



## AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : [ddoc-theses-contact@univ-lorraine.fr](mailto:ddoc-theses-contact@univ-lorraine.fr)

## LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

[http://www.cfcopies.com/V2/leg/leg\\_droi.php](http://www.cfcopies.com/V2/leg/leg_droi.php)

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Institut National Polytechnique de Lorraine

---

École Nationale Supérieure de Géologie de Nancy

École Doctorale RP2E

[H] 2003 CAUMON, G -

# Représentation, visualisation et modification de modèles volumiques pour les Géosciences

## THÈSE

Présentée et soutenue publiquement le 10 Mars 2003  
à l'École Nationale Supérieure de Géologie  
pour l'obtention du  
Doctorat de l'Institut National Polytechnique de Lorraine  
Spécialité Géosciences

par

Guillaume CAUMON

### Composition du jury

Président : Michel CARA

Rapporteurs : André JOURNEL  
Albert TARANTOLA

Examineurs : Jean-Pierre GRATIER  
Jean-Laurent MALLET  
Bruno LÉVY  
Charles H. SWORD, Jr.

---

Centre de Recherches Pétrographiques et Géochimiques  
Laboratoire d'Infographie et d'Analyse des Données  
Rue du Doyen Marcel Roubault – 54501 Vandoeuvre Cedex



# Remerciements

Avant tout, je tiens à remercier Jean-Laurent Mallet, mon directeur de thèse, qui, avec le projet Gocad, a su créer un cadre de travail aussi original que passionnant. Son enthousiasme et ses idées ont été un stimulant tout au long de ces trois années.

Cette thèse a été financée par la compagnie Chevron, devenue depuis ChevronTexaco. De ce fait, j'ai eu plaisir à de nombreuses discussions avec Chuck Sword. Ce travail doit beaucoup à ses commentaires pertinents et à ses indications précieuses.

Durant ces trois ans, la collaboration avec Bruno Lévy, de l'INRIA, m'a également beaucoup apporté. Ses conseils avisés et son dynamisme communicatif ont énormément contribué à ce travail.

Je remercie André Journal, de l'Université de Stanford et Albert Tarantola, de l'Institut de Physique du Globe de Paris, pour avoir accepté d'être rapporteurs de ce travail. Je suis également très honoré par la présence dans mon jury, de Michel Cara, directeur de l'École et Observatoire des Sciences de la Terre de Strasbourg, et de Jean-Pierre Gratier, directeur de l'Observatoire de Sciences de la Terre de Grenoble.

La deuxième salve de remerciements ira aux infatigables relecteurs de ce mémoire, qui ont grandement contribué à en améliorer la lisibilité. Un grand merci, donc, à Sophie pour ses abondantes et pertinentes notes, à Christian pour ses commentaires bienvenus, à Jean-Laurent pour ses corrections et conseils avisés, ainsi qu'à Fabrice et Laurent pour leurs remarques sur le Chapitre 1.

Plus généralement, le groupe de recherche Gocad, au sein duquel s'est effectué ce travail, constitue une grande famille que je tiens à remercier. En ordre aléatoire, merci et bonne continuation à ceux qui sont encore là, Laurent S., Laurent L., David, Luciano, Richard, François, Jean-Charles, Thomas, Christian, César, Emmanuel, Sophie, Pierre M., Pierre J., Bruno, Monique, Rémi, comme à ceux qui sont partis vers d'autres horizons, Arben, Stéphane, Isabelle, Magali, Bruno, Nathalie, Mathieu, Jérôme, Stefan, . . . Merci également aux T-Surfers (nouvellement Earth Decision Scientists) Fabien, Didier, Thierry (prompt rétablissement!), Nicolas, Joël, Olivier, Laurent, Matthieu, Étienne, Philippe H., David, Jean-Claude,

et à tous les autres qui ont contribué à la réussite du projet Gocad.

Un grand merci également à Monique, Élisabeth, Nicole, Armelle et Régine, pour leur soin et leur bonne humeur à régler tous nos problèmes pratiques.

Durant ces trois années, j'ai eu le plaisir de passer plusieurs mois en Californie et au Texas chez ChevronTexaco. Merci à Sébastien pour ses commentaires et suggestions sur ce travail, ainsi que pour les ballades Californiennes. Pour leur accueil, je remercie également Chuck, Amy, Cécile, Karen, Greg, Dee, Mike, João, Adwait, Connie, Jay, et bien sûr tous les joueurs de *soccer* de San Ramon.

À Nancy, merci à tous ceux qui m'ont entouré pendant ces trois années : étudiants de l'ENSG, chanteurs de la *Chorale U* et de *Sine Nomine*, musiciens des *Celtas Martos* et du *Big Bang*, vous avez souvent remplacé le soleil!

Enfin, je ne saurais terminer ces remerciements sans rendre hommage à mes parents, et à mon frère, que je sais toujours proches et attentionnés malgré la distance qui nous sépare. Leur présence et leur soutien sont une chance inestimable.

# Table des matières

Résumé . . . . .	9
Abstract . . . . .	10
<b>Introduction</b>	<b>11</b>
<b>1 Modélisation de formations stratifiées</b>	<b>15</b>
1.1 Introduction . . . . .	15
1.2 Problématique . . . . .	16
1.2.1 Données disponibles . . . . .	16
1.2.2 Cahier des charges . . . . .	17
1.3 Méthodes existantes de modélisation . . . . .	17
1.3.1 Modélisation génétique. . . . .	18
1.3.2 Modélisation inverse . . . . .	21
1.4 Utilisation de cartes de proportions . . . . .	24
1.4.1 Description de la méthode . . . . .	24
1.4.2 Limites de la méthode. . . . .	29
1.5 Visualisation de milieux stratifiés . . . . .	31
1.5.1 Utilisation de coupes . . . . .	31
1.5.2 Création d'une coupe . . . . .	33
1.6 Modifications sous contraintes . . . . .	35
1.6.1 Choix de la méthode . . . . .	35
1.6.2 Méthode mise en oeuvre . . . . .	38
1.6.3 Bilan et limites de la méthode proposée . . . . .	39
1.7 Application . . . . .	41
1.7.1 Présentation des données . . . . .	41
1.7.2 Application de la méthode . . . . .	42
1.8 Bilan et perspectives . . . . .	43
<b>2 Modèles volumiques définis par frontières.</b>	<b>45</b>
2.1 Introduction . . . . .	45
2.2 Enjeux d'une représentation par frontières . . . . .	46
2.2.1 Les b-reps : une représentation volumique parmi d'autres . . . . .	46
2.2.2 Applications en Géosciences . . . . .	47
2.2.3 Problématiques de recherche . . . . .	50
2.2.4 Bilan . . . . .	52
2.3 Structures de données utilisée . . . . .	54
2.3.1 État de l'art . . . . .	54
2.3.2 Représentation utilisée . . . . .	56
2.3.3 Extension de la structure : notion de contact . . . . .	56
2.4 Modification de modèles volumiques : état de l'art . . . . .	58

2.4.1	Opérations sur des modèles définis par frontières . . . . .	58
2.4.2	Déformation de modèles volumiques . . . . .	61
2.5	Modification de contacts dans une b-rep . . . . .	63
2.5.1	Approche proposée . . . . .	63
2.5.2	Définition d'une hiérarchie . . . . .	64
2.5.3	Définition de la zone d'influence . . . . .	67
2.5.4	Paramétrisation de la surface de glissement . . . . .	68
2.5.5	Modification interactive . . . . .	69
2.5.6	Retriangulation de la surface de glissement . . . . .	73
2.5.7	Gestion des changements de macro-topologie . . . . .	79
2.5.8	Mise à jour du modèle volumique . . . . .	83
2.5.9	Bilan . . . . .	83
2.6	Applications . . . . .	84
2.7	Conclusion . . . . .	85
<b>3</b>	<b>Modèles volumiques cellulaires</b>	<b>87</b>
3.1	Enjeux et Variété des Modèles Cellulaires . . . . .	88
3.1.1	Applications . . . . .	88
3.1.2	Taxonomie des modèles cellulaires . . . . .	89
3.2	Les méthodes de visualisation volumique . . . . .	90
3.2.1	Les techniques de visualisation . . . . .	91
3.2.2	Méthodes de calcul . . . . .	93
3.2.3	Méthodes d'optimisation . . . . .	95
3.3	Tranchage incrémental . . . . .	96
3.3.1	Principe . . . . .	96
3.3.2	Mise à jour des éléments actifs . . . . .	100
3.3.3	Calcul des polygones d'intersection . . . . .	101
3.3.4	Rendu à l'écran . . . . .	104
3.3.5	Bilan de la méthode . . . . .	105
3.4	Visualisation de grilles fortement hétérogènes . . . . .	106
3.4.1	Structure de données . . . . .	106
3.4.2	Construction . . . . .	107
3.4.3	Tranchage incrémental avec CIEL . . . . .	108
3.4.4	Résultats . . . . .	113
3.5	Visualisation de grilles faiblement hétérogènes . . . . .	115
3.5.1	Structure de données . . . . .	115
3.5.2	Construction . . . . .	118
3.5.3	Tranchage incrémental . . . . .	119
3.5.4	Limites de la méthode . . . . .	123
3.5.5	Résultats . . . . .	125
3.6	Grilles curvilinéaires . . . . .	126
3.6.1	Principe . . . . .	126
3.7	Comparaison des méthodes . . . . .	127
3.7.1	Comparaison sur des grilles curvilinéaires . . . . .	127
3.7.2	Perspectives . . . . .	128
3.8	Conclusion et perspectives . . . . .	128
	<b>Conclusion</b>	<b>131</b>
	<b>Bibliographie</b>	<b>150</b>

# Table des figures

1.1	Géométrie-type d'une séquence sédimentaire . . . . .	19
1.2	Principe de l'inversion stratigraphique . . . . .	22
1.3	Problématique et données disponibles . . . . .	24
1.4	Principe des cartes de proportions . . . . .	26
1.5	Contraintes au niveau des marqueurs de puits . . . . .	28
1.6	Prise en compte de failles syn- et post-sédimentaires (respectivement A et B). . . . .	29
1.7	Gestion des biseaux par la méthode . . . . .	31
1.8	Coupe dans un modèle stratifié . . . . .	32
1.9	Structure de données de coupe . . . . .	32
1.10	Raffinement d'une ligne par une surface triangulée . . . . .	36
1.11	Performances de la modification interactive . . . . .	40
1.12	Jeu de données du champ « <i>Sinusoidal</i> » . . . . .	41
1.13	Exemple d'utilisation de coupes dans un modèle stratifié. . . . .	42
1.14	Modèle cellulaire généré à partir du modèle par extrusion. . . . .	43
2.1	Représentation par frontières pour un modèle du sous-sol . . . . .	46
2.2	Principe des représentations par frontières . . . . .	47
2.3	Principe des SIG . . . . .	48
2.4	Validité d'un modèle géologique . . . . .	53
2.5	Principe des représentations combinatoires de b-reps . . . . .	55
2.6	Exemples de configurations non variété . . . . .	55
2.7	La structures d'arête radiale utilisée . . . . .	57
2.8	Notion de contact dans un modèle b-rep . . . . .	58
2.9	Exemples de modifications multi-résolution . . . . .	59
2.10	Suppression des cicatrices . . . . .	60
2.11	Découpage contraint de surfaces . . . . .	61
2.12	Exemple de déformation simple et contrainte . . . . .	62
2.13	Principe de la modification proposée . . . . .	63
2.14	Types de contacts et hiérarchie associée . . . . .	65
2.15	Ensembles de la région d'influence . . . . .	67
2.16	Paramétrisation d'une surface triangulée . . . . .	68
2.17	Mise en cohérence 3D du curseur . . . . .	72

2.18	Mise à jour de la surface de glissement . . . . .	74
2.19	Recherche des points co-localisés de $\mathcal{S}_g$ . . . . .	76
2.20	Respect d'arêtes dans une triangulation . . . . .	77
2.21	Problèmes d'appariements de noeuds . . . . .	78
2.22	Carte des coins . . . . .	80
2.23	Changements de topologie après la modification . . . . .	80
2.24	Structure simplifiée pour stocker un arrangement de segments . . . . .	81
2.25	Calcul d'intersections entre contacts . . . . .	83
2.26	Modification de contacts sur un modèle synthétique (1) . . . . .	84
2.27	Modification de contacts sur un modèle synthétique (2) . . . . .	85
3.1	Classification des représentations cellulaires. . . . .	90
3.2	Exemples de visualisation volumique . . . . .	91
3.3	Principe du tranchage incrémental . . . . .	97
3.4	Méthodes de calcul d'isovaleur . . . . .	98
3.5	Principe de l'algorithme . . . . .	100
3.6	Calcul combinatoire des polygones d'intersection . . . . .	103
3.7	La structure CIEL . . . . .	106
3.8	Les primitives de la structure CIEL : cellules polyédriques, demi-arêtes, et sommets. . . . .	107
3.9	Images générées avec la méthode CIEL : Grille 1 . . . . .	113
3.10	Images générées avec la méthode CIEL : Grille 2 . . . . .	114
3.11	Les primitives combinatoires d'un graphe cellulaire . . . . .	116
3.12	Table des cas de l'algorithme des <i>Marching Cubes</i> . . . . .	118
3.13	Structure de méta-cellule mise en place . . . . .	119
3.14	Propagation dans un graphe cellulaire lors du tranchage incrémental . . . . .	122
3.15	La grille stratigraphique faillée Mandaros, munie d'une propriété aléatoire lissée par DSI. . . . .	124
3.16	La grille stratigraphique faillée <i>FGrid</i> , rendue avec 700 tranches. . . . .	125
3.17	Primitive utilisée pour les graphes cellulaires structurés . . . . .	126
3.18	Le temps de calcul d'une image est proportionnel au nombre de polygones d'intersection . . . . .	128

## Résumé

La *modélisation volumique* du sous-sol a une grande importance applicative dans de nombreux domaines des géosciences tels que l'exploration et l'exploitation pétrolière, la géotechnique ou encore l'hydrogéologie. Cette thèse aborde trois des principaux défis de cette discipline :

- la *représentation informatique* des modèles volumiques doit être adaptée à la complexité des formations géologiques, ainsi qu'aux opérations effectuées sur les modèles du sous-sol ;
- la *visualisation* de ces modèles se doit de générer efficacement des images exploitables à partir de la totalité ou d'une partie du volume ;
- la *modification* de modèles volumiques, nécessaire dans de nombreuses applications, doit en particulier maintenir leur validité.

Dans un premier temps, ces trois aspects sont étudiés dans le cas de la modélisation d'architectures stratigraphiques.

Dans un deuxième temps, nous étudions deux représentations plus générales capables de traiter tous les types de formations géologiques. La première est une représentation *macroscopique* du sous-sol à base de surfaces triangulées (représentation par frontières), équivalente à une véritable carte géologique tridimensionnelle. La méthode proposée pour modifier ce type de modèle est décrite, ainsi que ses principaux intérêts en géosciences.

Générées à partir de ces macro-modèles, les grilles volumiques constituent la seconde représentation. Utilisées comme supports de calculs en ingénierie (méthodes d'éléments finis, géostatistique), ces grilles sont étudiées sous l'angle de la visualisation volumique. Pour cela, une méthode générique est décrite, puis déclinée aux différentes familles de grilles. La méthode permet d'explorer rapidement les grilles de plusieurs millions de cellules à l'aide de sections, de surfaces d'isovaleurs ou par rendu volumique.

## Abstract

Subsurface volume modeling has many geoscientific applications such as hydrocarbon exploration and production, geotechnics or hydrogeology. This thesis concerns three of the main challenges of volume modeling:

- *Computer representation* of solid models needs to account for complex geometries as encountered in geological formations, and to be appropriate to carry out operations on these models.
- *Visualization* of such models has to generate usable images efficiently of the whole or part of the model.
- *Modification* of solid models, needed in various applications, must maintain model validity.

In the first part, all three aspects above are studied for stratigraphic architecture modeling.

In the second part, this thesis deals more generally with two representations capable of handling two types of geological formations. The first one is a *macroscopic* subsurface description using triangulated surfaces, equivalent to a true solid geological map. For this type of representation, a real-time modification method is described, as well as its main applications in geosciences.

Generated from these macroscopic descriptions, three-dimensional grids make up the second general representation. Primarily used for finite elements or geostatistical computations, these grids are studied in a volume visualization perspective. Therefore, a generic visualization technique is described and declined to several types of grids. This method is convenient to explore efficiently volume data sets with several millions cells through sections, isosurfaces of direct volume rendering.

# Introduction

L'accroissement quantitatif des données d'observation depuis une trentaine d'années confère aux outils et méthodes informatiques une importance croissante dans le domaine des Géosciences. Ils apportent en effet, outre le stockage des informations sous une forme échangeable, une aide indispensable à l'interprétation et à l'intégration des données. Plus généralement, ils facilitent la modélisation et la visualisation des structures et des phénomènes de subsurface. La définition de tels outils et méthodes informatiques est l'objet de la *Géomodélisation*.

## Contexte d'étude

Le travail présenté dans ce mémoire, financé par la compagnie pétrolière CHEVRONTEXACO, a été réalisé au sein du projet Gocad, dirigé par le Professeur Jean-Laurent MALLET.

Ce projet entre dans le cadre de la Géomodélisation, qui regroupe l'ensemble des méthodes informatiques qui permettent de décrire le sous-sol à partir des informations disponibles. Pour cela, il s'agit principalement d'identifier les hétérogénéités de la subsurface par la représentation de *l'arrangement relatif des roches* d'une part et la *caractérisation de leurs propriétés physiques* d'autre part.

De manière simplifiée, les principales opérations de la Géomodélisation s'ordonnent de la manière suivante

1. traitement des données d'acquisition ;
2. intégration de ces données pour former un modèle surfacique constitué d'interfaces géologiques (failles et fractures, limites de couches, etc.) ;
3. définition d'une représentation volumique macroscopique continue à partir de ces interfaces ;
4. discrétisation en grilles de l'espace ainsi défini ;
5. calculs de propriétés physiques sur ces grilles ;
6. simulations d'écoulements, calculs de compaction ou de fracturation, etc.

Le travail présenté dans ce mémoire a trait plus particulièrement aux étapes 3 et 4, qui relèvent de la *modélisation volumique* du sous-sol.

## Problématiques et contributions

La modélisation volumique [Mäntylä, 1988] concerne la représentation informatique d'objets solides, ainsi que la construction et les opérations effectuées sur ces objets. Cette étude aborde sous trois angles la modélisation volumique dans le cadre des Géosciences.

Premièrement, la *représentation informatique* constitue la pierre fondatrice des méthodes volumiques. Elle détermine en effet les catégories d'objets et les types d'opérations réalisables par le système de modélisation. Elle est de plus la clé de l'efficacité, en temps et en mémoire, des différentes applications. Cette étude s'intéresse à deux représentations à base de surfaces triangulées : l'une permet de définir par extrusion la géométrie de formations stratifiées, alors que l'autre, plus générale, permet de traiter tous les types de géométries rencontrés en Géologie (réseaux de failles complexes, limites de couches évaporitiques, plis en fourreau, etc.). Pour cette dernière, deux conditions nécessaires de validité géologique sont définies. Différentes structures de représentations de grilles sont également présentées.

Le deuxième aspect concerne la *visualisation* des modèles volumiques, indispensable à leur bonne compréhension. Elle permet également par un contrôle visuel de vérifier la cohérence des modèles. Dans leur mise en oeuvre, les méthodes de visualisation doivent donc permettre d'appréhender la nature tridimensionnelle des objets. Pour cela, il est important [Möller *et al.*, 1999] que la génération d'une image soit suffisamment efficace pour permettre des changements de points de vue fluides. Pour la visualisation volumique de grilles, la rapidité des accès à l'information et la rapidité des traitements sont donc particulièrement critiques. À cet effet, ce travail propose un nouvel algorithme de visualisation de complexité optimale.

Le troisième aspect abordé dans cette thèse concerne la *modification* de modèles volumiques existants. Suite au comportement anormal d'un algorithme ou au manque de données, la visualisation de ces modèles peut en effet mettre en lumière des incohérences nécessitant les corrections manuelles d'un expert. En outre, les données d'acquisition fournissent du sous-sol une vision empreinte d'incertitudes. Des méthodes de calcul appliquées à un modèle volumique peuvent ainsi demander des réajustements géométriques, ce qui requiert habituellement une reconstruction complète du modèle. Dans ce mémoire, nous présentons une approche plus performante pour modifier interactivement des modèles volumiques, tout en préservant un certain nombre de contraintes géologiques.

## Organisation du mémoire

Ce mémoire s'organise autour de trois familles de représentations volumiques utilisées en Géosciences.

Le Chapitre 1 est consacré à la modélisation d'architectures stratigraphiques, particulièrement intéressante pour l'industrie pétrolière. Pour cela, la représentation par extrusion proposée par Mallet [2002] est utilisée. Nous montrons comment l'usage de coupes graphiques, lié à un système de modification sous contraintes permet à un spécialiste d'incorporer intuitivement sa connaissance au modèle géométrique.

Le Chapitre 2 se place dans un cadre plus large pour s'intéresser à une représentation volumique par frontières, équivalente à une carte géologique tridimensionnelle. Une définition des conditions nécessaires de validité donne lieu à une nouvelle approche de modifications contrainte de ce type de modèle. Contrairement aux approches existantes, la méthode définie ne nécessite pas de reconstruction complète du modèle, et permet donc de maintenir les attributs de la représentation du sous-sol (nombre et nom des régions, définitions des propriétés, etc.).

Pour de nombreuses applications qui nécessitent la résolution d'équations aux dérivées partielles, l'espace doit être discrétisé en grilles. Dans ces méthodes, la prise en compte des hétérogénéités du sous-sol s'appuie sur des grilles non structurées conformes à la géométrie des objets géologiques. Dans le Chapitre 3, nous présentons une méthode de visualisation efficace et polyvalente pour ce type de représentation. Cette méthode générique est adaptée à plusieurs types de grilles, depuis les plus générales aux plus spécifiques.



# Chapitre 1

## Modélisation de formations stratifiées

### 1.1 Introduction

Sur de nombreux affleurements de roches sédimentaires, il est possible de distinguer nettement une succession verticale de bancs de nature et d'épaisseur variables. L'observation de ces formations, dites *formations stratifiées* et de leur éventuel contenu en fossiles a permis l'élaboration des premières théories géologiques. Ainsi, dès la fin du XVII<sup>e</sup> Siècle, Steno [1669] établit les trois principes suivants :

- les couches plus récentes se déposent au-dessus des couches antérieures (*principe de superposition*);
- au moment du dépôt, les couches sont continues latéralement et sub-horizontales;
- les variations par rapport à ce schéma sont explicables par des événements postérieurs au dépôt.

Au XIX<sup>e</sup> Siècle, Lyell [1833] montre que les processus observés actuellement sont transposables dans le passé pour expliquer les formations géologiques (principe de l'*uniformitarisme*). Riche de ces théories et des études paléontologiques, la *stratigraphie*, étymologiquement « description des strates <sup>1</sup> », a ainsi permis d'établir une chronologie des formations.

Derrière l'apparente simplicité de ces principes, les formations stratifiées présentent une grande variabilité, du fait de la diversité des processus agissant sur la sédimentation et l'évolution des roches en profondeur [Cojan *et al.*, 1999 ; Biju-Duval, 1999]. De plus, elles peuvent avoir été affectées par des déformations ou des ruptures tectoniques après leur dépôt. Parallèlement, ces formations ont une grande importance industrielle, dans les domaines des ressources naturelles

---

<sup>1</sup> Dans une définition plus actuelle, la stratigraphie « étudie l'agencement, dans l'espace et dans le temps, des formations géologiques et des événements qu'elles matérialisent, afin de reconstituer l'histoire de la Terre et son évolution en fonction du temps » [Rey, 1997].

(pétrole, gaz, minerais, carrières), et de l'environnement (stockage de déchets, géotechnique, hydrogéologie). Dans ce cadre, il est primordial, à partir des informations disponibles sur une zone du sous-sol, de caractériser la *géométrie* des strates (*architecture stratigraphique*), ainsi que leur *lithologie* (propriété pétro-physiques).

Ce chapitre s'intéresse plus particulièrement à la modélisation d'architectures stratigraphiques. Après un bref exposé des types de données disponibles et des enjeux de la modélisation, nous passerons en revue les méthodes existantes de modélisation. Nous reviendrons ensuite sur la méthode adoptée par Mallet [2002], avant de montrer comment les méthodes de visualisation et de modification mises en oeuvre dans cette thèse permettent de contrôler et d'assurer une cohérence géologique des modèles générés.

## Terminologie

La description volumique de formations stratifiées s'appuie généralement sur la définition d'*unités lithostratigraphiques*, ensembles de lithologie relativement homogène dont les limites peuvent être reconnues [Rey, 1997]. Ces limites entre unités lithostratigraphiques, seront particulièrement importantes pour la représentation informatique des volumes rocheux. Comme cela est courant en géophysique, nous emploierons par la suite le terme d'*horizon* pour désigner ces discontinuités.

## 1.2 Problématique

### 1.2.1 Données disponibles

Qu'elles s'intéressent aux formations stratifiées ou non, les méthodes d'investigation du sous-sol, donnent lieu principalement à [Cojan *et al.*, 1999] :

- des données géophysiques, en particulier de sismique réflexion ;
- des données de sondage directes (cas de sondages carrotés) ou indirectes (enregistrements diagraphiques dans le puits).

Ces données de subsurface, couramment utilisées dans l'exploration et la production pétrolières, présentent des résolutions différentes. En effet, la résolution verticale de données sismiques est généralement de l'ordre de la dizaine de mètres [Henry, 1997], alors que les forages permettent un échantillonnage plus fin des mesures (résolution allant du millimètre pour un sondage carroté à la dizaine de centimètres pour les diagraphies). La précision des données de puits est également meilleure que celle du signal sismique, car celles-ci sont mesurées directement au niveau des formations traversées.

En revanche, si les images sismiques permettent d'appréhender l'architecture des formations par une couverture régulière de l'espace [Vail *et al.*, 1977], les données de forage sont ponctuelles et éparses. Elles comprennent tout au plus une mesure locale de pendage des marqueurs géologiques recoupés par le puits (pendagemétrie).

### 1.2.2 Cahier des charges

À partir des données de subsurface, nous voulons définir un système de modélisation quantitative d'architectures stratigraphiques. Les spécifications souhaitables d'un tel système sont à présent énoncées.

1. Le système de modélisation utilise une représentation volumique adéquate.
2. Cette représentation permet la prise en compte de géométries perturbées par les déformations et ruptures tectoniques survenues dans les roches entre le moment du dépôt et l'époque actuelle. En particulier, nous souhaitons traiter les failles inverses et les plis couchés.
3. Cette représentation est compacte en mémoire et facilement consultable.
4. Dans la mesure du possible, le système intègre les connaissances géologiques, c'est-à-dire les principes théoriques et les informations qualitatives.
5. Il se place dans la perspective d'une modélisation de faciès et de propriétés pétrophysiques des formations représentées. Cette dernière sera en effet essentielle dans l'évaluation de ressources disponibles dans un domaine du sous-sol.
6. Dans cette optique, le modèle de subsurface obtenu est suffisamment flexible pour obtenir facilement des représentations spécifiques à certaines applications.
7. Le système offre un traitement adapté des données de subsurface, en tenant compte en particulier de la hiérarchie des données relative à leur résolution variable.
8. Il permet d'évaluer les incertitudes relatives à un modèle du sous-sol.
9. Il permet la mise à jour rapide du modèle lorsque de nouvelles informations quantitatives ou qualitatives sont disponibles.

## 1.3 Méthodes existantes de modélisation

Les techniques existantes de modélisation numérique des séquences stratigraphiques se divisent en deux catégories complémentaires [Miall, 1997 ; Emery *et al.*, 1996 ; Watney *et al.*, 1999] :

- modélisation génétique (*forward modeling*) : un modèle simplifié synthétique est construit à partir d'un ensemble de conditions de dépôt et de règles empiriques.

- modélisation inverse : ces méthodes mettent l'accent sur les données disponibles dans une zone pour construire le modèle. La connaissance des facteurs géologiques liés à la mise en place et à l'évolution des formations n'est pas nécessaire. Dans cette catégorie, on distingue l'approche déterministe, où l'on cherche à représenter au mieux la réalité, de l'approche stochastique, où l'on génère des réalisations équiprobables d'un même modèle afin d'envisager plusieurs éventualités.

### 1.3.1 Modélisation génétique.

#### Contexte : La stratigraphie séquentielle

La stratigraphie séquentielle [Homewood *et al.*, 1999 ; Sloss, 1988 ; Miall, 1997 ; Emery *et al.*, 1996] définit un ensemble de concepts afin de mieux caractériser les formations stratifiées. En reliant les environnements de dépôts aux variations relatives du niveau marin, cette discipline a énormément modifié la façon d'approcher et de comprendre les formations sédimentaires depuis une cinquantaine d'années. Le principe fondamental est le suivant : les dépôts s'organisent en séquences continues, séparées par des discordances. Les séquences correspondent au passage d'un environnement de dépôt (par exemple plaine côtière) à un autre (offshore profond) par variation du niveau marin relatif (une transgression dans notre exemple).

L'utilisation de données sismiques, introduite par le groupe d'Exxon [Vail *et al.*, 1977 ; Vail, 1987 ; Wagoner *et al.*, 1987] a donné un nouveau souffle à la stratigraphie séquentielle. L'idée de ce groupe de chercheurs a été d'associer les réflecteurs sismiques à des surfaces remarquables correspondant à des variations de l'espace disponible pour la sédimentation (ou accommodation [Jervey, 1988]). La stratigraphie sismique permet en particulier d'identifier des discordances, des surfaces d'inondation [Galloway, 1989], ainsi que les terminaisons des couches, qui sont directement corrélables aux variations relatives du niveau marin. On distingue en particulier (FIG. 1.1) :

- les figures de rétrogradation, ou *onlap*, qui correspondent à une remontée du niveau de base.
- les figures de progradation, ou *downlap*, qui sont liées à une baisse du niveau de base. Les régressions marines se traduisent également par des variations latérales des dépôts en haut des formations (*offlap*). Les formations émergées peuvent alors être érodées, donnant lieu à des hiatus sédimentaires.

En mettant en relation un certain nombre de facteurs (environnement de dépôt, taux de sédimentation, eustatisme, subsidence, etc.), la stratigraphie séquentielle permet d'élaborer des modèles simplifiés d'évolution de bassins sédimentaires. On parle de modèles génétiques ou basés sur les processus. Nous allons à présent donner un aperçu de ces méthodes, afin de mettre en perspective l'approche inverse développée durant cette thèse.

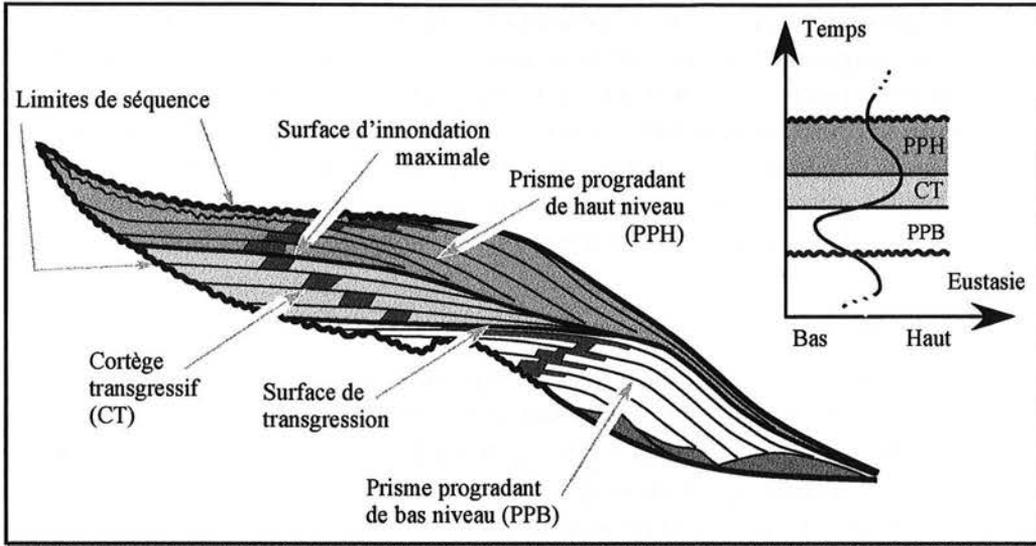


FIG. 1.1 – Géométrie type d'une séquence sédimentaires sur une plate-forme continentale (d'après [Homewood *et al.*, 1999]). Les architectures stratigraphiques enregistrent les variations du niveau marin (la subsidence est supposée constante).

### Modèles utilisant des bilans de matière

Dans l'un des premiers modèles conceptuel de sédimentation, introduit par Sloss [1962], l'agencement des sédiments est déterminée par le volume, la nature et le taux de dispersion de sédiments, ainsi que par l'espace disponible. Ces considérations ont permis l'élaboration d'un des premiers modèles numériques simplifiés en coupe [Harbaugh *et al.*, 1970] (décrit aussi dans [Emery *et al.*, 1996, p. 260]). Dans ce modèle, le flux de sédiments se propage sur une grille 2D ; chaque colonne de la grille subit une sédimentation fonction de l'espace d'accommodation ; le flux sortant d'une colonne est déterminé par conservation de la masse.

Le modèle de TURCOTTE et WILLEMANN [Turcotte *et al.*, 1983] permet quant à lui de prendre en compte les processus d'érosion comme une fonction de l'espace. Le flux sédimentaire varie aussi en fonction de la position horizontale, et le taux de subsidence est uniforme. Dans ce modèle, seules les variations eustatiques sont donc variables en fonction du temps.

Vers la fin des années 80, des approches plus flexibles permettent de mieux prendre en compte les paramètres, et de moduler leur influence dans le temps et dans l'espace. Cela se traduit par l'utilisation de courbes de variations eustatiques en fonction du temps, ainsi que de courbes donnant la géométrie initiale, le taux de sédimentation et le taux de subsidence du centre à la bordure du bassin sédimentaire [Burton *et al.*, 1987 ; Jervey, 1988 ; Reynolds *et al.*, 1991]. D'autres paramètres sont également pris en compte, comme la compaction et la flexure

lithosphériques liées à la charge sédimentaire [Reynolds *et al.*, 1991].

Ces techniques, conçues en premier lieu pour modéliser des dépôts détritiques, ne permettent pas de générer des modèles carbonatés réalistes, car aucun ne prend en compte la croissance récifale. Ainsi, des modèles de genèse de formations carbonatées ont également été définis [Lerche *et al.*, 1987 ; Bice, 1989 ; Demicco *et al.*, 1989], qui ajoutent aux paramètres vus précédemment des facteurs biotiques comme l'ensoleillement, la température, la salinité, etc.

### Modèles physiques

Les *méthodes physiques* mettent l'accent sur le transport des sédiments. Par exemple, JORDAN et FLEMINGS [Jordan *et al.*, 1991] utilisent une loi de diffusion simplifiée de la forme :  $\frac{\partial h}{\partial t} = K \cdot \frac{\partial^2 h}{\partial x^2}$ , où  $h$  est l'altitude du profil de dépôt,  $t$  le temps,  $x$  la position horizontale et  $K$  le coefficient de transport. Cette formulation, permet de traiter le dépôt et l'érosion de manière unifiée aussi bien sur les marges passives ( $K = 1000 \text{ m}^2 / \text{an}$ ) que dans des bassins d'avant-pays ( $K = 40\,000 \text{ m}^2 / \text{an}$ ).

### Modèles utilisant la logique floue

Plus récemment, Nordlund [1996, 1999] décrit un système de modélisation utilisant la *logique floue*. Plusieurs variables (profondeur, pente, taux de sédimentation, etc.) permettent de définir des ensembles flous à l'aide de fonctions d'appartenance (*membership function*). Par exemple, l'ensemble « domaine d'action de vagues » est caractérisé par une fonction linéaire par morceaux de la profondeur. Ces ensembles sont reliés par des lois comme : « si milieu profond alors peu de dépôts et dépôts fins ». La combinaison de plusieurs lois permet de définir un ensemble flou multivariable (*compound fuzzy set*). Les fonctions d'appartenance de ce nouvel ensemble permettent d'extraire les caractéristiques du modèle en tout point (pourcentage d'argile, épaisseur, etc.) Le système, en offrant la possibilité de modifier les règles, les ensembles et leurs fonctions d'appartenance, permet d'étudier facilement l'influence des différents facteurs sur la géométrie et la lithologie des séries sédimentaires. Il peut également s'adapter à différents types d'environnements, pour peu que les règles soient définies avec soin. Cependant, la prise en compte d'informations uniquement qualitatives (les lois) implique une solution qualitative par nature.

### Modèles intégrés

Les systèmes de modélisation génétique récents, comme SEDPACK [Kendall *et al.*, 1991, 1993 ; Csato *et al.*, 2002], SEDSIM [Tetzlaff *et al.*, 1989 ; Dyt *et al.*, 2001] ou PHILL [Bowman *et al.*, 1999], combinent dépôts détritiques et carbonatés, et offrent de nombreuses possibilités pour intégrer les différents paramètres.

Ils gèrent ainsi flexure continentale, compaction, subsidence différentielle, variations eustatiques, érosion, sources multiples de sédiments, etc.

### Apports et limites des modèles génétiques

Quelle que soit l'approche envisagée (conservation de la masse, hydrodynamique, logique floue), les modèles génétiques ont pour principale fonction de *tester des concepts*. Ils ont ainsi grandement contribué à l'évolution de la théorie stratigraphique au cours de ces vingt dernières années [Homewood *et al.*, 1999].

D'un point de vue pratique, les méthodes génétiques ont également de nombreuses applications. Par exemple [Kendall *et al.*, 1999], des réflecteurs sismiques ont été datés en les corrélant à un modèle synthétique construit à l'aide d'un cycle eustatique global [Haq *et al.*, 1987].

Du point de vue des représentations utilisées, la majorité de modèles génétiques s'appuie sur des grilles régulières. Ainsi, la représentation des failles normales en 2D [Kendall *et al.*, 1991], n'est possible qu'au prix d'un échantillonnage horizontal très dense, difficilement réalisable dans le cas de modèles tridimensionnels. De plus, les grilles régulières ou curvilinéaires utilisées par la plupart des méthodes génétiques permettent difficilement de gérer les failles inverses. Cette restriction liée à la représentation touche à une autre limite intrinsèque aux méthodes génétiques : elles se concentrent sur les processus de dépôt, et ne font pas de la prise en compte de données actuelles leur préoccupation principale. Le conditionnement de méthodes génétiques à des données est ainsi rendu délicat par les déformations tectoniques qui peuvent avoir eu lieu entre le moment du dépôt et l'époque actuelle. Contraindre un processus à honorer des données peut en outre introduire un biais dans les modèles générés. Enfin, l'ajustement manuel des paramètres d'entrée est fastidieux et permet rarement d'atteindre une précision convenable [Cross *et al.*, 1999].

De plus, d'un point de vue fondamental, la génération d'architectures sédimentaires est considérée par beaucoup comme non univoque : des causes différentes peuvent avoir les mêmes effets. Ainsi, la concordance d'un modèle synthétique avec des données de subsurface ne garantit pas forcément que les paramètres d'entrée de ce modèle correspondent à la réalité [Burton *et al.*, 1987].

### 1.3.2 Modélisation inverse

Dans le domaine de la modélisation stratigraphique [Watney *et al.*, 1999], le terme d'*inversion* est généralement employé pour parler de l'ajustement automatique des paramètres d'entrée d'un modèle génétique en fonction des données d'observation (FIG. 1.2). Pour cela, la théorie mathématique de l'inversion est utilisée [Tarantola, 1987]. Cette méthode apparaît ainsi très intéressante par rap-

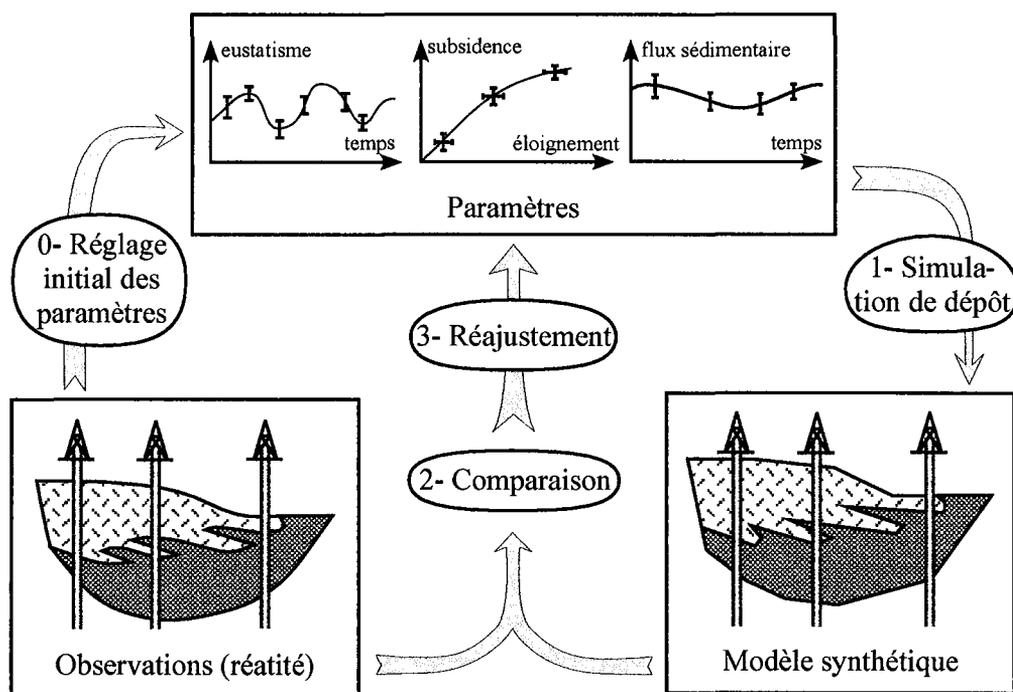


FIG. 1.2 – Principe de l'inversion stratigraphique (d'après [Cross *et al.*, 1999]). Après un évaluation initiale des paramètres stratigraphiques à partir des données d'observation (0), un premier modèle synthétique est créé (1). La boucle d'inversion peut alors commencer : une comparaison du modèle synthétique avec les données d'observation (2) permet de réajuster les paramètres (3) ; un nouveau modèle synthétique est alors généré (1). Le système inverse s'arrête lorsque les différences observées sont jugées suffisamment faibles.

port à un ajustement manuel des paramètres [Cross *et al.*, 1999] : plus rapide, elle permet également une quantification des erreurs commises par les estimations.

Cependant, les observations selon lesquelles des causes différentes produisent les mêmes effets [Burton *et al.*, 1987 ; Galloway, 1989] rendent *a priori* les méthodes inverses inapplicables à la modélisation de formations stratifiées. À partir de données synthétiques, CROSS et LESSENGER [Cross *et al.*, 1999, 2002] argumentent cependant que l'influence de chaque paramètre est séparable si l'épaisseur des cortèges sédimentaires est prise en compte par le modèle génétique. Ils définissent ainsi un système d'inversion stratigraphique, en imposant d'autres contraintes moins controversées (en particulier : qualité et la quantité de données disponibles, comportement déterministe du modèle génétique, conservation de la masse [Cross *et al.*, 2002]).

Par rapport à un ajustement manuel des paramètres, l'inversion stratigraphique permet de réduire considérablement le temps nécessaire à l'obtention d'un modèle synthétique. Ce temps reste cependant de l'ordre de la centaine d'heures

pour un modèle en coupe [Cross *et al.*, 1999]. Les performances d'une résolution de problème inverse en trois dimensions semblent donc encore prohibitives.

L'inversion stratigraphique, si elle permet de se rapprocher au plus près des observations, ne permet généralement pas une concordance exacte du modèle synthétique avec les mesures. En revanche, elle produit en général une marge d'erreur sur les paramètres d'entrée, qui peuvent servir à générer plusieurs réalisations [Cross *et al.*, 1999]. On entre alors dans une approche probabiliste.

Quoi qu'il en soit, le non-respect relatif des données d'acquisition reste une limite de l'inversion stratigraphique.

Dans cette thèse, nous considérons une définition plus large des méthodes inverses, qui ne s'appuient pas forcément sur un modèle génétique. Dans ce sens, la construction d'un modèle du sous-sol revient avant tout à interpoler les données disponibles.

Les méthodes géostatistiques [Deutsch *et al.*, 1997], surtout utilisées pour estimer des propriétés physiques peuvent être appliquées pour interpoler des géométries. Ainsi, Bourquin *et al.* [1997] utilisent un krigeage 2D sur une grille régulière pour produire des cartes isopaques.

Duan *et al.* [1999] proposent une simulation de paraséquences basée sur des méthodes syntaxiques. Dans ce système, les variations horizontales et verticales de lithofaciès sont établies par des règles grammaticales. La grammaire dérive directement des concepts de la stratigraphie séquentielle, par décomposition d'une paraséquence type en cortèges sédimentaires. Cette grammaire est par la suite utilisée dans les simulations conditionnelles. Ainsi, la grammaire joue conceptuellement le même rôle que les images d'entraînement en géostatistique à points multiples. Cette approche originale paraît très extensible ; cependant, la définition de la grammaire, qui dépend de l'environnement de dépôt est difficile à établir.

Goff [2000] propose une méthodologie pour simuler les architectures possibles d'une séquence sédimentaire. La méthode s'inscrit dans les simulations conjointes en domaine spectral, introduites dans [Gutjar *et al.*, 1997]. La géométrie d'un horizon  $k$ , approximée par une analyse spectrale par transformée de Fourier, est simulée à partir de l'horizon supérieur  $k - 1$ . Cette simulation est principalement conditionnée par un taux de conformité qui définit la variabilité de l'épaisseur verticale. La méthode permet également de prendre en compte des données sismiques ou des données de forage. Cependant, elle est mal adaptée pour gérer :

- les degrés de confiance variables des informations de subsurface ;
- les formations faillées ;
- les surfaces multivaluées suivant l'axe vertical (suite à un pli couché ou à un jeu de faille inverse).

Nous allons à présent décrire la méthode inverse des cartes de proportions [Mallet, 2002], sur laquelle repose notre travail.

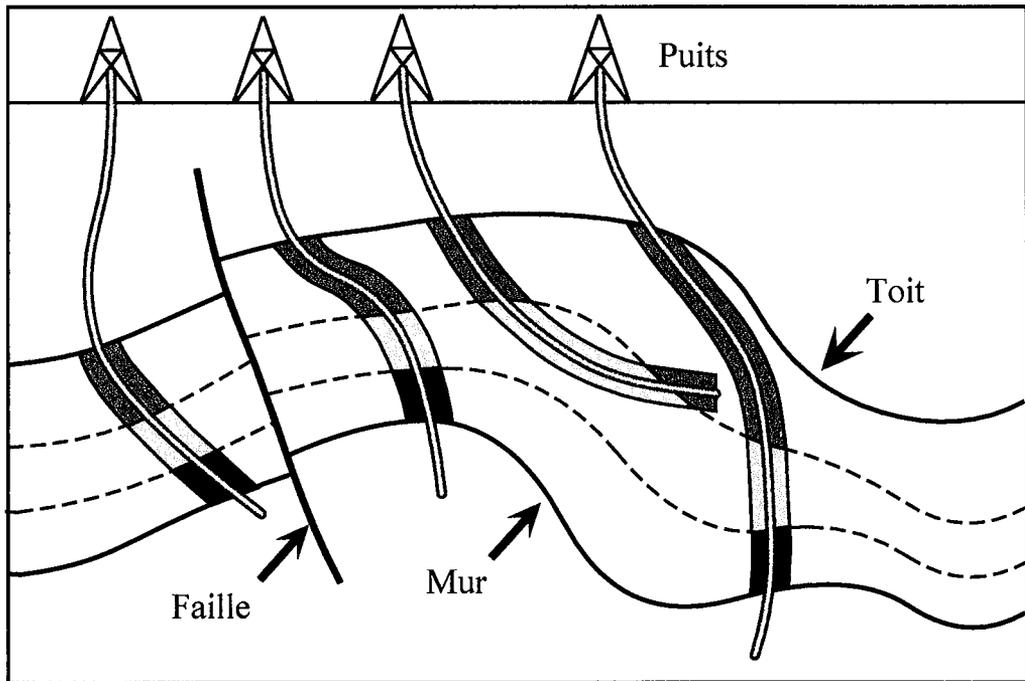


FIG. 1.3 – Problématique et données disponibles (vue en coupe). À partir d'horizons et de failles visibles sur une section sismique (en noir gras), le but est de déterminer une géométrie possible des couches intermédiaires non visibles sur la sismique (en pointillés), honorant les données de puits.

## 1.4 Utilisation de cartes de proportions

### 1.4.1 Description de la méthode

La méthode des cartes de proportion [Mallet, 2002, Chap. 8] s'appuie directement sur les données généralement disponibles dans les réservoirs pétroliers, à savoir pointés sismiques et marqueurs de puits (FIG. 1.3). Cette méthode, qui s'intéresse à la représentation d'architectures stratigraphiques, se situe donc résolument dans le cadre de la *stratigraphie géométrique* [Rey, 1997].

Cette partie rappelle le but et le fonctionnement des cartes de proportions présentées dans [Mallet, 2002].

#### Problématique

Dans les méthodes classiques de construction de modèles stratifiés, des pointés sismiques sont utilisés pour construire des surfaces représentant les horizons (voir par exemple [Duvina, 2000]). Ces surfaces sont ensuite modifiées pour tenir compte des informations disponibles aux puits (*marqueurs* donnant ponctuellement la position de l'horizon et éventuellement son pendage). Cependant,

dans le cas de couches de faible épaisseur relativement à leur extension horizontale, de telles modifications, sont particulièrement difficiles à gérer. En effet, des intersections entre horizons peuvent apparaître lorsqu'on les contraint à honorer les marqueurs de puits. Les méthodes de détection d'interférences [Moore *et al.*, 1988 ; Bergen, 1997] sont souvent coûteuses à mettre en place, car elles sont soumises aux problèmes de discrétisation des surfaces et à la précision limitée des ordinateurs.

### Représentation et méthodologie

Afin d'éviter les problèmes d'interférences, Mallet [2002] propose de représenter dans un même objet un ensemble de couches globalement conformes, éventuellement faillées. Cette représentation s'appuie sur une surface triangulée  $\mathcal{S}$ , et sur un champ de vecteurs défini en tout point de  $\mathcal{S}$  (FIG. 1.4). Ainsi, tout point  $\mathbf{x}(x, y, z)$  du volume d'intérêt  $\mathcal{V}$  peut être *projeté* sur  $\mathcal{S}$  dans la direction du champ de vecteurs. Réciproquement, l'opération d'*extrusion* permet, à partir d'un point  $\mathbf{x}_S \in \mathcal{S}$  et d'une distance à  $\mathcal{S}$  suivant cette direction d'obtenir un point dans le domaine  $\mathcal{V}$  (on parlera donc de *vecteurs d'extrusion*). En pratique, le champ de vecteurs est choisi de sorte à donner l'amplitude verticale de la formation, ce qui permet d'utiliser une distance normalisée entre 0 et 1, ou *proportion*. Il est ainsi très facile de représenter une famille de surfaces de même topologie en définissant des cartes de proportions sur la surface de référence.

