

Laboratoire des sciences de l'ingénieur,  
de l'informatique et de l'imagerie

---

École Doctorale Mathématiques,  
Sciences de l'Information et de l'Ingénieur

**THÈSE**

présentée pour obtenir le grade de  
**Docteur de l'Université de Strasbourg**  
**Mention : Informatique**

par  
**Lionel Untereiner**

**Représentation des maillages multirésolutions :  
application aux volumes de subdivision**

Soutenue publiquement le **8 novembre 2013**

**Membres du jury :**

**M. Bruno LÉVY** ..... *Rapporteur*  
Directeur de recherche à INRIA Nancy Grand-Est

**M. Guillaume DAMIAND** ..... *Rapporteur*  
Chargé de recherche HDR du CNRS au LIRIS à Lyon

**M. Georges-Pierre BONNEAU** ..... *Examinateur*  
Professeur à l'Université de Grenoble

**Mme Dominique BECHMANN** ..... *Directrice de thèse*  
Professeure à l'Université de Strasbourg

**M. David CAZIER** ..... *Directeur de thèse*  
Maître de conférences HDR à l'Université de Strasbourg



# REMERCIEMENTS

---

J'aimerais tout d'abord remercier Dominique Bechmann et David Cazier, d'avoir accepté de m'encadrer durant mon stage recherche en Master 2 (les brins n'étaient à l'époque pas encore mon « truc »), puis d'avoir renouvelé leur confiance pour ce travail de recherche doctoral. Je leur suis reconnaissant de la liberté de travail qu'ils m'ont laissé durant ces trois années.

Je remercie également MM. Bruno Lévy, directeur de recherche à INRIA Nancy Grand-Est et Guillaume Damiand, chargé de recherche du CNRS au LIRIS à Lyon pour avoir accepté de rapporter ce mémoire. Les remarques et échanges que nous avons pu avoir, ont permis d'en améliorer la qualité. Merci également à M. Georges-Pierre Bonneau, Professeur à l'Université de Grenoble, d'avoir accepté de participer au jury et de l'avoir présidé lors de la soutenance.

Ces quelques années passées dans le couloir IGG n'auraient pas été les mêmes sans les différentes personnes, parties et/ou arrivées, qui le composent. Ils ont rendu ces journées de travail (voir certaines soirées) bien plus amusantes grâce à un subtil mélange de discussions sérieuses et d'autres qui l'étaient un peu moins... Je ne citerai pas de noms afin de n'oublier personne, mais je suis sûr qu'ils se reconnaîtront.

Ces années ont également été pour moi l'occasion de découvrir l'enseignement par l'intermédiaire d'une mission enseignement au sein du département

informatique de l'UFR Mathématique-Informatique à l'Université de Strasbourg. J'aimerais remercier toutes les personnes avec qui j'ai eu le plaisir d'enseigner, et notamment Basile, qui a joué le rôle du guide lorsque j'ai débuté.

J'aimerais également remercier Noëlle, Yves, Antoine, Nathalie ainsi que tous les membres du bureau de l'École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur. Les activités et échanges que nous avons eu dans ce cadre ont été forts intéressants et m'ont permis de m'ouvrir à d'autres horizons que celui d'un travail de recherche.

J'aimerais également profiter de cette espace pour adresser un Merci particulier à Kenneth. Présent à mes côtés depuis le tout début de cette aventure universitaire, à m'écouter, et à supporter mes humeurs... Merci de ta patience !

Merci à Martine et Padelis de m'avoir accueilli, et ce, notamment pendant ces périodes délicates que sont la rédaction et la préparation de la soutenance. Grâce à vous, ce manuscrit respire un peu de cette petite brise marine particulière à La Brée-les-Bains.

Merci à Az et à Al. Sans vous, je ne serais sûrement pas arrivé jusque là.

Cette aventure n'aurait pas été possible sans le soutien de mes parents et de ma sœur. Je vous remercie de tout cœur pour votre patience et vos encouragements.

Enfin, j'aimerais remercier Mélina qui partage ma vie, mes doutes, et mes joies. Tu as été là pour moi à chaque étape de ce voyage. Lorsque ma motivation était chancelante, tu as su insuffler un vent d'énergie m'aidant à avancer.

# Table des matières

|  |           |
|--|-----------|
| <b>Introduction</b>  | <b>1</b>  |
| 1 Contexte . . . . .   | 1         |
| 2 Contributions . . . . .  | 2         |
| 2.1 Volumes de subdivision multirésolutions . . . . .            | 3         |
| 2.2 Représentation des maillages . . . . .                       | 4         |
| 3 Plan du mémoire . . . . .                                      | 5         |
| <br>   |           |
| <b>I Modèle topologique multirésolution</b>                      | <b>7</b>  |
| <br>   |           |
| <b>1 Modèles de représentation multirésolution</b>               | <b>9</b>  |
| 1.1 Introduction . . . . .                                       | 10        |
| 1.2 Structures topologiques de base . . . . .                    | 10        |
| 1.2.1 Structures ad hoc . . . . .                                | 11        |
| 1.2.2 Modèles structurés . . . . .                               | 15        |
| 1.3 Structures multirésolutions . . . . .                        | 28        |
| 1.3.1 Généralités . . . . .                                      | 28        |
| 1.3.2 Maillages progressifs tétraédriques . . . . .              | 29        |
| 1.3.3 Octree . . . . .   | 30        |
| 1.3.4 Hiérarchie par Bissection Régulière de Simplexes . . . . . | 33        |
| 1.3.5 Hiérarchie de Tétraèdres/Octaèdres . . . . .               | 36        |
| 1.3.6 Multi-Tessellation . . . . .                               | 38        |
| 1.3.7 Pyramides de cartes . . . . .                              | 39        |
| 1.4 Conclusion . . . . .   | 40        |
| <br>   |           |
| <b>2 Cartes combinatoires multirésolutions</b>                   | <b>43</b> |
| 2.1 Introduction . . . . .                                       | 43        |
| 2.2 Extension multirésolution en dimension quelconque . . . . .  | 44        |
| 2.2.1 Ensembles de brins . . . . .                               | 44        |
| 2.2.2 Relations indexées . . . . .                               | 45        |
| 2.2.3 Hypercartes multirésolutions . . . . .                     | 47        |
| 2.2.4 Cartes combinatoires multirésolutions . . . . .            | 48        |
| 2.3 Opérations de manipulation . . . . .                         | 50        |
| 2.3.1 Opérations élémentaires . . . . .                          | 51        |

|           |   |            |
|-----------|---|------------|
| 2.3.2     | Opérations de bas niveau . . . . .                                  | 53         |
| <b>II</b> | <b>Contributions</b>  | <b>61</b>  |
| <b>3</b>  | <b>Volumes de subdivision multirésolutions</b>                      | <b>63</b>  |
| 3.1       | Introduction . . . . .  | 64         |
| 3.1.1     | Volumes de subdivision . . . . .                                    | 64         |
| 3.1.2     | Applications des volumes de subdivision . . . . .                   | 67         |
| 3.1.3     | Volumes de subdivision multirésolutions . . . . .                   | 69         |
| 3.1.4     | Support du maillage multirésolution . . . . .                       | 71         |
| 3.2       | Génération des niveaux par raffinement primal . . . . .             | 72         |
| 3.2.1     | Génération régulière . . . . .                                      | 74         |
| 3.2.2     | Subdivision d'un polyèdre arbitraire . . . . .                      | 83         |
| 3.2.3     | Génération adaptative . . . . .                                     | 87         |
| 3.2.4     | Simplification . . . . .  | 92         |
| 3.2.5     | Niveau de subdivision d'une cellule . . . . .                       | 94         |
| 3.3       | Mise à jour des attributs de sommet : cas de la géométrie . . . . . | 96         |
| 3.3.1     | Schéma de subdivision . . . . .                                     | 97         |
| 3.3.2     | Schéma multirésolution . . . . .                                    | 102        |
| 3.4       | Génération adaptative de maillage pour la simulation . . . . .      | 106        |
| 3.5       | Autre raffinement : $\sqrt{3}$ volumique . . . . .                  | 109        |
| 3.5.1     | Génération régulière . . . . .                                      | 111        |
| 3.5.2     | Génération adaptative . . . . .                                     | 117        |
| 3.5.3     | Géométrie . . . . .   | 119        |
| 3.6       | Conclusion . . . . .  | 122        |
| <b>4</b>  | <b>Représentations du modèle</b>                                    | <b>123</b> |
| 4.1       | Introduction . . . . .  | 123        |
| 4.1.1     | Mise en œuvre monorésolution . . . . .                              | 125        |
| 4.1.2     | Analyse comparative de la performance . . . . .                     | 126        |
| 4.2       | Représentation multirésolution explicite . . . . .                  | 129        |
| 4.2.1     | Principe . . . . .  | 129        |
| 4.2.2     | Complexité temporelle . . . . .                                     | 131        |
| 4.2.3     | Complexité spatiale . . . . .                                       | 131        |
| 4.2.4     | Mise en œuvre et analyse de la performance . . . . .                | 135        |
| 4.3       | Représentation multirésolution implicite . . . . .                  | 139        |
| 4.3.1     | Principe . . . . .  | 139        |
| 4.3.2     | Complexité temporelle . . . . .                                     | 148        |
| 4.3.3     | Complexité spatiale . . . . .                                       | 151        |
| 4.3.4     | Mise en œuvre et analyse de la performance . . . . .                | 154        |
| 4.4       | Conclusion . . . . .  | 161        |
|           | <b>Conclusion</b>   | <b>163</b> |

---

|     |  |            |
|-----|--|------------|
| 1   | Rappel des contributions . . . . .                 | 163        |
| 1.1 | Généricité des méthodes . . . . .                  | 164        |
| 1.2 | Préservation de la cohérence topologique . . . . . | 165        |
| 1.3 | Efficacité des méthodes . . . . .                  | 165        |
| 1.4 | Flexibilité des méthodes . . . . .                 | 166        |
| 2   | Perspectives . . . . .                             | 167        |
|     | <b>Bibliographie</b>                               | <b>173</b> |





# INTRODUCTION

---

## 1 Contexte

La représentation d'un objet continu (réel ou synthétique) avec l'outil informatique nécessite une étape de discrétisation afin de pouvoir le représenter de manière finie en machine. Les objets sont généralement approximés par un certain nombre d'échantillons. Plus il y a d'échantillons, mieux l'objet numérique approxime l'objet continu. Ces échantillons sont la donnée initiale d'un grand nombre d'applications en informatique graphique.

Parmi ces objets, les données volumiques sont très répandues en informatique graphique, en visualisation scientifique et en calcul numérique. Elles sont issues d'acquisitions de données réelles (imagerie médicale, scanner 3D) ou générées de manière à résoudre des problèmes d'analyse numérique ou d'optimisation. La manipulation de ces données s'appuie usuellement sur une discrétisation en cellules (sommets, arêtes, faces, volumes, etc.) sur lesquelles s'appuient les calculs numériques. Pour permettre des traitements informatiques efficaces, ces cellules sont munies de relations d'adjacences et d'incidences pour former un maillage. De tels maillages sont la clé de voûte de la modélisation, de la simulation et de la visualisation de phénomènes physiques.

Afin d'accélérer les traitements, les applications utilisant ces maillages font

parfois appel à des calculs adaptatifs nécessitant des opérations de subdivision, de simplification ou de remaillage. Tous ces procédés impliquent la gestion de masses de données et rendent l'objet modélisé (c'est-à-dire le maillage) qui en résulte volumineux et complexe.

Une solution pour maîtriser la complexité de l'objet et des traitements numériques qui lui sont appliqués, consiste à le représenter à différentes échelles ou niveaux de détails. On parle alors de représentation multirésolution. Pour ce faire, on n'exploite plus un maillage unique mais une hiérarchie de maillages, ou une séquence de maillages emboîtés les uns dans les autres. Cette hiérarchie peut être construite de deux façons. La première opère dans le sens « grossier vers fin » par une opération récursive subdivisant chaque élément du maillage en un sous-ensemble d'éléments. Cette représentation est dite adaptative si la finesse n'est pas la même pour toutes les zones de l'objet : fin dans une zone d'intérêt et grossier ailleurs. La seconde opère dans le sens « fin vers grossier » par une opération de simplification permettant de fusionner des éléments du maillage.

Parmi les structures stockant et manipulant des maillages, les modèles à base topologique sont réputés garantir la cohérence et la robustesse des informations d'adjacence entre cellules (arêtes, faces, volumes, etc). Par contre, ils posent problème par la taille mémoire nécessaire à leur encodage et le manque d'information hiérarchique. À l'inverse, les modèles correspondant aux subdivisions de l'espace, régulières ou non, maîtrisent leur coût mémoire, mais ne permettent pas de définir des opérateurs topologiques robustes et génériques.

Les travaux de recherche que nous allons présenter s'intéressent à l'approche « grossier vers fin ». Nous verrons que, dans ce contexte, les modèles existants sont conçus pour des approches spécifiques rendant leur utilisation limitée aux applications pour lesquelles ils ont été pensés. Nous proposons dans ce travail des algorithmes et des modèles que nous voulons génériques et efficaces afin de pallier à ces défauts. Pour cela, nous utilisons comme socle de notre travail le cadre formel des cartes combinatoires [47, 114].

## 2 Contributions

Une carte combinatoire de dimension  $n$  représente la subdivision en cellules (sommets, arêtes, faces, volumes, etc.) d'un espace de dimension  $n$  orientable et fermé. Ce modèle est lié à un élément abstrait nommé brin, et fait appel à des notions de topologie combinatoire [33]. L'extension du modèle des cartes combinatoires au cadre multirésolution permet, comme nous allons le montrer,

de représenter un large éventail de maillages pour lesquels chaque niveau de résolution modélise la topologie d'une variété de dimension  $n$ .

Les structures spécifiques mentionnées auparavant ne représentent généralement pas toutes les cellules d'un maillage, ni l'ensemble des relations d'incidences ou d'adjacences liant ces cellules. Il est donc difficile d'élaborer des algorithmes génériques convenant à tout type de polyèdres ou d'utiliser sur un même modèle plusieurs schémas de subdivision différents. Les modèles à base de cartes combinatoires, quant à eux, ont déjà démontré leurs capacités en modélisation géométrique [78, 79]. Ils ont d'ailleurs déjà fait preuve de leur efficacité dans le cadre multirésolution pour la manipulation de surfaces de subdivision multirésolutions [68].

Notre première contribution relève de la formalisation dans le cadre des cartes combinatoires d'une famille d'algorithmes permettant de construire et parcourir une hiérarchie de maillages en dimension quelconque. Notre seconde contribution s'intéresse aux hiérarchies de maillages volumiques construites à partir d'un processus de subdivision de volumes, avec comme application la réécriture des schémas de subdivision volumique pour des polyèdres quelconques. Notre troisième contribution concerne deux modes de représentation construits à partir du modèle formel pour de telles hiérarchies de maillages.

## 2.1 Volumes de subdivision multirésolutions

Notre objet d'étude principal a été les volumes de subdivision multirésolutions. Un volume de subdivision est défini comme la limite d'un processus de raffinement. La forme de l'objet raffiné est définie par un schéma de subdivision. Une étape de ce processus peut être séparée en deux parties. D'abord, la connectivité est modifiée par le raffinement de la topologie. Ensuite, la position des sommets du maillage raffiné est calculée grâce à différents masques de subdivision. Il existe différents schémas de raffinement agissant sur des maillages de connectivité différente, et parallèlement des masques différents générant des volumes aux propriétés géométriques différentes.

Dans ce cadre, la plupart des structures de données existantes sont spécifiques à certains types de polyèdres (tétraèdres, octaèdres ou hexaèdres) et donc à certains types de raffinements. De par sa généralité, le modèle des cartes combinatoires nous permet de formaliser l'algorithme de raffinement en fonction de la dimension des cellules. Ceci nous autorise à combiner dans un même modèle des raffinements de volume à base de quadrisection de face triangulaire [26] ou encore à base de quadrisection de face polygonale [86].

De plus, il est possible de mélanger différents types de raffinements comme nous le montrerons avec un algorithme de raffinement mixte conservant les faces triangulaires et quadrangulaires. Enfin, ce modèle supporte également des changements de configuration plus complexe comme c'est le cas pour le raffinement  $\sqrt{3}$  volumique [15] qui opère par des raffinements de tétraèdres puis des basculements de faces et d'arêtes entre tétraèdres. Ces algorithmes démontrent la généralité de notre approche car leur conception nécessite de savoir décrire la combinatoire des objets manipulés.

La subdivision adaptative permettant de raffiner un objet à des profondeurs différentes, en suivant un critère donné, est gérée de manière totalement transparente grâce aux qualités intrinsèques du modèle, comme illustré figure 1. Les structures spécifiques ne représentent généralement pas toutes les cellules du maillage. Il est alors difficile de raffiner de manière adaptative sans avoir de problème de conformité aux cellules frontières séparant deux niveaux de résolution. Toutes les cellules présentes dans ces zones frontières sont raffinées avec une carte combinatoire. Ceci permet de maintenir la cohérence entre les cellules voisines malgré la différence de niveau et l'apparition de polyèdres quelconques.

Le parcours des voisinages dans un maillage peut rapidement devenir fastidieux dans le cas volumique, qui plus est, pour des polyèdres quelconques. Le modèle des cartes combinatoires nous apporte la complétude des parcours de voisinage. Ceci nous permet d'envisager des algorithmes de calcul de la géométrie, efficaces et pouvant s'appliquer à des maillages volumiques arbitraires.

## 2.2 Représentation des maillages

La représentation d'une hiérarchie de maillages est le sujet d'étude connexe aux algorithmes de construction d'une telle hiérarchie. Nous avons étudié deux manières différentes de représenter une hiérarchie de maillage générée par un processus de subdivision. Ces deux modes de représentation se posent aux deux extrêmes en terme de complexité spatiale et temporelle.

La première représentation que l'on qualifiera d'explicite stocke, pour chaque niveau de la hiérarchie, l'ensemble des relations des brins tout en évitant les redondances entre les niveaux successifs. Chaque niveau représente donc directement une carte combinatoire. Cette représentation est très efficace en terme de complexité des traitements mais plus coûteuse en terme de stockage d'information. Cette approche permet de représenter une hiérarchie aussi bien générée dans un sens « grossier vers fin » que dans un sens « fin vers grossier ».

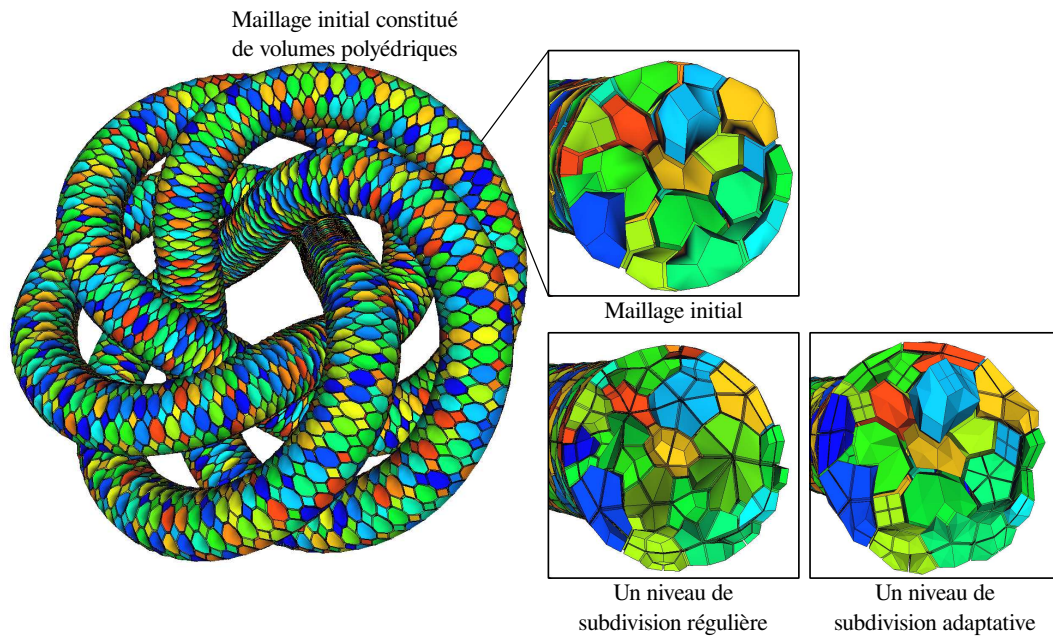


FIGURE 1 : Exemple de subdivision d'un maillage volumique polyédrique arbitraire utilisant une de nos méthodes. Les couleurs sont gardées d'un niveau de résolution au suivant afin d'identifier les volumes subdivisés. La subdivision adaptative consiste, dans cet exemple, à raffiner les volumes du bord du maillage.

À l'inverse, la seconde représentation, dite implicite ou compacte, ne stocke que le niveau le plus fin généré par un processus de subdivision primal. Grâce à la régularité du processus de subdivision primal, il est possible de réduire fortement la quantité d'information stockée. Nous proposons un mécanisme de reconstruction à la volée des informations non stockées. Ce mécanisme montre qu'il est ainsi possible de maintenir une hiérarchie implicite de cellules (sommets, arêtes, faces, volumes) à l'aide d'une simple structure de données mono-résolution. Cette représentation permet une économie d'occupation mémoire au détriment d'une augmentation des temps de calcul.

Les algorithmes et modèles de représentation présentés dans ces travaux ont été implantés au sein de la plate-forme logicielle CGoGN<sup>1</sup> développée dans l'équipe IGG.

### 3 Plan du mémoire

La manuscrit est organisé de la manière suivante :

---

1. <http://cgogn.unistra.fr>

Le premier chapitre présente les différents modèles de représentation multirésolution. Nous faisons d'abord un état de l'art des structures de données monorésolutions utilisées pour la représentation de maillages volumiques dans le cadre de la subdivision de volumes. Dans un second temps nous présentons des modèles plus généraux et notamment les modèles à base topologique dont sont issues les cartes combinatoires sur lesquelles nous reposons notre travail. Nous exposons ensuite les différents modèles et techniques de représentation et de manipulation de maillages multirésolutions.

Le second chapitre est consacré à l'extension multirésolution du modèle des cartes combinatoires en dimension quelconque. Nous utilisons la définition générale des hypercartes pour dériver de manière formelle la définition des cartes multirésolutions en dimension quelconque. Nous décrivons ensuite des opérations de manipulation de 3-cartes combinatoires dans le cas monorésolution. Puis, nous montrons comment étendre l'utilisation de ces opérations dans le cadre multirésolution.

Le troisième chapitre traite de l'application de notre modèle multirésolution à la représentation de volumes de subdivision multirésolutions. Nous proposons différents algorithmes de raffinement de volumes dans le cas régulier comme dans le cas adaptatif. Nous montrons ensuite l'utilisation de notre modèle dans le cadre de l'application de différents schémas de subdivision volumique. Nous proposons également une méthode de génération adaptative de maillage dans le cadre de la simulation électro-thermique de circuits intégrés.

Le quatrième chapitre aborde la représentation et l'implantation sous forme de structures de données concrètes des maillages multirésolutions. Nous proposons deux approches différentes à l'opposé l'une de l'autre. Nous analysons le coût théorique et les performances pratiques de ces deux représentations.

Enfin, le dernier chapitre conclut ce travail par un résumé de l'ensemble des apports et des limitations du modèle et des méthodes que nous proposons. Nous suggérons également diverses perspectives pour de futurs travaux. Ces idées se rapportent autant à des cadres applicatifs au sein desquels les cartes multirésolutions seraient un avantage qu'à des évolutions du modèle multirésolution en lui-même.

PREMIÈRE PARTIE

---

MODÈLE TOPOLOGIQUE  
MULTIRÉSOLUTION





---

# MODÈLES DE REPRÉSENTATION MULTIRÉSOLUTION

---

## Sommaire

---

|       |   |           |
|-------|---|-----------|
| 1.1   | Introduction . . . . .                                    | <b>10</b> |
| 1.2   | Structures topologiques de base . . . . .                 | <b>10</b> |
| 1.2.1 | Structures ad hoc . . . . .                               | 11        |
| 1.2.2 | Modèles structurés . . . . .                              | 15        |
| 1.3   | Structures multirésolutions . . . . .                     | <b>28</b> |
| 1.3.1 | Généralités . . . . .                                     | 28        |
| 1.3.2 | Maillages progressifs tétraédriques . . . . .             | 29        |
| 1.3.3 | Octree . . . . .  | 30        |
| 1.3.4 | Hierarchie par Bisection Régulière de Simplexes . . . . . | 33        |
| 1.3.5 | Hierarchie de Tétraèdres/Octaèdres . . . . .              | 36        |
| 1.3.6 | Multi-Tessellation . . . . .                              | 38        |
| 1.3.7 | Pyramides de cartes . . . . .                             | 39        |
| 1.4   | Conclusion . . . . .                                      | <b>40</b> |

---

Ce chapitre débute par une courte introduction à la représentation par des modèles numériques d'objets en informatique graphique. Nous présentons ensuite un certain nombre de structures de données ad hoc utilisées dans le cadre volumique. Puis, prenant un peu de distance avec ces modèles spécifiques, nous présentons des modèles structurés classiques utilisés en informatique graphique. Nous terminons par le modèle des cartes combinatoires qui est le socle du modèle multirésolution que nous étendons au chapitre suivant. Pour finir, nous nous intéresserons aux modèles et structures de données pour la représentation de maillages multirésolutions volumiques et/ou indépendants de la dimension.

## 1.1 Introduction

L'informatique graphique s'intéresse à la modélisation et à l'affichage d'objets ainsi qu'à leur animation et leur manipulation par un utilisateur. L'objectif est la représentation d'objets continus définis par une infinité de points dans un espace donné. Ces objets doivent cependant être discrétisés pour être représentés par l'outil informatique. Il existe deux méthodes pour la représentation de tels objets discrets.

La première méthode utilise une connaissance a priori des fonctions permettant de construire un objet. Elle consiste à représenter un objet par des fonctions en stockant les coefficients de ces dernières. Par exemple une sphère plongée dans un espace à trois dimensions peut être représentée de manière discrète par la position de son centre et la valeur de son rayon, connaissant la fonction liant ces paramètres.

La seconde méthode consiste à discrétiser l'objet en un ensemble de points prenant leurs coordonnées dans un espace donné. Cette discrétisation est subdivisée en un ensemble de primitives ou cellules plus ou moins structurées. Ces cellules, de différentes dimensions, sont liées entre elles par des relations d'incidences et d'adjacences. Ce partitionnement forme ce que l'on appelle un maillage.

Ainsi, les structures de données pour les maillages sont d'une importance fondamentale dans différents domaines tels que la modélisation géométrique, la génération de maillages, l'analyse par éléments finis, le traitement numérique de maillages, la visualisation ou encore la géométrie algorithmique. Nous nous sommes donc intéressés à cette seconde méthode. Nous verrons qu'il existe d'ailleurs un certain nombre de modèles bien établis en dimension 2 grâce à la modélisation géométrique. Ceci n'est pas le cas en dimensions supérieures.

## 1.2 Structures topologiques de base

Les structures de données volumiques mises en œuvre sont généralement étroitement liées au cadre d'application. Les opérations manipulant ces structures sont spécifiques à un traitement donné. Il est ainsi impossible d'utiliser ces structures dans un logiciel permettant d'appliquer des traitements différents sans envisager la conversion entre modèles.

Pour illustrer notre propos, nous décrivons les structures de données ad

hoc utilisées dans le contexte particulier de la subdivision de volumes. Ces structures de données permettent de représenter un maillage unique construit à l'aide d'opérations de raffinement particulières que nous décrirons le cas échéant.

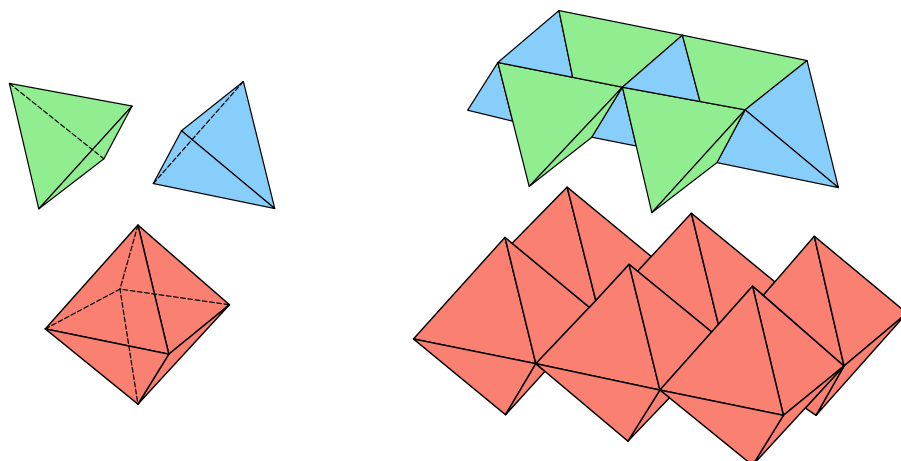
Nous poserons ensuite le cadre nécessaire à la définition d'un modèle plus général. Ce modèle nous permettra, par la suite, de décrire dans un cadre très général les raffinements appliqués aux cellules du maillage.

### 1.2.1 Structures ad hoc

#### Octet-truss

La structure en octet-truss [53] fait partie des structures de partitionnement de l'espace tridimensionnel. Elle est constituée de deux grilles complémentaires de tétraèdres et d'octaèdres assemblées de manière à remplir l'espace comme illustré par la figure 1.1. C'est une structure classiquement utilisée en mécanique [53], en électronique [76] ou encore en visualisation [66].

Dans cette structure, un octaèdre peut être découpé en quatre tétraèdres par une de ces diagonales. Cette structure est convertie puis implémentée sous la forme d'un maillage tétraédrique.



**FIGURE 1.1** : La structure octet-truss : des couches d'octaèdres (couleur saumon) agglomérées sous la forme de boîtes à œufs avec des tétraèdres (en vert et bleu) remplissant les trous. Les deux couleurs des tétraèdres permettent de distinguer les deux directions d'alignements.

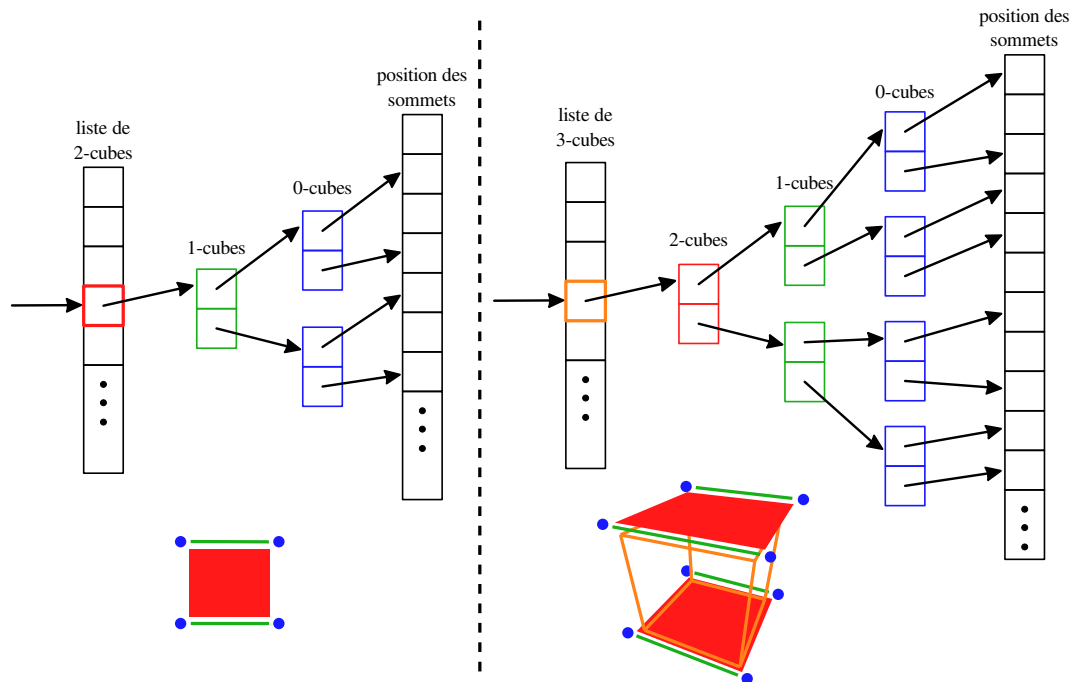


FIGURE 1.2 : Exemple d'un 2-cube et d'un 3-cube.

## Cubes en dimension quelconque

La seconde structure de données, issue de [4], supporte des maillages hypercubiques. Ainsi, un cube de dimension  $d$  (noté  $d$ -cube) est représenté de manière récursive où un 0-cube est un index de sommet et un  $d$ -cube (avec  $d > 0$ ) une liste de deux  $(d - 1)$ -cubes. Nous illustrons sur la figure 1.2 deux exemples : une face carrée (un 2-cube) est composée d'une liste de deux segments (deux 1-cubes), chaque 1-cube est composé d'une liste de deux sommets (deux 0-cubes) ; un hexaèdre (un 3-cube) est composé d'une liste de deux faces carrées (deux 2-cubes), etc.

Cette structure de données vient avec une opération de raffinement d'un  $d$ -cube  $C$  en  $2^d$  sous-cubes de dimension  $d$  également. Cette opération de création de nouveaux  $d$ -cubes est totalement liée à l'opération géométrique d'interpolation linéaire pour le calcul de la position des sommets. Cette formulation de l'opération de subdivision permet d'éviter l'emploi d'une structure de données hiérarchique devant gérer des positions de sommets à des niveaux différents.

Pour un  $d$ -cube  $C$ , l'algorithme commence par calculer de manière récursive la subdivision linéaire de deux  $(d - 1)$ -cubes compris dans  $C$ . Les deux listes résultantes de  $2^{d-1}(d - 1)$ -cubes sont appelées *top* et *bottom* respectivement. Ces deux listes *top* et *bottom* sont des subdivisions des faces top et bottom

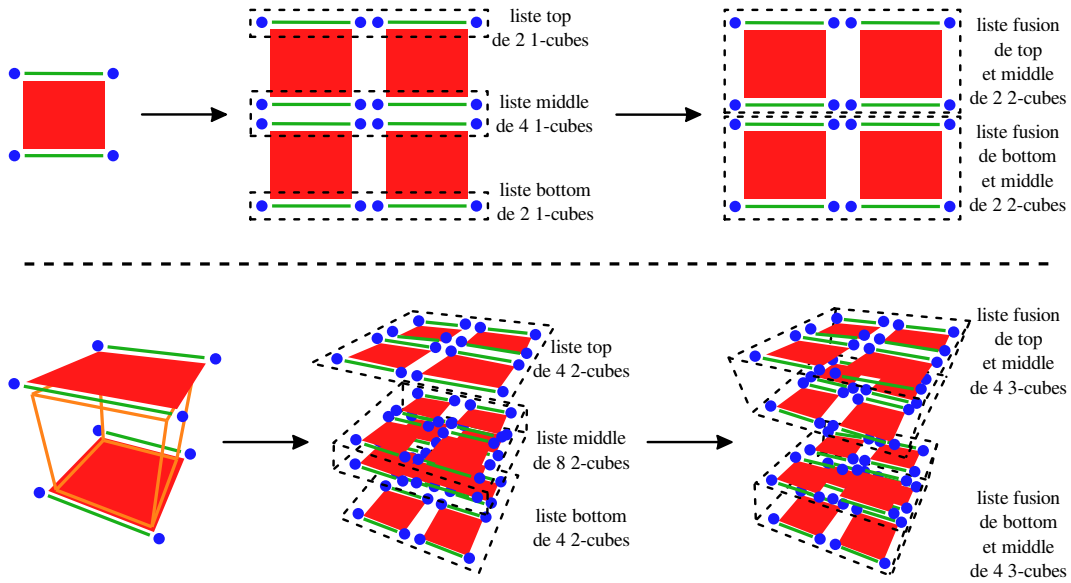


FIGURE 1.3 : Exemple de subdivision d'un 2-cube et d'un 3-cube.

de  $C$ . Dans un second temps, l'algorithme utilise la subdivision linéaire pour calculer la liste de  $2^{d-1}(d-1)$ -cubes appelée *middle*. Cette liste se trouve au milieu des listes *top* et *bottom*. Enfin, l'algorithme retourne  $2^{d-1}d$ -cubes des paires correspondantes de cubes dans *top* et *middle* et  $2^{d-1}d$ -cubes des paires correspondantes dans *middle* et *bottom*. La figure 1.3 illustre l'application de cet algorithme pour un 2-cube et un 3-cube.

La définition présentée ci-dessus permet la représentation de la topologie d'un seul  $d$ -cube. La topologie d'un maillage  $T$  de  $d$ -cubes est simplement composée d'une liste de plusieurs  $d$ -cubes. Pour assurer la consistance topologique de leur représentation, chaque nouveau sommet commun à plusieurs sous-cubes utilise le même index pour les différents sous-cubes auxquels il appartient. Ils utilisent une table de hachage indexée par le plus petit  $d$ -cube contenant chaque sommet pour maintenir ces indices. Par exemple, un sommet inséré dans une arête est indexé par les deux sommets incidents à cette arête.

### Structures indexées centrées volumes

La troisième structure de données, issue de [106], est une structure représentant explicitement les volumes. C'est une structure que l'on peut dériver de la structure présentée précédemment pour le cas tétraédrique/octaédrique. Cette structure consiste en une liste des volumes du maillage. Chaque volume

correspond à une liste d'indices dans un tableau regroupant la position des sommets. Leur algorithme ne nécessite pas de recherche de voisinage ou d'algorithmes permettant de traverser le maillage. Il n'y a donc pas besoin de stocker plus de connectivité.

Une seule opération est possible : l'ajout de nouveaux volumes tétraédriques ou octaédriques. Cette opération s'effectue en supprimant un élément de la liste puis en créant de nouveaux éléments tout en se servant des indices de sommets préexistants afin d'éviter toute duplication.

## Arêtes radiales modifiées

La dernière structure de données présentée, issue de [88], consiste en une version simplifiée de la structure de données d'arêtes radiales [95, 116] qui est une généralisation de la structure de données d'arêtes ailées aux complexes arbitraires. Elle est utilisée pour encoder la subdivision de maillages hexaédriques.

L'implémentation présentée consiste en quatre listes stockant les volumes, les faces, les arêtes et les sommets. Chaque entité topologique (volume, face, arête ou sommet) contient différentes listes représentant son voisinage topologique local. Ainsi, dans leur implémentation, un objet « face » consiste en une liste ordonnée d'arêtes orientées et une paire de pointeurs vers les volumes qui partagent cette face. Le tableau 1.1 répertorie les informations stockées pour chaque entité topologique apparaissant dans cette structure de données. Les entités topologiques « Use » représentent les relations d'incidence entre les objets volumes, faces et arêtes. Un tel objet stocke en plus l'orientation d'une arête (d'une face) en regard de la face (du volume) à laquelle elle appartient.

Des informations auxiliaires, incluant entre autre un drapeau indiquant si une entité est sur le bord ou à l'intérieur, la position géométrique des sommets, sont également stockées pour l'algorithme de subdivision.

## Conclusion

Ces structures de données sont spécifiques à un raffinement et, pour la plupart, ont l'avantage d'être compactes. Elles sont cependant moins souples dans leur utilisation car spécifiques à ce pourquoi elles ont été conçues. Nous allons voir comment, en nous donnant un cadre plus général, il est possible

|   |   |
|---|---|
| Volume  | Face  |
| Liste de Sommet<br>Liste de FaceUse                       | Liste d'ArêteUse<br>Volume voisin1, voisin2       |
| Arête   | Sommet  |
| Sommet point1, point2<br>Liste de Volume<br>Liste de Face | Liste de Volume<br>Liste de Face<br>Liste d'Arête |
| FaceUse   | ArêteUse  |
| Face face<br>Drapeau orientation                          | Arête arête<br>Drapeau orientation                |

**TABLE 1.1** : Informations stockées par la version modifiée des arêtes radiales.

d'obtenir une structure de données générique représentant des quasi-variétés de dimension  $n$  grâce à une unique entité topologique. De plus, nous verrons dans les chapitres suivants comment, grâce à ce cadre formel, nous pouvons dériver des algorithmes de raffinements très généraux et originaux.

## 1.2.2 Modèles structurés

Nous cherchons à représenter des objets discrétisés en cellules de différentes dimensions. Ces cellules partagent entre elles des relations d'incidences et d'adjacences. Ce type d'objets est appelé un complexe cellulaire.

Un objet de dimension  $n$  peut être vu comme l'union disjointe d'un ensemble de  $k$ -cellules ( $0 \leq k \leq n$ ) où  $k$  représente la dimension de la cellule. Le bord d'une  $k$ -cellule est défini récursivement comme l'union d'un ensemble de  $(k - 1)$ -cellules ( $1 \leq k \leq n$ ). Ces deux notions nous permettent de définir les notions d'incidence et d'adjacence. Deux cellules sont dites incidentes si l'une appartient au bord de l'autre. Deux cellules sont dites adjacentes si ces cellules sont de même dimension et s'il existe une cellule du complexe cellulaire incidente à ces deux cellules.

Ces informations d'incidence et d'adjacence entre cellules nous donnent uniquement une représentation combinatoire de la topologie de l'objet. On obtient donc un graphe abstrait sans information supplémentaire (géométrie, etc.) liée aux cellules des différentes dimensions. Un modèle de plongement peut ensuite être lié à ce graphe. Ceci permet d'associer à chaque entité de la subdivision topologique une information et ainsi de séparer la topologie des objets de leurs plongements.

Notons que l'espace de plongement n'est pas forcément lié à la dimension du graphe (c'est-à-dire de l'objet). En effet, on plonge ainsi classiquement un objet surfacique représenté par un graphe 2D dans un espace géométrique 3D, ou bien encore un objet volumique représenté par un graphe 3D dans un espace géométrique 3D.

## Graphe d'incidence

Une représentation simple de cellules et de leurs relations d'incidences consiste à les enregistrer dans un graphe appelé le graphe d'incidence [46]. Il s'agit d'un graphe orienté dont les nœuds correspondent aux cellules et où chaque arc orienté connecte une  $k$ -cellule à une  $(k - 1)$ -cellule de son bord. Ce graphe permet de représenter une grande diversité d'objets : des variétés ou non, orientable ou non, avec ou sans bord.

Cependant, dans ce type de représentation certaines requêtes de voisinage ne sont pas accessibles directement. Il est ainsi impossible de trouver toutes les  $(k + 1)$ -cellules incidentes à une  $k$ -cellule donnée. Cette requête requiert de traverser tout le graphe et est donc résolue à un coût prohibitif. Pour améliorer cette complexité, le graphe d'incidence peut être augmenté par des arcs de direction opposée reliant les  $k$ -cellules à leurs  $(k + 1)$ -cellules incidentes.

Le parcours de toutes les  $(k - 1)$ -cellules incidentes à une  $k$ -cellule, comme par exemple toutes les arêtes incidentes à une face, est très compétitif car directement accessible. À l'inverse, les opérations pour maintenir la cohérence du graphe sont d'une complexité rédhibitoire dans le cas général.

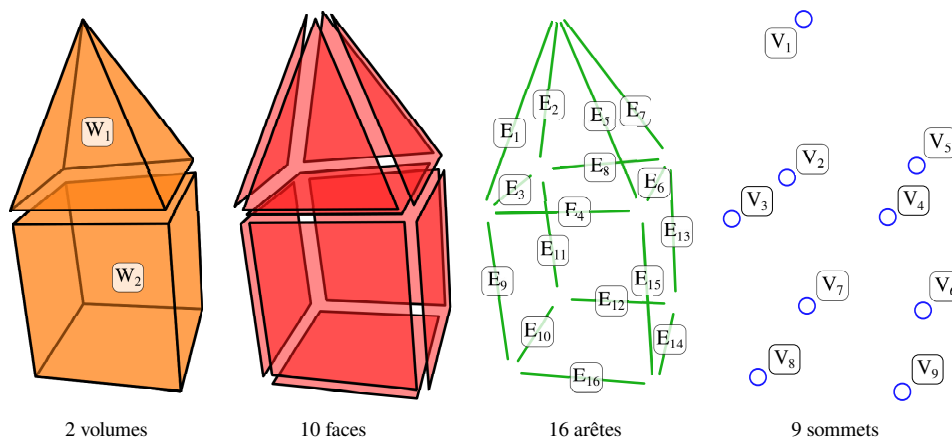


FIGURE 1.4 : Décomposition cellulaire d'un objet tridimensionnel.



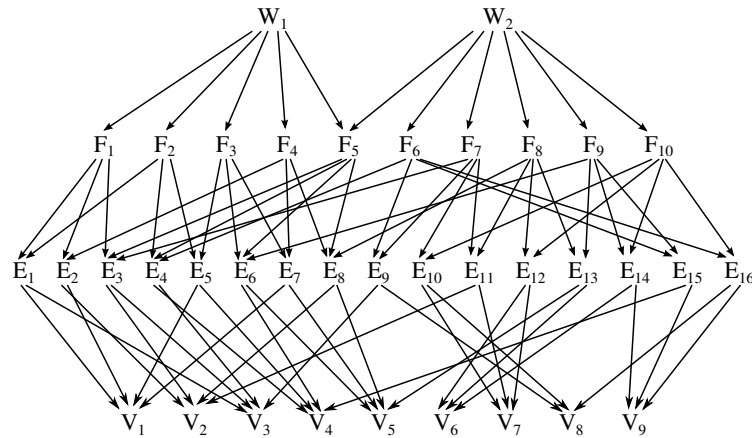


FIGURE 1.5 : Graphe d'incidence de l'objet en 3 dimensions de la figure 1.4.

La figure 1.4 illustre la décomposition cellulaire d'un objet tridimensionnel consistant en un cube et une pyramide. Les faces ne sont pas numérotées sur la figure pour une question de lisibilité. La figure 1.5 illustre le graphe d'incidence obtenu pour l'objet de la figure 1.4.

Une spécialisation au domaine de représentation des variétés cellulaires, plus restrictif mais englobant les objets de notre domaine d'étude, permet de dériver des modèles plus simples et plus efficaces.

**Définition 1.** On définit une variété de dimension  $n$  comme un objet dont tout point admet un voisinage homéomorphe à une boule de  $\mathbb{R}^n$ .

Par exemple en dimension 1, un objet est une 1-variété s'il admet en tout point un voisinage homéomorphe à un segment de droite. En dimension 2, un objet est une 2-variété s'il admet en tout point un voisinage homéomorphe à un disque. Une variété de dimension 2 s'appelle communément une surface. En dimension 3, un objet est une 3-variété s'il admet en tout point un voisinage homéomorphe à une sphère. On appelle volume une variété de dimension 3.

Une  $n$ -variété est dite orientable s'il est possible de définir une direction transportée par continuité en tout point de la variété. Dans le cas contraire, elle est dite non-orientable.

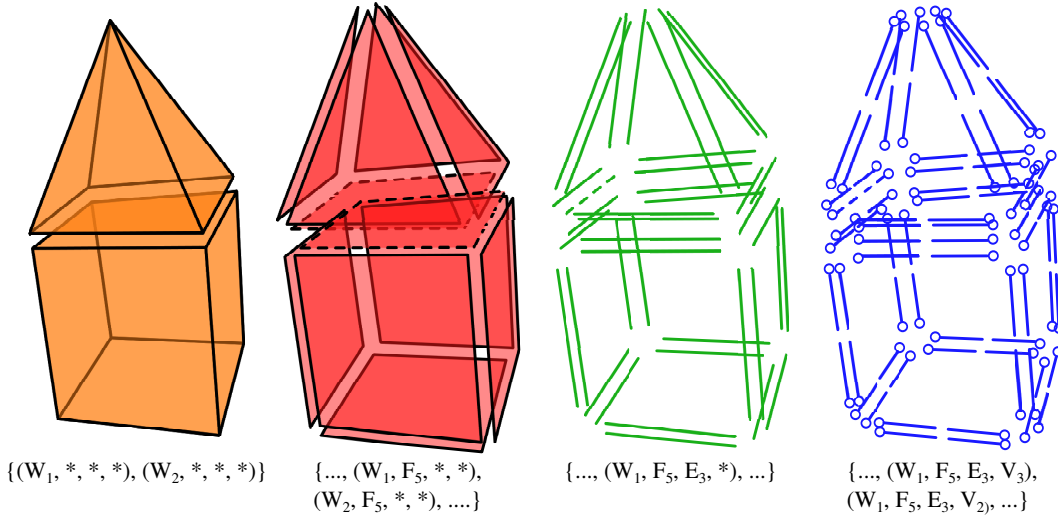
## Cell-tuples

À partir du graphe d'incidence restreint aux variétés cellulaires, Brisson [9, 10] a étudié la manière dont les cellules sont ordonnées autour de cellules de

dimensions inférieures définissant ainsi la structure algébrique de cell-tuples en dimension quelconque.

Étant donné la décomposition cellulaire d'un complexe de dimension  $d$ , un cell-tuple est défini comme une séquence ordonnée de cellules de dimension décroissante  $(C_n, C_{n-1}, \dots, C_1, C_0)$  tel que  $\forall i : 0 < i \leq n, C_i$  est incident à  $C_{i-1}$ . En d'autres termes, un cell-tuple correspond à un chemin dans le graphe d'incidence d'une cellule de dimension  $n$  jusqu'à une cellule de dimension 0 (un sommet).

La figure 1.6 montre la construction itérative de tous les cell-tuples générés par la décomposition cellulaire de la figure 1.4.

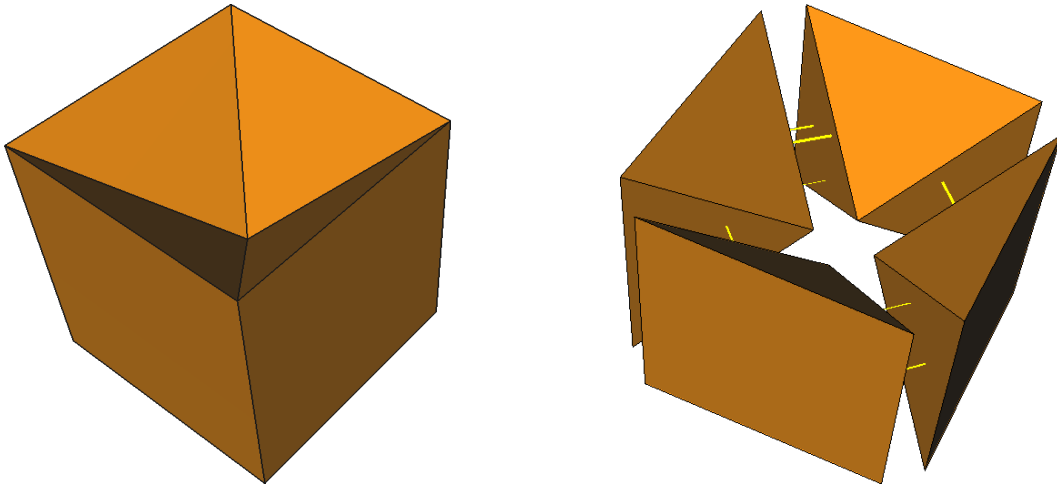


**FIGURE 1.6 :** Construction itérative des cell-tuples correspondants à la décomposition cellulaire de la figure 1.4. Tous les cell-tuples ne sont pas décrits pour ne pas alourdir la figure. Les cell-tuples indiqués correspondent aux cellules en trait pointillé.

Les relations d'adjacences sont définies sur les cell-tuples : deux cell-tuples sont dits  $i$ -adjacents si leur chemin dans le graphe d'incidence partage tout sauf les cellules de dimension  $i$ . Par exemple, dans la figure 1.6  $(W_1, F_5, E_3, V_3)$  et  $(W_1, F_5, E_3, V_2)$  sont 0-adjacents.

**Définition 2.** On appelle une quasi-variété de dimension  $n$  un objet de dimension  $n$  obtenu par assemblage de  $n$ -cellules le long de  $(n - 1)$ -cellules formant leur bord tel que chaque  $(n - 1)$ -cellule est incidente à au plus deux  $n$ -cellules.

Par construction, en dimension 2 une quasi-variété est également une variété. À l'inverse, pour un objet de dimension supérieure à 2, cet assemblage ne constitue plus nécessairement une variété comme illustré par la figure 1.7.



**FIGURE 1.7 :** Exemple d'une quasi-variété de dimension 3 qui n'est pas une variété : la boule pincée. Ce complexe est obtenu par l'assemblage de quatre pyramides le long de leurs faces ou 2-cellules (liens en jaune). Par contre, le voisinage du sommet central « pincé » n'est pas homéomorphe à une sphère de  $\mathbb{R}^3$ .

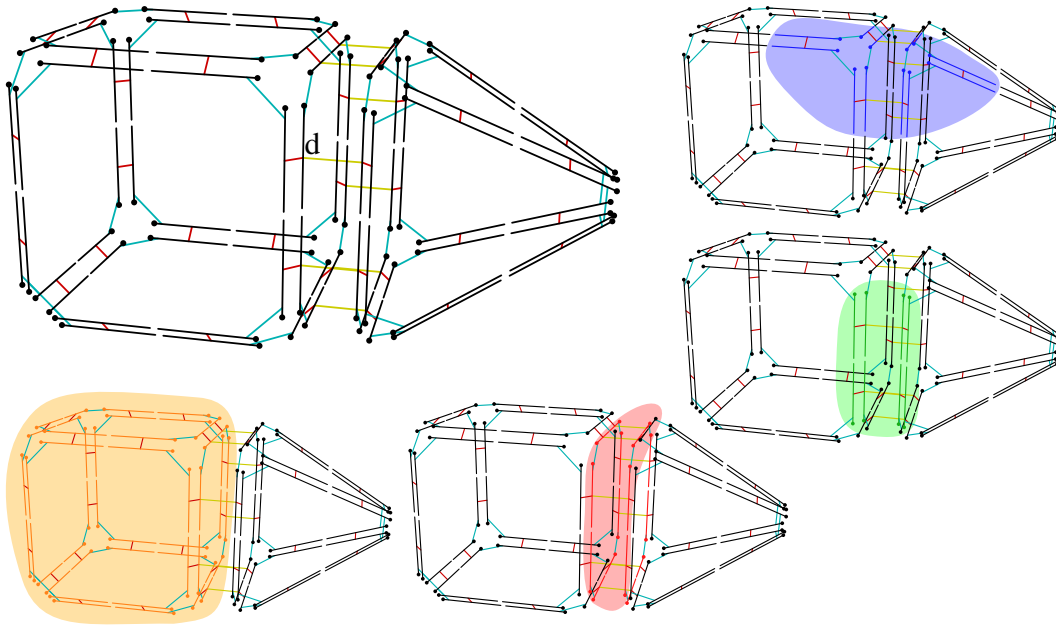
Dans ce contexte de décomposition cellulaire d'une quasi-variété, il a été montré [9, 80] que ces  $n + 1$  relations d'adjacences mettent les cell-tuples en bijection, excepté pour la  $n$ -adjacence au bord de l'objet où les cell-tuples n'ont pas de voisins. Basé sur ces définitions, les cartes généralisées (ou g-cartes) ont été proposées comme un modèle de représentation de la décomposition cellulaire de quasi-variétés de dimension  $n$ , orientables ou non.

## Cartes Généralisées

Une carte généralisée [78, 79] encode la décomposition cellulaire avec un ensemble  $B$  d'éléments abstraits appelés « brins » qui sont en bijection avec les cell-tuples. Un ensemble de  $n + 1$  fonctions  $\beta_i : B \rightarrow B, 0 \leq i \leq n$  sont définies en se basant sur les relations de  $i$ -adjacences des cell-tuples.

Suivant la propriété de bijection sur les  $i$ -adjacences des cell-tuples mentionnées précédemment, les fonctions  $\beta_i$  sont des involutions, c'est-à-dire des fonctions telles que  $\forall d \in B, \beta_i(\beta_i(d)) = d$ .

Des contraintes combinatoires expriment l'assemblage de cellules le long de leurs bords. Pour les fonctions  $\beta_i$ , ces contraintes s'expriment comme suit :  $\forall i, j, 0 \leq i < i+2 \leq j \leq n, \beta_i \circ \beta_j$  est une involution. Si la g-carte est construite à partir du graphe d'incidence de la décomposition d'une quasi-variété, alors ces contraintes sont automatiquement respectées. À l'inverse, ces contraintes



**FIGURE 1.8** : La figure du haut montre la  $g$ -carte correspond à la décomposition cellulaire de la figure 1.4. Les brins sont représentés comme les cell-tuples de la figure 1.6. Les fonctions  $\beta_0, \beta_1, \beta_2$  et  $\beta_3$  sont représentées respectivement par le blanc entre deux brins, les liens bleus, rouges et jaunes. Les quatre autres figures illustrent les ensembles de brins représentant en bleu un sommet, en vert une arête, en rouge une face et en orange un volume du brin  $d$ .

doivent être respectées lors de la construction par l'application d'opérations sur la  $g$ -carte.

**Définition 3.** Une carte généralisée de dimension  $n$ , ou  $n$ - $g$ -carte, est un  $(n + 2)$ -uplet tel que :

$$G = (B, \beta_0, \dots, \beta_n) \quad (1.1)$$

où :

- $B$  est un ensemble fini de brins
- $\forall i : 0 \leq i \leq n, \beta_i$  est une involution sur  $B$
- $\forall i, j : 0 \leq i < i + 2 \leq j \leq n, \beta_i \circ \beta_j$  est une involution

La figure 1.8 illustre une 3- $g$ -carte correspondant à la décomposition cellulaire de la figure 1.4.

Dans cette représentation, les cellules sont décomposées avec leurs relations d'incidences en un ensemble unique d'entités abstraites indépendantes de la dimension. Nous faisons les deux observations suivantes afin d'amener la notion de cellule dans cette représentation. Chaque brin identifie un ensemble

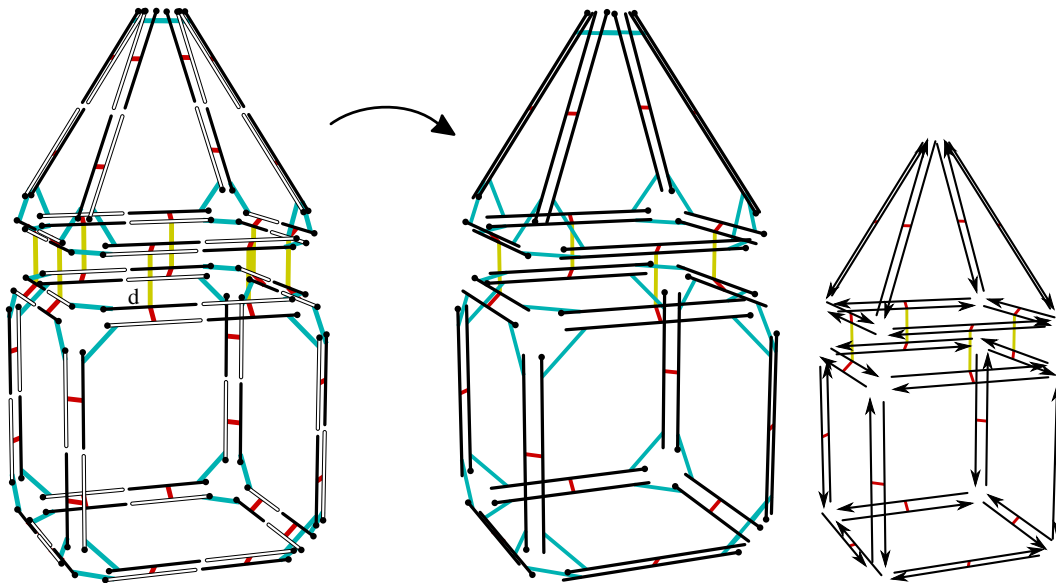
de  $n$  cellules de différentes dimensions, celles contenues dans les cell-tuples correspondants. Pour être plus précis, chaque brin représente de manière égale le sommet, l'arête, la face, etc. Réciproquement, chaque  $k$ -cellule (notée  $cell_k$ ) est représentée par un ensemble de brins, tous les brins dont le cell-tuple correspondant contient cette cellule. La figure 1.8 illustre ces notions. Une opération importante pour pouvoir examiner les relations d'adjacence et d'incidence entre cellules et de savoir énumérer les brins d'une cellule donnée.

Notons que d'après les définitions précédentes,  $\beta_i(d)$  est un brin qui partage toutes les cellules de  $d$  sauf celle de dimension  $i$ . Tous les autres fonctions  $\beta_j$  avec  $j \neq i$  conduisent à des brins qui partagent la même  $i$ -cellule que  $d$ . Il s'en suit qu'à partir d'un brin, l'ensemble des brins représentant la même  $i$ -cellule peuvent être obtenus en appliquant successivement toutes les fonctions qui maintiennent la cellule de dimension  $i$  inchangée, c'est-à-dire les  $\beta_j$  avec  $j \in \{0, 1, \dots, i-1, i+1, \dots, n\}$ . Un tel ensemble de brins est formellement défini par la notion d'orbite notée :  $\langle \beta_0, \dots, \beta_{i-1}, \beta_{i+1}, \dots, \beta_n \rangle$ . Par exemple, dans la figure 1.8, le sommet (ou  $cell_0$ ), l'arête (ou  $cell_1$ ), la face (ou  $cell_2$ ) et le volume (ou  $cell_3$ ) de  $d$  sont définies respectivement par les orbites suivantes :  $\langle \beta_1, \beta_2, \beta_3 \rangle (d)$ ,  $\langle \beta_0, \beta_2, \beta_3 \rangle (d)$ ,  $\langle \beta_0, \beta_1, \beta_3 \rangle (d)$  et  $\langle \beta_0, \beta_1, \beta_2 \rangle (d)$ .

Rappelons qu'une  $g$ -carte est capable de représenter des quasi-variétés orientables ou non orientables. Or pour un certain nombre d'applications, il est suffisant de se restreindre au domaine de représentation des objets orientables. Nous allons donc utiliser un modèle plus compact défini à partir de celui des cartes généralisées, les cartes combinatoires.

## Cartes combinatoires

L'orientation d'une carte généralisée peut être déterminée par un processus de coloration binaire de ses brins en suivant la règle : un brin d'une couleur ne peut seulement être lié qu'à des brins de l'autre couleur. En prenant un brin aléatoire, si tout l'objet peut être colorié de cette façon, alors l'objet est orientable. Dans ce cas, les brins de la  $g$ -carte sont partitionnés en deux ensembles  $B^{noir}$  et  $B^{blanc}$  de cardinalité identique, chacun de ces ensembles représentant une des deux orientations de l'objet. Plus formellement, pour tout brin  $d \in B$ , il est possible de construire l'ensemble des brins correspondant à l'orientation de  $d$  grâce à l'orbite  $\langle \beta_1 \circ \beta_0, \dots, \beta_n \circ \beta_0 \rangle (d)$ . Il est possible de construire un tel ensemble alternativement pour chaque orientation. La figure 1.9 illustre l'application de cette procédure sur la  $g$ -carte de la figure précédente.



**FIGURE 1.9 :** L'image de gauche montre les brins de la g-carte partitionné en deux ensembles correspondant chacun à une des deux orientations de l'objet. Au milieu, la carte combinatoire orientée est construite à partir du brin  $d$ . Les relations  $\phi_1 = \beta_1 \circ \beta_0$ ,  $\phi_2 = \beta_2 \circ \beta_0$  et  $\phi_3 = \beta_3 \circ \beta_0$  sont respectivement représentées en bleu, rouge et jaune. À droite, se trouve la représentation qui sera utilisée ultérieurement. La flèche sur chaque brin matérialise la relation  $\phi_1$  à la place du trait bleu précédent.

Il n'est pas nécessaire de garder ces deux ensembles car ces deux orientations représentent exactement le même objet dans une g-carte. Un de ces ensembles peut donc être supprimé permettant d'obtenir un modèle deux fois plus petit.

Une orientation d'une g-carte orientable est une carte combinatoire définie comme un ensemble de brins  $B$  ainsi que  $n$  fonctions  $\phi_i : B \rightarrow B, 1 \leq i \leq n$ , avec  $\phi_i = \beta_i \circ \beta_0$ .

**Définition 4.** Nous définissons une carte combinatoire de dimension  $n$  (ou  $n$ -carte) formellement par un  $(n + 1)$ -uplet tel que :

$$C = (B, \phi_1, \dots, \phi_n) \quad (1.2)$$

où :

- $B$  est un ensemble fini de brins
- $\phi_1$  est une permutation sur  $B$
- $\forall i : 2 \leq i \leq n, \phi_i$  est une involution sur  $B$
- $\forall i, j : 1 \leq i < i + 2 \leq j \leq n, \phi_i \circ \phi_j$  est une involution

La fonction  $\phi_1$  est une permutation qui lie les sommets ordonnés autour de

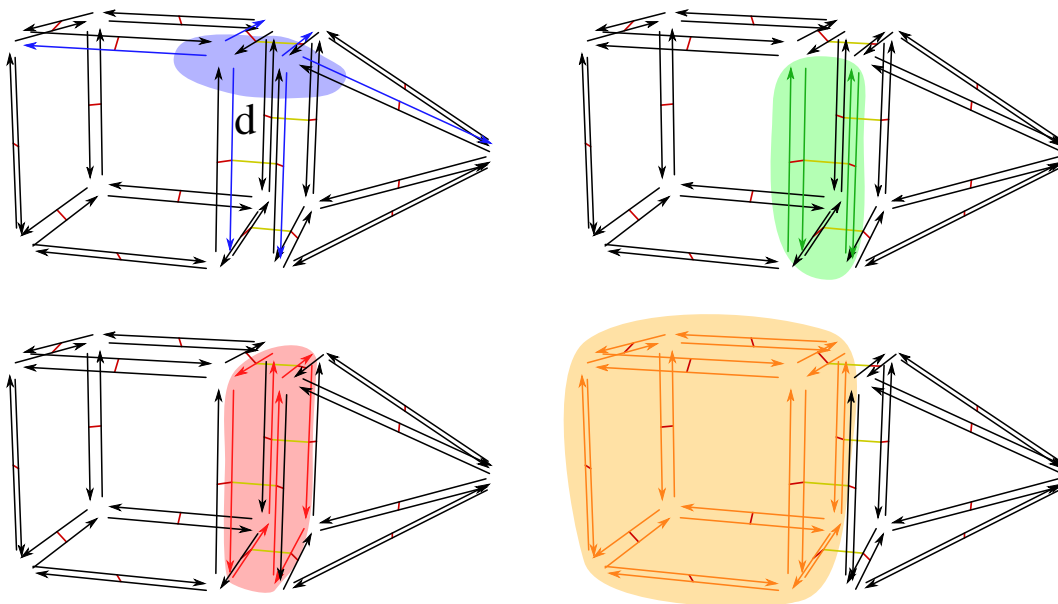


FIGURE 1.10 : Les quatre figures illustrent respectivement les ensembles de brins représentant un sommet, une arête, une face et un volume du brin  $d$  pour le formalisme dual.

faces orientées. Les fonctions  $\phi_i, 2 \leq i \leq n$  sont des involutions tels que stipulé par la contrainte exprimée sur les involutions  $\beta_i$  des  $g$ -cartes.

Dans une approche constructive, chacune de ces involutions permet de coudre des paires de cellules de dimension  $i$  le long de leurs cellules de bord communes de dimension  $i - 1$ . Comme pour les  $g$ -cartes, les orbites se définissent en commençant par n'importe quel brin, puis en appliquant toutes les fonctions qui maintiennent la cellule de dimension  $i$  inchangée. Les orbites définissant les cellules s'écrivent donc comme suit : pour les cellules de dimension  $i \geq 1$ , les ensembles de brins représentant les cellules sont définis par l'orbite :  $\langle \phi_1, \dots, \phi_{i-1}, \phi_{i+1}, \dots, \phi_n \rangle$ . Pour les sommets, l'orbite est définie par :  $\langle \phi_1 \circ \phi_2, \dots, \phi_1 \circ \phi_n \rangle$ . Par exemple, dans la figure 1.10, le sommet (ou  $cell_0$ ), l'arête (ou  $cell_1$ ), la face (ou  $cell_2$ ) et le volume (ou  $cell_3$ ) de  $d$  sont définies respectivement par les orbites suivantes :  $\langle \phi_1 \circ \phi_2, \phi_1 \circ \phi_3 \rangle$ ,  $\langle \phi_2, \phi_3 \rangle$ ,  $\langle \phi_1, \phi_3 \rangle$  et  $\langle \phi_1, \phi_2 \rangle$ .

Cette définition de la représentation des cartes combinatoires est dite *duale*. Il existe une représentation, donnant exactement les mêmes informations, appelée représentation *primale* de la carte.

**Définition 5.** Nous définissons une carte combinatoire primale de dimension  $n$  (ou  $n$ -carte) formellement par un  $(n + 1)$ -uplet tel que :

$$C = (B, \alpha_0, \dots, \alpha_{n-1}) \quad (1.3)$$

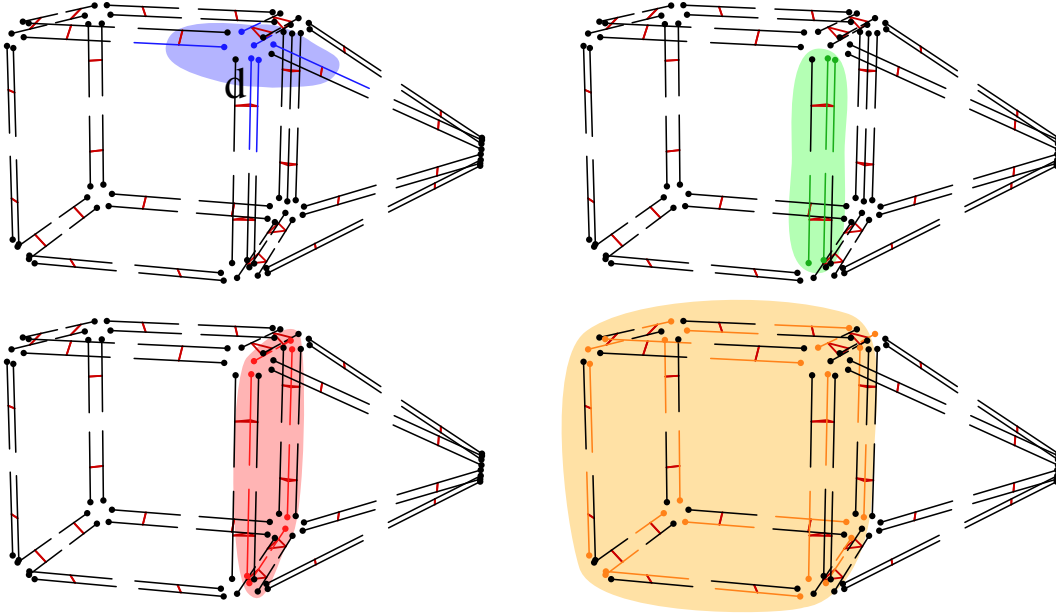


FIGURE 1.11 : Les quatre figures illustrent respectivement les ensembles de brins représentant un sommet, une arête, une face et un volume du brin  $d$  pour le formalisme primal. Les brins du volume constituant le bord de la 3-carte ne sont pas coloriés car ils ne sont pas représentés dans le formalisme dual.

où :

- $B$  est un ensemble fini de brins
- $\alpha_{n-1}$  est une permutation sur  $B$
- $\forall i : 0 \leq i \leq n-2, \alpha_i$  est une involution sur  $B$
- $\forall i : i < n-2$  et  $\forall j : i+1 < j \leq n-1, \alpha_j \circ \alpha_i$  est une involution

Ces deux représentations d'une même carte étant duales l'une de l'autre, il est possible d'exprimer les  $\alpha_i$  en fonction des  $\phi_i$  introduit précédemment [21]. Ainsi,  $\alpha_0 = \phi_n$  et  $\forall i : 1 \leq i \leq n-1, \alpha_i = \phi_i \circ \phi_n$ . Inversement, on peut exprimer les  $\phi_i$  en fonction des  $\alpha_i$  :  $\phi_n = \alpha_0$  et  $\forall i : 1 \leq i \leq n-1, \phi_i = \alpha_i \circ \alpha_0$ .

Pour cette représentation primale, les orbites permettant de parcourir les brins des cellules représentées s'expriment comme :  $\langle \alpha_1 \circ \alpha_0, \dots, \alpha_{n-1} \circ \alpha_0 \rangle$  pour le parcours d'une  $n$ -cellule et  $\langle \alpha_0, \dots, \alpha_{i-1}, \alpha_{i+1}, \dots, \alpha_{n-1} \rangle$  pour les  $i$ -cellules avec  $0 \leq i \leq n-1$ . Par exemple, dans la figure 1.11, le sommet (ou  $cell_0$ ), l'arête (ou  $cell_1$ ), la face (ou  $cell_2$ ) et le volume (ou  $cell_3$ ) de  $d$  sont définies respectivement par les orbites suivantes :  $\langle \alpha_1, \alpha_2 \rangle$ ,  $\langle \alpha_0, \alpha_2 \rangle$ ,  $\langle \alpha_0, \alpha_1 \rangle$  et  $\langle \alpha_1 \circ \alpha_0, \alpha_2 \circ \alpha_0 \rangle$ .

Le fondement mathématique des cartes combinatoires permet leur utilisation dans le cadre plus théorique de la spécification formelle [22] ou encore



la preuve d'algorithmes géométriques par l'utilisation de méthodes de preuves formelles à l'aide d'un assistant de preuve tel que Coq [11]. Il permet également de séparer les opérations définies sur le modèle, en terme d'ensembles et de permutations, de leur implantation en structures de données concrètes. Nous exploitons cette propriété dans les chapitres suivants.

## Cartes plongées

Les cartes combinatoires et cartes généralisées définissent uniquement la topologie de la décomposition cellulaire d'une quasi-variété de dimension  $n$ . Elles définissent de manière abstraite des cellules, représentées implicitement par des ensembles de brins, et les relations de voisinage entre ces cellules.

Afin de représenter et d'exploiter des propriétés d'autres natures sur ces objets, il est possible d'associer des attributs aux cellules des différentes dimensions. Typiquement, on associe un plongement géométrique dans  $\mathbb{R}^3$  aux sommets (0-cellules) pour visualiser des objets représentés par des cartes de dimension 2 ou 3. Il est tout à fait possible d'ajouter des attributs, comme par exemple une couleur aux faces ou encore une propriété mécanique comme un module d'élasticité aux volumes.

Afin d'attacher de manière consistante des attributs aux cellules, il faut que tout attribut attaché à une cellule soit lié à tous les brins de cette cellule. La solution la plus flexible consiste à associer un index à chaque cellule. Ainsi, toute donnée associée à cet index est alors associée à la cellule correspondante.

Nous définissons donc des fonctions additionnelles  $emb_i$  sur les cartes telles que  $\forall i : 0 \leq i \leq n, emb_i : B \rightarrow \mathbb{N}$  est la fonction qui associe à chaque brin l'index de sa cellule de dimension  $i$ . Une carte est dite « bien plongée » lorsque des contraintes sur cette application sont respectées. Il faut s'assurer que :  $\forall d, d' \in B, d' \in cell_i(d) \implies emb_i(d') = emb_i(d)$ . C'est à dire que pour chaque dimension plongée, tous les brins d'une même orbite soient associés au même index. Inversement, pour des questions d'efficacité, on peut imposer que :  $\forall d, d' \in B, emb_i(d) = emb_i(d') \implies d \in cell_i(d')$ . C'est à dire que pour chaque dimension plongée, les brins associés au même index appartiennent à la même orbite. Dans ce cas, l'application qui associe une orbite de dimension  $i$  à son index est bijective.

## Hypercartes

Les cartes combinatoires et les cartes généralisées sont des modèles topologiques définis par un ensemble de brins et un ensemble de permutations définies sur ces brins. La seule différence entre ces modèles réside dans les contraintes de construction.

Il est possible de définir un cas plus général en s'abstrayant de la sémantique des cell-tuples et en retirant les contraintes d'involution ou d'absence de point fixe sur les permutations. On appelle ce cadre général abstrait, commun à tous ces modèles, une hypercarte [30, 32, 31].

**Définition 6.** *Une hypercarte de dimension  $n$  est un  $(n + 1)$ -uplet tel que :*

$$H = (B, \alpha_0, \dots, \alpha_{n-1}) \quad (1.4)$$

où :

- $B$  est un ensemble fini de brins
- $\forall i : 0 \leq i \leq n - 1, \alpha_i$  est une permutation sur  $B$

Ce modèle permet d'exprimer tous les modèles issus des cartes. Ainsi, une  $n$ -carte est une hypercarte de dimension  $n$  à laquelle des contraintes ont été ajoutées, telles que données par la définition 5. De même, une  $n$ -g-carte est une hypercarte de dimension  $n + 1$  à laquelle des contraintes ont été ajoutées, telles que données par la définition 3. Il nous servira à définir de manière cohérente l'extension multirésolution des cartes combinatoires en dimension quelconque.

## Implémentations

Dans le contexte des maillages surfaciques, la structure en demi-arête [115] est la plus connue. Chaque demi-arête stocke un pointeur vers la demi-arête suivante, un second vers la précédente demi-arête dans sa face orientée, un troisième vers la demi-arête opposée, un quatrième vers le sommet incident et un cinquième vers la face incidente. Les arêtes ne sont usuellement pas explicitement représentées. Plusieurs bibliothèques logicielles mettent en œuvre cette structure de données : OpenMesh [8], surface\_mesh [109], CGAL [23].

La structure en demi-arête s'exprime dans le cadre formel des cartes comme une carte combinatoire de dimension 2. Chaque demi-arête correspond à un brin, le pointeur vers la demi-arête suivante correspond à la permutation  $\phi_1$ , le pointeur vers la demi-arête précédente à l'inverse de  $\phi_1$  ( $\phi_1^{-1}$  noté  $\phi_{-1}$  dans la

suite) et enfin la demi-arête opposée à l'involution  $\phi_2$ . En terme de plongement, les pointeurs vers le sommet incident et vers la face incidente sont identifiés par les relations  $emb_0$  et  $emb_2$ .

Pour les maillages volumiques, plusieurs structures de données ont été proposées telles que les facet-edge[43] ou encore les handle-face[84]. Ces dernières ont été spécialisées par la suite aux complexes simpliciaux 3-variétés par la structure de données CHF [73] afin de réduire leur complexité mémoire. Il y a également la structure Circular Incident Edge Lists (CIEL) [77] représentant la topologie d'une grille irrégulière afin d'extraire des iso-surfaces efficacement. Classiquement, la face incidente de la structure en demi-arête est appelée demi-face et stocke un pointeur vers la demi-face opposée du volume adjacent. Il n'existe pas de bibliothèques logicielles décrivant l'implantation et les performances de ce type de structures de données.

Ces structures de données pour les maillages volumiques s'exprime également comme des cartes combinatoires de dimension 3. Elles peuvent être vue comme l'assemblage de carte combinatoire de dimension 2 le long de faces communes par l'involution  $\phi_3$ . Cette fonction encode le pointeur vers la demi-face opposée mais sans avoir à introduire une nouvelle entité : la demi-face pour stocker la relation.

L'équivalence théorique de toutes ces représentations avec le modèle des cartes combinatoires permet de montrer qu'il y a une équivalence théorique entre ces structures de données et une structure de donnée mettant en œuvre le modèle formel.

Concernant les modèles à base de cartes, il existe trois mises en œuvre matures. Une première consiste en une implantation des 3-g-cartes dans Moka [94], une seconde consiste en un paquet pour la bibliothèque logicielle CGAL mettant en œuvre les cartes combinatoires  $nD$  [34] et enfin la bibliothèque CGoGN comprenant les cartes combinatoires  $nD$  et les cartes généralisées  $nD$  [24]. Une implantation compacte du modèle des 3-cartes combinatoires consistant à compresser la connectivité par un jeu de reformulation des relations du modèle a également été proposée [49].

À contre pied, la structure mise en œuvre par OpenVolumeMesh [71] mélange un graphe d'incidence pour l'accès aux incidences des  $(k - 1)$ -cellules à partir des  $k$ -cellules et une structure de demi-face qui stocke des listes pour les incidences vers les  $(k + 1)$ -cellules. Ce mélange permet des accès rapides pour le parcours des incidences. Cependant, le maintien des listes d'incidences, ou encore leurs reconstructions après l'application d'une série d'opérations modifiant la topologie, rend cette structure inefficace lorsque la connectivité re-

quiert d'être dynamique par la création ou la suppression de cellules, comme nous verrons plus en détail au chapitre 4.

## 1.3 Structures multirésolutions

### 1.3.1 Généralités

Les modèles, présentés dans la section précédente, permettent de représenter un maillage unique, c'est-à-dire une seule discrétisation d'un échantillonnage donné. Il est possible d'effectuer des opérations sur ce maillage afin d'ajouter ou de supprimer des cellules. Cependant le traitement s'effectue sur ce maillage et altère celui-ci sans retour possible.

Une représentation multirésolution d'un objet discrétisé doit permettre l'accès à une séquence de maillage représentant différentes versions du même objet. On parle alors dans ce cas de différents niveaux de résolution ou encore de différentes échelles d'un même objet. Ainsi, un bout de la séquence fait exister un objet dans sa version grossière et le bout opposé de la séquence fait exister l'objet dans sa version fine. Les étapes intermédiaires de la séquence permettent de faire varier le niveau de résolution.

La représentation multirésolution est utilisée comme support pour accélérer certains traitements. On peut citer notamment des applications de transmissions progressives, ou encore en visualisation scientifique afin d'extraire des informations pertinentes de manière sélective dans de gros volumes de données [28]. Elle permet également d'accélérer la détection de collision dans des subdivisions surfaciques ou volumiques. Par exemple en simulation de foule [65], le niveau de résolution le plus fin est utilisé comme structure accélératrice pour résoudre les requêtes de proximité alors que les niveaux plus grossiers sont utilisés pour le calcul de la trajectoire. En analyse numérique, une représentation multirésolution d'un problème permet d'adapter la taille du système linéaire à résoudre en fonction de divers critères. Il est possible de simuler des phénomènes naturels tels que les gaz [102], les liquides [100], la fumée [85], les corps déformables [19], etc.

On distingue deux approches pour la construction d'une représentation multirésolution. Il y a l'approche dite « grossier vers fin » qui, partant d'un maillage initial, crée des versions de plus en plus fines en raffinant successivement le maillage. On parle alors également de sur-échantillonnage. Puis, il y a l'approche dite « fin vers grossier » qui, partant d'un maillage initial, crée

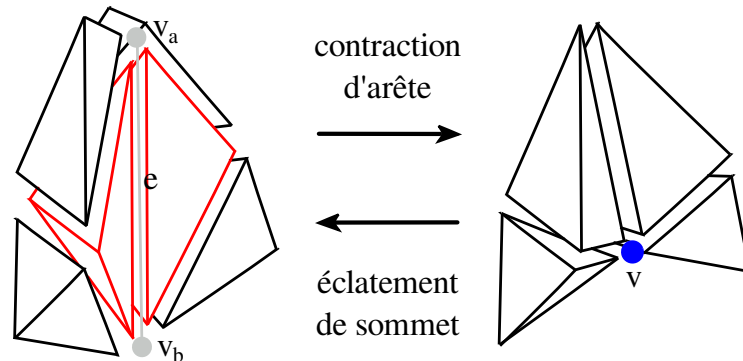


FIGURE 1.12 : Opération de contraction d'arête et d'éclatement de sommet.

des versions de plus en plus grossières en simplifiant le maillage de manière itérative. On parle alors de sous-échantillonnage.

La solution couramment utilisée [40] pour représenter un objet à différentes échelles consiste à stocker d'une part le maillage initial dans une structure standard « monorésolution », et d'autre part de disposer d'un ensemble de mises à jour qui, une fois appliquées au maillage initial, fournissent différentes résolutions du maillage représentant l'objet. En outre, il faut mettre en place une relation de dépendance entre ces mises-à-jour, pour permettre de les combiner, et d'extraire une représentation intermédiaire consistante.

### 1.3.2 Maillages progressifs tétraédriques

Les maillages progressifs tétraédriques (MPT) [110, 101, 113, 97, 55] sont une extension volumique des maillages progressifs proposés initialement par Hoppe [62]. Un MPT encode un maillage grossier et une séquence linéaire d'opérations de mise à jour. Ces opérations peuvent être appliquées de manière linéaire au maillage afin de le raffiner progressivement. L'extraction d'un maillage, pour ce type de modèle, est uniquement possible aux niveaux de résolution intermédiaires pouvant être obtenus par l'application successive d'une séquence finie d'opérations.

Le modèle MPT se construit à partir d'un maillage de référence et par l'application successive d'une opération de contraction d'arête produisant le maillage de base qui représente l'approximation tétraédrique grossière. L'opération de contraction d'arête, également appelée contraction « pleine » d'arête et illustrée figure 1.12, consiste à fusionner deux sommets ( $v_a, v_b$ ) incidents à une arête  $e$  en un unique sommet  $v$ . La position géométrique du sommet résultat se trouve au milieu de l'arête  $e$ .

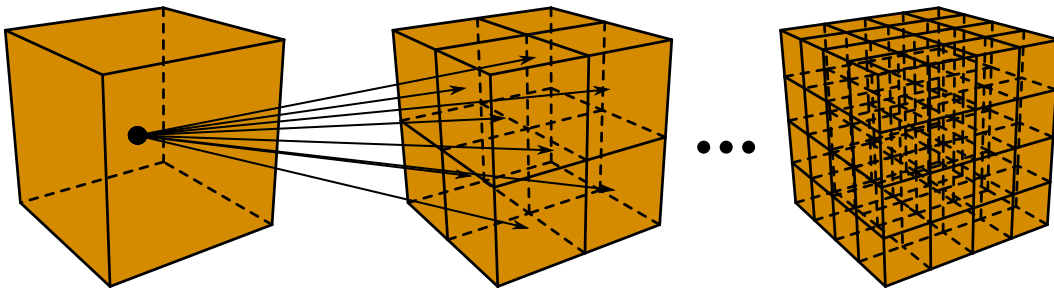


FIGURE 1.13 : Raffinement d'un octree.

Tout comme les maillages progressifs, le modèle MPT est formé d'un maillage de base  $M_0$  ainsi que d'une séquence inversée d'opérations d'éclatements de sommets. L'opération d'éclatement « plein » de sommet, illustré figure 1.12, éclate un sommet  $v$  en une arête  $e = (v_a, v_b)$  et partitionne l'ensemble des tétraèdres incidents au sommet  $v$  (également appelé l'étoile de  $v$ ) en tétraèdres incidents à  $v_a$  ou  $v_b$ .

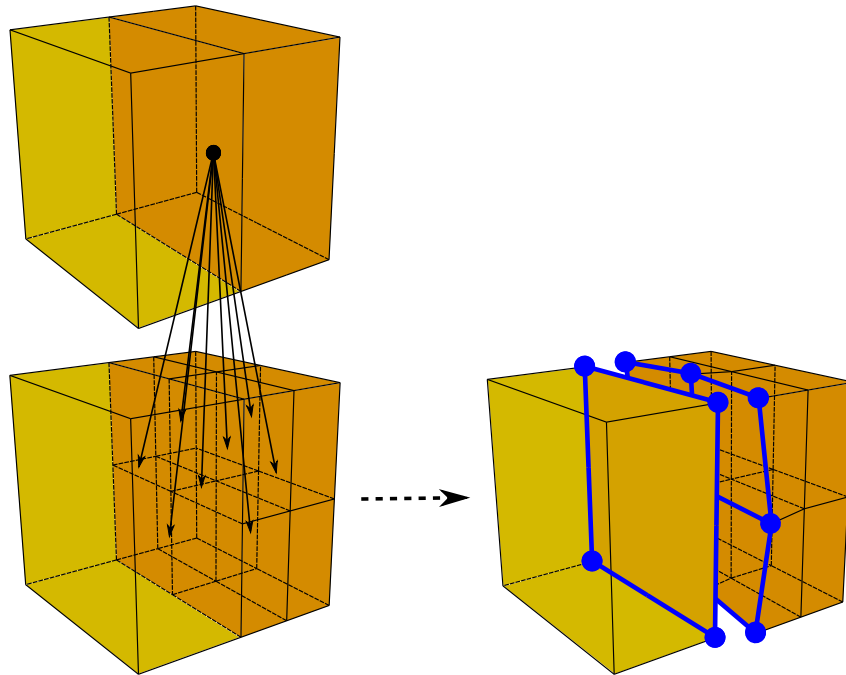
Dans [110], chaque éclatement de sommet est encodé en spécifiant le sommet éclaté  $v$  ainsi que l'éventail de faces triangulaires incidentes aux sommets qui seront éclatés en tétraèdres.

Dans [97], un encodage plus compacte du modèle MPT est proposé. Il est étudié à des fins de compression ou de transmission. Cet encodage consiste en un maillage de base ainsi qu'une série d'éclatements de sommets par lots effectués simultanément pour construire le prochain niveau de résolution.

### 1.3.3 Octree

Un arbre octal, ou octree [105], est d'abord un modèle de partitionnement de l'espace utilisé pour accélérer les traitements sur un ensemble de points plongés dans un espace tridimensionnel. Ce modèle s'utilise dans l'approche « grossier vers fin ». Il possède une opération de raffinement qui consiste à remplacer un volume par un ensemble de huit volumes. Cette opération de raffinement permet de créer une hiérarchie de volumes qui consiste en une arborescence, dans laquelle on adjoint à un nœud père huit nœuds fils. La figure 1.13 illustre ce type d'octree. La complexité de la recherche d'un élément dans ce modèle est en  $O(\log_8(n))$  où  $n$  est le nombre de nœuds de l'arbre. Une version du modèle d'octree, appelée R-octree, permettant de moduler le placement des plans de coupe, a été proposé [118].

Une utilisation par raffinement régulier consiste à créer un arbre octal com-



**FIGURE 1.14** : Raffinement adaptatif d'un octree. La génération de faces non conformes provoque l'apparition d'un trou topologique visible en bleu. Les sommets incidents à ce trou ont été déplacés afin de le rendre visible. Ce type de non-conformité est également appelé une jonction en T.

plet où toutes les opérations de raffinement sont effectuées pour chaque niveau. Ce modèle étant un outil de partitionnement, il ne possède pas de représentation topologique pour toutes les cellules : seuls les volumes sont explicitement représentés. Dans le cas d'un raffinement adaptatif, cette insuffisance entraîne des problèmes de conformité avec l'apparition de cellules non conformes, comme illustré par la figure 1.14. Les arêtes et les faces n'étant pas représentées, ces cellules ne sont pas découpées par le processus de raffinement et laissent apparaître un trou topologique lorsque des cellules voisines sont de niveaux différents. Afin de réduire ces problèmes, les méthodes utilisant un raffinement adaptatif limitent à un le nombre de niveaux de différence entre des cellules voisines.

La forme d'un objet généré par un octree est limitée à la géométrie d'un cube. Pour obtenir plus de flexibilité dans la géométrie des objets, il faut utiliser une forêt d'octrees. Partant d'une discrétisation initiale, ce modèle se réduit à une forêt d'arbres où chaque volume du maillage initial  $M^0$  est la racine d'un des arbres. Cette méthode est notamment utilisée pour raffiner des objets à la géométrie complexe par les méthodes d'Adaptive Mesh Refinement (AMR) [18].

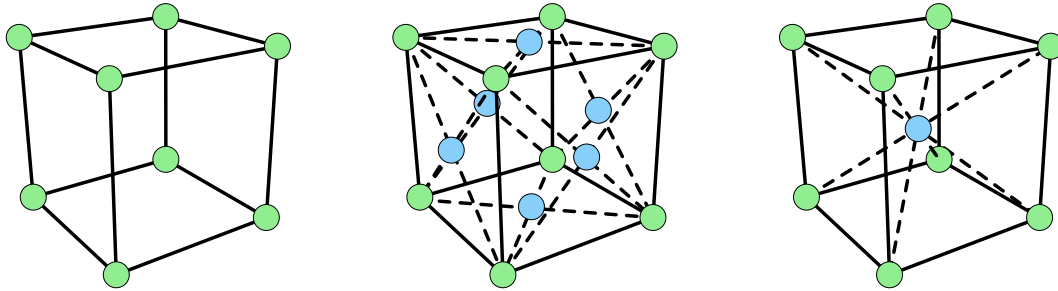


FIGURE 1.15 : Les trois types de pavages respectivement CC, FCC et BCC.

Afin de générer des nœuds à huit fils, un volume doit obligatoirement se découper en 8 volumes analogues au premier. Ceci limite l'utilisation de l'octree à un échantillonnage du domaine cubique cartésien et réduit d'autant le nombre possible d'éléments représentables. Ceci explique qu'il n'existe qu'un seul type de cellules représentables : le cube. Il est d'ailleurs possible d'étendre ce raffinement en dimension quelconque uniquement dans le cas hypercubique. Ainsi, la bissection régulière d'une cellule  $T$  de dimension  $d$  consiste à couper les cellules de dimension  $d > 0$  en deux par l'insertion d'une cellule de dimension  $d - 1$ . Cette opération décompose  $T$  en  $2^d$  cellules disjointes couvrant l'espace formé par  $T$ . Nous reparlerons du raffinement de tétraèdres, et plus généralement de simplexes en dimension quelconque, dans les sous-sections suivantes.

L'arrangement spatial Cubique Cartésien (CC) que définit un octree vient avec une position sur les axes perpendiculaires. Pour pallier à ce défaut de représentation discrète d'un domaine, deux autres pavages de l'espace tridimensionnel sont utilisés, notamment en génération de maillages [112] et en rendu de données par grilles [104]. Ces deux autres pavages sont basés sur deux types d'échantillonnages de l'espace tridimensionnel issus de la cristallographie. Ils peuvent être construits par l'intersection de deux grilles cubiques et nous les illustrons par la figure 1.15. La première, appelée Face Centered Cubic (FCC), ajoute des échantillons aux centres des faces. La seconde, appelée Body Centered Cubic (BCC), ajoute des échantillons aux centres des cubes.

La structure de Multiresolution Sphere Packing Tree (MSP-Tree) [64] permet de construire une représentation hiérarchique pour ce type de grilles. Leur travail est lié au problème d'empilement compact qui consiste à agencer des hypersphères dans un espace donné afin d'avoir la plus grande densité de sphères sans que celles-ci ne se recouvrent. À l'inverse de l'octree qui ne possède qu'une règle de raffinement, cette structure possède un ensemble de règles pour chacun des trois agencements. Pour une gestion cohérente de l'application de ces règles, les nœuds de l'arbre sont différenciés en plusieurs classes en fonction



des caractéristiques du raffinement qui leur est appliqué.

### 1.3.4 Hiérarchie par Bissection Régulière de Simplexes

Le schéma de raffinement en dimension quelconque de la sous-section 1.3.3 ne s'applique pas dans le cas des simplexes [117]. Si en dimension 2, le raffinement régulier d'un triangle génère quatre triangles similaires au premier illustré sur la figure 1.16a. Cette règle n'est plus valable dans les dimensions supérieures. Ainsi, le raffinement d'un tétraèdre génère quatre tétraèdres incidents aux sommets du premier ainsi qu'un octaèdre. Cet octaèdre est illustré sur la figure 1.16b. Il possède huit faces triangulaires construites en prenant une face triangulaire pour chacune des quatre faces d'origine et une face triangulaire pour chacun des quatre sommets de coin d'origine. Ces faces sont définies par les sommets des milieux d'arêtes. La décomposition en tétraèdres de cet octaèdre central n'est pas unique et dépend de la diagonale choisie. Il faut donc trouver des stratégies afin d'obtenir un raffinement stable, c'est-à-dire utilisant toujours une diagonale dans la même direction, ou gérer une hiérarchie avec plusieurs types d'éléments. Nous verrons dans la sous-section suivante que quelques travaux ont essayé de construire une hiérarchie tout en conservant les éléments octaédriques.