En termes plus formels, soit  $k$  fonctions de proportion  $\{p_1, \dots, p_k\}$  telles que <sup>2</sup> :

$$\forall \mathbf{x} \in \mathcal{S}, \quad p_i(\mathbf{x}) \in [0, 1], \quad \forall i \in \{1, \dots, k\} \quad (1.1)$$

Par extension, on définit également en tout point  $\mathbf{x}$  de  $\mathcal{S}$  les proportions  $p_0(\mathbf{x}) = 0$  et  $p_{k+1}(\mathbf{x}) = 1$ . Ainsi, la surface entre deux couches successives  $i$  et  $i + 1$ ,  $i \in [1, k]$  est définie à l'aide d'une fonction d'extrusion  $\mathbf{extr}$  selon :

$$\mathbf{x}_i(\mathbf{x}) = \mathbf{x} + p_i(\mathbf{x}) \cdot \mathbf{extr}(\mathbf{x}), \forall \mathbf{x} \in \mathcal{S} \quad (1.2)$$

Centrée autour de la surface de référence, cette représentation permet de vérifier facilement une stratigraphie géométriquement cohérente dans le processus de modélisation :

$$\forall i \in [1, k] \quad \text{et} \quad \forall \mathbf{x} \in \mathcal{S}, \quad 0 \leq p_i(\mathbf{x}) \leq p_{i+1}(\mathbf{x}) \leq 1 \quad (1.3)$$

D'un point de vue informatique, elle est très compacte en mémoire, car elle factorise les relations combinatoires de  $k$  surfaces à la seule surface de référence. De plus, l'utilisation de proportions normalisées entre 0 et 1 est plus stable numériquement que des épaisseurs. Signalons également qu'il n'y a pas d'équivalence directe entre proportion et épaisseur des couches. En effet (FIG. 1.4), le champ

---

<sup>2</sup> En pratique, ces fonctions sont estimées de façon univoque par interpolation trilineaire entre les valeurs aux noeuds de la surface de référence.

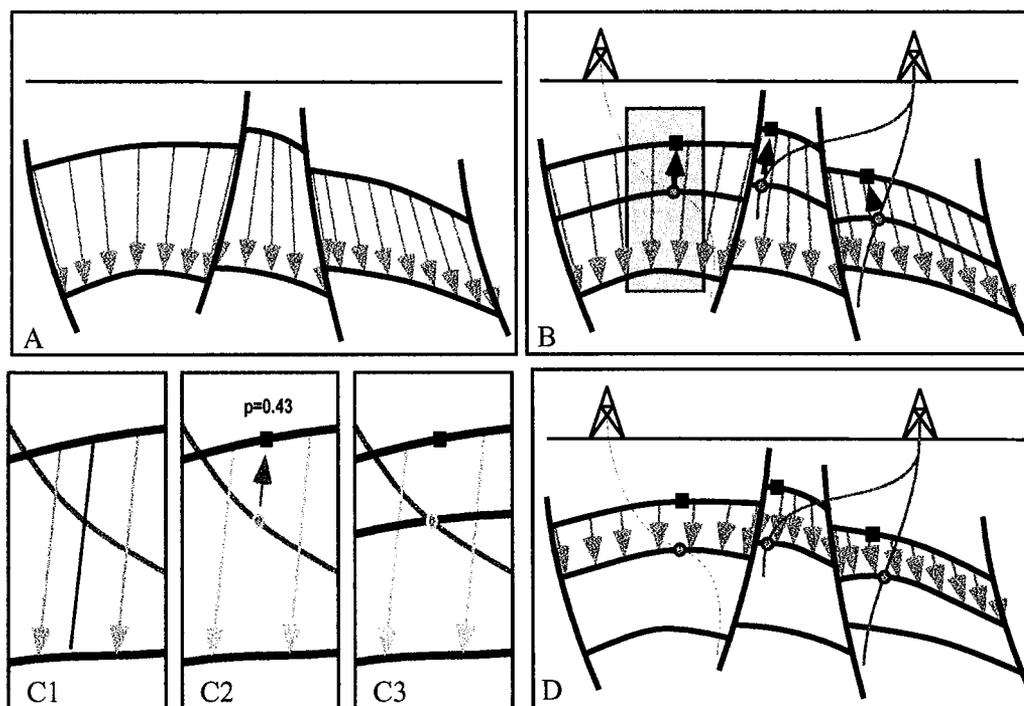


FIG. 1.4 – Représentation implicite des couches haute résolution à l'aide de cartes de proportion (vue en coupe). Un vecteur d'extrusion et un ensemble de proportions sont stockés en chaque noeud de la surface triangulée de référence  $S$  (A). Les proportions sont calculées à partir de l'image des marqueurs de puits sur  $S$ , puis interpolées (B et C). Les horizons intermédiaires peuvent ensuite être construits par simple translation de  $S$  le long de ses vecteurs d'extrusion (D).

de vecteurs  $\text{extr}$ , n'est pas nécessairement orthogonal à la stratigraphie. Les proportions sont donc reliées à l'épaisseur *apparente* de la formation le long des vecteurs.

Contrairement aux représentations classiques par cartes 2D, elle permet de prendre en compte des formations affectées par des failles inverses. Cependant, elle rend délicat le traitement de failles qui ne traversent pas la formation de part en part [Souche, 2003]. Toutes les limites entre strates partagent en effet la même topologie (celle de la surface de référence), et des cicatrices existeront donc même si le rejet est nul.

D'un point de vue géologique, ce type de représentation peut par exemple convenir à la modélisation de paraséquences, « succession relativement conforme de couches [...] délimitées par des surfaces d'inondation » [Wagoner *et al.*, 1987]. Plus généralement, la méthode permet de modéliser des couches non visibles sur la sismique, soit parce qu'elles ont une épaisseur trop faible (séquences des 4<sup>e</sup> et 5<sup>e</sup> ordres), soit parce que les contrastes d'impédance acoustique entre elles sont

trop faibles. De plus, par son attachement aux données et à des contraintes géométriques, elle est très générale et peut s'appliquer indifféremment à de nombreux environnements de dépôt.

En pratique, les cartes de proportions définies par Mallet [2002] s'utilisent de la manière suivante :

1. détection et construction de failles et de réflecteurs bien identifiés sur la sismique. Un réflecteur peut correspondre en particulier à une surface d'inondation, une discordance, etc. [Vail *et al.*, 1977] ;
2. pour deux surfaces successives  $\mathcal{S}_1$  et  $\mathcal{S}_2$ , calcul des vecteurs d'extrusion définissant le volume d'intérêt ;
3. initialisation et calculs des cartes de proportions entre  $\mathcal{S}_1$  et  $\mathcal{S}_2$  à partir des données aux puits ;
4. éventuellement, génération d'une grille stratigraphique Lévy [1999] pour réaliser des études de réservoir.

La première étape est fondamentale pour la suite du processus. La validité de l'interprétation du cadre structural et des principaux réflecteurs est en effet préalable à tout raffinement du modèle. À l'issue des méthodes de construction automatique (voir par exemple [Duvinaige, 2000 ; Ledez, 2002, 2003 ; Labrunye, 2004]), nous supposons donc dans ce qui suit que les horizons principaux et les failles sont cohérents.

Nous allons à présent développer les étapes 2 et 3 qui sont au coeur de la méthode proposée par [Mallet, 2002].

### Contraintes sur la fonction d'extrusion

La construction de la fonction d'extrusion utilise une comparaison des bords des surfaces  $\mathcal{S}_1$  et  $\mathcal{S}_2$ , qui correspondent soit aux limites du domaine d'étude, soit aux polygones de failles [Duvinaige, 2000]. Les relations ainsi définies permettent d'initialiser des vecteurs sur les bords de la surface de référence (FIG. 1.4-A). Les vecteurs sont ensuite interpolés sur toute la surface de référence avec la méthode DSI [Mallet, 1989, 1992]. Le champ de vecteurs ainsi obtenu est donc principalement contraint par les frontières du domaine et par le réseau de failles. Pour un noeud à l'intérieur de la surface, la direction d'un vecteur est proche de la moyenne des directions des vecteurs sur les noeuds voisins. En général, la méthode DSI évite donc que les vecteurs ne se croisent à l'intérieur du domaine.

### Calcul des cartes de proportions

Les proportions sur la surface sont calculées à partir de points (marqueurs de puits) ou de surfaces (interprétations partielles de données sismiques). En chaque point de donnée  $\mathbf{x}$  est donc disponible une information sur le type d'événement

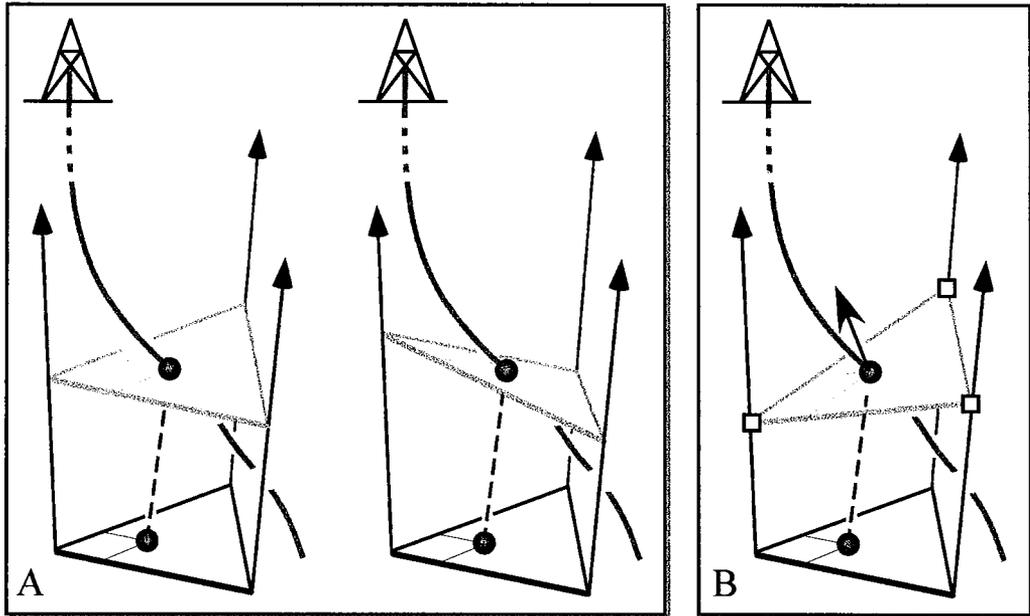


FIG. 1.5 – Deux types de contraintes DSI pour honorer les données de puits. Avec un point de contrôle (A), le triangle « virtuel », en grisé, dispose de degrés de liberté de rotation autour du marqueur de puits. Lorsque le pendage est disponible (B), les proportions sont calculées pour chaque sommet du triangle, et les noeuds du triangle virtuel sont fixés en noeuds de contrôle (carrés blancs).

stratigraphique identifié (nom ou âge de l'horizon, etc.). Pour chaque événement, une fonction de proportion  $p_i$  est créée sur la surface de référence ; cette fonction servira de support à l'interpolation DSI [Mallet, 1989, 2002].

Chacun de ces points est ensuite projeté sur la surface de référence en suivant les vecteurs d'extrusion. Les proportions correspondant aux données disponibles peuvent ainsi servir de contraintes DSI, dont la formulation mathématique est donnée dans [Mallet, 2002, p. 384].

Intuitivement, la contrainte utilisée est analogue à un noeud de contrôle barycentrique (*fuzzy control node*). En effet, elle assure que l'horizon interpolé passe exactement par le point de donnée (FIG. 1.5-A). La formulation de la contrainte est toutefois différente car elle s'applique à un triangle de l'horizon défini implicitement par extrusion d'un triangle de la surface de référence (par application de l'Équation 1.2 page 25). Dans certains cas, il se peut que des pendages soient disponibles au niveau des points de données. Dans ce cas, les trois valeurs de proportion aux sommets du triangle concerné sont fixées (noeuds de contrôle, ou *control nodes*) (FIG. 1.5-B).

La non-intersection des horizons est garantie quant à elle par une contrainte « dure » issue de l'Équation 1.3 page 25 : la proportion  $p_i$ ,  $i \in [1, k]$  est bornée par les proportions  $p_{i-1}$  et  $p_{i+1}$  en tout noeud de la surface de référence. Par

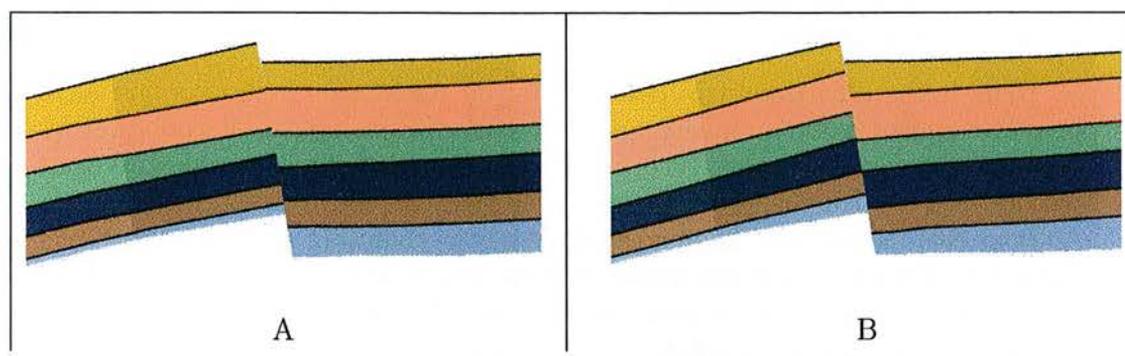


FIG. 1.6 – Prise en compte de failles syn- et post-sédimentaires (respectivement A et B).

convention,  $p_0$  et  $p_k$  sont des fonctions constantes représentant les limites du domaine étudié, égales respectivement à 0 et à 1.

En complément à ce traitement, il est également proposé de tenir compte d'informations implicites au niveau des puits, comme les « marqueurs fantômes ». Il s'agit simplement de tenir compte des couches traversées par les trajectoires de puits. Pour cela, la trajectoire du puits est projetée sur la surface de référence ; en chaque point compris entre un marqueur  $i$  et  $i + 1$ , des contraintes d'inégalité sont mises en place sur les valeurs de proportion  $p_i$  et  $p_{i+1}$  pour empêcher les horizons correspondants de recouper la trajectoire du puits.

La méthode des cartes de proportions permet également de prendre en compte différents types de failles. En effet, lorsqu'elles sont (FIG. 1.6) :

- syn-sédimentaires, les proportions sont interpolées sans maintenir la continuité de part et d'autre des lèvres de faille ;
- post-sédimentaires, une contrainte de continuité permet de garder une épaisseur continue des strates de part et d'autre des failles.

Ainsi, par l'intégration d'informations de différents types (principe stratigraphique de non-intersection, type de failles, données d'observations), la méthode DSI semble appropriée à la modélisation de milieux stratifiés.

### 1.4.2 Limites de la méthode.

Si elle permet de hiérarchiser les informations de subsurface et de représenter les milieux affectés par d'importantes déformations tectoniques, la méthode des cartes de proportions ne résout pas pour autant tous les problèmes. Ses principales limites sont liées d'une part à la fonction d'extrusion, d'autre part à l'interpolation des proportions. Nous allons à présent préciser ces deux points, avant de présenter les extensions mises en place durant cette thèse.

### Extrusion vectorielle

Dans le cas d'horizons très déformés ou de réseaux de failles complexes, il peut être difficile d'obtenir un champ de vecteurs convenable par interpolation des vecteurs d'extrusion. De plus, la corrélation de traces de failles entre deux horizons par des vecteurs [Duvinaige, 2000] impose que les failles soient des surfaces réglées.

Pour cela, la méthode a été généralisée par la définition d'un champ de vecteurs dans l'espace, contraint à être tangent aux failles et aux contacts faille / faille [Souche, 2002, 2003]. À partir de ce champ de vecteurs, une  *fibre*  peut être construite pour chaque noeud de la surface de référence à l'aide d'une méthode de Runge-Kutta [Abramowitz *et al.*, 1972]. Les fibres jouent ainsi le rôle de vecteurs d'extrusion curvilinéaires, ce qui permet de mieux prendre en compte les géométries de failles.

Les réseaux de failles complexes (en particulier failles synthétiques et antithétiques « en Y ») peuvent être traités par l'utilisation de deux réseaux de fibres superposés (l'un aligné sur le « / » et l'autre sur le « \ » du « Y »).

### Choix de l'interpolateur

Le choix de DSI comme interpolateur peut lui-même être discuté. Prenons l'exemple d'une surface de référence plane, et imaginons qu'un simple puits vertical est utilisé pour générer les cartes de proportions. Le résultat de l'interpolation aura tendance à homogénéiser l'épaisseur apparente des couches (minimisation de la rugosité). Par comparaison avec les modèles de la stratigraphie séquentielle (FIG. 1.1), ce comportement est peu réaliste. Dans une telle configuration, où si peu de données sont disponibles, il est difficile d'attendre plus de la méthode. Cependant, la prise en compte d'informations qualitatives connues par des études sédimentaires (milieu de sédimentation, la direction du flux sédimentaire, etc.) pourrait avoir une influence non négligeable sur le modèle final.

Plus généralement, la méthode a tendance à homogénéiser l'épaisseur apparente des couches loin des points de données. Cela a deux conséquences :

- les tendances définies par d'éventuelles données de pendage sont relativement peu extrapolées.
- l'extension horizontale des biseaux stratigraphiques (couches d'épaisseur nulle) est souvent sous-estimée par la méthode. En effet, la contrainte d'épaisseur nulle n'est installée qu'au niveau des puits dans lesquels la couche n'est pas visible ; l'interpolation confère une épaisseur non nulle à ce couches dès que l'on s'éloigne de ces puits (FIG. 1.7).

Là encore, l'incorporation de connaissances qualitatives permettrait d'améliorer la méthode. Cependant, du fait de la diversité des processus régissant la sédimentation et l'évolution des roches, la définition de méthodes inverses atteignant le niveau de réalisme des méthodes génétiques paraît à ce jour hors de portée. Pour cette raison, notre approche a été d'améliorer la méthode existante tout conservant son esprit. Ces améliorations consistent à raffiner le résultat ob-

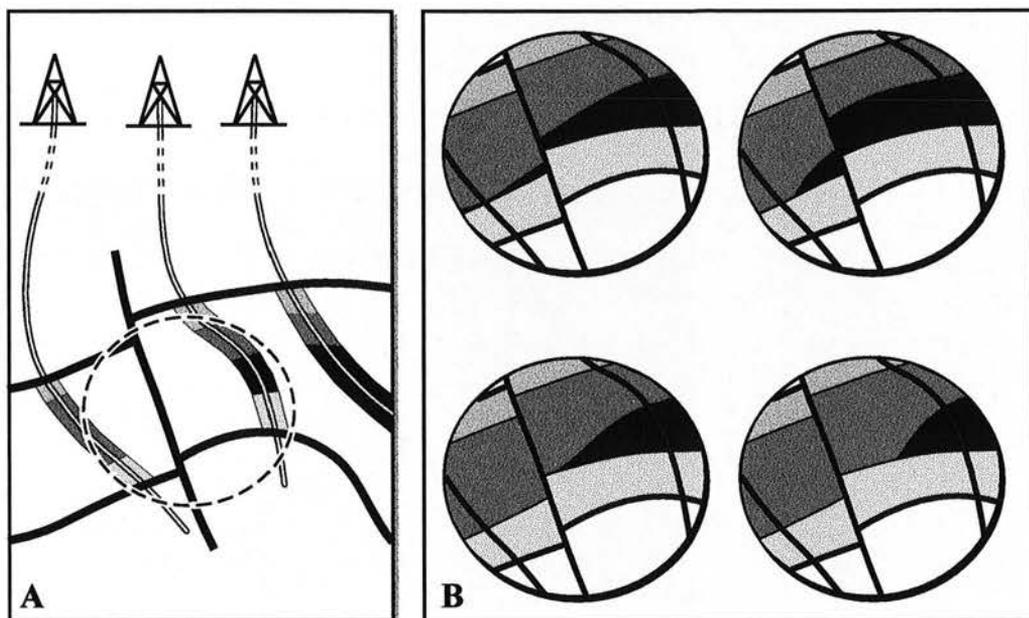


FIG. 1.7 – Ambiguïtés lors d'existence de biseaux. Lorsque des couches disparaissent d'un puits à l'autre (A), le biseau peut prendre des formes variées (B). L'interpolation des proportions par DSI (B, en haut à gauche) a tendance à créer des biseaux d'angle très faible, localisés autour des puits.

tenu automatiquement par des modifications interactives. La fin de ce chapitre est donc consacrée à la visualisation et au procédé de modification mis en oeuvre.

## 1.5 Visualisation de milieux stratifiés

Jusqu'alors, seuls les aspects géologiques et mathématiques de la méthode des cartes de proportions ont été abordés. Cependant, un modèle numérique du sous-sol est amené à évoluer lorsque de nouvelles informations sont disponibles. Comme nous venons de le voir, ces informations peuvent être apportées par un avis d'expert. Pour cela, il est nécessaire de fournir des outils de visualisation performants et explicites.

### 1.5.1 Utilisation de coupes

Au cours des étapes définies page 27, le seul contrôle possible sur le modèle final provient des données initiales. La visualisation du modèle stratigraphique lui-même ne peut se faire que par extrusion des surfaces ou d'une grille stratigraphique. Ces opérations sont certes le but ultime de la modélisation, mais peuvent prendre du temps et des ressources inutiles s'il s'agit simplement d'effectuer un

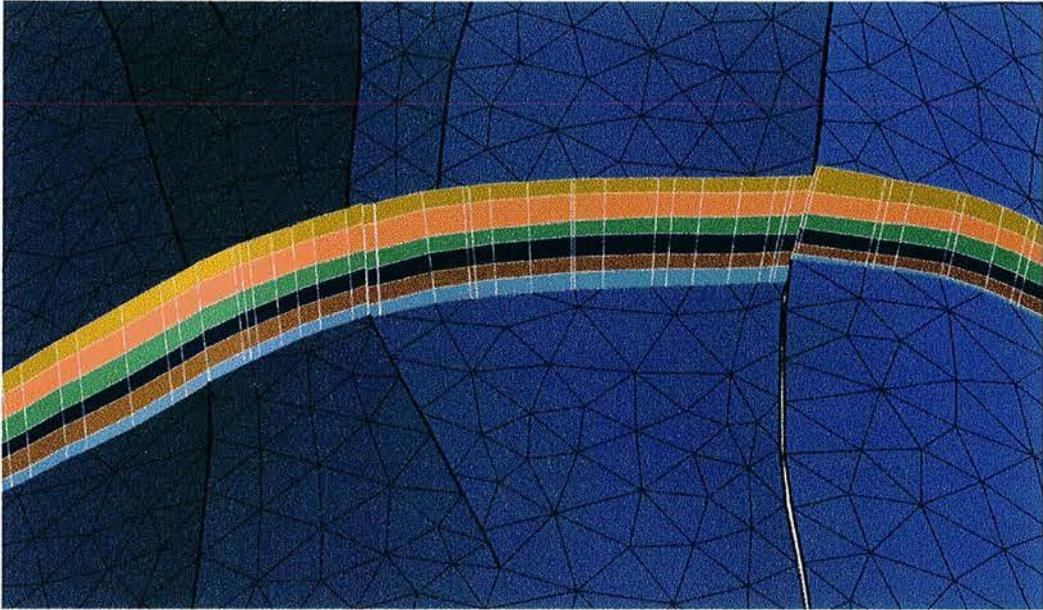


FIG. 1.8 – Une coupe dans un modèle stratifié est définie par une colonne vertébrale plongée sur la surface de référence. Les vecteurs d’extrusion et les proportions interpolés sur la surface permettent la représentation graphique.

contrôle qualité du modèle.

Pour cela, des coupes transversales alignées sur les vecteurs d’extrusion ont été mises en place. La représentation de ces coupes est extrêmement simple, et très proche de la représentation du milieu stratifié lui-même. Elle s’appuie sur l’extrusion d’une ligne polygonale  $\mathcal{L}$ , appelée *colonne vertébrale* de la coupe, par analogie avec les structures de représentations de chenaux [Viseur, 2001]. Cette ligne, projetée sur la surface de référence, est le support d’une fonction d’extrusion vectorielle **extr** définissant la surface de coupe. Elle comporte de plus  $k$  fonctions de proportions  $\{p_1, \dots, p_k\}$ , qui définissent les subdivisions de la coupe par les couches (FIG. 1.8).

En termes informatiques, une coupe peut être représentée par les structures de données de la FIG. 1.9.

<pre><b>struct</b> LigneCoupe {     Noeud noeuds[];     int nb_noeuds;     int nb_proportions; };</pre>	<pre><b>struct</b> Noeud {     float géométrie[3];     float extrusion[3];     float proportions[]; };</pre>
---	--

FIG. 1.9 – Structure de données simplifiée pour la représentation d’une coupe dans un milieu stratifié.

Le dessin d’une telle coupe (Algorithme 1.1) se fait en deux boucles imbri-

---

**Algorithme 1.1** Dessin d'une coupe dans un modèle stratifié. L'instruction **point** fait référence à la fonction de dessin de la librairie graphique 3D (par exemple OpenGL [Neider *et al.*, 1993]).

---

**Dessine\_coupe :**

**Entrée :**

$\mathcal{C}$  : Coupe

**Début**

LigneCoupe  $\mathcal{L} \leftarrow$  colonne vertébrale de  $\mathcal{C}$

**Pour**  $i$  **de** 1 **à**  $\mathcal{L}.nb\_proportions$

  // Activer une nouvelle bande de quadrilatères de couleur  $i$

**Pour**  $j$  **de** 0 **à**  $\mathcal{L}.nb\_noeuds - 1$

$n_{cour} : \text{Noeud} \leftarrow \mathcal{L}.noeuds[j]$

**point**( géométrie(  $n_{cour}$  ) +  $p_{i-1}(n_{cour}) \cdot \mathbf{extr}(n_{cour})$  )

**point**( géométrie(  $n_{cour}$  ) +  $p_i(n_{cour}) \cdot \mathbf{extr}(n_{cour})$  )

**FinPour**

**FinPour**

**Fin**

---

quées, l'une pour les couches, l'autre pour les segments de la colonne vertébrale<sup>3</sup>. L'exactitude de la coupe dépend quant à elle de la discrétisation de sa colonne vertébrale. Si la courbe est grossière relativement au maillage de la surface de référence, l'image générée sera peu fidèle au modèle. En contrepartie, une courbe plus dense donnera de meilleurs résultats, mais avec des performances moindres. Il est donc important d'obtenir une résolution adaptée dès la création de la coupe.

### 1.5.2 Création d'une coupe

Une résolution de la colonne vertébrale  $\mathcal{L}$  adaptée à la surface de référence  $\mathcal{S}$  peut être obtenue par le *coraffinement* de  $\mathcal{L}$  par  $\mathcal{S}$  [Conreux, 2001]. À l'issue de cette opération (FIG. 1.8) :

- chaque noeud de  $\mathcal{L}$  se situera sur la surface  $\mathcal{S}$  ;
- $\mathcal{L}$  ne sera recoupée par le maillage de  $\mathcal{S}$  qu'en ses noeuds.

Le coraffinement d'une ligne par une surface est également décrit dans [Grosse, 2002], pour définir les « Trimmed Triangulated Surfaces ». Par rapport à cette méthode, notre but est simplement de connaître le plongement de tout noeud  $n$

---

<sup>3</sup>Cet ordre de boucle permet d'accélérer le dessin par l'utilisation de bandes de quadrilatères (*quad strips*, [Neider *et al.*, 1993]). En effet, il y a généralement bien plus de segments sur la colonne vertébrale que de couches à dessiner.

---

**Algorithme 1.2** Coraffinement de la colonne vertébrale  $\mathcal{L}$  d'une coupe dans un milieu stratifié.

---

**Coraffiner\_ligne :**

**Entrée :**

$\mathcal{L}$  : LigneCoupe ; la colonne vertébrale de la coupe à construire

$\mathcal{S}$  : Surface : la surface de référence

**Début**

$n_{pre}$  : Noeud  $\leftarrow \mathcal{L}.noeuds[0]$

$plon_{pre}$  : Plongement  $\leftarrow$  recherche globale du plongement de  $n_{pre}$  sur

$\mathcal{S}$

**Pour**  $i$  **de** 1 **à**  $\mathcal{L}.nb\_noeuds - 1$

$n_{cur}$  : Noeud  $\leftarrow \mathcal{L}.noeuds[i]$

$plon_{pre} \leftarrow \text{coraffiner\_segment}(n_{pre}, plon_{pre}, n_{cur}, \mathcal{L}, \mathcal{S})$

**FinPour**

**Fin**

---

de la ligne  $\mathcal{L}$  dans la surface  $\mathcal{S}$ , défini par :

$$plongement(n, \mathcal{L}, \mathcal{S}) = \begin{cases} \{n_S, 0\} & \text{si } n \text{ est inclu topologiquement} \\ & \text{dans un noeud } n_S \in \mathcal{S} \\ \{a_S, 1\} & \text{si } n \text{ est inclu topologiquement} \\ & \text{dans une arête } a_S \in \mathcal{S} \\ \{t_S, 2\} & \text{si } n \text{ est inclu topologiquement} \\ & \text{dans un triangle } t_S \in \mathcal{S} \end{cases}$$

Cette information permet en effet de mettre la coupe à jour rapidement lorsque les vecteurs d'extrusion ou les proportions sont modifiés sur la surface.

Au moment de la création d'une coupe à partir d'une ligne polygonale quelconque nous allons donc :

1. projeter les noeuds de la ligne  $\mathcal{L}$  sur la surface ;
2. subdiviser les segments de  $\mathcal{L}$  qui recoupent le maillage de  $\mathcal{S}$  (*coraffinement* de  $\mathcal{L}$  par  $\mathcal{S}$ ) ;
3. stocker le plongement des noeuds de  $\mathcal{L}$  ;
4. calculer pour chaque noeud de  $\mathcal{L}$  les valeurs des vecteurs d'extrusion et les proportions en utilisant le plongement.

Le coraffinement de  $\mathcal{L}$  (Algorithme 1.2) est réalisé segment par segment (Algorithme 1.3). De façon similaire à [Conreux, 2001], cette opération fait intervenir un automate d'états finis (FIG. 1.10) pour déterminer le chemin entre deux noeuds (fonction *successeur*). Par rapport aux méthodes de coraffinement décrites dans [Euler, 1999 ; Conreux, 2001 ; Grosse, 2002], une seule recherche globale de plongement est effectuée pour toute la ligne. Les plongements et les noeuds issus du coraffinement sont ensuite déterminés de proche en proche. Lorsque les

---

**Algorithme 1.3** Coraffinement d'un segment de la colonne vertébrale  $\mathcal{L}$  d'une coupe. La recherche de **successeur** fait appel à l'automate d'états finis de la FIG. 1.10.

---

**coraffiner\_segment :**

**Entrée :**

$\mathcal{L}$  : LigneCoupe ; la colonne vertébrale de la coupe à construire

$\mathcal{S}$  : Surface ; la surface de référence

$n_1, n_2$  : Noeuds ; extrémités du segment à raffiner

$plong$  : Plongement ; plongement de  $n_1$  dans  $\mathcal{S}$

**Sortie :**

$plong$  : Plongement ; plongement de  $n_2$  dans  $\mathcal{S}$

**Début**

$\mathcal{P}$  : Plan  $\leftarrow$  plan passant par  $n_1, n_2$ , et localement normal à  $\mathcal{S}$

$n_{cour}$  : Noeud  $\leftarrow n_1$

**Faire**

$\{n_{cour}, plong\} \leftarrow \text{successeur}(n_{cour}, plong, \mathcal{P}, n_2)$

**insérer**  $n_{cour}$  dans  $\mathcal{L}$

**Tant que**  $n_{cour} \neq n_2$  **Fin**

---

deux extrémités d'un segment se situent dans deux blocs de faille différents, les relations entre bords de la surface de référence sont utilisées [Ait Ettajer, 1995 ; Cognot *et al.*, 1997].

## 1.6 Modifications de la géométrie des couches sous contraintes

La définition des coupes dans le modèle stratifié permet de visualiser le résultat de l'interpolation automatique décrit paragraphe 1.4, page 24. Un tel contrôle visuel a cependant peu d'intérêt s'il n'est pas possible par ailleurs de modifier le modèle.

### 1.6.1 Choix de la méthode

#### État de l'art

De nombreuses méthodes dédiées à la modification de modèles géométriques sont définies dans la littérature informatique. Ces opérations soulèvent en effet plusieurs problèmes informatiques.

Dans un logiciel de dessin, l'espace de modélisation, en deux dimensions, correspond à l'espace du périphérique d'entrée, généralement une souris. Le degré de

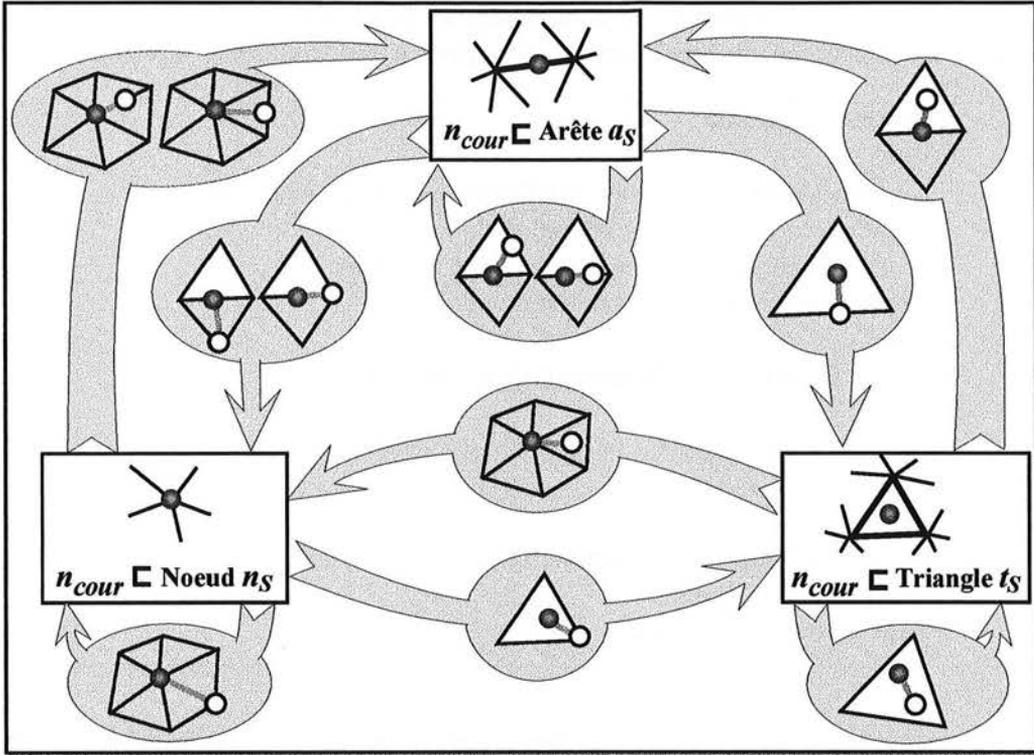


FIG. 1.10 – Automate d'états finis utilisé pour déterminer récursivement le co-raffinement d'un segment par une surface triangulée  $\mathcal{S}$  (fonction **successeur** dans l'Algorithme 1.3). À partir d'un noeud  $n_{cour}$  plongé dans  $\mathcal{S}$  (en gris foncé), et d'un plan  $\mathcal{P}$  contenant le segment (en gris), l'automate donne un nouveau noeud à insérer (en blanc), ainsi que son plongement.

liberté de rotation est bloqué dans la plupart des modes de dessin, ce qui permet de traduire tout mouvement de la souris par une translation dans le modèle.

Dans un espace à trois dimensions, les objets disposent non plus de trois, mais de six degrés de liberté (trois directions de translation et trois axes de rotation). Une souris classique, quant à elle, ne permet d'en contrôler que deux simultanément. L'écran lui-même permet difficilement d'appréhender la profondeur d'un modèle sans déplacer le point de vue.

Pour être univoque, la manipulation d'un objet dans un espace 3D demande donc, soit d'utiliser des périphériques spécifiques, soit d'imposer un certain nombre de *contraintes*. Les périphériques 3D sont relativement chers, peu répandus et restent souvent assez spécifiques et complexes [Zhai, 1998]. Aussi, de nombreuses méthodes ont été définies pour modifier des objets dans un environnement 3D à partir de périphériques bidimensionnels.

Une première approche consiste à définir plusieurs modes d'édition, chacun

fixant quatre degrés de libertés, et laissant à la souris le contrôle des deux autres. Par exemple, le déplacement de la souris est traduit par une translation dans un plan parallèle à l'écran de l'ordinateur [Nielson *et al.*, 1987]. Dans d'autres cas, les mouvements vont s'appuyer sur des objets graphiques temporaires (*3D widgets*) qui permettent de définir plus facilement un axe de rotation, de translation, etc. [Conner *et al.*, 1992 ; Zeleznick *et al.*, 1993]. Ce type d'approche est également utilisé avec les *manipulateurs* de la bibliothèque OpenInventor [Wernecke, 1994].

Quoi qu'il en soit, ces méthodes nécessitent en général de nombreuses interactions avant de parvenir au résultat souhaité. Au-delà de ces problèmes d'ergonomie, la précision de tels systèmes est souvent trop faible pour les exigences de la modélisation géométrique 3D [Hsu *et al.*, 1997]. Ainsi, plusieurs approches utilisent des parties du modèle pour contraindre les mouvements du curseur. C'est le cas du magnétisme (*snap-dragging*), introduit par [Bier *et al.*, 1986 ; Bier, 1990] : lorsque le curseur se trouve près d'un objet, il est automatiquement projeté sur cet objet. Le magnétisme s'intègre dans une approche plus générale d'*édition sous contraintes*, dans lesquelles des relations entre les objets définissent le comportement du modèle lors d'une modification. L'idée de mettre en place de telles relations remonte aux balbutiements de l'informatique graphique, en 1963 avec le visionnaire Sketchpad [Sutherland, 1963]. Cependant, le traitement des contraintes pose des problèmes de logique (le système est-il sous-déterminé ou sur-déterminé ? Les contraintes sont-elles cohérentes ?), de performance (le temps de résolution augmente de manière exponentielle avec le nombre de contraintes) et d'architecture logicielle. Sur les deux premiers aspects, le lecteur pourra consulter une étude bibliographique organisée autour des différents solveurs utilisés [Hower *et al.*, 1996]. Les aspects logiciels sont par ailleurs développés dans [Helm *et al.*, 1995] par les initiateurs des « design patterns » en programmation orientée objet [Gamma *et al.*, 1994].

Enfin, les méthodes de modification interactives soulèvent des problèmes inhérents à la nature et à la représentation même des modèles modifiés (cf. Chapitre 2).

### Approche choisie

Dans le cas de modèles stratifiés, le choix des contraintes est facile à définir : les couches ne doivent pas se croiser (Équation 1.3). De plus, par la représentation elle-même, chaque limite de couche ne peut se déplacer que par translation le long des vecteurs d'extrusion.

L'interface utilisateur, consiste simplement à sélectionner un horizon sur une coupe, et à le déplacer sous contraintes. Dans un premier temps une modification de la coupe elle-même a été envisagée. Cependant, il est vite apparu qu'une modification directe du modèle stratifié était plus intéressante. Elle permet en effet :

- de maintenir à tout instant la cohérence entre la coupe et le modèle ;

- de gérer facilement les coupes multiples ;
- de conserver aisément un modèle stratigraphique contraint par les données de puits ;
- de limiter la taille du code en s'appuyant sur les outils d'interpolation existants, donc de faciliter la maintenance.

## 1.6.2 Méthode mise en oeuvre

### Interprétation d'une modification

Dans l'approche mise en oeuvre, la modification d'un modèle correspond à l'ajout d'information par un expert. Comme la méthode de construction du modèle s'appuie sur l'interpolateur DSI, il est intéressant d'exprimer cette information sous la forme d'une contrainte DSI. La généralité de la méthode DSI [Mallet, 2002] confère à cette formulation plusieurs avantages. En premier lieu, elle permet de gérer simultanément les contraintes liées à la manipulation et les autres informations (données de puits en particulier). Du point de vue logiciel, cette homogénéité facilite la maintenance du programme, paramètre important dans un projet industriel. Enfin, cette formulation permet de sauvegarder l'information intégrée dans le modèle en tant que donnée. Ainsi, l'interprétation ne sera pas perdue si le modèle doit être reconstruit (par exemple, suite à l'ajout d'une faille).

Lors de la construction du modèle (Partie 1.4.1, page 24), nous avons vu que les points de données (puits) sont honorés exactement par l'interpolation DSI. Les contraintes installées sont en effet des *contraintes dures* (*hard constraints*), qui ont préséance sur les contraintes molles (*soft constraints*). Au cours de la modification, il est en général peu souhaitable d'ignorer le peu d'informations disponibles ; ainsi, la modification est traduite par une contrainte molle (point de contrôle ou *fuzzy control point*). Lorsque aucune autre contrainte n'est définie au voisinage d'un point de contrôle, celui-ci sera honoré correctement, si tant est que le poids de la contrainte soit suffisant par rapport au critère de rugosité. En revanche, le point de contrôle aura peu ou pas d'influence s'il n'est pas concordant avec à une donnée de puits voisine (représentée par un noeud de contrôle).

### Étapes de la modification

Nous pouvons à présent préciser les étapes de la modification d'un modèle stratifié. Pour l'utilisateur, l'opération consiste simplement à sélectionner un horizon sur une coupe, puis à le déplacer jusqu'à la position désirée. La modification procède ainsi en trois étapes : sélection et initialisation, manipulation, et terminaison.

1. Le point sélectionné doit se trouver sur la trace d'un horizon (proportion  $i$ ) sur la coupe. Si c'est le cas, l'image de ce point sur la surface de référence

$\mathcal{S}$  est calculée par projection suivant les vecteurs d'extrusion. Ce nouveau point peut ainsi servir de point de contrôle à la proportion  $p_i$  sur  $\mathcal{S}$ . À l'aide d'une distance d'influence (définie par l'utilisateur), les noeuds de  $\mathcal{S}$  concernés par la modification peuvent ensuite être identifiés.

2. Pour chaque déplacement du curseur, la proportion  $i$  du point de contrôle est recalculée. Seulement quelques itérations de DSI sur la région d'influence de  $\mathcal{S}$  permettent ensuite de mettre à jour le modèle. En effet, les positions du curseur aux instants  $t$  et  $t + dt$  sont généralement voisines. Ainsi, le nombre d'itérations nécessaires pour atteindre la convergence du système est beaucoup plus faible que dans un contexte normal. Ce petit nombre d'itérations est la clé de l'interactivité du système, pour peu que la zone d'influence soit suffisamment réduite (FIG. 1.11). Jusqu'à présent, la coupe n'a pas été modifiée et aucun retour visuel n'est donc encore effectué. Les proportions de la coupe sont donc mises à jour à partir du modèle, en utilisant le plongement des noeuds de la colonne vertébrale sur  $\mathcal{S}$ .
3. Lorsque la position désirée est atteinte, un signal est émis pour mettre à jour les observateurs du modèle. Cela permet en particulier de mettre en cohérence d'autres coupes définies dans le modèle stratifié.

En résumé, la méthode mise en oeuvre joue sur deux plans. Tout d'abord elle assure à tout moment le respect des données et la non-intersection des horizons par une interpolation locale du modèle stratifié et non de la coupe. Cette dernière reste ainsi une simple interface qui reste inféodée au modèle, et non l'inverse. Ensuite, les opérations effectuées durant le déplacement sont réduites au minimum afin de garantir l'efficacité de la méthode.

### 1.6.3 Bilan et limites de la méthode proposée

La modification des cartes de proportions est d'une utilisation très intuitive. De plus, la représentation proposée dans [Mallet, 2002] permet de maintenir facilement la cohérence du modèle.

La localité des modifications proposées, si elle permet d'assurer l'interactivité, présente toutefois une limite importante : l'utilisateur doit effectuer de nombreuses modifications pour éviter des ondulations artificielles des horizons : le long d'une section il est facile de vérifier la régularité d'un horizon ; en revanche, l'extension transversale d'une modification locale est limitée, ce qui n'apparaît pas sur la section modifiée. Il est donc important pour l'utilisateur d'utiliser des coupes croisées afin de mieux appréhender les effets en 3D d'une modification effectuée sur un objet 2D.

Concernant le problème de biseaux stratigraphiques soulevé au paragraphe 1.4.2, une manipulation interactive en coupes transversales reste un *pis-aller*. Il serait en effet intéressant de proposer une modification en carte directement sur la surface de référence. Cette modification pourrait s'inscrire dans un cadre plus

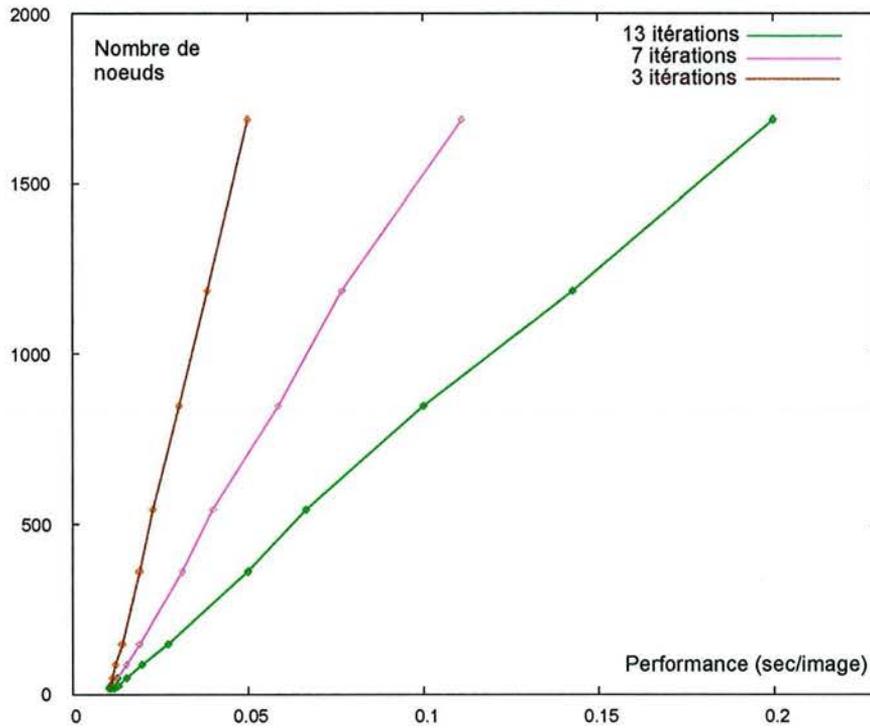


FIG. 1.11 – Interactivité de la modification. La vitesse de mise à jour est proportionnelle au nombre de noeuds dans la zone d'influence. Jusqu'à 200 noeuds environ, 3 itérations par image sont suffisantes pour atteindre la convergence. Au-delà, l'utilisateur doit déplacer le curseur lentement, ou augmenter le nombre d'itérations.

vaste, celui d'une modification interactive de lignes de niveau sur une surface triangulée. Le schéma pourrait en être le suivant :

1. L'utilisateur sélectionne une ligne de niveau sur la surface de référence.
2. Cette ligne est localement approximée par une courbe paramétrique facilement modifiable (par exemple une b-spline cubique ou une NURBS, voir [Farin, 2001 ; Piegl *et al.*, 1997]). Un ensemble de contraintes DSI est installé au niveau de cette courbe pour contraindre l'isovaleur.
3. Lorsque la courbe et les contraintes d'isovaleur associées sont déplacées sur la surface, la propriété est réinterpolée localement. Les lignes de niveau sont alors recalculées et dessinées par un algorithme rapide (par exemple celui proposé par Bajaj *et al.* [1996], cf. §3.3, p. 96).

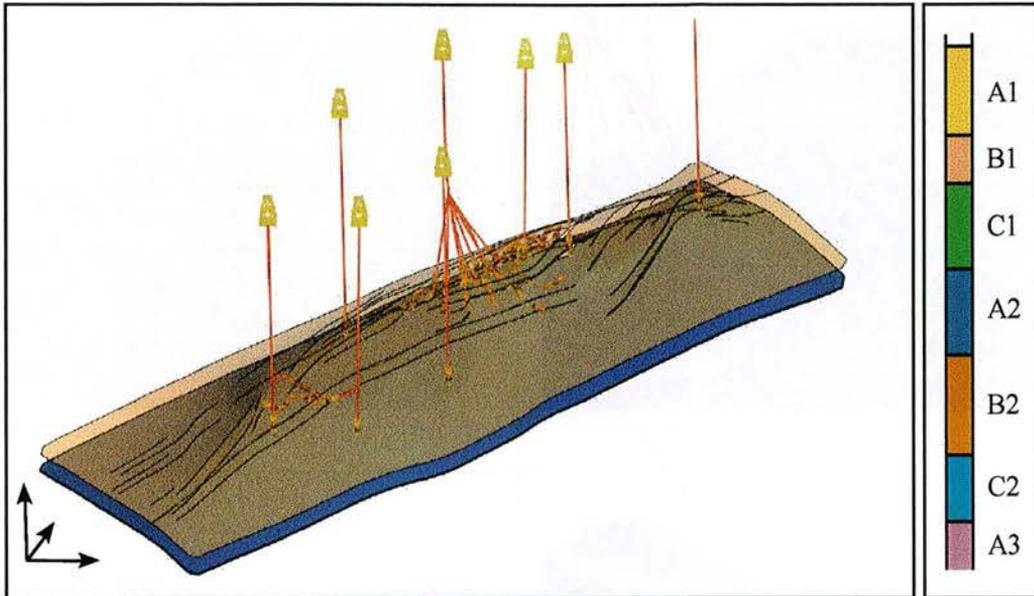


FIG. 1.12 – Jeu de données dit « *Sinusoidal* », fourni par CHEVRONTXACO (dilatation verticale :  $\times 2$ ). Les données comprennent les trajectoires et marqueurs de puits, le toit du réservoir (en transparence) et son mur, utilisé comme surface de référence. À droite, la succession stratigraphique de référence, observée dans le puits vertical n° 7.

## 1.7 Application

### 1.7.1 Présentation des données

Les améliorations apportées aux cartes de proportions durant cette thèse sont actuellement en cours d'intégration dans le logiciel Gocad. Ce travail est cependant utilisé depuis deux ans par la compagnie CHEVRONTXACO. Dans ce paragraphe, la méthode est appliquée à des données réelles, dont la compagnie a gracieusement accepté la publication. Cependant, pour des raisons de confidentialité, le nom du champ et le contexte géologique ne peuvent pas être divulgués. Cette application constitue donc plus une illustration de la méthode qu'une véritable étude de cas.

Le jeu de données est constitué de deux surfaces faillées définissant le toit et le mur d'un réservoir pétrolier. Sur les 21 puits forés dans ce réservoir, les marqueurs de sept horizons ont été identifiés (FIG. 1.12). Les sept puits verticaux ont permis de caractériser un réservoir à fort contraste entre une perméabilité verticale  $K_v$  faible et une perméabilité horizontale  $K_h$  plus forte. Cela explique les trajectoires « sinusoidales » des autres puits, destinées à d'augmenter le plus possible la productivité.