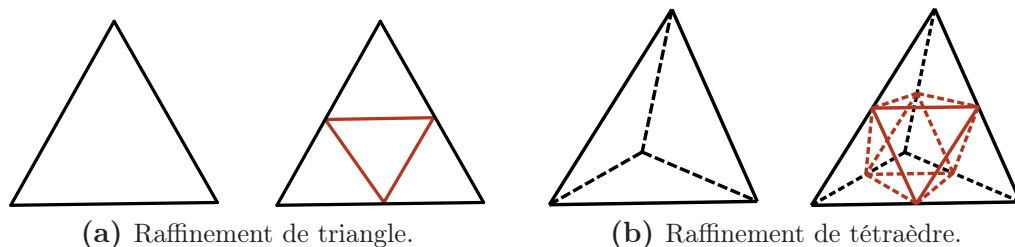


FIGURE 1.16 : Raffinement régulier de simplexe.

Une structure de partitionnement alternative pour les 3-variétés de dimension quelconque consiste à construire une hiérarchie de simplexes par une opération de bissection d'arête. Cette opération, contrairement à la précédente, est consistante en dimension. L'opération de bissection d'arête bissecte un simplexe de dimension  $d$  le long d'un hyperplan défini par le sommet  $v_e$  au milieu d'une arête  $e$  et les  $(d - 1)$  sommets non incidents à  $e$ . La figure 1.17 illustre cette opération en dimension 2 et 3.

Les arêtes d'une grille hypercubique de dimension  $d$ , partitionnée en simplexes de dimension  $d$ , peuvent être regroupées en plusieurs types. Nous illustrons ces différents types d'arêtes en dimension 2 et 3 par la figure 1.18. Comme

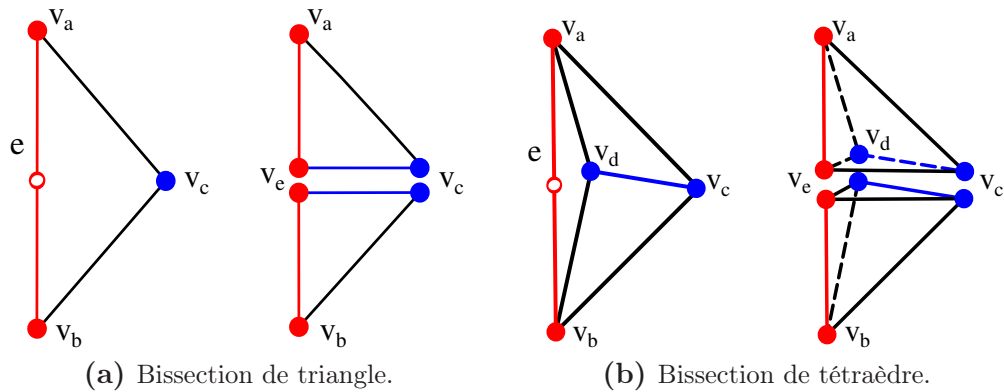


FIGURE 1.17 : Opération de bisection de simplexe le long d'un hyperplan défini par le point  $v_e$  d'une arête  $e$  et par tous les sommets du simplexe non incidents à  $e$ .

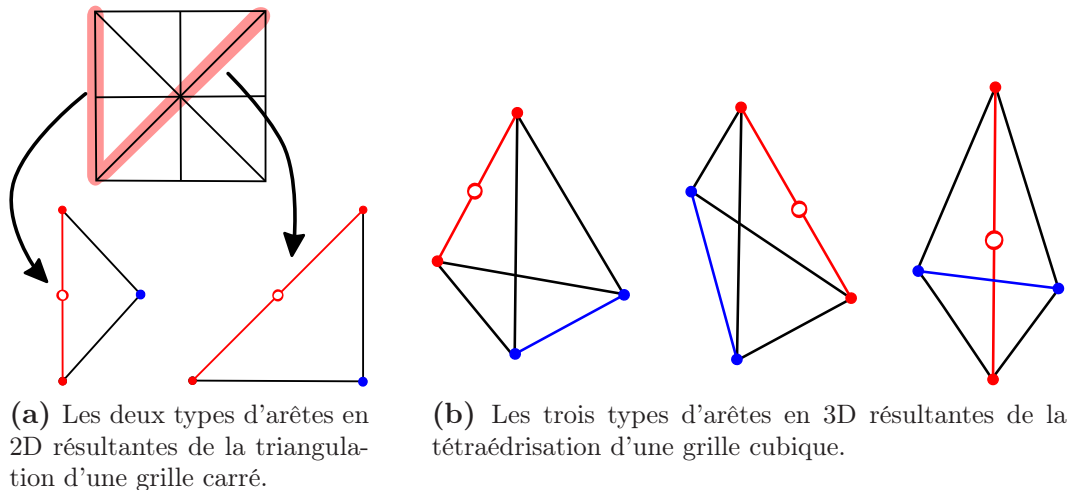
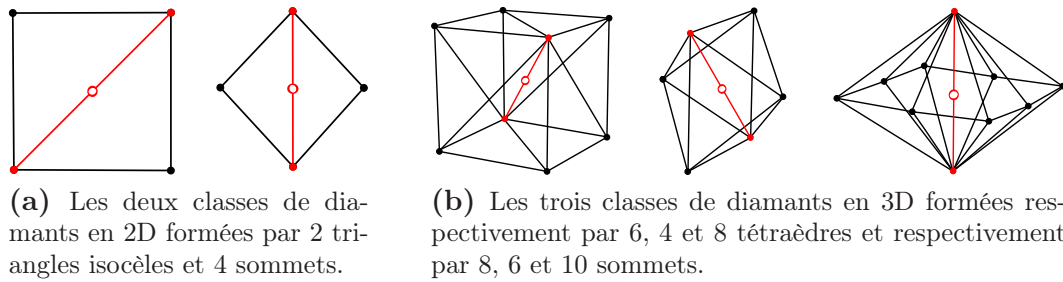


FIGURE 1.18 : Les différentes configurations d'arêtes à bissecter en 2D et 3D. L'arête bissectée est montrée en rouge.

illustré plus en détail par la figure 1.18a, ces arêtes correspondent aux différentes diagonales de la grille.

Il y a deux manières de considérer les éléments d'une hiérarchie construite par cette opération : d'une part, les simplexes eux-mêmes, d'autre part un ensemble de simplexes appelé diamant. Le raffinement régulier par une arête bissecte simultanément tous les simplexes incidents à cette arête. L'ensemble formé par ces simplexes autour de l'arête bissectée est usuellement appelé un diamant. La figure 1.19 illustre les deux classes de diamant en 2D et les trois classes de diamant en 3D.

Nous illustrons les structures de données utilisées dans le cas tétraédrique uniquement sachant que ces concepts s'étendent en dimension quelconque.



**FIGURE 1.19** : Les classes de diamants en 2D et 3D. L'arête bissectée est montrée en rouge.

Nous présentons de manière succincte cette famille de modèles et son mode de fonctionnement. Pour plus de détails le lecteur pourra se référer à l'état de l'art suivant [117].

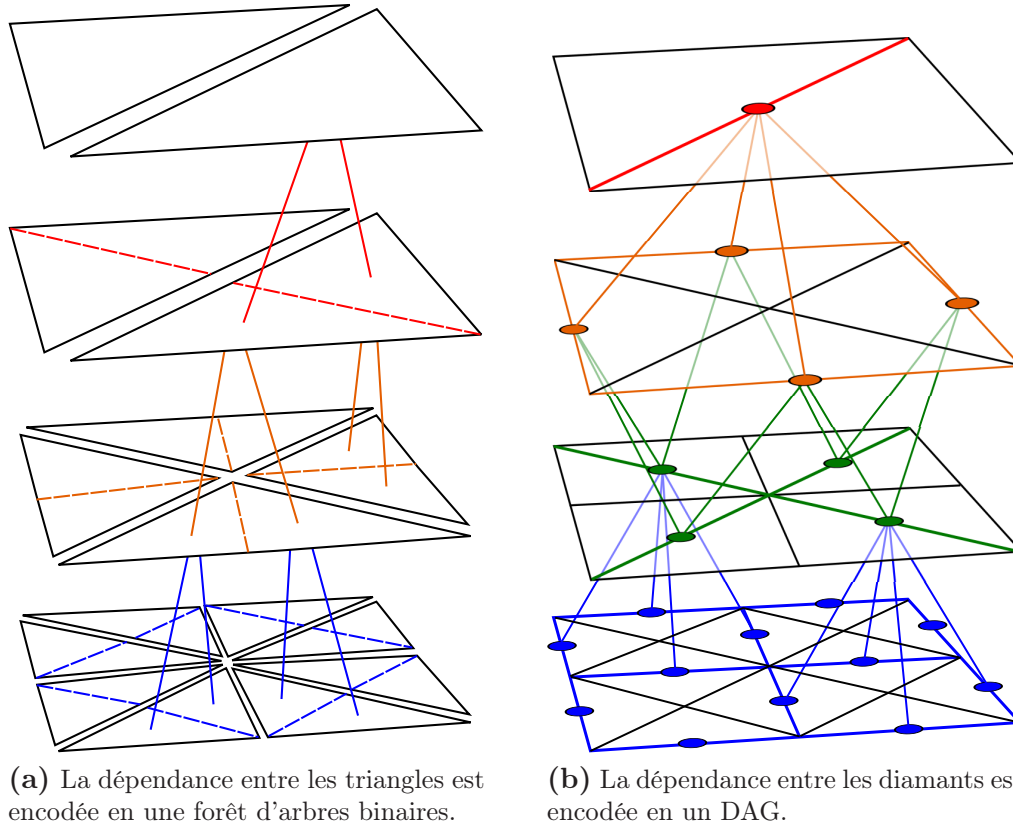
Il y a deux types de structures de données pour encoder la hiérarchie de maillages produit par l'opération de bisection d'arête : une forêt de tétraèdres ou un graphe orienté acyclique (en anglais Directed Acyclic Graph ou DAG) de diamants où les arcs encodent leurs dépendances directes.

La plupart des travaux utilisant ce type de structures de données relèvent de la décomposition d'un domaine cubique uniforme partitionné en tétraèdres. Les voisinages étant fixes, les relations d'incidences et d'adjacences peuvent être résolues de manière implicite. Il n'y a donc pas de connectivité à stocker, ce qui permet d'obtenir des structures de données efficaces.

Dans une forêt de tétraèdres, les racines correspondent aux six tétraèdres résultant de la découpe d'un cube initial. Les autres nœuds décrivent un tétraèdre  $t$  et les deux fils de  $t$  qui sont les deux tétraèdres obtenus en découpant  $t$  le long de sa plus longue arête. Chacun des arbres est un arbre binaire complet qui peut alors être encodé implicitement. Une forêt de tétraèdres étend directement le principe de la forêt de triangles utilisée en dimension 2 et que nous illustrons par la figure 1.20a. Notons en plus que la forêt n'a pas besoin d'être explicitement encodée sauf si des attributs doivent être associés aux tétraèdres à chaque niveau de résolution.

Un DAG peut être utilisé comme structure de données pour encoder les trois classes de formes de tétraèdres générées par le processus de bisection. La racine du DAG est la subdivision initiale d'un cube. Les autres nœuds du DAG décrivent un diamant et les arcs du graphe décrivent la relation de parenté entre les nœuds. Étant donné un diamant  $D$ , les parents de  $D$  sont les diamants qui doivent être coupés pour créer les tétraèdres de  $D$ . Les diamants créés lorsque  $D$  est à son tour coupé sont les enfants de  $D$ . Là encore le principe étend

directement la version en dimension 2 que nous illustrons par la figure 1.20b.



**FIGURE 1.20** : Illustration de deux structures de données dont la hiérarchie est générée par une opération de bisection d'arête.

Dans [59] une structure de données compacte pour encoder un DAG de diamant est proposée. La structure de DAG n'est pas enregistrée de manière explicite. Les informations sur les diamants ainsi que les informations sur les attributs sont attachées aux sommets dans une structure de données pour maillages tétraédriques.

### 1.3.5 Hiérarchie de Tétraèdres/Octaèdres

Une solution alternative consiste en un schéma de raffinement de tétraèdres et d'octaèdres [61, 60] dans lequel les octaèdres sont traités comme des éléments à part entière de la décomposition. Dans ce schéma, chaque tétraèdre est raffiné en quatre tétraèdres ainsi qu'un octaèdre. Chaque octaèdre est décomposé en six octaèdres incidents à chaque sommet de l'octaèdre initial, ainsi que huit tétraèdres correspondant aux faces triangulaires tronquées de l'octaèdre initial comme illustré par la figure 1.21.

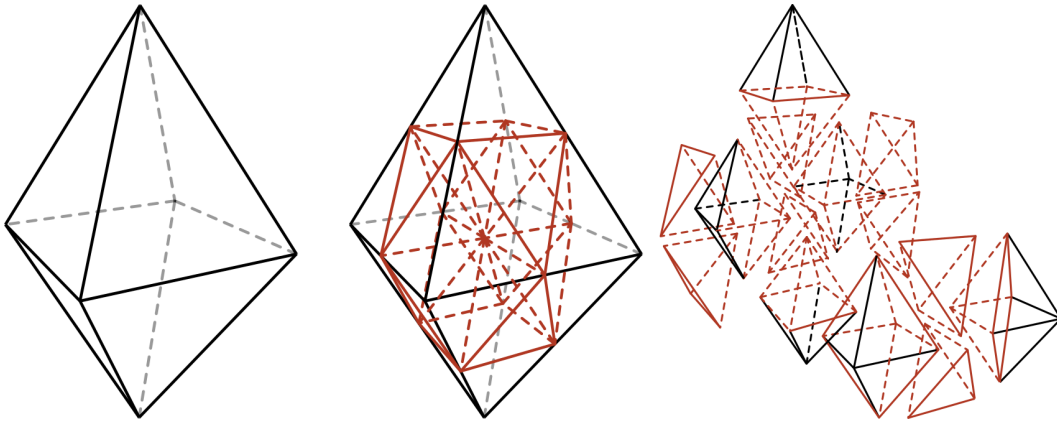


FIGURE 1.21 : Raffinement d'un octaèdre.

Dans ces travaux, le domaine est partitionné en utilisant une décomposition par des cellules sommets, arêtes et éléments (tétraédriques ou octaédriques). Chaque niveau de résolution est construit comme une nouvelle partition de ce domaine. Ainsi, le niveau de résolution  $G_k$  d'une partition  $T$  est défini tel que  $G_0$  contient tous les éléments de la partition  $T_0$  tandis que  $G_k$  contient tous les éléments  $T_k \setminus T_{k-1}$ , c'est-à-dire tous les nouveaux sommets, nouvelles arêtes et nouveaux éléments qui ne sont pas dans  $G_{k-1}$ . Ainsi, tous les éléments dans  $G_k$  sont assignés au niveau  $k$ . Pour finir, une grille à plusieurs niveaux représente toute cette hiérarchie de partitions.

Une hiérarchie régulière s'obtient en utilisant les raffinements que nous avons décrits dans le premier paragraphe. Il n'est pas possible de gérer simplement la création d'une hiérarchie adaptative car cette structure de données ne gère pas les éléments polyédriques. En effet, comme nous le verrons en détail au chapitre 3 le raffinement adaptatif engendre l'introduction de polyèdres quelconques. Leur technique, afin de gérer les éléments irréguliers produits par ce type de raffinement, est d'introduire la notion d'élément *virtuel*. Ces éléments *virtuels* n'existent pas dans la structure de données et doivent être créés, à la demande, si nécessaire. Dans le but de les reconnaître, les éléments parents de ceux-ci ont un état particulier.

Ainsi, l'algorithme de raffinement global est conduit par les différents états que peut prendre un élément. Ces états sont mémorisés à l'aide de drapeaux. Basé sur le concept d'éléments virtuels, un drapeau indique un des états suivants :

- Un élément est par défaut *régulier*.
- Il a l'état *irrégulier* si son fils est virtuel.
- Il a l'état *raffiné* si cet élément a été raffiné par une des règles du raffinement régulier.

- Enfin, il a l'état *marqué* pour indiquer que l'élément sera subdivisé de manière régulière.

L'algorithme raffine tous les éléments dont l'état est *marqué* pour le raffinement.

Enfin, pour garder une connectivité cohérente, deux éléments voisins de la hiérarchie ne peuvent différer que d'un niveau de raffinement au maximum.

### 1.3.6 Multi-Tessellation

La Multi-Tessellation (MT) est un modèle basé sur les maillages simpliciaux en dimension  $d$  [42]. Ce modèle étend le modèle de Multi-Triangulation [41, 103] aux dimensions 3 et supérieures. Il est indépendant de la dimension du complexe et de la stratégie de construction du modèle. Il est principalement utilisé dans l'approche « fin vers grossier ». La littérature à propos de ce modèle étant très riche, nous proposons ici un aperçu rapide du modèle et de quelques structures de données qui en sont issues.

Ce modèle est composé d'un maillage de base au niveau grossier associé à un ensemble de mises à jour. Ces mises à jour peuvent être appliquées localement au maillage afin de raffiner le maillage de base. Une mise à jour est constituée d'un ensemble de cellules (des tétraèdres dans le cas 3D) à retirer du maillage et d'un ensemble de cellules à insérer à la place de celles précédemment enlevées. L'application d'opérations sur le maillage induit un ordre partiel entre les mises à jour. En effet, une mise à jour  $u_1$  dépend d'une autre mise à jour  $u_2$  lorsque les tétraèdres introduits par  $u_2$  sont supprimés par  $u_1$ .

Une structure de données implémentant la Multi-Tessellation en 3D est proposée dans [39]. Les tétraèdres sont représentés comme des quadruplets d'index de sommets. Un DAG permet de conserver de manière générale l'ordre entre les mises à jour.

Dans [28], une structure de données dérivée pour les maillages tétraédriques irréguliers appelée Full-Edge Multi-Tessellation a été développée. Elle encode efficacement un modèle généré par des contractions « pleine » d'arêtes. La relation de dépendance directe est encodée grâce à une forêt d'arbres binaires de sommets, initialement proposée dans [48]. Considérons une opération de contraction pleine qui contracte une arête  $e = (v_a, v_b)$  en un sommet  $v$ . Un nœud de l'arbre correspond à une opération de contraction pleine et à son inverse l'opération d'éclatement de sommet. Il contient également un offset pour retrouver les positions de  $v_a$  et  $v_b$  à partir de  $v$  et vice versa et un masque

de bit pour partitionner l'ensemble de tétraèdres et identifier lesquels sont attachés au sommet  $v_a$  ou au sommet  $v_b$  après éclatement.

Dans [37], une structure dérivée pour les maillages tétraédriques irréguliers appelée Half-Edge Multi-Tessellation a été développée. Elle encode efficacement un modèle généré par des semi-contractions d'arêtes sous la forme d'un DAG. L'opération de semi-contraction d'une arête consiste à fusionner deux sommets  $(v_a, v_b)$  incidents à une arête  $e$  en un sommet  $v$  qui est soit le sommet  $v_a$ , soit le sommet  $v_b$ . La semi-contraction d'une arête  $(v_a, v_b)$  en un sommet  $v$  et son semi-éclatement de sommet sont encodés en stockant deux informations : les coordonnées du sommet  $v$  et un codage compact de la région d'influence de l'éclatement de sommet. Cette région correspond à une portion de l'étoile de  $v$  affectée par l'insertion du sommet  $v$ . Une structure plus efficace en coût mémoire a été proposée [38] dans laquelle la relation de dépendance directe est encodée implicitement à travers une extension de la forêt d'arbres binaires de sommets.

### 1.3.7 Pyramides de cartes

Les pyramides de cartes [12, 13, 14, 51, 50, 52] sont définies comme une extension du modèle de cartes combinatoires. Elles sont utilisées dans une approche « fin vers grossier ». Elles permettent de décrire une hiérarchie de subdivisions ainsi que les liens entre ces subdivisions afin de décrire un objet à différentes résolutions. Les applications ont trait à l'analyse et la segmentation multi-échelle d'images.

L'algorithme de segmentation d'une image fonctionne comme suit. D'abord, une carte combinatoire initiale est construite. Dans cette représentation, une face de la carte correspond à un pixel. Un nouveau niveau de la pyramide est construit à partir du niveau précédent par l'application des opérations de suppression ou de contraction de cellules. Ces opérations visent à fusionner des cellules voisines dans la carte afin de regrouper des régions d'intérêts suivant un critère donné. Ainsi, chaque niveau de la pyramide représente une segmentation de l'image. Pour être plus précis, cet algorithme contribue à former des régions de pixels de plus en plus larges au fur et à mesure de l'ajout de niveaux.

Les opérations utilisées pour construire des pyramides de cartes ont été définies au sein des cartes généralisées en dimension quelconque [36]. Leurs opérations inverses que sont, l'insertion et l'éclatement de cellules, ont également été définies en dimension quelconque [2].

Les opérations de suppression et de contraction sont à la base de la définition des pyramides de cartes généralisées [56, 57, 58]. La principale application de ces travaux concerne la segmentation multi-échelles d'images discrètes 3D. Dans cette application, chaque niveau est représenté par une 3-g-carte construite grâce à l'application des opérations de suppression ou de contraction sur la 3-g-carte de niveau précédent.

Ces travaux ont mené à différentes représentations, équivalentes en terme d'information topologique, du modèle des pyramides de cartes généralisées. Elles sont cependant à l'opposé en terme de complexité spatiale et temporelle. Une première représentation, dite « explicite », stocke séparément chaque n-g-carte correspondant à chacun des niveaux de la pyramide. Une seconde représentation, dite « implicite », stocke uniquement le niveau initial ainsi que les informations nécessaires à reconstruire les différents niveaux supérieurs. Enfin, la dernière représentation, dite « hiérarchique », stocke le niveau initial et pour chaque niveau les brins des cellules modifiées au niveau précédent.

## 1.4 Conclusion

La première section de ce chapitre a été l'occasion de présenter les modèles de base utilisés pour manipuler des maillages. Nous avons donné un aperçu rapide de structures de données ad hoc utilisées dans le cadre volumique. Ces structures sont basées sur une représentation spécifique aux opérations quelles sont sensées mettre en œuvre. Étant donné un objet, il est donc impossible d'appliquer des opérations différentes sans devoir développer des méthodes de conversion entre différentes représentations. Construire une représentation multirésolution en se basant sur ce type de structures de données reviendrait à développer autant de structures multirésolutions que de raffinements spécifiques.

À l'inverse, le modèle des cartes combinatoires sur lequel repose notre travail, bénéficie de la généralité et de la flexibilité d'un modèle mathématique bien défini. Ce cadre favorable est dû à plusieurs choses : d'une part, la représentation d'une entité unique simplifiant et généralisant l'écriture d'opérations modifiant la connectivité ; d'autre part, un nombre restreint et formalisé de contraintes à maintenir afin de garantir des propriétés du modèle sur les objets représentés. En utilisant les relations du modèle, il est possible de formuler toutes les requêtes de voisinage entre les différentes cellules représentées. Ainsi, même les requêtes non ordonnées comme les incidences par rapport à un sommet dans une 3-carte sont accessibles. Nous verrons que les algorithmes utilisant différentes cellules à des fins de calcul pourront tirer profit de cet



avantage.

Dans la seconde section de ce chapitre nous nous sommes intéressés aux modèles multirésolutions permettant de manipuler des maillages volumiques et de dimension supérieure. Une partie de ces modèles est basée sur la représentation d'un maillage à l'aide d'une structure monorésolution complétée d'un graphe ou d'un arbre représentant les dépendances entre un ensemble d'opérations de mise à jour.

Pour ce type de modèles, l'accès à un niveau de résolution, autre que celui représenté par le maillage initial, se construit par l'application successive d'opérations de mise à jour à la structure de données représentant le maillage courant. Lors de la mise à jour dynamique d'attributs associés aux cellules du maillage il faut parcourir l'ensemble des niveaux afin d'obtenir les cellules correspondantes. L'inconvénient est donc de devoir mettre à jour la topologie du maillage régulièrement. De plus, les différentes représentations implicites ou encore la non représentation de cellules à chaque niveau rend impossible le stockage d'attributs sur ces cellules de niveaux différents.

Enfin, les modèles sont souvent limités par une représentation sous-jacente où toutes les cellules ne sont pas représentées. Ceci réduit la subdivision de volumes à l'utilisation d'éléments cubiques ou rend impossible la gestion d'une hiérarchie adaptative de tétraèdres/octaèdres. Une solution alternative dans le cas de simplexe consiste à utiliser un raffinement par bisection d'arête mais cette opération centrée arête produit plus de cellules que nécessaires dans le cadre adaptatif. Cette absence de représentation de certaines cellules rend également impossible certaines requêtes de voisinage.

Les pyramides de cartes généralisées sont définies dans le cadre général des cartes généralisées permettant la représentation de quasi-variétés incluant les objets non orientables. Notre domaine d'études se limite aux objets orientables, nous avons donc choisi d'étendre les 2-cartes combinatoires multirésolutions [69]. Ceci nous permet d'obtenir une structure de données compétitive en coût mémoire. De plus, les opérations qui permettent la construction des niveaux de pyramides de g-cartes les rendent adaptées aux cadres de la manipulation d'images. Contrairement à ces travaux, nous voulons définir des opérations de subdivision plus générales et pouvant supporter une grande variété de schémas de subdivision.



---

# CARTES COMBINATOIRES MULTIRÉSOLUTIONS

---

## Sommaire

---

|       |   |           |
|-------|---|-----------|
| 2.1   | Introduction . . . . .                                      | <b>43</b> |
| 2.2   | Extension multirésolution en dimension quelconque . . . . . | <b>44</b> |
| 2.2.1 | Ensembles de brins . . . . .                                | 44        |
| 2.2.2 | Relations indexées . . . . .                                | 45        |
| 2.2.3 | Hypercartes multirésolutions . . . . .                      | 47        |
| 2.2.4 | Cartes combinatoires multirésolutions . . . . .             | 48        |
| 2.3   | Opérations de manipulation . . . . .                        | <b>50</b> |
| 2.3.1 | Opérations élémentaires . . . . .                           | 51        |
| 2.3.2 | Opérations de bas niveau . . . . .                          | 53        |

---

Dans ce chapitre notre objectif est de définir le modèle des cartes combinatoires multirésolutions en dimension  $n$ . Pour cela nous étendons la définition des 2-cartes combinatoires multirésolutions présentée dans [68].

## 2.1 Introduction

Les cartes combinatoires permettent de représenter des objets composés de cellules arbitraires. Elles garantissent un accès optimal à l'information topologique de la subdivision de l'espace représentée, quelque soit la dimension de l'objet. Gardant toutes ces propriétés, les cartes combinatoires multirésolutions de dimension  $n$  sont capables de représenter et manipuler efficacement un objet multirésolution quelque soit sa dimension.

Nous commençons par rappeler la définition des hypercartes multirésolu-

tions de dimension  $n$ . Cette définition nous permet d'exprimer le formalisme multirésolution dans le cadre d'un modèle indépendant de la dimension et du type d'objet représentable (orientable ou non, ouvert ou fermé, variété ou non). Grâce à cette définition des hypercartes multirésolutions de dimension  $n$ , nous pouvons ensuite écrire la définition des cartes combinatoires multirésolutions en dimension  $n$ . Cet héritage s'obtient en ajoutant à la définition des hypercartes multirésolutions les contraintes spécifiques au modèle des cartes combinatoires.

Par la suite, nous détaillons un certain nombre d'opérations nécessaires à la construction et à la manipulation de cartes combinatoires multirésolutions. Le cadre applicatif de ce travail se situe principalement en dimension 3 et donc, afin de gagner en efficacité, nous avons donc pris le parti d'exprimer et d'implanter nos opérations en dimension 3. Ceci étant, ces opérateurs peuvent être définis en dimension  $n$ .

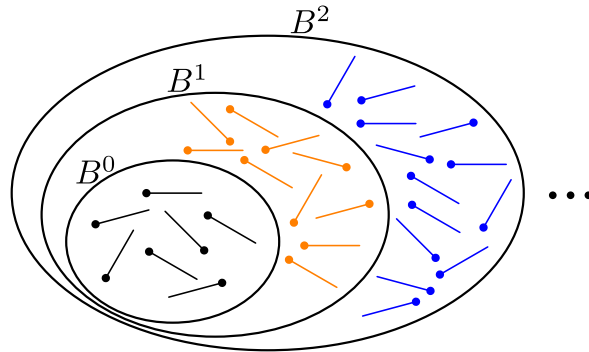
## 2.2 Extension multirésolution en dimension quelconque

### 2.2.1 Ensembles de brins

D'un point de vue très général, un modèle multirésolution permet de décrire une séquence de maillages représentant un objet avec différents niveaux de détails, généralement ordonnés du plus grossier au plus fin. Suivant le sens de construction, des cellules sont ajoutées ou supprimées entre chacun des maillages de la séquence par l'application d'opérateurs (subdivision, éclatement de sommet, contraction d'arête). L'application de ces opérations entraîne des modifications locales des liaisons topologiques entre les brins et donc entre les cellules et leurs relations d'adjacences et d'incidences.

Transposer cette idée aux hypercartes consiste à avoir une hypercarte multirésolution initiale à laquelle des brins sont successivement ajoutés ou détruits. Nous obtenons ainsi une séquence d'hypercartes débutant par une hypercarte grossière ayant peu de brins et finissant par une hypercarte fine ayant plus de brins.

Dans le cadre monorésolution classique, un unique ensemble de brins est utilisé. Les brins et leurs relations permettent de décrire une subdivision de l'espace donné. À l'inverse, une hypercarte multirésolution doit permettre de représenter plusieurs subdivisions de l'espace imbriquées les unes dans les autres et obtenues par un processus de construction. Les brins doivent donc être ar-



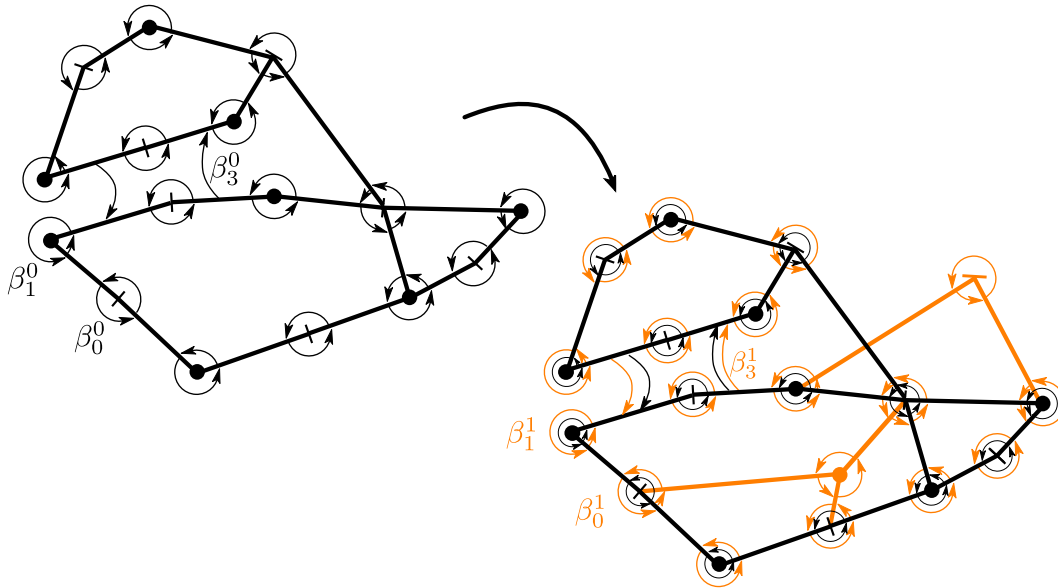
**FIGURE 2.1** : Ensembles de brins imbriqués dans une hypercarte multirésolution. Les brins noirs, oranges et bleus forment respectivement les ensembles  $N^0$ ,  $N^1$  et  $N^2$  (image issue de [68]).

rangés sous la forme d'ensembles imbriqués les uns dans les autres. Les relations topologiques liant ces brins peuvent rester inchangées ou être modifiées afin de former de nouvelles cellules. Cependant, les brins non modifiés sont conservés d'un niveau à l'autre. Plus formellement, chaque niveau de résolution  $k$  correspond à un ensemble de brins  $B^k$  tel que  $B^0 \subset B^1 \subset B^2 \subset \dots \subset B^l$ , où  $l$  est le niveau de résolution maximum. Lorsqu'un brin est inséré dans un niveau de résolution  $k$ , il fait parti intégrante des ensembles de brins décrivant toutes les hypercartes des niveaux supérieurs. Notons que  $B^0$  est l'ensemble des brins du niveau grossier et  $B^l$  l'ensemble des brins du niveau fin.

Cette relation d'imbrication des ensembles de brins est illustrée par la figure 2.1. Nous verrons au chapitre 4 comment ce principe d'imbrication des ensembles de brins nous permet de construire plusieurs représentations du même modèle qui se situent de part et d'autre sur l'échelle coût mémoire / coût en temps. Pour pouvoir représenter sans redondance cette hiérarchie de brins, nous avons besoin de définir  $N^k$  l'ensemble des brins ajoutés au niveau  $k$ . Pour tout  $k > 0$ , nous définissons  $N^k = B^k \setminus B^{k-1}$ . On peut alors définir les ensembles  $B^k$  de la manière suivant :  $B^k = \bigcup_{0 \leq m \leq k} N^m$ . Ainsi, un brin inséré au niveau  $k$  appartient à un seul ensemble  $N^k$  mais à tous les ensembles  $B^i$ , tels que  $k \leq i \leq l$

## 2.2.2 Relations indexées

Nous venons de définir différents ensembles de brins dans lesquels, les brins sont liés entre eux par des relations topologiques. Nous avons indiqué précédemment que lors du processus de construction de ces ensembles de brins les relations entre brins peuvent être modifiées. Ceci implique de pouvoir repré-



**FIGURE 2.2 :** Indexation des relations dans une hypercarte de dimension 3 ayant deux niveaux de résolution. Les relations de niveau 0 (en noir) sont recopiées au niveau 1 (en orange). La figure de droite illustre les trois types de relations possibles : les relations restant inchangées entre le niveau 0 et 1 composées uniquement de brins de niveau 0, les nouvelles relations introduites au niveau 1 composées uniquement de brins de niveau 1 et les relations modifiées du niveau 0 au niveau 1 composées à la fois de brins de niveau 0 et de brins de niveau 1.

sender différentes relations  $\beta_i$  pour chaque brin et chaque niveau de résolution. Par exemple, le niveau 1 est constitué des brins de l'ensemble  $B^1$ , composé des brins de  $N^0$  et de ceux de  $N^1$ . Il faut que les brins de l'ensemble  $N^1$ , que l'on vient d'ajouter, puissent s'insérer dans des permutations qui existaient déjà au niveau 0 entre les brins de  $N^0$ . Dans le cas contraire, les brins ajoutés à chaque niveau de résolution seraient totalement déconnectés des brins pré-existants dans les niveaux inférieurs. De plus, nous ne voulons pas perdre les informations topologiques qui existaient entre les brins dans les niveaux inférieurs. Ceci afin de pouvoir exploiter chaque niveau de manière indépendante.

La solution consiste à définir un ensemble de relations  $\beta_i$  pour chaque niveau de résolution  $k$ . On note ainsi  $\beta_i^k$ , la permutation  $\beta_i$  agissant sur les brins de l'ensemble  $B^k$ . Par exemple, la  $i$ -relation  $\beta_i^0$  lie les brins de  $B^0$  entre eux. Au niveau 1, la  $i$ -relation  $\beta_i^1$  lie les nouveaux brins de  $B^1$  entre eux ou modifie les liaisons  $\beta_i^0$  des brins déjà présents dans  $B^0$ . Nous illustrons sur la figure 2.2 les relations indexées dans une hypercarte de dimension 3 ayant deux niveaux consécutifs.

Plus généralement un brin doit disposer d'une relation différente par la

permutation  $\beta_i$  pour chaque niveau où il existe, c'est à dire pour chaque niveau supérieur ou égal au niveau auquel il a été inséré dans l'hypercarte. En d'autres termes, un brin de  $N^0$  doit porter les relations  $\beta_i^0, \dots, \beta_i^l$ , un brin de  $N^m$  doit porter les relations  $\beta_i^m, \dots, \beta_i^l$  et un brin de  $N^l$  porte uniquement la relation  $\beta_i^l$ .

### 2.2.3 Hypercartes multirésolutions

Nous définissons formellement une hypercarte multirésolution de dimension  $n$  comme un  $(n + 1)$ -uplet

$$H_{MR} = (\{B^k\}_{k \geq 0}, \{\beta_0^k\}_{k \geq 0}, \dots, \{\beta_{n-1}^k\}_{k \geq 0}) \quad (2.1)$$

où les  $B^k$  sont des ensembles de brins imbriqués les uns dans les autres, c'est-à-dire que tel que  $B^0 \subset B^1 \subset \dots \subset B^l$ , et pour tout  $k$  les  $\beta_i^k$  avec  $i \in 0, \dots, n$  sont des permutations sur les  $B^k$ .

Pour tout  $k \geq 0$ , le  $(n + 1)$ -uplet

$$H^k = (B^k, \beta_0^k, \dots, \beta_{n-1}^k) \quad (2.2)$$

est une hypercarte qui décrit l'objet au niveau de résolution  $k$ .

Dans l'état actuel, la définition n'impose aucune contrainte de construction sur les permutations d'un niveau ni sur leur conservation d'un niveau au suivant. Cette souplesse implique qu'il n'y pas forcément de cohérence entre les hypercartes des différents niveaux de résolution. Ainsi les relations existant entre les brins de  $N^0$  par la permutation  $\beta_i^0$  et les relations entre ces brins au niveau suivant par la permutation  $\beta_i^1$  peuvent être totalement différentes. Par extension, cette souplesse vaut également pour les cellules définies à chaque niveau. En particulier, il n'y a pas de contrainte de relation d'inclusion entre les cellules des différents niveaux.

Cependant, afin de construire des hypercartes cohérentes et donc valides, deux contraintes doivent être introduites dans le modèle. Rappelons que nous n'avons pas encore parlé d'attributs (géométrique, etc.) associés aux cellules par l'intermédiaire des brins. Les contraintes que nous allons introduire sont donc uniquement des contraintes topologiques liées à la construction des niveaux de résolution.

La première contrainte impose à l'hypercarte de niveau  $k$  de contenir un nombre de brins supérieur ou égal au nombre de brins présents dans l'hypercarte de niveau  $k - 1$ . Cette contrainte est liée à notre idée de départ

qui consiste à construire une séquence d'hypercartes dont le nombre de brins grossi à chaque niveau de résolution. Cette contrainte permet de construire une séquence cohérente d'objets sans pour autant faire d'hypothèse sur leur représentation. Un objet peut ainsi être totalement différent d'un niveau à l'autre.

La seconde contrainte impose que le  $(n+1)$ -uplet  $H^k$  défini à chaque niveau de résolution est bien une hypercarte conforme à la définition que l'on en a donné dans le chapitre précédent. Les opérations manipulant la topologie pour construire les différents niveaux doivent au minimum garantir l'intégrité de l'hypercarte correspondante à chaque niveau.

Cette définition et ces contraintes suffisent à construire une séquence d'hypercartes cohérente par niveau sans introduire de liens logiques entre les niveaux. La cohérence entre les niveaux sera introduite et maintenue par les opérateurs topologiques servant à la construction de la séquence multirésolution. En effet, pour être totalement générique, la cohérence n'est pas fixée au niveau du modèle mais est introduite par les différents modes de construction qui sont dépendants des domaines d'applications.

#### 2.2.4 Cartes combinatoires multirésolutions

A partir de la définition des hypercartes, il est possible de retrouver les modèles exprimables dans ce cadre en ajoutant les contraintes spécifiques à chaque modèle. Nous déduisons donc de l'extension multirésolution des hypercartes les cartes combinatoires multirésolutions de dimension  $n$ .

**Définition 7.** *Une carte combinatoire primale multirésolution de dimension  $n$  est une hypercarte multirésolution :*

$$C_{MR} = (\{B^k\}_{k \geq 0}, \{\alpha_0^k\}_{k \geq 0}, \{\alpha_1^k\}_{k \geq 0}, \dots, \{\alpha_{n-1}^k\}_{k \geq 0}) \quad (2.3)$$

telle que pour tout  $k \geq 0$ , le  $(n+1)$ -uplet

$$C^k = (B^k, \alpha_0^k, \alpha_1^k, \dots, \alpha_{n-1}^k) \quad (2.4)$$

est une carte combinatoire primale de dimension  $n$ . C'est-à-dire que les  $\alpha_i^k$  sont des involutions pour  $i$  allant de  $0$  à  $n-2$  et vérifient de plus la contrainte qui fait de  $\alpha_j^k \circ \alpha_i^k$  une involution sur  $B^k$  pour tout  $i < n-2$  et pour tout  $j$  tel que  $i+1 < j \leq n-1$ .

Comme pour les cartes combinatoires monorésolutions, nous pouvons exprimer la version duale d'une  $n$ -carte combinatoire multirésolution.



**Définition 8.** Une carte combinatoire duale multirésolution de dimension  $n$  est une hypercarte multirésolution

$$C_{MR} = (\{B^k\}_{k \geq 0}, \{\phi_1^k\}_{k \geq 0}, \{\phi_2^k\}_{k \geq 0}, \dots, \{\phi_n^k\}_{k \geq 0}) \quad (2.5)$$

telle que pour tout  $k \geq 0$ , le  $(n+1)$ -uplet

$$C^k = (B^k, \phi_1^k, \phi_2^k, \dots, \phi_n^k) \quad (2.6)$$

est une carte combinatoire duale de dimension  $n$ . C'est-à-dire que les  $\phi_i^k$  sont des involutions pour  $i$  allant de 2 à  $n$  et vérifient de plus la contrainte telle que pour tous les entiers  $i, j, 1 \leq i < i+2 \leq j \leq n$ , la relation  $\phi_i^k \circ \phi_j^k$  soit une involution sur  $B^k$ .

La contrainte de validité d'un niveau introduite pour les hypercartes multirésolutions est reprise ici. Ainsi, en forçant chacun des  $(n+1)$ -uplet  $C^k$  à être une  $n$ -carte valide, elle garantit que chacun des niveaux de résolution représentés est bien la subdivision en cellule d'une variété de dimension  $n$  orientable. Il faut ensuite veiller à ce que les opérations agissant sur une  $n$ -carte multirésolution respectent cette contrainte d'intégrité.

A chaque niveau de résolution  $k$ , les cellules de niveau  $k$  sont définies par des ensembles de brins de  $B^k$  formés par les orbites des  $\alpha_i^k$  ou  $\phi_i^k$ . Les orbites utilisées sont les mêmes que celles définies au chapitre 1 sur les  $n$ -cartes. La seule différence est l'introduction d'un paramètre  $k$  correspondant au niveau de résolution examiné.

Considérons donc une carte de dimension  $n$  et de niveau  $k$ , et définissons le parcours des cellules composées d'ensembles de brins de  $B^k$ . Dans le cas primal, le parcours de toutes les  $i$ -cellules avec  $0 \leq i \leq n-1$  s'exprime tel que :  $\langle \alpha_0^k, \dots, \alpha_{i-1}^k, \alpha_{i+1}^k, \dots, \alpha_{n-1}^k \rangle$ . Le parcours d'une  $n$ -cellule, quant à lui, s'exprime tel que :  $\langle \alpha_1^k \circ \alpha_0^k, \dots, \alpha_{n-1}^k \circ \alpha_0^k \rangle$ . Ainsi, dans le cas d'une 3-carte primale, les cellules sont parcourues grâce aux orbites suivantes :  $\langle \alpha_1^k, \alpha_2^k \rangle$  pour les sommets (0-cellules),  $\langle \alpha_0^k, \alpha_2^k \rangle$  pour les arêtes (1-cellules),  $\langle \alpha_0^k, \alpha_1^k \rangle$  pour les faces (2-cellules) et  $\langle \alpha_1^k \circ \alpha_0^k, \alpha_2^k \circ \alpha_0^k \rangle$  pour les volumes (3-cellules).

Pour le cas dual, on note d'abord  $\phi_0 = \phi_1^{-1}$  (également noté  $\phi_{-1}$ ). L'orbite sommet s'exprime comme :  $\langle \phi_1^k \circ \phi_2^k, \dots, \phi_1^k \circ \phi_n^k \rangle$ . Ensuite, l'orbite permettant de parcourir les brins d'une  $i$ -cellule est :  $\langle \phi_1^k, \dots, \phi_{i-1}^k, \phi_{i+1}^k, \dots, \phi_n^k \rangle$ , avec  $i \geq 1$ . Dans le cas d'une 3-carte duale, les sommets sont composés de l'ensemble des brins qui peuvent être atteints par la composition quelconque de permutations parmi  $\langle \phi_1^k \circ \phi_2^k, \phi_1^k \circ \phi_3^k \rangle$ , les arêtes par  $\langle \phi_2^k, \phi_3^k \rangle$ , les faces par  $\langle \phi_1^k, \phi_3^k \rangle$  et les volumes par  $\langle \phi_1^k, \phi_2^k \rangle$ .

## 2.3 Opérations de manipulation

Nous avons choisi d'utiliser et d'implanter la version duale du modèle des cartes combinatoires. Cette version du modèle permet une approche constructive, et donc plus intuitive, notamment pour monter en dimension dans la définition des opérations. En effet, une carte combinatoire duale peut être vue comme un assemblage de cellules de différentes dimensions. Ainsi, une 1-carte est un ensemble de cycles fermés. Une 2-carte est un ensemble de faces fermés composé de faces cousues à des faces adjacentes le long d'arêtes incidentes communes. Une 3-carte est un assemblage de surfaces fermés composé de surfaces cousues à des surfaces adjacentes le long de faces incidentes communes.

Les opérations topologiques sont donc décrites et illustrées avec leur version duale. Des versions primales existent, certaines sont implantées, mais elles ne sont pas décrites plus en détail dans cette section. Cependant nous basculerons les illustrations de certaines opérations en primal car elles expriment un sens différent qui sera utilisé dans le chapitre suivant.

Nous venons de définir formellement le modèle des cartes combinatoires multirésolutions en dimension  $n$ . Nous verrons dans le chapitre 4 que la structure de données sous-jacente supporte les cartes multirésolutions en dimension quelconque. Cependant, nous nous limitons à l'implantation efficace d'opérations en dimension 1, 2 et 3 permis par la définition constructive du modèle. Ainsi, les opérations sont définies par dimension de carte. Ceci nous permet de décrire des opérations en nous servant des opérations définies dans les dimensions inférieures. L'avantage est de pouvoir définir des opérations de manière récursive sur la dimension.

Nous allons donc introduire un certain nombre d'opérations qui manipulent la topologie de 3-cartes. Nous verrons qu'elles sont également utilisées pour manipuler une 3-carte multirésolution. Nous nous fixons comme contrainte, que chaque opération réalisée de manière atomique, vérifie les contraintes d'intégrité des cartes afin de laisser une carte cohérente après son application.

Nous nous fixons également comme contrainte de minimiser la complexité de chaque opération. Pour cela, nous définissons des opérations locales modifiant seulement les brins nécessaires à l'opération. Cette contrainte est d'autant plus simple à respecter lorsque l'on manipule une seule entité, à savoir le brin, plutôt que de mettre à jour plusieurs types de cellules.

Nous nous attarderons sur la partie topologique des opérations afin de bien comprendre les changements qui s'opèrent dans les relations topologiques entre

les brins. Il y a bien évidemment toute une gestion des plongements de cellules lors de l'application de ces opérations. Cependant, nous n'aborderons pas ces aspects dans cette section. Il faut simplement retenir que les cellules anciennes et nouvellement créées, après les transformations topologiques, sont toutes correctement plongées. Ceci signifie que tous les brins d'une orbite pointent correctement sur le même plongement, sans redondance ou recopie. Là encore, l'atomicité des opérations permet une gestion efficace en ne modifiant que les plongements nécessaires.

### 2.3.1 Opérations élémentaires

Nous commençons par détailler les opérations élémentaires concernant la gestion de la couture des brins par les différentes relations du modèle. Nous nous intéresserons également à l'insertion et à la suppression de brins dans une carte combinatoire. Nous commençons par décrire ces opérations dans une 3-carte monorésolution, avant de voir comment les adapter au sein d'une 3-carte multirésolution.