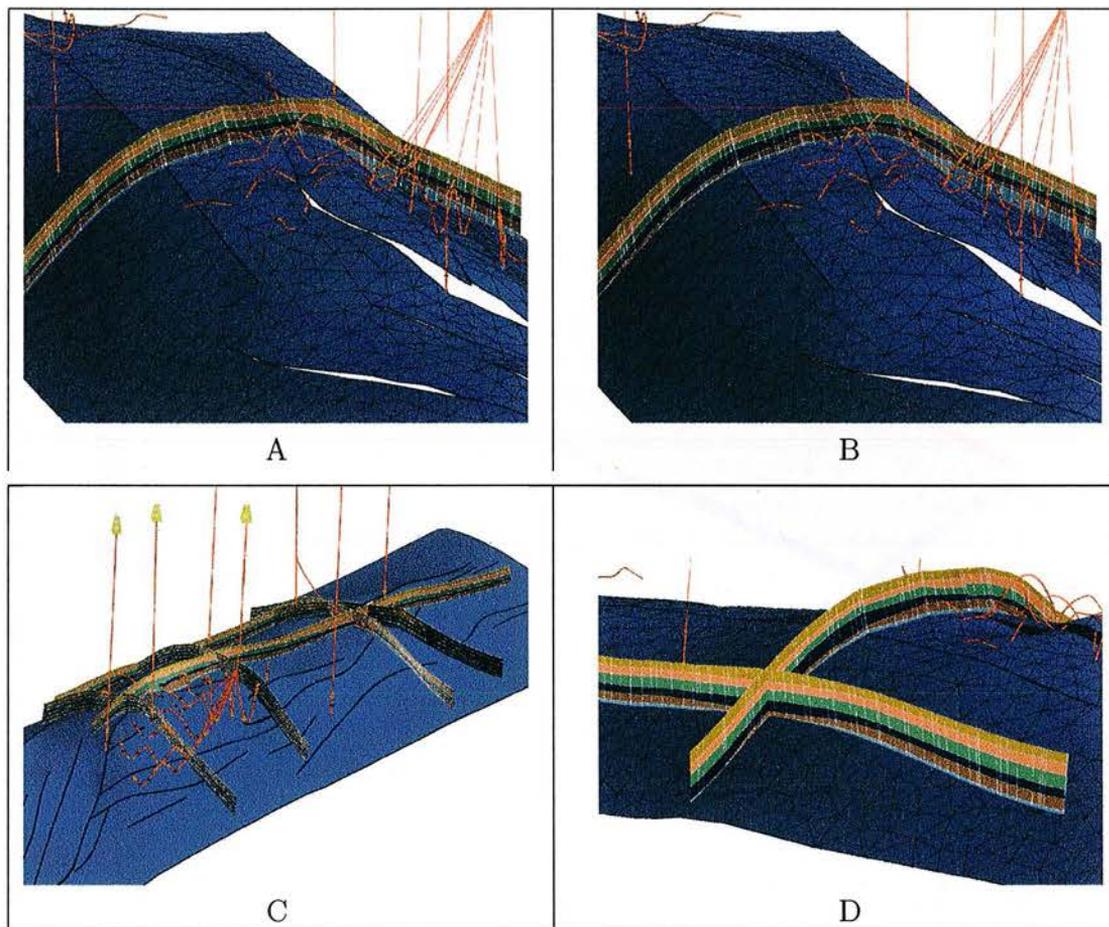


FIG. 1.13 – Exemple d'utilisation de coupes dans un modèle stratifié. Quelques modifications manuelles suffisent à imposer des biseaux stratigraphiques loin des données (A et B). La combinaison de coupes d'orientation différentes permet de visualiser l'étendue des modifications.

### 1.7.2 Application de la méthode

Des exemples d'application sont donnés par la FIG. 1.13. Les coupes sont tracées suivant des lignes arbitraires sur la surface de référence, et orientées selon les vecteurs d'extrusion. Près des données de puits, le modèle est contraint, et les géométries des couches sont donc difficilement modifiables. En revanche, suite par exemple à des simulations sur des modèles génétiques, il est possible d'imposer des biseaux stratigraphiques interactivement loin des données (FIG. 1.13-A,B). En combinant plusieurs coupes d'orientations complémentaires, il est possible d'échantillonner le modèle volumique de manière satisfaisante (FIG. 1.13-C). Cela permet en particulier de visualiser l'impact d'une modification dans toutes les directions (FIG. 1.13-B,-D).

Dans l'industrie pétrolière, la construction du modèle d'architectures strati-

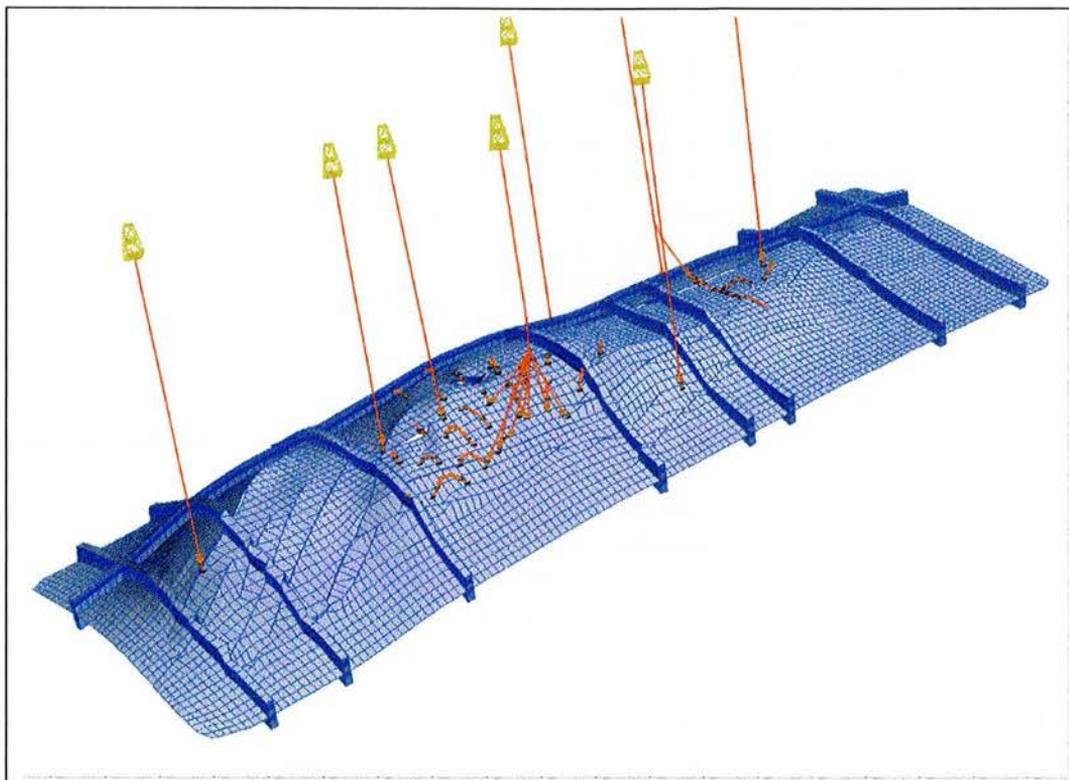


FIG. 1.14 – Modèle cellulaire généré à partir du modèle par extrusion.

graphiques n'est pas une fin en soi, mais une étape importante avant de créer des représentations cellulaires, ou grilles (cf. Chapitre 3), qui vont être des supports de calculs géostatistiques et de simulations d'écoulement. Dans cet exemple, une grille stratigraphique (FIG. 1.14) a été créée par extrusion grâce à la paramétrisation de la surface de référence [Lévy, 1999].

## 1.8 Bilan et perspectives

Par rapport aux objectifs fixés dans le paragraphe 1.2.2, la méthode des cartes de proportions définie dans [Mallet, 2002] utilise une représentation compacte du modèle, en factorisant les informations combinatoires de chaque horizon au niveau de la surface de référence. S'appuyant sur des surfaces triangulées, elle permet de représenter des formations faillées et plissées.

Le passage à d'autres représentations est permis par le modèle. En particulier la création de surfaces triangulées pour représenter explicitement les horizons est réalisable par extrusion. À l'aide d'une paramétrisation de la surface de référence [Lévy, 1999], il est également possible de définir des grilles tridimensionnelles. Ces grilles (voir Chapitre 3), pourront être utilisées pour effectuer des estimations ou

des simulations géostatistiques, des simulations d'écoulements, etc.

Les outils de visualisation et de modification interactive mis en place dans ce travail de thèse ajoutent de la flexibilité à la méthode, en permettant de contrôler visuellement les modèles générés et de prendre en compte des avis d'experts. D'un point de vue informatique, la méthode s'appuie sur des surfaces de coupes afin de lever toute ambiguïté quant à la position du curseur de souris. Cependant, contrairement aux apparences, la coupe n'est pas le moteur de la modification, mais constitue une interface pour modifier le modèle lui-même. Cela permet de maintenir à tout moment la cohérence du modèle, ce qui ne serait pas garanti par une simple modification de la coupe. Celle-ci, par un couplage étroit avec la surface de référence, est mise à jour de façon transparente pour l'utilisateur.

Comme nous l'avons vu section 1.4.2, la gestion de failles de géométries complexes peut se faire en utilisant des fibres en lieu et place des vecteurs d'extrusion [Souche, 2003]. Cette amélioration ne modifie pas le coeur de la méthode, et pourra donc être prise en compte facilement.

Par rapport à la méthode de Goff [2000], les modèles stratigraphiques que nous obtenons sont déterministes. Il serait intéressant de générer plusieurs réalisations équiprobables d'un modèle afin de mieux évaluer les incertitudes structurales. Pour cela, l'approche fréquentielle [Goff, 2000] pourrait être utilisée. Une autre approche, similaire à [Lecour, 2000], pourrait être d'appliquer une perturbation aléatoire aux proportions. Nulle au niveau des puits, cette perturbation pourrait être estimée par simulation gaussienne séquentielle [Deutsch *et al.*, 1997 ; Goovaerts, 1997].

À plus long terme, une meilleure caractérisation des formations stratifiées passe sans doute par l'intégration des concepts de la stratigraphie séquentielle [Homewood *et al.*, 1999 ; Miall, 1997] dans les modèles inverses. En complément de l'inversion stratigraphique [Bornholdt *et al.*, 1999 ; Cross *et al.*, 2002, 1999], l'utilisation de méthodes géostatistiques à points multiples [Caers, 2001 ; Strebelle, 2002] paraît une piste possible pour cette tâche.

# Chapitre 2

## Modèles volumiques définis par frontières.

### 2.1 Introduction

Dans le chapitre précédent, nous avons étudié comment une famille de surfaces triangulées pouvait être représentée et utilisée efficacement pour modéliser des architectures stratigraphiques. L'idée principale de cette représentation est de tirer parti de la conformité relative des horizons dans les formations stratifiées. En cela, la représentation par cartes de proportions est peu adaptée pour tenir compte de la variabilité des objets géologiques. Dans le cas général, il est en effet souhaitable de disposer d'une représentation capable de traiter :

- des réseaux de failles complexes ;
- des morphologies très perturbées (toit de couches salifères, plis couchés ou en fourreau, intrusions magmatiques) ;
- des corps de topologie variée – surfaces ouvertes, fermées, toriques, etc. – (minéralisations, intrusions, lentilles, etc.)

Pour gérer ce type de configurations, une approche consiste à définir un volume rocheux de caractéristiques relativement homogènes (minéralisation, couche, intrusion, ...) par ses limites (FIG. 2.1). En fait, ce concept de représentation est né dans le domaine de la conception assistée par ordinateur (CAO), en particulier pour modéliser des objets manufacturés. Dans le domaine de la modélisation volumique [Mäntylä, 1988 ; Hoffmann, 1989a], ce type de représentation porte le nom de *représentation par frontières* ou *b-rep* (*boundary representation*).

Consacré à ces représentations et à leur modification pour des applications géologiques, ce chapitre est organisé comme suit. Tout d'abord, l'accent est mis sur l'importance des b-reps en Géomodélisation, ainsi que sur les principaux thèmes de recherche liés à ces représentations. La structure de données utilisée est ensuite présentée. Le contexte de travail ainsi défini, la méthode de modifica-

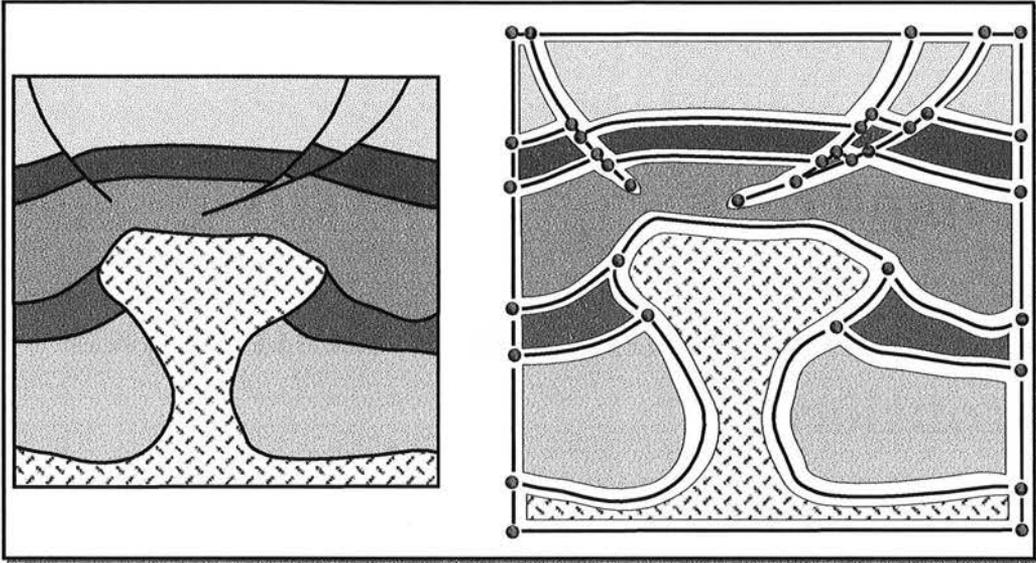


FIG. 2.1 – Un modèle géologique en deux dimensions (à gauche) peut être représenté par frontières (à droite). Les régions, délimitées par des failles, des horizons, des surfaces intrusives, peuvent être regroupées en couches ou en blocs de failles. Ce type de représentation peut être étendu en trois dimensions.

tion de b-reps mise au point durant cette thèse [Caumon *et al.*, 2002, 2003b] est présentée.

## 2.2 Enjeux d'une représentation par frontières

### 2.2.1 Les b-reps : une représentation volumique parmi d'autres

La modélisation volumique [Mäntylä, 1988 ; Hoffmann, 1989a] concerne les représentations et les algorithmes s'appliquant à des solides. Elle propose principalement trois types de représentations :

- les partitions ou subdivisions d'espace, en particulier les arbres octaux (*octrees*) [Jackins *et al.*, 1980 ; Mäntylä, 1988 ; Samet, 1990] ;
- la géométrie solide constructive, ou approche booléenne [Requicha *et al.*, 1980 ; Mäntylä, 1988], qui consiste à assembler de simples primitives géométriques (cubes, sphères, cylindres, etc.) par les opérations d'intersection ( $\cap$ ), d'union ( $\cup$ ) et de différence ( $\setminus$ ) ;
- les représentations par frontières [Braid, 1975], dans lesquelles les volumes sont définis par leurs limites (FIG. 2.2).

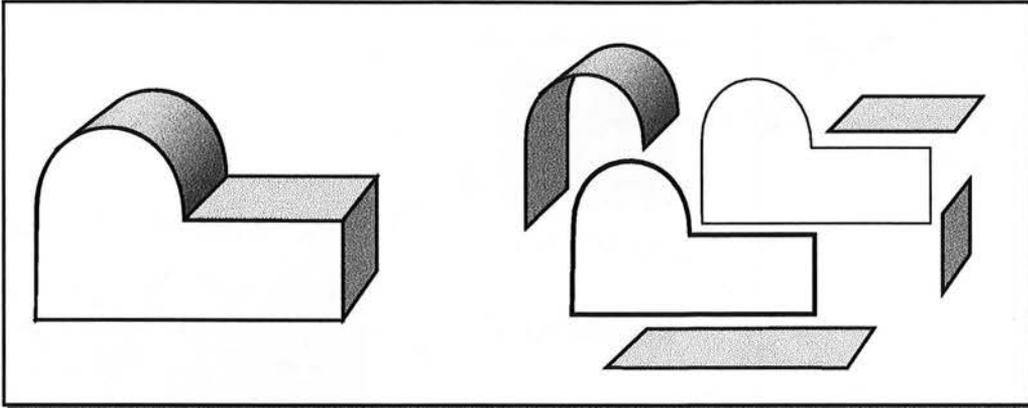


FIG. 2.2 – Principe de la modélisation volumique par frontières : le volume est représenté par ses faces, qui sont jointes suivant leurs bords.

Il existe également une quatrième famille de représentations, les décompositions cellulaires, ou grilles, qui feront l'objet du Chapitre 3. En pratique, les grilles sont utilisées pour effectuer des calculs (géostatistique, méthodes d'éléments finis), mais sont mal adaptées pour représenter la *géométrie* des solides de manière flexible. Elles sont ainsi rarement utilisées directement en modélisation volumique, mais plutôt générées à partir des autres représentations dans des applications d'ingénierie.

Sans entrer dans le détail de ces représentations <sup>1</sup>, les b-reps offrent une flexibilité et une précision plus élevées que les autres approches. En effet, les frontières de solides peuvent être représentées par de nombreux types de surfaces orientables.

De plus, dans le cas de modèles géologiques, leur construction à partir des données du sous-sol est conceptuellement aisée, car elle consiste à assembler entre elles les surfaces géologiques – horizons, failles, discordances, etc. – pour définir des volumes fermés.

Avant de présenter les principaux sujets de recherches sur les b-reps, nous allons à présent montrer leur importance dans le domaine des Géosciences en étudiant quelques-unes de leurs applications.

### 2.2.2 Applications en Géosciences

Depuis une quinzaine d'années, les b-reps sont régulièrement utilisées pour la modélisation informatique de structures géologiques. Cette partie passe en revue trois champs d'applications qui sont historiquement les moteurs du développement des b-reps en Géosciences.

<sup>1</sup> Pour une comparaison étendue des différentes représentations volumiques, voir [Foley *et al.*, 1997, Chap. 12]

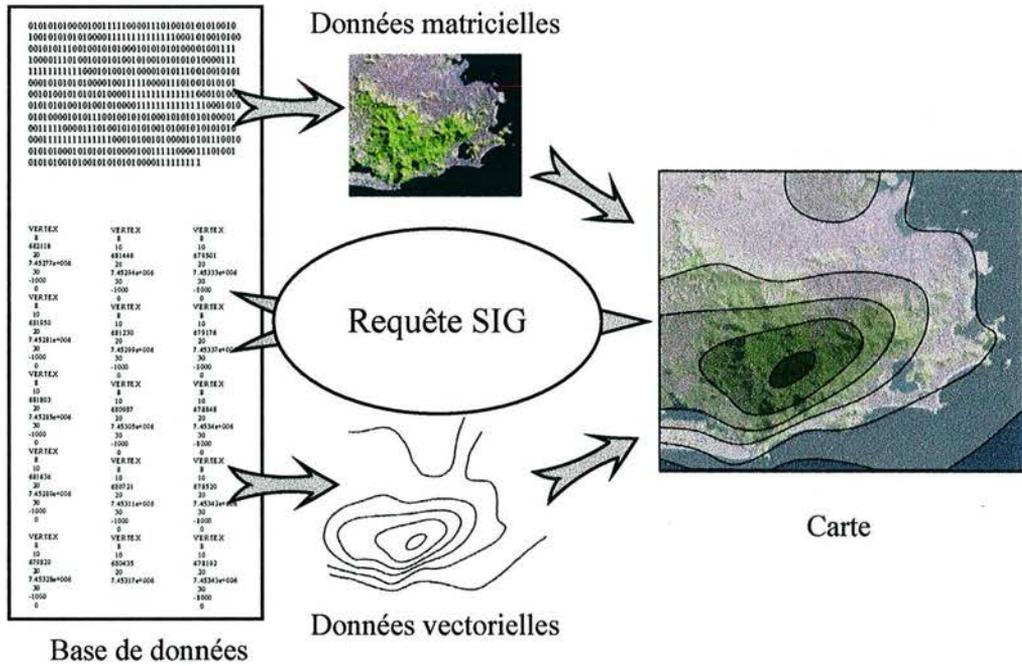


FIG. 2.3 – Principe des systèmes géographiques intégrés : à partir d'une base de données géographiques, une requête formulée par l'utilisateur permet d'obtenir l'information nécessaire pour générer rapidement une carte.

## Systèmes d'Information Géographique

Fondement des nouvelles techniques de cartographie numérique, les systèmes d'information géographique (SIG) sont couramment utilisés dans des domaines aussi divers que les sciences de l'environnement, l'urbanisme, l'agriculture, l'exploration minière ou les travaux publics. Ces systèmes utilisent en général un langage formel pour effectuer de requêtes sur des bases de données. Les informations ainsi sélectionnées sont ensuite utilisées pour produire des cartes (FIG. 2.3). Ces informations sont constituées essentiellement par :

- des images (représentation matricielle), relativement figées et peu flexible dans leur forme, leur précision et leur résolution ;
- des ensembles de polygones, de lignes et de points (représentation vectorielle), plus précis, facilement modifiables et interprétables.

Dans le domaine des Géosciences, les SIG ont de nombreuses applications [Bonham-Carter, 1994]. Cependant, les systèmes géographiques traditionnels en deux dimensions permettent difficilement de percevoir la nature tridimensionnelle des objets géologiques. L'extension des SIG à la représentation 3D est ainsi un sujet actif de recherche [Turner, 1992 ; Raper, 1989 ; Breunig, 1999 ; Apel, 2003].

Dans ces systèmes, le pendant des images 2D est constitué par des grilles régulières 3D, par exemple des cubes sismiques. Plus proches de l'interprétation et des applications, des données vectorielles peuvent également être disponibles. À ce titre, les b-reps, représentations vectorielles solides, sont très intéressantes car elles permettent [Kemp, 2002] :

- de travailler directement sur des volumes géologiques, sans subir de perte d'informations par rapport à une discrétisation de ces volumes en grille régulière ;
- de définir une carte 3D dans laquelle les légendes concernent les points, les lignes, les polygones et surtout les régions volumiques ; une carte 2D peut ainsi être vue comme une carte 3D d'épaisseur nulle.
- de combiner des informations hétérogènes (blocs structuraux, blocs définis par leur saturation en eau, huile ou gaz, galeries et autres cavités, etc.).

Pour un grand nombre d'applications des SIG, la modification interactive des modèles est une nécessité [Turner, 1992, Chap. 4]. En particulier, elle permet à l'utilisateur de contrôler la validité d'une représentation du sous-sol, et de la rectifier si nécessaire.

### Modélisation de bassin et fracturation

La modélisation de bassin est un autre domaine d'applications des b-reps. Les caractéristiques d'un système pétrolier dans un bassin sédimentaire sont conditionnées par l'évolution de ce bassin au cours du temps. La modélisation de bassin a pour ambition de reconstituer les variations temporelles des conditions de pression et de température ainsi que la déformation subie par les roches, qui conditionnent la maturation et la migration de la matière organique. Les équations aux dérivées partielles (EDP) qui régissent cette évolution sont généralement résolues par des méthodes de discrétisation, qui s'appliquent sur des maillages volumiques (cf. Chapitre 3).

Dans [Mello *et al.*, 1997], ces représentations cellulaires peu flexibles sont inféodées à une *description macroscopique* du sous-sol par une b-rep plus facilement modifiable. Chaque région de l'espace sert de cadre au maillage d'éléments finis. Lorsqu'une région est déformée (par le jeu d'une faille, le fluage d'une couche évaporitique), le maillage volumique est recalculé. Cette représentation permet également d'appliquer des calculs parallèles séparément sur chaque région. De nouveau, la modification du modèle b-rep est au coeur du système de modélisation de bassin utilisé.

### Traitement géophysique

Les premières utilisations de b-reps dans le domaine des Géosciences sont liées à des applications géophysiques [Gjøystdal *et al.*, 1985 ; Sword, 1991 ; Frøyland

*et al.*, 1993]. En effet, à l'issue d'une campagne de sismique réflexion, les géophysiciens disposent d'images du sous-sol en temps double, c'est-à-dire le temps qu'a mis le signal sismique pour se propager de l'émetteur au réflecteur, puis du réflecteur au géophone. Au cours du traitement de ces données [Claerbout, 1985 ; Henry, 1997], les vitesses de propagation sont utilisées pour la *migration* du signal sismique. Cette opération est essentielle dans l'interprétation sismique car elle permet d'obtenir une image plus fidèle de la géométrie du sous-sol, par diminution des effets de pendage et de courbure des réflecteurs et par traitement des phénomènes de diffraction. La migration profondeur permet en outre de convertir les temps d'arrivée en profondeur.

La justesse de la migration passe par la connaissance des vitesses de propagation des ondes dans le sous-sol. Naturellement, ces champs de vitesses ne sont pas connus directement, et doivent faire l'objet d'estimations. Pour cela, l'*inversion* est habituellement utilisée [Tarantola, 1987]. Dans ce processus itératif, un modèle de vitesses en temps est d'abord migré en profondeur. Ce modèle en profondeur permet de construire un signal sismique synthétique par tracé de rayon. La comparaison de ce signal avec les données réelles permet de réajuster le modèle de vitesse ou la géométrie des couches. Un nouveau cycle peut alors commencer.

Dans l'élaboration et l'ajustement du modèle de vitesses, la géométrie des interfaces géologiques en profondeur n'est généralement pas fixée. Il est donc important de pouvoir déplacer une limite de couche facilement tout en conservant la validité du modèle. Dans cette optique, plusieurs approches [Gjøystdal *et al.*, 1985 ; Guiziou *et al.*, 1990 ; Sword, 1991 ; Frøyland *et al.*, 1993 ; Euler, 1999] associent un modèle de vitesses analytique à des représentations par frontières. Dans ces modèles, une fonction mathématique est associée à chaque région de l'espace afin de décrire les variations de vitesse en fonction de divers paramètres. Par exemple, la vitesse peut être constante (dans le cas d'évaporites), augmenter linéairement avec la profondeur réelle ou relative à un réflecteur, etc. Des ajustements de modèles de vitesse sont également possibles par l'utilisation de tracé de rayon, qui permet par exemple de prendre en compte des variations latérales de vitesse au sein d'une couche [Guiziou *et al.*, 1996].

Dans [Wiggins *et al.*, 1993], une description du sous-sol par des b-reps est également utilisée pour représenter un champ de vitesses. Cependant, les vitesses ne sont pas définies analytiquement dans les régions du modèle, mais sur un maillage tétraédrique, ce qui rend leur persistance plus difficile à maintenir si des interfaces sont déplacées.

### 2.2.3 Problématiques de recherche

À travers trois applications, nous venons de voir en quoi les b-reps sont particulièrement attractives pour la Géomodélisation. Cependant, ces représentations présentent intrinsèquement un certain nombre de problèmes, et font ainsi l'objet de nombreuses recherches dans le domaine de la modélisation solide. Ces re-

cherches concernent principalement la définition de structures de données adaptées (§2.3), et le maintien de ces structures au cours de modifications (§2.4 et §2.5). De manière sous-jacente, ce dernier point fait appel à des considérations sur la *validité* des modèles volumiques.

En effet, si leur expressivité <sup>2</sup> est la force des b-reps, leur faiblesse réside principalement dans les difficultés rencontrées pour établir la *validité* d'un modèle [Sakkalis *et al.*, 2000].

### Validité représentative

En théorie [Requicha, 1980 ; Mäntylä, 1988], la validité représentative (*representational validity*) d'une b-rep est assurée si le ou les solides représentés sont *réalisables* dans un espace tridimensionnel. En d'autres termes, il convient de s'assurer que les éléments topologiques qui décrivent une b-rep (régions, faces, lignes et points) :

1. Ne se recouvrent pas. Cela implique que l'intersection de deux régions voisines est au plus égale à l'interface entre ces deux régions. Similairement, l'intersection de deux faces n'est possible qu'au niveau de leurs bords, etc. Cette condition, dite de *non-intersection* spécifie également qu'un élément ne peut pas s'auto-intersecter.
2. Aient une extension limitée dans l'espace ; c'est la condition de *finitude*.

Dans la pratique, la validité d'un modèle nécessite [Sakkalis *et al.*, 2000] :

- des mécanismes qui forcent *l'intégrité topologique* du modèle, comme les opérateurs d'EULER [Baumgart, 1975] ;
- des méthodes géométriques qui assurent une *concordance de la géométrie* avec les informations topologiques.

En particulier, lors de la construction d'un modèle b-rep, des intersections entre les frontières sont calculées afin de contraindre les bords des surfaces à être coïncidents. Dans cette opération, la validité topologique est relativement aisée à assurer. En revanche, la cohérence géométrique est plus délicate à obtenir. En effet, les opérations utilisant la géométrie sont en général coûteuses en terme de calcul et soumises à la précision numérique limitée des ordinateurs [Hoffmann, 1989b]. Afin de minimiser l'impact de cette précision limitée, l'élaboration d'algorithmes géométriques [Boissonnat *et al.*, 1995] s'appuie sur la *logique des prédicats*, qui consiste à consulter de manière exacte la base de données géométrique. Par exemple, des prédicats peuvent signaler si des arêtes du plan se recoupent. En revanche, leur point d'intersection ne peut pas être calculé exactement, ce qui est susceptible de générer des incohérences dans le modèle. Ainsi, les calculs

---

<sup>2</sup>Une représentation est dite expressive si elle permet de traiter une grande variété d'objets avec précision [Mäntylä, 1988, p. 51].

d'intersection s'avèrent très délicats à mettre en oeuvre, ce qui peut donner lieu à des modèles non valides.

### Validité géologique

Parallèlement à la validité représentative, des contraintes supplémentaires peuvent être formulées pour définir la validité géologique d'un modèle b-rep. Pour cela, la caractérisation du contexte géologique (style de déformation, type de roche, etc.) peut apporter des éléments intéressants. De manière plus générale, deux conditions minimales, facilement vérifiables à partir des informations combinatoires peuvent être proposées [Caumon *et al.*, 2003b].

**Première condition de validité :** les failles sont les seules interfaces dont les bords peuvent être libres (auquel cas le bord correspond à un déplacement nul des blocs). Si d'autres types d'interfaces, par exemple un horizon ou un contact huile-eau, ont des bords libres, la cohérence géologique n'est pas assurée. Il convient donc soit de supprimer les parties de surface contenant les bords libres (FIG. 2.4-A), soit de sceller ces bord contre d'autres surfaces (FIG. 2.4-B).

**Deuxième condition de validité :** Les limites de couches (horizons, surfaces d'érosion, ...) ne peuvent se recouper qu'en leurs bords *externes*. En d'autres termes, il ne peut y avoir d'intersection franche (« en X ») entre deux limites de couches.

Dans notre approche de modifications des b-reps (§2.5), nous considérons des modèles déjà construits et valides. Les modifications proposées se doivent donc de maintenir cet état des choses.

### 2.2.4 Bilan

Aujourd'hui, la plupart des logiciels intégrés de modélisation géologique disposent d'une représentation volumique par frontières [Assa *et al.*, 1998 ; Hubeli, 2002 ; Euler, 1999]. En effet, quel que soit le domaine d'application, les b-reps :

- peuvent représenter de manière compacte tous les types de structures géologiques ;
- permettent d'honorer correctement les données du sous-sol, en utilisant directement les surfaces résultant de l'interprétation ;
- sont précises ; en particulier, elles ne sont pas sujet à l'effet de moiré (*aliasing*) caractéristique des grilles régulières ou curvilinéaires ;
- permettent de définir des modèles analytiques de propriétés ;
- constituent une description macroscopique du sous-sol, qui peut servir en particulier pour définir des maillages pour les méthodes de discrétisation.

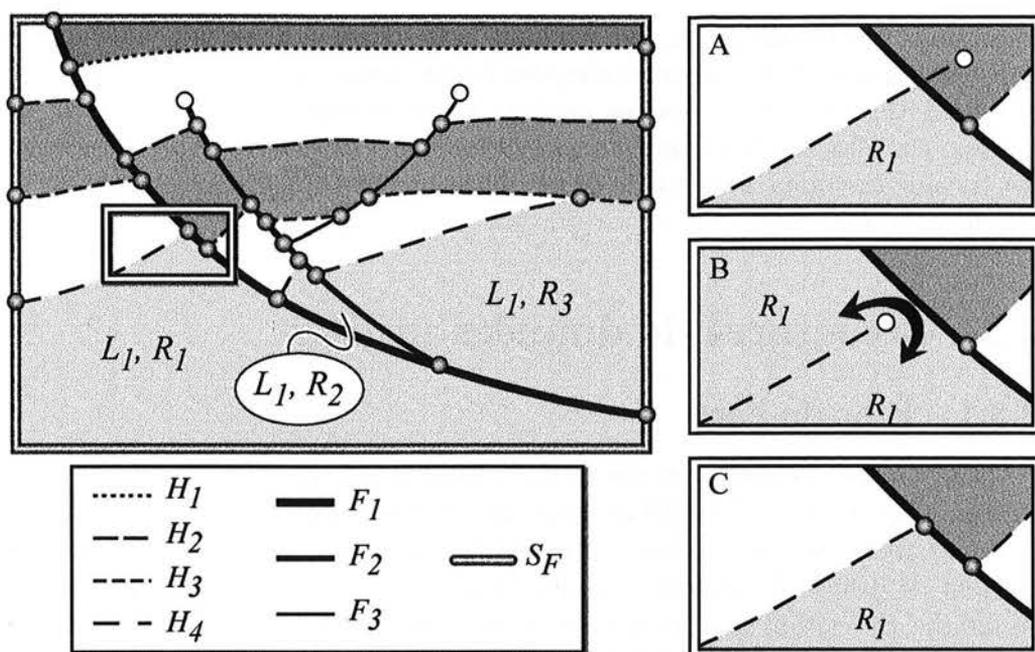


FIG. 2.4 – À gauche, un exemple de modèle géologique valide, constitué de limites de couches  $H_i$ , de failles  $F_j$  et d'une limite de domaine  $S_F$ . Les bords libres, figurés par les cercles blancs, sont légaux pour les surfaces de faille, mais indiquent des incohérences sur les autres surfaces (encarts A et B).

Cependant, aussi exactes que soient les b-reps, elles ne servent qu'à approximer la réalité à partir d'informations partielles. Une prise en compte optimale de ces informations passe ainsi souvent par plusieurs réajustements successifs d'un modèle. Dans certains cas, ces réajustements sont effectués automatiquement (par exemple, par un processus d'inversion), mais ils peuvent aussi faire intervenir interactivement un expert.

De tels mécanismes sont également intéressants pour la prise en compte de nouvelles données d'acquisition, lorsque celles-ci n'imposent pas une réinterprétation complète du modèle.

De plus, les modèles géologiques sont empreints d'incertitudes, dont il est important de quantifier l'impact avant une prise de décision (mise en exploitation, forage de puit, choix d'un type de fondation, etc.). Pour cela, une approche [Lecour, 2000] consiste à étudier l'incidence de perturbations aléatoires appliquées à un modèle de référence. Bien entendu, ces perturbations géométriques doivent maintenir la validité du modèle, ce qui nécessite des mécanismes de mise à jour des structures de données utilisées.

Dans tous les cas de figure, la modification d'un modèle existant est préférable à sa reconstruction. Plus rapide, elle permet en particulier de conserver des in-

formations non géométriques, par exemple la description analytique des vitesses dans chaque région. La fin de ce chapitre est organisée de la façon suivante. Les représentations informatiques des modèles b-reps utilisées sont tout d'abord présentées, car elles conditionnent les algorithmes mis en oeuvre. Après une revue des travaux existants, la méthode de modification développée dans cette thèse est ensuite présentée.

## 2.3 Structures de données utilisées

### 2.3.1 État de l'art

Dans un des premiers systèmes de modélisation volumique appliqué aux Géosciences, Gjølystdal *et al.* [1985] utilisent une représentation solide à base de surfaces paramétriques. Les surfaces, assemblées par des opérations booléennes, permettent de définir des régions volumiques auxquelles sont associées des vitesses sismiques. En évitant une évaluation explicite des intersections entre interfaces, cette représentation permet une modification rapide de leur géométrie, mais sans garantir l'invariance du nombre de régions. De plus, la résolution des modèles est limitée par l'usage d'une grille régulière pour vérifier leur validité.

Cette approche [Gjølystdal *et al.*, 1985] mise à part, la plupart des représentations par frontières [Baumgart, 1975 ; Weiler, 1986 ; Mäntylä, 1988 ; Rossignac *et al.*, 1990 ; Frøyland *et al.*, 1993 ; Lamboglia, 1994 ; Lévy, 1999 ; Grosse, 2002] sépare nettement géométrie (position des éléments dans l'espace) et topologie (relations de voisinage entre ces éléments), comme le montre la FIG. 2.5.

Brièvement <sup>3</sup>, les représentations combinatoires s'appuient sur une décomposition des modèles en 0-, 1-, 2-, et 3-cellules (respectivement sommets, arêtes, faces et polyèdres). Le graphe d'incidence (FIG. 2.5-B) est une manière intuitive de représenter les relations combinatoires entre cellules. Cependant, le stockage de cette information utilise des structures de données plus efficaces. La complexité de ces structures dépend en particulier de la *variété* ou *non variété* des objets à représenter par le système.

Un objet *n-variété* est un objet de dimension  $n$  dont tout point a un voisinage homéomorphe à un disque de dimension  $n$  <sup>4</sup>. Ainsi, une surface triangulée variété peut être représentée par un ensemble de triangles adjacents, chaque côté de triangle bordant au plus deux triangles. En revanche, dans une surface non variété, une arête peut être incidente à plus de deux triangles, ce qui nécessite

<sup>3</sup>Une description plus étendue des structures de données combinatoires pour les représentations par frontières est disponible dans [Bardis *et al.*, 1992]

<sup>4</sup>Un objet est homéomorphe à un autre objet si l'on peut passer de l'un à l'autre par une déformation continue. Ainsi, tous les objets de la FIG. 2.5-C sont homéomorphes les uns aux autres.

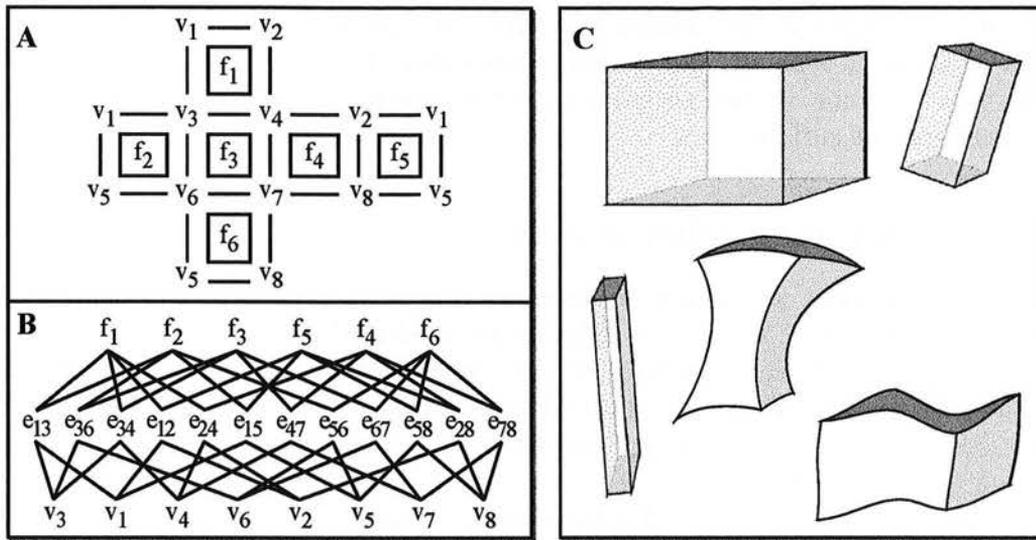


FIG. 2.5 – Représentations d'un objet : un objet peut être décomposé en cellules (A). Les relations de voisinage entre ces cellules (topologie) peuvent être représentées par un *graphe d'incidence* (B). Pour une même topologie, il existe plusieurs réalisations géométriques du modèle (C).

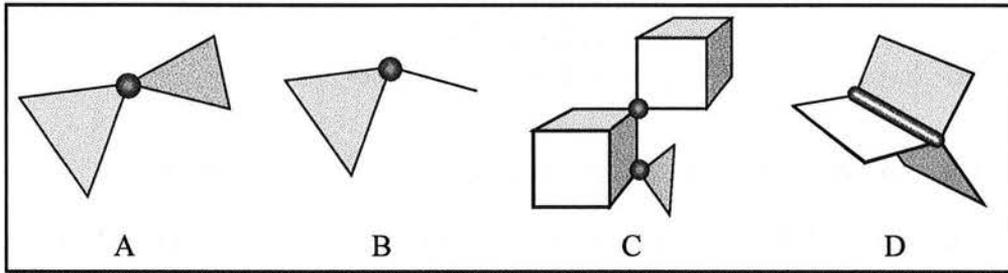


FIG. 2.6 – Exemples de configurations non variété. Les singularités non variété ponctuelles (A à C) ont peu d'intérêt en géologie. En revanche, les configurations non variété linéaires (D) sont courantes, par exemple au contact entre une surface de faille et une limite de couche.

des structures de données plus complexes, comme les arêtes radiales introduites par Weiler [1986] (FIG. 2.7, p. 57).

Dans le cas de modèles géologiques, les structures non variété que l'on souhaite représenter sont relativement restreintes (FIG. 2.6). Nous manipulons en effet des surfaces, qui sont au contact les unes des autres par des lignes (bord d'horizon contre faille ou contre intrusion, etc.). Ainsi il est possible de considérer les surfaces non plus comme des objets 2D non variété mais comme les faces de polyèdres. C'est le principe des Cartes Généralisées Hiérarchiques [Lévy,

1999 ; Grosse, 2002]. Des contacts non variété de type surface contre ligne peuvent éventuellement avoir un intérêt pour représenter l'intersection d'une trajectoire de puit avec une surface géologique ; à une résolution suffisante, le forage est cependant une surface,

### 2.3.2 Représentation utilisée

Les structures de données des modèles b-reps utilisées dans ce travail s'appuient sur les arêtes radiales introduites par Weiler [1986]. Suite aux travaux de [Lamboglia, 1994], ces structures ne sont cependant pas employées pour représenter les surfaces triangulées du modèle. En effet, l'information non variété du modèle peut être représentée par des arêtes radiales greffées autour des bords des surfaces (FIG. 2.7).

Les composantes variété des frontières du modèle volumique sont quant à elles représentées par un système de demi-arêtes [Conraud, 1997], relativement similaire à DCEL (*Doubly Connected Edges List*) [Muller *et al.*, 1978 ; Preparata *et al.*, 1985].

Parmi les bords de ces composantes variété, on distingue les *bords externes*, correspondant aux terminaisons des surfaces, des autres bords, dits *bords internes*. Parmi les bords internes, nous reviendrons par la suite sur les *cicatrices*, qui sont dues à des contacts avec d'autres surfaces : continues géométriquement, les cicatrices sont constituées de deux bords co-localisés.

### 2.3.3 Extension de la structure : notion de contact

En complément des d'arêtes radiales définies au niveau des triangles des surfaces, une structure de plus haut niveau a été mise en place durant cette thèse, afin de regrouper plusieurs arêtes radiales contiguës et de même topologie. Un *contact*  $\mathcal{C}$  (FIG. 2.8) est ainsi défini par l'ensemble des arêtes radiales  $a_0, \dots, a_k$  telles que, pour  $i \in [0, k[$  :

- $a_i$  est contiguë à  $a_{i+1}$  ;
- le nombre d'éléments autour de  $a_i$  est le même que le nombre d'éléments autour de  $a_{i+1}$  ;
- l'ordre et l'orientation des surfaces est le même autour de  $a_i$  qu'around de  $a_{i+1}$ .

Le regroupement des arêtes radiales a plusieurs intérêts :

- il permet de détecter certaines fuites entre deux régions, par exemple lorsque deux arêtes radiales adjacentes ont des orientations contradictoires ;
- d'un point de vue logiciel, il simplifie et accélère les consultations topologiques de haut niveau (par exemple, connaître les deux bords d'une cicatrice de faille au niveau de son contact avec un horizon) ;

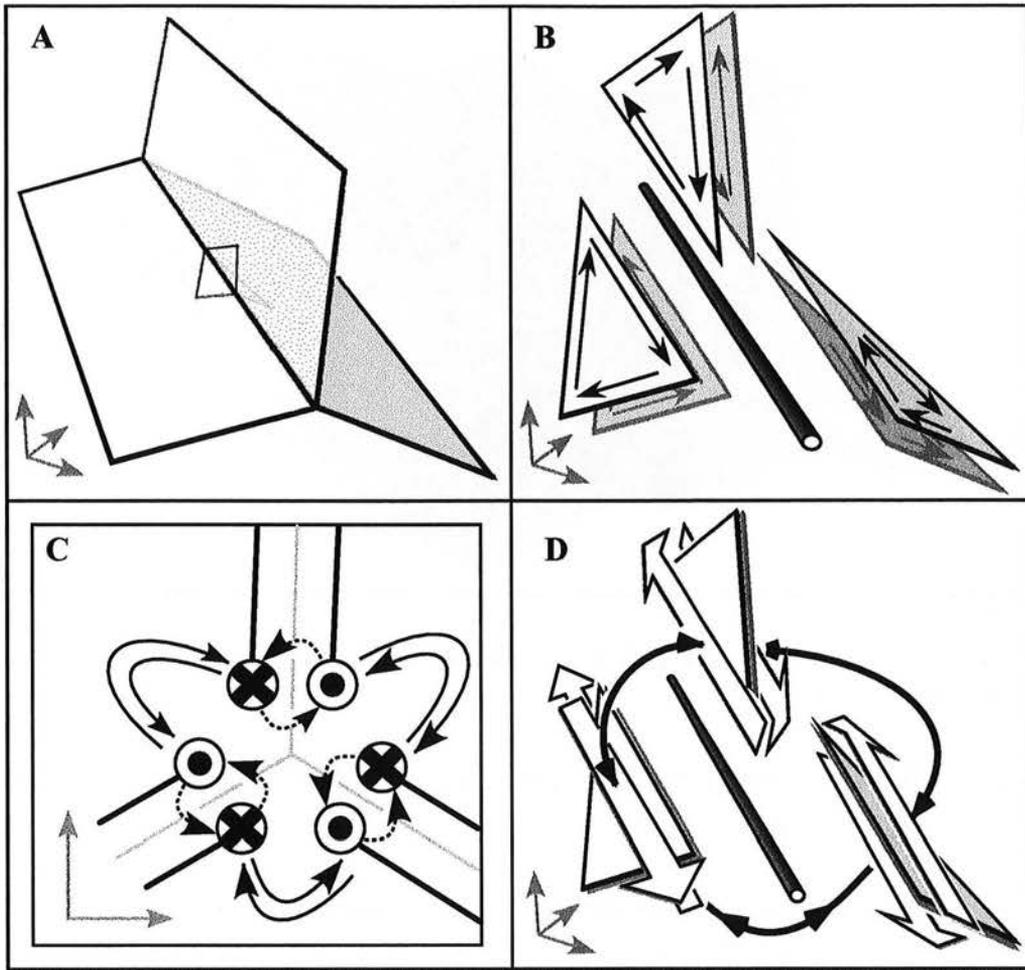


FIG. 2.7 – Représentation d'un contact non variété entre plusieurs surfaces triangulées (A). Dans une surface variété, chaque triangle peut être vu comme deux triangles orientés superposés et collés l'un à l'autre (B). La structure d'arêtes radiales [Weiler, 1986] se surimpose à la représentation variété pour lier les triangles par leur bord libre (C : vue en coupe; D : vue éclatée). Elle permet de passer d'un côté à l'autre d'une surface (flèches en pointillé), et de changer de face triangulaire dans une région polyédrique (flèches en trait plein).

- il permet de cataloguer les contacts par *types* comme dans [Lecour, 2000, Chap. 8];
- au niveau de l'interface utilisateur, il permet d'individualiser et de visualiser rapidement les contacts suivant leur type (faille/faille, horizon/faille, horizon/intrusion, etc.).

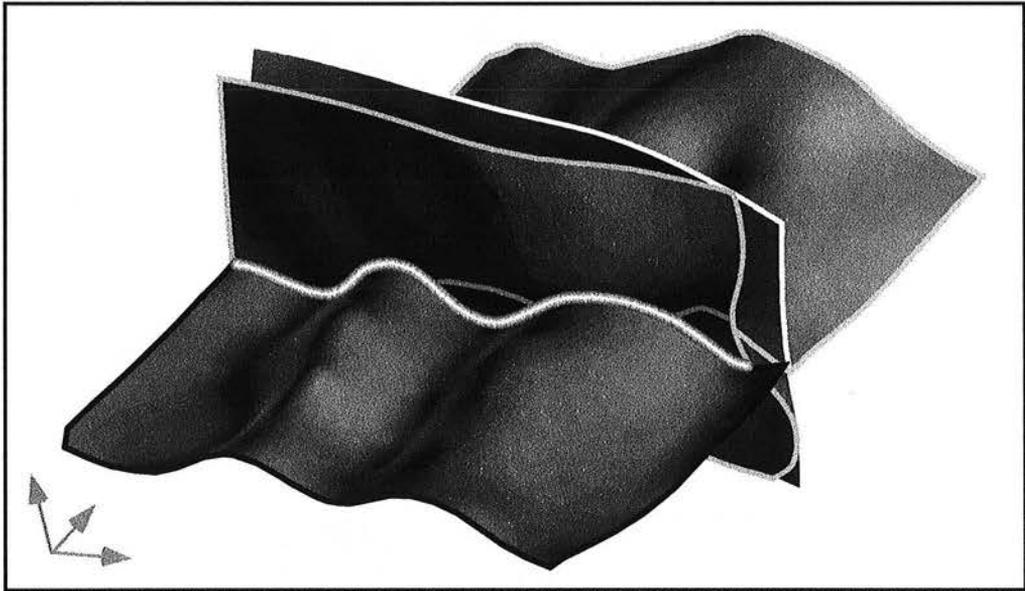


FIG. 2.8 – Un contact (surligné en dégradé) regroupe les arêtes radiales contiguës de même topologie.

## 2.4 Modification de modèles volumiques : état de l'art

Quelles que soient les structures de données utilisées, leur mise à jour lors de modifications de modèles volumiques est reconnue comme difficile [Raghothama *et al.*, 1998]. Cela est dû, d'une part à la relative complexité des structures mises en jeu, et d'autre part à la difficulté du maintien de la validité des b-reps. Avant de présenter l'approche mise en place dans ce travail, cette partie dresse un bilan des méthodes de modifications de modèles b-reps.