#### Carte monorésolution

Les brins d'une 3-carte peuvent être cousus par 3 relations différentes, la permutation  $\phi_1$  et les involutions  $\phi_2$  et  $\phi_3$ . Nous dirons qu'un brin qui n'est pas cousu à un autre brin par une relation  $\phi_n$  (avec  $1 \leq n \leq 3$ ) est en point fixe par la relation  $\phi_n$ .

Dans une 1-carte, il n'y a pas d'opérations de couture et découture par la relation  $\phi_1$ . Nous rappelons le principe des deux opérations élémentaires, illustrées figure 2.3a, qui permettent de fusionner et de séparer deux cycles. Ces deux opérations sont importantes car elles permettent de s'assurer que le cycle fermé formé par les brins de la permutation est encore un cycle fermé après leur application. Les opérations présentées plus avant utilisent le cas particulier où l'un des deux cycles est formé par un seul brin. Dans ce cas, ces opérations permettent d'ajouter ou d'enlever un brin dans une face. Nous nous donnons également une opération de création d'un brin de 1-carte. Cette opération crée un brin et le lie à lui même par la relation  $\phi_1$  formant ainsi un cycle de longueur un.

Dans une 2-carte, l'opération de couture de deux brins par  $\phi_2$  prend deux brins  $d$  et  $e$  et les lie ensemble de manière symétrique. L'opération inverse

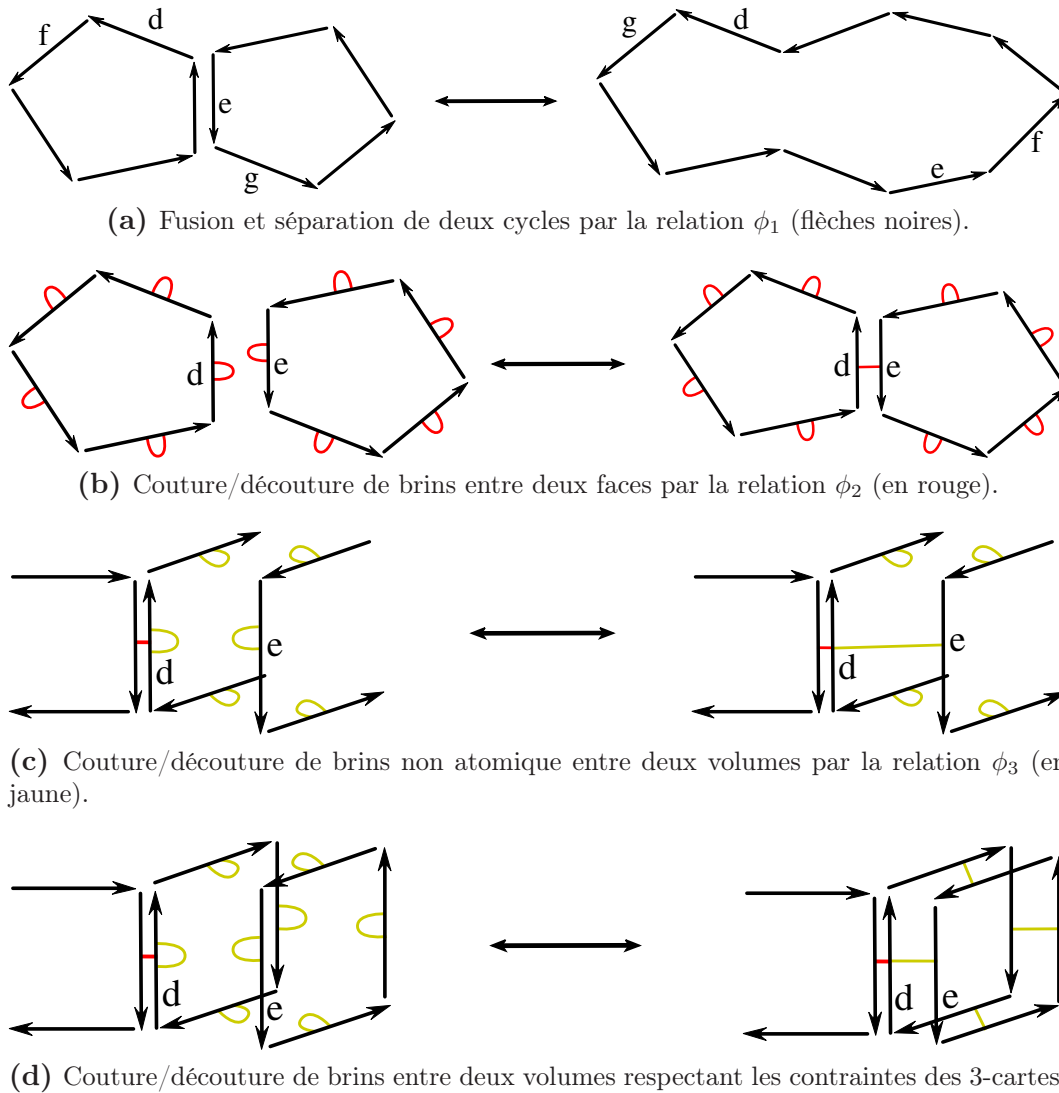


FIGURE 2.3 : Opérations élémentaires duales de couture et découpe.

effectue la découpe des deux brins. Ces deux opérations sont illustrées par la figure 2.3b. Nous nous donnons également une opération de création d'un brin de 2-carte. Cette opération utilise l'opération de création d'un brin de 1-carte et met ce nouveau brin en point fixe par la relation  $\phi_2$ .

Dans une 3-carte, l'opération de couture de deux brins par  $\phi_3$  prend deux brins  $d$  et  $e$  et les lie ensemble de manière symétrique. L'opération inverse effectue la découpe des deux brins. Ces deux opérations sont illustrées par la figure 2.3c. Cependant, cette opération rend la 3-carte invalide car il faut respecter une autre contrainte sur les 3-cartes impliquant que les brins de deux cycles distincts soient tous liés entre eux par la relation  $\phi_3$ . Ainsi, nous rappelons le principe des deux opérations respectant cette contrainte permettant

la couture et la découture de deux volumes le long de cycles ayant le même nombre de brins comme illustrées par la figure 2.3d. Toutes les opérations de bas niveau que nous détaillerons plus loin utilisent ces deux opérations de couture et découture de volumes. Toutefois, lors d'ajouts ou suppressions locaux de brins, comme nous le verrons par la suite, les opérations de couture et découture de deux brins par  $\phi_3$  restent disponibles.

## Carte multirésolution

Nous avons défini précédemment que dans une carte combinatoire multirésolution les relations entre les brins sont définies à chaque niveau de résolution. La construction des niveaux de résolution consiste d'abord à dupliquer les brins et leurs relations du niveau  $k$  au niveau  $k+1$ . Dans le cas d'une 3-carte multirésolution, chaque carte  $C^k$  dispose d'une permutation  $\phi_1$  et de deux involutions  $\phi_2$  et  $\phi_3$  qui lient les brins de l'ensemble  $B^k$ . Afin de modifier ce nouveau niveau, il faut ensuite se placer au niveau  $k+1$  dans lequel des brins de niveau  $k+1$  peuvent être ajoutés en utilisant les opérations élémentaires présentées précédemment. Ainsi, les opérations élémentaires de couture et découture dans le cadre multirésolution sont les mêmes que celles que l'on vient de décrire pour les 3-cartes monorésolutions. Ces opérations agissent sur les brins présents au niveau  $k+1$ , donc introduits dans un niveau inférieur ou égal à  $k+1$ .

Ces opérations élémentaires nous permettent de construire des opérations de bas niveau effectuant de petites opérations atomiques modifiant principalement une seule cellule (c'est-à-dire un sous-ensemble restreint de brins). Nous verrons plus tard comment ces opérations de bas niveau nous permettent de fabriquer des algorithmes efficaces et génériques pour la construction de niveaux de résolution.

### 2.3.2 Opérations de bas niveau

Nous allons maintenant illustrer un certain nombre d'opérations de bas niveau nécessaires par la suite à la création d'algorithmes plus avancés et dont une spécification algébrique est donnée par [21]. D'abord, nous expliquons leur fonctionnement dans une 3-carte monorésolution. Puis, nous montrons comment ces opérations sont utilisées dans le cadre multirésolution. Nous expliquons notamment comment il est possible d'appliquer ces opérations entre deux niveaux de résolution différents afin de lier ces niveaux ensemble en liant des brins des différents niveaux.

## Découpe d'arête

La découpe d'une arête consiste à couper cette arête en deux morceaux en y insérant un nouveau sommet de valence 2. Le fait d'encoder la représentation duale, nous permet de profiter d'une vue constructive et d'écrire l'opération pour chaque dimension de carte combinatoire. Ces opérations sont appelées insertion et suppression de sommet dans une arête lorsqu'elles sont vues dans la représentation primale.

Nous commençons par rappeler le principe de l'opération de découpe d'arête pour une 1-carte illustrée figure 2.4. Un nouveau brin  $e$  est créé puis introduit dans le cycle de  $d$ .

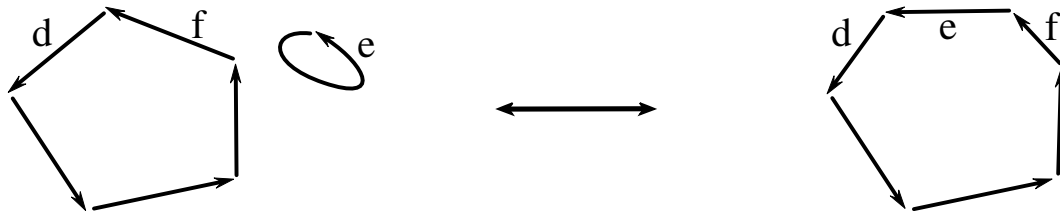


FIGURE 2.4 : Découpe et découpe inverse d'une arête - Insertion et suppression d'un sommet dans une arête en dimension 1.

Nous rappelons ensuite l'opération de découpe d'arête pour une 2-carte illustrée figure 2.5. L'opération de la 1-carte est appelée pour chaque face de 2-carte autour de l'arête de la 2-carte. Dans ce cas, deux nouveaux brins  $d_2$  et  $e_2$  sont introduits dans la carte. Ils sont insérés dans les permutations  $\phi_1$  de chaque cycle de brins autour de l'arête concernée par l'opération, c'est à dire les brins  $d$  et  $e$ . Les involutions  $\phi_2$  sont ensuite mises à jour afin de lier les anciens brins aux nouveaux brins pour former la nouvelle arête de 2-carte.

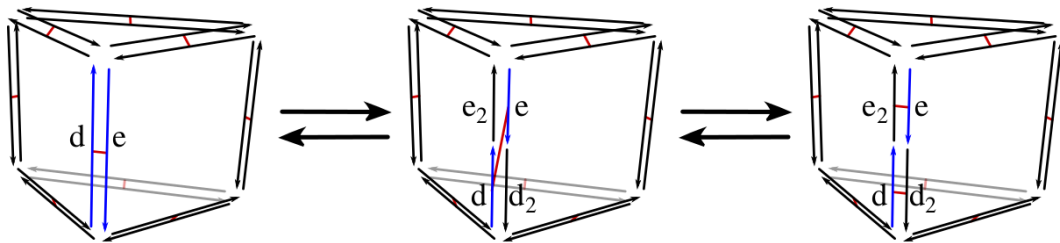


FIGURE 2.5 : Découpe et découpe inverse d'une arête - Insertion et suppression d'un sommet dans une arête en dimension 2.

Cette opération de 2-carte est ensuite appelée pour chaque couple de brins de 2-carte autour de l'arête dans la 3-carte comme illustrée figure 2.6. Les involutions par  $\phi_3$  sont mises à jour au fur et à mesure de la traversée de l'arête.

Sur cette figure, tous les volumes autour de l'arête ne sont pas représentés à des fins de lisibilité.

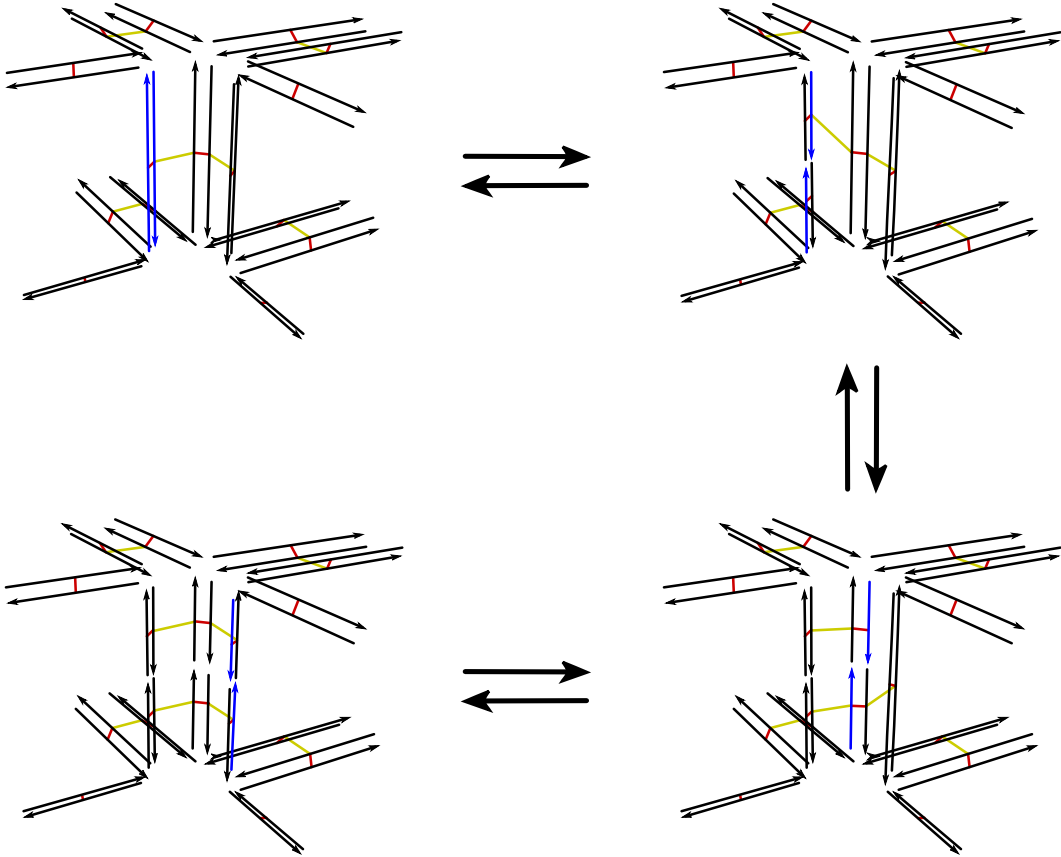


FIGURE 2.6 : Découpe et découpe inverse d'une arête en dimension 3.

Les modifications des relations décrites ci-dessus sont effectuées en parcourant une seule fois l'orbite arête dans une 3-carte. La complexité de cette opération est donc linéaire en nombre de brins de l'arête.

L'opération inverse, également illustrée figure 2.5 et figure 2.6, consiste à supprimer un sommet de valence 2 du maillage. Là encore, nous utilisons l'opération inverse de la 2-carte qui consiste d'abord à supprimer les brins  $d_2$  et  $e_2$  en utilisant l'opération de la 1-carte de suppression d'un brin d'un cycle, puis à recoudre les brins  $d$  et  $e$  entre eux par  $\phi_2$  mais également aux successeurs de  $d_2$  et  $e_2$  par  $\phi_1$ . Cette opération est appliquée à chaque morceau de 2-carte autour de l'arête. Parallèlement, il faut à nouveau recoudre correctement par  $\phi_3$  les brins des volumes voisins. Étant l'opération inverse de la découpe d'arête, cette opération a également une complexité linéaire en nombre de brins de l'arête.

Ces deux opérations se représentent dans le formalisme primal et nous les

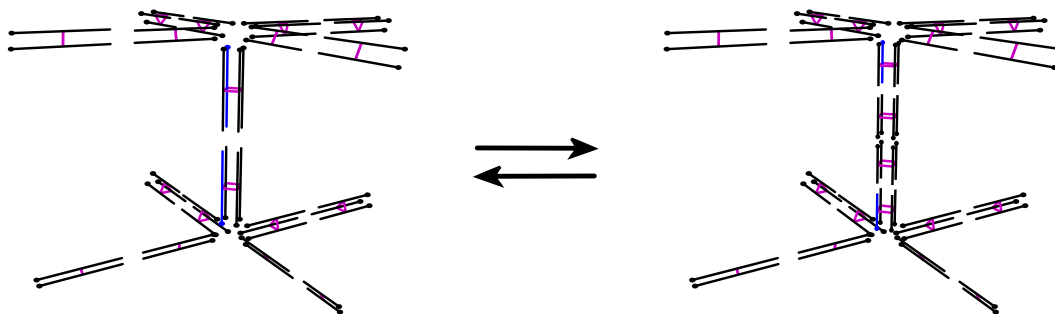


FIGURE 2.7 : Insertion et suppression d'un sommet dans une arête en dimension 3.

illustrons par la figure 2.7.

### Découpe de face

La découpe d'une face en deux faces revient à insérer une arête entre deux sommets de cette face. Nous illustrons ici cette opération appliquée entre deux brins d'une face à quatre côtés.

Rappelons qu'une face de 3-carte est constituée de deux faces orientées de 2-carte cousus le long de leurs bords par la relation  $\phi_3$ . Suivant notre démarche constructive, nous appliquons l'opération de découpe de face d'une 2-carte sur chaque face orientée. L'opération de découpe de face d'une 2-carte, illustrée figure 2.8, utilise l'opération de fusion de cycle de 1-carte qui permet d'ajouter un brin entre deux autres brins dans un cycle en fusionnant un cycle constitué d'un brin avec un autre cycle (illustrée figure 2.4). Deux nouveaux brins sont ajoutés, entre les brins  $d$  et  $e_0$ , avant le brin  $e$ , ainsi que entre  $e$  et  $d_0$ , avant le brin  $d$ . Ces deux nouveaux brins sont ensuite cousus par  $\phi_2$ . Les nouveaux brins d'arêtes créés pour chaque face orientée sont ensuite recousus entre eux par la relation  $\phi_3$  comme illustré figure 2.9.

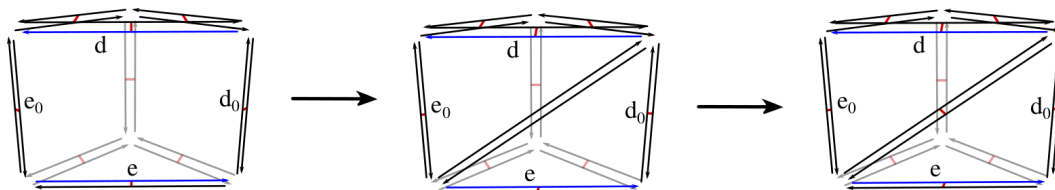


FIGURE 2.8 : Découpe de face - Insertion d'une arête dans une face en dimension 2.

Cette opération met en œuvre quatre brins de la face initiale quelque soit le degré de cette face. La complexité de cette opération est donc constante.



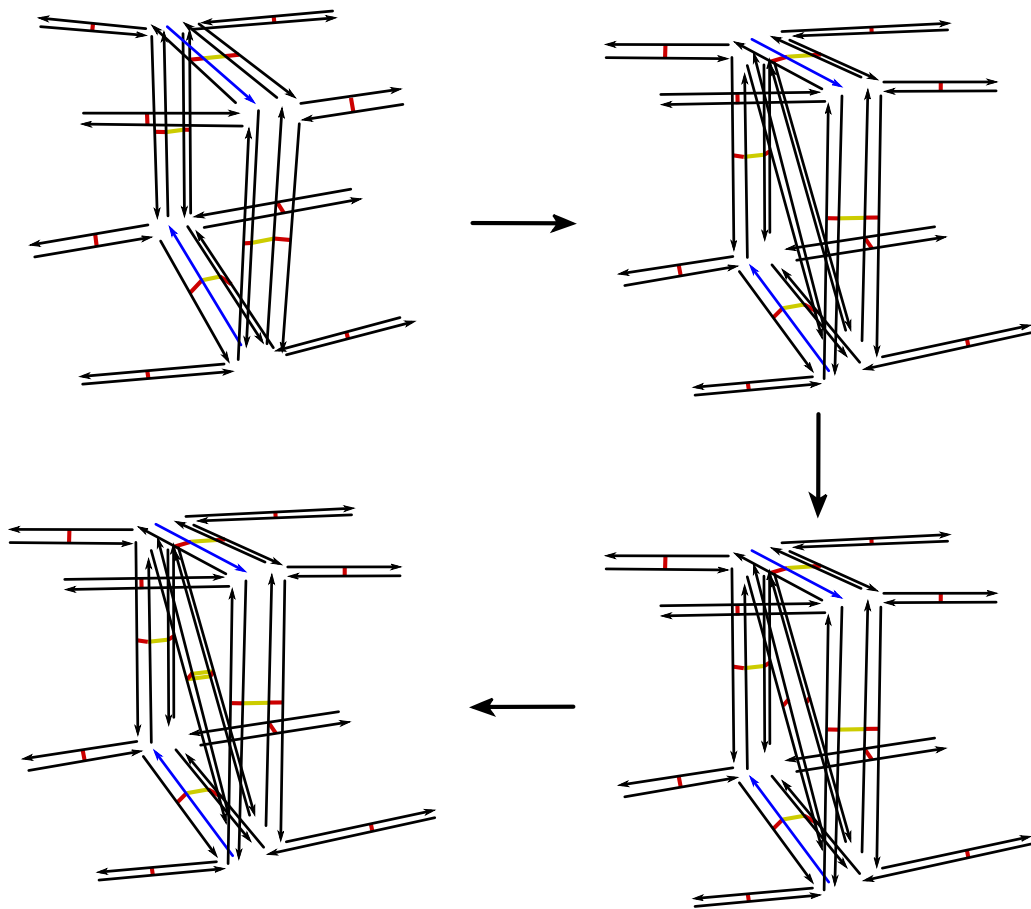


FIGURE 2.9 : Découpe de face en dimension 3.

Cette opération se représente dans le formalisme primal et nous l'illustrons par la figure 2.10.

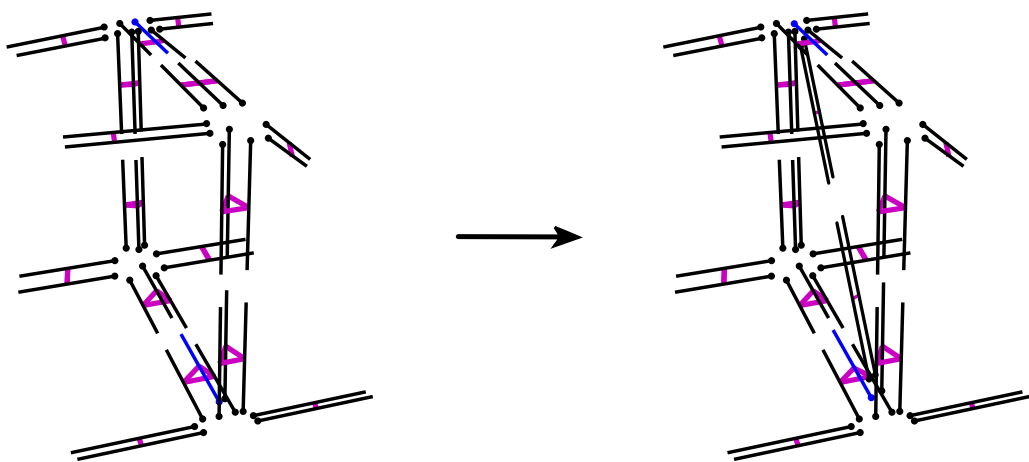


FIGURE 2.10 : Insertion d'une arête dans une face en dimension 3.

## Découpe de volume

La découpe d'un volume en deux volumes revient à insérer une face le long d'un chemin d'arête de ce volume. La figure 2.11 illustre cette opération appliquée à un volume le long des brins marqués en bleu.

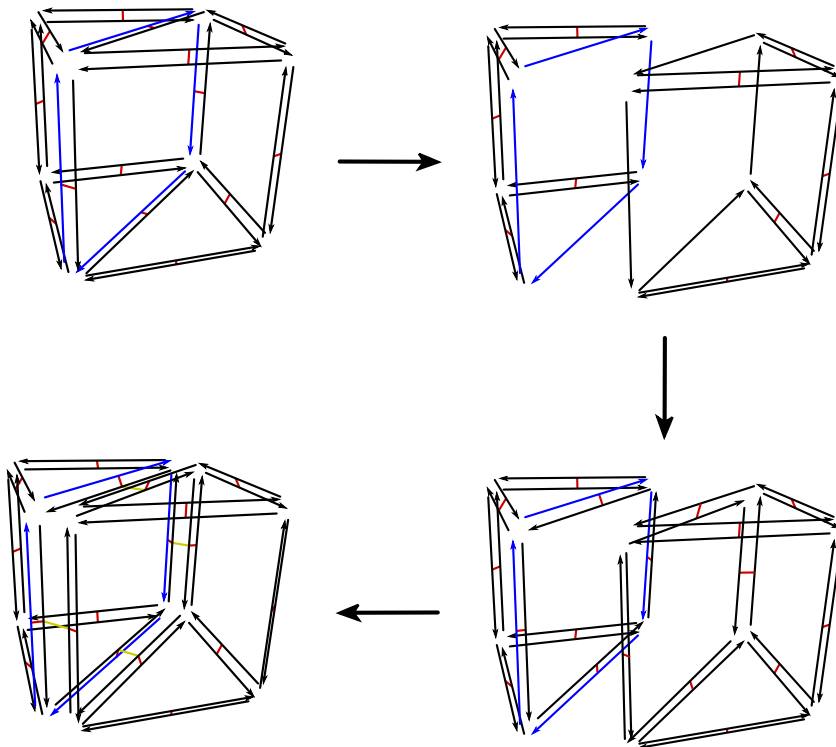


FIGURE 2.11 : Découpe d'un volume.

La découpe d'un volume consiste en la découpe d'une 3-carte le long d'un chemin d'arête sur cette 3-carte. Nous utilisons donc ici une opération de 2-carte qui découd les arêtes par  $\phi_2$  en itérant sur les brins du chemin. Nous créons ensuite deux faces de 2-carte dont le degré est égal à la longueur du chemin d'arête. Ces deux faces sont cousues de part et d'autre du chemin que l'on vient de découdre. Enfin, l'opération de la 3-carte coud les deux faces orientées entre elles en utilisant l'opération de couture de volumes.

Cette opération met en œuvre  $n$  brins du chemin d'arête du volume initial nécessaire à la découpe. La complexité de cette opération est donc linéaire en la taille du chemin de découpe.

Cette opération se représente dans le formalisme primal et nous l'illustrons par la figure 2.12.

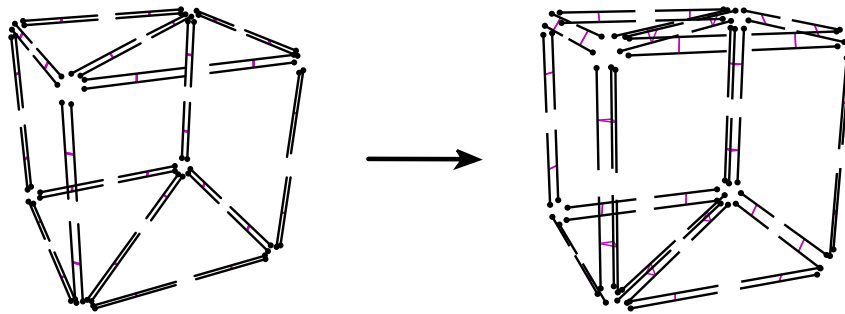


FIGURE 2.12 : Insertion d'une face dans un volume.

## Fusion de volumes

La fusion de deux volumes revient à supprimer la face commune à ces deux volumes. La figure 2.13 illustre cette opération appliquée à deux volumes cousus par une face à quatre côtés.

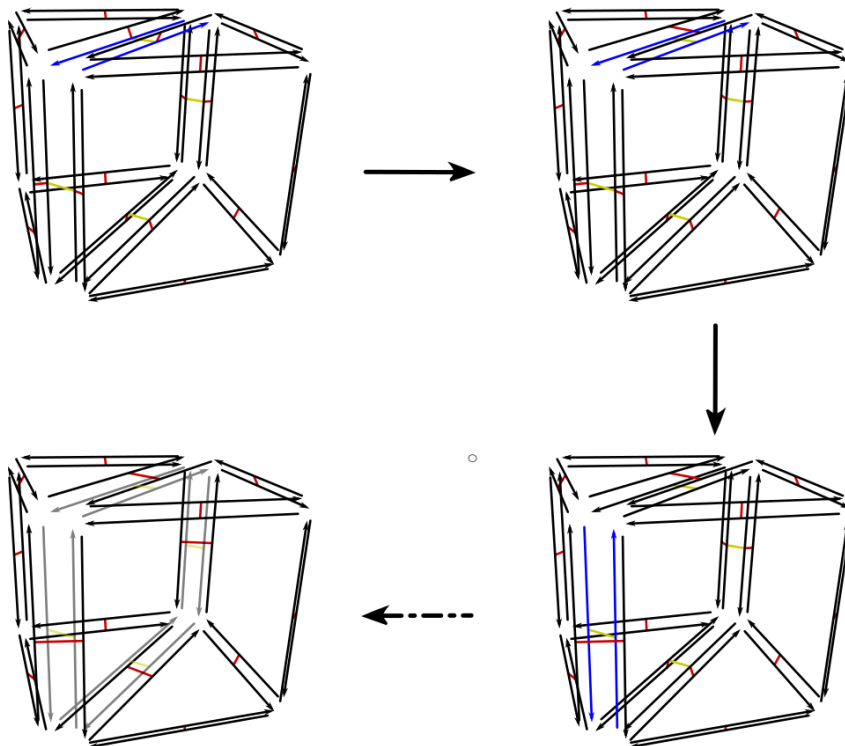


FIGURE 2.13 : Fusion de deux volumes le long de leur face commune.

La fusion de deux volumes consiste à fusionner deux 2-cartes le long de deux faces (reliées ici par la  $\phi_3$ ). Nous utilisons donc ici une opération de 2-carte qui consiste à traverser dans une même itération les arêtes des deux faces, la première face dans le sens trigonométrique par la relation  $\phi_1$  et la seconde face

dans le sens anti-trigonométrique par la relation  $\phi_{-1}$ . Pour chaque arête, il faut découper par  $\phi_2$  la face à supprimer puis coudre par  $\phi_2$  les brins des volumes laissés libre par cette involution comme illustré par la figure 2.13. Les deux cycles sont ensuite soit supprimés, soit gardés (comme dans notre exemple en gris clair).

Cette opération met en œuvre  $n$  arêtes de la face comprise entre les deux volumes. La complexité de cette opération est linéaire en la taille de la face.

Cette opération se représente dans le formalisme primal et nous l'illustrons par la figure 2.14.

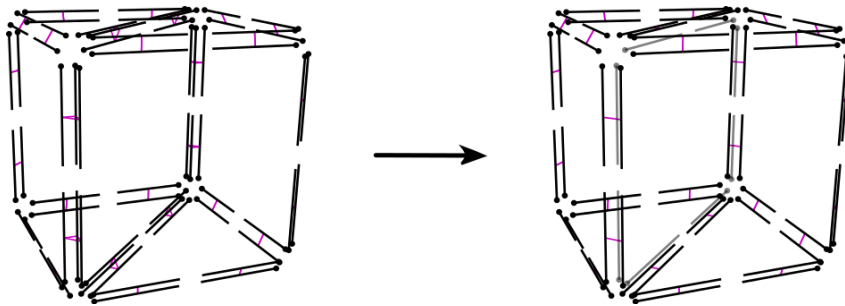


FIGURE 2.14 : Fusion de deux volumes le long de leur face commune en primal.

## Opérations multirésolutions

La construction d'un nouveau niveau de résolution  $k + 1$  s'effectue en copiant le niveau de résolution  $k$ . Remarquons que les opérations de bas niveau sont des compositions d'opérations élémentaires agissant sur des brins présents au niveau  $k + 1$ . L'application des opérations de bas niveau dans le cadre multirésolution est une généralisation de l'application des opérations élémentaires dans le cadre multirésolution.

Nous détaillerons dans le chapitre 3 l'enchaînement de ces opérations entre deux niveaux de résolution dans le cadre de maillages multirésolutions construits à l'aide d'algorithmes de subdivision.

DEUXIÈME PARTIE

---

CONTRIBUTIONS



---

# VOLUMES DE SUBDIVISION MULTIRÉSOLUTIONS

---

## Sommaire

---

|       |   |            |
|-------|---|------------|
| 3.1   | Introduction . . . . .  | <b>64</b>  |
| 3.1.1 | Volumes de subdivision . . . . .                                    | 64         |
| 3.1.2 | Applications des volumes de subdivision . . . . .                   | 67         |
| 3.1.3 | Volumes de subdivision multirésolutions . . . . .                   | 69         |
| 3.1.4 | Support du maillage multirésolution . . . . .                       | 71         |
| 3.2   | Génération des niveaux par raffinement primal . . . . .             | <b>72</b>  |
| 3.2.1 | Génération régulière . . . . .                                      | 74         |
| 3.2.2 | Subdivision d'un polyèdre arbitraire . . . . .                      | 83         |
| 3.2.3 | Génération adaptative . . . . .                                     | 87         |
| 3.2.4 | Simplification . . . . .  | 92         |
| 3.2.5 | Niveau de subdivision d'une cellule . . . . .                       | 94         |
| 3.3   | Mise à jour des attributs de sommet : cas de la géométrie . . . . . | <b>96</b>  |
| 3.3.1 | Schéma de subdivision . . . . .                                     | 97         |
| 3.3.2 | Schéma multirésolution . . . . .                                    | 102        |
| 3.4   | Génération adaptative de maillage pour la simulation . . . . .      | <b>106</b> |
| 3.5   | Autre raffinement : $\sqrt{3}$ volumique . . . . .                  | <b>109</b> |
| 3.5.1 | Génération régulière . . . . .                                      | 111        |
| 3.5.2 | Génération adaptative . . . . .                                     | 117        |
| 3.5.3 | Géométrie . . . . .   | 119        |
| 3.6   | Conclusion . . . . .  | <b>122</b> |

---

Ce chapitre présente les capacités de notre modèle multirésolution dans le cadre d'une hiérarchie obtenue dans le sens « grossier vers fin ». Nous illustrons notre propos en dimension 3 pour des volumes de subdivision.

Nous commençons par introduire le principe de la subdivision de maillages et différentes applications des volumes de subdivision. Nous détaillons ensuite

la manière dont les cartes combinatoires permettent de décrire efficacement les changements topologiques au sein d'objets de dimension 3 ou supérieur. Elles offrent ainsi un support efficace pour la représentation et la manipulation de tels objets. Nous finissons en exposant quelques expérimentations.

## 3.1 Introduction

Notre principal objectif est de montrer que nous pouvons offrir un support robuste, efficace et générique aux volumes de subdivision, et plus largement à tous maillages issus d'un processus de subdivision. Nous commençons ainsi par donner l'intuition de la construction d'objets par un processus de subdivision. Nous n'aborderons pas les propriétés mathématiques à la base de la construction de tels objets qui ne sont pas au cœur de nos travaux.

Les volumes de subdivision, et plus largement tous maillages de subdivision, établissent un lien entre les objets représentés de façon continue par des fonctions et ceux représentés sous la forme d'une discrétisation en cellules. Cette méthode permet de représenter l'infinité des points d'une surface, d'un volume, d'un polytope par la limite d'un processus de raffinement partant d'une discrétisation initiale.

Les recherches dans le domaine de la subdivision ont d'abord et avant tout été motivées par l'industrie de l'animation afin de remplacer les modèles paramétriques dont le contrôle est rendu plus difficile notamment par la gestion de la connexion inter-carreaux. Une des propriétés des techniques à base de subdivision est qu'elles permettent de modéliser des objets ayant une topologie quelconque. Par topologie quelconque, nous entendons des objets dont le genre et la discrétisation est arbitraire, mais qui restent des variétés (ou quasi-variétés).

### 3.1.1 Volumes de subdivision

La subdivision est un processus itératif de construction d'un maillage à partir d'un polytope ou d'un maillage initial appelé maillage de contrôle. Ce processus agit sur le maillage en insérant de nouveaux sommets et en modifiant les sommets existants dans le maillage de contrôle du modèle cible. Chaque schéma s'applique sur un type particulier de maillage de contrôle et converge vers un modèle paramétrique (courbe, surface, volume) limite ayant des propriétés particulières. Lorsque le processus termine, on obtient un maillage de



subdivision visuellement plus « fin ».

Les surfaces de subdivision ont été introduites pour la première fois en 1978 par Catmull et Clark [20] et par Doo et Sabin [44]. Elles sont la généralisation des travaux de Chaikin [25] sur le schéma de subdivision de courbes. Depuis, de nombreux schémas ont été développés. On peut citer le schéma de subdivision de Loop [83], Butterfly [45], le schéma de subdivision primal/dual [120], ou le schéma de subdivision  $\sqrt{3}$  [67], etc.

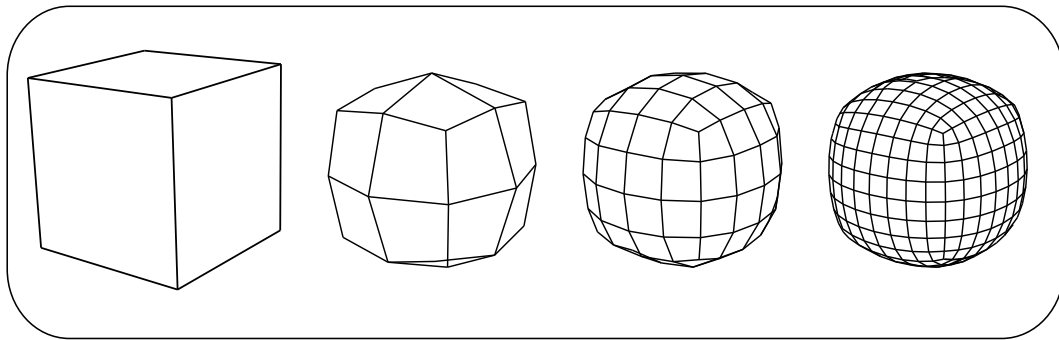


FIGURE 3.1 : Illustration du schéma de subdivision surfacique de Catmull-Clark.

A l'inverse, les travaux sur les volumes de subdivision sont apparus bien plus tard. Les pionniers sont MacCracken et Joy [86] qui généralisent le schéma de Catmull et Clark aux maillages polyédriques. Puis, au début des années 2000, une vague de travaux sur les schémas de subdivision volumique apparaît. Les travaux de MacCracken et Joy sont reformulés et améliorés [4]. Apparaît également une généralisation du schéma du Loop aux tétraèdres [26], puis aux tétraèdres/octaèdres [106]. On retrouve une généralisation du schéma Butterfly aux maillages hexaédriques [88], mais également un schéma interpolant pour des maillages tétraédriques [27]. Plus récemment, une adaptation du schéma  $\sqrt{3}$  aux maillages tétraédriques [15] a été proposée.

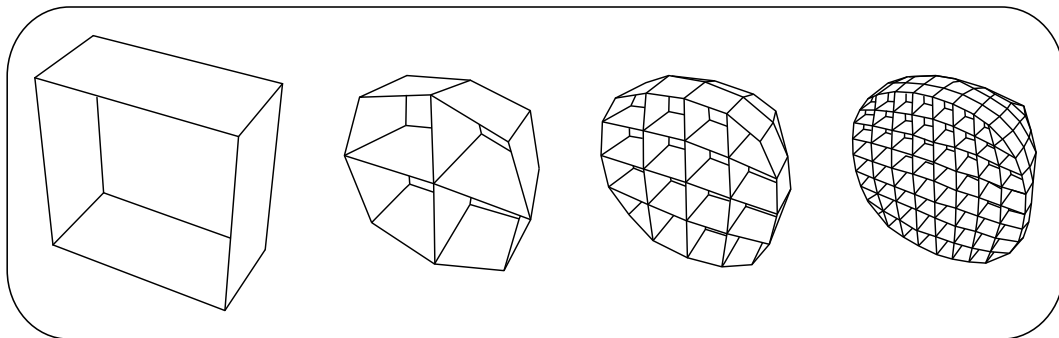


FIGURE 3.2 : Illustration du schéma de subdivision volumique de MacCracken-Joy.

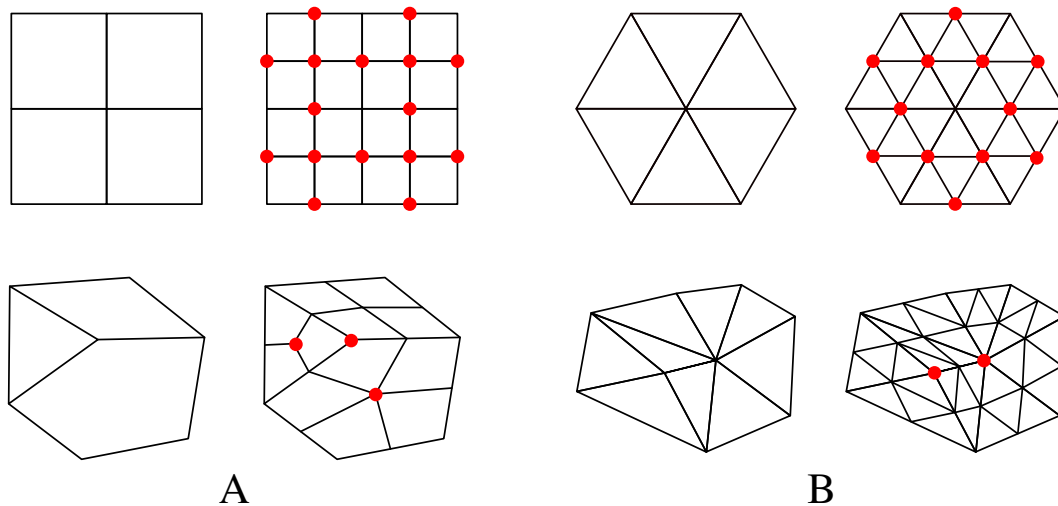
Quelques travaux se sont intéressés à la subdivision en dimension quel-

conque. On peut citer les travaux de Pascucci [99, 98] qui étendent le schéma primal/dual en dimension 3, puis en dimension quelconque.

Les algorithmes de subdivision, quelque soit la dimension pour laquelle ils s'appliquent, peuvent être découpés en une composante topologique et une composante géométrique. La composante géométrique consistant à appliquer un schéma géométrique aux sommets peut, elle-même, être découpée en deux catégories : les schémas approximants et les schémas interpolants.

Dans le cas d'un schéma approximant, les sommets du maillage initial ne sont pas sur la surface ou le volume limite. Après chaque étape de subdivision, les sommets de l'étape précédente sont déplacés ou éventuellement supprimés, ce qui donne en général un meilleur aspect. L'utilisateur a cependant moins de contrôle sur le résultat final. Dans le cas d'un schéma interpolant, les sommets du maillage initial font partie de la surface ou du volume limite. Les nouveaux sommets peuvent se déplacer après chaque étape de subdivision et doivent rester sur la surface ou le volume limite.

La composante topologique, également appelée raffinement, modifie quant à elle la connectivité du maillage. Cette composante peut, elle-même, être découpée en deux catégories : les schémas primaux et les schémas duaux.



**FIGURE 3.3 :** Illustration d'une étape de subdivision primale d'une surface pour un raffinement quadrangulaire (A) et triangulaire (B). Les sommets introduits sont représentés en rouge sur la ligne du haut. Les sommets irréguliers sont représentés en rouge sur la ligne du bas.

## Schémas primaux

Pour une surface, chaque face du maillage initial est subdivisée en plusieurs faces. Les anciens sommets sont conservés. De nouveaux sommets sont insérés sur les arêtes du maillage subdivisé, et éventuellement à l'intérieur des faces du maillage subdivisé. De nouvelles arêtes liant ces sommets sont ajoutées. Pour finir, des faces sont ajoutées le long de ces arêtes. Le degré des faces générées dépend du raffinement utilisé (quadrangulaire ou triangulaire), comme illustré par la figure 3.3.

L'extension aux volumes s'effectue simplement. La procédure de subdivision d'une face est appliquée à toutes les faces du maillage, autrement dit aux différentes faces des polyèdres du maillage. Un nouveau sommet est ajouté à l'intérieur de chaque volume, ainsi que de nouvelles arêtes liant ce sommet aux sommets des faces et des arêtes du bord du polyèdre. Le type des polyèdres générés dépend du raffinement utilisé sur les faces du polyèdre.

## Schémas duaux

Pour une surface, une nouvelle face est créée pour chaque arête et pour chaque sommet. Les anciens sommets sont donc éclatés en autant de nouveaux sommets que de faces adjacentes à cet ancien sommet. L'éclatement d'une arête (ayant 2 faces adjacentes) crée une nouvelle face de degré 4. Une face créée dans un ancien sommet est de degré  $d$ , où  $d$  est le nombre d'arêtes incidentes à cet ancien sommet. Les anciennes faces principales sont maintenues sur le maillage subdivisé, comme illustré par la figure 3.4.

Là encore, l'extension aux volumes est simple à comprendre. Un nouveau volume est créé pour chaque arête et pour chaque sommet du maillage. Le volume créé dans une arête est homéomorphe à un cylindre dont la base est un polygone de degré  $n$ , où  $n$  est le degré de cette ancienne arête. Le volume créé dans un sommet est homéomorphe à une sphère dont le nombre de faces est de degré  $m$ , où  $m$  est le nombre d'arêtes incidentes à cet ancien sommet. Les anciens volumes principaux sont maintenus sur le maillage subdivisé.

### 3.1.2 Applications des volumes de subdivision

Les travaux sur les volumes de subdivision trouvent leurs origines dans diverses applications qui vont de la modélisation géométrique à l'analyse nu-

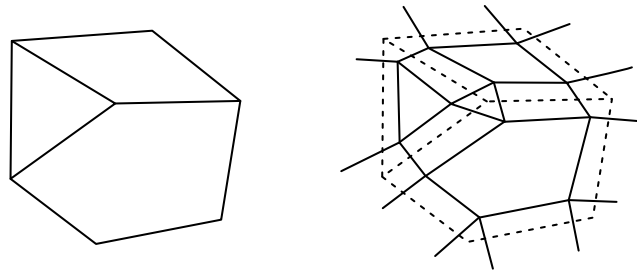


FIGURE 3.4 : Illustration d'une étape de subdivision duale d'une surface.

mérique.

La déformation de forme libre, ou encore FFD pour Free Form Deformation [107, 54], est un ensemble de techniques permettant de déformer un objet en appliquant une distorsion à l'espace environnant cet objet via un outil de contrôle de la déformation. La majeure partie des travaux sur les volumes de subdivision se placent dans ce contexte. Les volumes de subdivision permettent de libérer totalement la forme et la topologie de l'outil de contrôle fourni à l'utilisateur. Ils permettent de remplacer les modèles paramétriques pour lesquels il est difficile de maintenir la continuité lors de la déformation par le déplacement des points de contrôle.

L'idée directrice de ces techniques consiste à générer par des raffinements successifs une séquence de maillages indépendants les uns des autres. Chacun de ces maillages représente une partition de l'espace englobant l'objet à déformer. Les sommets de chacun des maillages de la séquence, du plus grossier au plus fin, servent de points de contrôle pour la déformation. La finesse du contrôle de la déformation de l'objet et l'étendue du rayon d'action de la déformation dépendent du maillage choisi dans la séquence, ou plus exactement du niveau d'édition.

Ces travaux ont donné lieu au développement d'outils de plus haut niveau. McDonnell et al. [92] présentèrent un outil de sculpture dynamique d'objets. Les objets sont plongés dans une cage construite à l'aide de volumes de subdivision. Un système masse-ressort permet de réduire le nombre de degrés de liberté de la cage. Ils développèrent également une version de leur outil utilisant une Méthode par Éléments Finis [91] (abrégé FEM pour Finite Element Method) pour maintenir la cohérence dans les déformations qu'ils améliorèrent par la suite [93]. McDonnell et al. ont également présenté un outil de sculpture de surfaces implicites [89]. Les données scalaires sont interpolées en utilisant le schéma de subdivision et différents outils de modélisation de haut-niveau sont développés.

Des travaux récents, notamment ceux de Hugues [63], tentent de réduire l'écart entre la Conception Assisté par Ordinateur (C.A.O) et la simulation numérique grâce à une technique appelée analyse iso-géométrique. Il s'agit d'utiliser des fonctions de base servant à définir la géométrie, par exemple la base de fonction des splines, pour approcher les solutions d'équations aux dérivées partielles. En comparaison de méthodes FEM classiques cette approche hérite des avantages de l'utilisation de ce type de fonctions en fournissant par exemple une représentation précise, voire exacte, des géométries mises en œuvre. Cependant, elle hérite également des inconvénients que nous avons abordé ci-dessus.

Plus récemment, des travaux ont entrepris de transposer ces concepts initialement conçus pour des objets représentés de manière continue aux volumes de subdivision. Il s'agit de tirer profit de la possibilité pour les volumes de subdivision de représenter des maillages volumiques non structurés ayant une topologie arbitraire. Ces méthodes consistent à utiliser la même base de fonctions pour la représentation de la géométrie issue de la subdivision et pour l'intégration des éléments pendant la simulation. La première approche est basée sur les maillages tétraédriques en utilisant une subdivision par un schéma  $\sqrt{3}$  volumique [16]. La deuxième approche est basée sur des maillages hexaédriques utilisant le schéma de subdivision de MacCracken et Joy [17].

Enfin, les volumes de subdivision ont également été exploités dans le cadre d'une technique de rendu volumique direct appelée « splatting » [90]. La technique du splatting consiste à combiner plusieurs textures entre elles en jouant notamment sur la transparence des différentes couches. L'objectif de cette approche est d'avoir un rendu efficace d'un maillage volumique utilisé par des outils de déformation sans avoir à manipuler un maillage volumique très fin qui serait gourmand en mémoire.

### 3.1.3 Volumes de subdivision multirésolutions

Le terme multirésolution recouvre plusieurs aspects suivant les domaines auxquels il se rapporte. Nous utilisons ici le terme multirésolution dans son sens large où l'on dira qu'un objet est étudié à plusieurs échelles (ou niveaux de résolutions).

Une première approche de la multirésolution est de considérer que seule l'information topologique est modifiée en générant une séquence de maillages emboîtés par raffinement et subdivision linéaire.

Cette technique est utilisée dans le cadre de la déformation de forme libre utilisant des cages volumiques où la construction d'une cage de forme et de topologie quelconque peut être laborieuse. On trouve ainsi des travaux proposant la génération d'une hiérarchie de cages s'adaptant à un objet par subdivision successive d'une octree [96].

Cette technique a également été utilisée dans le cadre de la génération de maillages hexaédriques [3] à partir de données voxels. L'algorithme développé dans ce travail utilise alternativement des phases de subdivision adaptative d'une grille puis de lissage utilisant un schéma de subdivision volumique afin que le bord du maillage tende vers une surface lisse qui approxime au mieux les données voxels.

Une approche plus fondamentale consiste à associer nos données discrètes au paradigme de l'analyse multirésolution. L'analyse multirésolution est une théorie d'analyse du signal qui permet de décomposer un signal en coefficients renfermant deux informations : une information spatiale qui sera pour nous un attribut associé à une cellule (par exemple la position géométrique) ainsi que la connectivité, et une information fréquentielle qui correspond à l'échelle d'influence du coefficient sur la courbe, la surface ou le volume. Les ondelettes ont trouvé des champs d'application très variés comme l'analyse numérique [29], le traitement du signal, le traitement d'image [87], la visualisation et l'informatique graphique [7].

D'ailleurs, l'analyse multirésolution de surfaces de subdivision est un outil très développé. Ce mécanisme a donné lieu à des outils de traitement sophistiqués qui permettent de faire de l'édition multirésolution [121, 119] de surface, de la compression de maillage [1], ou encore du filtrage de maillages bruités.

Dans le cas volumique, une approche utilisant la technique du Lifting Scheme [111], issu de la théorie des ondelettes, a été proposée par Bertram [5]. Cette technique apporte une bonne approximation des maillages aux niveaux plus grossiers et minimise les vecteurs de détails dans le cadre du filtrage de maillages bruités.

Dans le cadre de la déformation dynamique, Capell et al. [19] ont proposé une approche utilisant les volumes de subdivision comme support à une base hiérarchique de fonctions. Cette méthode permet d'adapter la simulation en fonction du contexte.

D'autres travaux concernent la représentation multirésolution d'une grille de voxels à l'aide d'un schéma  $\sqrt[3]{2}$  [81]. Cette technique permet de faire varier l'échantillonnage en fonction du point de vue en gardant une bonne approxi-

mation lors du sous-échantillonnage. Une technique similaire est utilisée pour des données voxels variant dans le temps [82].

### 3.1.4 Support du maillage multirésolution

Les outils d'analyse et de traitement multirésolution, populaire pour la manipulation de surfaces, ont du mal à percer dans le cadre volumique. Afin de se placer dans un cadre propice à l'avènement de ces techniques manipulant des volumes de subdivision multirésolutions, les maillages volumiques gagneraient à posséder un support efficace et générique pour la représentation multirésolution. Cependant, une telle représentation doit répondre à certaines exigences, similaires à celles qui sont utilisées pour les surfaces de subdivision multirésolutions.

Dans une application du type édition ou déformation multirésolution, l'utilisateur doit pouvoir accéder et manipuler aisément les cellules à chaque niveau intermédiaire. Le modèle doit donc permettre un accès efficace à chacun des maillages intermédiaires correspondant à ces niveaux.

Comme nous l'avons vu précédemment, la combinaison affine de sommets d'un voisinage donné permet de calculer la position d'un sommet de la subdivision. Dans le cas de l'analyse multirésolution, une mise à jour des positions est effectuée après chaque édition du maillage. Ce calcul, et plus largement le calcul associé à tout type d'attributs (propriétés de matériaux, etc.), doit donc se faire de manière efficace. Les informations de voisinage étant extrêmement sollicitées, le modèle nécessite donc de pouvoir effectuer des requêtes d'adjacence performantes.

Ce modèle doit également supporter la subdivision adaptative de maillage. Il faut pouvoir raffiner un volume à une profondeur variable selon un critère donné. Ceci permet d'économiser de la mémoire en terme d'informations topologiques et géométriques car cela évite de générer toutes les cellules et toutes les informations relatives à ces cellules. Il faut également pour cela un critère qui permette de dire si oui ou non la cellule doit être subdivisée.

La principale difficulté d'une telle représentation d'objets volumiques réside dans la combinatoire des objets. Le degré des sommets, des faces, des volumes augmente considérablement. Cette complexité entraîne un surcoût dans le développement d'algorithmes de subdivision. Afin de gagner en souplesse et en généricité, une solution consiste à exprimer des raffinements différents à l'aide d'un même modèle représentant des maillages quelconques. Ce modèle doit

pouvoir supporter plusieurs schémas de subdivision indépendamment de la structure de données sous-jacente.

Au vu de la complexité des objets volumiques, il devient donc difficile de gérer une telle représentation avec une structure ad hoc. Il est donc préférable de pouvoir décrire de manière formelle les transformations topologiques appliquées aux objets. Ce modèle doit donc être robuste pour la description d’algorithmes efficaces mais également pour l’enregistrement des informations d’adjacences entre cellules arbitraires (sommets, arêtes, faces, volumes) dans une hiérarchie volumique quelconque.

Les cartes combinatoires multirésolutions procurent le cadre nécessaire à la résolution de ces contraintes. Dans les sections suivantes, nous allons montrer comment nous pouvons formaliser les changements de combinatoire et mettre en évidence leurs apports dans le cadre d’une représentation multirésolution.

## 3.2 Génération des niveaux par raffinement primal

Une étape de raffinement consiste à ajouter un certain nombre de cellules à un maillage ainsi que leurs relations d’incidences. Les modèles « classiques », à base de graphe d’incidence, permettent de réaliser ces insertions sur la représentation même si, en pratique, leur implémentation reste inefficace (voir chapitre 4). Dans le cas des cartes combinatoires, il faut insérer de nouveaux brins pour représenter ces cellules et reconfigurer localement les relations ( $\phi_i$  ou  $\alpha_i$ ) des brins appartenant aux cellules impliquées dans le raffinement. Ces opérations nécessitent une analyse précise des relations et de l’ordre des cellules dans les orbites à reconstruire mais permettent de modifier un nombre minimal de relations pour rester efficace.

Détaillons d’abord comment les niveaux de résolution sont générés dans une 3-carte multirésolution pour un raffinement primal. Ce type de raffinement opère une subdivision des arêtes, des faces, et des volumes du maillage. La création d’un niveau de résolution peut se faire de manière régulière, ou adaptative, suivant un critère donné. Nous présentons d’abord trois raffinements de maillages volumiques (quadrisection de faces polygonales, triangulaires et mixtes) utilisant les opérations de bas niveaux présentées au chapitre 2. Nous esquissons ensuite la possibilité, permise grâce au formalisme des cartes combinatoires et à une récurrence sur la dimension des cellules, de formaliser le raffinement primal en dimension quelconque. Pour finir, nous présentons la subdivision adaptative et ses spécificités.



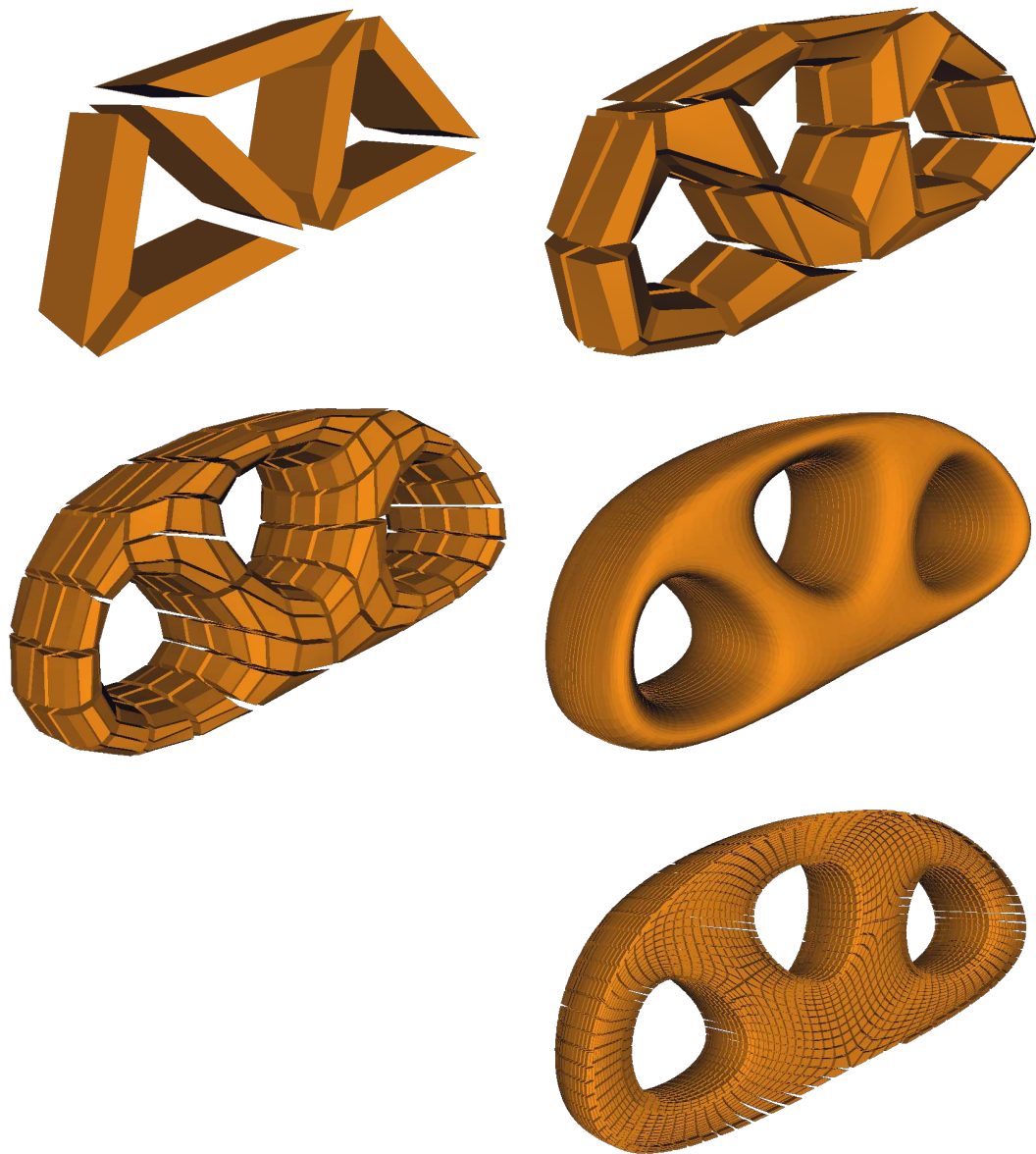


FIGURE 3.5 : Exemple de maillage obtenu par subdivision régulière.

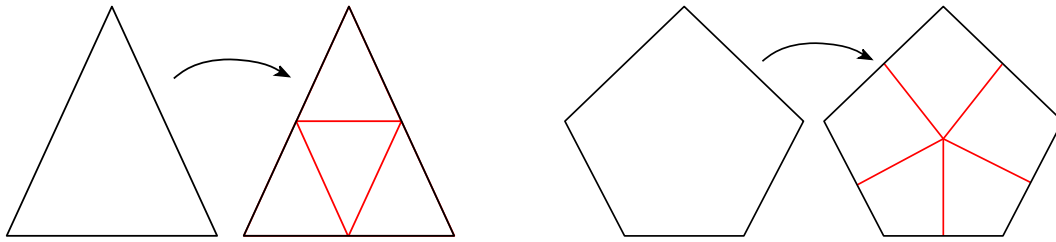


FIGURE 3.6 : Quadrisection de faces triangulaires et polygonales.

### 3.2.1 Génération régulière

La génération régulière des niveaux de résolution consiste à subdiviser toutes les cellules du maillage d'un niveau donné au niveau suivant. La figure 3.5 illustre une subdivision régulière d'un maillage polyédrique en utilisant le schéma multirésolution de Bertram. Le raffinement d'un polyèdre est lié au raffinement de ses faces. Nous distinguerons par la suite les raffinements de polyèdres en fonction du raffinement appliqué à leurs faces : la quadrisection de faces triangulaires ou la quadrisection de faces polygonales illustrées figure 3.6. Mis à part le premier exemple, et pour des raisons de clarté, nous illustrerons les algorithmes de génération régulière sur un seul polyèdre à la fois.

#### Raffinement par quadrisection de faces polygonales

Le premier raffinement opère sur des maillages polyédriques en effectuant une quadrisection de faces polygonales. L'image 3.7 illustre le raffinement d'un hexaèdre. D'abord, toutes les arêtes du maillages sont subdivisées. Ensuite, les faces sont subdivisées. Pour cela, un sommet est introduit au milieu de chacune d'elles. De nouvelles arêtes reliant ces sommets centraux aux sommets placés aux milieux des arêtes sont ensuite insérées. Pour finir, de nouvelles faces sont insérées dans les nouvelles arêtes introduites précédemment. Cette dernière opération permet de former les nouveaux polyèdres. Au final, on obtient un nouveau polyèdre par sommet de chaque ancien polyèdre.

Nous allons illustrer les étapes de notre méthode de raffinement par une quadrisection de faces entre deux niveaux consécutifs  $i$  et  $i + 1$  dans une 3-carte multirésolution sur un hexaèdre. La première étape consiste à créer un nouveau niveau  $i + 1$  comme une copie du niveau courant  $i$  et de considérer comme nouveau niveau courant ce niveau  $i + 1$ .

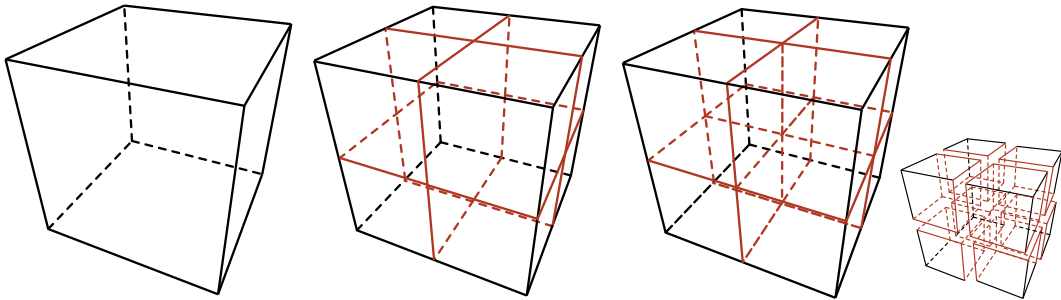


FIGURE 3.7 : Quadrisection d'un hexaèdre.

La figure 3.8 illustre le raffinement d'une face dans une 3-carte multirésolution. D'abord, toutes les arêtes du maillage sont coupées en deux morceaux en insérant de nouveaux sommets (figure 3.8-B). Cette étape consiste à appliquer sur toutes les arêtes l'opération de découpe d'arête décrite dans le chapitre 2. Pour chaque ancienne face, on utilise l'opération de découpe de face, décrite au chapitre 2, entre deux nouveaux sommets insérés issus de deux anciennes arêtes contiguës (figure 3.8-C). Puis, l'opération de découpe d'une arête est ensuite utilisée au niveau  $i + 1$  sur cette nouvelle arête afin d'insérer un sommet qui sera le sommet central de la subdivision de face (figure 3.8-D). Enfin, de nouvelles arêtes sont insérées entre ce sommet central et les autres sommets issus de la subdivision des arêtes de la face d'origine (figure 3.8-E puis 3.8-F).

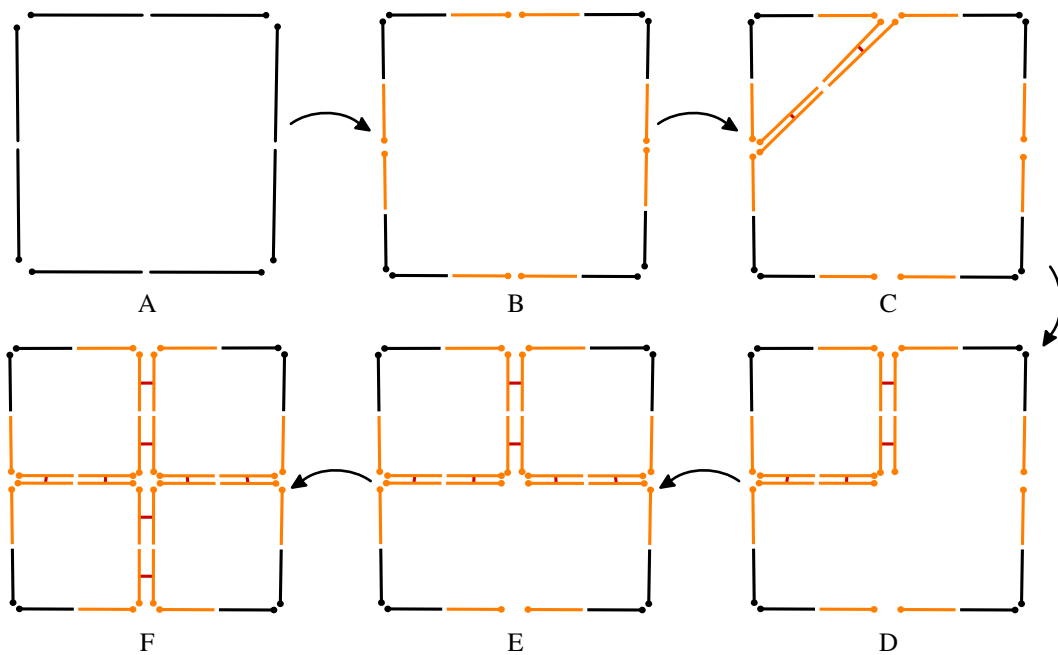


FIGURE 3.8 : Quadrisection de faces quadrangulaires au sein d'une 3-carte multirésolution.

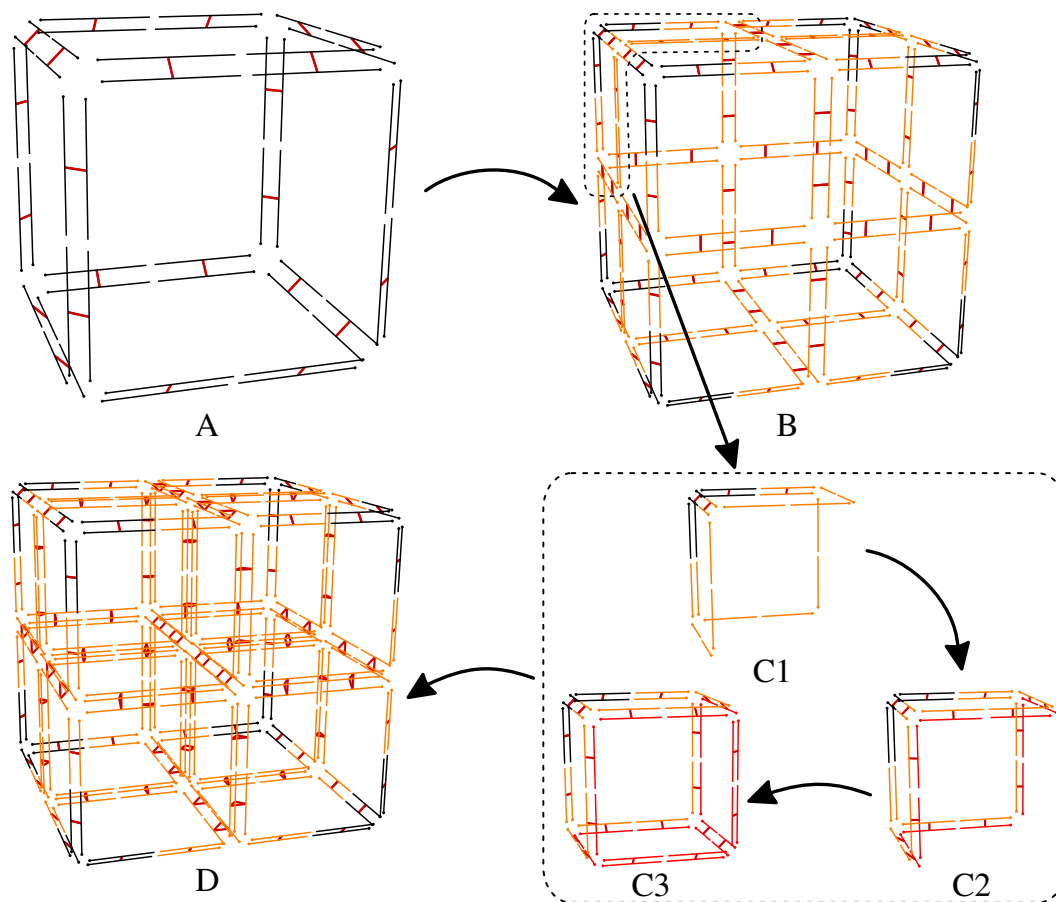


FIGURE 3.9 : Quadrissection d'hexaèdres au sein d'une 3-carte multirésolution.

La figure 3.9 illustre le raffinement d'un volume dans une 3-carte multirésolution. La première étape consiste à appliquer l'algorithme précédent à toutes les faces d'un volume (figure 3.9-B). Ensuite, l'opération de découpe de volume est utilisée afin de découper une ombrelle le long d'un chemin d'arête autour de chaque sommet de niveau  $i$  (figure 3.9-C1). Ce chemin d'arête est constitué des arêtes de niveau  $i + 1$  insérées lors du raffinement des faces. Cette opération crée une face, en rouge sur la figure 3.9-C2, dont le degré est égal à la longueur du chemin d'arête précédent. Notons que dans notre exemple cette face a 6 arêtes. Cette face est, à son tour, raffinée comme précédemment en ajoutant un sommet central (figure 3.9-C3). Pour finir, les arêtes de niveau  $i + 1$ , introduites par le raffinement des faces, sont utilisées pour coudre les volumes voisins, de niveau  $i + 1$ , le long de leurs faces voisines de niveau  $i + 1$ . Les sommets centraux de tous les coins d'un volume de niveau  $i$  subdivisés au niveau  $i + 1$  sont ainsi automatiquement fusionnés en un unique sommet central grâce à cette dernière opération (figure 3.9-D).

À l'inverse des surfaces où la quadrissection de faces produit toujours des

faces quadrangulaires, en volumique le raffinement de certains polyèdres ne produit pas que des hexaèdres. Une classification des polyèdres générés selon le degré de leurs sommets a été donnée par [86].

Le degré d'un sommet  $V$  d'un volume  $W$  est définie comme le nombre d'arêtes de  $W$  incidente à  $V$ . Considérons un sommet  $V$  d'un volume  $W$  de degré  $n$ . Le processus de raffinement décrit plus haut produit un nouveau volume  $W_1$  incident au sommet  $V$  :  $W_1$  contient  $2n$  faces de degré 4, 2 sommets de degré  $n$  et  $2n$  sommets de degré 3. Les volumes générés par une étape de raffinement sont appelés des volumes de type- $n$ . Nous illustrons des volumes de type- $n$  obtenus à partir de pyramides dans la figure 3.10. Prenons l'exemple d'un sommet  $s_0$  de degré 4 d'un volume  $w$ . Le processus de raffinement générera un volume ayant 8 faces de degré 4, 2 sommets de degré 4 et 8 sommets de degré 3. Un volume de ce type est illustré en trait noir sur la figure 3.10b. Après un pas de subdivision, tous les nouveaux volumes générés peuvent être classifiés en volume de type- $n$ .

Le raffinement d'un volume de type- $n$  produit deux volumes de type- $n$  et  $2n$  volumes de type-3. Le volume de type-3 est usuellement appelé un hexaèdre et possède des faces à 4 côtés et des sommets de degré 3. La répétition du processus de raffinement génère un arrangement de ce type de volumes excepté autour d'un nombre fini de volumes de type- $n$  cassant cette régularité et que nous illustrons sur la figure 3.10b. Ainsi, pour  $n > 3$  le nombre de volumes de type- $n$  double à chaque application de l'algorithme de raffinement.

L'utilisation des cartes combinatoires permet de gérer correctement l'apparition de ce type de polyèdre. Une manière d'éviter d'obtenir ce type de volumes est de construire un maillage initial où les sommets des volumes sont tous de degré 3, ce qui est le cas pour l'hexaèdre, le prisme ou encore le tétra-

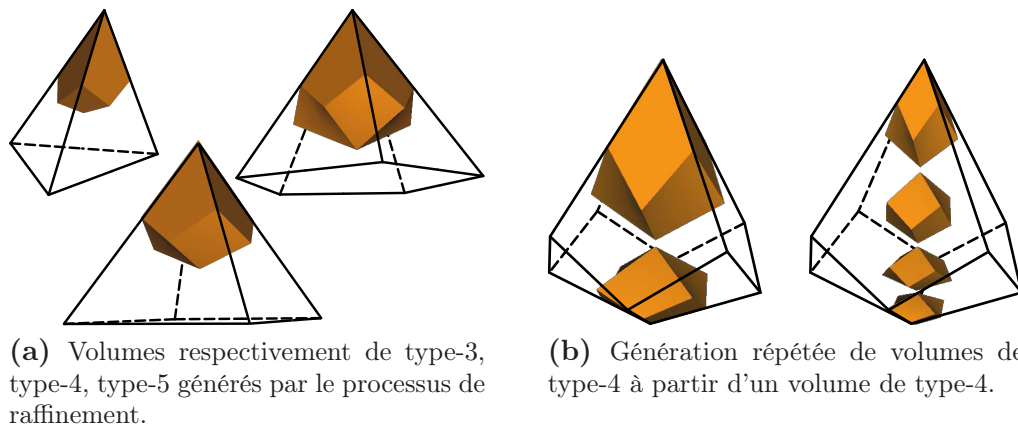


FIGURE 3.10 : Illustration du raffinement de pyramides dont le degré de la base varie.

èdre. L'apparition de ce type de polyèdres peut être gênant notamment pour certaines méthodes numériques. Cette raison a motivé l'élaboration d'un nouveau type de raffinement appelé mixte qui mélange la quadrisection de faces polygonales et la quadrisection de faces triangulaires. L'application de ce raffinement à un polyèdre produit, pour des cas précis, des polyèdres de même type que le polyèdre initial.

### Raffinement par quadrisection de faces triangulaires

Le second raffinement opère sur les maillages tétraédriques ou tétraédriques/octaédriques en effectuant une quadrisection de faces triangulaires. Le raffinement de ces deux polyèdres à faces triangulaires que sont le tétraèdre et l'octaèdre est illustré par les figures 3.11 et 3.12. Pour la subdivision des faces, de nouveaux sommets sont introduits en subdivisant chaque arête du maillage. Ces nouveaux sommets sont reliés entre eux par de nouvelles arêtes. Puis, de nouvelles faces sont introduites entre ces nouvelles arêtes. Dans le cas du raffinement d'un tétraèdre, l'opération s'arrête là. Dans le cas du raffinement d'un octaèdre, ces nouvelles faces sont elles aussi raffinées en utilisant une opération de triangulation de face. Cette dernière opération forme les nouveaux polyèdres : un octaèdre par ancien sommet de chaque polyèdre initial et un tétraèdre par ancienne face de chaque polyèdre initial.

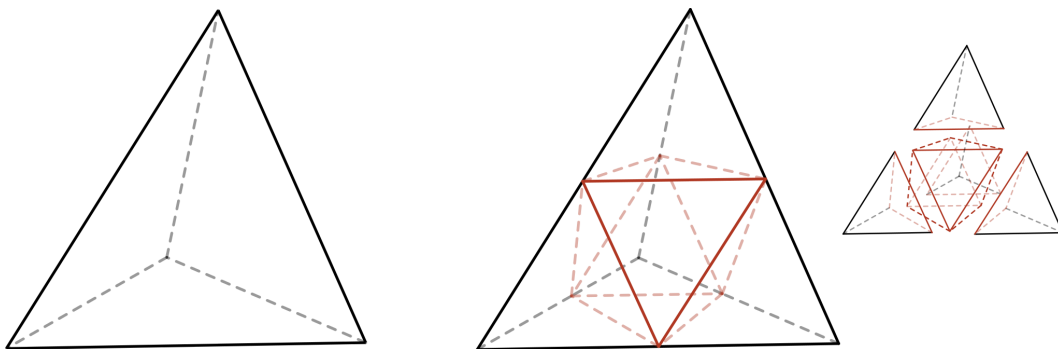


FIGURE 3.11 : Quadrisection de tétraèdres.

Nous allons illustrer les étapes du raffinement d'un tétraèdre et d'un octaèdre par une quadrisection de faces triangulaires entre deux niveaux consécutifs  $i$  et  $i + 1$  dans une 3-carte multirésolution. De la même manière que pour le raffinement précédent, le niveau  $i + 1$  est créé comme un copie du niveau  $i$ .

La figure 3.13 illustre le raffinement d'une face dans une 3-carte multirésolution. D'abord, toutes les arêtes du maillage sont découpées en deux par

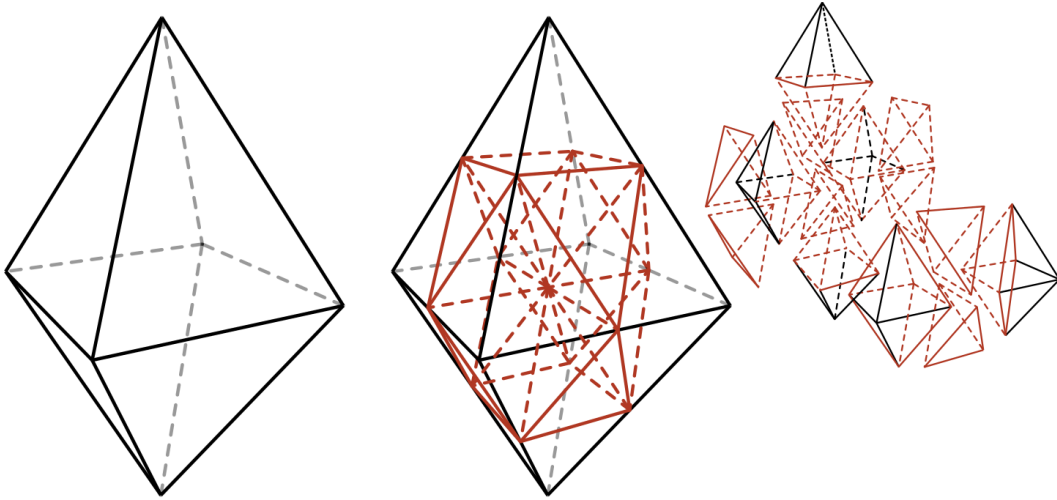


FIGURE 3.12 : Quadrisection d'octaèdres.

l'insertion d'un nouveau sommet (figure 3.13-B). Cette étape consiste à appliquer sur toutes les arêtes, l'opération de découpe d'arête décrite dans le chapitre 2. De nouvelles arêtes sont ensuite introduites entre tous ces nouveaux sommets par l'intermédiaire d'une opération de découpe de face décrite dans le chapitre 2 (figure 3.13-C).

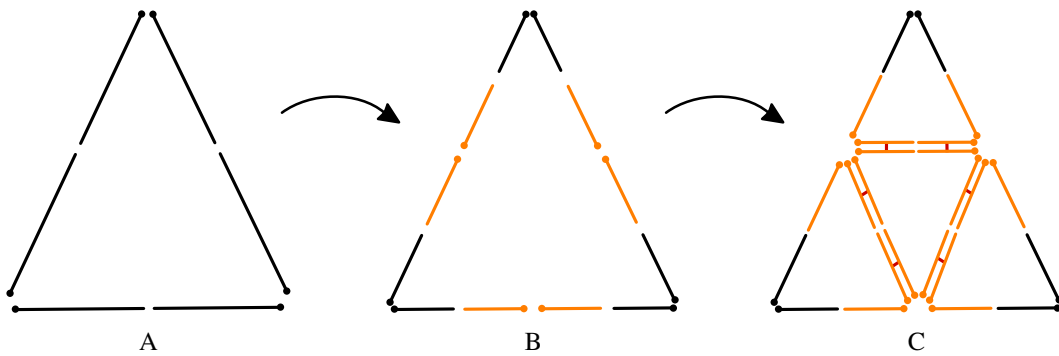


FIGURE 3.13 : Quadrisection de faces triangulaires au sein d'une 3-carte multirésolution.

La figure 3.14 illustre le raffinement d'un tétraèdre dans une 3-carte multirésolution. La première étape consiste à appliquer l'algorithme précédent à toutes les faces d'un tétraèdre (figure 3.14-B). De nouvelles faces sont ensuite introduites le long des arêtes nouvellement créées à l'aide de l'opération de découpe de volume également décrite dans le chapitre 2 (figures 3.14-C, 3.14-D, 3.14-E et 3.14-F). Cet algorithme insère quatre faces de degré 3 formant quatre tétraèdres appuyés sur les sommets de départ.

La figure 3.15 illustre le raffinement d'un octaèdre dans une 3-carte multiré-

solution. La première étape consiste à appliquer l'algorithme de raffinement de face sur toutes les faces d'un octaèdre (figure 3.15-B). De nouvelles faces sont ensuite introduites le long des arêtes nouvellement créées à l'aide de l'opération de découpe de volume décrite dans le chapitre 2 (figure 3.15-C1). Ces nouvelles faces sont, à leur tour, subdivisées par un algorithme de triangulation d'une face (figure 3.15-C2). La figure 3.15-C3 est équivalente à la figure 3.15-C2. Le nouveau sommet inséré a simplement été déplacé pour montrer l'octaèdre généré. À cette étape de l'algorithme, nous avons six octaèdres ainsi qu'un polyèdre résultant des découps successives dans l'octaèdre initial (figure 3.15-D). Le polyèdre résultant est illustré dans l'encadré en pointillé de la figure 3.15-D. C'est un polyèdre dont des sommets sont géométriquement pincés sur un sommet central obtenu par la création des octaèdres. La dernière étape de notre méthode consiste découper dans ce polyèdre les différents tétraèdres. Pour effectuer cette opération l'algorithme consiste à échanger, sur ce polyèdre, les coutures par  $\alpha_2$  des arêtes incidentes au sommet central afin de former les différents tétraèdres (figure 3.15-E).

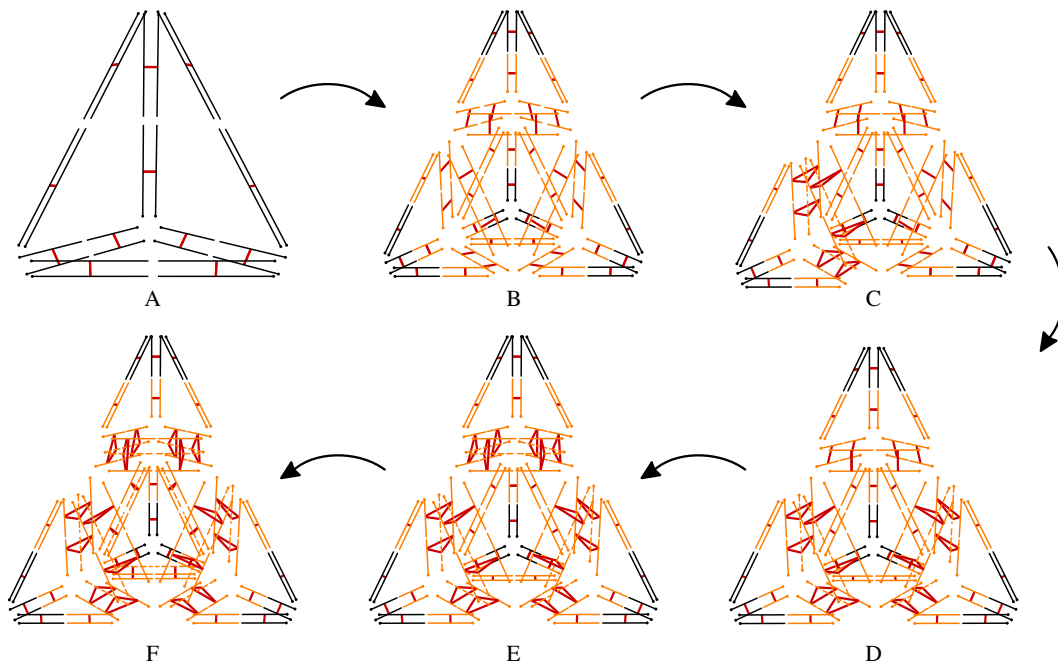


FIGURE 3.14 : Quadrissection de tétraèdres au sein d'une 3-carte multirésolution.

Ces deux algorithmes sont liés par le raffinement de leurs faces. Grâce au contrôle local des modifications de la topologie, il est possible d'élaborer un seul algorithme pouvant s'appliquer à ces deux polyèdres. En calculant le degré des faces ajoutées, ou plus exactement la longueur du chemin d'arête servant à la découpe de volume, il est possible de décider quel polyèdre est en train d'être raffiné.



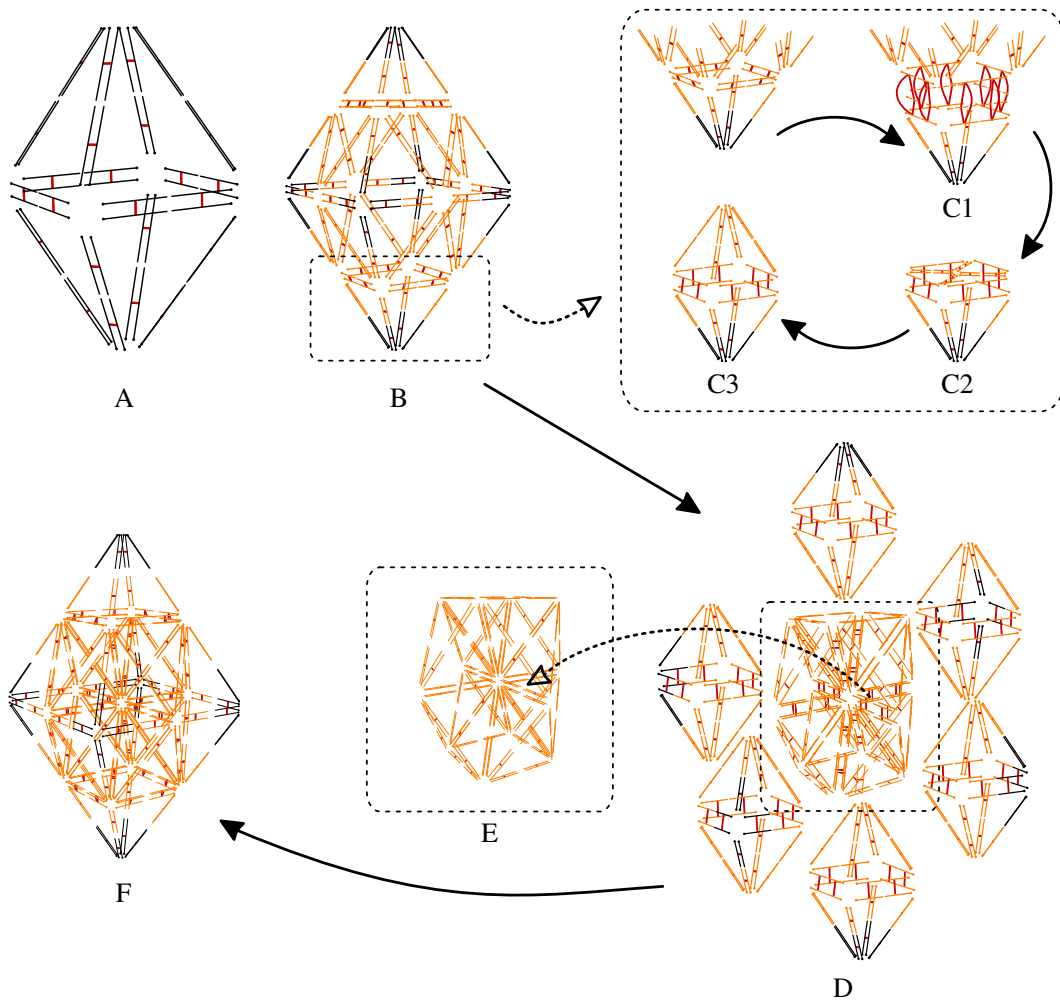


FIGURE 3.15 : Quadrisection d'octaèdres au sein d'une 3-carte multirésolution.

### Raffinement par quadrisection de faces mixtes

Le troisième raffinement opère sur des maillages polyédriques en effectuant une quadrisection de faces mélangeant les opérations de quadrisection triangulaire et polygonale. L'image 3.16 illustre ce raffinement sur un prisme. D'abord, toutes les arêtes du maillages sont subdivisées. Chaque face est ensuite raffinée en fonction de son degré, soit par l'opération de quadrisection triangulaire, soit par l'opération de quadrisection polygonale. Pour finir, de nouvelles faces sont introduites le long des arêtes introduites précédemment. Ces nouvelles faces sont elles aussi raffinées grâce aux opérations de quadrisection triangulaire ou polygonale. Cette dernière opération permet de former les nouveaux polyèdres.

Nous allons illustrer les étapes du raffinement mixte entre deux niveaux consécutifs  $i$  et  $i + 1$  dans une 3-carte multirésolution. De la même manière

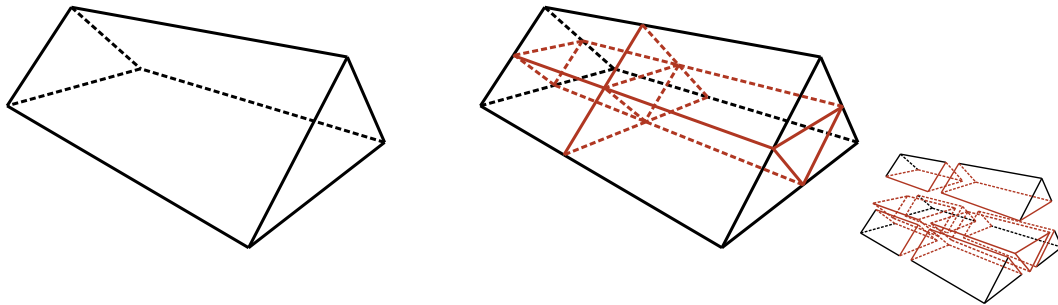


FIGURE 3.16 : Quadrisection de prismes.

que pour les raffinements précédents, le niveau  $i + 1$  est créé comme une copie du niveau  $i$ .

Toutes les arêtes du maillage sont coupées en deux par l'insertion d'un nouveau sommet. Ensuite, l'opération de découpe d'une face est exécutée en fonction du degré de celle-ci. Comme pour les algorithmes précédents de nouvelles faces sont introduites le long des nouvelles arêtes à l'aide de l'opération de découpe de volume. Le chemin d'arêtes à construire pour découper un coin dépend cette fois-ci du degré de chaque face d'un volume autour du sommet de niveau  $i$ . Le chemin d'arêtes est constitué d'une arête par face triangulaire et de deux arêtes par face polygonale.

Le traitement appliqué à cette nouvelle face par ombrelle est discriminé par le degré du sommet du coin (sommet de niveau  $i$ ), ainsi que par le degré de cette nouvelle face insérée comme illustré figure 3.17. Ainsi pour un coin de degré trois, il y a quatre cas différents suivant la permutation des faces triangulaires et quadrangulaires autour de ce coin.

**cas A :** Pour une face de degré 3, autrement dit 3 faces triangulaires autour du coin, nous sommes dans le cas du tétraèdre.

**cas B :** Pour une face de degré 4, c'est-à-dire 1 face quadrangulaire et 2 faces triangulaires, nous sommes dans le cas d'une pyramide.

**cas C :** Pour une face de degré 5, c'est-à-dire 2 faces quadrangulaires et 1 face triangulaire, nous sommes dans le cas d'un prisme.

**cas D :** Pour une face de degré 6, c'est-à-dire 3 faces quadrangulaires, nous sommes dans le cas d'un hexaèdre.

Pour un coin de degré 4, nous traitons le cas où toutes les faces autour du sommet sont triangulaires comme pour un octaèdre (figure 3.17-E). Enfin, pour tous les cas non énumérés, l'algorithme retombe dans le cas général en appliquant une quadrisection de faces polygonales.

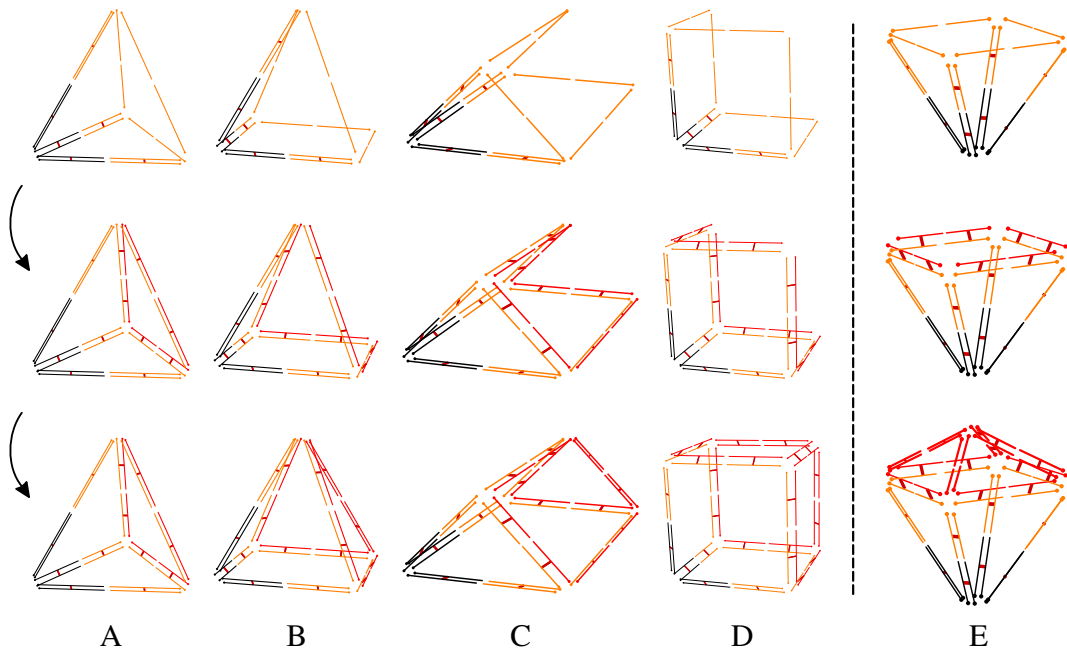


FIGURE 3.17 : Discrimination pour la subdivision mixte au sein d'une 3-carte multirésolution.