### 2.4.1 Opérations sur des modèles définis par frontières

#### Méthodes variationnelles et multi-résolution

Les capacités hétérogènes des ordinateurs, la complexité variable des codes de calculs justifie souvent de travailler sur plusieurs versions de résolutions différentes d'un même modèle. Pour cela, de nombreuses approches multi-résolution ont été définies en infographie pour traiter des surfaces 2-variétés. Dans [Hubeli, 2002], des méthodes sont définies pour décimer, subdiviser, lisser, et éditer des b-reps non variétés. Ces techniques permettent par exemple d'éliminer les variations haute fréquence de maillages non variétés. À l'inverse, il est possible de modifier les basses fréquences d'un modèle tout en conservant les détails haute fréquence,

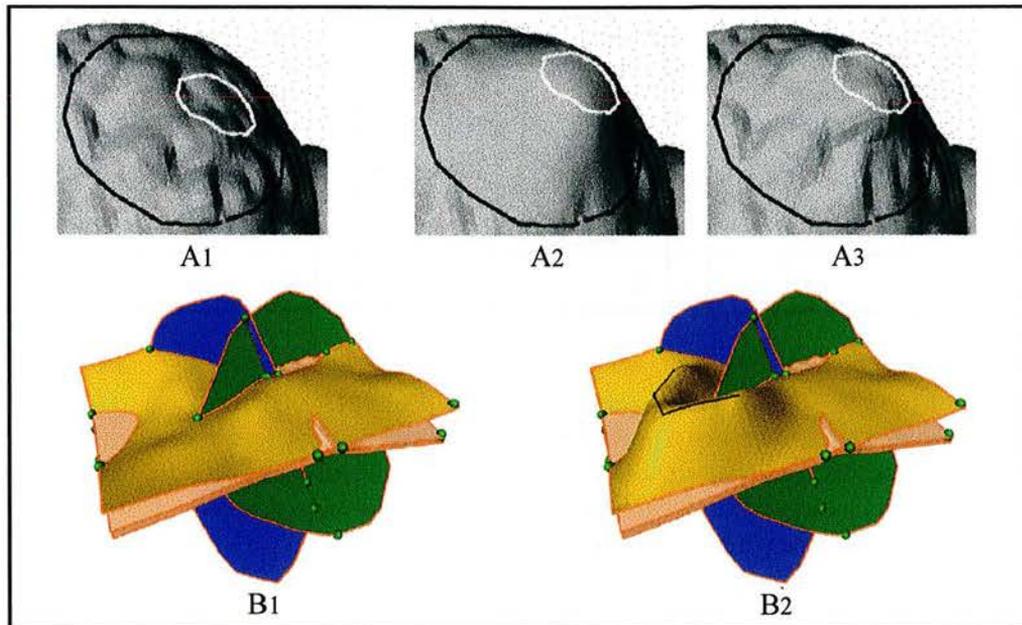


FIG. 2.9 – Exemple de modifications multi-résolution (images A et B issues respectivement de [Kobbelt, 2000] et [Hubeli, 2002], avec l'aimable autorisation des auteurs). Les méthodes variationnelles permettent modifier une surface (A1) en appliquant un filtre passe-bas (A2) ou bien en conservant les niveaux de détail (A3). Sur des modèles non variété (B), ce type de modification s'effectue à bords fixés.

comme dans [Zorin *et al.*, 1997 ; Kobbelt, 2000].

Intéressantes pour de nombreuses applications, ces approches variationnelles ont cependant l'inconvénient de s'effectuer à topologie constante. En effet, elles ne modifient par la définition des régions volumiques, et les contacts non variété entre surfaces sont figés.

### Mise à jour

Comme tous les modèles du sous-sol, les modèles volumiques sont amenés à évoluer lorsque de nouvelles informations sont disponibles. On peut ainsi vouloir ajouter, retirer ou déplacer une interface du modèle volumique. Pour cela, Euler [1999] ; Euler *et al.* [1998, 1999] propose un ensemble de méthodes pour :

1. Gérer le retrait d'une interface, en particulier en supprimant les cicatrices [Euler, 1999, Chap. 2], (FIG. 2.10). Après fusion des bords appariés, cette opération retire les noeuds fusionnés pour revenir au maillage avant découpage. Le maintien d'un modèle volumique valide nécessite toutefois de conserver les extrémités des cicatrices.
2. Permettre l'insertion de surfaces déjà découpées dans le modèle par l'opération de *découpage contraint* (cf. [Euler *et al.*, 1998] et [Euler, 1999, Chap.

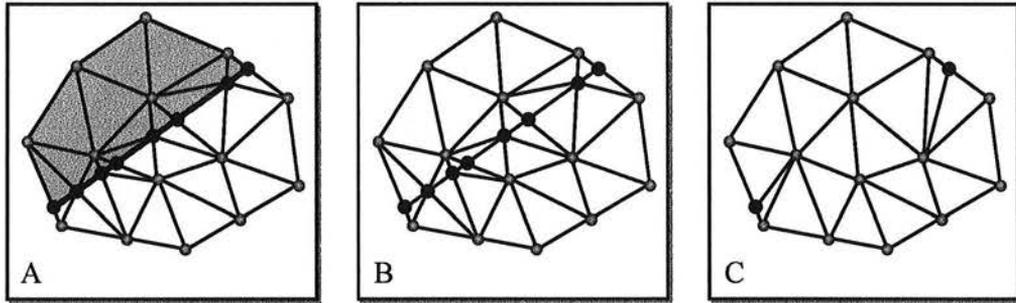


FIG. 2.10 – Suppression des cicatrices (d’après [Euler, 1999]). Sur une surface coupée par une autre surface (A), les cicatrices sont retirées par fusion des deux bords coïncidents (B), puis par retrait et retriangulation des noeuds liés à l’intersection, symbolisés en noir (C). Afin de garantir la continuité avec d’autres cicatrices, les noeuds du bord ne sont jamais retirés.

3]). Comme le montre la FIG. 2.11, la méthode projette le bord d’une surface contre une autre surface. Chaque noeud du bord est ainsi associé à une position sur un triangle cible. La ligne est ensuite raffinée entre ces deux triangles afin de se trouver exactement sur la surface cible <sup>5</sup>. Son maillage est ensuite inséré dans les triangles des deux surfaces pour générer un contact fermé.

3. Simplifier les bords d’interfaces contigus après une opération de découpage [Euler, 1999, Chap. 4]. Cette opération vise à réduire le nombre de triangles de faible *aspect ratio* <sup>6</sup>.

Du fait de la complexité des structures de données et des algorithmes vérifiant la validité des modèles b-reps, peu de méthodes ont été définies pour permettre leur modification en temps réel. Parallèlement, Wyatt *et al.* [1992] constatent que « [...] nous devons accroître nos efforts pour évaluer des approches nouvelles et originales afin de rendre [le] processus de modification plus satisfaisant et accessible aux êtres humains ».

Avant de présenter l’approche mise en oeuvre dans cette thèse, nous passons à présent en revue les méthodes existantes pour modifier des modèles volumiques.

<sup>5</sup>Euler [1999] utilise l’arbre de recouvrement minimal entre les deux triangles cibles pour trouver le raffinement de l’intersection. Dans le paragraphe 1.5.2 p. 33, nous avons étudié une méthode analogue pour échantillonner correctement le modèle stratifié par une coupe.

<sup>6</sup>L’*aspect ratio* caractérise la forme d’un triangle. On peut utiliser par exemple le rapport  $R$  du rayon du cercle inscrit sur le rayon du cercle circonscrit. Dans ce cas, la grandeur  $2R$  varie entre 1 pour un triangle équilatéral, et 0 pour un triangle dont les trois sommets sont alignés

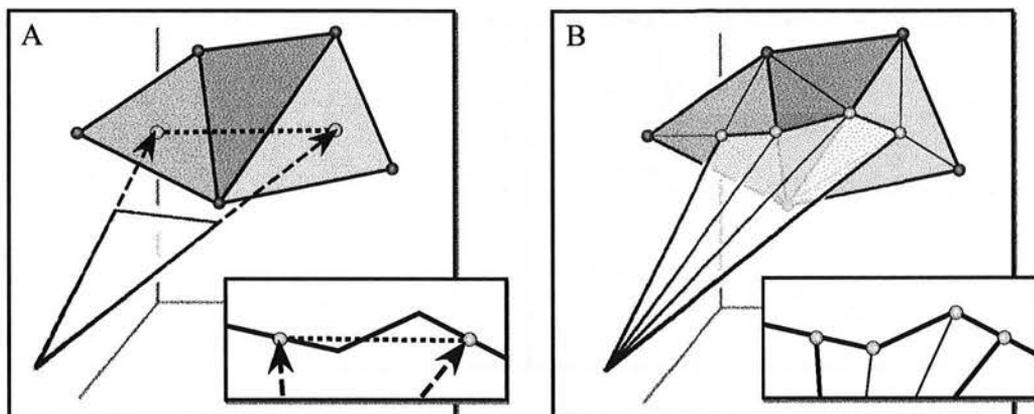


FIG. 2.11 – Découpage contraint de surfaces (d'après [Euler *et al.*, 1998]). Le bord de la surface à insérer dans le modèle est projeté sur la surface à découper (A). La ligne de projection (en pointillés) est raffinée pour se situer exactement sur la surface cible. La ligne peut ensuite être insérée dans le maillage des deux surfaces (B).

### 2.4.2 Déformation de modèles volumiques

Les déformations de modèles géométriques tridimensionnels font l'objet de nombreuses études en informatique graphique et en conception assistée par ordinateur. Deux grandes familles d'approches se dégagent de ces travaux :

- les déformations de forme libre, ou FFD (*Free-form deformation*) ;
- les méthodes utilisant des surfaces déformables.

#### Les déformations de formes libres

Ces techniques <sup>7</sup>, introduites par SEDERBERG et PARRY [Sederberg *et al.*, 1986], s'appuient sur une fonction de transformation d'un espace initial en un nouvel espace déformé. Intuitivement, l'espace peut être assimilé à un bloc de gelée contenant le modèle ; la déformation du bloc de gelée induit une déformation des objets qu'il contient. En pratique, l'espace est discrétisé en une grille régulière, et la fonction définie par un vecteur de déplacement en chaque noeud de la grille. Dans chaque cellule, la fonction est obtenue par interpolation polynomiale (spline tricubique par exemple). Cette définition formelle permet également d'utiliser différentes représentations de la fonction. Ainsi, différents types de grilles peuvent être utilisés en fonction des besoins [Coquillart, 1990].

Dans un contexte de modifications interactives, une résolution inverse du problème est couramment utilisée : à partir de contraintes de déplacement des objets eux-mêmes, l'espace discrétisé par la grille est déformé [Hsu *et al.*, 1992]. Ce type

<sup>7</sup>Pour une étude bibliographique plus poussée de méthodes de FFD, voir [Grosse, 2002].

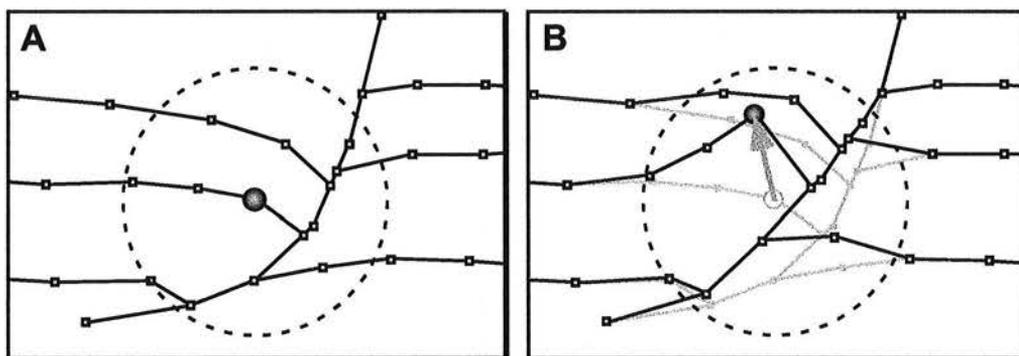


FIG. 2.12 – Exemple de déformation simple et contrainte [Borrel *et al.*, 1994]. Au début de la modification (A), le point sélectionné définit centre de la région d'influence. En fonction de leur distance à ce point, tous les noeuds de la région d'influence sont déplacés lorsque le point central est déplacé.

d'approche a été appliqué récemment à des modèles géologiques dans le cadre de l'inversion tomographique [Grosse, 2002]. Une autre variante des FFDs permet une modification *locale* d'un modèle. Dans ce schéma, dit déformations simples et contraintes, ou SCODEF (*simple constrained deformation*) [Borrel *et al.*, 1994], une modification est définie localement dans une région d'influence. le déplacement est défini par une fonction radiale décroissante du centre vers la limite de la région d'influence (FIG. 2.12-A).

Les méthodes de FFD sont très génériques, car elles s'appliquent sans distinction à tous les objets discrets par déplacement de leurs noeuds. Cependant, le comportement d'une FFD dans une modification de modèle géologique peut être indésirable. Par exemple (FIG. 2.12-B), il n'est sans doute pas souhaitable de modifier la géométrie d'une faille lorsque l'utilisateur déplace un horizon. Une FFD « par morceaux », c'est-à-dire qui ne s'appliquerait qu'à certains objets (l'horizon dans notre exemple), si elle permet un meilleur comportement géologique, ne permet pas en revanche de maintenir la cohérence topologique du modèle.

### Les surfaces déformables

La notion de surface déformable [Montagnat *et al.*, 2001] regroupe différentes représentations et approches pour modifier la forme de surfaces. En règle générale, ces méthodes font intervenir d'une part un modèle combinatoire et d'autre part un modèle physique. Le modèle combinatoire est utilisé pour conserver l'invariance topologique de la surface en cours de déformation. Le modèle physique permet quant à lui de gérer des problèmes sous-contraints par la minimisation de l'énergie de déformation, ou un critère analogue.

L'approche développée dans ce travail s'inscrit dans les méthodes de surfaces

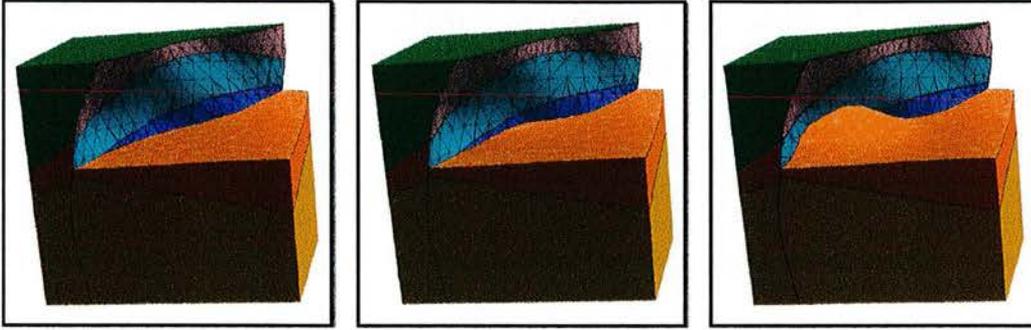


FIG. 2.13 – Principe de la modification proposée : une interface du modèle (l’horizon supérieur) est déplacée à l’intérieur des deux régions qu’elle délimite. Les contacts entre la surface déformée et ses voisines sont maintenus.

déformables. Comme nous allons le voir, elle s’applique en effet à des surfaces triangulées (modèle combinatoire), modifiées par des courbes paramétriques et par la méthode d’interpolation DSI [Mallet, 2002] (modèle physique).

## 2.5 Modification de contacts dans une b-rep

Dans l’évocation des méthodes existantes, nous avons vu que les modifications génériques comme les déformations simples et contraintes ou de forme libre [Borrel *et al.*, 1994 ; Grosse, 2002] ne sont pas toujours adaptées aux problèmes géologiques. Les approches conçues spécifiquement pour traiter séparément les interfaces d’une b-rep requièrent quant à elles d’importants calculs [Euler, 1999] ; elles sont donc difficilement applicables dans un contexte de modification en temps réel. Pour cette raison, nous avons développé une *approche locale* qui permet de modifier les interfaces d’une b-rep tout en conservant ses contacts [Caumon *et al.*, 2002, 2003b].

### 2.5.1 Approche proposée

#### Principe

Inscrite dans les approches de surfaces déformables [Montagnat *et al.*, 2001], la méthode mise en oeuvre (FIG. 2.13) consiste à *modifier séparément* les interfaces d’un modèle existant. Par rapport aux approches classiques de déformation de surfaces, des contraintes supplémentaires sont à prendre en compte pour maintenir la validité du modèle définie au paragraphe 2.2.3 p. 52.

**Contrainte 1** : le contact géométrique et topologique entre l’interface déformée et les interfaces adjacentes doit être maintenu (respect de la première condition de validité p. 52).

**Contrainte 2** : la surface déformée doit se déplacer uniquement à l'intérieur des deux régions qu'elle sépare (respect de la deuxième condition de validité p. 52).

Pour autant, il ne s'agit pas d'une modification à topologie constante. En effet, le maillage de certaines surfaces du modèle est modifié pour maintenir la fermeture des régions du modèle. De plus, la définition des deux régions affectées par le déplacement d'une interface peut varier.

Une autre spécificité de la méthode est de pouvoir raisonner *localement*. En effet, les modèles du sous-sol couramment utilisés comportent des interfaces constituées de plusieurs dizaines de milliers de triangles. De ce fait, sur les machines d'aujourd'hui, certains traitements de ces surfaces (interpolation, calcul d'intersections, etc.) sont difficilement compatibles avec une modification en temps réel.

En pratique, la méthode a été mise en place et testée dans le cas particulier de la modification de contacts. C'est en effet à ce niveau que peuvent se produire des artefacts dans les méthodes de construction du modèle. De plus, les contacts entre interfaces sont généralement peu marqués sur les données sismiques à cause des phénomènes de diffraction lors de la propagation du signal. Pour l'utilisateur, l'outil mis en place apparaît donc comme un moyen intuitif et efficace pour rectifier si nécessaire la géométrie des contacts.

## Méthodologie

Comme pour la modification interactive de cartes de proportions (§1.6), la déformation d'un contact s'effectue au sein d'une région d'influence ajustable par l'utilisateur. Notre approche se déroule fondamentalement en trois temps.

1. Au moment de la *sélection* d'un point du contact, la surface à déplacer et ses liens avec les autres surfaces sont identifiés. Toutes les initialisations nécessaires aux étapes suivantes sont effectuées.
2. Lors du *déplacement du curseur*, le bord et l'intérieur de la surface modifiée sont mis à jour par interpolation.
3. Lorsque la position désirée est atteinte, la topologie du modèle volumique est mise à jour.

Nous allons à présent décrire plus précisément les principales et les opérations intervenant au cours de ces étapes.

### 2.5.2 Définition d'une hiérarchie

Avant tout, il est nécessaire d'établir une hiérarchie entre les différents types de surfaces triangulées impliquées dans le contact. Cette hiérarchie sera ensuite utilisée pour déterminer une *surface de glissement*  $\mathcal{S}_g$ , dont la géométrie est fixée

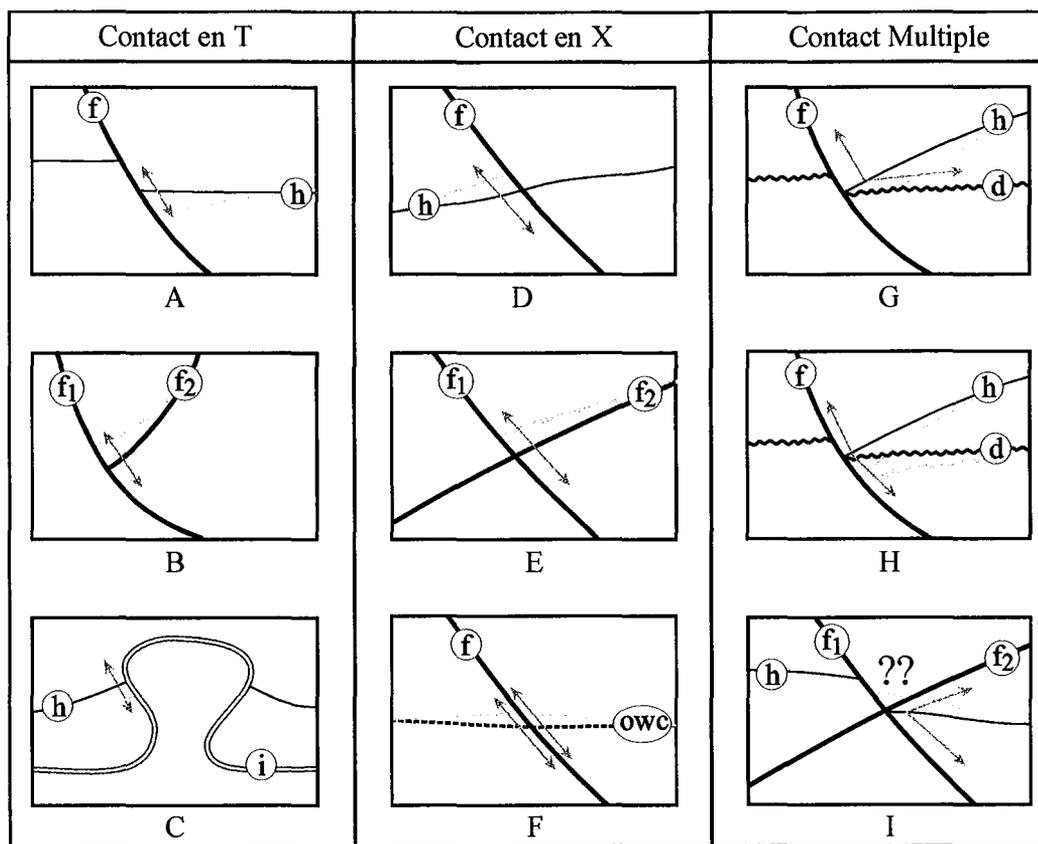


FIG. 2.14 – Types de contacts et hiérarchie associée. Dans chaque cas, la surface après déformation est indiquée en grisé ; et ses degrés de liberté sont indiqués par une double flèche. Les contacts font intervenir failles (f), horizons (h), discordances (d), surfaces intrusives (i) ou interfaces eau-huile (owc).

au cours de la modification, et une *surface déformée*  $\mathcal{S}_d$ . Elle permettra de définir un comportement prévisible pour l'utilisateur, et de maintenir la cohérence topologique et géologique du modèle.

Pour différentes familles de contacts rencontrées, une hiérarchie a été définie (FIG. 2.14). Dans tous les cas, la topologie du contact est utilisée en priorité. S'il y a lieu, les types et âges géologiques peuvent également servir à définir  $\mathcal{S}_g$  et  $\mathcal{S}_d$ .

Dans un premier cas de figure, deux surfaces  $\mathcal{S}_1$  et  $\mathcal{S}_2$  forment un contact « en T », de type  $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_1\}$ . Ce type de contact est rencontré fréquemment dans les modèles géologiques (FIG. 2.14-A, B, C) : horizon/faille, horizon/discordance, horizon/surface intrusive, faille-faille, etc. Dans ce cas, la topologie du contact impose *a priori* le glissement du bord de  $\mathcal{S}_2$  sur  $\mathcal{S}_1$ .

Deux surfaces  $\mathcal{S}_1$  et  $\mathcal{S}_2$  peuvent également former un contact « en X », de type  $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_1, \mathcal{S}_2\}$ . En pratique (FIG. 2.14-D, E, F), cette situation est courante au moment de la construction du modèle volumique : lorsque deux surfaces triangulées se recoupent géométriquement, leur intersection topologique est effectuée. Un contact en "X" apparaît donc si aucune opération n'est réalisée ultérieurement (interpolation, suppression d'une composante connexe, etc.). Dans ce cas, la hiérarchie est déterminée à l'aide du type géologique des interfaces. Par exemple, un horizon se déplacera le long d'une faille et non l'inverse. Lorsque deux failles se rencontrent, l'âge sera utilisé afin de faire glisser la plus ancienne le long de la plus récente.

Pour un contact « en X », la méthode doit permettre de maintenir ou non la continuité géométrique de la surface déformée  $\mathcal{S}_d$  pendant et après la déformation. Par exemple, la continuité est souhaitable lors du déplacement d'une surface de contact huile-eau, indépendamment des formations traversées (FIG. 2.14-F). En revanche, il est sans doute souhaitable de rompre cette continuité si l'utilisateur souhaite introduire un rejet sur une faille (FIG. 2.14-D, E). Pour cela, l'angle sous lequel le contact est visualisé est utilisé pour déplacer le bord visible de  $\mathcal{S}_d$  et laisser le bord caché par  $\mathcal{S}_g$  inchangé.

Parmi les configurations impliquant deux surfaces, les contacts « en  $\Gamma$  », de type  $\{\mathcal{S}_1, \mathcal{S}_2\}$ , associent deux bords externes de surface. Dans ce cas, très rare en pratique, les informations géologiques (nature de la surface et âge) peuvent être utilisées pour déterminer la surface de glissement.

Enfin, plus de deux surfaces peuvent être impliquées et former un contact multiple (FIG. 2.14-G, H, I). Dans ce cas, la topologie ou les types géologiques ne suffisent pas forcément à définir le comportement prévisible ou désiré par l'utilisateur. Dans le cas d'un contact « multi-T » [Lecour, 2000] (par exemple  $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_1\}$ ), une stratégie peut consister à déplacer de la même manière les bords de plusieurs horizons (ici,  $\mathcal{S}_2$ ,  $\mathcal{S}_3$  et  $\mathcal{S}_4$ ) suivant la même surface de glissement ( $\mathcal{S}_1$  dans notre exemple, voir aussi FIG. 2.14-H). Dans certains cas cependant, il n'est pas souhaitable de maintenir de tels contacts qui n'ont pas de raison d'être géologique. Pour cette raison, notre approche par défaut est de déplacer le bord d'une interface sur les deux surfaces de part et d'autre de cette interface (FIG. 2.14-G). Cette solution, si elle permet de résoudre l'ambiguïté géométriquement et topologiquement, n'est cependant pas idéale car elle risque d'invalider la cohérence géologique du modèle (FIG. 2.14-I).

Par souci de simplification, nous allons à présent nous attacher au cas d'un contact « en T » impliquant deux surfaces. Les adaptations de la méthode aux autres cas seront évoquées dans le paragraphe 2.5.9 p. 83.

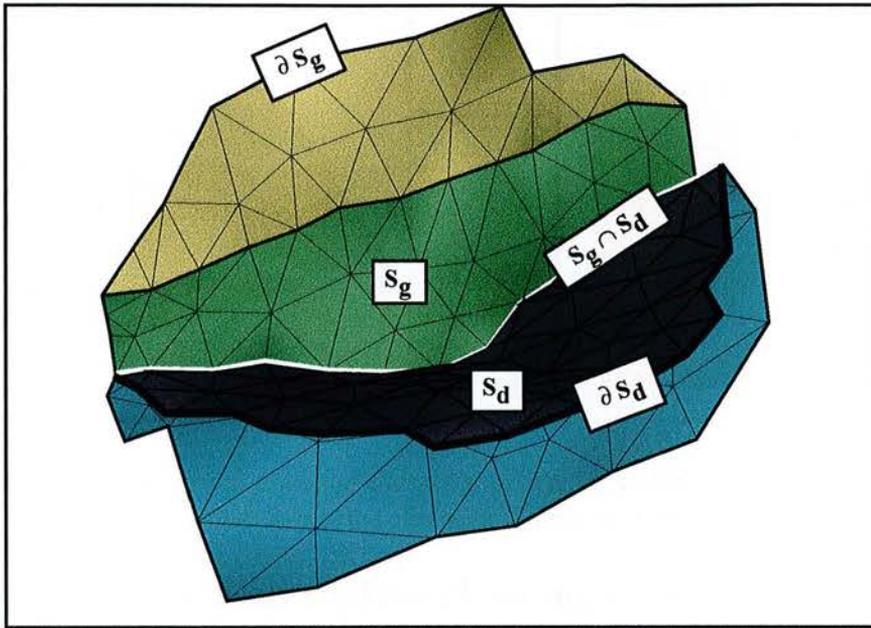


FIG. 2.15 – Les ensembles de la région d'influence : Les triangles, les arêtes et les noeuds sont classés suivant leur appartenance à la surface de glissement  $\mathcal{S}_g$ , la surface à déformer  $\mathcal{S}_d$ . Les bords  $\partial\mathcal{S}_g$  et  $\partial\mathcal{S}_d$ , limites de la distance d'influence ou bords libres des surfaces sont également identifiés. Un dernier ensemble regroupe les éléments du contact à modifier  $\mathcal{S}_g \cap \mathcal{S}_d$ .

### 2.5.3 Définition de la zone d'influence

En préalable à toute modification, il est nécessaire de définir le domaine d'intérêt. Cela implique (FIG. 2.15) :

- de séparer les surfaces autour du contact en sous-ensembles à l'aide d'une distance d'influence autour du point sélectionné ;
- d'identifier la surface  $\mathcal{S}_d$  qui sera déformée et la surface de glissement  $\mathcal{S}_g$ , qui va guider le déplacement de  $\mathcal{S}_d$  ;
- de détecter les bords  $\partial\mathcal{S}_d$  et  $\partial\mathcal{S}_g$  des surfaces  $\mathcal{S}_d$  et  $\mathcal{S}_g$  non impliqués dans le contact. Ces bords, libres ou limites de la zone d'influence, sont en effet fixés car ils garantissent la continuité de la zone modifiée avec le reste du modèle.

Comme toutes les opérations suivantes seront effectuées localement, les termes de « surface de glissement » et de « surface déformée » seront désormais employés pour faire référence à la partie de ces interfaces appartenant à la zone d'influence.

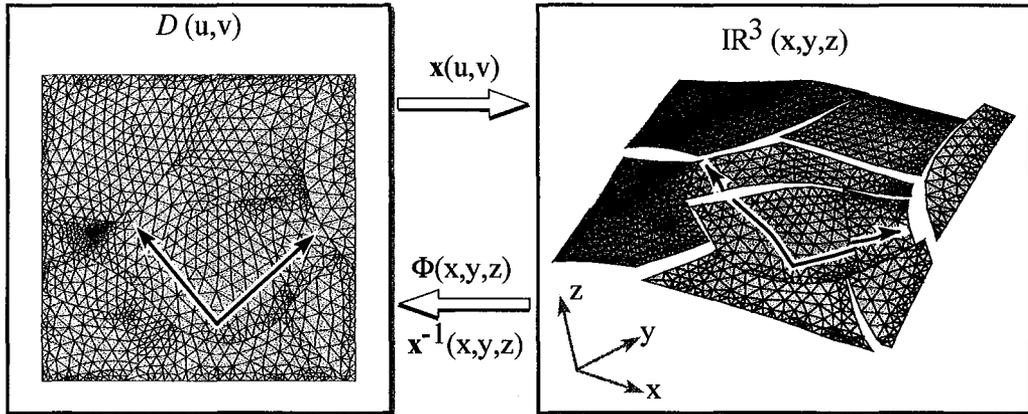


FIG. 2.16 – Paramétrisation  $\mathbf{x}(u, v)$  et paramétrisation inverse  $\Phi(x, y, z)$  mettent en relation un domaine du plan  $\mathcal{D}$  avec une surface  $\mathcal{S}$  définie dans  $\mathbb{R}^3$ .

#### 2.5.4 Paramétrisation de la surface de glissement

La modification d'un contact se doit de maintenir la fermeture des régions concernées. Pour cela, le bord de la surface déformée doit rester le plus près possible de la surface de glissement et le maillage de celle-ci doit être modifié. Pour cela, l'approche existante de retrait des cicatrices et de découpage contraint [Euler, 1999], n'est pas directement utilisable, car elle n'est pas définie localement.

Nous proposons donc une nouvelle approche, dans laquelle la surface de glissement est localement ramenée à un plan. Cela permettra par la suite :

- de déplacer facilement le bord de la surface déformée dans un espace en deux dimensions (l'espace paramétrique) ;
- de retriangler rapidement la surface de glissement pour tenir compte de la nouvelle position du contact.

Mathématiquement, ce passage d'une surface dans un espace tridimensionnel à un domaine 2D est appelé *paramétrisation*. Une surface triangulée  $\mathcal{S}$  sera dite paramétrisée si, et seulement si, il existe les deux fonctions  $\mathbf{x}$  et  $\Phi = \mathbf{x}^{-1}$  associant tout point de  $\mathcal{S}$  et son image dans un domaine  $\mathcal{D}$  du plan (FIG. 2.16) :

$$\mathbf{x} : \mathcal{D} \longrightarrow \mathcal{S}$$

$$(u, v) \longmapsto \mathbf{x}(u, v) = \begin{bmatrix} \mathbf{x}^x(u, v) \\ \mathbf{x}^y(u, v) \\ \mathbf{x}^z(u, v) \end{bmatrix}$$

$$\Phi : \mathcal{S} \longrightarrow \mathcal{D}$$

$$(x, y, z) \longmapsto \Phi(x, y, z) = \begin{bmatrix} u(x, y, z) \\ v(x, y, z) \end{bmatrix}$$

Pour être paramétrable, une surface traitée doit être homéomorphe à un plan. Dans notre cas, cela est garanti, même sur une surface fermée (lentille, sel), pour

peu que la distance d'influence soit suffisamment petite. Si ce n'est pas le cas, une paramétrisation par morceaux peut éventuellement être envisagée.

Le calcul d'une paramétrisation fait l'objet de nombreuses recherches en informatique graphique <sup>8</sup> [Floater, 1997 ; Lévy *et al.*, 1998, 2002]. Dernièrement la paramétrisation a également été appliquée dans le cadre de la restauration d'horizons plissés et faillés [Massot, 2002].

Pour cela, l'algorithme d'initialisation par projection défini dans [Massot, 2002] a été adapté pour travailler localement (Algorithme 2.1). Afin de s'abstraire des structures de données radiales du modèle volumique, il a également été repensé dans une approche générique. Pour cela, l'algorithme demande en tout noeud une position géométrique  $\mathbf{x}$ , un compteur  $c$ , un vecteur normal à la surface  $\mathbf{n}$ , un couple de valeurs  $u$  et  $v$  pour stocker les coordonnées paramétriques, ainsi que leur gradient  $\nabla u$  et  $\nabla v$ .

### 2.5.5 Modification interactive

Pour chaque déplacement du curseur, il est tout d'abord nécessaire d'assurer sa cohérence par rapport au modèle. Ensuite, le bord  $\partial\mathcal{S}_d$  de la surface  $\mathcal{S}_d$  peut être déformé et la surface  $\mathcal{S}_d$  mise à jour.

#### Contraintes sur la position du curseur

Comme nous l'avons vu pour la modification de modèles stratigraphiques (§1.6.1, p. 35), le curseur de la souris correspond à une ligne  $l$  dans l'espace de visualisation du modèle. Sa profondeur est donc indéterminée. Afin de lever cette indétermination, de nombreuses approches fixent un déplacement à profondeur constante, ou bien utilisent des éléments d'interface tridimensionnels fixant les degrés de liberté [Conner *et al.*, 1992 ; Wernecke, 1994 ; Gleicher, 1992]. Dans le cas de la modification de contact, le curseur doit idéalement se déplacer sur la surface de glissement. Pour cela, une première approche peut être de calculer l'intersection entre la ligne  $l$  et la surface de glissement par lancer de rayon (*ray casting*) [Foley *et al.*, 1997, p. 701],[Möller *et al.*, 1999, Chap. 10]. Cependant, une telle méthode ne permet pas d'obtenir un point valide si le curseur ne recoupe pas la surface. La mise en oeuvre du lancer de rayon est de plus spécifique aux objets utilisés comme contraintes. Pour cela, une seconde approche générique a été mise en oeuvre, en utilisant le principe du *magnétisme* [Bier, 1990].

Dans ce système (Algorithme 2.2), la position du curseur est contrainte par un ensemble d'éléments géométriques  $E$ . Ces éléments peuvent être de des segments, des triangles ou des constructions plus complexes comme des lignes polygonales ou paramétriques. Pour chaque type d'élément géométrique, la méthode requiert

---

<sup>8</sup> Les espaces paramétriques sont en effet utilisés pour de nombreuses applications comme le plaquage de textures [Lévy, 2001], la recherche de lignes géodésiques [Floater, 1997] ou encore la retriangulation adaptative de surfaces [Alliez *et al.*, 2002].

---

**Algorithme 2.1** Paramétrisation interactive (d'après [Massot, 2002]).
 

---

**Entrée :** $N$  : ensemble de noeuds (connexes) de  $\mathcal{S}_g$  $T$  : ensemble de triangles (connexes) de  $\mathcal{S}_g$  $n_{source}$  : un noeud de  $\mathcal{S}_g$  choisi comme point de départ **Début****Pour chaque** noeud  $n$  **de**  $N$ 

$$c(n) \leftarrow u(n) \leftarrow v(n) \leftarrow 0 \quad ; \quad \nabla u(n) \leftarrow \nabla v(n) \mathbf{0}$$

**Fin Pour**

$$c(n_{source}) \leftarrow 1$$

 $\nabla u(n_{source}) \leftarrow$  un vecteur unitaire localement aligné sur la surface

$$\nabla v(n_{source}) \leftarrow \nabla u(n_{source}) \times \mathbf{n}(n_{source})$$

 $P_{courante}$  : Pile de noeuds  $\leftarrow \emptyset$  $P_{suivante}$  : Pile de noeuds  $\leftarrow \emptyset$ **empiler**  $n_{source}$  sur  $P_{courante}$ **Tant que**  $P_{courante}$  **non vide** $n_{cour}$  : noeud  $\leftarrow$  **retirer\_tête**(  $P_{courante}$  )**Pour Chaque** triangle  $T$  autour de  $n_{cour}$  $\mathbf{n}_T$  : vecteur normal au triangle  $T$  $\mathbf{r}$  : vecteur  $\leftarrow (\nabla u(n_{cour}) \times \nabla v(n_{cour})) \times \mathbf{n}_T$  $\mathbf{g}_u$  : rotation de  $\nabla u(n_{cour})$  autour de  $\mathbf{r}$  d'un angle de arcsin  $\|\mathbf{r}\|$  $\mathbf{g}_v$  : vecteur  $\leftarrow \mathbf{g}_v \times \mathbf{n}(n_{cour})$ .**Pour Chaque** noeud  $n_i$  voisin de  $n_{cour}$  dans  $T$ 

$$u(n_i) \leftarrow \frac{c(n_i) \cdot u(n_i) + u(n_{cour}) + \mathbf{g}_u \cdot [\mathbf{x}(n_{cour}) - \mathbf{x}(n_i)]}{c(n_i) + 1}$$

$$v(n_i) \leftarrow \frac{c(n_i) \cdot v(n_i) + v(n_{cour}) + \mathbf{g}_v \cdot [\mathbf{x}(n_{cour}) - \mathbf{x}(n_i)]}{c(n_i) + 1}$$

$$\nabla u(n_i) \leftarrow \frac{c(n_i) \cdot \nabla u(n_i) + \mathbf{g}_u}{c(n_i) + 1}$$

$$\nabla v(n_i) \leftarrow \frac{c(n_i) \cdot \nabla v(n_i) + \mathbf{g}_v}{c(n_i) + 1}$$

$$c(n_i) \leftarrow c(n_i) + 1$$

**Si**  $c(n_i) = 1$ **Empiler**  $n_i$  sur  $P_{suivante}$ **Fin Si****Fin Pour****Fin Pour****permuter**  $P_{courante}$  et  $P_{suivante}$ **Fin Tant que****Fin**

une méthode de projection, qui donne l'image d'un point quelconque de  $\mathbb{R}^3$  sur l'élément. Ainsi, l'image de tout point  $\mathbf{p}$  de la ligne de vue  $l$  peut être obtenue par un parcours de  $E$  pour en trouver l'élément le plus proche (étape 1). Afin de garantir la cohérence entre la position du curseur à l'écran et la position projetée

**Algorithme 2.2** Mouvement contraint du curseur

---

**Entrée :**  
 $E$  : éléments devant contenir le curseur  
 $l$  : ligne correspondant au curseur  
 $\mathbf{p} = (x_p, y_p, z_p)$  : position initiale du curseur

**Sortie :**  
 $\mathbf{p}_{im} = (x_{p_{fin}}, y_{p_{fin}}, z_{p_{fin}})$  : position de l'image du curseur sur  $E$

**Début :**  
 $dist_{prec}$  : float  $\leftarrow \infty$   
 $\mathbf{p}_{im}$  : point  $\leftarrow \mathbf{p}$   
 $progrès$  : booléen  $\leftarrow$  vrai

**Tant que**  $progrès$   
 // 1- Projeter le curseur sur l'élément le plus proche  
**Pour chaque** élément  $e$  de  $E$  :  
 $\mathbf{p}_{cur} = (x_{p_{cur}}, y_{p_{cur}}, z_{p_{cur}}) \leftarrow \text{projection}(\mathbf{p}, e)$   
 $dist$  : float  $\leftarrow \text{distance}(\mathbf{p}, \mathbf{p}_{cur})$   
**Si**  $dist < prev\_dist$   
 $\mathbf{p}_{im} \leftarrow \mathbf{p}$   
 $dist_{prec} \leftarrow dist$   
**FinSi**  
**FinPour**  
 // 2- Projeter le point obtenu sur la ligne de vue  $l$   
 $\mathbf{p}' \leftarrow \text{projection}(\mathbf{p}_{im}, l)$   
 $dist \leftarrow \text{distance}(\mathbf{p}, \mathbf{p}')$   
**Si**  $dist \geq prev\_dist$   
 $progrès \leftarrow$  faux  
**Fin si**  
**Fin Tant que**  
**Fin**

---

$\mathbf{p}_{im}$ , cette image est projetée orthogonalement sur la ligne de vue (étape 2). Ce processus itératif est répété jusqu'à la convergence.

Graphiquement, le comportement du système est illustré par la FIG. 2.17. Lorsque l'ensemble  $E$  est relativement normal à la direction de vue, la convergence est très rapide. Elle est plus lente lorsque les contraintes sont sub-parallèles à la direction de vue. En fonction de la position de départ, il est possible que le point projeté reste localisé au niveau d'un minimum local (FIG. 2.17-C); dans ce cas, un changement de la direction de vue permet facilement de déplacer le curseur jusqu'à la position désirée.

Afin de garantir la rapidité de la méthode, une variante de l'étape 1 de l'algorithme 2.2, utilise une recherche *locale* dans  $E$  à la place d'une recherche globale. Intéressante lorsque  $E$  contient une grande quantité d'éléments, cette approche

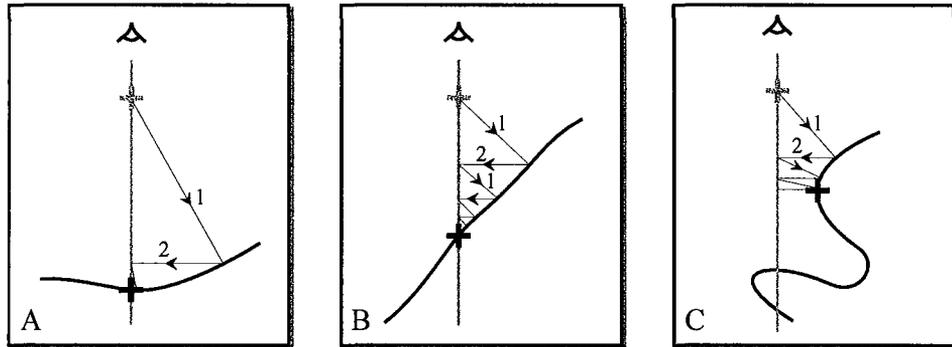


FIG. 2.17 – Processus itératif de mise en cohérence du curseur : exemple d'une ligne polygonale. La direction de vue est indiquée en grisé. Le nombre d'itérations dépend de la profondeur initiale du curseur et de l'inclinaison de la ligne de contrainte relativement à la direction de vue (A et B). Dans certains cas (C), la progression peut être bloquée par un minimum local, ce qui demande un changement de point de vue.

locale est néanmoins plus sensible aux minima locaux. De plus, elle impose une fonctionnalité supplémentaire aux éléments pour accéder à leur connectivité.

Dans son principe, cette mise en cohérence du curseur n'est guère différente du magnétisme [Bier, 1990]. La programmation générique permet en revanche de rendre cette méthode très intuitive et évolutive. Ainsi, il suffit de définir une méthode de projection pour généraliser le magnétisme à tout type d'élément géométrique.

Dans le cas qui nous intéresse, cette méthode a été appliquée pour contraindre le déplacement du curseur par la surface de glissement. L'ensemble  $E$  est donc constitué par les triangles de cette surface.

### Interpolation du bord $\partial S_d$

Pour chaque nouvelle position du curseur sur la surface de glissement, il convient de mettre à jour le bord  $\partial S_d$  de la surface déformée. Grâce à la paramétrisation de la surface de glissement, cette opération revient à déformer une courbe dans un espace à deux dimensions.

Ce problème classique en dessin et conception assistés par ordinateur est habituellement traité à l'aide de courbes paramétriques [Farin, 2001 ; Piegl *et al.*, 1997]. Dans ce travail, le bord à déplacer est approximé par des polynômes d'Hermite par morceaux. Chaque section de courbe est donc définie par ses extrémités  $\mathbf{p}_0$  et  $\mathbf{p}_1$  et les tangentes associées  $\mathbf{t}_0$  et  $\mathbf{t}_1$ . Un point  $\mathbf{p}$  de la courbe situé à une abscisse  $t \in [0, 1]$  est donné par :

$$\mathbf{p}(t) = (2t^3 - 3t^2 + 1) \cdot \mathbf{p}_0 + (t^3 - 2t^2 + t) \cdot \mathbf{t}_0 + (-2t^3 + 3t^2) \cdot \mathbf{p}_1 + (t^3 - t^2) \cdot \mathbf{t}_1$$

Au point de raccordement entre deux morceaux de courbe, l'égalité des tangentes assure une continuité  $C^1$ . En revanche, l'ajout d'un point de contrôle sur une courbe sans modifier les courbes voisines nécessite un réajustement des normes des tangentes, qui entraîne une continuité de type  $G^1$  entre deux courbes adjacentes.

Dans la mise en oeuvre de cette interpolation, seulement trois points de contrôle sont choisis par défaut, aux extrémités et au centre du bord de surface à déplacer. Cela permet une modification rapide par un simple mouvement de souris. Le point faible de cette simplification est son effet de filtre passe-bas : elle ne permet pas de représenter des ondulations à haute fréquence du contact. Dans la plupart des cas, ces variations ne sont pas forcément significatives et il est souhaitable de les éliminer. Cependant, dans le cas où l'utilisateur souhaite conserver ou ajouter des hautes fréquences, le système permet de raffiner le modèle interactivement en ajoutant des points de contrôle. Dans ce cas, il est nécessaire de les déplacer un à un pour obtenir la forme désirée.

En pratique, la méthode mise en oeuvre est suffisamment flexible pour obtenir le résultat désiré. Cependant, il serait intéressant de proposer un traitement plus élégant et automatique des aspects multi-résolution, par exemple en utilisant des ondelettes [Stollnitz *et al.*, 1995].

### Interpolation de la surface $\mathcal{S}_d$

Une fois son bord  $\partial\mathcal{S}_d$  déformé, la surface  $\mathcal{S}_d$  doit être interpolée. Par souci de simplicité, la méthode DSI [Mallet, 2002] a été choisie, mais d'autres approches utilisant par exemple des surfaces paramétriques [Farin, 2001] ou des approches multi-résolution [Kobbelt, 2000 ; Hubeli, 2002] pourraient être utilisées.

Quel que soit le schéma d'interpolation, la cohérence du modèle volumique impose que d'éventuelles interférences entre l'interface déformée et les autres interfaces du modèle soient détectées. Pour cela, de nombreux mécanismes de détection d'interférence sont décrits dans la littérature infographique [Lin *et al.*, 1998 ; Jiménez *et al.*, 2001].

### 2.5.6 Retriangulation de la surface de glissement

Au cours du déplacement, le contact du modèle volumique n'est pas conservé, car le maillage de la surface de glissement  $\mathcal{S}_g$  n'est plus concordant avec la nouvelle position du bord  $\partial\mathcal{S}_d$ . Pour résoudre ce problème, l'approche de retrait des cicatrices et de découpage contraint [Euler, 1999] est peu adaptée, car elle n'exploite pas les spécificités de la modification, en particulier son caractère local.

En pratique, la méthode suivante a donc été définie. Lors de la déformation, les noeuds de  $\mathcal{S}_g$  situés sur le contact sont déplacés en bloc avec le bord  $\partial\mathcal{S}_d$ . Afin de garantir le non-recouvrement des triangles incident à ces noeuds avec le reste de  $\mathcal{S}_g$ , une triangulation locale de la surface de glissement est effectuée.

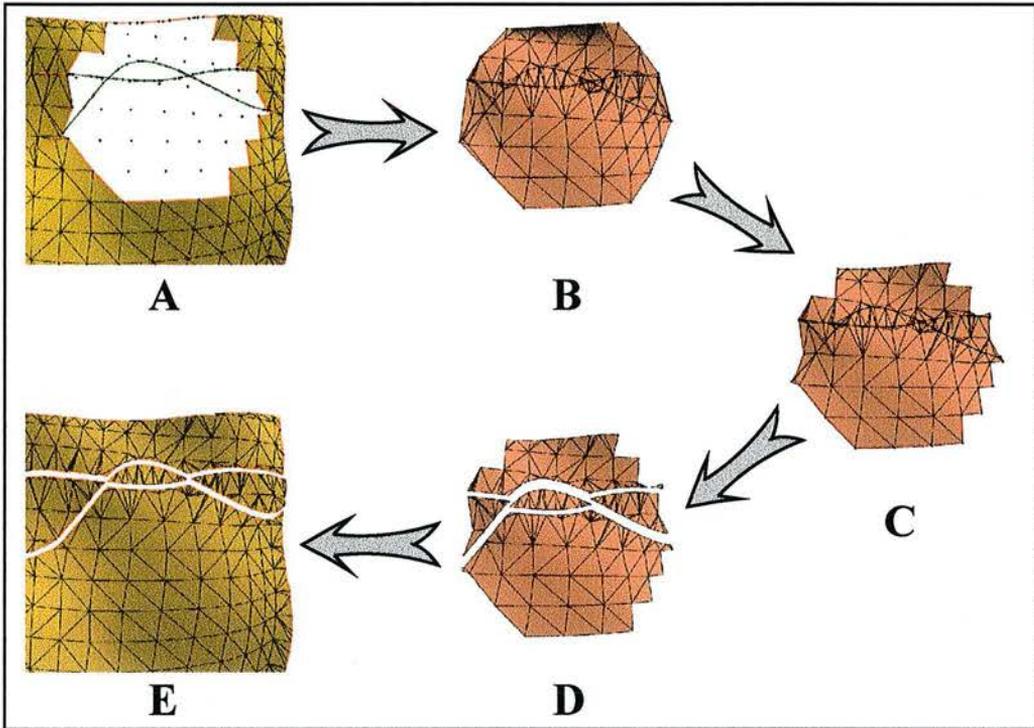


FIG. 2.18 – La mise à jour de la surface de glissement  $S_g$ . Après la destruction des triangles existants (A), une triangulation de Delaunay est effectuée sur la zone d'intérêt à partir des points initiaux de  $S_g$  et de ceux du contact déplacé (B). Les arêtes des cicatrices et des limites externes de  $S_g$  sont ensuite honorées (C), avant de joindre l'ensemble à la partie fixe du modèle (D et E)

En préalable à cette triangulation, le maillage de la surface de glissement  $S_g$  est bien entendu détruit (FIG. 2.18-A). De plus, les éventuelles intersections entre contacts sont insérées comme nouveaux noeuds de  $S_g$  (la gestion de ces intersections sera décrite en détail au paragraphe 2.5.7, p. 79).

La triangulation à proprement parler procède ainsi :

1. Calcul d'une triangulation initiale à partir de l'ensemble de noeuds  $\mathcal{N}_{S_g}$ , contenant les points initiaux de  $S_g$  et les points du contact à leur nouvelle position. Ce calcul est effectué dans le plan grâce à la paramétrisation de  $S_g$  (FIG. 2.18-B).
2. Permutations d'arêtes pour respecter les cicatrices et les limites externes de la zone modifiée (FIG. 2.18-C).
3. Retrait des triangles compris entre le bord de la zone et l'enveloppe convexe des noeuds  $\mathcal{N}_{S_g}$  (FIG. 2.18-C).
4. Découpage de la triangulation suivant les cicatrices et suture avec la partie non modifiée du modèle (FIG. 2.18-D et E).

Nous allons maintenant préciser les principales opérations définies pour la mise en oeuvre de cette retriangulation.

### Gestion des noeuds co-localisés

En deux dimensions, une triangulation peut être définie [Boissonnat *et al.*, 1995, Chap. 11] comme un complexe simplicial pur, connexe, et sans face singulière<sup>9</sup>.

Dans le cadre de notre travail, cette définition a deux conséquences :

- l'ensemble des points initiaux  $\mathcal{N}_{\mathcal{S}_g}$  doit être filtré afin de conserver un noeud et un seul pour chaque position géométrique (en pratique, les noeuds co-localisés se situent au niveau des cicatrices) ;
- la connexité de la nouvelle triangulation  $\mathcal{T}_{\mathcal{S}_g}$  devra être altérée pour maintenir des contacts valides dans le modèle volumique (insertion de cicatrices).

L'information combinatoire du modèle volumique est ainsi utilisée pour ségréguer un ensemble de *noeuds fantômes*  $\mathcal{N}_{\mathcal{S}_g}^f$ , qui seront ignorés par l'algorithme de triangulation. La création de cet ensemble (Algorithme 2.3, FIG. 2.19) se fait par deux itérations imbriquées autour des noeuds du bord de  $\mathcal{S}_g$ . L'information variété est utilisée pour obtenir les arêtes du bord au voisinage d'un noeud. L'information non variété (arêtes radiales, voir p. 56) permet quant à elle de traverser les cicatrices liées aux contacts.

Après application de l'Algorithme 2.3, les noeuds utilisés en entrée du calcul de la triangulation locale sont donc donnés par :

$$\mathcal{N}_{\mathcal{S}_g}^* = \mathcal{N}_{\mathcal{S}_g} \setminus \mathcal{N}_{\mathcal{S}_g}^f$$

Les noeuds fantômes sont quant à eux conservés en arrière plan, et seront utilisés pour reconstruire localement une surface  $\mathcal{S}_g$  en cohérence avec les contacts du modèle volumique.

### Triangulation de Delaunay

Parmi les nombreuses méthodes de triangulation définies en géométrie algorithmique [Boissonnat *et al.*, 1995, Chap. 12,17], la triangulation de Delaunay offre de bonnes performances en temps (pour  $n$  points, elle peut être calculée en temps proportionnel à  $n \log n$  [Boissonnat *et al.*, 1995, p. 438]), et possède des

---

<sup>9</sup>Un complexe simplicial  $\mathcal{C}$  de dimension 2 est constitué par un ensemble de triangles (2-simplexes), d'arêtes (1-simplexes) et de sommets (0-simplexes) tels que l'intersection de deux  $k$ -simplexes de  $\mathcal{C}$ ,  $k \in 0, 1, 2$ , est soit vide, soit égale à un  $(k - 1)$ -simplexe de  $\mathcal{C}$ . Il est *pur* si, et seulement si, tous les  $(k - 1)$ -simplexes de  $\mathcal{C}$  sont incidents à au moins un  $k$ -simplexe de  $\mathcal{C}$ . Une ensemble simplicial  $\mathcal{C}$  est *connexe* si, et seulement si, le graphe d'adjacence de ses triangles est connexe. Une *face singulière* est un simplexe dont le voisinage est non variété (voir p. 55).

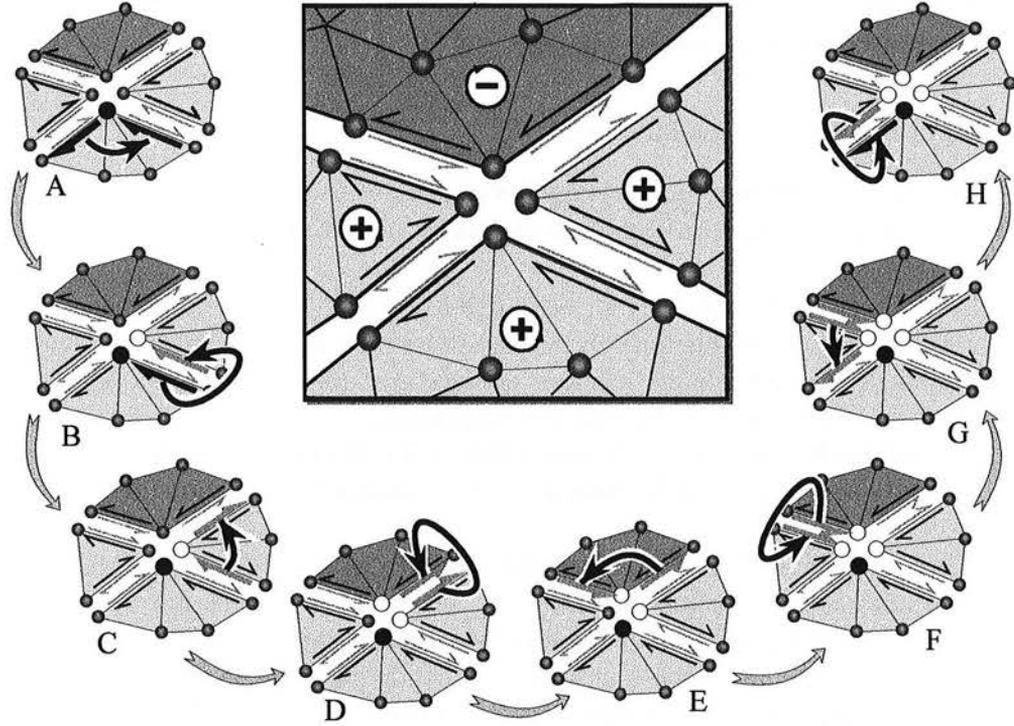


FIG. 2.19 – Recherche des points co-localisés de  $\mathcal{S}_g$  (vue éclatée). L'idée est de tourner sur  $\mathcal{S}_g$  autour de chaque noeud du bord. En utilisant les arêtes radiales, il est possible de traverser les cicatrices à l'autre (B,D,F,H); au sein de chaque secteur, la rotation utilise l'orientation de la surface (A,C,E,G). Au cours de la recherche le premier noeud (en noir) est conservé dans l'ensemble  $\mathcal{N}_{\mathcal{S}_g}$ . Tous ses successeurs (en blanc) sont déplacés dans  $\mathcal{N}_{\mathcal{S}_g}^f$ .

caractéristiques adaptées à la modélisation d'objets naturels [Conraud, 1997 ; Mello *et al.*, 1997].

Dans ce travail, cette triangulation est calculée dans l'espace paramétrique (§2.5.4, p. 68) à l'aide de l'algorithme itératif de Bowyer-Watson [Bowyer, 1981 ; Watson, 1981], implémenté par Conraud [1997].

Fondamentalement, la triangulation de Delaunay d'un ensemble de points  $\mathcal{P}$  du plan assure que l'intérieur du cercle circonscrit à chaque triangle ne contient aucun point de  $\mathcal{P}$ . Cette propriété a plusieurs conséquences pour notre approche de retriangulation locale :

- le bord de la triangulation de Delaunay  $\mathcal{T}_{\mathcal{S}_g}$  est l'enveloppe convexe des points  $\mathcal{N}_{\mathcal{S}_g}^*$ , alors que le bord de la zone retriangulée peut présenter des concavités ;
- la triangulation de Delaunay ne respecte pas forcément les contacts du modèle volumique.

---

**Algorithme 2.3** Recherche des noeuds fantômes sur la surface de glissement (algorithme simplifié)

---

**Entrée :**

$\mathcal{A}_{S_g}$  : arêtes radiales de la surface de glissement

**Sortie :**

$\mathcal{N}_{S_g}^f$  : noeuds fantômes de la surface de glissement

**Début**

**Pour chaque arête  $a$  de  $\mathcal{A}_{S_g}$  faire**

$v_0, v_1$  : noeuds  $\leftarrow$  extrémités de  $a$

**Pour  $i$  de 0 à 1 Faire**

**Si  $v_i$  n'est pas marqué**

**marquer  $v_i$**

// Voir FIG. 2.19-A,C,E,G

$a'$  : arête radiale obtenue en tournant autour de  $v_i$  sur  $S_g$

**Tant que  $a' \neq a$  Faire**

// Voir FIG. 2.19-B,D,F,H

$a' \leftarrow$  arête radiale située en vis-à-vis de  $a'$  sur  $S_g$

$v'$  : noeud  $\leftarrow$  extrémité  $i \oplus 1$  de  $a'$

**marquer  $v'$**

**ajouter  $v'$  à  $\mathcal{N}_{S_g}^f$**

**Fin Tant que**

**Fin Si**

**Fin Pour**

**Fin Pour**

**Fin**

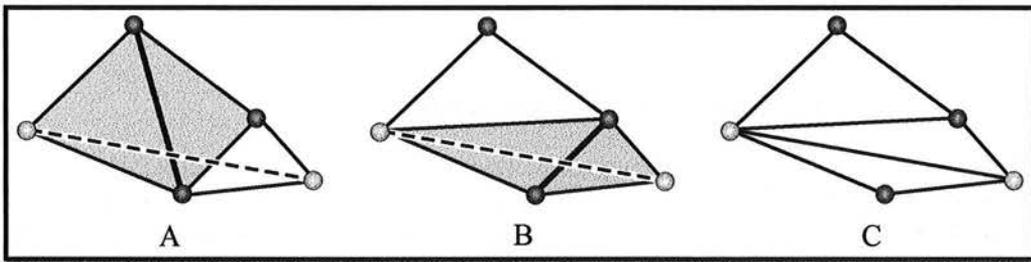


FIG. 2.20 – Une triangulation de Delaunay (A) ne respecte pas forcément toutes les arêtes des contacts (en pointillé). Obtenue par un basculement itératif d'arêtes (A, B), la triangulation contrainte [Chew, 1989] honore quant à elle parfaitement les contacts. (D'après [Conraud, 1997])

Un traitement supplémentaire, aboutissant à une *triangulation de Delaunay contrainte* [Chew, 1989] est donc nécessaire pour retrouver les arêtes des bords de  $S_g$  dans la triangulation  $\mathcal{T}_{S_g}$ . En résumé, cette opération consiste à détecter

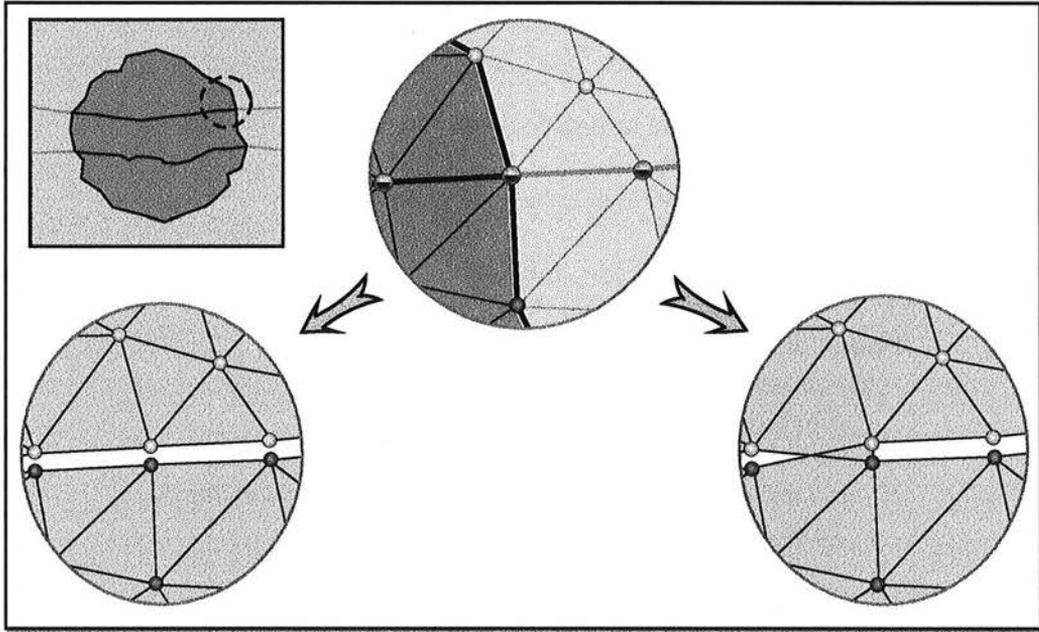


FIG. 2.21 – Les noeuds co-localisés peuvent poser des problèmes lors de l'appariement de la partie retriangulée avec l'extérieur.

les couples de points à connecter dans la triangulation originale, puis à effectuer itérativement des basculements d'arêtes (FIG. 2.20, [Conraud, 1997]).

Une fois la triangulation contrainte obtenue, les triangles de  $\mathcal{T}_{S_g}$  compris entre les limites de la zone d'influence et l'enveloppe convexe des points  $\mathcal{N}_{S_g}^*$  peuvent être éliminés (FIG. 2.18-C, p. 74).

### Jonction avec l'extérieur de la zone modifiée

Nous venons de voir comment une triangulation  $\mathcal{T}_{S_g}$  en conformité avec la nouvelle position du contact pouvait être obtenue. Par sa définition même,  $\mathcal{T}_{S_g}$  constitue un ensemble *connexe* de triangles. La mise à jour globale de la surface de glissement nécessite donc le clivage des arêtes  $\mathcal{T}_{S_g}$  situées sur un contact (FIG. 2.18-D, p. 74), ainsi que la suture de  $\mathcal{T}_{S_g}$  avec l'extérieur de la zone modifiée (FIG. 2.18-E, p. 74).

Le clivage d'une arête ne pose pas de problème particulier ; il fait partie des opérations classiques de modifications locales de maillages [Conreaux, 2001]. En revanche, la jonction avec l'extérieur peut donner lieu à des incohérences (FIG. 2.21). Il est en effet impossible de déterminer comment apparier certains noeuds de la frontière de la zone retriangulée à l'aide de critères géométriques.

Pour cette raison, une structure combinatoire, appelée *carte des coins*, a été conçue afin de stocker les relations de voisinage de certains noeuds co-localisés.

Dans la triangulation  $\mathcal{T}_{\mathcal{S}_g}$ , un *coin* est un noeud situé à la jonction entre au moins trois arêtes contraintes. La carte des coins identifie chaque secteur de  $\mathcal{S}_g$  par :

- un noeud  $n_c$  situé sur le coin, qui est soit le noeud de  $\mathcal{N}_{\mathcal{S}_g}^*$  référencé dans  $\mathcal{T}_{\mathcal{S}_g}$ , soit un noeud fantôme de  $\mathcal{N}_{\mathcal{S}_g}^f$  ;
- les voisins de  $n_c$  appartenant au même secteur de surface, et notés  $v_0(n_c)$  et  $v_{10}(n_c)$ .

En pratique, la définition carte des coins est inspirée des Cartes Généralisées [Lienhardt, 1993 ; Lévy *et al.*, 2002] (FIG. 2.22) :

- dans chaque secteur, deux *brins* liés par une involution <sup>10</sup>  $\alpha_1$  permettent d'accéder au noeud du coin et à ses voisins situés au bord du secteur ;
- de part et d'autre d'un bord interne de la surface  $\mathcal{S}_g$  deux brins sont connectés par une involution  $\alpha_2$ .

La construction et la lecture d'une carte des coins fait intervenir un algorithme analogue à celui de la détection des noeuds fantômes (FIG. 2.19, p. 76 et Algorithme 2.3, p. 77). Lors de la construction, le maillage de la surface et les arêtes radiales sont utilisés pour initialiser les relations entre brins ; après le calcul de la triangulation, l'application des involutions  $\alpha_1$  et  $\alpha_2$ , fournit une liste ordonnée des coins co-localisés ; la consultation de leur voisinage permet alors de connecter les noeuds de la surface  $\mathcal{S}_g$  correctement.

À l'issue de cette étape de retriangulation, le maillage de la surface de glissement est ainsi en conformité avec les bords des surfaces voisines.

## 2.5.7 Gestion des changements de macro-topologie

### Problèmes posés

Au cours du déplacement du contact, la topologie du modèle peut subir des modifications importantes (FIG. 2.23). En effet, le nombre, la disposition et les bords des différentes composantes connexes de la surface de glissement  $\mathcal{S}_g$  sont susceptible de varier. Dans certains cas, certaines intersections entre contacts de  $\mathcal{S}_g$  doivent donc être déplacées, ajoutées ou retirées.

### Calcul d'intersection des contacts

Le succès de la triangulation contrainte de  $\mathcal{S}_g$  impose que les arêtes correspondant aux cicatrices ne se recoupent qu'en leurs extrémités. Il convient donc d'insérer préalablement d'éventuelles intersections entre contacts dans la surface de glissement, c'est-à-dire en fait d'ajouter les cartes des coins correspondantes.

---

<sup>10</sup>Une involution est une fonction  $f$ , qui composée par elle-même, retourne l'identité ( $f \circ f \equiv Id$ )

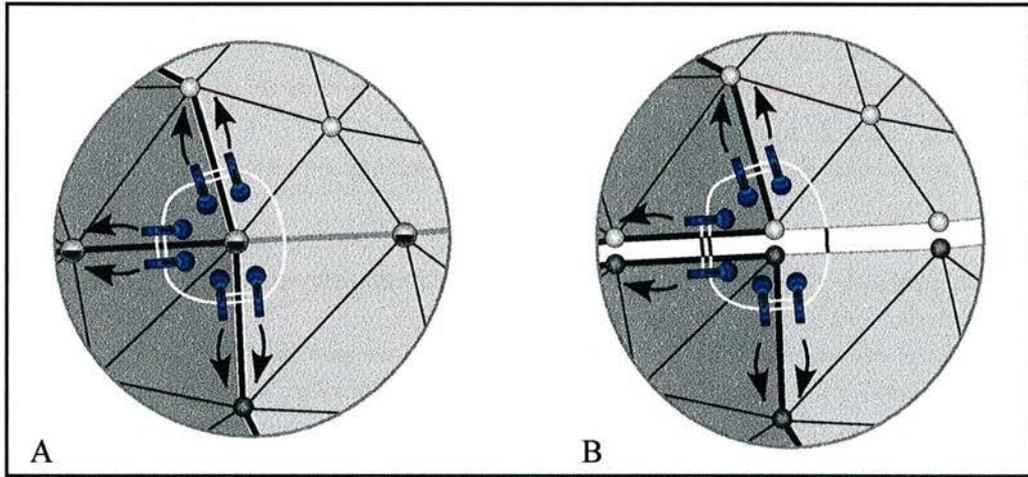


FIG. 2.22 – La carte des coins utilisée, en vue normale (à gauche) et en vue explosée (à droite). Chaque arête issue du coin est identifiée de manière unique par un brin (tête d'épingle). Les brins sont liés deux à deux par des involutions  $\alpha_1$  au sein du même secteur (simple trait blanc) et  $\alpha_2$  entre deux secteurs adjacents (double trait blanc). L'extérieur de la zone d'influence (en gris clair) est considéré comme un seul secteur.

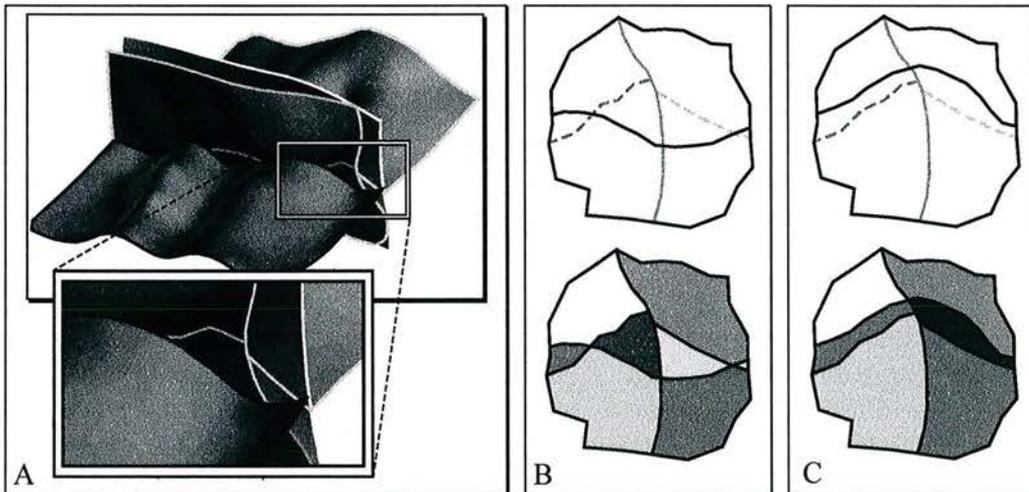


FIG. 2.23 – La modification peut entraîner des changements radicaux dans la topologie de la surface de glissement, en particulier apparition ou disparition de composantes connexes.

Grâce à l'espace paramétrique, cette opération revient donc à calculer les intersections entre un ensemble de segments du plan.

Ce problème classique en géométrie algorithmique [Hobby, 1999 ; Boissonnat *et al.*, 2000] a été abordé ici de manière générique, c'est-à-dire indépendamment de

<pre> <b>struct</b> Seg {     float extremités[2][2];     SubSeg* premier; }; </pre>	<pre> <b>struct</b> SubSeg {     Seg* parent;     SegInters* extrem[2]; }; </pre>
<pre> <b>struct</b> SegInters {     list&lt;SubSeg*&gt; étoile; }; </pre>	

FIG. 2.24 – Structure simplifiée pour stocker un arrangement de segments

la structure de données utilisée. Nous en donnons à présent une brève description.

Inspirée de l'algorithme de BENTLEY-OTTMANN, la méthode mise en oeuvre procède par balayage suivant l'abscisse  $x$  pour calculer dans un premier temps l'arrangement des segments <sup>11</sup> à l'aide de deux prédicats géométriques :

**intersect**( $s_1, s_2$ ) détermine si deux segments  $s_1$  et  $s_2$  s'intersectent ; il est construit autour du prédicat exact d'orientation de trois points dans le plan [Shewchuk, 1996] ;

**i\_avant<sub>x</sub>**( $s, s_1, s_2$ ) détermine si l'intersection de  $s_1$  avec  $s$  a une abscisse  $x$  inférieure à l'intersection de  $s_2$  avec  $s$ . Pour des raisons pratiques, ce prédicat est calculé en double précision, mais il est prouvé [Boissonnat *et al.*, 2000] qu'il peut être évalué de manière exacte.

Afin de minimiser les erreurs numériques, les intersections ne sont insérées que dans un deuxième temps. Les deux prédicats sont ainsi toujours évalués à partir des points initiaux, ce qui assure la cohérence de l'arrangement obtenu.

En pratique, la méthode utilise une structure combinatoire très simple pour stocker l'arrangement (FIG. 2.24). Chaque segment de l'arrangement (**Seg**) est construit à partir d'un segment initial. Il peut être subdivisé en plusieurs sous-segments (**SubSeg**), chacun délimité par deux intersections (**SegInters**). La liste ordonnée des sous-segments est également disponible pour chaque intersection.

Le balayage insère les segments un à un dans une liste de segments actifs (Algorithme 2.4). À l'étape courante, cette liste est amputée des segments situés à gauche du plan de balayage, correspondant à l'abscisse minimale du segment à activer. Les intersections détectées parmi les segments restants sont ensuite insérées dans l'arrangement par des créations de sous-segments, ordonnés autour de l'intersection. Lorsqu'un segment est recoupé à plusieurs reprises, le prédicat **i\_avant<sub>x</sub>** est utilisé pour ordonner les sous-segments correctement (fonction **subdiviser**).

<sup>11</sup>Un arrangement de segments est une structure combinatoire qui décrit l'union des segments sous la forme d'un graphe plan [Boissonnat *et al.*, 2000].

---

**Algorithme 2.4** Calcul de l'arrangement de segments dans le plan

---

**Entrée :** $\mathcal{E}$  : ensemble de segments du plan**Sortie :** $\mathcal{A}$  : arrangement des segments du plan (ensemble de Segs)**Début****trier**  $\mathcal{E}$  par valeurs d'abscisse minimale croissante. $segs\_actifs$  : liste des segments actifs  $\leftarrow \emptyset$ **Pour chaque** segment  $s$  de  $\mathcal{E}$     **Pour chaque** segment  $s'$  de  $segs\_actifs$         **Si** recouvrement(  $s, s'$  )            **Si** intersect(  $s, s'$  )                subdiviser(  $s, s'$  )        **Fin Si**    **Sinon**        Retirer  $s'$  de  $segs\_actifs$     **Fin Si**    **Fin Pour**        ajouter  $s$  à  $segs\_actifs$     **Fin Pour****Fin**

---

Une fois l'arrangement calculé, la position des intersections est évaluée et les noeuds correspondants sont créés. Dans sa mise en oeuvre actuelle, l'évaluation s'effectue en double précision, suffisante dans la plupart des cas. En théorie cependant, une telle subdivision de segments peut donner lieu à de nouvelles intersections non existantes dans l'arrangement initial. Il serait donc préférable d'utiliser pour ce calcul les méthodes plus robustes d'arrondi par magnétisme (*snap-rounding*) [Hobby, 1999].

**Retrait d'intersections**

Dans l'hypothèse où l'utilisateur effectue de nombreuses modifications successives, les insertions d'intersections se traduisent par un sur-raffinement du maillage des surfaces. Pour éviter cela, il est donc nécessaire de retirer les noeuds d'intersection correspondant aux bords déplacés avant la modification. Cette opération, purement combinatoire, identifie les groupes de noeuds au croisement du contact à déplacer et d'un autre contact, et les supprime par résorption [Conreux, 2001] (FIG. 2.25). Encore une fois, l'information contenue dans la carte des coins peut être utilisée à cet effet.

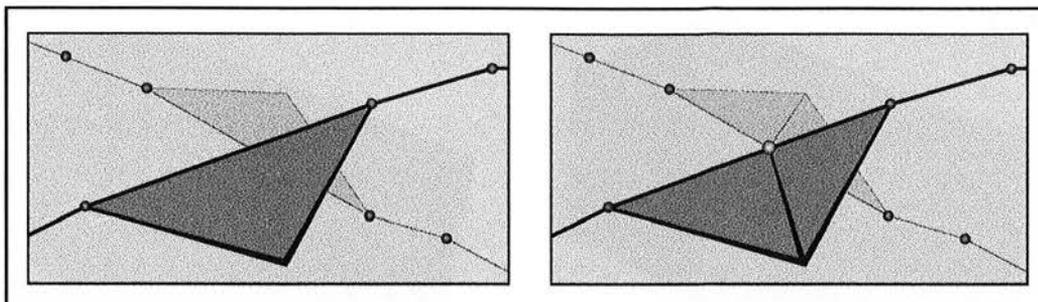


FIG. 2.25 – Les intersections de contacts sur la surface de glissement doivent être calculées et insérées dans le maillage des interfaces correspondantes. Réciproquement, les intersections entre le contact déplacé et les autres contacts doivent être retirées avant la modification pour éviter un sur-raffinement du maillage.

### Cohérence du modèle volumique

Lors de chaque opération d'insertion ou de retrait d'intersection, la cohérence du contact doit être conservée. Pour cela, les opérations analogues doivent être effectuées sur toutes les surfaces du contact (FIG. 2.25). Pour cela, l'information combinatoire contenue dans les arêtes radiales (§2.3.2 p. 56) est utilisée.

### 2.5.8 Mise à jour du modèle volumique

Une fois la surface de glissement mise à jour, ses contacts avec les autres surfaces du modèle peuvent être réévalués par un nouveau tri d'arêtes radiales.

L'invariance des régions au cours de la modification permet de maintenir si nécessaire la définition des couches et le modèle analytique de propriétés. En particulier, il est ainsi possible de maintenir la définition analytique des vitesses de propagation dans le cadre de la tomographie sismique.

### 2.5.9 Bilan

Dans la méthode proposée, la déformation d'un simple contact « en T » (FIG. 2.14, p. 65) s'appuie principalement sur la paramétrisation de la surface de glissement, qui permet d'une part de contraindre la déformation et d'autre part de mettre son maillage à jour efficacement.

Signalons toutefois que le type de déformation proposée offre moins de degrés de liberté que les contraintes 1 et 2 définies au paragraphe 2.5.1. En effet, les extrémités de contacts restent figées afin de maintenir la continuité entre les interfaces modifiées et les autres interfaces du modèle volumique. Plus généralement, la méthode offrirait plus de flexibilité si elle permettait le déplacement continu d'une interface le long de plusieurs surfaces de glissement adjacentes. Pour cela, il

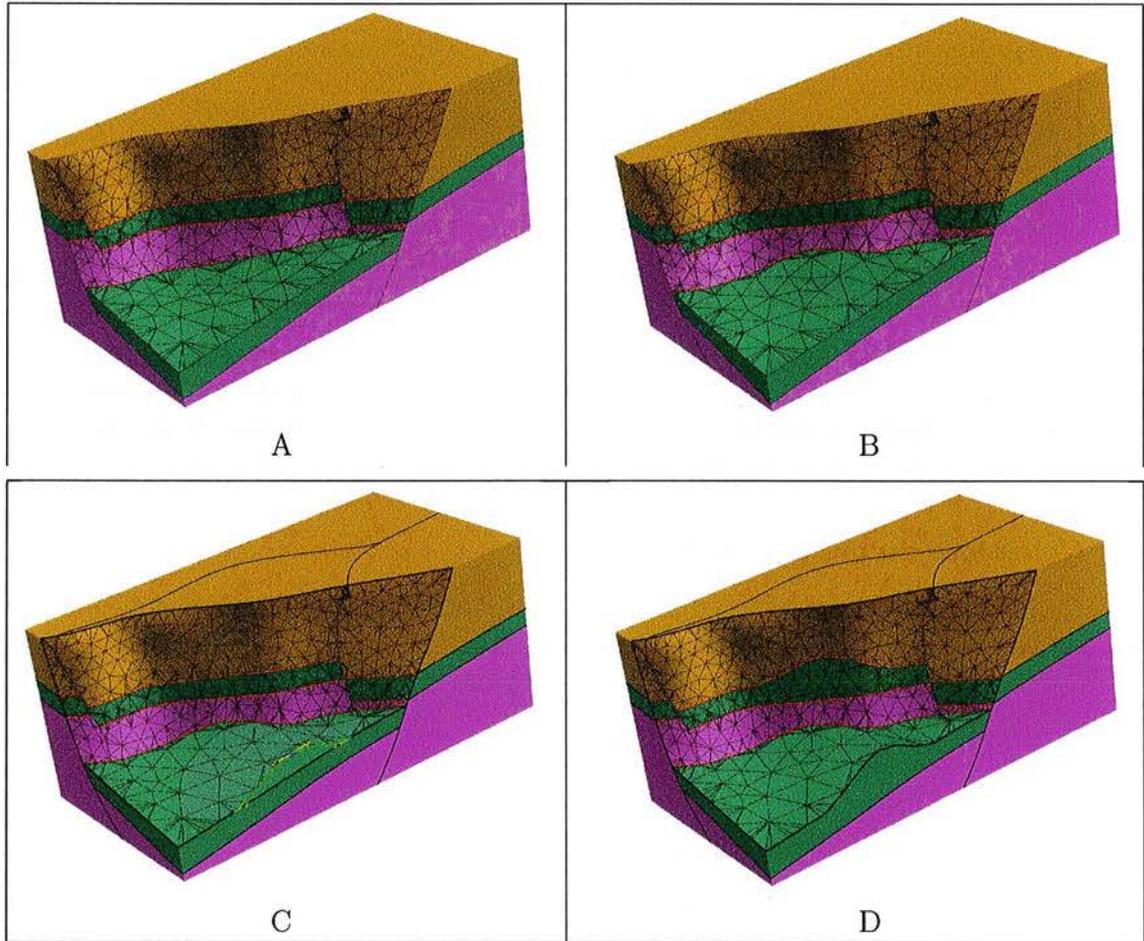


FIG. 2.26 – Modification de contacts sur un modèle synthétique (1).

est tout à fait envisageable de définir une paramétrisation continue sur plusieurs surfaces, et de les retriangler séparément. Le respect des cicatrices dans cette triangulation par morceaux assurerait en effet la validité du modèle obtenu.

## 2.6 Applications

La modification locale de b-reps a été appliquée à plusieurs modèles synthétiques (FIGS. 2.26 et 2.27). Dans tous les cas, les modifications sont réalisées en temps réel, avec des temps d'attente d'au plus quelques secondes.

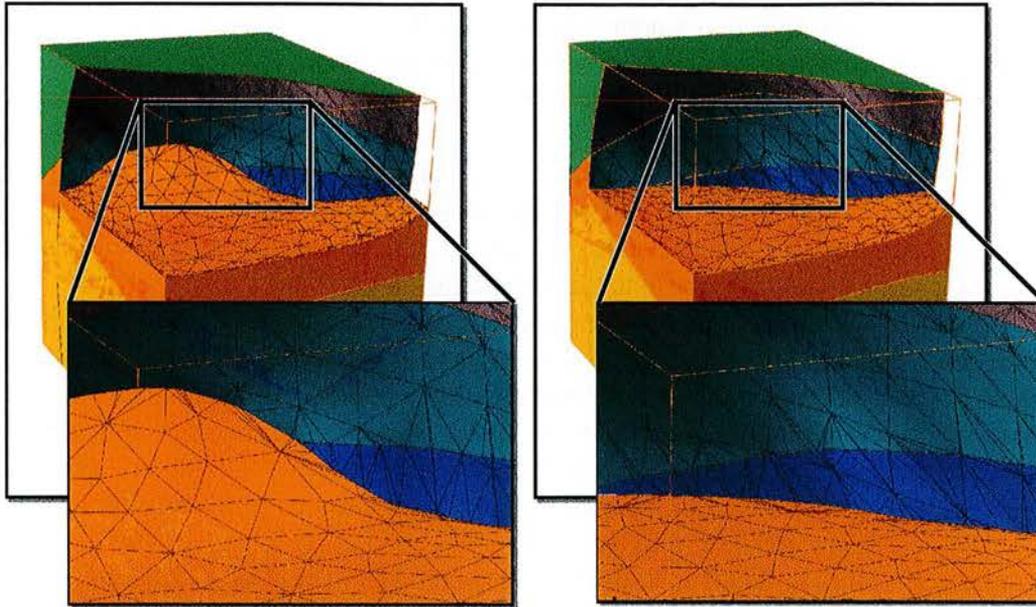


FIG. 2.27 – Modification de contacts sur un modèle synthétique (2).

## 2.7 Conclusion

Dans ce chapitre, nous avons étudié une description *continue* du sous-sol capable de représenter des géométries complexes. Cette description par frontières est utilisable en elle-même pour de nombreuses applications comme la tomographie sismique ou dans des systèmes géographiques intégrés.

Pour cette représentation, deux conditions minimales de validité géologiques ont été introduites. Ces deux conditions nous ont permis de spécifier des contraintes pour déplacer une interface dans un modèle tout en maintenant sa validité. La méthode développée pour respecter ces contraintes a été décrite. Sa principale originalité réside dans l'usage d'une paramétrisation des surfaces adjacentes à la surface déplacée. Cette paramétrisation permet en effet de maintenir les contacts géométriquement pendant la déformation. De plus, elle est utilisée dans la mise à jour des structures combinatoires du modèle volumique par retriangulation.

Par rapport au découpage contraint de surfaces [Euler, 1999], cette approche réduit considérablement le nombre de calculs d'intersections, ce qui améliore sa stabilité. En particulier, elle empêche l'apparition de fuites entre deux régions du modèle.

Les outils et concepts développés dans cette méthode font par ailleurs l'objet d'une industrialisation par la compagnie *Earth Decision Sciences* pour l'édition interactive de rejets de faille sur des modèles surfaciques.

Plusieurs extensions à ce travail sont envisagées. Tout d'abord, les contraintes

de validité géologique pourraient être systématiquement utilisées pour détecter d'éventuels problèmes lors de la création de ces modèles.

Concernant la modification elle-même, l'accent a été mis dans sa mise en oeuvre sur les aspects interactifs. Dans ce cadre, il s'est avéré que la simple visualisation des interfaces ou des régions du modèle n'était pas très ergonomique. Il serait donc appréciable de disposer de méthodes efficaces de visualisation par coupes transversales. Dans le Chapitre 3 (§3.3.3, p. 101), nous présentons un algorithme combinatoire qui pourrait être utilisé à cette fin.

Plus généralement, il serait souhaitable d'appliquer la méthode de modification dans d'autres contextes, où le déplacement n'est pas défini par l'utilisateur mais calculé. L'aspect local de la modification proposée pourrait être limitant dans ce type applications. Pour s'en affranchir, des résultats récents pour définir une paramétrisation par morceaux pourraient être utilisés [Lévy *et al.*, 2002].

Parmi les méthodes automatiques de modification, nous pouvons citer en particulier des algorithmes de mise en cohérence structurale [Samson, 1996 ; Léger *et al.*, 1997 ; Massot, 2002] ou le résultat l'inversion tomographique [Guiziou *et al.*, 1996 ; Grosse, 2002]. En fonction de ces applications, des contraintes supplémentaires pourraient être utilisées, comme la constance du volume, ou le respect de styles de déformation (cisaillement simple, glissement banc sur banc, etc.).

# Chapitre 3

## Modèles volumiques cellulaires

Les modèles représentés par extrusion (Chapitre 1) ou par frontières (Chapitre 2) permettent une *représentation continue* du sous-sol. Pour de nombreuses applications en Géosciences cependant, ces modèles sont peu utilisables tel quels. Une première limite provient de leur modèle de propriétés : une définition analytique dans les régions de l'espace, si elle est intéressante lorsque peu d'informations sont disponibles, ne permet pas facilement de respecter les données ponctuellement observées, par exemple au niveaux de puits de forage.

Une seconde limite est liée à la nature même des calculs que l'on souhaite effectuer sur les volumes. En effet, de nombreuses méthodes numériques (calculs de déformation, simulations d'écoulements), utilisent une résolution nécessitent une *discrétisation* des volumes en cellules élémentaires afin de pouvoir appliquer ces calculs.

Ces *représentations cellulaires*, qualifiées également de *grilles* ou *maillages* volumiques, font l'objet de ce chapitre. Ces structures de données sont sujettes à de nombreuses recherches en sciences exactes et appliquées, dans des disciplines aussi variées que les Géosciences, la Médecine, l'Aéronautique ou la Physique Nucléaire. Les enjeux associés à ces objets volumiques sont en effet variés : il s'agit de manipuler rapidement des volumes considérables de données, de les visualiser en temps réel de manière intuitive, de créer des maillages de géométrie et de topologie complexes, etc.

Dans ce travail, nous nous sommes particulièrement intéressés aux problèmes intimement liés de la représentation et de la visualisation de ces objets.

Dans une première partie (§ 3.1), ce chapitre présente les principaux enjeux des modèles cellulaires. Il est y rappelé en particulier que la variété des applications utilisant des grilles donne lieu à plusieurs catégories d'objets de complexité variable. Après un état de l'art des méthodes de visualisation volumique (§ 3.2), la deuxième partie présente un *cadre algorithmique générique* pour visualiser ces familles de grilles par des sections, des surfaces d'isovaleur ou du rendu volumique (§ 3.3). Enfin, la troisième partie (§ 3.4 et 3.5) présente des applications de cette méthode aux différentes familles de grille.

## 3.1 Enjeux et Variété des Modèles Cellulaires

### 3.1.1 Applications

De la géophysique aux simulations d'écoulement, les grilles volumiques interviennent dans de nombreuses opérations de Géomodélisation. À travers l'étude de quelques applications, cette partie présente les différents types de grilles utilisés en Géosciences.

#### Données d'acquisition

Dans la majorité des processus d'acquisition indirecte (scanner, résonance magnétique, sismique réflexion, etc.), le signal obtenu est stocké sous forme discrète dans des tableaux. Les voisinages entre points de données et leur position géométrique sont ainsi connus en fonction de leur indice dans le tableau. On parle dans ce cas de *grilles cartésiennes*, ou *régulières*.

#### Méthodes Geostatistiques

Les méthodes géostatistiques [Deutsch *et al.*, 1997 ; Goovaerts, 1997] constituent un cadre mathématique pour l'étude de variables spatiales. L'idée principale de la géostatistique est de considérer les corrélations entre points de l'espace pour effectuer des estimations ou des simulations.

Comme les corrélations spatiales sont généralement plus marquées dans l'espace de dépôt, ces supports peuvent être alignés sur la stratigraphie, permettant, dans une certaine mesure, une remise à plat virtuelle du modèle. On parle alors de *grille curvilinéaire*

En pratique, les méthodes géostatistiques sont surtout appliquées sur des grilles régulières ou curvilinéaires afin d'accélérer les recherches de voisinage. Conceptuellement cependant, la théorie géostatistique est indépendante du support volumique. À ce titre, la librairie générique GsTL [Rémy *et al.*, 2002], constitue un cadre élégant et efficace pour des applications à tous les types de grilles.

#### Simulations d'écoulements

Dans le domaine pétrolier, l'estimation de perméabilités par des méthodes géostatistiques est couramment utilisée pour appliquer des méthodes de simulation d'écoulement. Dans l'exploitation de réservoirs pétroliers, ces simulations permettent par exemple de gérer au mieux l'emplacement et le débit des puits d'injection et de pompage. Dans le domaine de l'environnement, elles peuvent être utilisées pour étudier la migration de polluants dans le sol.

Ces méthodes font intervenir essentiellement [Erterin *et al.*, 2001] la *loi de Darcy*, et le principe de *conservation de la masse*. La Loi de Darcy permet de mettre en relation la vitesse laminaire de filtration  $\mathbf{v}_p$  d'une phase incompressible

$p$  de viscosité  $\mu_p$  et de perméabilité relative  $k_p$ , à sa différence de potentiel  $\nabla\Phi$ , dans un milieu poreux de perméabilité  $\bar{\mathbf{k}}$  :

$$\mathbf{v}_p = \bar{\mathbf{k}} \cdot \frac{k_p}{\mu_p} \nabla\Phi \quad (3.1)$$

La conservation de la masse, spécifique quant à elle que la différence entre la quantité de fluide entrant et sortant d'un volume de roche donné correspond à une accumulation. En pratique, pour un élément de volume  $V$  et d'aire extérieure  $A$  orientée en tout point par une normale  $\mathbf{n}$ , la conservation de la masse d'un composant  $c$  peut s'écrire en fonction de sa concentration  $\omega_c$ , de la porosité  $\phi$  et du taux de saturation  $S_p$  de chaque phase  $p$  [Verma, 1996] :

$$\underbrace{\oint_A \sum_p \omega_c \cdot \mathbf{v}_p \cdot \mathbf{n} \, dA}_{\text{bilan du flux à travers } A} = \underbrace{\frac{\partial}{\partial t} \int_V \sum_p \omega_c \cdot S_p \cdot \phi \, dV}_{\text{accumulation dans } V} \quad (3.2)$$

Dans de nombreuses méthodes numériques [Erterin *et al.*, 2001], ces équations sont appliquées à des ensembles de cellules hexaédriques dont l'empilement définit des grilles régulières. Dans ce cas, de brusques variations spatiales de la nature des roches ou des conditions de pression nécessitent des grilles très denses, qui sont prohibitives en taille et en temps de calcul. Pour cela, des approches utilisant des *Grilles Flexibles* ont été définies [Aziz, 1993 ; Verma, 1996 ; Sukumar *et al.*, 2001]. Constituées de cellules de forme et de taille variables, ces grilles peuvent se conformer aux géométries souvent complexes des objets géologiques. Certaines hétérogénéités du sous-sol (liées à la stratigraphie, aux réseaux de failles, à la présence de chenaux . . .), sont ainsi prises en compte directement par la géométrie des grilles, ce qui permet de réduire considérablement le nombre de cellules et le temps de calcul par rapport à une discrétisation régulière, tout en améliorant la précision des simulations.

Plus généralement, toutes les méthodes numériques de résolution d'équations aux dérivées partielles (éléments finis, différences finies, . . .) utilisent des discrétisations du domaine en grilles. La validité des calculs numériques pose en général des contraintes souvent contradictoires sur la forme et la taille des cellules polyédriques utilisées [Verma, 1996 ; Sukumar *et al.*, 2001]. La prise en compte de ces contraintes pour différentes applications donne ainsi lieu à de nombreux algorithmes de génération de maillage [Conraud, 1997 ; Owen, 1998 ; Lepage, 2003].

### 3.1.2 Taxonomie des modèles cellulaires

Au cours de la présentation de quelques applications des représentations cellulaires volumiques, plusieurs types ont été définis. Pour les problèmes de représentation et de visualisation qui nous intéressent, la distinction entre ces types

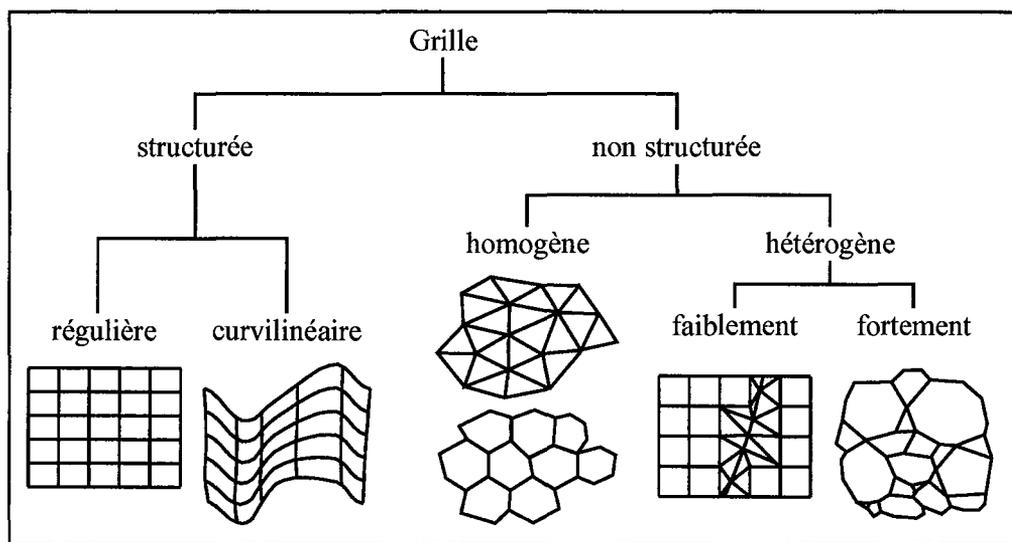


FIG. 3.1 – Classification des représentations cellulaires.

de grilles est également importante. Dans la suite de ce chapitre, la classification résumée par la figure 3.1 sera donc utilisée.

Les *grilles structurées* constituent des réseaux de noeuds, dans lesquels une maille élémentaire est répétée périodiquement. Les sommets de ces grilles sont ainsi accessibles par le schéma d'indexage de grille. La recherche de voisinage est donc rapide, mais le respect de géométries complexes induit des déformations de cellules qui peuvent nuire à la validité des calculs.

Lorsque les connections entre noeuds se présentent pas de régularité, on parle de *grilles non structurées*. L'*homogénéité* de ces maillages est relative aux types de cellules polyédriques présents. Une grille tétraédrique est ainsi dite homogène, alors qu'une grille comportant tétraèdres, prismes et hexaèdres est qualifiée d'hétérogène. Dans ce travail, nous distinguons également les grilles *faiblement hétérogènes*, dans lesquelles les types de cellules sont prédéfinis, des grilles *fortement hétérogènes*, constituées de polyèdres quelconques.

Dans la suite de ce chapitre, nous nous intéressons aux problèmes liés de la représentation et de la visualisation de ces différentes familles de grilles.

## 3.2 Les méthodes de visualisation volumique

La visualisation volumique [Kaufman, 1996] consiste à générer des images à partir de données tridimensionnelles. Les enjeux de cette visualisation sont principalement de deux ordres :

- dès lors que l'on veut visualiser l'intérieur d'une grille, il est nécessaire

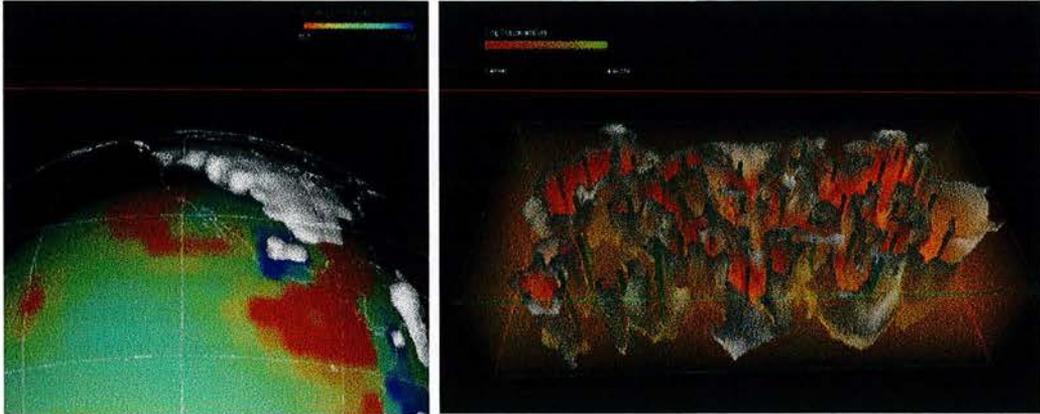


FIG. 3.2 – Deux exemples de visualisation volumique. À gauche, des hétérogénéités mantelliques sont imagées à l'aide d'extraction de surfaces. À droite, l'usage conjugué de rendu volumique et de surfaces d'isovaleur met en valeur les régions de forte perméabilité dans un réservoir pétrolier. (Avec l'aimable autorisation du *Lawrence Berkeley National Laboratory*, <http://www-vis.lbl.gov>).

d'extraire l'information de sorte que l'image générée soit *non ambiguë* et *significative* ;

- le volume de données à traiter est habituellement important, ce qui rend les accès et la génération d'images beaucoup plus lentes que pour des objets de dimension inférieure.

Au-delà de la transmission d'information à des non-spécialistes, la visualisation volumique présente un grand intérêt dans l'interprétation et l'analyse de données volumiques (FIG. 3.2).

### 3.2.1 Les techniques de visualisation

La visualisation de données volumiques scalaires (température, amplitude, porosité, pression, etc.) fait principalement appel à deux catégories de techniques. La première consiste à *extraire* des surfaces, lignes ou points du volume pour les représenter en utilisant les techniques classiques de rendu de points, lignes et triangles [Foley *et al.*, 1997]. Dans la seconde, l'ensemble des données volumiques est représenté en même temps sur une image 2D (*rendu volumique*).

#### Méthodes d'extraction

La plus courante des méthodes d'extraction, consiste à visualiser des lignes de champ ou des surfaces équipotentielles (*surfaces d'isovaleur* ou *iso-surfaces*). Ces lignes et surfaces sont alors dessinées à l'écran à l'aide de modèles d'éclaircissement simplifiés [Foley *et al.*, 1997].

Ce type de méthode permet rapidement de mettre l'accent sur des parties significatives de données volumiques. Un volume de données peut ainsi être exploré par des plans de coupe (iso-surfaces de la forme  $ax + by + cz + d = 0$ ) ou par des iso-surfaces arbitraires.

Cependant, par leur nature même, les méthodes d'extraction ne produisent qu'une vue partielle du volume [Kaufman, 1996]. Pour cette raison, Martha *et al.* [1997] proposent, sur le principe des iso-surfaces, de dessiner l'extérieur de zones comprises entre deux valeurs seuils. Ce type de méthode peut permettre de mieux percevoir certaines parties du volume que des iso-surfaces, en créant directement des solides b-reps. Cette méthode demeure cependant limitée pour visualiser une grille dans son ensemble.

### Rendu Volumique

Née au milieu des années 1980 [Kajiya *et al.*, 1984 ; R. A. Drebin, 1988 ; Sabella, 1988], la technique du rendu volumique génère une image dans laquelle chaque élément de volume apporte sa contribution. Dans cette optique, le volume est considéré comme un milieu semi-opaque qui laisse se propager la lumière.

Un modèle de rendu volumique est défini par UPSON et KEELER [Upson *et al.*, 1988] pour calculer l'intensité lumineuse en tout point de l'image à générer. Dans ce modèle, l'intensité  $I_\varphi$  de la longueur d'onde  $\varphi$  est accumulée pour chaque pixel sur un rayon  $R$  traversant le volume :

$$I_\varphi = \int_R \left\{ \underbrace{\alpha(s) \cdot att(s) \cdot I_{R_\varphi}(s)}_{\text{éclairage standard}} + \underbrace{(1 - \alpha(s)) \cdot fc_\varphi}_{\text{couleur de fond}} \right\} ds \quad (3.3)$$

où  $\alpha(s)$  est la fonction d'opacité à l'abscisse  $s$  sur le rayon,  $att(s)$  une fonction d'atténuation (pouvant éventuellement être ignorée), et  $I_{R_\varphi}(s)$  l'intensité lumineuse en  $s$ .