### 3.2.2 Subdivision d'un polyèdre arbitraire

Nous proposons dans cette section de formaliser dans le cadre des cartes combinatoires les relations d'incidences à reconstruire pour un raffinement à base de quadrisection polygonale de faces en dimension quelconque. Nous montrons ici comment le modèle des cartes, et notamment la définition implicite des cellules, permet de retrouver et de générer simplement toutes les relations d'incidences et d'adjacences entre toutes les cellules nouvellement créées. Ainsi, expliciter sur les brins les relations à reconstruire pour le raffinement de faces, puis de volumes, nous permet d'amener les notations et idées afin de donner un aperçu de la généralisation de notre méthode à des cellules de dimension quelconque.

#### Ordre des arêtes autour d'un sommet

Considérons, pour commencer, la subdivision d'une face et l'ordre des arêtes autour du sommet inséré. La figure 3.18a montre une face pentagonale avec les relations  $\alpha_0$  et  $\alpha_1$  en rouge et vert. La figure 3.18b montre cette face subdivisée avec un sommet central. L'objectif consiste à trouver l'ordre des arêtes autour de ce sommet central (représenté par la flèche bleue), pour pouvoir définir les

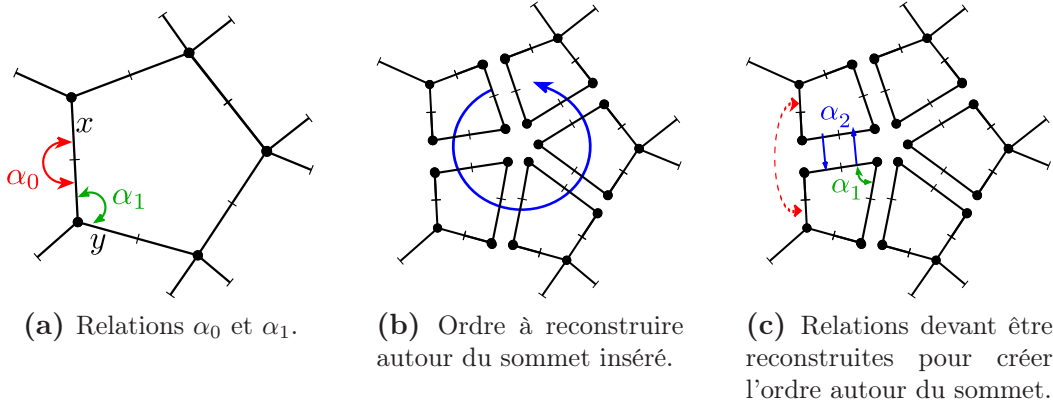


FIGURE 3.18 : Relations à reconstruire dans une face subdivisée.

relations entre les brins insérés pour former ce sommet.

La figure 3.18c montre une vue éclatée des faces générées par le raffinement. La combinaison  $\alpha_1 \circ \alpha_2$  permet de traverser les arêtes comme souhaité. Pour reconstituer l'ordre autour du sommet, il est donc suffisant de générer correctement la relation  $\alpha_2$  entre les faces ajoutées. Nous remarquons que ces relations sont directement liées aux relations  $\alpha_0$  des brins des arêtes subdivisées (voir flèches en pointillés rouges sur la figure 3.18c). La force de notre méthode est la possibilité d'alterner entre la représentation primale et duale afin de propager cette information durant cette étape de subdivision et entre les niveaux de résolution.

Maintenant, décrivons l'algorithme pour la subdivision d'une face de niveau  $i$ . Dans un premier temps, toutes les arêtes sont coupées en deux en ajoutant deux brins de niveau  $i + 1$  comme montré en orange figure 3.19a. Pour tout niveau  $i$ , cette étape reconstruit les relations comme suit (avec  $y = \alpha_1^i(\alpha_0^i(x))$ ) :

$$\alpha_1^{i+1}(\alpha_0^{i+1}(x)) = \alpha_0^{i+1}(\alpha_1^i(y))$$

Les sommets de la face originale forment des « coins » séparés visibles dans la figure 3.19a. La deuxième étape consiste à former les faces émergentes en reliant les sommets de niveau  $i + 1$  par deux arêtes, ce qui signifie 4 brins au niveau  $i + 1$  (illustré figure 3.19b).

L'étape finale (illustrée figure 3.19c) consiste à coudre par  $\alpha_2$  au niveau  $i + 1$  les paires d'arêtes attachées aux brins connectés par  $\alpha_0$  au niveau  $i$  (brins noirs). Pour tout niveau  $i$ , cette étape reconstruit les relations comme suit (avec  $y = \alpha_1^i(\alpha_0^i(x))$ ) :

$$\alpha_2^{i+1}(\alpha_1^{i+1}(\alpha_0^{i+1}(x))) = \alpha_1^{i+1}(\alpha_0^{i+1}(\alpha_1^i(y)))$$

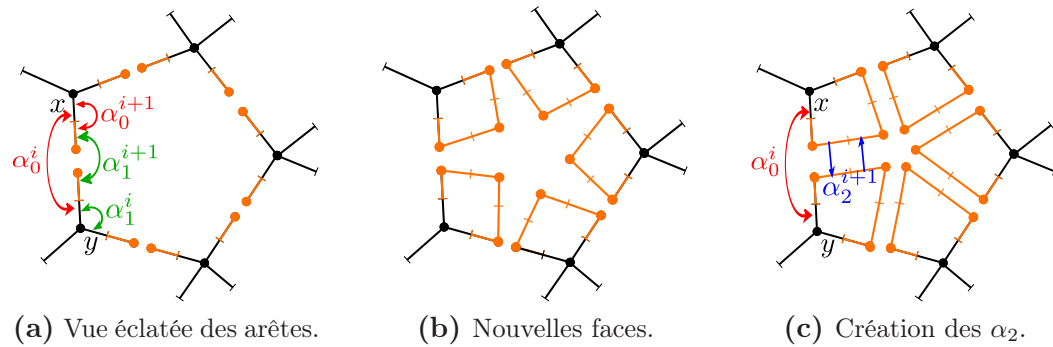


FIGURE 3.19 : Étapes de subdivision d'une face.

Finalement, l'ordre des arêtes autour du sommet central est reconstruit de manière totalement implicite et indépendamment du degré de la face initiale. Ce principe est étendu à l'ordre des faces autour d'une arête intérieure d'un volume dans la sous-section suivante.

### Ordre des faces autour d'une arête

Considérons maintenant la subdivision d'un volume illustrée par la figure 3.20. Concentrons nous en particulier sur l'ordre des faces autour des arêtes (en rouge sur la figure 3.20b). Notons  $\phi_2 = \alpha_2 \circ \alpha_0$ . Nous savons de la définition d'une 3-carte que cette relation est une involution. Ceci signifie que  $\phi_2(\phi_2(d)) = d$  pour tout brin  $d$ . Rappelons ici que la relation  $\phi_2$  lie un brin d'une face au brin de la face adjacente appartenant au même volume. Suivant la même approche que pour l'ordre des arêtes autour d'un sommet, nous décomposons la permutation  $\alpha_2$  pour la reconstruire en  $\alpha_2 = \phi_2 \circ \alpha_0$ . La figure 3.20c montre les faces autour de l'arête surlignée en rouge et la figure 3.20d montre une vue éclatée de ces faces. Nous observons que la combinaison  $\phi_2 \circ \alpha_0$  permet de traverser toutes les faces de la façon désirée. Ainsi, pour reconstruire la séquence de faces il est suffisant de générer correctement les relations  $\alpha_0$  entre les faces ajoutées.

### Topologie d'un polyèdre subdivisé

Les outils présentés ci-dessus sont utilisés pour définir élégamment n'importe quelle subdivision d'un polyèdre. D'abord, les faces sont subdivisées en leur ajoutant un sommet central. A cette étape, la couture des arêtes autour de chaque sommet central n'est pas effectuée. Il en résulte une ombrelle (ou

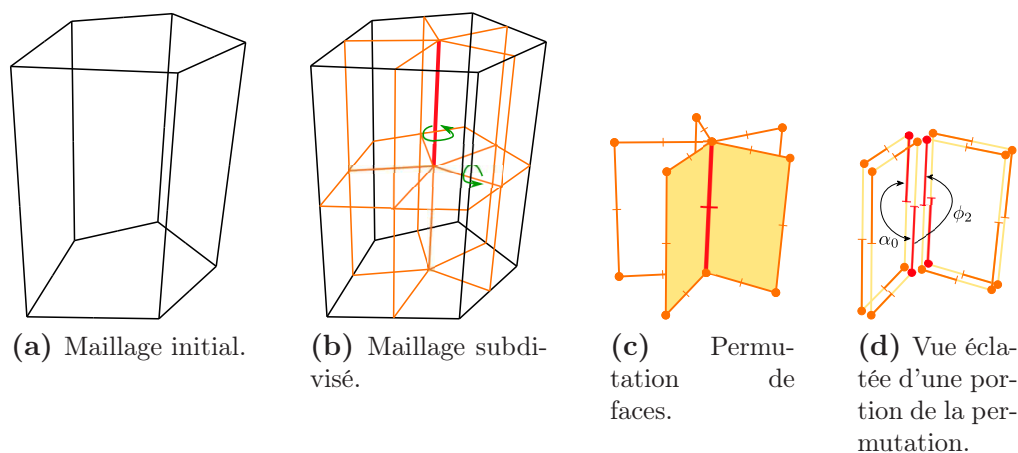


FIGURE 3.20 : Permutation de faces autour d'une arête.

un coin) composée de faces quadrangulaires (illustrée figure 3.21a) autour de chaque sommet initial. Dans un second temps, cette ombrelle est automatiquement complétée pour former le futur volume. Pour ce faire, une face topologique est créée le long des arêtes de l'ombrelle (illustrée figure 3.21b). Cette face topologique est, à son tour, subdivisée en ajoutant un sommet central. Ce nouveau sommet est positionné au centre du volume subdivisé (illustration figure 3.21c).

À ce stade, nous avons généré un volume par sommet initial du polyèdre (cf. figure 3.21d). Les relations précédemment laissées libres peuvent maintenant être complétées pour ordonner les faces internes des volumes générés autour de leur arête commune (cf. flèche verte figure 3.21d). En particulier, pour chaque paire de sommets du maillage initial, nous considérons les deux brins  $x$  et  $y$  cousus par  $\alpha_0$  au niveau  $i$ . Leurs images par  $\alpha_0^{i+1}$  sont les sommets des deux faces incidentes. Ces deux faces sont fusionnées en cousant leurs brins deux à deux avec la relation  $\phi_2$ . Après cette étape, toutes les relations topologiques entre les cellules créées (sommets, arêtes, faces, volumes) sont reconstruites. Ceci démontre combien notre algorithme est générique puisqu'il permet de subdiviser tout volume, quelque soit son degré et le degré de ses faces.

### Subdivision en dimension quelconque

Le processus de subdivision peut s'étendre de manière récursive en dimension supérieure. Nous donnons ici un aperçu du principe. Comme précédemment, nous définissons la relation  $\phi_{n-1} = \alpha_{n-1} \circ \alpha_0$  qui représente l'ordre cyclique des cellules de dimension  $n - 1$  autour de cellules de dimension  $n - 2$

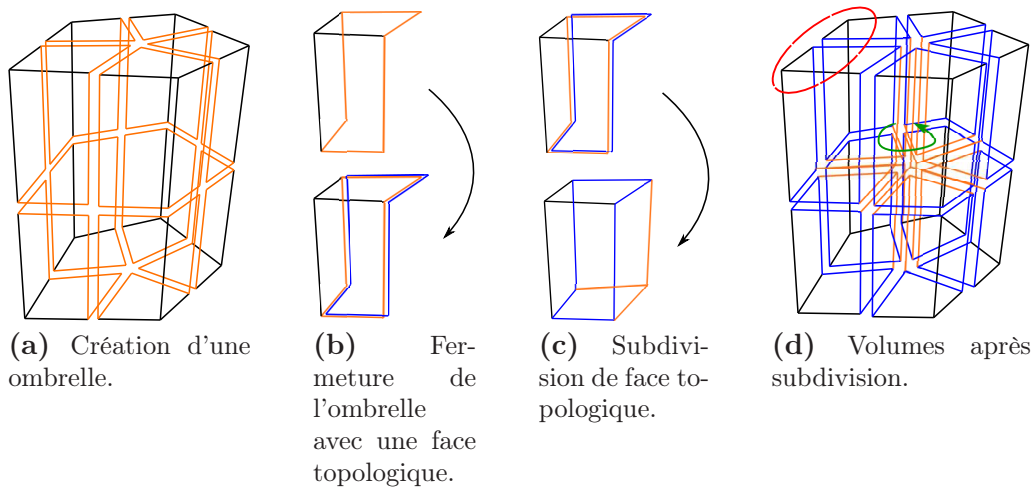
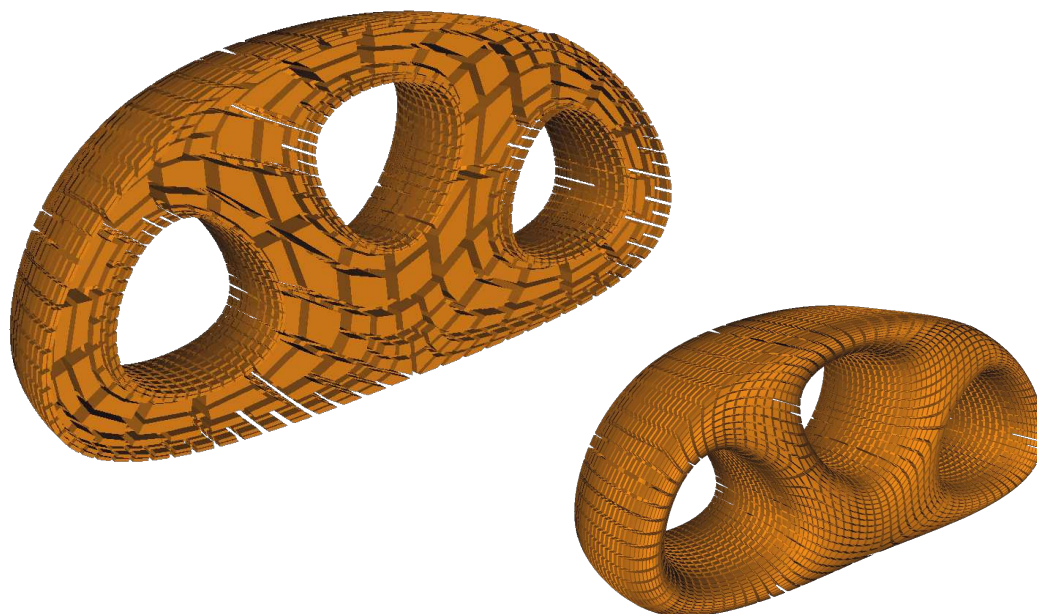


FIGURE 3.21 : Subdivision topologique.

dans une carte de dimension  $n$ . La subdivision d'un hypervolume  $v$  de dimension  $n$  à un niveau de résolution  $l$  commence par la subdivision des cellules de dimension  $n - 1$  formant son bord. La couture autour du sommet central des cellules de dimension  $n - 1$  générées n'est pas effectuée. Nous obtenons des ensembles de cellules de dimension  $n - 1$  liés formant des ombrelles de dimension  $n - 1$  autour du sommet initial de  $v$  avec des bords ouverts où la couture n'a pas été effectuée. Ces ombrelles sont fermées avec une cellule de dimension  $n - 1$ . Ces cellules de dimension  $n - 1$  sont ensuite subdivisées à la dimension  $n - 1$ . Ceci génère toutes les cellules nécessaires pour reconstruire le sommet central. Ces cellules sont cousues avec la relation  $\alpha_0$ . Les relations  $\alpha_0$  au niveau  $l$  sont requises afin de trouver les couples de cellules qui doivent être cousus au niveau  $l + 1$ . Grâce à la combinatoire des cartes de dimension  $n$ , cette procédure reconstruit implicitement la relation  $\phi_{n-1}$  et ainsi toutes les relations de voisinage nécessaires entre les cellules insérées au niveau  $l + 1$ .

### 3.2.3 Génération adaptative

La génération d'un niveau de manière adaptative consiste à choisir, selon un critère donné, un ensemble de volumes qui seront raffinés au niveau suivant. Le critère de raffinement est différent suivant l'application visée. Par exemple, on peut envisager de raffiner les volumes d'un maillage dans une zone d'interaction, notamment pour découper un maillage, ou encore dans la zone d'insertion d'un outil. Dans ce type d'application raffiner le maillage permet d'avoir des propriétés mécaniques plus précises dans une zone particulière de l'objet. Un critère plus simple consiste à successivement subdiviser seulement



**FIGURE 3.22** : Exemple de maillage obtenu par subdivision adaptative où le bord est plus subdivisé que l'intérieur.

les volumes du bord afin de lisser la surface de l'objet sans générer trop de volumes intérieurs. La figure 3.22 montre un exemple de maillage dont seules les cellules du bord ont été subdivisées afin de lisser géométriquement l'objet. Le bord du maillage obtenu est le même que celui de la figure 3.5 seulement avec moins de volumes intérieurs. Les notions nécessaires afin de maintenir la consistance des niveaux lors du raffinement sont les mêmes quelque soit le critère utilisé. Nous allons donc détailler comment ce type de raffinement est géré grâce aux 3-cartes multirésolutions. Nous rappellerons la notion de *niveau de subdivision* d'une cellule et nous l'appliquerons au cas 3D afin de pouvoir adapter dynamiquement le niveau de subdivision des volumes d'un maillage quelconque.

Prenons l'exemple de la figure 3.23. À gauche, deux hexaèdres ( $c_1$  et  $c_2$ ) de niveau 0 ont été extraits d'un maillage quelconque. À droite, l'hexaèdre  $c_2$  a été subdivisé de manière adaptative au niveau 1. À l'inverse du cas régulier, toutes les cellules du maillage n'ont pas été subdivisées. Ce maillage est donc composé de cellules dont la topologie est hétérogène.

Pour être plus précis, dans une 3-carte combinatoire, une face entre deux polyèdres est commune aux deux. La subdivision d'un niveau  $i$  au niveau  $i + 1$  d'un des polyèdres entraîne la subdivision de la face commune. Ainsi, les brins de cette face vus du polyèdre non subdivisé sont traversés au niveau  $i$ . La face est toujours considérée comme non subdivisée. À l'inverse, les brins de cette



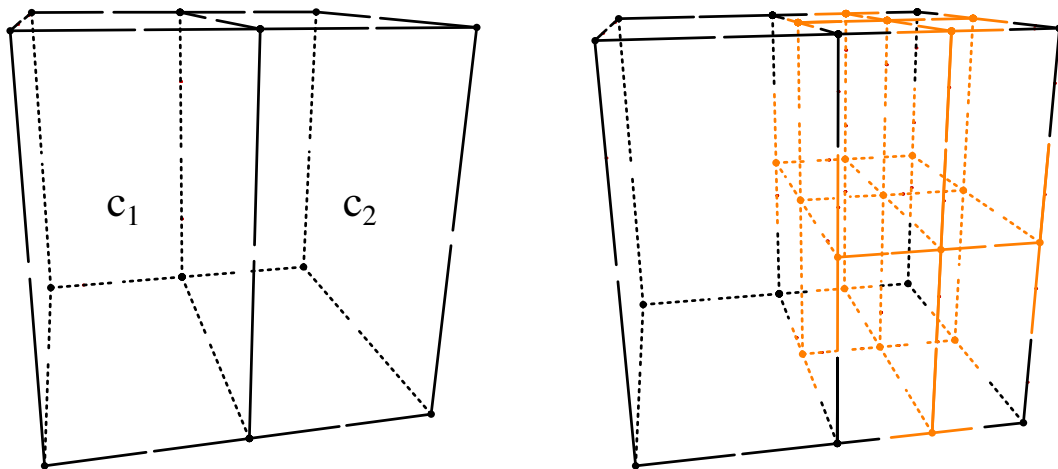


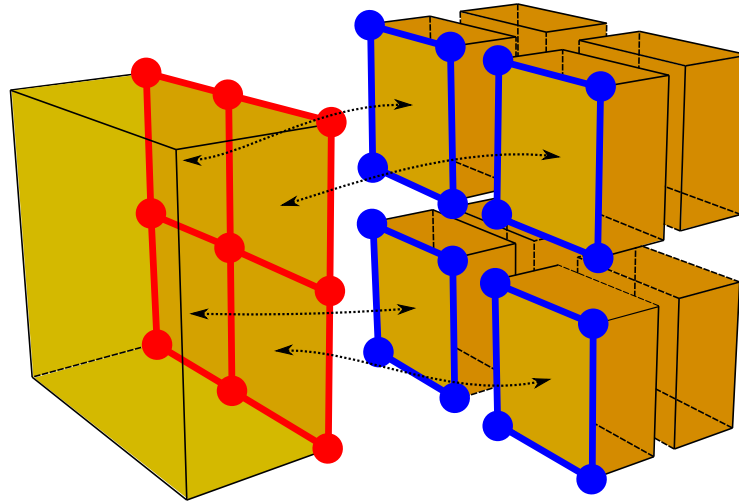
FIGURE 3.23 : Subdivision adaptative d'un hexaèdre dans un maillage

face vues du polyèdre subdivisé sont traversés au niveau  $i + 1$ . Ainsi, la même face est vue comme subdivisée. Nous exploitons ici la capacité du modèle des cartes combinatoires à représenter des cellules arbitraires.

Dans l'exemple de la figure 3.24 l'hexaèdre du fond à été raffiné en huit hexaèdres. La face adjacente aux deux hexaèdres de niveau  $i$  est subdivisée de manière adaptative au niveau  $i + 1$ . Elle est matérialisée en rouge pour le polyèdre de gauche et en bleu pour les hexaèdres de droite. Le polyèdre aux faces rouges est un hexaèdre au niveau  $i$ . Au niveau  $i + 1$ , c'est un polyèdre de niveau  $i$  à neuf faces ayant quatre faces de niveau  $i$  à cinq côtés, une face de niveau  $i$  à quatre côtés et quatre faces de niveau  $i + 1$  à quatre côtés. Les polyèdres à face bleu sont des hexaèdres de niveau  $i + 1$  ayant six faces à quatre côtés.

Pour subdiviser l'hexaèdre  $c_1$  de la figure 3.23 au niveau 1, l'algorithme de raffinement ne peut pas simplement raffiner le polyèdre comme nous l'avons fait jusqu'à maintenant. En effet, une des faces de ce polyèdre n'est plus du même niveau que les autres. L'algorithme procéderait donc au raffinement d'un polyèdre à neuf faces. Celui-ci serait donc incorrect car il n'en résulterait pas huit cubes comme attendu. Pour ce faire, il est nécessaire de traiter correctement les cellules (arêtes et faces) incidentes aux polyèdres ayant différents niveaux de subdivision. Pour résoudre ce problème, nous allons faire intervenir la notion de niveau de subdivision d'une cellule.

La subdivision du volume  $c_1$  au niveau 1 commence par la subdivision de chacune de ces faces. Pour chaque face, il faut d'abord vérifier si elle a déjà été subdivisée au niveau 1. Pour cela, il faut commencer par tester si toutes les arêtes de la face ont déjà été subdivisées au niveau 1. Si au moins une arête de



**FIGURE 3.24 :** Illustration d'une face incidente à des volumes ayant un niveau de subdivision différent.

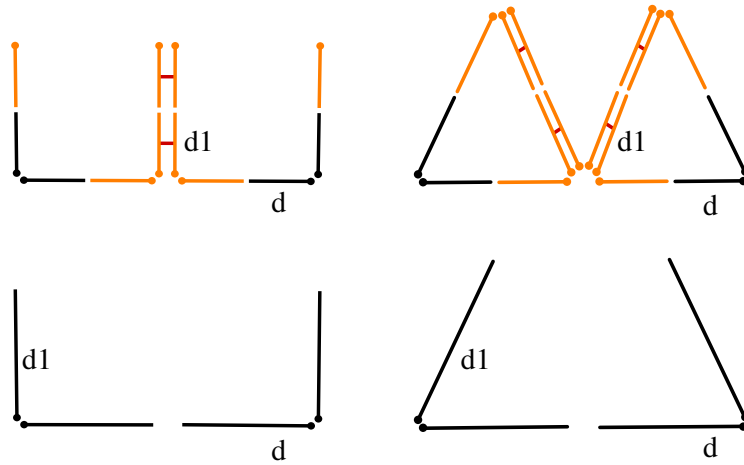
cette face n'a pas été subdivisée au niveau 1, alors la face n'a pas encore été subdivisée au niveau 1. Si toutes les arêtes ont été subdivisées au niveau 1, il faut tester si la face elle-même a été subdivisée. En effet, nous pourrions nous retrouver dans le cas où toutes les faces adjacentes à cette face par une arête (les volumes voisins) aient été subdivisées mais pas la face elle-même.

Pour vérifier que cette face a été subdivisée au niveau 1, on cherche un brin de cette face au niveau 1. Puis, on teste le niveau de ce brin pour vérifier qu'il est égal au niveau courant, en l'occurrence 1. Étant donné un brin  $d$  de niveau 0 de la face dont on cherche à tester le niveau, un brin de cette face au niveau 1 est obtenu en prenant le brin  $d_1 = \alpha_1^1(\alpha_0^1(d))$  comme illustré par la figure 3.25.

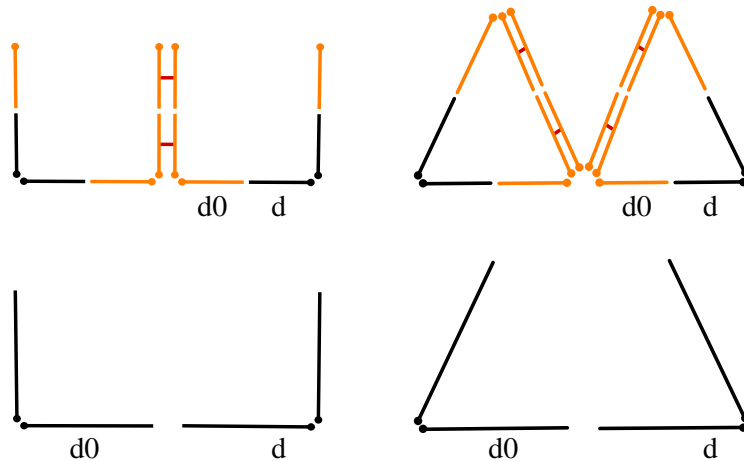
Dans le cas où une face n'a pas encore été subdivisée, l'algorithme continue par la subdivision des arêtes de cette face. L'algorithme commence ici par vérifier si chaque arête de la face a déjà été subdivisée au niveau 1. Si une arête n'a pas encore été subdivisée au niveau 1, elle l'est à ce moment là.

Afin de vérifier si une arête a déjà été subdivisée, on cherche un brin de cette arête au niveau 1. Puis, on teste le niveau de ce brin pour vérifier qu'il est égal au niveau courant, en l'occurrence 1. Étant donné un brin  $d$  de niveau 0 de l'arête dont on cherche à tester le niveau, un brin de cette arête au niveau 1 est obtenu en prenant le brin  $d_0 = \alpha_0^1(d)$  comme illustré par la figure 3.26.

Lorsque toutes les arêtes de cette face ont correctement été subdivisées, le raffinement de face peut être exécuté au niveau 1. Pour la subdivision de la



**FIGURE 3.25 :** Extraits de deux faces (triangulaire et polygonale) subdivisées (en haut) et non subdivisées (en bas). Dans le cas où la face n'a pas été subdivisée (images du bas), le brin  $d_1 = \alpha_1^1(\alpha_0^1(d))$  est un brin de niveau 0. Dans le cas où la face a été subdivisée (images du haut), le brin  $d_1$  est un brin de niveau 1.



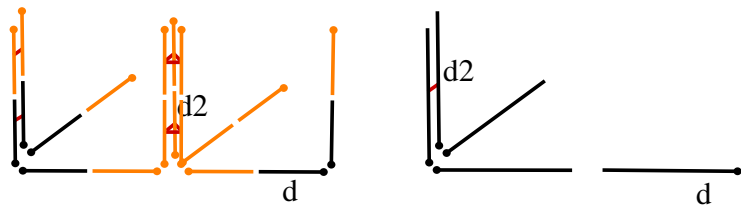
**FIGURE 3.26 :** Extraits de deux faces (triangulaire et polygonale) subdivisées (en haut) et non subdivisées (en bas). Dans le cas où l'arête n'a pas été subdivisée (images du bas), le brin  $d_0 = \alpha_0^1(d)$  est un brin de niveau 0. Dans le cas où l'arête a été subdivisée (images du haut), le brin  $d_0$  est un brin de niveau 1.

face, les algorithmes de quadrisection polygonale ou triangulaire peuvent être appliqués indifféremment, comme nous l'avons vu dans la section précédente. Lorsque toutes les faces du volume ont correctement été subdivisées, le raffinement de volume peut-être exécuté au niveau 1. Là encore, les algorithmes de raffinement vus précédemment s'appliquent directement.

Lors de la mise à jour du maillage, il faut s'assurer que la subdivision d'un volume est autorisée. Ainsi, avant de subdiviser le volume  $c_1$  au niveau 1, il

faut tester et s'assurer que ce volume n'a pas encore été subdivisé au niveau 1. Il faut commencer par tester si le niveau du volume est inférieur au niveau courant, car dans ce cas, le volume n'a pas encore été subdivisé. Dans le cas contraire, il faut tester si toutes les faces du volume ont déjà été subdivisées au niveau suivant. Si tel est le cas, alors, de la même manière que pour le test sur les faces, il faut tester si le volume lui-même a déjà été subdivisé. En effet, nous pouvons nous retrouver dans le cas où toutes les faces du volume ont été subdivisées car tous les volumes adjacents à ce volume par leur face ont été subdivisés mais pas le volume lui-même.

Afin de vérifier si le volume a été subdivisé, on cherche un brin de ce volume au niveau 1. On vérifie ensuite si le niveau de ce brin est bien égal à 1. Étant donné un brin  $d$  de niveau 0 du volume dont on cherche à tester le niveau, un brin de ce volume au niveau 1 est obtenu en prenant le brin  $d_2 = \alpha_2^1(\alpha_1^1(\alpha_0^1(d)))$  comme illustré par la figure 3.27.



**FIGURE 3.27** : Extraits d'un volume subdivisé (à gauche) et non subdivisé (à droite). Dans le cas où le volume n'a pas été subdivisé, le brin  $d_2$  est un brin de niveau 0. Dans le cas où le volume a été subdivisé, le brin  $d_2 = \alpha_2^1(\alpha_1^1(\alpha_0^1(d)))$  est un brin de niveau 1. Ce brin appartient alors à une face intérieure.

Tous ces algorithmes fonctionnent de manière générale quelque soit le niveau de résolution en remplaçant 0 et 1 par  $i$  et  $i + 1$ . Pour tous ces tests, nous exploitons largement les capacités du modèle des cartes combinatoires pour parcourir aisément les adjacences et incidences à des niveaux différents.

### 3.2.4 Simplification

Certaines applications nécessitent de mettre à jour dynamiquement la topologie du maillage. Dans ce cadre, suivant un critère donné, certaines cellules doivent être raffinées. À l'inverse, il se peut que pour d'autres cellules, le niveau de subdivision ne soit plus adapté au critère. Nous pouvons alors procéder à une simplification de ces cellules.

La simplification s'effectue en inversant les opérations de raffinement vu précédemment. Cette opération s'effectue entre deux niveaux consécutifs à

l'image des opérations de raffinement. De manière similaire au raffinement adaptatif de volumes, des contraintes doivent être ajoutées afin de procéder à une simplification correcte des cellules du maillage. En effet, lors de la subdivision d'une cellule de niveau  $i$  vers un niveau  $i + 1$ , il faut s'assurer qu'elle n'est pas déjà subdivisée à ce niveau. De manière analogue, lors de la simplification d'une cellule du niveau  $i + 1$  au niveau  $i$ , il faut s'assurer qu'elle a été subdivisée au niveau  $i + 1$  et seulement au niveau  $i + 1$ . En effet, si la cellule avait été subdivisée à des niveaux plus élevés, l'atomicité de l'opération entre deux niveaux consécutifs ne peut être respecté.

La simplification d'un volume de niveau  $i + 1$  au niveau  $i$  commence par l'application d'une opération inverse de la subdivision permettant de détruire toutes les faces intérieures en fusionnant les volumes adjacents par leurs faces en utilisant l'opération de fusion de volumes décrite dans le chapitre 2. Cette opération permet également de détruire le sommet central créé par un certain nombre de raffinements. Il faut ensuite simplifier les faces du volume. L'algorithme commence par vérifier si chaque face du volume peut être simplifiée du niveau  $i + 1$  au niveau  $i$ . Une face peut être simplifiée si les deux volumes adjacents à cette face ne sont pas subdivisés et si la face elle-même n'est subdivisée que sur un seul niveau, le niveau  $i + 1$ . Si tel est le cas, la face est simplifiée par fusion des faces de niveau  $i + 1$  entre elles le long de leurs arêtes communes. Pour finir, il faut demander à chaque face la simplification de ses arêtes. L'algorithme commence donc par interroger chaque arête pour savoir si elle a été subdivisée au niveau  $i + 1$ . Si tel est le cas, il faut vérifier qu'aucun volume incident à cette arête ne soit subdivisé. En effet, si un autre volume autour de cette arête est subdivisé, des arêtes formant les faces subdivisées des volumes voisins seraient adjacentes à cette arête. Si tel est le cas, il faut finir par vérifier, comme pour les cellules précédentes, que cette arête n'est pas subdivisée plus avant que le niveau  $i + 1$ . Si ces conditions sont remplies, l'arête peut être simplifiée en utilisant l'opération de découpe inverse d'arête décrite dans le chapitre 2.

Avant de procéder à la simplification d'un volume afin de mettre à jour la topologie du maillage, il faut tester si ce volume n'a été subdivisé qu'une seule fois. Pour cela, on teste d'abord si le volume a été subdivisé du niveau  $i$  au niveau  $i + 1$ . Si tel est le cas, pour chacun des volumes de niveau  $i + 1$  créé par le raffinement, il faut tester si à leur tour ils ont été subdivisés plus avant. Si aucun d'entre eux n'a été subdivisé, ce volume peut être simplifié. Dans le cas contraire, il ne peut pas l'être.

Nous tirons encore une fois parti de la force des cartes combinatoires qui nous permettent de parcourir les cellules à différents niveaux. En effet, chaque volume de niveau  $i + 1$  possède au moins une face commune avec les faces du

volume de niveau  $i$ . Il est donc possible en parcourant le niveau  $i$  de passer par chacun de volumes de niveau  $i + 1$  afin de le tester à moindre coût en traversant les faces du volume de niveau  $i$ .

### 3.2.5 Niveau de subdivision d'une cellule

Comme exposé précédemment par la figure 3.24, une face peut-être vue de manière différente suivant le niveau duquel on la regarde. Il est ainsi nécessaire de connaître et de définir des moyens de calculer le niveau d'une face pour pouvoir interpréter correctement sa topologie. Dans cette section, nous définissons plus formellement la notion de niveau de subdivision d'une cellule.

Pour décrire formellement ce qu'est une subdivision adaptative, nous définissons la notion de niveau de subdivision d'une cellule de dimension  $k$  dans une  $n$ -carte de niveau  $i$ . Nous appelons  $cell_k^i(b)$  l'ensemble de brins de la  $k$ -cellule du brin  $b$  au niveau  $i$ . Nous définissons le niveau de subdivision d'une  $k$ -cellule d'un brin  $b$  comme étant égal au plus petit niveau  $j$  tel que  $cell_k^j(b') \subset cell_k^i(b)$ , avec  $b' \in cell_k^i(b)$ . L'inclusion dans la formule précédente signifie que le niveau d'une cellule est défini comme le plus petit niveau tel que le bord de cette cellule peut-être subdivisé mais pas la cellule.

Définissons  $lev(b)$  le niveau d'un brin  $b$ . Pour déterminer le niveau d'une face  $f$  dans une 3-carte de niveau  $i$ , nous commençons par un de ses brins  $b$  avec le niveau d'insertion le plus petit,  $lev(b) \leq i$ . En effet, le niveau d'une face est nécessairement entre le niveau minimum de ses brins et le niveau  $i$  de la 3-carte considérée. D'abord, on teste si les brins de la face du brin  $b$  au niveau  $lev(b)$  sont contenus dans la face de  $b$  au niveau  $i$ . Si oui, alors  $f$  est une face de niveau  $lev(b)$  dans la carte de niveau  $i$ . Dans le cas contraire, on teste si les brins de la face de  $b$  au niveau  $lev(b) + 1$  sont contenus dans la face de  $b$  au niveau  $i$ . On boucle sur ce test de niveau jusqu'à trouver le plus petit niveau pour lequel les brins de la face de  $b$  sont contenus dans la face  $f$  au niveau  $i$ . Au plus, ce niveau est le niveau  $i$  lui-même.

Par exemple, pour la face  $f_1$  sur figure 3.28, nous utilisons le brin  $b_1$  inséré dans la carte de niveau 0, puis nous commençons par traverser la face de  $b_1$  au niveau 0 (les brins en noir). Or tous ces brins noirs sont contenus dans la face de  $b_1$  au niveau 2 (brins noir, orange et bleu). En d'autres termes, nous traversons toute la face et ne traversons que des brins de niveau 0. Nous pouvons en conclure que  $f_1$  est une face de niveau 0. Pour la face  $f_2$ , nous commençons par le brin  $b_2$  inséré dans le carte de niveau 0. Le brin suivant dans la face de  $b_2$  au niveau 0 n'est pas contenu dans la face de  $b_2$  au niveau

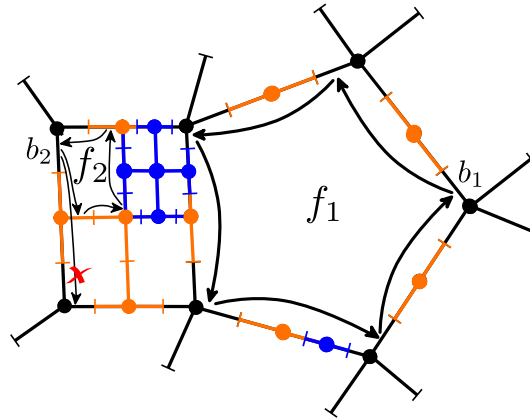


FIGURE 3.28 : Exemple du calcul du niveau d'une face : illustration pour les faces  $f_1$  et  $f_2$ , commencent respectivement par les brins  $b_1$  et  $b_2$ .

2. Nous continuons en traversant la face de  $b_2$  au niveau 1 et nous voyons que tous les brins de la face sont contenus dans la face de  $b_2$  au niveau 2. Nous en concluons que  $f_2$  est une face de niveau 1. En d'autres termes, avec les brins de niveau 0, on change de face mais pas avec les brins de niveau 1. Pour rester dans la même face, nous devons la considérer au niveau 1.

Utilisant cette information du niveau de subdivision d'une cellule, la subdivision adaptative peut s'étendre en dimension quelconque. Nous donnons ici un aperçu du principe. La subdivision d'un hypervolume  $v$  de dimension  $n$  du niveau de résolution  $i$  au niveau  $i + 1$  commence par la subdivision des cellules de dimension  $n - 1$  formant son bord. La subdivision de cette cellule de dimension  $n - 1$  commence par la recherche de son niveau. Si le niveau de cette cellule est déjà supérieur au niveau  $i + 1$ , cela implique que l'hypervolume adjacent a déjà été subdivisé. L'algorithme poursuit alors avec la prochaine cellule de dimension  $n - 1$ . Dans le cas contraire (la cellule de dimension  $n - 1$  est de niveau  $i$ ) le processus récursif de subdivision continue sur les cellules de dimension inférieures. Lorsque que le processus récursif arrive sur la condition d'arrêt (les cellules de dimension 1), les cellules non subdivisées de la dimension 0 à la dimension  $n - 1$  sont subdivisées. Nous pouvons ensuite utiliser les outils présentés dans les sections précédentes pour construire et lier topologiquement au niveau  $i + 1$  les couples de cellules nécessaires. Lorsque le processus est terminé nous obtenons des hypervolumes liés avec les relations topologiques au niveau  $i$  et au niveau  $i + 1$ . Appliquer cet algorithme de la sorte permet de gérer l'adaptativité de manière transparente dans la hiérarchie en dimension quelconque.

### 3.3 Mise à jour des attributs de sommet : cas de la géométrie

Rappelons que le processus de subdivision se décompose en deux phases, une phase de mise à jour de la connectivité du maillage puis une phase de mise à jour des attributs. Dans le cadre de la gestion et de la mise à jour d'attributs dans une hiérarchie volumique générée par un processus de subdivision primal, nous nous sommes intéressé aux attributs de sommet qui portent ici l'information géométrique.

Chaque phase de raffinement topologique produit deux types de sommets : les anciens sommets présents avant la subdivision et les nouveaux sommets introduits par le processus de raffinement. Les nouveaux sommets, également appelés sommets impairs, sont positionnés en un point calculé par l'application des masques de subdivision sur les anciens sommets. Les anciens sommets sont appelés des sommets pairs. Leur position est mise à jour dans le cas approximant. L'ensemble des sommets pairs et impairs forme le maillage de niveau  $i + 1$ . L'application d'un schéma de subdivision rend les sommets du niveau  $i + 1$  dépendant des sommets du niveau  $i$  et produit donc un lien ascendant entre les différents niveaux de résolution. En utilisant le vocabulaire de traitement du signal, ce processus est également appelé la synthèse de la géométrie d'un maillage.

Dans le cadre d'une utilisation multirésolution consistant à éditer un maillage à différents niveaux ou encore à filtrer du bruit à certaines fréquences, le recalcul de la géométrie pour accéder à des niveaux inférieurs va faire perdre l'information des niveaux supérieurs précédemment obtenue. Il faut donc ajouter un lien descendant entre les niveaux de résolution afin de déduire un maillage de niveau  $i$  à partir du maillage de niveau  $i + 1$  tout en conservant l'information nécessaire à la reconstruction, appelée détail, du maillage de niveau  $i + 1$  modifié. En traitement du signal, ce processus est appelé l'analyse de la géométrie d'un maillage.

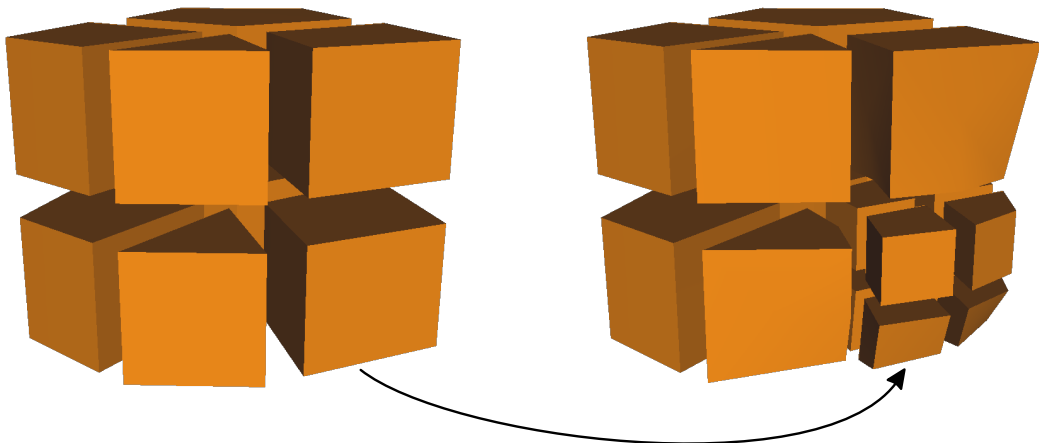
Le schéma multirésolution dont nous parlerons par la suite est décrit par le paradigme du *lifting scheme* [111]. Ce paradigme comprend une formulation algorithmique souple et efficace définissant un processus d'analyse et un processus de synthèse appelé un banc de filtres. Il s'agit de représenter les filtres sous la forme d'une série de plusieurs masques simples. La méthode du *lifting scheme* permet également d'effectuer le calcul « in situ », c'est-à-dire sans utilisation de mémoire supplémentaire comme la position des sommets à des niveaux inférieurs. Cette optimisation est rendue possible par une dépendance mutuelle des sommets pairs et impairs d'un même niveau pour le calcul des



masques.

Pour les surfaces, les objets utilisés sont le plus souvent des objets fermés, sans bord. Il est donc rare de devoir appliquer un masque de subdivision spécifique au bord lors de la manipulation d'objets de cette dimension. À l'inverse, dans le cas volumique un objet plongé dans  $\mathbb{R}^3$  n'est pas un volume fermé. Nous manipulons donc des objets à bord. Lors du calcul de la nouvelle géométrie des sommets, il nous faut donc différencier les sommets du bord des sommets intérieurs. D'un point de vue pratique, nous nous servons ici encore de la vision constructive de la version duale du modèle des cartes combinatoires. Il est en effet possible de parcourir la 2-carte formant le volume du bord afin d'appliquer le schéma de subdivision correspondant aux cellules du bord.

Dans ce cadre, nous avons expérimenté deux approches. La première vise à appliquer un schéma de subdivision après une étape du raffinement topologique. La seconde, orientée analyse multirésolution, consiste à utiliser un schéma multirésolution formulé sous la forme d'un banc de filtres. Nous illustrons ces deux approches dans le cadre des 3-cartes multirésolutions afin de montrer les avantages à utiliser notre modèle générique pour construire une hiérarchie volumique. Il donne accès au parcours d'orbites à des niveaux différents, ce qui est nécessaire pour l'application d'un schéma de subdivision ou multirésolution.



**FIGURE 3.29** : Raffinement adaptatif puis application d'un schéma de subdivision d'un niveau  $i$  au niveau  $i + 1$  dans une 3-carte multirésolution.

### 3.3.1 Schéma de subdivision

Nous illustrons d'abord l'application d'un schéma de subdivision qui procède au calcul de la position des sommets de niveau  $i + 1$  suivant des masques

impliquant les sommets de niveau  $i$  tout en parcourant les orbites de niveau  $i$ . Cette méthode est notamment utilisée dans le cadre adaptatif où le voisinage de niveau  $i+1$  n'est pas accessible dans les zones frontières entre deux niveaux.

Nous illustrons les algorithmes de calcul au sein d'une 3-carte multirésolution pour le schéma de MacCracken et Joy [86]. Une application adaptative de ce schéma sur un maillage composé de quatre hexaèdres, deux prismes à base triangulaire et deux prismes à base pentagonale, est illustrée par la figure 3.29. Ce schéma de subdivision consiste en une extension du schéma de subdivision de Catmull-Clark [20]. Nous montrons comment les algorithmes de calcul de la position d'un sommet pour ce schéma peuvent être implantés dans une structure de données issue des 3-cartes multirésolutions.

Les masques utilisés par ce schéma sont illustrés par les dessins des figures 3.31, 3.32, 3.33 et 3.34 pour les sommets intérieurs et par les dessins de la figure 3.30 pour les sommets du bord. Les couleurs sur ces figures représentent les différentes orbites qui interviennent lors du calcul. La couleur verte correspond aux sommets pairs et impairs dont on cherche à calculer la position. L'orbite topologique de ces sommets existe car l'étape de raffinement a eu lieu précédemment. La couleur bleue correspond à l'orbite de niveau  $i$  dont on doit parcourir des cellules incidentes afin de calculer la nouvelle position. Enfin, les variations de la couleur rouge correspondent aux cellules incidentes à l'orbite en bleu nécessaires au calcul de la nouvelle position.