D'autres modèles [Sabella, 1988 ; Krüger, 1990], permettent quant à eux d'émuler les phénomènes physiques de propagation de la lumière dans des milieux inhomogènes. Dans ces modèles, le volume est constitué par des particules émettant et diffractant de la lumière ; la densité de particules dans une région élémentaire de la grille est fonction de la valeur scalaire de cette région.

Dans tous les cas, les modèles d'éclairage ne sont pas tant conçus pour générer une image réaliste qu'une image facilement *interprétable*. Pour permettre à l'utilisateur de privilégier certaines valeurs de propriété, une *fonction de transfert* entre valeur scalaire et opacité est généralement utilisée. La définition d'une fonction de transfert appropriée aux données fait l'objet de nombreuses recherches [Kniss *et al.*, 2002 ; Gerhardt *et al.*, 2002].

Les intérêts de ce type de méthode sont nombreux dans le domaine des Géosciences. Dans un réservoir pétrolier, une bonne compréhension de la géométrie des écoulements (FIG. 3.2) peut ainsi contribuer largement à la prise de décision.

### 3.2.2 Méthodes de calcul

La conversion de la grille en l'image fait intervenir principalement deux types d'approche, suivant que les calculs sont ordonnés dans l'espace de l'image ou dans l'espace de la grille [Kaufman, 1996].

#### Méthodes ordonnées suivant l'image (*screen-order*)

Dans la première catégorie, un rayon rectiligne est émis vers la grille depuis l'oeil de l'observateur à travers chaque pixel de l'image (lancer de rayon, ou *ray casting*) [Kajiya *et al.*, 1984 ; Sabella, 1988 ; Garrity, 1990 ; Levoy, 1990 ; Wilhelms *et al.*, 1990]. Pour le rendu d'iso-surfaces [Tuy *et al.*, 1984], seule la première intersection du rayon avec l'iso-surface est utilisée pour déterminer la couleur du pixel. Dans le cas du rendu volumique, toutes les intersections du rayon avec les cellules de la grille doivent être calculées et intégrées.

Pour cela, la méthode du « balayage paresseux » (*Lazy Sweep*) [Silva *et al.*, 1997] ordonne les calculs suivant une ligne de balayage correspondant à un plan horizontal. Pour chaque plan, une liste des arêtes intersectées est utilisée pour intégrer la couleur des pixels sans considérer l'ensemble du volume. Cette opération requiert un nouveau balayage de gauche à droite dans le plan horizontal, qui utilise à la fois des informations combinatoires et géométriques. L'algorithme de tranchage incrémental développé dans cette thèse présente des points communs avec cette méthode. Nous verrons toutefois que l'exploitation plus poussée des informations combinatoires confère à notre algorithme une complexité plus faible <sup>1</sup>

Pour un même objet, les temps de calcul sont donc d'autant plus importants que l'image à générer est grande ; en revanche, il est relativement aisé de paralléliser ce type d'algorithme, en affectant seulement une partie de l'image à chaque processeur [Parker *et al.*, 1999 ; Kutluca *et al.*, 2000].

Du point de vue de la qualité et de la précision, les images générées présentent des imperfections. Certaines cellules, trop petites pour être détectées ne sont ainsi pas visibles sur l'image finale. L'écart de couleur entre pixels adjacents peut également donner lieu à des effets de « franges de moiré » (*aliasing*). Cela explique que les méthodes basées-écran soient aujourd'hui rarement utilisées pour l'extraction d'iso-surfaces. Cependant, lorsque la rapidité de la méthode n'est pas prioritaire, ces deux problèmes peuvent être réduits en augmentant la taille de l'image.

<sup>1</sup>La complexité  $O(f(n))$  d'un algorithme détermine sa rapidité d'exécution fonction de la taille  $n$  de ses données d'entrée. Dans ce travail, les complexités sont toutes données *en moyenne*.

### Méthodes ordonnées suivant l'objet (*object-order*)

La majorité des méthodes d'extraction d'iso-surfaces [Lorensen *et al.*, 1987 ; Bajaj *et al.*, 1996 ; Itoh *et al.*, 1996] relève de cette catégorie. Ces surfaces sont en effet plus aisément et fidèlement calculées à partir de la grille que par lancer de rayon, qui ne produit qu'un échantillonnage. Le principe de l'extraction en elle-même est de procéder par morceaux [Lorensen *et al.*, 1987] : pour chaque cellule de la grille, le ou les polygones d'isovaleur sont extraits puis dessinés à l'écran.

Dans le cas du rendu volumique, toutes les cellules de la grille sont projetées sur l'écran dans l'ordre de visibilité [Upson *et al.*, 1988 ; Shirley *et al.*, 1990]. Le calcul d'un ordre de visibilité correct constitue une des principales difficultés dans la mise en oeuvre de ces méthodes. Certaines grilles présentent en effet des cycles qui rendent impossible le calcul de cet ordre.

Williams [1992] décrit ainsi un algorithme pour ordonner des maillages constitués de cellules convexes. En considérant l'orientation des faces des cellules relativement au point de vue, cette méthode distingue ainsi les *faces entrantes* des *faces sortantes* pour construire un graphe d'adjacence orienté (voir aussi §3.5.3, p 119). L'ordre de visibilité est ensuite obtenu par un tri topologique du graphe. Cette méthode, comme ses dérivées [Silva *et al.*, 1998 ; Comba *et al.*, 1999], présente deux limites. D'une part, elle ne permet de traiter que des grilles sans cycle de visibilité. D'autre part, elle est relativement sensible aux cellules et aux faces dégénérées. Une approche plus robuste, mais limitée aux tétraédrisations de Delaunay, est proposée par Cignoni *et al.* [1998]. En effet, la longueur entre la tangente au cercle circonscrit d'un tétraèdre et le point de vue constitue une relation d'ordre pour le calcul de visibilité. Un algorithme de tri rapide peut ainsi être utilisé.

La méthode de projection de cellules a également fait l'objet de nombreuses recherches. Dans l'algorithme fondateur de SHIRLEY et TUCHMAN [Shirley *et al.*, 1990], les tétraèdres sont convertis en trois à quatre triangles en fonction du point de vue, qui sont ensuite dessinés par la carte graphique. Les valeurs de couleur et d'opacité entre les sommets de ces triangles sont calculées par interpolation linéaire, ce qui donne lieu à des artefacts si la fonction de transfert n'est pas linéaire [Stein *et al.*, 1994]. Plusieurs auteurs [Stein *et al.*, 1994 ; Röttger *et al.*, 2000] utilisent ainsi des cartes de textures pour obtenir des images correctes.

Les images générées sont ainsi plus représentatives que celles obtenues par lancer de rayon, car chaque cellule apporte sa contribution à l'image finale. De plus, la projection permet d'éviter les effets de moiré.

Au niveau du temps de calcul, le rendu par projection, est essentiellement dépendant de la taille de la grille, et peu sensible à la taille de l'image. Cette approche est ainsi relativement plus adaptée que les méthodes basées-écran pour générer de grandes images précises. En revanche, comme la majorité du traite-

ment s'effectue dans l'espace de la grille, il est plus difficile de paralléliser ce type de méthode [Ma *et al.*, 1997].

Dans ce travail, une variante des méthodes de projection est utilisée (§3.3, p. 96). Dans son article fondateur, Yagel *et al.* [1996] propose en effet de découper la grille par plusieurs plans parallèles à l'écran. Les tranches ainsi obtenues sont rendues d'arrière en avant efficacement grâce à la carte graphique. Cette approche présente plusieurs avantages :

- elle laisse latitude à l'utilisateur pour trouver le compromis entre la justesse de l'image (déterminée par le nombre de tranches) et la vitesse de rendu ;
- elle exploite les importantes capacités de calcul des cartes graphiques modernes ;
- elle permet de traiter dans le même algorithme générique l'extraction d'iso-surfaces et le rendu volumique (§3.3, p. 96)

### 3.2.3 Méthodes d'optimisation

Fondamentalement, les méthodes de visualisation volumique n'ont guère évolué depuis 1992. Une large majorité de travaux de ce domaine concerne en revanche l'optimisation des méthodes existantes. L'accroissement des capacités de calcul et de mémoire entraîne en effet une augmentation notable de la taille des grilles à visualiser. Parallèlement, si les codes de calcul peuvent prendre plusieurs heures voire plusieurs jours, la rapidité des méthodes de visualisation reste primordiale pour la bonne compréhension d'objets tridimensionnels. L'accélération des méthodes de visualisation volumique repose principalement sur la parallélisation des algorithmes, évoquée dans le paragraphe précédent, sur l'utilisation des processeurs graphiques, et sur l'optimisation des parcours et des traitements effectués sur les grilles.

#### Utilisation des cartes graphiques

Les capacités des processeurs graphiques standards sont d'ores et déjà utilisées dans de nombreuses approches de rendu volumique. Les méthodes de projection [Shirley *et al.*, 1990 ; Wittenbrink, 1999] et le tranchage incrémental [Yagel *et al.*, 1996 ; Lévy *et al.*, 2001] exploitent ainsi principalement la rasterisation de polygones. Westermann [2001] tire avantage des textures en deux dimensions pour rééchantillonner des grilles tétraédriques en grilles régulières. Des piles de textures 2D peuvent également servir directement à générer des images [Rezk Salama *et al.*, 2000]. Lorsqu'elles sont disponibles sur les processeurs graphiques, les textures volumiques permettent d'améliorer la précision de la projection de cellules sans perte de performance [Röttger *et al.*, 2000]. Plusieurs types d'opérations utilisent également des combinaisons d'image pour réaliser des calculs d'iso-surfaces ou du rendu volumique [Westermann *et al.*, 1997, 1998].

Enfin, des processeurs graphiques ont récemment été développés spécifiquement pour le rendu volumique de grilles régulières [Pfister *et al.*, 1999]. Le gain en performance de ce type de matériel semble plaider à moyen terme pour des stratégies de rééchantillonnage des grilles non structurées [Westermann *et al.*, 1998]. À ce titre, le tranchage incrémental semble tout à fait adapté.

### Optimisation des accès et du traitement des grilles

De manière générale, l'élimination des traitements inutiles est sans doute une des meilleures stratégies d'optimisation. Pour les méthodes de rendu volumique, cela implique souvent d'initialiser des structures de données efficaces préalablement au rendu proprement dit. Pour cela, l'information peut être organisée suivant :

**la géométrie des objets** ; les arbres octaux accélèrent ainsi les recherches dans les grilles [Wilhelms *et al.*, 1992 ; Westermann *et al.*, 1999 ; Hormann *et al.*, 2002] ; les partitions binaires de l'espace [Comba *et al.*, 1999] sont également utilisées pour accélérer le calcul d'ordre de visibilité ;

**le champ de valeurs à visualiser** ; dans un espace de valeurs scalaires unidimensionnel, chaque cellule d'une grille peut être décrite simplement par son minimum et son maximum, ce qui permet d'optimiser l'extraction d'isosurfaces [Livnat *et al.*, 1996 ; Bajaj *et al.*, 1996] ; des méthodes fréquentielles sont également utilisées pour réduire la complexité du rendu volumique [Kaufman, 1996] ;

**les informations combinatoires** ; certaines techniques de projection utilisent ainsi les connexions entre cellules pour réduire de moitié les projections de faces ; plus généralement, toutes les méthodes mettant en jeu une propagation exploitent les connectivités du maillage [Itoh *et al.*, 1996 ; Bajaj *et al.*, 1996 ; Silva *et al.*, 1997 ; Lévy *et al.*, 2001]. Les méthodes multi-résolution permettent également d'accélérer les traitements [Cignoni *et al.*, 1997, 2000 ; Farias *et al.*, 2000], mais restent limitées aux grilles régulières et aux maillages tétraédriques.

## 3.3 Le tranchage incrémental : une méthode de visualisation progressive et polyvalente

### 3.3.1 Principe

Pour la visualisation volumique de maillages non structurés, nous avons choisi de développer une approche de tranchage incrémental, initialement introduite par Yagel *et al.* [1996] pour le rendu volumique de grilles tétraédriques (Figure 3.3). De manière similaire à [Giertsen, 1992 ; Wilhelms *et al.*, 1996 ; Westermann *et al.*,

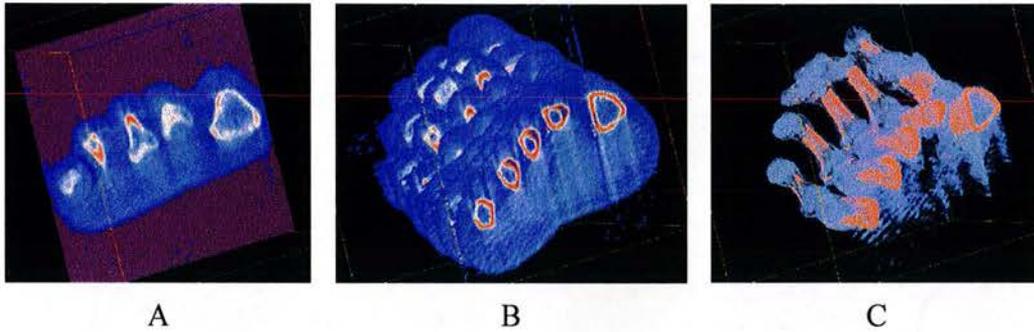


FIG. 3.3 – Principe du tranchage incrémental pour le rendu volumique. Plusieurs tranches sont calculées, et dessinées d'arrière en avant (A et B). En modifiant la fonction de transfert de transparence et en augmentant le nombre de tranches (C), les zones internes deviennent visibles. (Données médicales *Philips Research*).

1997 ; Silva *et al.*, 1997], cette méthode décompose un problème tridimensionnel en un ensemble de problèmes bidimensionnels. Pour le rendu volumique, cette approche consiste en effet à découper le volume en tranches, dessinées d'arrière en avant en utilisant la transparence.

En pratique, cette méthode peut être exprimée de manière plus générale : il s'agit de calculer un ensemble de surfaces d'isovaleurs entre deux valeurs scalaires  $\varphi_{min}$  et  $\varphi_{max}$ , séparées par un intervalle  $\Delta\varphi$ . Dans cette formulation, le rendu volumique est un cas particulier où la valeur scalaire est la profondeur  $z$  dans le repère des coordonnées de vue (*viewing coordinates*).

### Calcul d'iso-surfaces

Pour un algorithme naïf, ce calcul de  $N_\varphi$  surfaces d'isovaleur sur une grille contenant  $N_c$  cellules demande de chercher des intersections entre chaque cellule et chaque surface. Le temps d'exécution d'un tel algorithme est donc proportionnel à  $N_\varphi \cdot N_c$ , alors que, en moyenne, seulement  $N_\varphi \cdot N_c^{2/3}$  polygones d'intersection sont effectivement dessinés. Ainsi, Itoh *et al.* [1996] proposent d'accélérer le calcul en évitant de considérer les cellules non intersectées par l'iso-surface. Pour cela, deux ensembles de cellules sont pré-calculés (FIG. 3.4) :

- un arbre de recouvrement reliant l'ensemble des noeuds extrêmes, ayant une valeur inférieure ou supérieure à leurs voisins ;
- l'ensemble des noeuds du bord, triés en fonction de leurs valeurs scalaires.

Pour chaque valeur à calculer, la surface d'isovaleur est déterminée par propagation à partir de ces ensembles. Dans un principe similaire, la méthode des *graines de contours* (*contour seeds*) [Bajaj *et al.*, 1996] utilise un ensemble plus réduit de cellules-graines, pré-calculé en considérant l'intervalle de variation dans chaque cellule (FIG. 3.4-B). Ainsi, toute composante connexe d'isovaleur recoupe au moins une cellule-graine, qui permet d'initier la propagation.

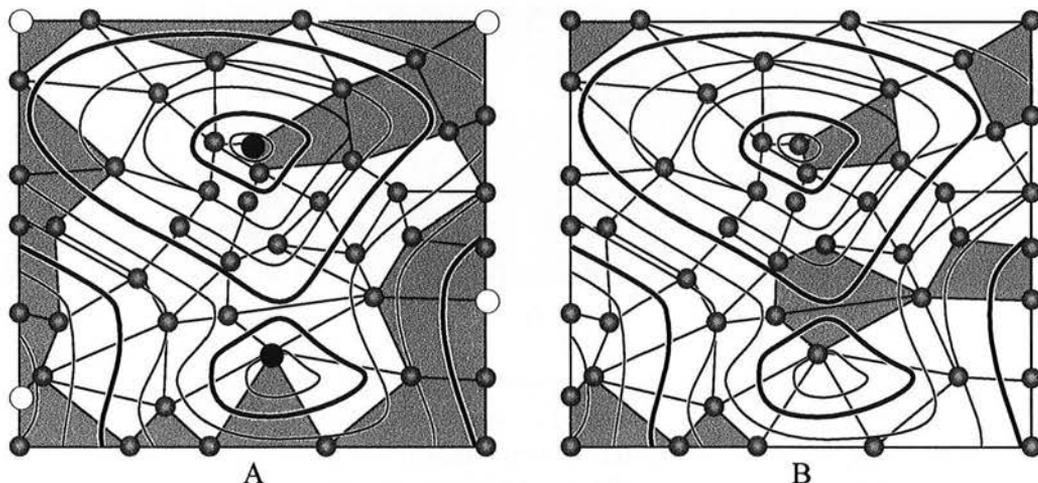


FIG. 3.4 – Deux méthodes de calcul d'isovaleur. Dans [Itoh *et al.*, 1996] (A), les isovaleurs sont calculées par propagation à partir d'un graphe reliant les sommets minima (ronds blancs) aux sommets maxima (ronds noirs), et des cellules du bord (en grisé). Dans la méthode (B) des *graines de contours* [Bajaj *et al.*, 1996], l'ensemble des cellules initiant la propagation, ou cellules-graines (en grisé) est réduit au minimum. On peut en effet constater que l'espace des propriétés est entièrement recouvert par les cellules-graines.

Dans ces deux approches, La propagation du contour se fait en utilisant le maillage de la grille [Howie *et al.*, 1994]. Cette étape fait intervenir soit une recherche en largeur d'abord (*breadth-first search*), soit une recherche en profondeur d'abord (*depth-first search*).

### Exploitation de la cohérence

Pour le calcul d'une famille de surfaces d'isovaleurs, les méthodes par propagation ne sont cependant pas optimales, car elles n'exploitent pas la cohérence [Foley *et al.*, 1997] entre deux isovaleurs successives. Pour cela, une structure d'*éléments actifs* (Algorithme 3.1) peut être utilisée plus efficacement, comme dans [Yagel *et al.*, 1996 ; Silva *et al.*, 1997].

Ainsi, l'algorithme fondateur pour effectuer un tranchage incrémental [Yagel *et al.*, 1996] utilise une liste d'arêtes actives. Entre deux iso-surfaces, celle-ci est mise à jour à l'aide d'une table pré-calculée de taille  $N_\varphi$  (*EdgeBuckets*). Pour une tranche  $\varphi$ , l'entrée correspondante de la table donne un ensemble d'arêtes; parmi elles, les arêtes jusque-là inactives sont ajoutées à la liste d'éléments actifs, alors que les arêtes actives en sont retirées. Cette structure de données présente deux principales limites. Premièrement, elle doit être recalculée lorsque l'utilisateur modifie le nombre de tranches. Deuxièmement, chaque arête est stockée deux fois dans la table, ce qui reproduit les informations combinatoires par ailleurs

**Algorithme 3.1** Dessin d'une grille par tranchage incrémental**Entrée :** $\mathcal{G}$  : Grille $\varphi_{min}, \varphi_{max}, \Delta\varphi$  : valeurs extrêmes et intervalle entre les iso-surfaces.**Début** $\varphi \leftarrow \varphi_{min}$  $elems\_actifs$  : file  $\leftarrow \emptyset$  $elems\_balayés$  : file  $\leftarrow \emptyset$  $minima\_triés$  : liste  $\leftarrow \mathbf{minima}(\mathcal{G})$  $minimum\_index$  : entier  $\leftarrow 0$ **Tant que**  $\varphi < \varphi_{max}$ **mise\_à\_jour\_eactifs**( $\varphi$ ,  $elems\_actifs$ ,  $elems\_balayés$ ,  $minima\_triés$ ,  $minimum\_index$ )

// Cf. Algorithme 3.2

**Pour chaque** élément  $e$  de  $elems\_actifs$ Dessiner l'intersection de l'isovaleur  $\varphi$  avec l'élément  $e$ 

// Cf. Algorithmes 3.6 et 3.9

**Fin Pour** $\varphi \leftarrow \varphi + \Delta\varphi$ **échanger**  $elems\_actifs$  et  $elems\_balayés$ **Fin Tant que****Fin**

disponibles dans le maillage de la grille.

Par rapport à cette structure, il est donc plus efficace d'utiliser directement les connexions entre cellules [Howie *et al.*, 1994 ; Itoh *et al.*, 1996 ; Bajaj *et al.*, 1996]. C'est le cas de l'algorithme de tracé de rayon par « *balayage paresseux* » (*lazy sweep*) [Silva *et al.*, 1997], qui utilise les cellules connectées à chaque sommet pour mettre à jour sa liste d'arêtes actives entre deux plans de balayage. Comme une propagation locale est susceptible d'ignorer les concavités du maillage (Figure 3.5-A), cette méthode utilise également les arêtes du bord de la grille dans la propagation. De façon similaire, l'algorithme de balayage, *ZSWEEP* [Farias *et al.*, 2000], utilise une liste de cellules actives, qui est mise à jour en parcourant les cellules autour des sommets, ainsi que les cellules du bord.

Afin de réduire le temps de mise à jour des éléments actifs d'une tranche à l'autre, une alternative à la vérification de tous les éléments du bord consiste à n'utiliser que des extrema locaux [Itoh *et al.*, 1996 ; Koyamada *et al.*, 1995 ; Bajaj *et al.*, 1996].

Dans notre approche, un élément est dit extrémal s'il ne peut pas être atteint par propagation locale. Une définition plus précise dépend donc du mode de propagation ; elle sera donnée plus loin en fonction du type de grille considérée.

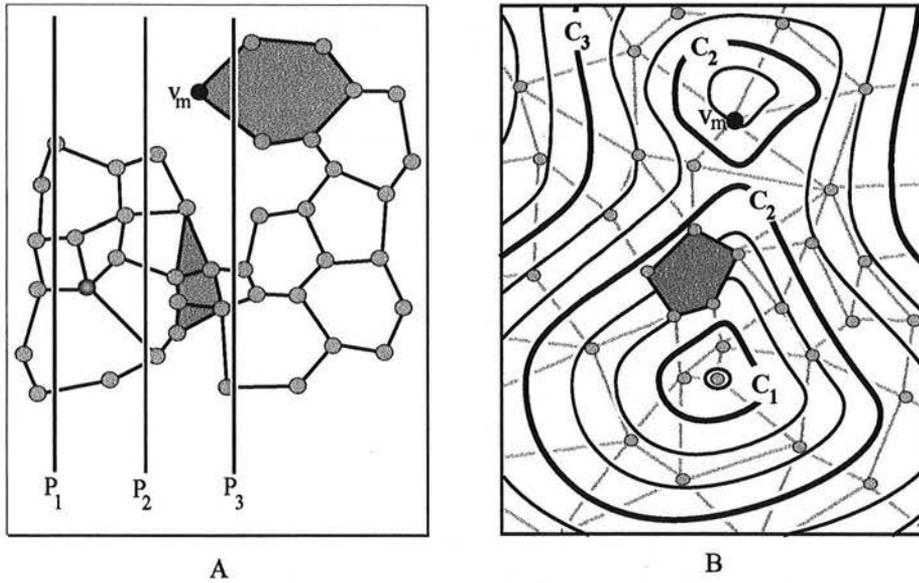


FIG. 3.5 – L’algorithme de tranchage incrémental calcule un ensemble d’isovaleurs géométriques (A) ou sur une propriété scalaire (B). La liste d’éléments actifs est mise à jour entre deux tranches successives en utilisant les connexions entre les cellules. Les cellules qui ne sont pas intersectées (en gris foncé), sont ainsi utilisées uniquement pour la propagation. D’une tranche à l’autre, les minima locaux balayés ( $v_m$ ) sont pris en compte.

Bien entendu, l’ensemble de ces éléments doit être recalculé pour tout changement du champ de valeurs scalaires considéré. En pratique, le nombre d’éléments extrêmes  $n_e$  est très inférieur au nombre total de cellules du bord, ce qui permet un traitement plus rapide que dans [Silva *et al.*, 1997 ; Farias *et al.*, 2000]. Une fois déterminé, l’ensemble des éléments extrémaux est trié par valeurs croissantes (pour les minima locaux) ou décroissantes (pour les maxima locaux), afin de réduire le nombre d’éléments à considérer entre deux tranches. Cette opération peut se faire avec un tri par sélection en un temps proportionnel à  $n_e \cdot \log N_e$ . Dans le cas du rendu volumique, un tri est théoriquement nécessaire pour chaque nouveau point de vue. Plus simplement, il est possible de trier ces éléments suivant les trois directions d’espace ( $x, y, z$ ) pour les deux directions de propagation, soit six ensembles ( $x+, x-, y+, y-, z+, z-$ ). Au moment du dessin, l’ensemble à consulter est alors choisi en fonction de l’axe le plus proche de la normale au plan de l’écran.

### 3.3.2 Mise à jour des éléments actifs

L’algorithme générique de propagation (Algorithme 3.2 et Figure 3.5) découle des considérations précédentes. Dans la mise en oeuvre de la mise à jour, plu-

sieurs approches ont été envisagées. Dans [Lévy *et al.*, 2001], une liste doublement chaînée est utilisée pour stocker les éléments actifs. Lorsqu'un élément reste actif entre la tranche  $\varphi$  et la tranche  $\varphi + \Delta\varphi$ , il est ajouté en fin de liste. Dans [Caumon *et al.*, 2003a], cette liste unique est remplacée par deux files, *elems\_actifs* et *elems\_balayés*. Cela simplifie d'une part les structures en évitant le retrait d'éléments situés en milieu de liste. D'autre part, cela permet de distinguer facilement les éléments intersectés de ceux qui ne sont utilisés que pour la propagation (Figure 3.5). Pour la tranche de valeur  $\varphi$ , la file *elems\_actifs* contient tous les éléments intersectés. Lors de la propagation à la tranche de valeur  $\varphi + \Delta\varphi$ , *elems\_actifs* et *elems\_balayés* sont échangées ; les minima locaux compris entre  $\varphi$  et  $\varphi + \Delta\varphi$  sont ensuite ajoutés à *elems\_balayés* ; enfin, suivant que leur valeur maximale est supérieure ou inférieure à  $\varphi + \Delta\varphi$ , les éléments de cette file sont respectivement ajoutés à *elems\_actifs* ou désactivés. La désactivation d'un élément consiste à chercher parmi ses voisins ceux qui n'ont pas encore été traités. Comme cette opération dépend de la structure de données utilisée, l'algorithme de désactivation sera donné *in extenso* dans les paragraphes 3.4 et 3.5.

### 3.3.3 Calcul des polygones d'intersection

Une fois mise à jour la liste des éléments actifs, l'algorithme de tranchage incrémental (Algorithme 3.1) calcule et dessine l'iso-surface  $\varphi$ . Cela nécessite de trouver pour chaque cellule polyédrique  $c$  le polygone  $p_\varphi$  d'isovaleur  $\varphi$ .

#### Principe

En première approximation, le calcul de  $p_\varphi$  suppose une variation monotone du champ scalaire sur la cellule polyédrique  $c$ . D'autre part, comme dans la plupart des algorithmes de calcul d'iso-surfaces [Lorensen *et al.*, 1987 ; van Gelder *et al.*, 1994 ; Itoh *et al.*, 1996 ; Bajaj *et al.*, 1996], la valeur scalaire est interpolée linéairement entre deux sommets connectés par une arête. Il est ainsi aisé de calculer les coordonnées des sommets du polygone  $p_\varphi$ .

Pour déterminer l'ordre de ces sommets, plusieurs approches sont présentées dans la littérature. Dans [Yagel *et al.*, 1996], un calcul purement géométrique d'enveloppe convexe est effectué. L'algorithme de balayage paresseux utilise quant à lui à la fois la géométrie et les informations combinatoires : les intersections sont d'abord triées par abscisse croissante ; les côtés des polygones d'intersection sont ensuite ordonnés autour de chaque point.

À ces deux méthodes s'appuyant sur la géométrie, un algorithme purement combinatoire plus efficace a été préféré. De manière similaire à [Bloomberg, 1988], il est en effet possible de trouver l'ordre des sommets d'intersection en tournant autour des faces du polyèdre d'une intersection à l'autre (Figure 3.6-A). Contrairement à la méthode des *Marching Cubes* [Lorensen *et al.*, 1987],

---

**Algorithme 3.2** Mise à jour de la file d'éléments actifs.

---

**mise\_à\_jour\_eactifs****Entrée :** $\varphi$  : isovaleur $elems\_balayés$  : file des éléments intersectés par l'isovaleur précédente $minima\_triés$  : liste des minima locaux triés par ordre croissant $minimum\_index$  : index du dernier minimum rencontré.**Sortie :** $elems\_actifs$  : file des éléments intersectés par l'isovaleur  $\varphi$ **Début****vider**  $elems\_actifs$ 

// Étape 1 : insérer les minima balayés

**Tant que**  $\varphi_{min}(sorted\_minima[minimum\_index]) < \varphi$ **et**  $minimum\_index < \text{nombre de minima}$  $elem$  : Élément  $\leftarrow minima\_triés[minimum\_index]$ **ajouter**  $elem$  à  $elems\_balayés$ **Fin Tant que**

// Étape 2 : propagation utilisant la connectivité

**Pour chaque** Élément  $elem$  **de**  $elems\_balayés$ **Si**  $\varphi_{max}(elem) > \varphi$ **activer\_element**(  $elem$  )**ajouter**  $elem$  à  $elems\_actifs$ **Sinon**// insérer les successeurs de  $elem$  à la fin de  $elems\_balayés$ **désactiver**(  $elem, elems\_balayés$  )

// Cf. Algorithmes 3.5 et 3.8

**Fin Si****Fin Pour****Fin**

---

cette méthode peut être appliquée à tout type de cellule polyédrique, car elle ne nécessite pas forcément l'usage d'une table pré-calculée.

La mise en oeuvre de cet algorithme dépend de la structure de donnée combinatoire utilisée, et donc du type de grille à visualiser. Une formulation générale pour des grilles fortement hétérogènes sera donc donnée dans le paragraphe 3.4, et une version indexée pour les autres types de grilles sera donc présentée dans le paragraphe 3.5.

Signalons également que le même algorithme pourrait être adapté pour calculer efficacement des coupes dans un modèle b-rep (Chapitre 2).

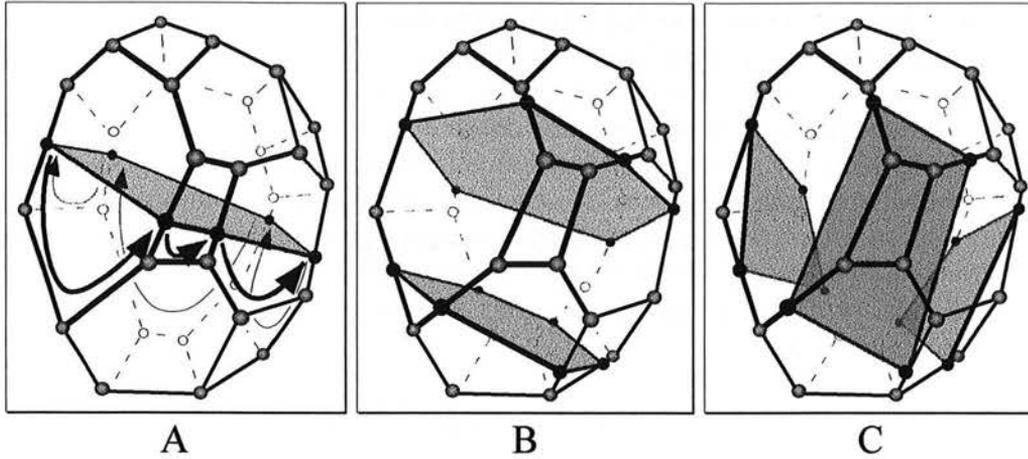


FIG. 3.6 – En partant d’une arête intersectée sur une cellule, l’ordre des sommets du polygone d’intersection est donné en tournant autour des face du polyèdre (A). Naturellement, cette méthode ne permet pas de résoudre les configurations ambiguës (B et C). De telles configurations peuvent se produire dans des cellules concaves, ou dans des cellules convexes lorsque les variations de propriétés ne sont pas monotones.

### Gestion des ambiguïtés

Le problème des configurations ambiguës dans le calcul d’iso-surfaces est bien connu pour les grilles régulières [van Gelder *et al.*, 1994]. Il n’a en revanche pas fait l’objet d’études poussées pour des grilles non structurées, sans doute car les grilles tétraédriques ne sont jamais ambiguës (en effet, l’interpolation linéaire d’un champ scalaire sur un tétraèdre est toujours cohérente mathématiquement). Dans le cas de grilles hétérogènes, le problème des ambiguïtés reste donc ouvert. Cette section présente quelques pistes de recherche pour une résolution de ces configurations.

En pratique, une configuration est dite ambiguë s’il existe plusieurs manières de joindre les points d’isovaleur  $\varphi$  sur les arêtes d’une cellule polyédrique. Cela se produit lorsque l’intersection de l’iso-surface avec la cellule est constituée de plusieurs composantes connexes (Figures 3.6-B et 3.6-C), ou que le polygone d’intersection contient des trous. De tels cas de figure peuvent apparaître principalement :

- si la fréquence de variation du champ scalaire est trop élevée pour satisfaire à l’approximation monotone et linéaire vue au paragraphe précédent (Figures 3.6-B et 3.6-C) ;
- si la cellule polyédrique présente des concavités, ce qui est rarement le cas pour les grilles de simulation numérique d’écoulement.

Par elle-même, la méthode proposée permet de traiter tous les cas où aucune face n'est intersectée plus de deux fois. Lorsque ce n'est pas le cas, différentes stratégies peuvent être utilisées pour choisir comment relier les points.

L'approche la plus efficace est sans doute de décomposer la cellule en tétraèdres et de raisonner sur chacun d'eux individuellement. Ce type de méthode, s'il permet de fournir une solution, n'est cependant pas idéal, car il n'impose pas que la polygonisation respectera le gradient du champ scalaire. Ainsi, une estimation du gradient au niveau des sommets pourrait être utilisée [van Gelder *et al.*, 1994].

À l'intérieur d'une cellule tétraédrique, la propriété  $\varphi$  est habituellement définie linéairement par :

$$\varphi(x, y, z) = a.x + b.y + c.z + d \quad (3.4)$$

Dans ce modèle, le gradient  $\nabla\varphi = [a, b, c]^t$  est donc considéré comme constant dans chaque tétraèdre, et peut être calculé à partir de la géométrie  $(x_i, y_i, z_i)_{i=[0..3]}$  et des valeurs  $(\varphi_i)_{i=0,1,2,3}$  aux sommets :

$$\underbrace{\begin{bmatrix} x_0 & y_0 & z_0 & 1 \\ x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}}_X = \underbrace{\begin{bmatrix} \varphi_0 \\ \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}}_B \quad (3.5)$$

Pour un sommet de la grille, la valeur du gradient est estimée par la moyenne des valeurs dans les tétraèdres voisins. Pour une cellule polyédrique quelconque, le système  $A \cdot X = B$  est surdéterminé. Dans ce cas, il est proposé de résoudre le système au sens des moindres carrés [Golub *et al.*, 1996] par :  $A^t A \cdot X = A^t B$ . En pratique, ce type d'estimation donne des résultats corrects si la variation du gradient est effectivement linéaire ou sub-linéaire dans la cellule. Elle pourra donc donner de bons résultats pour des ambiguïtés dues à des cellules concaves, mais ne sera pas utilisable pour des variations non monotones du champ scalaire. Dans ce cas, d'autres méthodes d'estimation doivent être étudiées, comme dans [Möller *et al.*, 1997 ; Neumann *et al.*, 2000].

### 3.3.4 Rendu à l'écran

Lorsqu'une iso-surface a été calculée, elle peut être dessinée directement à l'écran à l'aide d'une librairie graphique comme OpenGL [Neider *et al.*, 1993]. Au cours de cette opération, l'*ombrage* peut améliorer la perception de la géométrie et des hétérogénéités de manière significative [Foley *et al.*, 1997, Chap. 16]. Les calculs d'ombrages peuvent être effectués directement par les cartes graphiques à partir des normales en chaque sommet de polygones. Pour une iso-surface, la normale est par définition colinéaire au gradient  $\nabla\varphi$  du champ scalaire  $\varphi$ , et peut donc être pré-calculée à partir de l'Équation 3.5.

Dans le cas du rendu volumique, le dessin des tranches d'arrière en avant permet d'utiliser les fonctions de transparence de la carte graphique. Pour chaque pixel de l'image, les valeurs de couleur et de transparence  $\{\mathbf{i}_s = (r_s, v_s, b_s), \alpha\}$  sont interpolées sur chaque polygone d'intersection  $p_\varphi$ ; la nouvelle intensité  $\mathbf{i}_d$  du pixel est alors combinée à son ancienne valeur par :  $\mathbf{i}_d = \mathbf{i}_d \cdot \alpha + \mathbf{i}_s \cdot (1 - \alpha)$ .

En pratique, une interpolation linéaire de la couleur et de la transparence ne permet pas de traiter correctement les fonctions de transfert non linéaires [Stein *et al.*, 1994]. Comme le remarquent Yagel *et al.* [1996], une interpolation de la propriété elle-même est préférable, mais n'est pas possible avec OpenGL. Pour cela, nous avons proposé dans [Lévy *et al.*, 2001] une interpolation de texture 1D par la carte graphique, qui s'effectue rapidement, et prend en compte fidèlement les fonctions de transfert non linéaires.

### Gestion des cas particuliers

En théorie, rien n'empêche une isovaleur de passer exactement par un sommet de la grille. En pratique, ce cas de figure peut ralentir considérablement l'algorithme, car il introduit des tests supplémentaires au coeur de l'algorithme. Pour éviter cela, un pré-traitement consiste à normaliser les coordonnées et les valeurs scalaires de la grille en coordonnées entières et paires. Les isovaleurs peuvent être quant à elles normalisées en coordonnées entières et impaires, ce qui évite les cas dégénérés.

### 3.3.5 Bilan de la méthode

Le tranchage incrémental permet de répondre aux deux principales opérations de la visualisation volumique. Il permet en effet de produire des séries d'iso-surfaces ou d'effectuer du rendu volumique de manière unifiée. L'algorithme présenté dans cette partie constitue également un cadre générique indépendant du type de grille.

Les applications de cette méthode aux différentes familles de modèles cellulaires vont à présent être décrites. Dans chacun des cas, après une description de la structure de données utilisée, nous allons :

1. Spécifier le type d'élément (sommet, arête, cellule ...) utilisé pour la propagation.
2. Indiquer explicitement comment les successeurs d'un élément balayé sont déterminés en utilisant l'information combinatoire (fonction **désactiver** dans l'Algorithme 3.2).
3. Caractériser précisément les minima locaux, qui ne peuvent pas être atteints par cette propagation.
4. Définir comment les intersections sont dessinées à partir d'un élément actif.

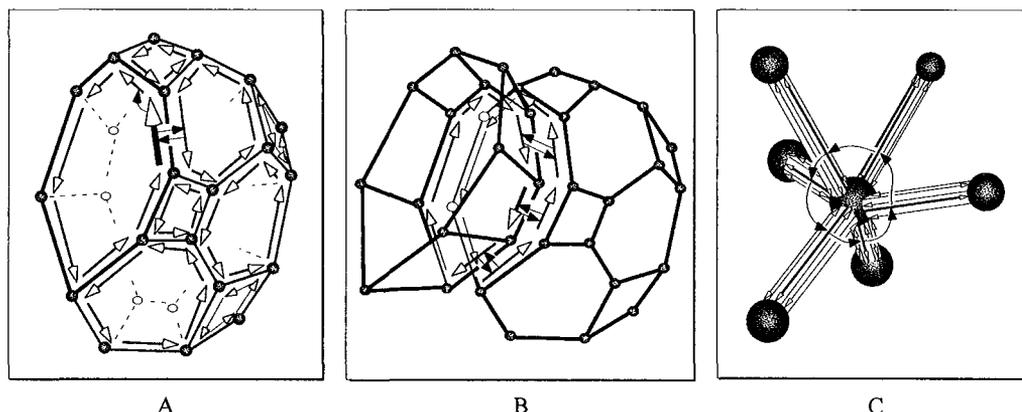


FIG. 3.7 – La notion de demi-arête orientée (flèches à tête blanche) est au coeur de la structure CIEL. Chaque demi-arête pointe vers son sommet d'arrivée, son successeur `succ_face` dans la face, et vers la demi-arête d'orientation opposée `autre_face` dans la même cellule (flèches à tête noire, Figure A). Un lien `autre_cell` permet également de passer d'une cellule à l'autre (B). Enfin, chaque sommet permet d'accéder directement à la liste chaînée des demi-arêtes qui en sont issues (C).

## 3.4 Tranchage incrémental de grilles fortement hétérogènes

### 3.4.1 Structure de données

Le traitement de grilles fortement hétérogènes nécessite une structure de données suffisamment flexible pour représenter des cellules polyédriques comportant un nombre arbitraire de faces, d'arêtes et de sommets. Dans ce travail, nous avons traité le cas de cellules homéomorphes à des sphères<sup>2</sup>. Les grilles représentées sont ainsi des *complexes cellulaires CW réguliers* (*regular CW-Complexes*, où CW signifie *closure-finite, weak-topology*) [Lundell *et al.*, 1969]. Les grilles peuvent avoir des bord concaves, et être constituées de plusieurs composantes connexes. La plupart des représentations existantes pour ces types de grilles s'appuient sur des théories de topologie combinatoire comme les cartes combinatoires [Edmonds, 1960], ou les cartes généralisées [Lienhardt, 1989] (pour une description plus détaillée, voir par exemple [Lévy, 1999 ; Conreux, 2001 ; Hubeli, 2002 ; Mallet, 2002]).

Dans ce travail, nous avons conçu et implanté la structure CIEL, listes circulaire d'arêtes incidentes (*Circular Incident Edges Lists*). Comme le montrent les FIGS. 3.7 et 3.8, cette représentation s'organise autour de la notion de demi-arête

<sup>2</sup> Concernant la géométrie, la validité des calculs d'intersection (§3.3.3), n'est assurée que pour des cellules convexes et à faces planes.

<pre> <b>struct</b> DemiArête {   DemiArête* succ_face;   DemiArête* succ_sommet;   DemiArête* autre_face;   DemiArête* autre_cell;   Sommet* sommet;   Cell* cell;   bool marqué; }; </pre>	<pre> <b>struct</b> Sommet {   float géométrie[3];   float propriété;   float gradient[3];   DemiArête* arête; };  <b>struct</b> Cell {   DemiArête* arête;   bool marqué; }; </pre>
--	--

FIG. 3.8 – Les primitives de la structure CIEL : cellules polyédriques, demi-arêtes, et sommets.

orientée, comme dans [Mäntylä, 1988, Chap. 10] ou [Kettner, 1998]. Dans CIEL, chaque demi-arête pointe vers son sommet d'arrivée (sommet), et vers son successeur dans la même face (`succ_face`, FIG. 3.7-A), ce qui permettra de tourner autour des faces d'une cellule pour ordonner les points d'intersection. Afin de pouvoir changer de face au sein d'une même cellule, une demi-arête référence également la demi-arête d'orientation opposée sur la face adjacente (`autre_face`, 3.7-A). Le passage d'une cellule à l'autre est permis par le lien `autre_cell` liant deux demi-arêtes d'orientation opposée (FIG. 3.7-B). En principe, ces trois types de relation suffisent à décrire complètement les relations combinatoires entre les demi-arêtes. Cependant, une quatrième relation, `succ_sommet` (FIG. 3.7-C) a été ajoutée afin d'optimiser les accès lors du tranchage incrémental. Cette relation permet en effet d'obtenir directement toutes les arêtes issues d'un sommet, alors que leur recherche dans une structure classique comme DCEL [Muller *et al.*, 1978 ; Preparata *et al.*, 1985] nécessite l'emploi d'une pile [Conreux, 2001].

D'un point de vue global, une grille représentée par CIEL stocke un tableau de sommets, un tableau de demi-arêtes et un tableau de cellules. Ce dernier n'est pas explicitement nécessaire, mais il permet de simplifier l'algorithme de dessin en marquant les cellules intersectées. Avant de préciser comment cette structure est utilisée dans le cadre du tranchage incrémental, nous allons maintenant décrire la construction d'une grille à partir d'un fichier ou d'une base de données.

### 3.4.2 Construction

La manipulation de grilles fortement hétérogènes nécessite des formats de fichiers spécifiques qui contiennent toute l'information combinatoire de la grille. Pour cela, il est courant d'énumérer les sommets de la grille, puis les cellules polyédriques. Chaque cellule est définie par ses faces polygonales, elles-mêmes

spécifiées par des listes ordonnées de leurs sommets <sup>3</sup>.

La construction d'une grille à partir de ce type de fichier construit tout d'abord le tableau  $S$  des noeuds de la grille, qui permet d'associer un index  $s_i$  à un sommet  $S[s_i]$ . Les cellules de la grille sont ensuite créées face par face. Pour chaque face, la méthode procède en deux temps. Premièrement, les demi-arêtes sont formées, et soudées aux demi-arêtes existantes de la cellule (Algorithme 3.3). Deuxièmement, la face est connectée le cas échéant aux cellules voisines (Algorithme 3.4).

Afin de limiter les recherches par la suite, un traitement est effectué après la construction proprement dite. En effet, une grille CIEL contient plusieurs demi-arêtes de même orientation par arête. Comme ces groupes de demi-arêtes sont accessibles à l'aide des liens `autre_face` et `autre_cell`, les listes d'arêtes incidentes (lien `succ_sommet`) peuvent être facilement simplifiées sans perte d'information.

### Complexité

Pour une grille contenant  $N_c$  cellules, l'algorithme de construction se fait en temps proportionnel à  $k \cdot N_c$ , où  $k$  est une constante dépendant du type de grille. Le facteur  $k$  peut être estimé par le produit des nombre moyens de faces par cellule  $\bar{f}_c$ , de sommets par face  $\bar{s}_f$ , et d'arêtes autour d'un sommet  $\bar{a}_s$ . En pratique,  $k$  est de l'ordre de  $6 \times 4 \times 6 = 144$  pour une grille hexaédrique, de  $4 \times 3 \times \sim 10 \simeq 120$  pour un maillage tétraédrique de Delaunay et de  $\sim 10 \times \sim 5 \times 4 \simeq 200$  pour un maillage dual de Voronoï. Dans le cas où les sommets sont donnés indépendamment, un pré-traitement doit être effectué pour fusionner les sommets co-localisés. Ce type d'opération fait intervenir des structures de recherche utilisant la géométrie, qui requièrent en général un temps proportionnel à  $N_c \cdot \log N_c$ .

### 3.4.3 Tranchage incrémental avec CIEL

Avec la représentation CIEL, nous disposons de toute l'information nécessaire au rendu graphique de grilles fortement hétérogènes. Comme la structure de donnée elle-même, l'algorithme de tranchage incrémental est centré autour de la notion de demi-arête.

---

<sup>3</sup>Dans ce type de description, l'information combinatoire est donnée de manière redondante, car les faces internes de la grille sont décrites deux fois. Les faces pourraient éventuellement être définies à part et identifiées par des index, aux dépens de temps de chargement et d'écriture plus lents.

**Algorithme 3.3** Construction d'une face de cellule polyédrique.

---

```

créer_face
  Entrée :
     $V$  : tableau de sommets de la grille
     $\{s_1, \dots, s_k\}$  : index des  $k$  sommets de la face à créer
  Sortie :
     $A_f = \{a_1, \dots, a_k\}$  : tableau de  $k$  demi-arêtes de la face
  Début
     $A_f[1] \leftarrow \text{créer\_demi\_arête}(s_1, s_2)$ 
    Pour  $i = 2$  à  $k$ 
       $A_f[i] \leftarrow \text{créer\_demi\_arête}(s_i, s_{i\oplus 1})$ 
       $A_f[i-1].\text{succ\_face} \leftarrow A_f[i]$ 
      // Mettre à jour la liste des demi-arêtes autour de  $V[s_i]$ 
      Si  $V[s_i].\text{arête} \neq \emptyset$ 

         $A_f[i].\text{succ\_sommets} \leftarrow V[s_i].\text{arête}$ 
      Sinon
         $A_f[i].\text{succ\_sommets} \leftarrow A_f[i]$ 
      Fin Si
       $V[s_i].\text{arête} \leftarrow A_f[i]$ 
      // Connecter  $A_f[i]$  aux faces adjacentes de la cellule en construction
       $a$  : demi-arête  $\leftarrow V[s_{i\oplus 1}].\text{arête}$ 
       $\text{trouvé}$  : booléen  $\leftarrow$  Faux
      Tant que  $a \neq \emptyset$  et Non Trouvé
        Si  $a.\text{sommets} = V[s_i]$ 
           $A_f[i].\text{autre\_face} \leftarrow a$ 
           $a.\text{autre\_face} \leftarrow A_f[i]$ 
           $\text{trouvé} \leftarrow$  Vrai
        Fin Si
         $a \leftarrow a.\text{succ\_sommets}$ 
      Fin Tant que
    Fin Pour
     $A_f[k].\text{succ\_face} \leftarrow A_f[1]$ 
  Fin

```

---

**Mise à jour des arêtes actives**

De façon similaire au balayage paresseux [Silva *et al.*, 1997], la mise à jour des files d'éléments actifs fait essentiellement intervenir la liste des demi-arêtes incidentes disponible aux sommets de la grille (Algorithme 3.5). À ce titre, la simplification évoquée au paragraphe 3.4.2 est importante pour assurer l'efficacité de la méthode, car elle permet de minimiser le nombre d'éléments de cette liste.

**Algorithme 3.4** Connexion de deux cellules partageant une face polygonale.

---

```

coudre_face
  Entrée :
     $V$  : tableau de sommets de la grille
     $\{s_1, \dots, s_k\}$  : sommets de la face
     $A_f = \{a_1, \dots, a_k\}$  : tableau de  $k$  demi-arêtes de la face
  Début
    // Trouver deux demi-arêtes  $a'_2$  et  $a'_1$  connectant respectivement  $s_3$  à
     $s_2$  et  $s_2$  à  $s_1$ .
     $a'_2$  : demi-arête  $\leftarrow V[s_3].\text{arête}$ 
     $\text{trouvé}$  : booléen  $\leftarrow$  Faux
    Tant que  $a'_2 \neq \emptyset$  et non  $\text{trouvé}$ 
       $a'_1$  : demi-arête  $\leftarrow a'_2.\text{succ\_face}$ 
      Si  $a'_2.\text{sommet} = V[s_2]$  et  $a'_1.\text{sommet} = V[s_1]$ 
        // Parcourir la face contenant  $a'_1$ , et connecter les arêtes avec le
        lien  $\text{autre\_cell}$ .
        Pour  $i = k \dots 1$ 
           $a'_1 \leftarrow a'_1.\text{succ\_face}$ 
           $a'_1.\text{autre\_face} \leftarrow A_f[i]$ 
           $A_f[i].\text{autre\_face} \leftarrow a'_1$ 
        Fin Pour
         $\text{trouvé} \leftarrow$  Vrai
      Fin Si
       $a'_2 \leftarrow a'_2.\text{succ\_sommet}$ 
    Fin Tant que
  Fin

```

---

**Arêtes minimales**

D'après ce schéma de propagation, les demi-arêtes qui doivent être étudiées entre deux tranches (Algorithme 3.2 p. 102, étape 1) sont celles qui partent de sommets ayant une valeur  $\varphi$  inférieure à celle de leurs voisins.

**Dessin des arêtes actives**

Dernière brique élémentaire de l'algorithme générique, le dessin d'une iso-surface à partir de la file des demi-arêtes actives est résumé par l'Algorithme 3.6. Pour chaque arête active, les liens `succ_face` et `autre_face` permettent de tourner autour de la cellule polyédrique (Figure 3.6 p. 3.6). Lorsque le polygone d'intersection est dessiné, la cellule est marquée afin d'éviter plusieurs passages. Cela implique que les cellules intersectées soient démarquées entre chaque tranche. Dans l'algorithme 3.6, les instructions `début_polygone`, `fin_polygone` et `sommet` correspondent à des fonctions de la librairie graphique [Neider *et al.*, 1993].

---

**Algorithme 3.5** La propagation entre deux tranches successives s'appuie sur les listes de demi-arêtes incidentes.

---

**désactiver**

**Entrée :**

$a_{in}$  : demi-arête à désactiver

$elems\_balayés$  : file des demi-arêtes dépassées par le plan de balayage

**Début**

$a_{succ}$  : demi-arête  $\leftarrow a_{in}$  .sommet .arête

$a_{cour}$  : demi arête  $\leftarrow a_{succ}$

**Faire**

**Si**  $a_{cour}$  .sommet . $\varphi > a_{in}$  .sommet . $\varphi$  **et non**  $a_{cour}$  .marqué

**ajouter**  $a_{cour}$  **à la fin** de  $elems\_balayés$

$a_{cour}$  .marqué  $\leftarrow$  **vrai**

**Fin Si**

$a_{cour} \leftarrow a_{cour}$  .succ\_sommet

**Tant que**  $a_{cour} \neq a_{succ}$

**Fin**

---

La fonction **intersection** retourne quant à elle les coordonnées de l'intersection entre une demi-arête et une isovaleur obtenues par interpolation linéaire. En pratique, elle retourne également la couleur du point par interpolation de texture, ainsi que la normale à l'isovaleur pour permettre l'ombrage.

## Optimisation

Dans l'Algorithme 3.6, il est aisé de voir que les intersections entre une arête active et l'iso-surface sont calculées pour chaque cellule incidente à une arête. Il pourrait donc sembler souhaitable de conserver l'intersection en mémoire afin d'éviter ces calculs redondants. En pratique, cela était effectué dans la première version de la méthode [Lévy *et al.*, 2001]. Cependant, comme le montre le Tableau 3.1, les performances sont légèrement moins bonnes que lorsque les intersections sont recalculées.

Surprenant au premier abord, ce résultat peut s'expliquer par le coût en mémoire du stockage des intersections dans des structures d'arêtes, ce qui augmente significativement la taille de la grille. Il semble en effet que les accès mémoire soient le facteur limitant de la méthode, plus que la quantité de calculs du processeur central <sup>4</sup>

---

<sup>4</sup>Cette observation a été effectuée sur plusieurs types de PC. Il est possible que, sur des architectures dotées de processeurs plus lents mais gérant mieux les accès mémoire (par exemple des SGI), le stockage des arêtes soit avantageux.

**Algorithme 3.6** Dessin des polygones d'isovaleur autour d'une arête active.

---

```

dessiner_cellules_autour_arête
  Entrée :
     $a_{in}$  : demi-arête de départ
     $\varphi$  : isovaleur
  Début
     $a_{cour} \leftarrow a_{in}$ 
    Faire // itération sur toutes les cellules autour de  $a_{in}$ 
      Si non  $a_{cour}$ .cell.marqué
         $a_{cour}$ .cell.marqué  $\leftarrow$  vrai
        début_polygone
           $a_{face} \leftarrow a_{cour}$ 
          Faire // itération sur les faces d'une cellule
            sommet( intersecter ( $a_{face}$ ,  $\varphi$ ) )
            Faire // itération sur les arêtes d'une face
               $a_{face} \leftarrow a_{face}$ .succ_face
              Tant que non  $a_{face}$ .marqué
                 $a_{face} \leftarrow a_{face}$ .autre_face
              Tant que  $a_{face} \neq a_{cour}$ 
            fin_polygone
          Fin Si
         $a_{cour} \leftarrow a_{cour}$ .autre_face.autre_cell
      Tant que  $a_{cour} \neq a_{in}$ 
    Fin

```

---

	sans ombrage	avec ombrage	taille en mémoire
Intersections recalculées	14.05	15.51	229,7 MB
Intersections stockées	15.68	16.51	302,1MB

TAB. 3.1 – Temps de dessin (en s.) pour 300 tranches dans une grille de 95 308 cellules polyédriques : le stockage des intersections entre les arêtes et les isovaleurs n'est pas plus rapide que des calculs redondants. (Calculs effectués sur un Pentium III 1,13 GHz, 512 MB)

### Complexité

Le temps d'exécution de l'algorithme de tranchage incrémental peut être séparé en temps de dessin, et temps mis pour mettre à jour les structures d'éléments actifs.

Pour une grille contenant  $N_a$  arêtes, avec une moyenne de  $\bar{c}_a$  cellules autour de chaque arête, la recherche des cellules intersectées à partir de la file d'arêtes actives prend un temps proportionnel à  $N_\varphi \cdot N_a^{2/3} \cdot \bar{c}_a$ , où  $N_\varphi$  est le nombre de tranches. Le rendu d'une cellule s'effectue quant à lui en temps constant, proportionnel

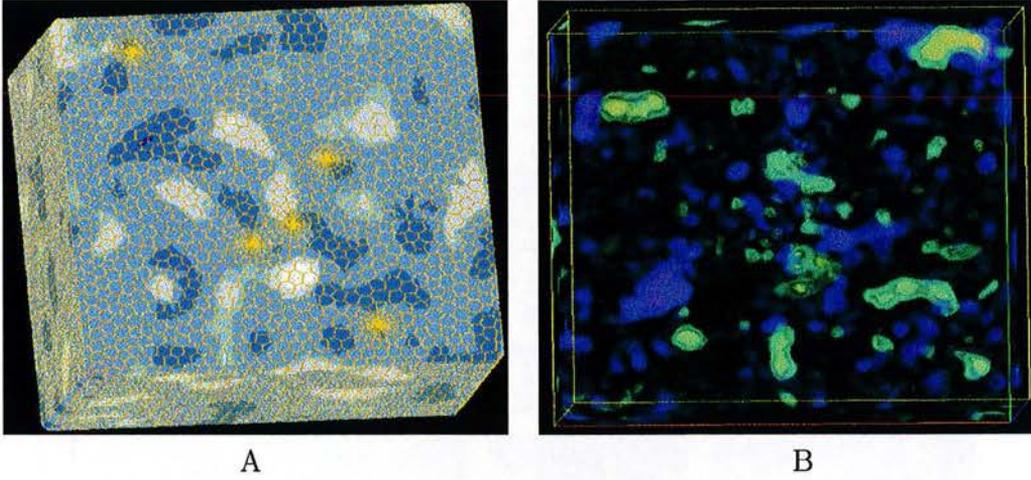


FIG. 3.9 – Images de la grille 1 générées avec la méthode. Le maillage de Voronoi comprend 95 308 cellules polyédriques ; l'image B de rendu volumique est obtenue avec 500 tranches.

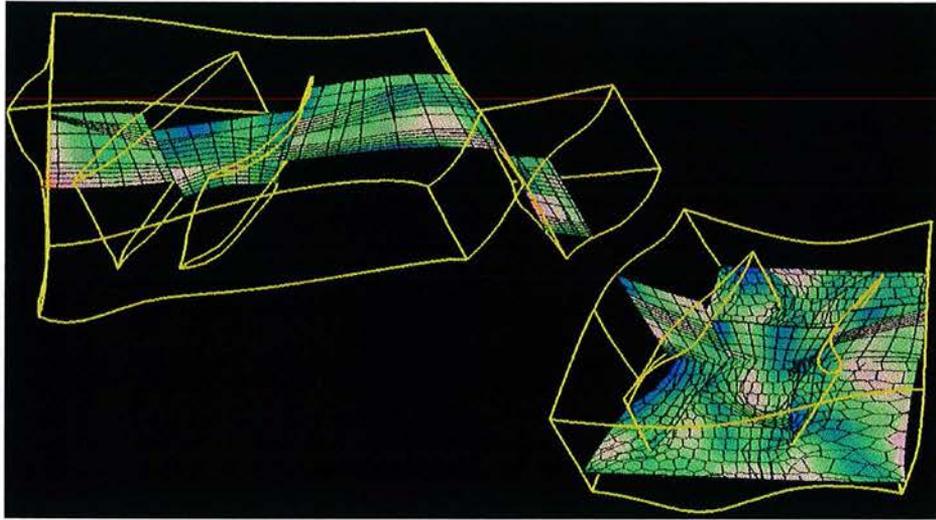
au nombre moyen de sommets par face  $\bar{s}_f$  multiplié par le nombre moyen de sommet par polygone d'isovaleur  $\bar{s}_i$ . Le dessin de  $N_\varphi$  tranches s'effectue donc en un temps variant *linéairement* en fonction du nombre d'intersections. Cet algorithme est donc plus efficace que le balayage paresseux [Silva *et al.*, 1997], qui trie les intersections.

Lors de la propagation d'une tranche à l'autre, chaque demi-arête active est traitée pour déterminer si elle doit être désactivée ou non. Soit  $k$  la probabilité qu'une demi-arête soit désactivée entre deux iso-surfaces successives, et  $\bar{a}_s$  le nombre moyen d'arêtes autour d'un sommet de la grille. En supposant que chaque arête est intersectée au moins par une iso-surface, la propagation est en  $O((k \cdot \bar{a}_s + (1 - k)) \cdot N_a^{2/3})$ .

Ainsi, dès lors que la grille est correctement échantillonnée, le temps d'exécution global dépend *linéairement* du nombre d'intersections dessinées. En pratique, sur une grille de densité de cellules homogène, la performance sera optimale pour un nombre de tranches est supérieur à  $N_a^{1/3}$ . Ce nombre assure que chaque cellule apporte sa contribution à l'image finale. En deçà, la grille est sous-échantillonnée, et la complexité de la propagation est non linéaire. Au delà, la complexité est linéaire, et l'image obtenue gagne en qualité, en diminuant les effets de moiré.

#### 3.4.4 Résultats

La méthode de rendu avec CIEL a été testée sur plusieurs grilles non structurées. Dans tous les cas, les champs de valeurs scalaires ont été créés *ab nihilo*. La grille 1 (FIG. 3.9) est un diagramme de Voronoï obtenu par dual d'une grille tétraédrique [Lepage, 2003]. La grille 3 (FIG. 3.10) a été obtenue par extrusion



A



B

FIG. 3.10 – Images de la grille 2 générées avec la méthode. Cette grille, créée par extrusion, est tranchée par deux plans (A), et visualisée avec 700 tranches (B).

d'une surface polygonale ([Conreux, 2001]).

Pour chaque grille, le nombre d'éléments ainsi que l'occupation en mémoire est donnée dans le TAB. 3.2. De manière prévisible, la plus grande partie de la mémoire est occupée par les structures de demi-arêtes; c'est en effet à ce niveau qu'est capturée la majeure partie de l'information combinatoire.

Les performances sont quant à elles mentionnées dans le TAB. 3.3. Dans ce tableau, le temps de chargement correspond à la lecture du fichier de données sur le disque et à la création de la structure CIEL. Le pré-traitement consiste à calculer les gradients en chaque cellule, et à chercher et trier les minima locaux.

	Grille 1	Grille 2
Sommets	604 232 (16,9 MB)	38 997 (1,7 MB)
Cellules	95 308 (0,8 MB)	18 340 (0,1 KB)
Demi-arêtes	7 220 766 (202,2 MB)	628 920 (17,7 MB)
Minima locaux	0.2 MB	5 KB
TOTAL	220.1 MB	19,5 MB

TAB. 3.2 – Taille et occupation mémoire pour deux grilles fortement non structurées.

		Grille 1	Grille 2
Chargement		74,6	6
Pré-traitement		5,3	0,1
Rendu volumique	100	7,1	0,4
	500	15,9	0,8
	1000	27,2	1,2

TAB. 3.3 – Performances obtenues pour le chargement et la visualisation de deux grilles fortement non structurées, en s.

Les délais par image en rendu volumique sont donnés respectivement pour 100, 400 et 1000 tranches. Ces temps d'attente sont légèrement variables selon l'angle de vue, nous avons figuré les meilleures performances. Une analyse plus poussée des résultats, en relation avec la complexité de l'algorithme sera donnée au Paragraphe 3.7, p. 127.

## 3.5 Visualisation de grilles faiblement hétérogènes

### 3.5.1 Structure de données

Dans la section précédente, nous avons vu que la visualisation de grilles fortement hétérogènes nécessitait des structures de données flexibles, occupant un espace important en mémoire. Pour représenter et visualiser des grilles faiblement hétérogènes plus efficacement, nous avons eu l'idée est d'utiliser leur degré de structuration plus élevé. Typiquement, ces grilles sont constituées de cellules de types prédéfinis, par exemple des tétraèdres, des hexaèdres ou des prismes. Ce type d'information n'est pas directement utilisable avec des représentations centrées autour des arêtes, car le nombre de cellules autour d'une arête reste variable. En revanche, il est possible d'exploiter cette information en utilisant un *graphe cellulaire* (*CGraph*). Chaque noeud du graphe est défini par un type et une géométrie de cellule ; les arcs du graphe permettent de connecter deux cellules voisines par leur face commune.

<pre> <b>struct</b> CGraphCell {   MetaCell* description;   Sommet* sommets[];   CGraphCell* voisins[];   Marques marques_faces;   bool fantôme;   bool active; }; </pre>	<pre> <b>struct</b> Sommet {   float géométrie[3];   float propriété;   float gradient[3]; }; </pre>
---	--

FIG. 3.11 – Les primitives combinatoires d’un graphe cellulaire. En s’appuyant sur l’information disponible dans les méta-cellules, les cellules comprennent uniquement un tableau de sommets, un tableau de voisins, et des marques booléennes (en pratique, `marques_faces`, `fantôme` et `active` peuvent être stockées dans le même ensemble de bits).

La structure de données proposée est donc composée de deux parties :

- une partie *dynamique* contenant les sommets et les cellules de la grille;
- une table *statique* de méta-cellules, contenant les différents types combinatoires de cellules dans la grille.

### Description dynamique d’un Graphe Cellulaire

Par définition, la partie dynamique d’un graphe cellulaire comprend toutes les informations spécifiques à une grille. Cela inclut le graphe d’adjacence entre cellules d’une part, et leur géométrie et le champ de valeurs qu’elles contiennent d’autre part. Ainsi, la structure de graphe cellulaire est constituée par un tableau de sommets et un tableau de cellules (FIG. 3.11). Comme dans CIEL, un sommet contient une position dans l’espace ainsi que la valeur de propriété en ce point. Une cellule connaît son type combinatoire, indexé dans la table des méta-cellules. Grâce à cette information, le stockage des sommets et des voisins d’une cellule peut se faire de manière très compacte dans deux tableaux.

Les faces polygonales n’ont pas de représentation explicite dans la structure proposée. Toutefois, elles sont accessibles à travers l’indigage des voisins des cellules. Pour le rendu, nous verrons cependant que les faces sont utilisées. En effet la face d’un polyèdre peut être *entrante*, *sortante* ou *neutre* par rapport à une direction de propagation donnée [Williams, 1992]. Un ensemble de bits est donc utilisé pour stocker cette orientation (`marques_faces`). Deux autres bits de marquage sont également disponibles pour activer une cellule et pour l’ignorer, par exemple si elle contient des faces dégénérées.

### Description statique des cellules

Chaque type de cellule de la grille est décrit par une *méta-cellule*, contenant l’information combinatoire du polyèdre (FIG. 3.13). Pour cela un schéma de nu-

mérotation des sommets, arêtes et faces est utilisé. Un ensemble de tableaux stocke les relations entre ces cellules, qui seront utilisées pour construire et générer des images de la grille :

**nb\_sommets**, **nb\_arêtes** et **nb\_faces** donnent le nombre de primitives de dimension 0, 1, et 2 de la cellule. Cela détermine sa taille et sa complexité combinatoire ;

**arête\_vers\_sommets** permet de passer d'un index d'arête aux index des deux sommets qu'elle relie.

**sommet\_vers\_arêtes** permet de savoir si deux sommets sont reliés par une arête, et d'accéder le cas échéant à son index ;

**voisins\_dans\_face** donne le prédécesseur et le successeur d'un sommet dans une face.

Comme elles ne sont stockées qu'une seule fois par type de cellule, ces matrices d'adjacence et d'incidence ne sont pas les plus compacts possibles, mais sont conçues pour permettre des *accès rapides*. Ainsi, une représentation de matrice creuse pour, par exemple **sommet\_vers\_arêtes**, aurait un coût en temps trop élevé, pour un gain en mémoire négligeable.

Dans la méthode des *Marching Cubes* [Lorensen *et al.*, 1987], la topologie des cellules hexaédriques est également utilisée pour stocker les types possibles des polygones d'intersection dans une table des cas (FIG. 3.12). Dans cette méthode, l'utilisation de symétries permet de réduire le nombre de cas de 256 à 15.

Sur le même principe, les descriptions de cellules contiennent, en plus des informations d'incidence et d'adjacence, des informations relatives aux intersections d'une cellule avec une iso-surface. Comme les symétries ne sont pas connues par avance, le nombre de cas possibles sur une cellule comprenant  $n_s$  sommets est de  $2^{n_s}$ , suivant que la valeur de chaque sommet est inférieure ou supérieure à l'isovaleur  $\varphi$ . En pratique, la table des cas généralisée contient essentiellement deux tableaux de taille  $2^{n_s}$  ; pour une configuration donnée :

**taille\_config** donne le nombre d'arêtes intersectées ;

**config** contient la séquence ordonnée d'arêtes.

Généralisation de la table des cas des *Marching Cubes* [Lorensen *et al.*, 1987], ce système souffre des mêmes limites concernant les ambiguïtés : certaines configurations peuvent être résolues par plusieurs polygones d'intersection différents. Dans la mise en oeuvre de la méthode, un troisième tableau, **config\_ambiguë**, a été utilisé afin d'identifier ces configurations. Comme dans le cas de CIEL, les ambiguïtés pourraient être résolues par tétraédrisation, même si rien n'assure que cela produise un résultat conforme au gradient de propriétés. La structure de graphe cellulaire est également adaptée à cette méthode. Il est en effet possible

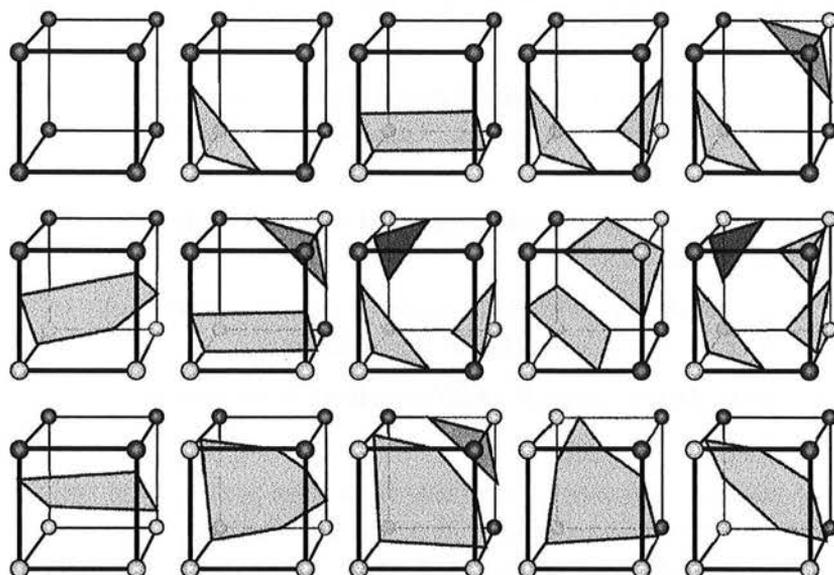


FIG. 3.12 – La table des cas de l’algorithme des *Marching Cubes* [Lorensen *et al.*, 1987]. Une analyse des symétries permet de réduire les  $2^8 = 256$  configurations à ces 15 cas de figure.

d’ajouter un schéma de tétraédrisation virtuel à la structure de méta-cellule utilisée. Encore une fois, le coût en mémoire d’un tel stockage est négligeable car chaque méta-cellule n’est stockée qu’une fois par type de cellule.

Plus généralement, la structure minimaliste définie FIG. 3.13 est tout à fait extensible pour contenir les informations supplémentaires nécessaires à d’autres algorithmes.

### 3.5.2 Construction

La construction d’un graphe cellulaire à partir d’un fichier procède en deux étapes. Dans un premier temps, une description des types de cellules est utilisée pour construire la table des méta-cellules. Dans le cas de grilles homogènes (hexaédriques ou tétraédriques), cette description est implicite ; dans le cas général de grilles faiblement hétérogènes, elle doit être explicitement donnée par une énumération des faces de chaque type de cellule, comme pour les grilles fortement hétérogènes (Cf. §3.4.2). À partir d’une description de polyèdre, une cellule abstraite, dépourvue de géométrie – peut être construite à l’aide de CIEL (ou d’une structure équivalente), comme indiqué par l’Algorithme 3.3. Par des parcours de la cellule à l’aide des liens `succ_face` et `autre_face`, il est facile d’initialiser

```

struct MetaCell {
    int nb_sommets ;
    int nb_arêtes ;
    int nb_faces ;
    int arête_vers_sommets[nb_arêtes][2] ;
    int sommet_vers_arêtes[nb_sommet][nb_sommets] ;
    int voisins_dans_face_[nb_faces][nb_sommets][2] ;
    int taille_config[nb_configs] ;
    bool config_ambiguë[nb_configs] ;
    int config[nb_configs][nb_inters_config] ;
    int config_face[nb_configs][nb_inters_config] ;
};

```

FIG. 3.13 – Structure de méta-cellule mise en place. Une description de cellule utilise des schémas d'indexage pour définir les connexions entre les sommets, les arêtes et les faces du polyèdre.

la structure de méta-cellule décrite FIG. 3.13. Une telle construction permet également de charger des grilles sans *a priori* sur les cellules qu'elles contiennent.

Dans un deuxième temps (Algorithme 3.7), les cellules et sommets de la grille sont créés et initialisés. Leur énumération dans le fichier d'entrée doit comprendre :

- pour les sommets, leurs coordonnées et leur valeur scalaire ;
- pour les cellules, l'index de la méta-cellule et la liste des sommets, dans un ordre conforme à la numérotation de la méta-cellule.

Cette information permet de créer une « soupe de polyèdres », dans laquelle les cellules ne portent pas d'informations d'adjacence. Pour connecter les cellules entre elles, une liste transitoire de cellules incidentes à chaque sommet est donc créée. Comme dans la création de CIEL, les faces partageant trois sommets consécutifs sont connectées.

### Complexité

La méthode de construction proposée est linéairement dépendante du nombre de cellules de la grille  $N_c$ . Plus précisément, sa complexité en temps est en  $O(N_c \cdot \bar{f}_c^2 \cdot \bar{s}_c \cdot \bar{c}_s)$ , où  $\bar{f}_c$ ,  $\bar{s}_c$  et  $\bar{c}_s$  sont respectivement les nombres moyens de faces par cellule, de sommets par cellule, et de cellules partageant un sommet.

### 3.5.3 Tranchage incrémental

Alors que la structure CIEL est centrée autour de la notion de demi-arête, les graphes cellulaires factorisent l'information combinatoire au niveau des cellules.

**Algorithme 3.7** Construction d'un graphe cellulaire

---

**Entrée :**  
 $M$  : table des méta-cellules  
 $S$  : Tableau des sommets de la grille **Début**  
// Étape 1 : Construction des cellules sans les connecter  
**Pour chaque** polyèdre  $p = \{type, s_1, \dots, s_k\}$   
 $p.description \leftarrow M[type]$   
**Pour**  $i = 1 \dots k$   
 $p.sommet[i] \leftarrow S[s_i]$   
**Fin Pour**  
**Fin Pour**  
// Étape 2 : connecter les cellules adjacentes  $sommet\_vers\_cells$  : table  
d'incidence des cellules aux sommets  $\leftarrow \emptyset$   
**Pour chaque** polyèdre  $p$  de la grille  
**Pour**  $i = 1 \dots k$   
**ajouter**  $p$  à  $sommet\_vers\_cells[p.sommet]$   
**Fin Pour**  
**Fin Pour**  
**Pour chaque** polyèdre  $p$  de la grille  
**Pour**  $i = 1 \dots p.description.nb\_faces$   
Sommet  $s_1, s_2, s_3 \leftarrow$  trois sommets successifs de la face  $i$   
**Pour chaque** polyèdre  $p'$  dans  $sommet\_vers\_cells[s_2]$   
**Si**  $p' \neq p$  et  $p'$  contient  $v_1, v_2, v_3$   
 $i_v \leftarrow$  face de  $p'$  contenant  $s_1, s_2, s_3$   
 $p.voisin[i] \leftarrow p'$   
 $p'.voisin[i_v] \leftarrow p$   
**Fin Si**  
**Fin Pour**  
**Fin Pour**  
**Fin Pour Fin**

---

Naturellement, l'algorithme de tranchage incrémental pour visualiser des grilles faiblement hétérogènes utilise une file de *cellules* actives.

### Propagation

La mise à jour des éléments actifs entre deux tranches utilise les adjacences entre cellules, de manière similaire à l'algorithme de résolution de visibilité *MPVO* [Williams, 1992]. Plus précisément, l'ordre de propagation – d'arrière en avant pour le rendu volumique – fait intervenir l'orientation des faces de cellules par rapport à l'isovaleur  $\varphi$ . Stockée dans le tableau *marques\_faces*, cette orientation est pré-calculée en comparant la normale à la face considérée au gradient

---

**Algorithme 3.8** Désactivation d'une cellule durant la propagation.

---

**désactiver****Entrée :** $c$  : cellule $cells\_balayées$  : file des cellules balayées**Début****Pour**  $i = 1 \dots c.description.nb\_faces$ **Si** la face  $i$  de  $c$  est sortante**Ajouter**  $c.voisin[i]$  à  $cells\_balayées$ **Fin Si****Fin Pour****Fin**

---

de d'isovaleur. Or, comme l'orientation réelle des cellules n'est pas directement disponible dans C-Graph, le calcul des normales à partir des sommets d'une face n'est connu qu'au signe près. Pour choisir une normale pointant toujours vers l'extérieur, deux stratégies ont été testées, dans l'hypothèse de cellules convexes :

- par l'angle entre la normale calculée et le vecteur reliant le barycentre de la face au barycentre de la cellule ;
- par le calcul du volume signé de la cellule [Gelder, 1995].

Les deux méthodes fonctionnent correctement dans la plupart des cas. En revanche, la méthode utilisant le volume est apparue plus robuste pour des cellules possédant des faces d'aires très faibles, ou nulles. Dans les deux cas cependant, la précision numérique limitée des calculs peut produire des résultats inexacts. Pour cette raison, les cellules contenant des faces dégénérées ou de volume non significatif sont simplement ignorées (à l'aide la marque `fantôme`), de manière similaire à [Williams, 1992]. Afin d'éviter toute incohérence et de réduire les calculs, les orientations des faces sont également initialisées de manière cohérente pour chaque couple de cellules adjacentes (en d'autres termes, la face entrante d'une cellule est forcément face sortante dans la cellule voisine).

Les marques des faces de cellules permettent une mise à jour rapide des structures d'éléments actifs par les Algorithmes 3.2 et 3.8 : lorsqu'une cellule balayée n'est plus intersectée, ses successeurs potentiellement actifs sont donnés par ses faces sortantes.

**Cellules minimales**

Les cellules ne pouvant être atteintes par ce mécanisme de propagation doivent être traitée entre chaque isovaleur. Il s'agit de cellules dont *toutes* les faces entrantes sont sur le bord de la grille.

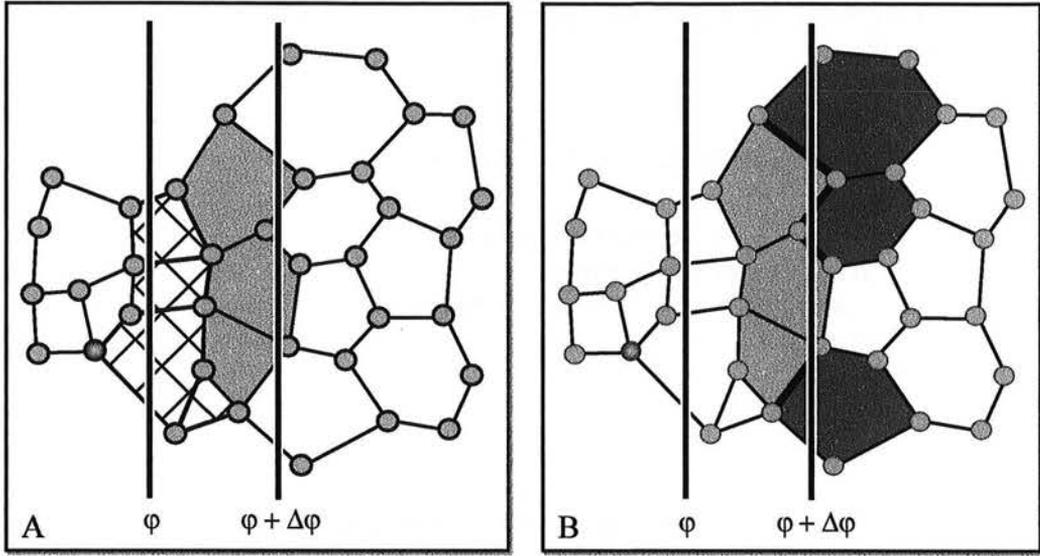


FIG. 3.14 – Propagation dans un graphe cellulaire lors du tranchage incrémental. Entre une isovaleur  $\varphi$  et la suivante ( $\varphi + \Delta\varphi$ ), les successeurs (en gris) des cellules désactivées (hachurées) sont ajoutés aux éléments balayés (FIG. A, Algorithme 3.8). Comme la désactivation ne permet pas d'obtenir l'ensemble des cellules intersectées, l'algorithme de dessin (Algorithme 3.9) détecte les cellules manquantes dans la liste d'éléments actifs (en gris foncé, FIG. B), à l'aide des faces intersectées.

### Dessin des cellules actives

Une cellule intersectée peut être dessinée très rapidement à partir de la table des configurations disponibles dans les méta-cellules. Chaque sommet possède en effet une valeur supérieure (codée 1) ou inférieure (codée 0) à l'isovaleur  $\varphi$ . Pour l'ensemble des sommets d'une cellule, un nombre entier peut ainsi être généré, donnant l'entrée dans la table des cas de la méta-cellule (fonction `calculer_config` dans l'Algorithme 3.9). La liste ordonnée des arêtes intersectées ainsi obtenue est ensuite utilisée pour calculer les sommets du polygone d'intersection (Algorithme 3.9).

Comme le montre la FIG. 3.14, l'algorithme de mise à jour entre deux isosurfaces ne permet pas d'atteindre toutes les cellules intersectées. L'algorithme de propagation pourrait être modifié pour détecter ces cellules, mais de manière non optimale. En effet, les cellules oubliées sont toujours accessibles par les faces sortantes des cellules actives recoupées par l'isovaleur  $\varphi$ . Ainsi, l'algorithme de dessin permet de détecter les cellules oubliées lors de la propagation en utilisant directement l'information des faces intersectées (pour cela, la table `config_face` a été ajoutée à la structure de méta-cellule).

---

**Algorithme 3.9** Dessin d'une cellule et mise à jour de la file des éléments actifs.
 

---

**dessiner\_et\_activer****Entrée :**

$c$  : cellule à dessiner  
 $\varphi$  : isovaleur courante  
 $cellules\_actives$  : file des cellules actives

**Début**

$meta$  : Meta-Cellule  $\leftarrow c.description$   
 $conf$  : entier  $\leftarrow \text{calculer\_config}(c, \varphi)$

**début\_polygone****Pour**  $i = 1 \dots meta.taille\_config$ 

// Dessiner l'intersection courante

 $a_{cur}$  : entier  $\leftarrow meta.config[conf][i]$  $s_1, s_2$  : entiers  $\leftarrow meta.arête\_vers\_sommets[a_{cur}]$ **sommet**( **intersection**(  $c, s_1, s_2, \varphi$  ) )

// Ajouter si nécessaire le voisin en fin de file des cellules actives

 $c_{suiv}$  : cellule  $\leftarrow c.voisin[meta.config\_face[conf][i]]$ **Si non**  $c_{suiv}.active$   **ajouter**  $c_{suiv}$  à  $cellules\_actives$ **Fin Si****Fin Pour****fin\_polygone****Fin**

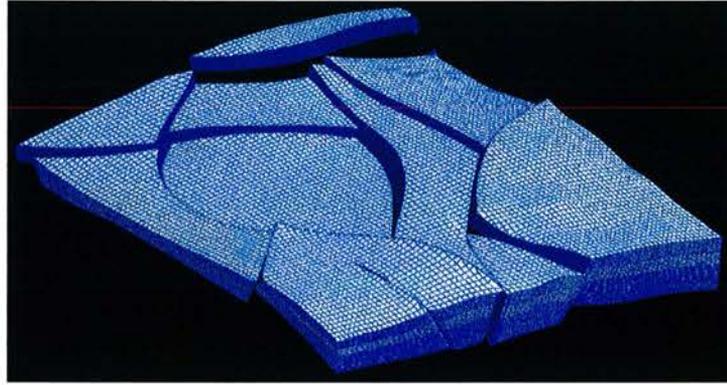

---

### Complexité

La complexité du rendu est sensiblement la même dans le cas de grilles faiblement hétérogènes que dans le cas de grilles fortement hétérogènes (p. 112). Le dessin seul de  $N_\varphi$  isovaleurs s'effectue en  $O(N_\varphi \cdot N_c \cdot \bar{s}_c \cdot \bar{s}_i)$ , où  $N_c$  est le nombre total de cellules, et  $\bar{s}_c$  et  $\bar{s}_i$  représentent les nombre moyens de sommets par cellule et de sommets par polygone d'isovaleur respectivement. Ainsi, dès lors que le nombre d'isovaleurs échantillonne la grille de manière représentative, la propagation entre deux isovaleurs successives nécessite un temps dépendant linéairement du nombre de cellules intersectées.

### 3.5.4 Limites de la méthode

Comme dans l'algorithme d'ordonnement de polyèdres *MPVO* [Williams, 1992], les cellules ou les faces dégénérées posent des problèmes pour calculer le statut des faces. Une orientation incorrecte des faces pose d'importants problèmes dans l'algorithme de visualisation, car des cycles peuvent apparaître, ce qui donne lieu à des boucles infinies.



A

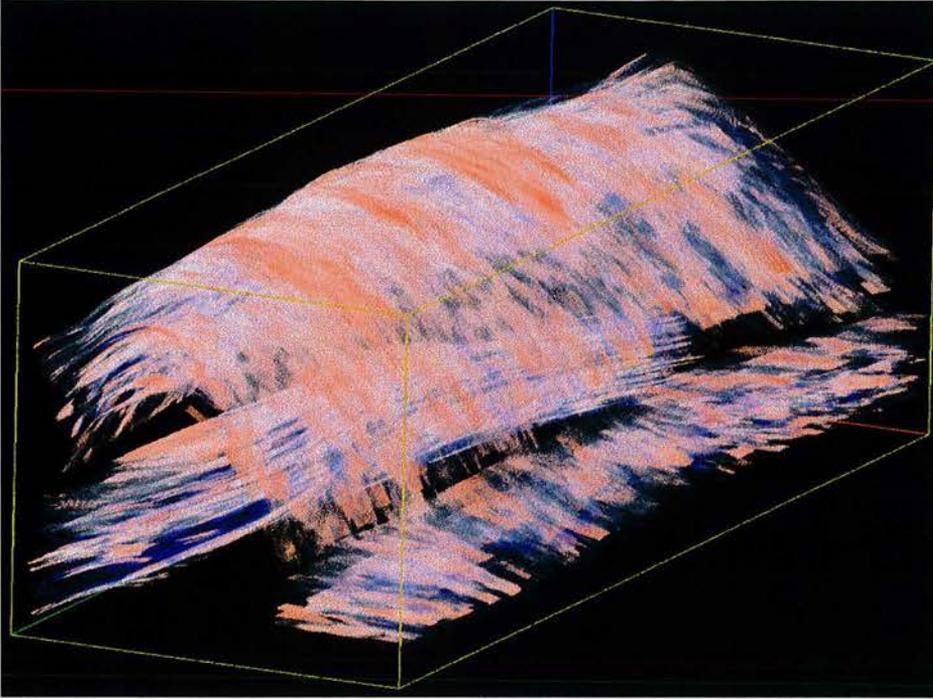


B

FIG. 3.15 – La grille stratigraphique faillée Mandaros, munie d'une propriété aléatoire lissée par DSI.

Dans l'approche développée, les cellules et faces dégénérées sont détectées dès le chargement de l'objet, et marquées comme fantômes. Par la suite elles sont ainsi ignorées par la méthode de propagation.

La suppression virtuelle des dégénérescences n'est cependant pas idéale. En effet, s'il est difficile d'éviter un calcul de normales aux faces, le calcul de volumes signés pourrait être éliminé par l'utilisation plus efficace et plus robuste de l'information combinatoire. En effet les connexions entre cellules pourraient être utilisées pour déterminer si un polyèdre a la même orientation que sa méta-cellule ou l'orientation opposée. La structure de *graphe cellulaire orientée* ainsi définie

FIG. 3.16 – La grille stratigraphique faillée *FGrid*, rendue avec 700 tranches.

	Mandaros	FGrid
Sommets	236 874 (10,4 MB)	1 159 400 (51 MB)
Cellules	206 912 (5 MB)	1 085 364 (26 MB)
Minima locaux	1,7 MB	25 KB
TOTAL	17,1 MB	77 MB

TAB. 3.4 – Taille et occupation mémoire des deux grilles stratigraphiques faillées.

présenterait également un intérêt pour tout algorithme utilisant l'orientation de la grille.

### 3.5.5 Résultats

En pratique, nous avons testé les graphes cellulaires sur des grilles stratigraphiques faillées. Les travaux de Lepage [2003] devraient bientôt permettre de tester la méthode sur d'autres types de grilles faiblement hétérogènes.

La première grille utilisée à été créée à partir du jeu de donnée réel *Mandaros*, fourni par Elf. Elle contient une propriété aléatoire qui a été lissée par DSI (FIG. 3.15). La seconde grille stratigraphique, *FGrid*, porte quant à elle une simulation géostatistique à haute fréquence (FIG. 3.16).

L'occupation mémoire et les performances pour ces deux grilles sont résumées

		Mandaros	FGrid
Chargement		22,6	117
Pré-traitement		1,3	6
Rendu volumique	100	0,6	7,1
	500	1,9	11,1
	1000	3,6	18,3

TAB. 3.5 – Performances obtenues pour le chargement et la visualisation des deux grilles stratigraphiques faillées avec les graphes cellulaires, en s.

```

struct SommetCell {
    float géométrie[3];
    float propriété;
    float gradient[3];
    Marques marques;
};

```

FIG. 3.17 – La primitive d'un graphe cellulaire structuré contient des informations relatives aux sommets (géométrie, propriétés, gradient) et aux cellules (marques des faces entrantes ou sortantes, et marques des cellules actives / virtuelles).

dans les TABS. 3.4 et 3.5. Les grandeurs figurant dans ces tableaux sont décrites au Paragraphe 3.4.4, p. 113.

## 3.6 Grilles curvilinéaires

### 3.6.1 Principe

Développée pour les grilles non structurées faiblement hétérogènes, la structure de graphe cellulaire est également applicable aux grilles non structurées homogènes ainsi qu'aux grilles structurées. Cependant, alors qu'une représentation explicite par graphe est nécessaire pour tous les types de grilles non structurées, elle devient superflue pour les grilles structurées, dans lesquelles les connectivités entre cellules sont implicites. Comme les grilles régulières sont encore beaucoup utilisées en Géosciences (grilles stratigraphiques), une structure a été définie pour leur appliquer la méthode du tranchage incrémental.

Le principe de cette représentation est exactement le même que pour les graphes cellulaires, d'où le nom de *Graphes Cellulaires structurés*. Dans cette représentation, seule une méta-cellule hexaédrique est utilisée. La partie de la grille dynamique est pour sa part stockée dans un tableau 3D de `SommetCell`, structure hybride contenant des informations relatives à la fois aux sommets et aux cellules FIG. 3.17. Afin d'éviter les effets de bord, une rangée de cellules virtuelles

est rajoutée en bordure de la grille. De manière classique, le schéma d'indigage du tableau permet de trouver facilement les relations d'adjacence entre cellules.

## 3.7 Comparaison des méthodes

### 3.7.1 Comparaison sur des grilles curvilinéaires

Les trois spécialisations de l'algorithme générique ont été réalisées pour exploiter au mieux les spécificités de chaque type de grille. Ainsi, les graphes cellulaires permettent de factoriser une partie importante de l'information combinatoire dans les méta-cellules. Mais constituent-ils pour autant une amélioration significative par rapport listes d'arêtes ?

Pour répondre à cette question, la même grille curvilinéaire a été visualisée successivement avec CIEL, CGRAPH et SCGraph. Les variations, résumées dans le TAB. 3.6 concernent le temps de chargement et de rendu de la grille. Pour charger la grille avec CIEL, nous avons eu recours à un PC doté de 2 GB de mémoire et d'un biprocesseur à 2,4 GHz pour effectuer ces tests.

Comparée à la structure CIEL, les approches utilisant les cellules offrent bien entendu de meilleurs résultats car elles sont plus compactes en mémoire. Les images obtenues sont quant à elles strictement identiques quelle que soit la méthode.

		CIEL	CGraph	SCGraph
Taille mémoire (MB)		1452	108.4	69.2
Chargement (s.)		83	45	20.1
Rendu Volumique (s)	100	12	2.9	2.7
	500	30	16.6	13.59
	1000	53	31.3	27.17

TAB. 3.6 – Comparaison de CIEL, des graphes cellulaires et des graphes cellulaires structurés sur une grille curvilinéaire ().

En complément des tableaux de chiffres précédents, qui ont surtout pour objet de donner des ordres de grandeur, la FIG. 3.18 montre expérimentalement l'optimalité de notre algorithme. Pour générer ce graphe, nous avons chargé des grilles cubiques régulières de taille variable, et mesuré les performances pour y calculer 200 tranches. Le nombre d'intersections a été estimé comme  $N_c^{2/3}$ , où  $N_c$  est le nombre de cellules de la grille.

En utilisant l'information combinatoire, notre algorithme a donc la même complexité que celui de Yagel *et al.* [1996], sans pour autant utiliser des structures annexes coûteuses à initialiser et à maintenir.

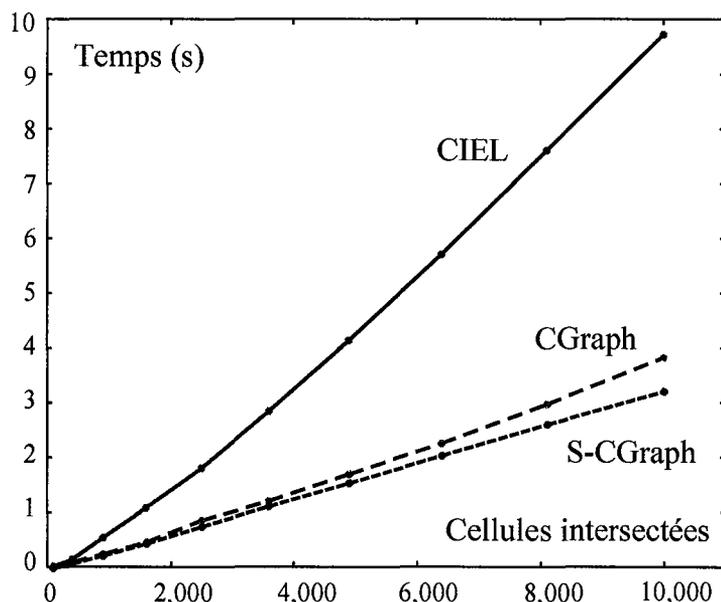


FIG. 3.18 – Le temps de calcul d’une image est proportionnel au nombre de polygones d’intersection dessinés (pour  $N_c$  cellules, il y a en moyenne  $N_c^{2/3}$  cellules intersectées par tranche).

### 3.7.2 Perspectives

La factorisation des informations combinatoires dans les méta-cellules permet de réduire considérablement la consommation des grilles en mémoire. Il serait donc intéressant d’étudier la rentabilité d’une représentation de grilles fortement hétérogènes par des graphes cellulaires. Pour cela, la création dynamique des méta-cellules vue au paragraphe 3.5.2 pourrait être utilisée. Au cours du chargement, la comparaison entre méta-cellules serait alors nécessaire. À cet effet, un algorithme inspiré de [Hilaga *et al.*, 2001] pourrait être mis en place.

## 3.8 Conclusion et perspectives

Dans ce chapitre, la visualisation des différents types de grilles utilisées en Géosciences ainsi que dans d’autres domaines scientifiques a été étudiée. Utilisant le concept du tranchage incrémental initialement proposé par Yagel *et al.* [1996], un algorithme générique a été développé pour permettre à la fois le rendu volumique et le calcul de séries d’iso-surfaces. Cette méthode est optimale pour le rendu volumique, car le temps de rendu est linéairement dépendant du nombre de polygones d’intersection dessinés. Signalons toutefois que la méthode proposée n’est pas toujours optimale pour le calcul d’isovaleurs car la complexité de

la propagation n'est pas linéaire pour un sous-échantillonnage de la grille ; elle constitue donc seulement un pis-aller par rapport à la méthode des *graines de contour* [Bajaj *et al.*, 1996].

Les structures de données ont été adaptées pour chaque type de grille, avec dans l'esprit d'optimiser l'algorithme de visualisation et de réduire au maximum leur occupation en mémoire. À cet égard, nous avons constaté expérimentalement que ces deux objectifs étaient concordants pour des grilles de taille courante (entre quelques dizaines de milliers et quelques millions de cellules). L'accès à l'information est donc le facteur limitant de la méthode de visualisation mise en oeuvre.

Dans sa conception, l'algorithme proposé utilise dès que possible l'information combinatoire aux dépens de la géométrie, ce qui permet :

- d'éliminer les structures annexes ayant des coûts de stockage et de maintenance élevés (par exemple les *edge buckets* dans [Yagel *et al.*, 1996]) ;
- de diminuer la complexité par rapport à d'autres approches utilisant le balayage [Silva *et al.*, 1997] ;
- d'améliorer la robustesse des algorithmes par rapport aux méthodes numériques. En ce sens, le développement de graphes cellulaires orientés (§3.5.4 p. 123) présenterait un intérêt certain.

Du point de vue de la visualisation elle-même, plusieurs évolutions pour ce travail sont envisagées. Elles concernent :

- la gestion des ambiguïtés apparaissant lors de variations non monotones de propriété, ou dans des cellules concaves (§3.3.3 p. 103) ;
- une estimation non linéaire des propriétés dans des cellules polyédriques quelconques, liée à une meilleure estimation du gradient [Möller *et al.*, 1997 ; Neumann *et al.*, 2000] ;
- l'amélioration de l'interface avec l'utilisateur. Dans le tranchage incrémental, l'utilisateur contrôle le nombre d'isovaleurs dessinées à l'écran, et peut ainsi trouver un compromis entre la vitesse et la qualité du rendu. Cependant, pour le rendu volumique, la fonction de transfert reste la même lorsque le nombre de tranches varie, ce qui a tendance à opacifier les images lorsque le nombre de tranches augmente ;
- la recherche de fonctions de transfert adaptées à certain type de données géophysiques ou géologiques [Vickery *et al.*, 2002 ; Gerhardt *et al.*, 1998, 2002] ;

Bien entendu, la méthode de visualisation décrite dans ce chapitre ne constitue pas une fin en soi. Elle se veut un outil supplémentaire pour l'analyse et l'interprétation des données volumiques rencontrées dans les domaines des Géosciences. L'esthétique des images générées ne doit donc en aucun cas prendre le pas sur leur aspect informatif.

Plus généralement, les structures définies dans ce travail, en particulier les graphes cellulaires et la notion de méta-cellules pourraient être utilisées pour d'autres opérations de Géomodélisation. Plus compacte que les Cartes Généralisées [Lévy, 1999 ; Conreux, 2001], une variante de la structure CIEL est précisément en cours de développement dans Gocad.

Par ailleurs, l'algorithme de propagation utilisé pour la visualisation pourrait être adapté aux recherches de voisinage nécessaires en géostatistique [Rémy *et al.*, 2002]. Une telle méthode faciliterait ainsi la mise en oeuvre des méthodes géostatistiques directement sur les grilles utilisées en simulation d'écoulement.

# Conclusion

## Bilan

La modélisation tridimensionnelle des structures du sous-sol fait appel à différents types de modèles volumiques en fonction des besoins. Pour passer des données de subsurface à un modèle volumique satisfaisant, de nombreuses étapes restent nécessaires. Lors de chacune de ces étapes, un contrôle visuel s'impose comme une nécessité. En cours de construction, il permet d'effectuer des choix d'interprétation et de vérifier la validité des représentations intermédiaires. Plus généralement, il permet de comprendre et de valider les résultats obtenus par les méthodes de Géomodélisation. Dans certains cas, suite au comportement anormal d'un algorithme ou au manque de données, les géométries observées peuvent apparaître peu réalistes. Dans ce cadre, des modifications interactives sous contraintes sont commodes pour intégrer une expertise géologique dans un modèle du sous-sol.

Dans ce mémoire, nous avons étudié et mis en place des méthodes de visualisation et de modification géométrique pour trois types de représentations volumiques utilisées en Géosciences.

Dans le Chapitre 1, la méthode de modélisation d'architectures stratigraphiques de Mallet [2002] a été présentée. La représentation utilisée est définie par l'extrusion d'une surface de référence, éventuellement faillée. Les améliorations apportées dans cette thèse concernent principalement la définition de coupes modifiables dans ces modèles volumiques. Ces coupes permettent de visualiser précisément et, le cas échéant, de rectifier la géométrie des couches tout en respectant les données d'observation. Plutôt qu'une édition des coupes elles-mêmes, qui risquerait d'invalidier la cohérence du modèle volumique, ce dernier est modifié directement sous contraintes. Un couplage étroit entre la coupe et le modèle volumique permet de visualiser les effets de la modification.

L'objet du Chapitre 2, était une autre représentation à base de surfaces triangulées, dite représentation par frontières. Nous y avons introduit deux conditions

nécessaires pour assurer la validité géologique de ces modèles. Dans ce travail, elles nous ont servi à définir une méthode pour déformer sous contraintes certaines interfaces, sans pour autant avoir à reconstruire l'ensemble du modèle volumique. Mise en place dans un contexte de manipulation interactive, cette approche utilise essentiellement des retriangulations locales dans un espace paramétrique. Elle ne dépend pas de la méthode de déformation et pourrait donc être également adaptée à des méthodes automatiques de réajustement de géométrie comme l'inversion tomographique.