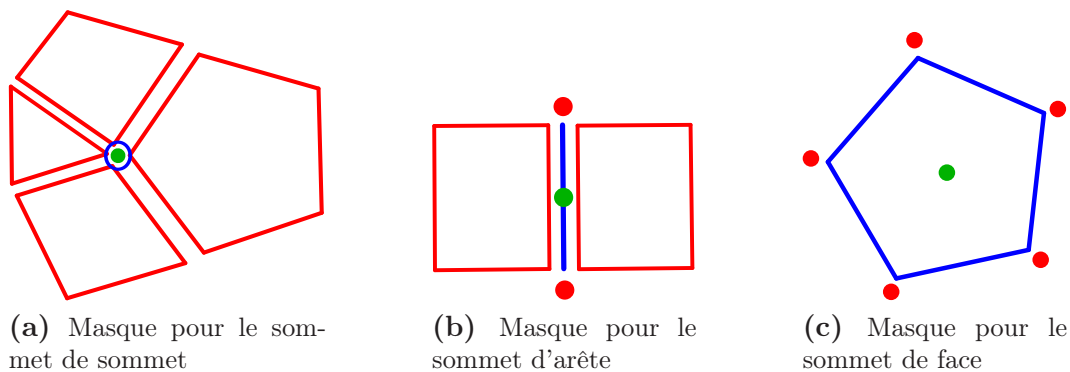


FIGURE 3.30 : Masques utilisés pour le calcul du schéma de subdivision de MacCracken et Joy pour les sommets du bord.

Afin d'organiser les différents masques nous introduisons la notion de « sommet d'orbite ». Cette notion exprime l'idée de sommet inséré au niveau  $i+1$  dans une orbite de niveau  $i$ . Il y a quatre types de sommet pour décrire les masques présentés ci-dessous. Un sommet de sommet est un sommet pair, c'est-à-dire un sommet existant au niveau  $i$  dont on recalcule la position au niveau  $i+1$ . Un sommet d'arête est un sommet impair inséré au niveau  $i+1$  dans une

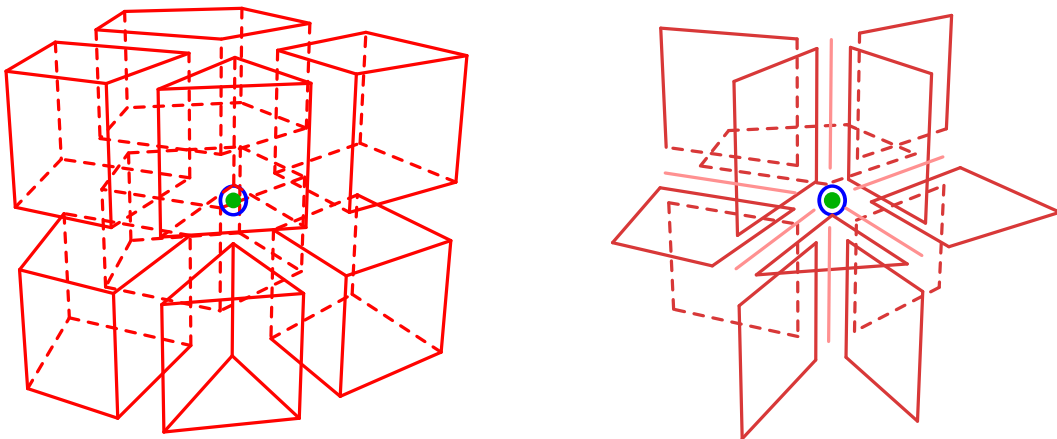
arête de niveau  $i$ . Un sommet de face est un sommet impair inséré au niveau  $i + 1$  dans une face de niveau  $i$ . Ce sommet est donc uniquement présent dans un schéma de subdivision s'appuyant sur un raffinement par quadrisection de faces polygonales. Enfin un sommet de volume est un sommet impair inséré au niveau  $i + 1$  dans un volume de niveau  $i$ .

### Sommet de sommet

Étant un schéma approximant, il faut mettre à jour la position des sommets pairs. Les masques des figures 3.30a et 3.31 servent au calcul de la nouvelle position de ces sommets. Partant d'un brin  $d$  du sommet de niveau  $i + 1$ , on vérifie d'abord si c'est un sommet du bord ou non. Dans le cas d'un sommet intérieur, on commence par récupérer un brin de l'orbite à traverser au niveau  $i$ , ici un brin du sommet de niveau  $i + 1$ . L'algorithme consiste ensuite en une moyenne pondérée de positions calculées à partir :

- du barycentre de chaque volume incident à l'orbite sommet du brin  $d$  (figure de gauche),
- du barycentre de chaque face incidente à l'orbite sommet du brin  $d$  (en rouge sombre sur la figure de droite),
- du barycentre de chaque arête incidente à l'orbite sommet du brin  $d$  (en rouge pâle sur la figure de droite).

Le résultat de ce calcul est affecté au sommet du brin  $d$  au niveau  $i + 1$ .



**FIGURE 3.31** : Masque pour le calcul de la position au niveau  $i + 1$  pour un sommet de niveau  $i$ .

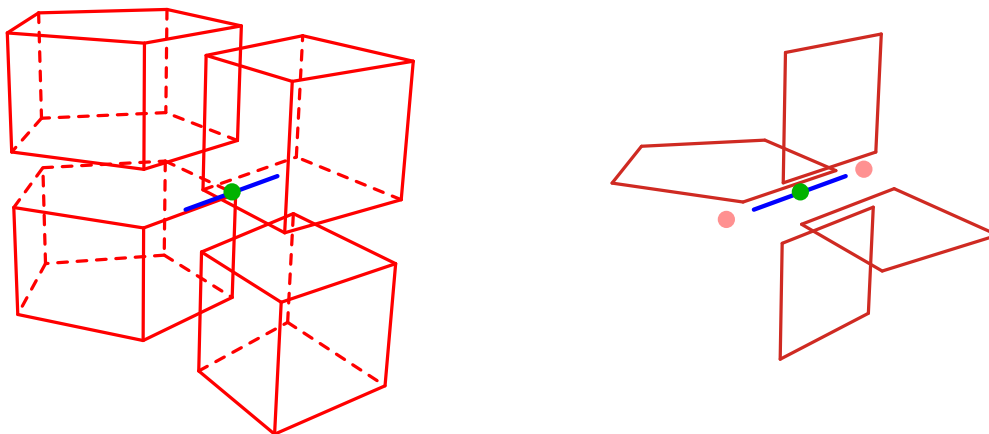
Dans le cas d'un sommet du bord, on cherche un brin  $db$  du sommet sur la face du bord. Nous pouvons ensuite, en utilisant le brin  $db$ , appliquer le schéma de Catmull-Clark au niveau  $i$  pour un sommet dont le voisinage nécessaire est illustré sur la figure 3.30a.

### Sommet d'arête

Les masques des figures 3.30b et 3.32 permettent de calculer la position du sommet inséré au niveau  $i + 1$  dans une arête de niveau  $i$ . Partant d'un brin  $d$  du sommet de niveau  $i + 1$ , on commence par récupérer un brin  $d2$  de l'orbite à traverser au niveau  $i$ , ici un brin de l'arête de niveau  $i$  (un brin de l'arête bleu). On vérifie ensuite, grâce au brin  $d2$ , si c'est une arête du bord ou non. Dans le cas d'une arête intérieure, l'algorithme consiste en une moyenne pondérée de positions calculées à partir :

- du barycentre de chaque volume incident à l'orbite arête du brin  $d2$ ,
- du barycentre de chaque face incidente à l'orbite arête du brin  $d2$ ,
- du barycentre de l'arête du brin  $d2$ .

Le résultat de ce calcul est affecté au sommet du brin  $d$  au niveau  $i + 1$ .



**FIGURE 3.32** : Masque pour le calcul de la position du sommet introduit dans une arête de niveau  $i$ .

Dans le cas d'une arête du bord, on cherche un brin  $db$  de l'arête de la face du bord. Puis, en utilisant le brin  $db$ , nous pouvons appliquer le schéma de Catmull-Clark pour un sommet d'arête dont le voisinage nécessaire est illustré figure 3.30b.

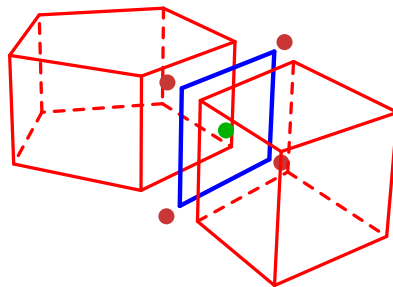
### Sommet de face

Les masques des figures 3.30c et 3.33 permettent de calculer la position du sommet inséré au niveau  $i + 1$  dans une face de niveau  $i$ . Partant d'un brin  $d$  du sommet de niveau  $i + 1$ , on commence par récupérer un brin  $d2$  de l'orbite à traverser au niveau  $i$ , ici un brin de la face de niveau  $i$  (un brin de la face en bleu). On vérifie ensuite grâce au brin  $d2$  si c'est une face du bord ou non. Dans

le cas d'une face intérieure, l'algorithme consiste en une moyenne pondérée de positions calculées à partir :

- du barycentre de chaque volume incident à l'orbite face du brin  $d2$ ,
- du barycentre de la face du brin  $d2$ .

Le résultat de ce calcul est affecté au sommet du brin  $d$  au niveau  $i + 1$ .

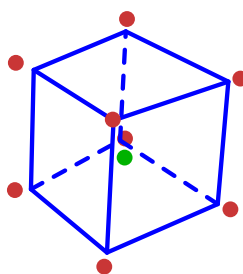


**FIGURE 3.33** : Masque pour le calcul de la position du sommet introduit dans une face de niveau  $i$ .

Dans le cas d'une face du bord, on cherche un brin  $db$  de la face du bord. Puis, en utilisant le brin  $db$ , nous pouvons appliquer le schéma de Catmull-clark pour un sommet de face dont le voisinage nécessaire est illustré figure 3.30c.

### Sommet de volume

Le dernier masque, illustré par la figure 3.34, permet de calculer la position du sommet inséré au niveau  $i + 1$  dans un volume de niveau  $i$ . Partant d'un brin  $d$  du sommet de niveau  $i + 1$ , on commence par récupérer un brin  $d2$  de l'orbite à traverser au niveau  $i$ , ici un brin du volume de niveau  $i$  (un brin du volume en bleu). Il n'y a pas de cas du bord à traiter ici. Nous pouvons directement appliquer l'algorithme qui consiste en une moyenne pondérée calculée à partir des sommets du volume du brin  $d2$ . Le résultat de ce calcul est affecté au sommet du brin  $d$  au niveau  $i + 1$ .



**FIGURE 3.34** : Masque pour le calcul de la position du sommet introduit dans un volume de niveau  $i$ .

### 3.3.2 Schéma multirésolution

Nous illustrons maintenant l'application d'un schéma multirésolution formulé sous la forme d'un banc de filtres. Cette méthode a l'avantage de ne nécessiter qu'un plongement par sommet pour tous les niveaux de résolution où ce sommet existe. Nous illustrons l'application d'un schéma multirésolution qui procède au calcul de la position des sommets de niveau  $i + 1$  suivant des masques impliquant les sommets de niveau  $i + 1$  en parcourant les orbites de niveau  $i$ .

Nous illustrons les algorithmes de synthèse de la géométrie au sein d'une 3-carte multirésolution pour le schéma de Bertram [5]. Une application de ce schéma sur un maillage est illustrée par la figure 3.5. Nous montrons comment les algorithmes de calcul de la position d'un sommet pour ce schéma peuvent être implantés dans une structure de données issue de 3-cartes multirésolutions.

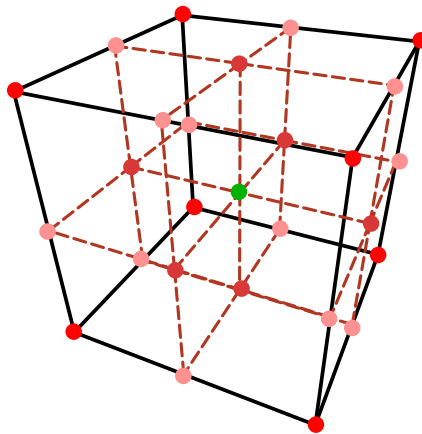
Dans les illustrations suivantes, le niveau  $i$  est représenté par des traits plein noir et le niveau  $i + 1$  est représenté par des traits pointillés rouge. La couleur verte correspond au sommet de niveau  $i + 1$  dont on cherche à calculer la position. Les variations de la couleur rouge correspondent aux différents sommets incidents nécessaires au calcul de la nouvelle position.

#### Sommets impairs

Le filtre appliqué aux sommets impairs est découpé en trois parties, une pour chaque type de sommets insérés. La première partie, illustrée par la figure 3.35, permet de calculer la position du sommet inséré dans un volume de niveau  $i$ . L'algorithme consiste, pour chaque volume de niveau  $i$ , à ajouter à la position du sommet du centre au niveau  $i + 1$  une contribution de ses voisins. Cette contribution consiste en une moyenne pondérée formée par les barycentres respectifs de :

- l'ensemble des sommets de niveau  $i + 1$  recopiés du niveau  $i$  incidents au volume (sommets en rouge),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les arêtes de niveau  $i$  incidentes au volume (sommets en rose pâle),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les faces de niveau  $i$  incidents au volume (sommets en rouge foncé).

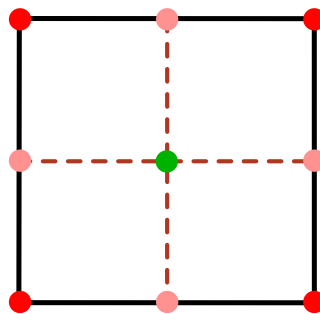
La seconde partie, illustrée par la figure 3.36, permet de calculer la position du sommet inséré dans une face de niveau  $i$ . L'algorithme consiste, pour chaque face de niveau  $i$ , à ajouter à la position du sommet de la face au niveau  $i + 1$



**FIGURE 3.35** : Masque pour le calcul de la position du sommet introduit dans un volume de niveau  $i$ .

une contribution de ses voisins. Cette contribution consiste en une moyenne pondérée formée par les barycentres respectifs de :

- l'ensemble des sommets de niveau  $i + 1$  copiés du niveau  $i$  incidents à la face (sommets en rouge),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les arêtes de niveau  $i$  incidentes à la face (sommets en rose pâle).



**FIGURE 3.36** : Masque pour le calcul de la position du sommet introduit dans une face de niveau  $i$ .

La troisième partie, illustrée par la figure 3.37, permet de calculer la position du sommet inséré dans une arête de niveau  $i$ . L'algorithme consiste, pour chaque arête de niveau  $i$ , à ajouter à la position du sommet de l'arête au niveau  $i + 1$  une contribution de ses voisins. Cette contribution consiste en une moyenne pondérée formée par le barycentre des sommets incidents à l'arête.



**FIGURE 3.37** : Masque pour le calcul de la position du sommet introduit dans une arête de niveau  $i$ .

## Sommets pairs

Étant un schéma approximant, il faut mettre à jour la géométrie des sommets pairs. Ce schéma possède donc un filtre pour les sommets pairs, également découpé en trois parties, une pour chaque type de sommets à mettre à jour.

La première partie permet de calculer la position du sommet de niveau  $i$  au niveau  $i + 1$ . L'algorithme consiste, pour chaque sommet de niveau  $i$ , à ajouter à la position du sommet au niveau  $i + 1$  une contribution de ses nouveaux voisins. Cette contribution est différente si le sommet est sur le bord ou non.

Dans le cas d'un sommet intérieur, illustré à droite sur la figure 3.38, elle consiste en une moyenne pondérée formée par les barycentres respectifs de :

- l'ensemble des sommets de niveau  $i + 1$  insérés dans les volumes de niveau  $i$  incidents au sommet (sommets en rouge),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les faces de niveau  $i$  incidentes au sommet (sommets en rouge foncé),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les arêtes de niveau  $i$  incidentes au sommet (sommets en rose pâle).

Dans le cas d'un sommet du bord, illustré à gauche sur la figure 3.38, elle consiste en une moyenne pondérée formée par les barycentres respectifs de :

- l'ensemble des sommets de niveau  $i + 1$  insérés dans les faces de niveau  $i$  incidentes au sommet (sommets en rouge).
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les arêtes de niveau  $i$  incidentes au sommet (sommets en rose pâle).

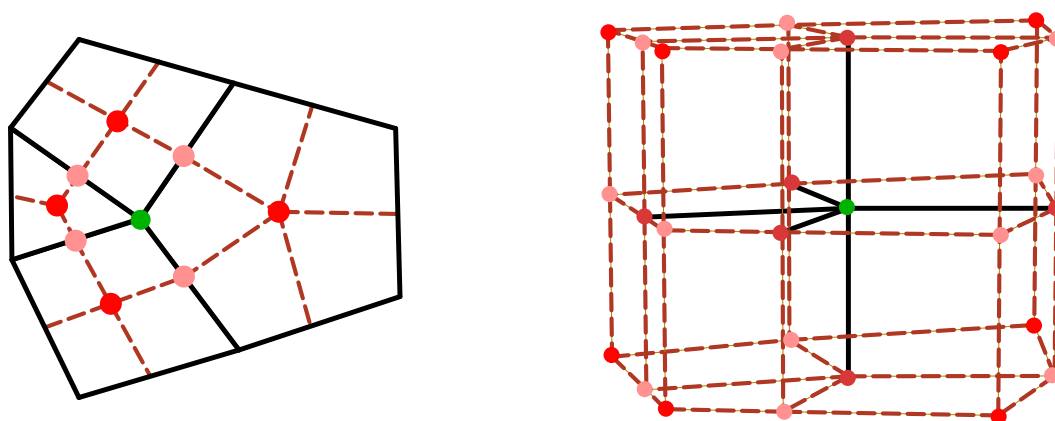


FIGURE 3.38 : Masque pour le calcul de la position au niveau  $i + 1$  pour un sommet de niveau  $i$ .

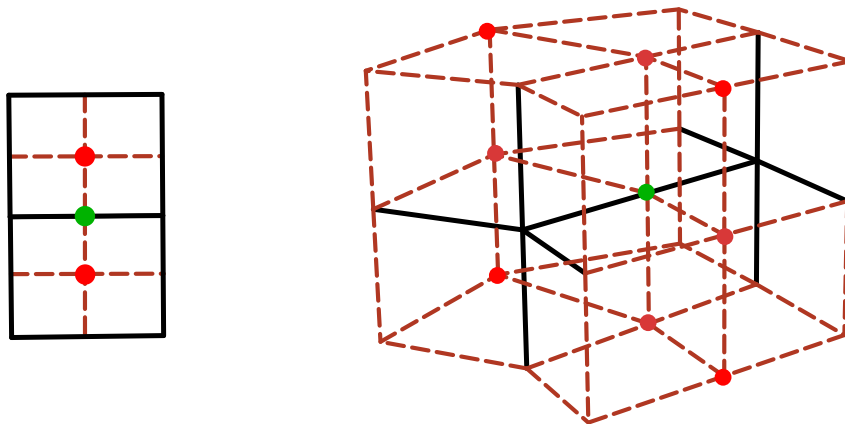
La seconde partie permet de calculer la position d'un sommet inséré dans une arête de niveau  $i$ . L'algorithme consiste, pour chaque arête de niveau  $i$ ,



à ajouter à la position du sommet au niveau  $i + 1$  une contribution de ses nouveaux voisins. Cette contribution est différente si l'arête est sur le bord ou non. Dans le cas d'une arête intérieure, illustrée à droite sur la figure 3.39, elle consiste en une moyenne pondérée formée par les barycentres respectifs de :

- l'ensemble des sommets de niveau  $i + 1$  insérés dans les volumes de niveau  $i$  incidents à l'arête (sommets en rouge),
- l'ensemble des sommets de niveau  $i + 1$  insérés dans les faces de niveau  $i$  incidentes à l'arête (sommets en rouge foncé).

Dans le cas d'une arête du bord, illustré à gauche sur la figure 3.39, elle consiste en une moyenne pondérée formée par l'ensemble des sommets de niveau  $i + 1$  insérés dans les faces de niveau  $i$  incidentes à l'arête.



**FIGURE 3.39** : Masque pour le calcul de la position du sommet introduit dans une arête de niveau  $i$ .

La troisième partie, illustrée par la figure 3.40, permet de calculer la position d'un sommet inséré dans une face de niveau  $i$ . L'algorithme consiste, pour chaque face de niveau  $i$  n'étant pas du bord, à ajouter une moyenne pondérée formée par le barycentre de l'ensemble des sommets de niveau  $i + 1$  insérés dans les volumes de niveau  $i$  incidents à la face.

Un dernier filtre appelé filtre de correction ou de mise à l'échelle multiplie les sommets pairs – les sommets insérés dans les arêtes et les sommets insérés dans les faces – par des coefficients. Les filtres d'analyse, quant à eux, consistent à prendre chacune de ces opérations dans l'ordre inverse et à soustraire ou diviser la contribution des voisins à la place de l'additionner ou de la multiplier.

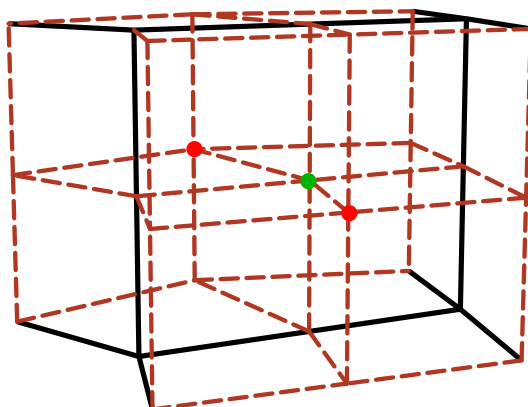


FIGURE 3.40 : Masque pour le calcul de la position du sommet introduit dans une face de niveau  $i$ .

### 3.4 Génération adaptative de maillage pour la simulation

Dans le cadre de ce projet de recherche nous avons collaboré avec une équipe du département Électronique du Solide, Systèmes et Photonique du laboratoire ICube. L'un de leurs thèmes de recherche concerne le développement d'un simulateur électro-thermique intégré à un environnement de CAD électrique standard. Ce simulateur sera utilisé pour effectuer l'analyse électro-thermique nécessaire à la validation des performances, dans leur environnement, de systèmes électroniques intégrés planaires ou 3D. Le comportement électrique est modélisé en utilisant une approche par éléments finis à base d'éléments cartésien en utilisant des méthodes de simulation multirésolution (ou multi-grille). Le réseau thermique 3D, quant à lui, est construit à partir des informations stockées dans le schéma électrique à savoir : la dimension de la puce électronique, l'emplacement des composants, etc.

Ainsi, leur problématique est double. D'une part, le langage interprété disponible dans leur environnement de simulation ne permet pas d'obtenir des performances en temps raisonnable pour la création et la subdivision du maillage. D'autre part, l'utilisateur doit retoucher manuellement le maillage généré après subdivision car il ne correspond pas forcément au maillage souhaité. Ceci est dû au fait qu'un critère de subdivision adaptative est difficile à implanter dans l'environnement de simulation.

Ainsi, pour obtenir un nombre minimum d'éléments tout en gardant une bonne précision, le maillage doit être fin près des zones ayant des changements significatifs dans le profil de température (c'est-à-dire les zones avec une densité de puissance élevée) et grossier dans les zones où le profil de température est plat ou à faible gradient. Par la suite, les zones où le maillage est fin seront

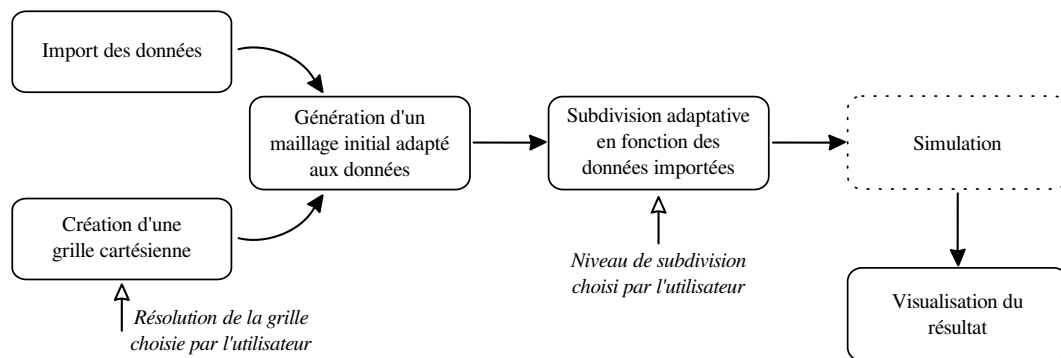


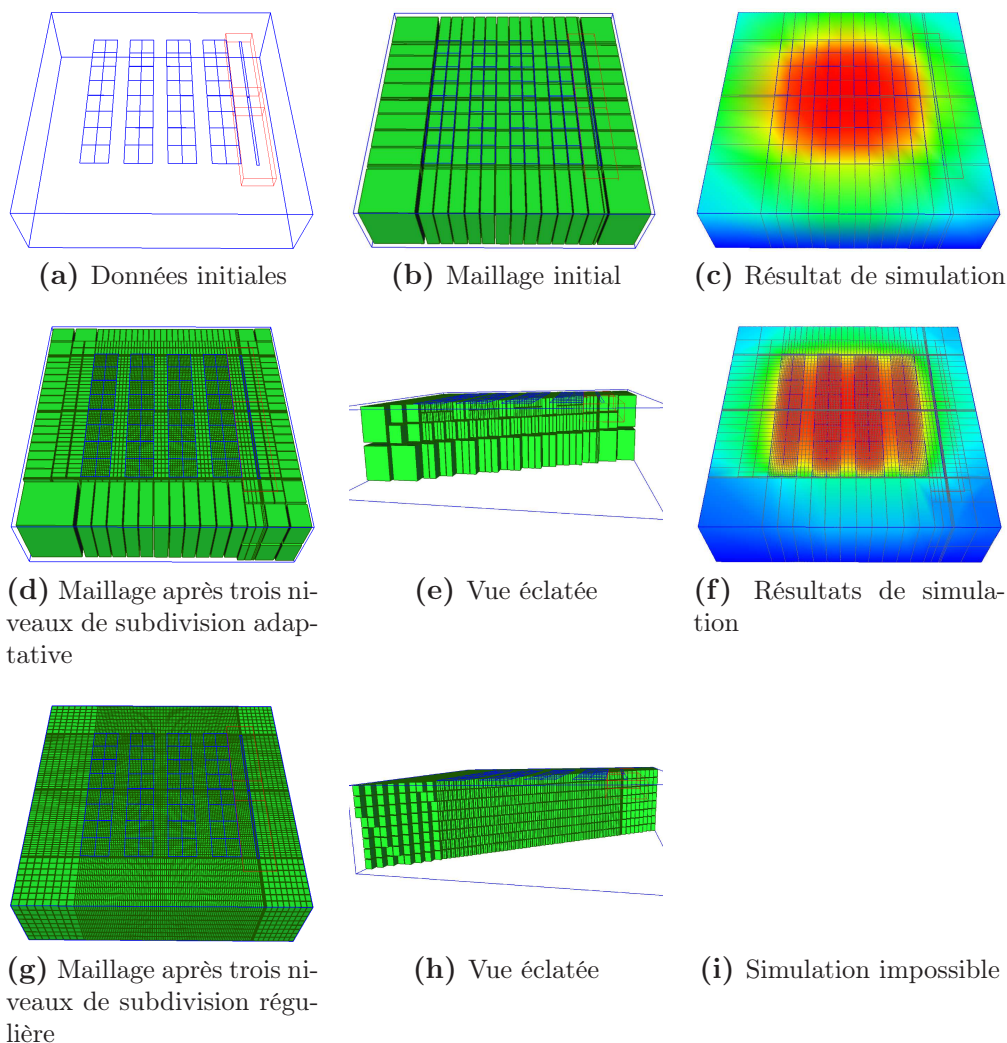
FIGURE 3.41 : Diagramme de flux des données.

désignées par des zones d'influence des composants.

Nous avons utilisé nos méthodes de subdivision adaptative pour la génération de maillage dont les zones d'intérêts sont connues a priori comme étant les zones d'influences des composants. Remarquons que l'utilisation d'un maillage ayant des jonctions en « T » est possible grâce à l'utilisation de méthodes de simulation multi-grilles. Nous avons donc mis au point la méthode visible sur la figure 3.41. Cette méthode se décompose en plusieurs étapes. D'abord, on importe les données qui sont constituées de différents types de boîtes englobantes. En parallèle, il y a création d'un maillage formant une grille régulière dont la résolution est choisie par l'utilisateur. Puis, par des étapes de raffinements successifs, ces deux données nous permettent d'obtenir un maillage initial adapté aux données d'entrées. L'utilisateur choisit ensuite le niveau de résolution qu'il estime nécessaire dans les zones d'influences des composants. Le maillage est ensuite subdivisé en tenant compte de ce critère. Enfin, le maillage est fourni au simulateur qui exécute la simulation sur un certain nombre de pas de temps. Les données résultats de la simulation sont extraites puis plongées sur les sommets du maillage à des fins de visualisation.

Nous montrons ici le résultat des expérimentations mené en utilisant le circuit intégré, servant à valider expérimentalement le simulateur, développé pendant la thèse de Jean-Christophe Krencker [72]. Ce circuit intégré est composé de deux résistances placées de part et d'autre d'une matrice  $4 \times 4$  de capteurs de températures. La figure 3.42 illustre le processus utilisé dans cette application en partant des données à importer (la première colonne) à la visualisation des résultats de la simulation (dernière colonne).

Sur la première ligne de la figure nous pouvons d'abord observer les données à importer constituées d'un ensemble de boîtes englobantes (figure 3.42a). L'emplacement des composants électroniques sur le circuit est illustré par les boîtes englobantes bleues et les zones d'influences sont illustrées en rouge.



**FIGURE 3.42 :** Illustration du pipeline de l'application. La première ligne montre les données importées, le maillage généré à partir de ces données et le résultat de la simulation sur ce maillage. La seconde ligne montre les mêmes informations après trois niveaux de subdivision adaptative. La troisième ligne montre ces informations après trois niveaux de subdivision régulière.

L'algorithme commence par générer un maillage adapté au données importées (576 sommets et 255 volumes) et dans lequel sont plongées les informations concernant les composants et les zones d'influences (figure 3.42b). Enfin, le maillage est exporté et utilisé par le simulateur. À des fins de visualisation nous avons importé les résultats de la simulation pour chaque pas de temps sur chaque sommet. Une carte de couleur pour un pas de temps est visible sur la figure 3.42c.

La seconde ligne montre le maillage résultant après trois niveaux de sub-

division adaptative des polyèdres inclus dans les zones d'influences. Une vue générale du maillage est montrée par la figure 3.42d et une vue en coupe par la figure 3.42e. Ce maillage contenant 29510 sommets et 19218 volumes peut facilement être simulé dans l'environnement de CAD. En outre, en comparaison au maillage initial, nous pouvons visualiser sur la figure 3.42f que la subdivision adaptative améliore la précision du résultat de la simulation.

La troisième ligne montre le maillage résultant après trois niveaux de subdivision régulière du maillage. Une vue générale du maillage est montrée par la figure 3.42g et une vue de coupe par la figure 3.42h. Ce type de maillage contenant 149193 sommets et 130560 volumes est trop volumineux pour être utilisé par le simulateur. Il n'y a donc pas de résultats à visualiser.

Grâce à notre modèle supportant la subdivision adaptative et l'efficacité des parcours topologiques, le nombre de sommets décroît de 80% et le nombre de volumes décroît lui de 85%. Notre méthode autorise donc la simulation fine de ce type de circuit intégré dans un environnement de CAD électrique standard sans intervention humaine sur le maillage.

### 3.5 Autre raffinement : $\sqrt{3}$ volumique

Le schéma  $\sqrt{3}$  volumique proposé par Umlauf dans [15, 16] permet une subdivision de maillages tétraédriques ne générant que des éléments tétraédriques, à l'inverse du raffinement tétraédrique/octaédrique présenté section 3.2. Ce mécanisme de subdivision trouve son intérêt dans des applications nécessitant l'utilisation univoque de tétraèdres comme c'est le cas pour certaines simulations numériques. Le raffinement se décompose en deux types d'opérations : les opérations de basculement ou « swap » et les opérations de raffinement que l'on appellera « split ». Une opération de basculement modifie localement la connectivité sans changer le nombre ou la position des sommets du maillage [74]. En comparaison de méthodes qui opèrent par une bisection d'arête, le nombre de tétraèdres générés à l'avantage de croître par un facteur six pour une étape de subdivision régulière contre un facteur huit pour la bisection.

La figure 3.43 illustre le principe de ce raffinement dans le cas régulier sur un maillage tétraédrique. Les traits jaunes sur les images illustrent la relation volumique  $\phi_3$  liant les tétraèdres entre eux. D'abord, une subdivision de tous les tétraèdres du maillage en quatre tétraèdres, appelée *split 1-4*, est effectuée en insérant un nouveau sommet en leurs centres (figure 3.43-B). Toutes les faces intérieures des anciens tétraèdres subissent ensuite une opération de basculement d'une configuration à deux tétraèdres vers une à trois tétraèdres,

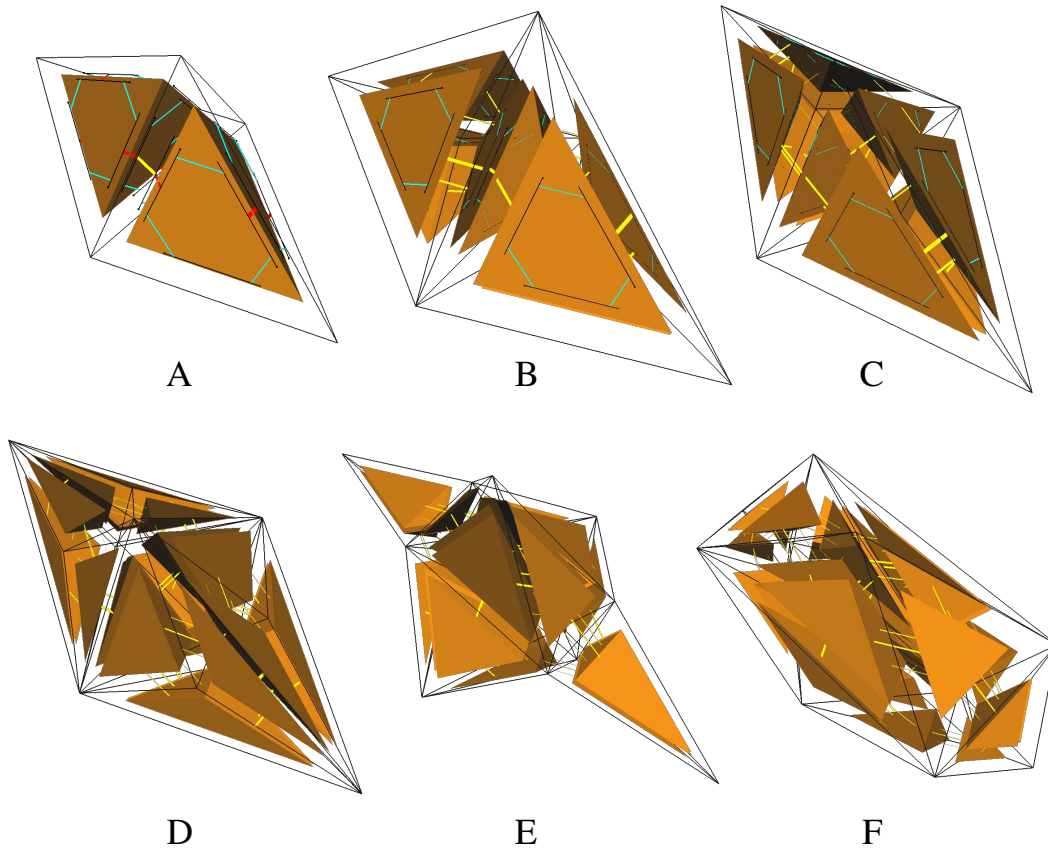


FIGURE 3.43 : Exemple de subdivision d'un niveau  $i$  au niveau  $i + 1$  d'un maillage tétraédrique.

c'est-à-dire un *swap 2-3* (figure 3.43-C). Les tétraèdres du bord, quant à eux, sont subdivisés en trois tétraèdres par l'insertion d'un nouveau sommet dans la face du bord de chacun de ces tétraèdres, c'est-à-dire un *split 1-3* (figure 3.43-D). Les arêtes du bord sont ensuite basculées en utilisant une opération de basculement généralisé (appelée *swapGen 3-2*) opérant le changement d'une configuration de  $m$  tétraèdres vers  $2(m - 2)$  tétraèdres (figure 3.43-E). Nous allons détailler toutes ces opérations dans le cadre de la construction d'un nouveau niveau de résolution à l'aide des 3-cartes multirésolutions.

Nous verrons également que ce raffinement peut être utilisé afin de construire un niveau de résolution de manière adaptative. Les raffinements utilisés sont les mêmes que dans le cas régulier mais ne sont pas exécutés dans les mêmes conditions.

Dans l'article [15] la géométrie de la subdivision est calculée par un processus d'optimisation. Ce processus consiste à placer un sommet intérieur au centre de ces 1-voisins, puis à changer de configuration à l'aide de l'opération

de basculement généralisé jusqu'à satisfaire au mieux un critère de qualité sur les tétraèdres. La géométrie des sommets du bord, quant à elle, se calcule grâce au masque de subdivision du schéma  $\sqrt{3}$  surfacique (figure 3.43-F). Nous n'avons pas opté pour ce processus d'optimisation qui rend la connectivité d'un nouveau niveau de résolution non déterministe.

La terminologie introduite précédemment sur la parité des sommets du maillage lors de la subdivision reste valide. Les sommets pairs sont donc les sommets du maillage conservés d'un niveau à l'autre. Les sommets impairs sont les nouveaux sommets insérés au centre des tétraèdres intérieurs et au centre des faces pour les tétraèdres du bord. Nous verrons dans la dernière sous-section comment retrouver ces deux types de sommets dans la hiérarchie de maillage.

Nous pouvons noter que ce raffinement peut être appliqué à tout maillage polyédrique dont les polyèdres sont à faces triangulaires. En effet, la première opération consiste à ajouter un point central à un polyèdre afin de former des tétraèdres entre ce point central et les faces du polyèdre de départ. Il est donc par exemple possible d'appliquer ce raffinement à un maillage tétraédrique/octaédrique.

### 3.5.1 Génération régulière

Nous allons illustrer dans l'ordre de leur application les différentes opérations de raffinement entre deux niveaux consécutifs  $i$  et  $i + 1$  dans une 3-carte multirésolution. À l'instar de la construction d'un niveau de subdivision à l'aide de raffinements primaires, la première étape consiste à créer un nouveau niveau de résolution  $i + 1$  comme une copie du niveau courant  $i$  et de considérer comme nouveau niveau courant ce niveau  $i + 1$ .

#### Opération de split 1-4

La figure 3.44 illustre l'opération de split 1-4 d'un tétraèdre dans une 3-carte multirésolution. Cette opération se décompose en plusieurs étapes créant chacune un tétraèdre.

La première étape consiste à utiliser l'opération de découpe de volume sur une face du tétraèdre avec pour chemin de découpe les brins de cette face. Cette opération crée un volume « plat » constitué de deux faces triangulaires.

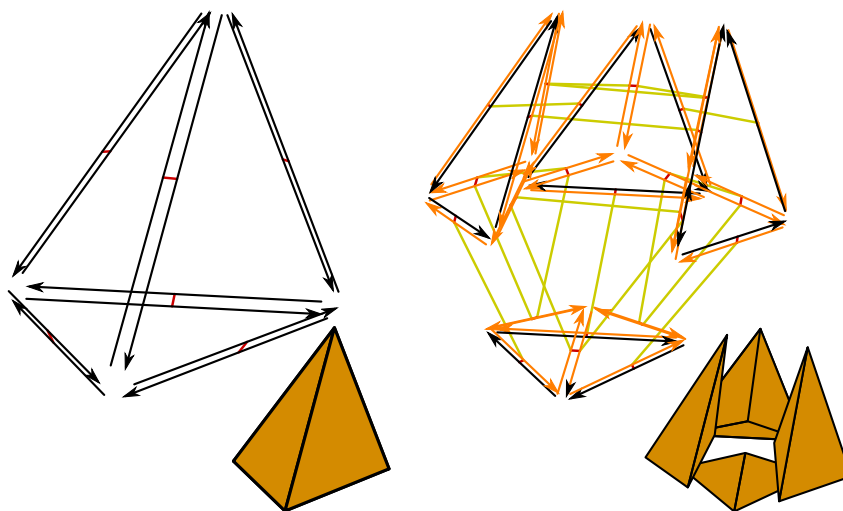


FIGURE 3.44 : Opération de split 1-4 au sein d'une 3-carte multirésolution.

Cette face nouvellement créée est ensuite triangulée en y insérant un point central. Ces deux opérations ont permis de découper un premier tétraèdre à l'intérieur du tétraèdre initial tout en laissant un polyèdre à six faces.

L'étape suivante consiste à utiliser l'opération de découpe de volume entre un brin du sommet nouvellement créé et un brin d'une face du tétraèdre initial, et plus précisément du sommet opposé. Cette opération crée un volume « plat » constitué de trois faces dont une à quatre côtés. L'opération de découpe de face est ensuite utilisée afin de créer le second volume en découpant la face de quatre côtés en deux.

À ce stade de l'opération nous avons deux tétraèdres et un polyèdre à six faces triangulaires. Pour finir, l'opération de découpe de volume est utilisée afin de découper ce polyèdre en deux tétraèdres. Cette opération renvoie un brin de l'orbite sommet ainsi créée.

### Opération de swap 2-3

La figure 3.45 illustre l'opération de swap 2-3 d'un tétraèdre dans une 3-carte multirésolution. Cette opération se décompose en plusieurs opérations de bas niveau transformant une configuration à deux tétraèdres en une configuration à trois tétraèdres. Afin de créer une hiérarchie cohérente, les brins qui, au niveau  $i$ , appartiennent à une face, doivent être associés à l'orbite arête après l'opération de basculement.



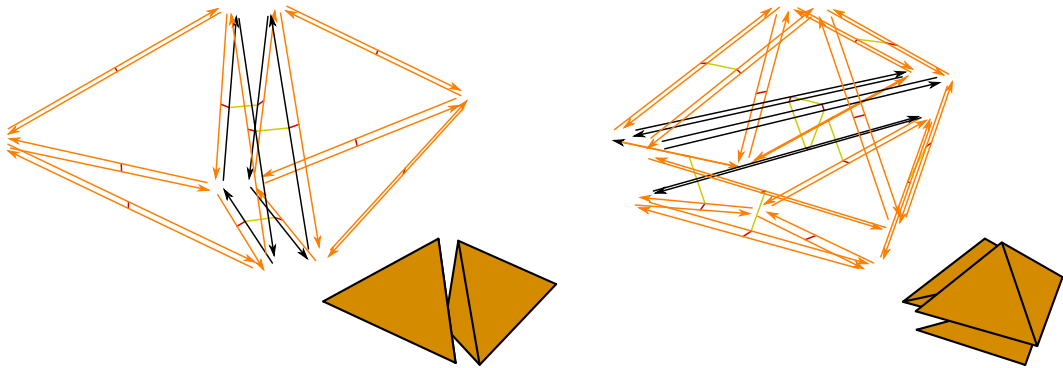


FIGURE 3.45 : Opération de swap 2-3 au sein d'une 3-carte multirésolution.

La première opération consiste à fusionner les deux tétraèdres le long de leur face commune. Cette face commune n'est pas supprimée lors de l'utilisation de cette opération dans le cadre multirésolution.

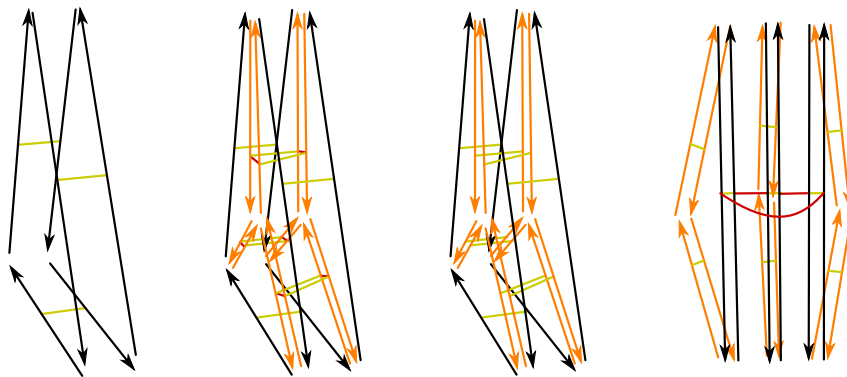


FIGURE 3.46 : Opération de transformation d'une orbite face en une orbite arête dans une 3-carte multirésolution.

On utilise une opération qui va permettre de transformer les brins qui forment deux faces en brins formant une arête. Cette opération est illustrée par la figure 3.46. D'abord la face est triangulée au niveau  $i + 1$  puis, chaque brin est déconnecté de son voisin par  $\phi_2$ . Les brins de niveau  $i$  sont recousus par  $\phi_2$  pour former la nouvelle orbite arête. Pour finir, cette permutation de faces est insérée dans le volume à six faces laissé libre précédemment. Pour être plus précis, chaque brin de niveau  $i + 1$  laissé libre par  $\phi_2$  est inséré donc son homologue du polyèdre à 6 faces. Cette opération renvoie un brin de l'orbite arête ainsi créée.

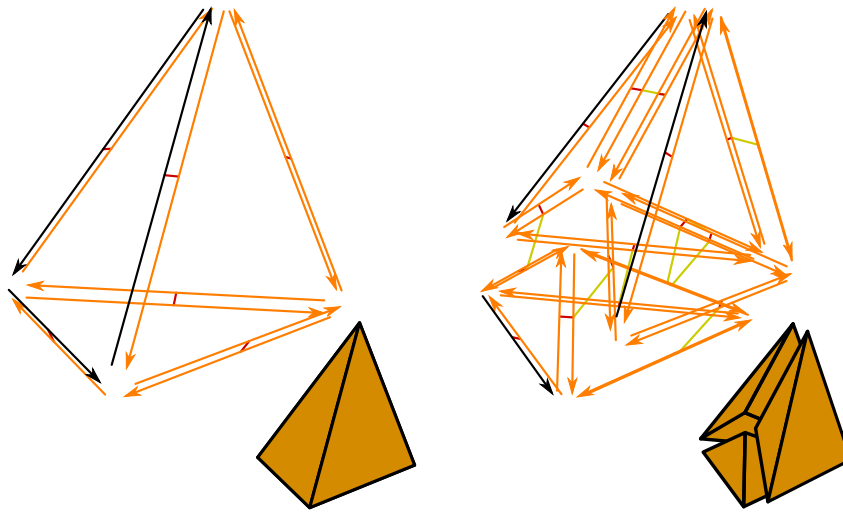


FIGURE 3.47 : Opération de split 1-3 au sein d'une 3-carte multirésolution.

### Opération de split 1-3

La figure 3.47 illustre l'opération de split 1-3 d'un tétraèdre dans une 3-carte multirésolution. À l'instar de l'opération de split 1-4, cette opération se décompose en plusieurs étapes créant chacune un tétraèdre.

D'abord, on applique l'opération de triangulation à la face dans laquelle un sommet doit être inséré. Il en résulte un polyèdre à six faces triangulaires.

Comme précédemment, on découpe un premier polyèdre « plat » à partir d'une face du tétraèdre initial entre le sommet inséré dans une face et le sommet opposé. La face intérieure est ensuite coupée en deux grâce à l'opération de découpe de face. Ces deux opérations ont permis de découper un premier tétraèdre à l'intérieur du tétraèdre initial tout en laissant un polyèdre à six faces.

Pour finir, l'opération de découpe de volume est utilisée afin de découper ce polyèdre en deux tétraèdres. Cette opération renvoie un brin de l'orbite sommet ainsi créée.