Pour sa part, le Chapitre 3 a mis l'accent sur les représentations volumiques cellulaires, et plus précisément sur les grilles non structurées, adaptée à la résolution numérique d'équations aux dérivées partielles. Du fait de la taille et de la complexité combinatoire des grilles non structurées, il est difficile d'extraire et d'ordonner l'information qu'elles contiennent de manière efficace. Dans ce cadre, nous avons défini un nouvel algorithme de visualisation par tranchage incrémental. De complexité optimale, cet algorithme générique permet de générer des images de rendu volumique, des coupes, ou des surfaces d'isovaleur. La mise en oeuvre de cet algorithme pour différents types de grilles s'est traduite par la définition de structures de données efficaces et complètes.

Au point de vue pratique, la plupart des méthodes décrites dans cette thèse ont été développées dans un plugin du géomodeleur Gocad. Certaines, comme la modification d'architectures stratigraphiques ou l'édition de rejets de faille font l'objet d'industrialisation par la compagnie *Earth Decision sciences*. Les structures définies dans le Chapitre 3 ont simplement été conçues pour répondre aux opérations de chargement et de visualisation de grilles non structurées. Elles servent également de base à des travaux en cours plus généraux qui comprennent entre autres la génération de maillages non structurés dans Gocad.

## Perspectives

Les perspectives de notre étude sont de plusieurs ordres. Concernant les représentations par frontières, les critères de validité géologique définis dans ce travail nous semblent ouvrir la voie à des méthodes de mise en cohérence automatique. De telles méthodes présenteraient un grand intérêt lors de la construction des modèles pour s'affranchir d'éventuelles erreurs numériques, ou plus simplement pour guider l'interprétation.

La mise en oeuvre de la modification de b-reps elle-même est encore perfectible. Il reste en effet à généraliser les différentes composantes de la méthode pour autoriser un déplacement simultané sur plusieurs surfaces de glissement. De plus,

il serait opportun de tester et comparer des différentes méthodes de détection d'interférence Lin *et al.* [1998] ; Jiménez *et al.* [2001] pour améliorer la méthode.

Dans un autre ordre d'idées, la retriangulation nécessaire lors de la mise à jour du modèle peut être problématique pour certains algorithmes Grosse [2002]. Pour cette raison, il serait intéressant d'appliquer la méthode de modification à la représentation de « Trimmed TSurfs » Grosse [2002], qui permettrait de laisser intacte la triangulation.

Durant cette thèse, nous nous sommes également concentrés sur les problèmes liés de visualisation et de représentation des grilles. Leur modification interactive n'a donc pas été traitée. Pour aborder ce problème, une méthode pourrait consister à combiner la méthode d'édition b-reps étudiée dans le Chapitre 2 avec des méthodes locales de maillage volumique François *et al.* [2000].

Enfin, l'utilisation des principes et des structures de données définis dans le Chapitre 3 pour des usages plus généraux est dans la continuité de ce travail. L'application de ces représentations à des méthodes de génération de maillage de calculs géostatistiques ou de simulation d'écoulement passe en particulier par une généralisation et des spécification plus poussées pour les Graphes Cellulaires.



# Bibliographie

- ABRAMOWITZ, M. & STEGUN, I., Eds (1972). - Handbook of Mathematical Functions, volume 55 of *Applied Mathematic.* - U.S. Government Printing Office, 10 edition. 30
- AIT ETTAJER, T. (1995). - Modélisation de Surfaces Géologiques Complexes sous Contraintes Géométriques. Ph.D. thesis. - INPL, Nancy, France. 35
- ALLIEZ, P., MEYER, M., & DESBRUN, M. (2002). - Interactive geometry remeshing. - *ACM Transactions on Graphics (Proc. Siggraph)*, **21**(3), 347–354. 69
- APEL, M. (2003). - 3D GIS. Ph.D. thesis. - Freiberg University. 48
- ASSA, S., CELNICKER, G., & RAMSHORN, C. (1998). - Feature-based geometric modeling for geoscience. *In* : 3D Modeling of Natural Objects : A Challenge for the 2000's. - Gocad ENSG Conference. 52
- AZIZ, K. (1993). - Reservoir simulation grids : Opportunities and problems. - *JPT*, **45**(7), 658–663. 89
- BAJAJ, C., PASCUCCI, V., & SCHIKORE, D. (1996). - Fast isocontouring for improved interactivity. *In* : Proc. Symp. on Volume visualization. - 39–46. ACM. 40, 94, 96, 97, 98, 99, 101, 129
- BARDIS, L. & PATRIKALAKIS, N. (1992). - Topological structures for generalized boundary representations. Geometric Algorithms in CS368. Handout 21. 54
- BAUMGART, B. (1975). - A polyhedron representation for computer vision. *In* : AFIPS Nat. Conf. Proc., volume 44. - 589–596. 51, 54
- BERGEN, G. V. (1997). - Efficient collision detection of complex deformable models using AABB trees. - *Journal of Graphic Tools*, **2**(4). <http://www.win.tue.nl/cs/tt/gino/solid>. 25
- BICE, D. (1989). - Synthetic stratigraphy of carbonate platform and basin systems. - *Geology*, **16**, 703–706. 20
- BIER, E. (1990). - Snap-dragging in three dimensions. - *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, **24**, 193–204. 37, 69, 72
- BIER, E. & STONE, M. (1986). - Snap-dragging. - *Computer Graphics*, **20**(4), 233–240. 37

- BIJU-DUVAL, B. (1999). - Géologie Sédimentaire – Bassins, environnements de dépôts, formation du pétrole. - Technip. 735 pp. 15
- BLOOMENTHAL, J. (1988). - Polygonalization of implicit surfaces. - *Computer Aided Geometric Design*, 5(4), 53–60. 101
- BOISSONNAT, J.-D. & PREPARATA, F. (2000). - Robust plane sweep for intersecting segments. - *SIAM J. Comput.*, 29(5), 1401–1421. 80, 81
- BOISSONNAT, J.-D. & YVINEC, M. (1995). - Géométrie Algorithmique. Collection informatique. - Ediscience international. 540 pp. 51, 75
- BONHAM-CARTER, G. F. (1994). - Geographic Information Systems for Geoscientists : Modelling with GIS. - Pergamon Press. 414 pp. 48
- BORNHOLDT, S., NORDLUND, U., & WESTPHAL, H. (1999). - Inverse stratigraphic modeling using genetic algorithms. In : HARBAUGH J., WATNEY W., RANKEY E., SLINGERLAND R., GOLDSTEIN R., & FRANSEEN E., Eds, *Numerical Experiments in Stratigraphy : Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume 62, 85–90. SEPM Special Publication. 44
- BORREL, P. & RAPPOPORT, A. (1994). - Simple constrained deformations for geometric modeling and interactive design. - *ACM Transactions on Graphics*, 13(2), 137–155. 62, 63
- BOURQUIN, S., VAIRON, J., & LE STRAT, P. (1997). - Three-dimensional evolution of the keuper of the paris basin based on detailed isopach maps of the stratigraphic cycles : tectonic influences. - *Geologische Rundschau*, 86(3). 23
- BOWMAN, S. A. & VAIL, P. R. (1999). - Interpreting the stratigraphy of the Baltimore Canyon section, offshore New Jersey with PHIL, a stratigraphic simulator. In : HARBAUGH J., WATNEY W., RANKEY E., SLINGERLAND R., GOLDSTEIN R., & FRANSEEN E., Eds, *Numerical Experiments in Stratigraphy : Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume 62, 117–138. SEPM Special Publication. 20
- BOWYER, A. (1981). - Computing dirichlet tessellations. - *Computer Journal*, 24(2), 162–166. 76
- BRAID, I. (1975). - The synthesis of solid bounded by many faces. - *Communications of the ACM*, 18, 209–216. 46
- BREUNIG, M. (1999). - An approach to the integration of spatial data and systems for a 3d geo-information system. - *Computer and Geosciences*, 25(1), 39–48. 48
- BURTON, R., KENDALL, C. G., & LERCHE, I. (1987). - Out of our depth : on the impossibility of fathoming eustasy from the stratigraphic record. - *Earth Science Review*, 24, 237–277. 19, 21, 22
- CAERS, J. (2001). - Geostatistical reservoir modeling using statistical pattern

- recognition. - *Journal of Petroleum Science and Engineering*, **29**(3), 177–188.  
44
- CAUMON, G., SWORD, C. H., & MALLET, J.-L. (2002). - Interactive editing of sealed geological 3d models. *In* : Terra Nostra, (Proceedings of IAMG), volume 04. - 75–80. 46, 63
- CAUMON, G., LÉVY, B., & PAUL, J.-C. (2003a). - Combinatorial data structures for volume rendering unstructured grids. - *IEEE Transactions on Visualization and Computer Graphics (Soumis)*, **xx**(xx). 101
- CAUMON, G., SWORD, C. H., & MALLET, J.-L. (2003b). - Constrained modifications of non-manifold b-rep models. *In* : ELBER G. & SHAPIRO V., Eds, *Proc. ACM Symp. on Solid Modeling and Applications*. - -. 46, 52, 63
- CHEW, L. P. (1989). - Constrained delaunay triangulations. - *Algorithmica*, **4**(1), 97–108. 77
- CIGNONI, P., MONTANI, C., PUPPO, E., & SCOPIGNO, R. (1997). - Multi-resolution modeling and visualization of volume data. - *IEEE Trans. Visual Comput. Graphics*, **3**(4), 352–369. 96
- CIGNONI, P., MONTANI, C., & SCOPIGNO, R. (1998). - Tetrahedra based volume visualization. *In* : HEGE H.-C. & POLTHIER K., Eds, *Mathematical Visualization*, 3–18. Springer Verlag. 94
- CIGNONI, P., FLORIANI, L. D., MAGILLO, P., PUPPO, E., & SCOPIGNO, R. (2000). - Volume visualization of large tetrahedral meshes on low cost platforms. *In* : Proc. NSF/DoE Lake Tahoe Workshop on Hierarchical Approximation and Geometrical Methods for Scientific Visualization. - . 96
- CLAERBOUT, J. (1985). - *Imaging the Earth Interior*. - Blackwell Publishing (out of print). <http://sepwww.stanford.edu/sep/prof/index.html>. 50
- COGNOT, R., AÏT ETTAJER, T., & MALLET, J.-L. (1997). - Modeling discontinuities on faulted geological surfaces. *In* : SEG technical program. - 1711–1718. 35
- COJAN, I. & RENARD, M. (1999). - *Sédimentologie*. - Dunod. 418 p. 15, 16
- COMBA, J., KLOSOWSKI, J. T., M., N., MITCHELL, J. S. B., SILVA, C. T., & WILLIAMS, P. L. (September 1999). - Fast polyhedral cell sorting for interactive rendering of unstructured grids. - *Computer Graphics Forum*, **18**(3), 369–376. 94, 96
- CONNER, D. B., SNIBBE, S. S., HERNDON, K. P., ROBBIN, D. C., ZELENICK, R. C., & DAM, A. V. (1992). - Three-dimensional widgets. - *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, **25**, 183–188. 37, 69
- CONRAUD, J. (1997). - Génération de maillages de simplexes pour la modélisation d'objets naturels. Ph.D. thesis. - INPL, Nancy, France. 56, 76, 77, 78, 89
- CONREAUX, S. (2001). - Modélisation de 3-Variétés à base topologique : appli-

- cation à la géologie. Ph.D. thesis. - INPL, Nancy, France. 33, 34, 78, 82, 106, 107, 114, 130
- COQUILLART, S. (1990). - Extended free-form deformation : A sculpturing tool for 3D geometric modeling. - *Computer Graphics*, **24**(4), 187–196. 61
- CROSS, T. A. & LESSENGER, M. A. (1999). - Construction and application of a stratigraphic inverse model. *In* : HARBAUGH J., WATNEY W., RANNEY E., SLINGERLAND R., GOLDSTEIN R., & FRANSEEN E., Eds, *Numerical Experiments in Stratigraphy : Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume 62, 69–83. SEPM Special Publication. <http://www.mines.edu/Academic/geology/gsrp/Pubs/publications.htm>. 21, 22, 23, 44
- CROSS, T. A. & LESSENGER, M. A. (2002). - The scientific basis for stratigraphic inversion. submitted to Geological Society of America Bulletin. <http://www.mines.edu/Academic/geology/gsrp/Pubs/publications.htm>. 22, 44
- CSATO, I. & KENDALL, C. G. (2002). - Modeling of stratigraphic architectural patterns in extensional settings – towards a conceptual model. - *Computer and Geosciences*, **28**(3), 351–356. 20
- DEMICO, R. V. & SPENCER, R. J. (1989). - Maps : A basic program to model accumulation of platform sediments. - *Computer and Geosciences*, **15**, 95–105. 20
- DEUTSCH, C. V. & JOURNEL, A. G. (1992, 1997). - GSLIB Geostatistical Software Library and user's guide. Applied Geostatistics. - Oxford University Press, 2nd edition. 384 pp. 23, 44, 88
- DUAN, T., GRIFFITHS, C. M., & JOHNSEN, S. O. (1999). - A new approach to reservoir heterogeneity modelling : conditional simulation of 2-d parasequences in shallow marine depositional systems using an attributed controlled grammar. - *Computers and Geosciences*, **25**, 667–681. 23
- DUVINAGE, I. (2000). - Création et mise en cohérence de modèles structuraux à partir d'horizons extraits de données sismiques tridimensionnelles. Ph.D. thesis. - INPL, Nancy, France. 24, 27, 30
- DYT, C. & GRIFFITHS, C. (2001). - Modelling carbonates with fuzzy logic and its integration into the forward stratigraphic modelling package sedsim. *In* : Proc. Exploration Geodynamics. - <http://www.ned.dem.csiro.au/research/solidMech/Geodynamics/ChapmanConference/AbstractsReceived/Abstracts-A-Z.html>. 20
- EDMONDS, J. (1960). - A combinatorial representation for polyhedral surfaces. - *Notices Amer. Math. Soc.*, **7**. 106
- EMERY, D. & MYERS, K. J. (1996). - Sequence Stratigraphy. - Blackwell Science. 17, 18, 19
- ERTERIN, T., ABOU-KASSEM, J. H., & KING, G. R. (2001). - Basic Applied

- Reservoir simulation. SPE textbooks. - SPE. 88, 89
- EULER, N. (1999). - Modélisation volumique, Contraintes et Libertés. Ph.D. thesis. - INPL, Nancy, France. 34, 50, 52, 59, 60, 63, 68, 73, 85
- EULER, N., SWORD, C. H., & DULAC, J.-C. (1998). - A new tool to seal a 3d earth model : a cut with constraints. *In : Society of Exploration Geophysicists 68th Annual Meeting.* - . 59, 61
- EULER, N., SWORD, C. H., & DULAC, J.-C. (1999). - Editing and rapidly updating a 3d earth model. *In : Society of Exploration Geophysicists 69th Annual Meeting.* - . 59
- FARIAS, R., MITCHELL, J., & SILVA, C. T. (2000). - Zsweep : An efficient and exact projection algorithm for unstructured volume rendering. *In : Proc. of ACM/IEEE Volume Visualization Symposium.* - . 96, 99, 100
- FARIN, G. (2001). - Curves and Surfaces for Computer Aided Geometric Design : a practical guide. *Computer Graphics and Geometric Modeling.* - Morgan-Kaufmann, 5th edition. 520 pp. 40, 72, 73
- FLOATER, M. (1997). - Parametrization and smooth approximation of surface triangulations. - *Computer Aided Geometric Design*, **14**(3), 231-250. 69
- FOLEY, J., DAM, A. V., FEINER, S., & HUGHES, J. (1990, 1997). - *Computer Graphics Principles and Practice.* - Addison-Welsey, 2e edition. 47, 69, 91, 98, 104
- FRANÇOIS, V. & CUILLIÈRE, J.-C. (2000). - Automatic remeshing applied to model modification. - *Computer Aided Design*, **32**(7), 433-444. 133
- FRØYLAND, L. A., LAKSÅ, A., STROM, K., & PAJCHEL, J. (1993). - A 3d cellular, smooth boundary representation modelling system for geological structures. *In : EAEG 55th Meeting and technical Exhibition.* - . 49, 50, 54
- GALLOWAY, W. E. (1989). - Genetic stratigraphic sequences in basin analysis I : architecture and genesis of flooding surface bounded depositional units. - *AAPG Bulletin*, **73**, 125-142. 18, 22
- GAMMA, E., HELM, R., JOHNSON, R., & VLISSIDES, J. (1994). - *Design Patterns : Elements of Reusable Object-Oriented Software.* - Addison Wesley. 37
- GARRITY, M. P. (November 1990). - Raytracing irregular volume data. - *Computer Graphics (San Diego Workshop on Volume Visualization)*, **24**(5), 35-40. 93
- GELDER, A. V. (1995). - Efficient computation of polygon area and polyhedron volume. *In : ALAN W P., Ed, Graphics Gems V*, 35-41. Academic Press. 121
- GERHARDT, A., PAIVA, A., SCHMIDT, A. E., MARTHA, L. F., CARVALHO, P. C., & GATTASS, M. (1998). - Aspects of 3d seismic data volume rendering. *In : 3D Modeling of Natural Objects : A Challenge for the 2000's.* - Gocad ENSG Conference. 129

- GERHARDT, A., MACHADO, M., SILVA, P. M., & GATTASS, M. (2002). - Enhanced visualization of 3-d seismic data. *In* : Proc. SEG Annual Meeting. - Society of Exploration Geophysicists. 92, 129
- GIERTSEN, C. (March 1992). - Volume visualization of sparse irregular meshes. - *IEEE Computer Graphics & Applications*, **12**(2), 40–48. 96
- GJØYSTDAL, H., REINHARDSEN, J. E., & ÅSTEBØL, K. (1985). - Computer representation of complex three-dimensional geological structures using a new solid modeling technique. - *Geophysical Prospecting*, **33**(8), 1195–1211. 49, 50, 54
- GLEICHER, M. (1992). - Intergating constraints and direct manipulation. - *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, **25**, 171–174. 69
- GOFF, J. A. (2000). - Simulation of stratigraphic architecture from statistical and geometrical characterizations. - *Mathematical Geology*, **32**(7). 23, 44
- GOLUB, G. H. & LOAN, C. F. V. (1996). - Matrix Computations. Studies in the Mathematical Sciences. - John Hopkins University Press, 3rd edition. 694 pp. 104
- GOOVAERTS, P. (1997). - Geostatistics for natural resources evaluation. Applied Geostatistics. - Oxford University Press. 483 pp. 44, 88
- GROSSE, O. (2002). - Mise en cohérence automatique d'un modèle géologique 3D légèrement perturbé. Ph.D. thesis. - INPL. 33, 34, 54, 56, 61, 62, 63, 86, 133
- GUIZIOU, J.-L., COMPTE, P., GUILLAUME, P., SCHEFFERS, B. C., RIEPEN, M., & DER WERFF, T. V. (1990). - SISTRE : a time-to-depth conversion tool applied to structurally complex 3-d media. *In* : Proc. SEG Annual Meeting. - 1267–1270. Society of Exploration Geophysicists. 50
- GUIZIOU, J.-L., MALLET, J.-L., & MADRIAGA, R. (1996). - 3-d seismic reflection tomography on top of the gocad depth modeler. - *Geophysics*, **61**(5), 1499–1510. 50, 86
- GUTJAR, A., BULLARD, B., & HATCH, S. (1997). - General joint conditional simulations using a fast fourier transform method. - *Mathematical Geology*, **29**(3), 361–389. 23
- HAQ, B. U., HARDENBOL, J., & VAIL, P. R. (1987). - Chronology of fluctuating sea levels since the Triassic (250 million years ago to present). - *Science*, **235**, 1156–1167. 21
- HARBAUGH & BONHAM-CARTER (1970). - Computer Simulation in Geology. - Wiley-Interscience Inc. 574 pp. 19
- HELM, R., HUYNH, T., MARRIOTT, K., & VLISSIDES, J. (1995). - An object-oriented architecture for constraint-based graphical editing. *In* : LAFFRA C., BLAKE E. H., DE MEY V., & PINTADO X., Eds, *Object Oriented Program-*

- ming for Graphics*. Springer Verlag. 37
- HENRY, G. (1997). - La Sismique Réflexion. - Editions Technip. 200 pp. 16, 50
- HILAGA, M., SHINAGAWA, Y., KOHMURA, T., & KUNII, T. L. (2001). - Topology matching for fully automatic similarity estimation of 3dshapes. *In* : Computer Graphics (Proc. Siggraph 2001). - 203–212. 128
- HOBBY, J. D. (1999). - Practical segment intersection with finite precision output. - *Computational Geometry Theory and Applications*, **13**(4), 199–214. 80, 82
- HOFFMANN, C. M. (1989a). - Geometric and Solid Modeling : An Introduction. - Morgan Kaufmann. 338 pp. 45, 46
- HOFFMANN, C. M. (1989b). - The problems of accuracy and robustness in geometric computation. - *IEEE Computer*, **22**(3), 31–41. 51
- HOMEWOOD, P. W., MAURIAUD, P., & LAFONT, F. (1999). - Best Practice in Sequence Stratigraphy for exploration and reservoir engineers, volume 25 of *Mémoires, Bulletin Centre de Recherche Exploration-Production*. - Elf Aquitaine. 81 pp. 18, 19, 21, 44
- HORMANN, K., LABSIK, U., MEISTER, M., & GREINER, G. (2002). - Hierarchical extraction of iso-surfaces with semi-regular meshes. *In* : LEE K. & PATRIKALAKIS N. M., Eds, *Proc. Seventh ACM Symp. Solid Modeling*. - ACM Press, 53–58. 96
- HOWER, W. & GRAF, W. H. (1996). - A bibliographical survey of constraint-based approaches to cad, graphics, layout, visualization, and related topics. - *Knowledge-Based Systems*, **9**(7), 449–464. 37
- HOWIE, C. T. & BLAKE, E. H. (1994). - The mesh propagation algorithm for isosurface construction. - *Computer Graphics Forum*, **13**(3), 65–74. 98, 99
- HSU, C., ALT, G., HUANG, Z., BEIER, E., & BRÜDERLIN, B. (1997). - A constraint-based manipulator toolset for editing 3d objects. *In* : HOFFMANN C. & BRONSVORT W., Eds, *Fourth Symposium on Solid Modeling and Applications*. - 168–180. 37
- HSU, W. M., HUGHES, J. F., & KAUFMAN, H. (1992). - Direct manipulation of free-form deformations. - *Computer Graphics (Proc. Siggraph)*, **26**(2), 177–184. 61
- HUBELI, A. (2002). - Multiresolution techniques for non-manifolds. Ph.D. thesis. - ETH Zürich. <http://graphics.ethz.ch/hubeli/Nemesi/Publications.html>. 52, 58, 59, 73, 106
- ITOH, T., YAMAGUCHI, Y., & KOYAMADA, K. (October 1996). - Volume thinning for automatic isosurface propagation. - *Proc. IEEE Visualization*, 303–310. 94, 96, 97, 98, 99, 101
- JACKINS, C. L. & TANIMOTO, S. L. (1980). - Octrees and their use in repre-

- senting three-dimensional objects. - *Computer Graphics and Image Processing*, **14**, 249–270. 46
- JERVEY, M. T. (1988). - Quantitative geological modeling of siliclastic rock sequences and their seismic expression. *In* : WILGUS C. K., HASTINGS B. S., KENDALL C. G. S. C., POSAMENTIER H. W., ROSS C. A., & WAGONER J. C. V., Eds, *Sea level changes – an integrated approach*, volume 42, 47–69. SEPM Special Publication. 18, 19
- JIMÉNEZ, P., THOMAS, F., & TORRAS, C. (2001). - 3D collision detection : A survey. - *Computers and Graphics*, **25**(2), 269–285. 73, 133
- JORDAN, T. E. & FLEMINGS, P. B. (1991). - Large-scale stratigraphic architecture, eustatic variation, and unsteady tectonism : a theoretical evaluation. - *Journal of Geophysical Research*, **96**(B4), 6681–6699. 20
- KAJIYA, J. T. & HERZEN, B. P. V. (1984). - Ray tracing volume densities. - *Computer Graphics*, **18**(3), 165–174. 92, 93
- KAUFMAN, A. (1996). - Volume visualization. - *ACM Computing Surveys*, **28**(1), 165–167. 90, 92, 93, 96
- KEMP, E. D. (2002). - Communication personnelle. 49
- KENDALL, C. G. & SEN, A. (1999). - Use of sedimentary simulations for dating sequence boundaries and measuring the size of eustatic sea level changes : an example from the neogene of the bahamas. *In* : HARFF J., LEMKE W., & STATTEGGER K., Eds, *Computerized Modeling of Sedimentary Systems*, 291–306. Springer. 21
- KENDALL, C. G., STROBEL, J., CANNON, R. L., BEZDEK, J., & BISWAS, G. (1991). - The simulation of sedimentary fill of basins. - *Journal of Geophysical Research*, **96**(B4), 6911–6929. 20, 21
- KENDALL, C. G., WHITTLE, G. L., R, E., MOORE, P. D., CANNON, R. L., & HELLMANN, D. R. (1993). - Computer sedimentary simulation models sequence stratigraphy. - *Oils and Gas Journal*, **91**, 46–51. 20
- KETTNER, L. (1998). - Designing a data structure for polyhedral surfaces. *In* : Proc 14th Annu. ACM Symposium on Computational Geometry. - 146–154. 107
- KNISS, J., KINDLMANN, G. L., & HANSEN, C. D. (2002). - Multidimensional transfer functions for interactive volume rendering. - *IEEE Transactions on Visualization and Computer Graphics*, **8**(3), 270–285. 92
- KOBBELT, L. P. (2000). - Discrete fairing and variational subdivision for freeform surface design. - *The Visual Computer*, **16**, 142–158. 59, 73
- KOYAMADA, K. & ITOH, T. (1995). - Fast generation of spherical slicing surfaces for irregular volume rendering. - *Visual Computer*, **11**(3), 166–175. 99
- KRÜGER, W. (1990). - The application of light transport theory to the visuali-

- zation of 3d scalar data fields. *In* : Proc. IEEE Visualization '90. - 273–280. 92
- KUTLUCA, H., KURÇ, T. M., & AYKANAT, C. (2000). - Image-space decomposition algorithms for sort-first parallel volume rendering of unstructured grids. - *The Journal of Supercomputing*, **15**(1), 51–93. 93
- LABRUNYE, E. (2004). - Analyse multi-variable d'attributs sismiques (titre provisoire). Ph.D. thesis. - INPL, Nancy, France. 27
- LAMBOGLIA, K. (1994). - Modélisation volumique de surfaces non-manifold. Ph.D. thesis. - INPL, Nancy, France. 54, 56
- LECOUR, M. (2000). - Modélisation des incertitudes concernant la géométrie des failles. Impact sur les études volumétriques des réservoirs pétroliers. Ph.D. thesis. - INPL, Nancy, France. 44, 53, 57, 66
- LEDEZ, D. (2002). - Euclidean distance mapping : Geological applications. *In* : Terra Nostra, (Proceedings of IAMG), volume 04. - 25–30. 27
- LEDEZ, D. (2003). - Méthodes implicites : applications aux géosciences (titre provisoire). Ph.D. thesis. - INPL, Nancy, France. 27
- LÉGER, M., THIBAUT, M., GRATIER, J.-P., & MORVAN, J.-M. (1997). - A least-squares method for multisurface unfolding. - *J. Struct. Geol.*, **19**(5), 735–743. 86
- LEPAGE, F. (2003). - Générations de maillages de simplexes (titre provisoire). Ph.D. thesis. - INPL, Nancy, France. 89, 113, 125
- LERCHE, I., DROMGOOLE, E., KENDALL, C. G., WALTER, L. M., & SCATURO, D. (1987). - Geometry of carbonate bodies : A quantitative investigation of factors influencing their evolution. - *Carbonates and Evaporites*, **2**, 15–42. 20
- LEVOY, M. (July 1990). - Efficient ray tracing of volume data. - *ACM Trans. on Graphics*, **9**(3), 245–261. 93
- LÉVY, B. (1999). - Topologie Algorithmique, Combinatoire et Plongement. Ph.D. thesis. - INPL, Nancy, France. 27, 43, 54, 55, 106, 130
- LÉVY, B. (2001). - Constrained texture mapping for polygonal meshes. *In* : Computer Graphics (Proc. Siggraph). - 417–424. ACM. 69
- LÉVY, B. & MALLET, J.-L. (1998). - Non-distorted texture mapping for sheared triangulated meshes. *In* : Computer Graphics (Proc. Siggraph). - ACM. 69
- LÉVY, B., CAUMON, G., CONREAU, S., & CAVIN, X. (2001). - Circular incident edge lists : a data structure for rendering complex unstructured grids. *In* : Proc. IEEE Visualization. - 191–198. 95, 96, 101, 105, 111
- LÉVY, B., PETITJEAN, S., RAY, N., & MAILLOT, J. (2002). - Least square conformal maps for automatic texture generation. - *ACM Transactions on Graphics (Proc. Siggraph)*, **21**(3), 362–371. 69, 79, 86

- LIENHARDT, P. (1989). - Subdivisions of n-dimensional spaces and n-dimensional generalized maps. *In* : Proc. Annual ACM Symposium on Computational Geometry. - 228-236. 106
- LIENHARDT, P. (1993). - N-dimensional generalized combinatorial maps and cellular quasi-manifold. Rapport de recherche 93/04, Université Louis Pasteur, Département Informatique. 79
- LIN, M. & GOTTSCHALK, S. (1998). - Collision detection between geometric models : A survey. *In* : Proc. IMA Conference on Mathematics of Surfaces. - . 73, 133
- LIVNAT, Y., SHEN, H.-W., & JOHNSON, C. R. (March 1996). - A near optimal isosurface extraction algorithm using the span space. - *IEEE Trans. Visual Comput. Graphics*, **2**(1), 73-84. 96
- LORENSEN, W. E. & CLINE, H. E. (1987). - Marching cubes : a high resolution 3d surface construction algorithm. - *Computer Graphics (Proc. Siggraph)*, **21**(4), 163-170. 94, 101, 117, 118
- LUNDELL, A. T. & WEINGRAM, S. (1969). - The topology of CW Complexes. - Van Nostrand Reinhold Company. 106
- LYELL, C. (1830-1833). - Principles of geology. - John Murray, London. <http://www.esp.org/books/lyell/principles/facsimile/>. 15
- MA, K.-L. & CROCKETT, T. W. (November 1997). - A scalable parallel cell-projection volume rendering algorithm for three-dimensional unstructured data. *In* : PAINTER J., STOLL G., & MA K.-L., Eds, *Proc. IEEE Parallel Rendering Symposium*. - IEEE, 95-104. 95
- MALLET, J.-L. (1989). - Discrete smooth interpolation in geometric modeling. - *ACM Transactions on Graphics*, **8**(2), 121-144. 27, 28
- MALLET, J.-L. (1992). - Discrete smooth interpolation. - *Computer-Aided Design*, **24**(4), 263-270. 27
- MALLET, J.-L. (2002). - Geomodeling. Applied Geostatistics. - Oxford University Press. 624 pp. 13, 16, 23, 24, 25, 27, 28, 38, 39, 43, 63, 73, 106, 131
- MÄNTYLÄ, M. (1988). - An Introduction to Solid Modeling. - Computer Science Press. 401 pp. 12, 45, 46, 51, 54, 107
- MARTHA, L. F., DE CARVALHO, M. T., & DE BEAUCLAIRE SEIXAS, R. (1997). - Volume contouring of generic unstructured meshes. - *Journal of the Brazilian Computer Society*, **3**(3), 43-51. 92
- MASSOT, J. (2002). - Implémentation de méthodes de restauration équilibrée 3D. Ph.D. thesis. - INPL, Nancy, France. 69, 70, 86
- MELLO, U. T. & HENDERSON, M. E. (1997). - Techniques for including large deformations associated with salt and fault motion in basin modeling. - *Marine and Petroleum Geology*, **14**(5), 551-564. 49, 76

- MIALL, A. D. (1997). - The Geology of Stratigraphic Sequences. - Springer-Verlag. 433 pp. 17, 18, 44
- MÖLLER, T. & HAINES, E. (1999). - Real-Time Rendering. - A.K. Peters Ltd. 482 pp. 12, 69
- MÖLLER, T., MACHIRAJU, R., MUELLER, K., & YAGEL, R. (1997). - A comparison of normal estimation schemes. *In* : IEEE Visualization. - 19-26. 104, 129
- MONTAGNAT, J., DELINGETTE, H., & AYACHE, N. (2001). - A review of deformable surfaces : topology, geometry and deformation. - *Image and Vision Computing*, **19**, 1023-1040. 62, 63
- MOORE, M. & WILHELMS, J. (1988). - Collision detection and response for computer animation. *In* : Computer Graphics (Proc. Siggraph), **22**(4) :289-298. - . 25
- MULLER, D. & PREPARATA, F. (1978). - Finding the intersection of two convex polyhedra. - *Theoretical Computer Science*, **7**, 217-236. 56, 107
- NEIDER, J., DAVIS, T., & WOO, M. (1993). - OpenGL Programming Guide. - Addison-Wesley. 33, 104, 110
- NEUMANN, L., CSÉBFALVI, B., KÖNIG, A., & GRÖLLER, E. (2000). - Gradient estimation in volume data using 4D linear regression. - *Computer Graphics Forum (Eurographics 2000)*, **19**(3), 351-358. 104, 129
- NIELSON, G. M. & OLSEN, D. R. (1987). - Direct manipulation techniques for 3d objects using 2d locator devices. *In* : CROW F. & PIZER S. M., Eds, *Proceedings of the 1986 workshop on Interactive 3D graphics*. - . 37
- NORDLUND, U. (1996). - Formalizing geological knowledge with an example of stratigraphic modeling using fuzzy logic. - *Journal of Sedimentary Research*, **66**(4), 689-698. 20
- NORDLUND, U. (1999). - FUZZIM : forward stratigraphic modeling made simple. - *Computers and Geosciences*, **25**(4), 449-456. 20
- OWEN, S. (1998). - A survey of unstructured mesh generation technology. *In* : Proc. 7th International Meshing RoundTable. - . 89
- PARKER, S., PARKER, M., LIVNAT, Y., SLOAN, P.-P., HANSEN, C., & SHIRLEY, P. (1999). - Interactive ray tracing for volume visualization. - *IEEE Trans. Visual. Comput. Graphics*, **5**(3), 238-250. 93
- PFISTER, H., HARDENBERGH, J., KNITTEL, J., LAUER, H., & SEILER, L. (1999). - The volumepro real-time ray-casting system. *In* : Computer Graphics (Proc. Siggraph). - 251-260. 96
- PIEGL, L. A. & TILLER, W. (1997). - The NURBS Book. Monographs in Visual Communications. - Springer Verlag, second edition. 650 pp. 40, 72
- PREPARATA, F. & SHAMOS, M. (1985). - Computational Geometry : An Intro-

- duction. - Springer-Verlag. 56, 107
- R. A. DREBIN, L. CARPENTER, P. H. (1988). - Volume rendering. - *Computer Graphics*, **22**(4), 65–74. 92
- RAGHOTHAMA, S. & SHAPIRO, V. (1998). - Boundary representations deformation in parametric solid modeling. - *ACM Transactions on Graphics*, **17**(4), 259–286. 58
- RAPER, J., Ed (1989). - Three Dimensional Applications in Geographical Informations Systems. - Taylor and Francis. 189 pp. 48
- RÉMY, N., SHTUKA, A., LÉVY, B., & CAERS, J. (2002). - GsTL : The geostatistical template library in c++. - *Computer and Geosciences*, **28**(8), 971–979. 88, 130
- REQUICHA, A. A. G. (1980). - Representation of rigid solids : Theory, methods and systems. - *ACM Computing Surveys*, **12**(4), 437–464. 51
- REQUICHA, A. A. G. & VOELCKER, H. B. (1980). - Constructive solid geometry. *In* : Production Automation Project Tech. Memo 25, Rochester University. - . 46
- REY, J., Ed (1997). - Stratigraphie, Terminologie Française, volume 19 of *Mémoires, Bulletin Centre de Recherche Exploration-Production*. - Elf Aquitaine. 15, 16, 24
- REYNOLDS, D. J., STECKLER, M. S., & COAKLEY, B. J. (1991). - The role of the sediment load in sequence stratigraphy : the influence of flexural isostasy and compaction. - *Journal of Geophysical Research*, **96B**, 6931–6949. 19, 20
- REZK SALAMA, C., ENGEL, K., BAUER, M., GREINER, G., & ERTL, T. (2000). - Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. *In* : Siggraph/Eurographics Workshop on Graphics Hardware. - 109–119. 95
- ROSSIGNAC, J. R. & O'CONNOR, M. A. (1990). - Sgc : A dimensional-independent model for pointsets with internal structures and incomplete boundaries. *In* : WOZNY M. J., TURNER J. U., & PREIS K., Eds, *Geometric Modelling for Product Engineering*, 145–180. Elsevier Science Publishers. 54
- RÖTTGER, S., KRAUS, M., & ERTL, T. (2000). - Hardware-accelerated volume and isosurface rendering based on cell-projection. *In* : IEEE Visualization. - . 94, 95
- SABELLA, P. (1988). - A rendering algorithm for visualizing 3d scalar fields. - *Computer Graphics*, **22**(4), 51–58. 92, 93
- SAKKALIS, T., SHEN, G., & PATRIKALAKIS, N. M. (2000). - Representational validity of boundary representation models. - *Computer Aided Design*, **32**(12), 719–726. 51
- SAMET, H. (1990). - Applications of Spatial Data Structures : Computer Gra-

- phics, Image Processing and GIS. Computer Science. - Addison Wesley. 507 pp. 46
- SAMSON, P. (1996). - Équilibrage de structures géologiques 3D dans le cadre du projet Gocad. Ph.D. thesis. - INPL, Nancy, France. 86
- SEDERBERG, T. W. & PARRY, S. R. (1986). - Free-form deformation of solid geometric models. *In* : EVANS D. C. & ATHAY R. J., Eds, *Computer Graphics (SIGGRAPH '86 Proceedings)*, volume 20. - 151–160. 61
- SHEWCHUK, J. (1996). - Robust adaptative floating-point geometric predicates. *In* : Symposium on Computational Geometry. - 141–150. ACM. 81
- SHIRLEY, P. & TUCHMAN, A. (November 1990). - A polygonal approximation to direct scalar volume rendering. - *Computer Graphics (San Diego Workshop on Volume Visualization)*, 24(5), 63–70. 94, 95
- SILVA, C. T. & MITCHELL, J. S. B. (April - June 1997). - The lazy sweep ray casting algorithm for rendering irregular grids. - *IEEE Trans. Visual. Comput. Graphics*, 3(2). 93, 96, 97, 98, 99, 100, 109, 113, 129
- SILVA, C. T., MITCHELL, J. S. B., & WILLIAMS, P. L. (October 1998). - An exact interactive time visibility ordering algorithm for polyhedral cell complexes. *In* : Proc. Volume Visualization Symposium. - IEEE, 87–94. 94
- SLOSS, L. L. (1962). - Cf page 26 of emery96. - *Tutu*, 0, 0–0. 19
- SLOSS, L. L. (1988). - Fourty years of sequence stratigraphy. - *Geological Society of America Bulletin*, 100, 1661–1665. 18
- SOUCHE, L. (2002). - Integrating complex fault network in horizon and reservoir modeling : a 3d parameterized space-based approach. *In* : Terra Nostra, (Proceedings of IAMG), volume 04. - 47–52. 30
- SOUCHE, L. (2003). - Modélisation de failles (titre provisoire). Ph.D. thesis. - INPL, Nancy, France. 26, 30, 44
- STEIN, C., BECKER, B., & MAX, N. (1994). - Sorting and hardware assisted rendering for volume visualization. *In* : KAUFMAN A. & KRUEGER W., Eds, *Proc. Symp. on Volume Visualization*. - 83–90. 94, 105
- STENO, N. (1669). - De solido intra solidium naturaliter contento dissertationis prodromus. Florence. 15
- STOLLNITZ, E. J., DEROSE, T. D., & SALESIN, D. H. (1995). - Wavelets for computer graphics : A primer, part 2. - *IEEE Computer Graphics and Applications*, 15(4), 75–85. <http://grail.cs.washington.edu/projects/wavelets/article/>. 73
- STREBELLE, S. (2002). - Conditional simulation of complex geological structures using multiple-point statistics. - *Mathematical Geology*, 34(1). 44
- SUKUMAR, N., MORAN, B., SEMENOV, A. Y., & BELIKOV, V. V. (2001). - Natural neighbor galerkin methods. - *Intl. J. for Numerical Methods in Engi-*

- neering*, **50**, 1–27. 89
- SUTHERLAND, I. E. (1963). - Sketchpad : a man-machine graphical communication system. Ph.D. thesis. - Massachusetts Institute of Technology. Technical Report 296, Lincoln Laboratory. 37
- SWORD, C. H. (1991). - Building flexible interactive, geologic models. *In* : 61st Annual International Meeting. - Society of Exploration Geophysicists, 1465–1467. 49, 50
- TARANTOLA, A. (1987). - Inverse Problem Theory. - Elsevier. 630 pp. 21, 50
- TETZLAFF, D. M. & HARBAUGH, J. W. (1989). - Simulating Clastic Sedimentation. - Van Nostrand Reinhold. 202 pp. 20
- TURCOTTE, D. L. & WILLEMANN, R. J. (1983). - Synthetic cyclic stratigraphy. - *Earth and Planetary Science Letters*, **63**, 89–96. 19
- TURNER, A. K., Ed (1992). - Three-Dimensional Modeling with Geoscientific Information Systems, volume 354 of *NATO-ASI, Math. and Phys. Sciences*. - Kluwer Academic Publishers. 443 pp. 48, 49
- TUY, H. K. & TUY, L. T. (1984). - Direct 2-d display of 3-d objects. - *IEEE Computer Graphics and Applications*, **4**(10), 29–33. 93
- UPSON, C. & KEELER, M. (August 1988). - Vbuffer : Visible volume rendering. - *Computer Graphics (Proc. SIGGRAPH)*, **22**(4), 59–64. 92, 94
- VAIL, P. R. (1987). - Seismic stratigraphy interpretation using sequence stratigraphy, part 1 : seismic stratigraphy interpretation procedure. *In* : BALLY A. W., Ed, *Atlas of Seismic Stratigraphy*, volume 27, chapter 1, 1–10. AAPG Studies in Geology. 18
- VAIL, P. R., MITCHUM, R. M., TODD, R. G., WINDMIER, J. M., THOMPSON, S., SANGREE, J. B., BUBB, J. N., & HATLEDID, W. G. (1977). - Seismic stratigraphy and global changes of sea level. *In* : PAYTON C. E., Ed, *Seismic Stratigraphy – Applications to hydrocarbon exploration*, volume 26, 49–212. AAPG Memoir. 17, 18, 27
- VAN GELDER, A. & WILHELMS, J. (October 1994). - Topological considerations in isosurface generation. - *ACM Trans. on Graphics*, **13**(4), 337–375. 101, 103, 104
- VERMA, S. (1996). - Flexible grids for reservoir simulation. Ph.D. thesis. - Stanford University, School of Earth Sciences. 89
- VICKERY, R., KEEN, T. R., MOORHEAD, R. J., BROU, R. J., CARRUTH, D. W., & M.DOANE, S. (2002). - Volume visualization of 5d sedimentation models. *In* : ERBACHER R. F., CHEN P. C., GROEHN M., ROBERTS J. C., & WITTENBRINK C. M., Eds, *Electronic Imaging Science and Technology Visualization and Data Analysis 2002*. - Proc. SPIE 4665, 12 pages. 129
- VISEUR, S. (2001). - Simulation stochastique basée-objet de dépôts fluviaux.

- Ph.D. thesis. - INPL, Nancy, France. 32
- WAGONER, J. C. V., MITCHUM, R. M., POSAMENTIER, H. W., & VAIL, P. R. (1987). - Seismic stratigraphy interpretation using sequence stratigraphy, part 2 : key definitions of sequence stratigraphy. *In* : BALLY A. W., Ed, *Atlas of Seismic Stratigraphy*, volume 27, chapter 1, 11–14. AAPG Studies in Geology. 18, 26
- WATNEY, W. L., RANKEY, E. C., & HARBAUGH, J. (1999). - Perspectives on stratigraphic simulation models : current approaches and future opportunities. *In* : HARBAUGH J., WATNEY W., RANKEY E., SLINGERLAND R., GOLDSTEIN R., & FRANSEEN E., Eds, *Numerical Experiments in Stratigraphy : Recent Advances in Stratigraphic and Sedimentologic Computer Simulations*, volume 62, 3–21. SEPM Special Publication. 17, 21
- WATSON, D. F. (1981). - Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. - *Computer Journal*, **24**(2), 162–166. 76
- WEILER, K. (1986). - The Radial Edge structure : a topological representation for non-manifold geometric boundary modeling. *In* : Proc. of the IFIG WG 5.2. - . 54, 55, 56, 57
- WERNECKE, J. (1994). - The Inventor Mentor : programming Object-oriented 3D graphics with Open Inventor, Release 2. - Addison Welsey. ISBN 0-201-62495-8. 37, 69
- WESTERMANN, R. (2001). - The rendering of unstructured grids revisited. *In* : Eurographics IEEE TCVG Symposium on Visualization 2001. - . 95
- WESTERMANN, R. & ERTL, T. (July 1998). - Efficiently using graphics hardware in volume rendering applications. - *Computer Graphics (Proc. SIGGRAPH)*, 169–178. 95, 96
- WESTERMANN, R. & ERTL, T. (November 1997). - The vsbuffer : Visibility ordering of unstructured volume primitives by polygon drawing. - *IEEE Visualization '97*, 35–42. 95, 96
- WESTERMANN, R., KOBELT, L., & ERTL, T. (1999). - Real-time exploration of regular volume data by adaptive reconstruction of iso-surfaces. - *Visual Computer*, **15**(2). 96
- WIGGINS, W., ALBERTIN, U. K., & STANKOVIC, G. (1993). - Building 3-d depth-migration velocity models with topological objects. *In* : Proc. 63rd Annual International Meeting. - Soc. of Expl. Geophys., 170–173. 50
- WILHELMS, J. & GELDER, A. V. (July 1992). - Octrees for faster isosurface generation. - *ACM Trans. on Graphics*, **11**(3), 201–227. 96
- WILHELMS, J., CHALLINGER, J., ALPER, N., & RAMAMOORTHY, S. (1990). - Direct volume rendering of curvilinear volumes. - *Computer Graphics (San Diego Workshop on Volume Visualization)*, **24**(5), 41–48. 93

- WILHELMS, J. P., GELDER, A. V., TARANTINO, P., & GIBBS, J. (October 1996). - Hierarchical and parallelizable direct volume rendering for irregular and multiple grids. - *IEEE Visualization '96*, 57-64. 96
- WILLIAMS, P. L. (April 1992). - Visibility-ordering meshed polyhedra. - *ACM Trans. on Graphics*, **11**(2), 103-126. 94, 116, 120, 121, 123
- WITTENBRINK, C. M. (1999). - Cellfast : Interactive unstructured volume rendering. *In* : IEEE Conference on Visualization - Late Breaking Hot Topics. - . 95
- WYATT, K. D., TOWE, S. K., LAYTON, J. E., WYATT, S. B., SEGGERN, D. H. V., & BROCKMEIER, C. A. (1992). - Ergonomics in 3-d depth-migration. *In* : 62nd Annual International Meeting. - Soc. of Expl. Geophys., 944-947. 60
- YAGEL, R., REED, D. M., LAW, A., SHIH, P.-W., & SHAREEF, N. (October 1996). - Hardware assisted volume rendering of unstructured grids by incremental slicing. *In* : Proc. Volume Visualization Symposium. - IEEE, 55-62. 95, 96, 98, 101, 105, 127, 128, 129
- ZELEZNICK, R. C., HERNDON, K. P., ROBBINS, D. C., HUANG, N., MEYER, T., PARKER, N., & JOHN F, H. (1993). - An interactive 3d toolkit for constructing 3d widgets. - *Computer Graphics (Proc. Siggraph)*, **27**, 81-84. 37
- ZHAI, S. (1998). - User performance in relation to 3d input device design. - *Computer Graphics*, **32**(4), 50-54. <http://www.almaden.ibm.com/u/zhai/papers/siggraph/final.html>. 36
- ZORIN, D., SCHRÖDER, P., & SWELDENS, W. (1997). - Interactive Multiresolution Mesh Editing. - *Computer Graphics (Proc. Siggraph)*, **31**, 259-268. 59

**AUTORISATION DE SOUTENANCE DE THESE  
DU DOCTORAT DE L'INSTITUT NATIONAL  
POLYTECHNIQUE DE LORRAINE**

o0o

VU LES RAPPORTS ETABLIS PAR :

**Monsieur André JOURNEL , Professeur, PED, Stanford University**

**Monsieur Albert TARANTOLA, Professeur, DDS, Institut de Physique du Globe de Paris**

Le Président de l'Institut National Polytechnique de Lorraine, autorise :

**Monsieur CAUMON Guillaume**

à soutenir devant un jury de l'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE,  
une thèse intitulée :

**"Représentation, visualisation et modification de modèles volumiques pour les  
Géosciences »**

NANCY BRABOIS  
2, AVENUE DE LA  
FORET-DE-HAYE  
BOITE POSTALE 3  
F - 5 4 5 0 1  
VANCEUVRE CEDEX

en vue de l'obtention du titre de :

**DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE LORRAINE**

Spécialité : « **Géosciences** »

Fait à Vandoeuvre, le 10 février 2003

Le Président de l'I.N.P.L.,

L. SCHUFFENECKER





## Résumé

La *modélisation volumique* du sous-sol a une grande importance applicative dans de nombreux domaines des Géosciences. Cette thèse aborde cette discipline sous trois aspects fondamentaux.

Premièrement, la *représentation informatique* des modèles volumiques détermine les opérations et les types formations géologiques réalisables. Nous étudions ainsi une représentation par extrusion adaptée à la modélisation d'architectures stratigraphiques d'une part, et deux approches plus générales d'autre part :

- une représentation continue du sous-sol à base de surfaces (représentation par frontières), équivalente à une carte géologique tridimensionnelle ;
- une représentation cellulaire (grille), collection de cellules polyédriques générée à partir de ces macro-modèles pour effectuer des calculs d'ingénierie.

Le deuxième aspect concerne la *visualisation* de ces modèles, indispensable à une bonne compréhension du sous-sol. Pour cela, de nouveaux algorithmes sont présentés pour extraire efficacement des coupes, des surfaces d'iso-valeur ou effectuer du rendu volumique.

Enfin, des méthodes générales de *modification* géométrique sont définies. Réalisées dans le cadre de manipulations interactives, elles mettent l'accent sur le maintien de la cohérence géologique du modèle tridimensionnel.

---

## Representing, visualizing and modifying geoscientific solid models

### Abstract

Subsurface volume modeling has many applications in Geosciences. This thesis is concerned with three of the main challenges of volume modeling.

First, *computer representation* of solid models determines how much complex geometries can be accounted for, and how appropriately operations can be carried out. Therefore, an extrusion-based representation for modeling stratigraphic architecture is studied, as well as two more general approaches:

- A boundary representation, which appears as a 3D geological map, is made up of continuous spatial regions separated by surfaces.
- A cellular representation, or grid, is a collection of polyhedral cells generated from these macro-models for engineering computations.

Second, *Visualization* of volume models has to generate informative images for helping the understanding of the subsurface. New algorithms are presented for efficiently extracting cross-sections and isovalue surfaces or direct volume rendering.

Third, general geometrical *Modification* are defined. Implemented as interactive manipulation techniques, their primary concern is to maintain geological consistency through the use of constraints.

---

### Mots-clés

Modélisation Volumique, Géomodélisation, Architecture Stratigraphique, B-rep, Grille, Interaction, Déformation, Visualisation Volumique.