### Opération de swap 2-2 et de swap 4-4

La figure 3.48 illustre l'opération de swap 2-2 d'un tétraèdre dans une 3-carte multirésolution. Cette opération se décompose en plusieurs opérations

de bas niveau transformant une configuration à deux tétraèdres en une autre configuration à deux tétraèdres en faisant basculer une des arêtes incidentes aux deux tétraèdres. En modifiant la connectivité entre deux tétraèdres de la sorte, cette opération ne peut être utilisée que sur deux tétraèdres du bord du maillage.

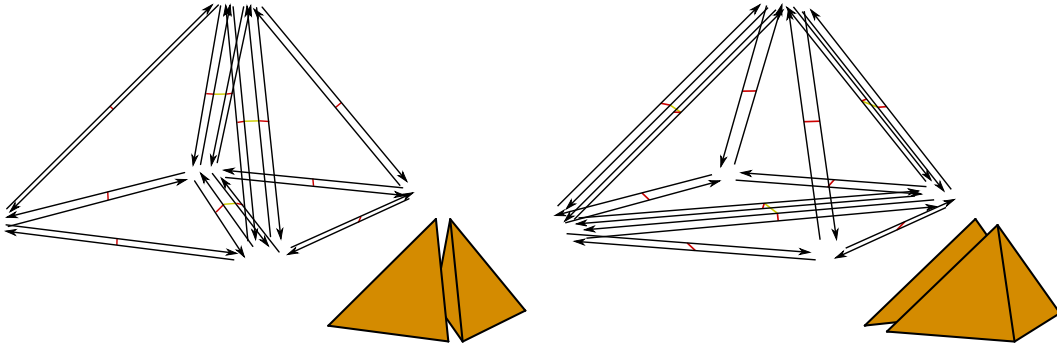


FIGURE 3.48 : Opération de swap 2-2 au sein d'une 3-carte multirésolution.

La première opération consiste à fusionner les deux tétraèdres le long de leur face commune. Cette face commune n'est pas supprimée lors de l'utilisation de cette opération dans le cadre multirésolution. L'arête choisie au départ est basculée par une opération de basculement d'arête. Enfin, l'opération de découpe de volume est utilisée le long du chemin d'arête traversant le volume. Cette opération insère la face déconnectée lors de l'opération de fusion. Cette opération renvoie un brin de l'orbite arête modifiée.

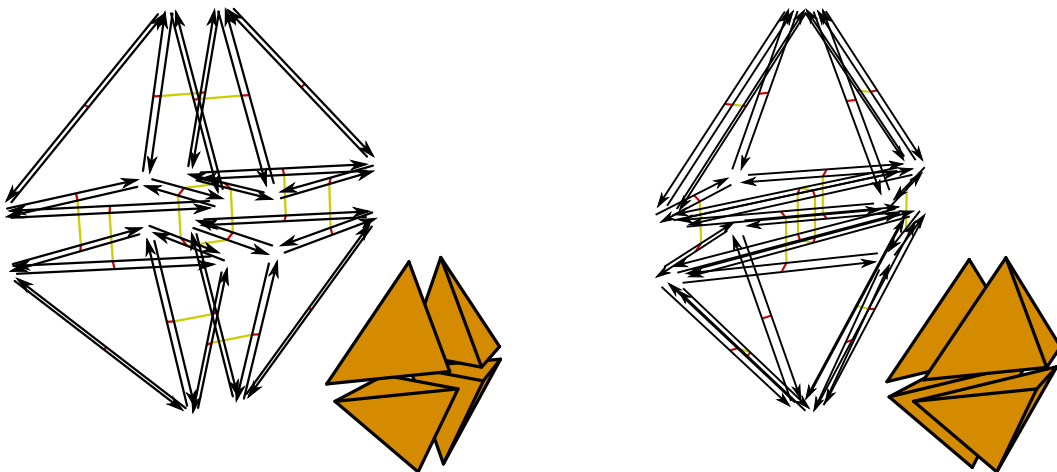
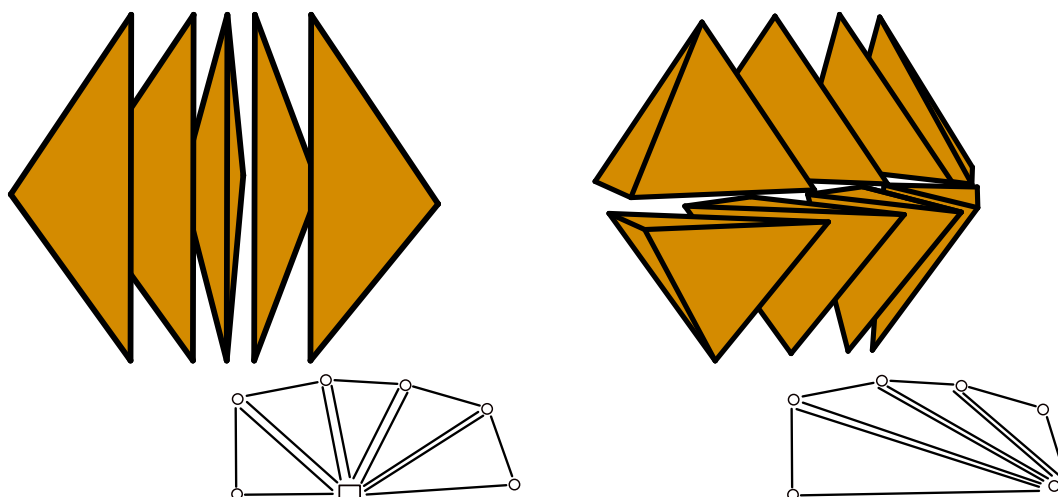


FIGURE 3.49 : Opération de swap 4-4 au sein d'une 3-carte multirésolution.

L'opération de swap 4-4, illustrée par la figure 3.49, est utilisée sur des arêtes intérieures. Elle transforme une configuration de quatre tétraèdres autour d'une arête en une autre configuration à quatre tétraèdres en faisant basculer l'arête incidente aux quatre tétraèdres. La première opération consiste à

découdre deux tétraèdres de leurs voisins afin de se ramener à deux configurations de deux tétraèdres. Nous pouvons ensuite utiliser l'opération de swap 2-2 sur chacune de ces deux configurations. Enfin, il faut recoudre les deux nouvelles paires de tétraèdres générées afin de reconstituer une configuration de quatre tétraèdres. Cette opération renvoie également un brin de l'orbite arête modifiée.

### Opération de swap généralisé



**FIGURE 3.50** : Opération de swap généralisé 3-2 au sein d'une 3-carte multirésolution pour une arête du bord. Les figures du bas représentent une vue de dessus de la permutation de faces. Les faces sont dessinées en trait noir, les sommets en cercles noirs et l'arête basculée en un carré noir.

Les opérations de basculement présentées ci-dessus peuvent être généralisées en deux opérations que nous illustrons par la figure 3.50.

La première, appelée swap généralisé 3-2 (ou suppression d'arête), s'applique à une arête reliant deux sommets  $a$  et  $b$  et dont le nombre de tétraèdres incident est supérieur ou égal à 3. Cette opération supprime l'arête entre les deux sommets  $a$  et  $b$  et la remplace par un éventail de faces. Ainsi, soit  $n$  le nombre de tétraèdres autour de l'arête, cette opération génère  $(n - 2)$  faces et  $2(n - 2)$  tétraèdres pour une arête intérieure et  $2(n - 1)$  tétraèdres pour une arête du bord.

La seconde opération, appelée swap généralisé 2-3 (ou suppression de faces), s'applique à un ensemble de faces comprises entre deux sommets  $a$  et  $b$ . Cet ensemble de faces est supprimé et une nouvelle arête est insérée entre les deux

sommets à la place des faces. Cette opération remplace  $2(n - 2)$  tétraèdres par  $n$  tétraèdres.

Il a été montré [108] que ces opérations de swap généralisé peuvent être décomposées en une suite d'opérations de swap 2-3 ou 3-2 maintenant la topologie intermédiaire cohérente. Seulement, cette suite d'opération peut générer des tétraèdres géométriquement dégénérés ou inversés. Il a ensuite été montré [75] qu'il existe toujours une séquence d'opérations de swap 2-3, 3-2, 2-2, 4-4 remplaçant le swap généralisé qui ne crée pas de tétraèdres dégénérés ou inversés. L'algorithme 1 décrit cette séquence d'opérations pour le basculement généralisé 3-2 qui est utilisé dans notre raffinement.

---

**Algorithme 1** Algorithme effectuant le swapGen 3-2.

---

```

1: fonction SWAPGEN3-2( $d$ )
2:    $n \leftarrow edgeDegree(d)$  ▷ Calcul le degré de l'arête
3:   si  $n \geq 4$  alors
4:      $dit \leftarrow d$ 
5:     si  $isBoundaryEdge(dit)$  alors ▷ Dans le cas d'une arête du bord
6:       pour  $i = 0, i < n - 2, i = i + 1$  faire
7:          $dit \leftarrow phi2(swap2-3(dit))$ 
8:       fin pour
9:        $swap2-2(dit)$ 
10:    sinon ▷ Dans le cas d'une arête intérieure
11:      pour  $i = 0, i < n - 4, i = i + 1$  faire
12:         $dit \leftarrow phi2(swap2-3(dit))$ 
13:      fin pour
14:       $swap4-4(phi3(phi2(dit)))$ 
15:    fin si
16:    sinon si  $n = 3$  alors
17:       $swap3-2(d)$ 
18:    sinon
19:       $swap2-2(d)$ 
20:    fin si
21: fin fonction

```

---

### 3.5.2 Génération adaptative

À l'instar de la génération adaptative d'un niveau par subdivision primaire, le raffinement  $\sqrt{3}$  volumique permet également de subdiviser de manière adaptative un maillage. Cette méthode a deux avantages. Elle ne génère que des éléments tétraédriques, à l'inverse des raffinements primaires. Elle limite le

nombre de tétraèdres générés en comparaison de l'approche régulière.

La règle concernant le raffinement adaptatif est simple et étend la terminologie de [68] pour le raffinement  $\sqrt{3}$  surfacique. Il existe deux types de tétraèdres que l'on appellera pair et impair. Les tétraèdres dits impairs sont ceux obtenus par la subdivision en quatre d'un tétraèdre avant une opération de swap 2-3 pour les faces intérieures ou avant la subdivision en trois pour les tétraèdres du bord. Ces tétraèdres sont à la frontière entre les zones ayant des niveaux de subdivision différents. Les tétraèdres pairs sont tous les autres tétraèdres.

Le raffinement adaptatif nécessite bien évidemment de savoir calculer le niveau de subdivision d'un volume. La définition générale donnée précédemment reste valide dans ce contexte. Cependant, nous sommes dans le cas particulier où toutes les faces sont triangulaires et tous les volumes des tétraèdres. Nous pouvons donc simplifier le calcul du niveau de subdivision d'une cellule. Nous définissons le niveau d'une face triangulaire du maillage comme étant le maximum des niveaux d'insertion de ses brins. Nous définissons de plus le niveau d'un tétraèdre comme étant le maximum des niveaux de ses faces.

En d'autre terme, un tétraèdre impair est alors défini comme étant un tétraèdre dont un tétraèdre voisin est de niveau inférieur. Inversement, tous les autres tétraèdres, c'est-à-dire les tétraèdres dont les tétraèdres voisins sont de niveau supérieur ou égal, sont donc pairs.

La subdivision adaptative d'un tétraèdre se déroule donc en suivant l'algorithme 2.

---

**Algorithme 2** Algorithme de subdivision adaptative d'un tétraèdre par un raffinement  $\sqrt{3}$  volumique.

---

```

1: si t est pair alors
2:   split1-4
3:   pour tout tétraèdre voisin  $t_i$  de t faire
4:     si  $t_i$  est impair et de niveau  $i + 1$  alors
5:       si  $t_i$  est un tétraèdre ayant une face du bord alors
6:         split1-3
7:         swapGen2-3
8:       sinon
9:         swap2-3
10:      fin si
11:    fin si
12:  fin pour
13: fin si

```

---

La figure 3.51 illustre le principe de la subdivision adaptative des tétraèdres intérieurs sur un détail d'un maillage tétraédrique. Cette subdivision suit la règle simple selon laquelle si un tétraèdre est subdivisé mais que son voisin ne l'est pas, leur face commune n'est simplement pas basculée par l'intermédiaire du basculement des deux tétraèdres incidents à cette face. Ainsi, d'abord deux tétraèdres sont sélectionnés pour la subdivision par l'opération de split 1-4. Le tétraèdre commun aux deux est ensuite basculé par l'opération de swap 2-3 (figure 3.51-B). Le tétraèdre bleu sur la figure 3.51-C représente un tétraèdre impair car il est issu d'un tétraèdre subdivisé et son tétraèdre voisin ne l'est pas.

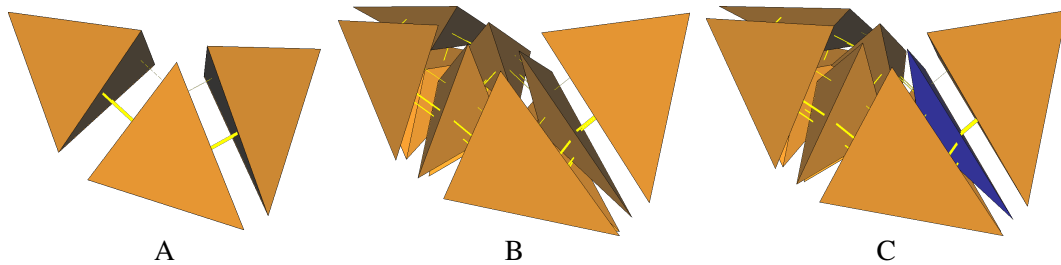


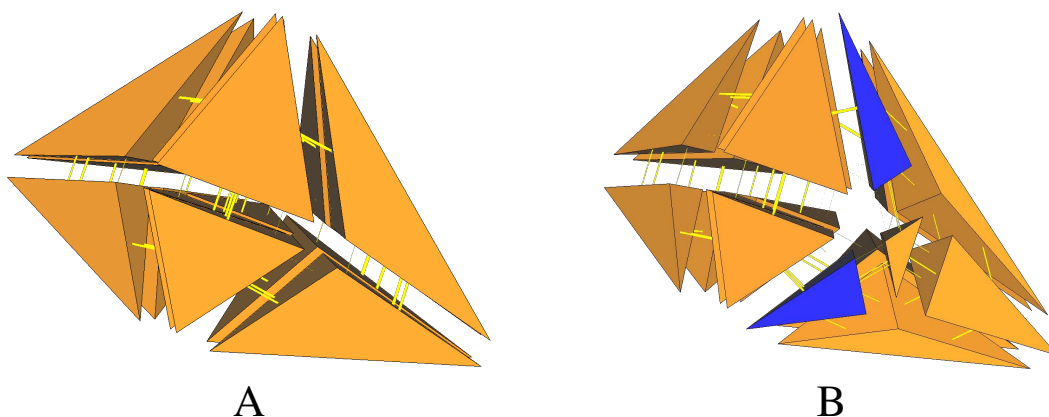
FIGURE 3.51 : Détail d'un pas de raffinement adaptatif avec le raffinement  $\sqrt{3}$  volumique dans le cas intérieur au sein d'une 3-carte multirésolution.

La figure 3.52 illustre le principe de la subdivision adaptative des tétraèdres du bord sur un détail d'un maillage tétraédrique. Dans ce cas, la subdivision suit la règle selon laquelle si un tétraèdre du bord est subdivisé mais que son voisin du bord ne l'est pas, leur arête commune n'est simplement pas basculée par l'intermédiaire du basculement généralisé swapGen 2-3. Ainsi, d'abord deux tétraèdres du bord sont sélectionnés pour la subdivision par l'opération split 1-3 (figure 3.52-B). On remarque que l'arête incidente aux deux tétraèdres subdivisés est basculée. À l'inverse, les deux tétraèdres bleus sur la figure 3.52-B sont des tétraèdres impairs car ils sont issus d'un tétraèdre subdivisé et leur tétraèdre voisin ne l'est pas.

### 3.5.3 Géométrie

Notre travail s'est concentré sur la création et la gestion cohérente d'une hiérarchie de maillages tétraédriques. Nous montrons dans cette sous-section les mécanismes permettant de définir des masques de subdivision en fonction du voisinage local dans une hiérarchie construite par un raffinement  $\sqrt{3}$  volumique multirésolution.

Il est possible de reprendre pour ce raffinement la terminologie des sommets



**FIGURE 3.52** : Détail d'un pas de raffinement adaptatif avec le raffinement  $\sqrt{3}$  volumique dans le cas du bord au sein d'une 3-carte multirésolution.

pairs et des sommets impairs. Les sommets impairs sont les nouveaux sommets insérés au centre des faces et des tétraèdres. Les sommets pairs sont les anciens sommets du maillage, c'est-à-dire les sommets d'origines des faces et des tétraèdres. Nous utilisons un schéma particulier pour les sommets du bord et un autre pour les sommets intérieurs. Nous avons choisi d'adapter la géométrie décrite par l'article [15] pour les sommets intérieurs et nous rappelons que les sommets du bord utilisent le masque du schéma  $\sqrt{3}$  surfacique [67]. Il est bien évidemment possible de définir d'autres coefficients pour ces masques.

À chaque niveau de résolution d'une subdivision régulière les sommets pairs sont toujours composés uniquement de brins nouvellement insérés du fait des opérations utilisées. Les sommets impairs sont eux composés de nouveaux brins et de brins provenant des niveaux précédents. Cette propriété vaut pour les sommets du bord et pour les sommets intérieurs.

La géométrie associée aux sommets du bord est calculée en utilisant les masques illustrés dans la figure 3.53. Le masque de gauche sert au calcul de la position des sommets impairs et celui de droite au calcul la nouvelle position des sommets pairs.

La géométrie associée aux sommets intérieurs est calculée en utilisant les masques illustrés dans la figure 3.54 Un sommet impair (masque de gauche) est placé au barycentre du tétraèdre de niveau précédent. Un sommet pair (masque de droite) est placé au barycentre de ses voisins du niveau courant.

Pour le calcul de la position des sommets impairs, il est nécessaire de savoir calculer le barycentre d'un tétraèdre du niveau précédent. Grâce à notre opération de swap 2-3 qui transforme les brins d'une face en brins d'une arête,



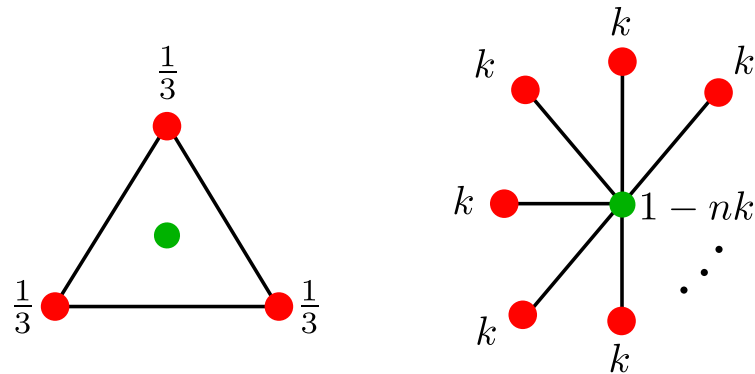


FIGURE 3.53 : Masques de subdivision des sommets du bord. Le poids  $k$  est celui proposé par [67] et est égal à  $\frac{4-2\cos\frac{2\pi}{n}}{9}$ .

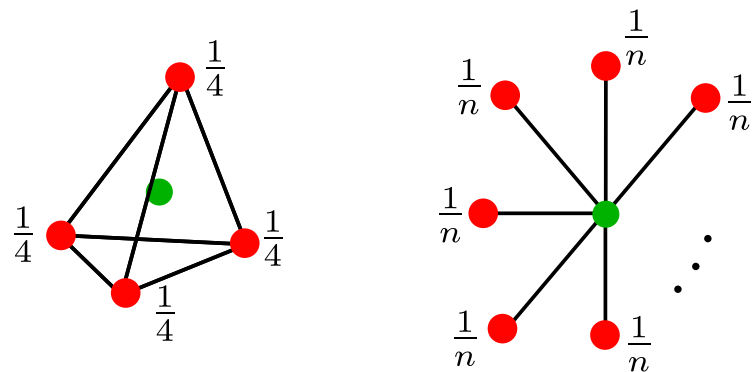


FIGURE 3.54 : Masques de subdivision des sommets de l'intérieur.

le nouveau sommet inséré au centre d'un tétraèdre de niveau  $i$  est composé de brins du niveau précédent appartenant au tétraèdre. Cette remarque vaut également pour les sommets du bord où l'opération de swap généralisé bascule les brins de manière à ce qu'un nouveau sommet soit composé de brins de triangles du niveau précédent. Il suffit alors de récupérer un de ces anciens brins et de se placer au niveau précédent pour calculer le barycentre de la face ou du tétraèdre.

Dans le cas d'une subdivision adaptative, il manque une information pour le calcul des sommets pairs. En effet, si tous les tétraèdres n'ont pas été raffinés comme dans le cadre régulier, il n'est pas possible de retrouver les voisins directement. La position de ce type de sommets doit être calculée grâce au sommets voisins du niveau précédent. Afin de retrouver le niveau précédent, nous pouvons récupérer un ancien brin de ce sommet comme précédemment.

## 3.6 Conclusion

Ce chapitre a été l'occasion de montrer comment les cartes combinatoires peuvent servir de support topologique aux volumes de subdivision multirésolutions. Nous avons d'ailleurs illustré comment le cadre formel des cartes combinatoires nous permet d'envisager de décrire formellement le raffinement primal en dimension quelconque.

Cette représentation nous permet de construire une hiérarchie cohérente d'un niveau au suivant, topologiquement consistante et générique. Elle permet un accès au maillage d'un niveau donné sans reconstruire ni calculer les informations topologiques entre les cellules. Elle permet de disposer à chaque niveau de résolution d'une carte combinatoire complète représentant toutes les cellules et leurs relations d'incidences et adjacences. Ces facultés permettent également de disposer d'attributs pour chaque cellule (sommet, arête, face, volume) à chaque niveau de résolution. Ce sont ces propriétés qui nous différencient d'autres structures de données ou modèles multirésolutions.

Dans le cas d'un schéma de subdivision, ces avantages nous permettent d'accéder au voisinage nécessaire sans avoir à parcourir la hiérarchie et donc sans être dépendant de la profondeur de la structure multirésolution. On peut étendre ce principe à tout type de méthodes de calcul et d'applications.

Nous avons également montré que les cartes combinatoires supportent une grande variété de raffinements aussi bien primaux que spécifiques aux maillages tétraédriques. Ainsi nous avons illustré l'utilisation du modèle pour les schémas primaux de MacCracken et Joy [86], de Bertram [5] ou encore le  $\sqrt{3}$  volumique [15]. Cette capacité à supporter une topologie quelconque nous permet également de gérer sans problème de conformité la topologie d'une subdivision adaptative.

---

# REPRÉSENTATIONS DU MODÈLE

---

## Sommaire

---

|       |  |            |
|-------|--|------------|
| 4.1   | Introduction . . . . .                               | <b>123</b> |
| 4.1.1 | Mise en œuvre monorésolution . . . . .               | 125        |
| 4.1.2 | Analyse comparative de la performance . . . . .      | 126        |
| 4.2   | Représentation multirésolution explicite . . . . .   | <b>129</b> |
| 4.2.1 | Principe . . . . .                                   | 129        |
| 4.2.2 | Complexité temporelle . . . . .                      | 131        |
| 4.2.3 | Complexité spatiale . . . . .                        | 131        |
| 4.2.4 | Mise en œuvre et analyse de la performance . . . . . | 135        |
| 4.3   | Représentation multirésolution implicite . . . . .   | <b>139</b> |
| 4.3.1 | Principe . . . . .                                   | 139        |
| 4.3.2 | Complexité temporelle . . . . .                      | 148        |
| 4.3.3 | Complexité spatiale . . . . .                        | 151        |
| 4.3.4 | Mise en œuvre et analyse de la performance . . . . . | 154        |
| 4.4   | Conclusion . . . . .                                 | <b>161</b> |

---

Ce chapitre expose les différentes représentations que nous proposons pour les cartes combinatoires multirésolutions. Nous commençons par introduire le protocole d'évaluation de ces représentations. Puis, pour chaque représentation, nous explicitons son principe de fonctionnement, avant de procéder à une analyse de la complexité théorique en temps et en espace ainsi qu'une analyse de la complexité pratique avec des mesures de performances des algorithmes.

## 4.1 Introduction

Dans les chapitres précédents, nous avons présenté le modèle formel et différentes opérations, exprimées dans ce modèle, permettant de construire une

hiérarchie de maillages volumiques. Nous proposons maintenant de dériver deux représentations du modèle des cartes combinatoires multirésolutions. La première, dite explicite, représente les relations du modèle pour chaque brin à chaque niveau de la hiérarchie. Les relations de chaque niveau sont donc immédiatement accessibles. La seconde, dite implicite ou compacte, ne représente que les relations du modèle au niveau le plus fin. Les relations intermédiaires sont reconstruites grâce à une connaissance a priori de la construction de la hiérarchie. Cette représentation n'est d'ailleurs disponible que dans le cadre précis d'une hiérarchie construite par un raffinement primal. Afin d'évaluer nos deux types de représentations, nous avons procédé à des comparaisons de la complexité théorique de nos modèles et de la complexité pratique de nos implantations.

Dans le but d'exprimer de la manière la plus indépendante possible des conditions pratiques d'exécution et d'implantation, nous donnons un ordre de grandeur du coût en temps et en espace des différentes représentations. Cependant, il n'existe pas de structure de données standard pour représenter des maillages volumiques multirésolutions générés par un processus de subdivision. Nous faisons donc un parallèle avec les surfaces de subdivision multirésolutions généralement représentées par une forêt d'arbres quaternaires ou quadtree. Nous prendrons donc ici son pendant volumique qui est la forêt d'arbres octals ou octree. Nous donnerons une comparaison de la complexité théorique entre notre modèle multirésolution explicite et une forêt d'octree. Puis, nous donnerons une comparaison de la complexité théorique entre notre modèle multirésolution implicite et notre modèle multirésolution explicite.

Dans un second temps, nous donnons une mesure précise des complexités temporelles et spatiales pour une configuration de machine de bureau donnée. L'utilisation des volumes de subdivision multirésolutions produit un grand nombre de données. Il devient donc nécessaire de produire des structures de données efficaces. Nous donnerons une comparaison pratique entre une carte combinatoire monorésolution et une carte combinatoire multirésolution explicite. Puis, nous donnerons une comparaison pratique entre carte combinatoire multirésolution explicite et carte combinatoire multirésolution implicite.

Nous commençons par expliciter la mise en œuvre monorésolution des cartes combinatoires disponible dans la librairie de modélisation à base topologique CGoGN [24]. Nous ferons ensuite une analyse comparative des performances de cette implantation par rapport à d'autres structures couramment employées. Le reste du chapitre s'articule autour des deux représentations, de leur principe de fonctionnement et de leurs études de complexités.

### 4.1.1 Mise en œuvre monorésolution

Par le passé, les cartes combinatoires ont déjà été mises en œuvre, par exemple au sein des modeleurs Topofil [6] et Moka [94], du noyau de CGAL [34, 35], et comme structure de données multirésolution en dimension 2 [68]. Nos prototypes ont été réalisés comme des extensions du noyau de modeleur CGoGN /sigɔŋ/ [24] implantant différents modèles à base de cartes et développé depuis quelques années au sein de l'équipe IGG de ICube. Pour cette raison, nous décrivons rapidement la mise en œuvre des cartes combinatoires monorésolutions en dimension  $n$  disponible dans cette plate-forme logicielle écrite en C++.

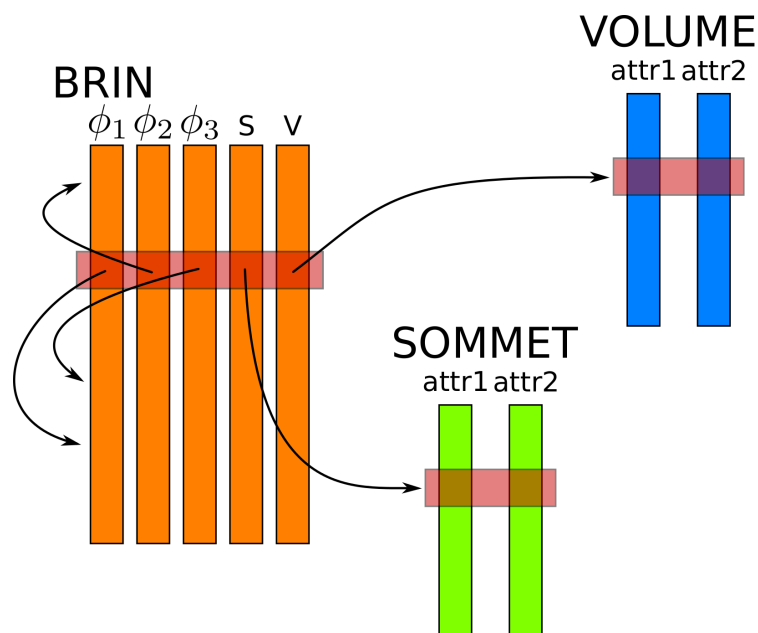
Les deux objectifs principaux de cette plate-forme logicielle sont la généralité et l'efficacité. L'implantation sous-jacente est ainsi capable de supporter un modèle quelque soit sa dimension. Cette faculté permet d'associer des plongements à n'importe quelle dimension de cellule. Il est donc également possible de fournir de manière exhaustive les parcours des voisinages pour toutes les incidences ou toutes les adjacences entre cellules.

L'entité élémentaire de ces représentations à base de carte est le brin. Chaque brin doit stocker, suivant la dimension de la carte, un nombre variable de relations topologiques ainsi que des indices vers les cellules plongées. Pour chaque dimension de cellule plongée, un nombre variable d'attributs doivent pouvoir être associés aux indices des cellules.

Pour répondre aux exigences ci-dessus, une carte peut être encodée comme un ensemble de conteneurs d'attributs : un pour les brins et un pour chaque dimension plongée. Chaque conteneur stocke un ensemble d'attributs pouvant être dynamiquement ajoutés ou supprimés et dont les données sont de type quelconque. Tous les attributs d'un conteneur donné ont le même nombre d'éléments. Chaque élément est identifié par un indice et notamment les brins. La figure 4.1 illustre les conteneurs d'une 3-carte ayant un plongement de sommet et de volume. Le conteneur de brins a trois attributs  $\phi_1$ ,  $\phi_2$  et  $\phi_3$  stockant pour chaque brin  $d$  les indices dans le conteneur de brins de  $\phi_1(d)$ ,  $\phi_2(d)$  et  $\phi_3(d)$ . Ce conteneur a également les attributs **S** et **V** stockant pour chaque brin les indices des sommets et des volumes dans les conteneurs d'attributs de sommets et de volumes qui lui sont associés. En pratique, un conteneur est stocké sous la forme d'un ensemble de tableaux chaînés les uns aux autres.

Cette implantation fournit différents types de mécanismes :

- des mécanismes de bas niveaux permettant par exemple de modifier la taille d'un tableau et ainsi trouver un équilibre entre la fragmentation en mémoire et une allocation mémoire précoce ou encore pour allouer ou



**FIGURE 4.1 :** Les conteneurs d'une carte combinatoire monorésolution de dimension 3 ayant des attributs de sommet et de volume. Les conteneurs de brin stockent les relations  $\phi_1$ ,  $\phi_2$  et  $\phi_3$  vers d'autres brins plus des liens S et V vers les sommets et volumes plongés.

désallouer des tableaux à la volée ;

- des mécanismes de plus haut niveau permettant de rendre les parcours génériques afin de masquer leur complexité.

Ce dernier point est d'autant plus intéressant en volumique où beaucoup de parcours sont non ordonnés et nécessitent un usage incessant de marquages. Il y a également un mécanisme de parcours appelé « Quick » permettant de mettre en cache les parcours globaux et les parcours locaux lorsque ceux-ci sont répétés plusieurs fois sans changement dans la connectivité.

Le lecteur trouvera de plus amples renseignements sur les mécanismes mis en œuvre dans cet article [70]. Nous verrons dans les sections suivantes comment nous avons essayé d'étendre ce noyau de modelleur afin de construire une représentation multirésolution tirant partie de tous ces mécanismes.

#### 4.1.2 Analyse comparative de la performance

Dans cette sous-section nous comparons la mise en œuvre monorésolution des cartes combinatoires décrites ci-dessus (CGoGN) avec d'autres structures de données utilisées pour la représentation de maillages volumiques :



**FIGURE 4.2** : Maillages utilisés pour l'évaluation monorésolution. Le modèle Coverrear (a) avec 4051 sommets, 20650 arêtes, 29767 faces et 13167 volumes. Le modèle Buddha (b) avec 45177 sommets, 264375 arêtes, 413744 faces et 194545 volumes.

les cartes combinatoires dans CGAL [34, 35] (CGAL LLC) et OpenVolume-Mesh [71] (OVM). Nous avons utilisé la procédure d'analyse comparative proposée dans [71] pour évaluer et comparer la performance des différentes structures de données. Cette procédure consiste en différents algorithmes évaluant les fonctionnalités fondamentales de ces bibliothèques telles que l'itération à travers des éléments du maillage, les requêtes de voisinage ou encore la modification de la connectivité.

CGAL LCC est la structure de données représentant les Complexes Cellulaires Linéaires de CGAL, c'est-à-dire une carte combinatoire dans laquelle un point est associé à chaque sommet. OVM est une structure de données hybride implantant d'une part, des relations d'incidences entre demi-arêtes et demi-faces (comparable à une 3-carte combinatoire) dans des listes ordonnées d'éléments, et d'autre part, des relations d'incidences induites par un graphe d'incidence. Ces dernières relations peuvent d'ailleurs être désactivées mais sont cependant nécessaires pour accéder au parcours de certains voisinages. Dans ce cas, il est donc nécessaire de les (re)-calculer après application d'opérations de modifications de la connectivité.

Nous effectuons les tests suivants :

- *Circulator* : pour chaque sommet, énumérer ses volumes incidents puis, pour chaque volume énumérer ses sommets.
- *Circulator2* : pour chaque sommet, énumérer ses sommets adjacents par un volume.
- *Barycenter* : calculer et stocker le barycentre de chaque volume.
- *Smoothing* : déplacer chaque sommet au barycentre des sommets adjacents.

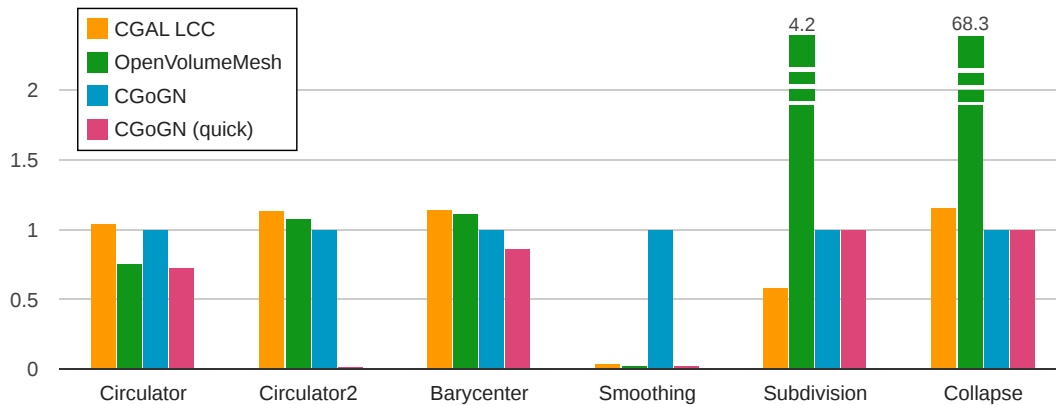


FIGURE 4.3 : Performances temporelles. Les valeurs sont exprimées relativement aux performances de CGoGN.

- *Subdivision* : appliquer un split 1-4 insérant un sommet central à chaque volume d'un maillage tétraédrique.
- *Collapse* : appliquer une série de contractions d'arêtes en sélectionnant à chaque fois la plus petite arête du maillage.

La figure 4.2 montre les maillages volumiques utilisés pour nos mesures de performances. Les résultats obtenus sont présentés dans la figure 4.3. Globalement, les mises en œuvre à base de cartes combinatoires sont aussi performantes que OVM pour les parcours. Cependant, OVM obtient les moins bonnes performances lorsque la connectivité est modifiée.

Le test *Circulator* utilise les relations d'incidences qui sont stockées dans le cas de OVM et traversées dynamiquement pour les deux implémentations des cartes combinatoires. Lorsque l'on active les parcours en mode « Quick », les performances de CGoGN sont comparables ou même meilleures que celle de OVM. Le test *Circulator2* utilise les relations d'adjacences qui ne sont pas directement accessibles dans OVM. Dans ce cas, LCC, OVM et CGoGN obtiennent des performances similaires. Lorsque l'on active les parcours en mode « Quick », CGoGN surpasse largement les deux autres. Pour le test de *Smoothing*, le parcours de la topologie est très mauvais pour CGoGN. Cependant, l'activation du mode « Quick » permet de rendre les performances de CGoGN comparables à LCC et OVM. Pour les algorithmes modifiant la connectivité du maillage, les performances de CGoGN sont comparables à celles de LCC et bien meilleures que celles de OVM qui doit reconstruire les informations d'incidences et d'adjacences après chaque opération pour le test de *Collapse*.

Notre méthode a l'avantage de ne pas stocker de relation d'incidence particulière pour ne pas ralentir les opérations modifiant la connectivité. Cependant, une mise en cache du voisinage pour des parcours rapides est présente



de manière optionnelle. Ainsi, les relations du modèle sont toujours accessibles avec la possibilité de les accélérer si nécessaire. Les bonnes performances des opérations modifiant la connectivité nous donne un avantage certain pour la construction d'une représentation multirésolution efficace quelque soit la dimension.

## 4.2 Représentation multirésolution explicite

Dans cette section, nous présentons le principe de la représentation explicite d'une carte combinatoire multirésolution. Puis, nous comparerons la complexité théorique en temps et en espace avec un modèle multirésolution classique. Pour finir, nous illustrerons notre mise en œuvre et nous ferons une analyse comparative des performances avec une carte combinatoire monorésolution.

### 4.2.1 Principe

Le principe de cette représentation est de coïncider complètement à la définition donnée au chapitre 2. Ainsi, pour tout niveau  $l$ , on stocke la carte combinatoire  $M^l$  du niveau de résolution  $l$  tout en réutilisant les brins de niveaux précédents pour enrichir l'ensemble de brins de chaque niveau. Ainsi, un brin, dans une carte combinatoire multirésolution, est muni de relations topologiques  $\phi_i^l$  ou  $\alpha_i^l$  pour chaque niveau  $l$  entre son niveau d'insertion et le niveau maximum de la carte. Chaque brin doit connaître son niveau d'insertion ainsi que ces relations  $\phi_i^l$  ou  $\alpha_i^l$  pour chaque niveau où il existe. Ce principe fonctionne indépendamment de la dimension de carte.

Cette représentation a l'avantage de permettre d'accéder directement aux relations du modèle à chaque niveau, ce qui a pour effet de parcourir directement, à chaque niveau, l'ensemble de brins formant les cellules.

Il n'existe aucune autre structure de données utilisée pour représenter des volumes de subdivision multirésolutions. Nous avons donc choisi de la comparer à la forêt d'octree.

Comme nous l'avons vu au chapitre 1, un octree opère une subdivision d'un octant ou huit octants à la manière du raffinement de faces polygonales sur un hexaèdre. Chaque volume du maillage initial est représenté par la racine d'un arbre donc chaque nœud est un volume et possède huit fils (un pour chaque

sous volume obtenu lors du raffinement). Au final, on obtient une forêt d'octree dont l'ensemble des racines constitue le maillage de niveau 0.

Rappelons que ce modèle permet de construire une hiérarchie régulière par raffinement régulier des cubes à chaque niveau de l'arbre. Il permet également une construction adaptative en stoppant la construction de l'arbre à des hauteurs différentes suivant un critère donné. Dans ce cas, toutes les feuilles de l'arbre ne partagent pas la même profondeur dans l'arbre.

Ce modèle est de conception simple mais, comme nous venons de le voir, rigide d'utilisation. En effet, ce modèle possède de nombreuses limitations qui sont une généralisation des limitations du quadtree en dimension supérieure, valable quelque soit la dimension de ce type de représentation. Ainsi, dans le cas surfacique, un quadtree permettait de représenter aussi bien une quadrisection de quadrangles ou de triangles. Il n'est pas possible de généraliser ce comportement au cas volumique à cause de la non uniformité et de la non régularité des cellules représentables, comme nous l'avons vu dans le chapitre 1. Il n'est donc pas envisageable de représenter un raffinement de tétraèdre/octaèdre et encore moins les raffinements plus exotiques tels que le raffinement mixte ou le schéma  $\sqrt{3}$  volumique.

De plus, un octree est avant tout un modèle de partitionnement de l'espace et non un modèle de représentation d'une partition cellulaire d'une variété de dimension 3. Les relations topologiques entre les cellules de la subdivision sont donc inexistantes et seules les volumes sont représentés. Le fait de ne pas représenter toutes les cellules de la subdivision mais seulement les volumes engendre plusieurs types de problèmes. Le premier problème est l'impossibilité d'associer des attributs à des cellules comme les arêtes ou les faces. Le second problème est l'apparition d'inconsistances topologiques aux frontières entre les zones de différents niveaux lors de la construction ou de l'accès à une hiérarchie construite de manière adaptative.

Une approche utilisant des cartes combinatoires permet de maintenir la cohérence topologique du maillage pendant les opérations de raffinement, et ce, que ces opérations soient effectuées de manière adaptative ou non. En effet, toutes les cellules de la partition sont représentées, et leurs relations d'incidences sont toutes présentes et accessibles. La possibilité d'avoir des maillages polyédriques permet de supporter de nombreux algorithmes de raffinement dans un seul et même modèle.

Nous présentons la complexité spatiale théorique optimale d'un algorithme de subdivision primale pour une 3-carte combinatoire multirésolution représentée de manière primale. En effet, sous cette forme seule la relation  $\alpha_0^l$  est

modifiée à chaque niveau. Ceci nous permet d'exprimer un rapport dans les conditions optimales entre notre modèle et une représentation d'une forêt d'octree que l'on décrira également. Nous présentons ensuite une mise en œuvre pratique profitant de manière incrémentale des propriétés de la mise en œuvre monorésolution duale.

#### 4.2.2 Complexité temporelle

Les requêtes d'adjacences sont les opérations les plus couramment exécutées pour des volumes de subdivision multirésolutions. Elles sont utiles lors de l'application d'un schéma de subdivision aussi bien que pour l'exécution d'algorithmes plus complexes.

Dans une forêt d'octree, chaque hexaèdre du maillage initial représente la racine d'un arbre octal. Cette structure peut être vue comme un graphe *non orienté* acyclique. Les requêtes de voisinage dans ce type de structure de données sont résolues en temps constant amorti. Cela signifie que le pire cas de cet algorithme ne peut se produire plusieurs fois consécutivement et permet de borner la complexité de l'algorithme. L'algorithme classique consiste à remonter l'arbre jusqu'à la racine parente puis à chercher la cellule adjacente. Les requêtes d'adjacences au niveau racine sont résolus en  $O(\log(n))$  où  $n$  est le niveau maximum, c'est-à-dire la taille de l'arbre le plus profond.

Dans le cas d'une 3-carte multirésolution, chaque maillage de la hiérarchie représente une 3-carte directement accessible. Les requêtes d'adjacences sont donc exécutées en temps constant. La complexité des algorithmes plus évolués comme parcourir le voisinage d'une cellule est linéaire en la taille du voisinage considéré.

#### 4.2.3 Complexité spatiale

Dans cette section nous comparons le coût théorique d'une 3-carte multirésolution avec une forêt d'arbres octals d'hexaèdres dans le cadre d'une subdivision régulière. Ceci est le pire cas au niveau de l'espace mémoire nécessaire. Toutefois, afin de préciser notre propos, nous commencerons par dissocier le coût des modèles eux-mêmes de leur extension multirésolution.

### Coût standard

Soit  $h$  le nombre de brins dans une 3-carte. Pour compter le coût total de l'information topologique, il nous faut compter le nombre de références (pointeur ou index suivant l'implantation) stockés pour chaque brin. Le calcul pour les relations  $\alpha_0$ ,  $\alpha_1$  et  $\alpha_2$  est assez simple puisque chaque brin dispose d'une unique liaison. Il y a donc en tout  $h$  références stockés pour chacune des trois relations. Il nous faut compter également une référence pour chaque brin vers une structure stockant les plongements de sommet. Ceci nous donne un coût de :

$$4 \times h$$

Dans le cas d'une forêt d'octree dont on ne garderait que les racines, le coût se résume au nombre d'hexaèdres présents dans le maillage. En effet, la structure est équivalente à un tableau de dimension 3 où chaque case correspond à un hexaèdre. Cependant, il nous faut rappeler qu'à l'inverse des cartes combinatoires, ceci n'est pas une représentation topologique complète car il n'y a pas de relation liant entre elles toutes les cellules du maillage.

Afin de compléter les informations topologiques disponibles dans ce type de structure, nous nous devons d'ajouter un minimum d'informations décrivant la topologie d'un polyèdre pour que la structure de données définisse un objet volumique. Chaque hexaèdre contiendra ainsi une liste de ses faces, avec pour chaque face une liste de ses arêtes. Ceci nous donne 4 références par face et donc 24 références pour les 6 faces. Enfin, chacune des 12 arêtes contient 2 références vers les sommets. Ceci ajoute 24 références en plus. Il faut en plus ajouter à cela les 8 références vers le tableau de plongement des sommets. Ainsi, pour  $f$  le nombre d'hexaèdres du maillage, nous obtenons un coût de :

$$56 \times f$$

Un hexaèdre étant composé de 24 brins dans une 3-carte, nous savons que :

$$f = \frac{h}{24}$$

Le ratio entre les deux structures de données à cette étape est donc de :

$$\frac{4 \times h}{\frac{56}{24} \times h}$$

Ce qui nous donne :

$$\frac{12}{7} \simeq 1,7$$

La structure des cartes combinatoires est donc à cette étape de notre calcul 70% plus coûteuse en terme d'espace. Ceci est normal car contrairement aux cartes combinatoires, il manque encore des informations topologiques à la structure ad hoc présentée ci-dessus. En effet, nous y avons seulement ajouté le minimum afin de décrire un volume. Il en résulte que toutes les requêtes de voisinage ne sont pas disponibles immédiatement avec la forêt d'octree mais doivent être calculées. Par exemple, pour obtenir toutes les faces autour d'une arête, il faut commencer par chercher le volume adjacent à une des faces de cette arête. Puis, dans la liste des faces de ce volume, il faut chercher celle possédant l'identifiant d'arête correspondant. Enfin, il faut recommencer l'opération. Dans une 3-carte combinatoire, les brins des faces sont reliés entre-eux par la relation volumique  $\alpha_2$  qu'il suffit de parcourir.

### Cas multirésolution

Soit  $h_0$  le nombre de brins au niveau 0 du maillage, et  $k$  le nombre maximal de niveaux de la structure multirésolution. Soit  $D$  le nombre de brins dans une 3-carte multirésolution au niveau  $k$ .

Chaque hexaèdre étant coupé en 8 nouveaux hexaèdres, le nombre de brins est multiplié par un facteur 8 à chaque étape de subdivision. Nous avons donc :

$$D = h_0 \times 8^k$$

Pour compter le coût total de l'information topologique, il nous faut compter le nombre de références stockés pour chaque brin, en particulier des relations  $\alpha_0, \alpha_1, \alpha_2$  et du plongement. Le calcul des relations  $\alpha_1$  et  $\alpha_2$  est assez simple puisque chaque brin dispose d'une unique liaison  $\alpha_1$  ou  $\alpha_2$ . Il y a donc en tout  $D$  (ou  $h_0 \times 8^k$ ) références stockés pour chacune des deux relations et de même pour le plongement, ce qui nous fait :

$$3 \times (h_0 \times 8^k)$$

Pour la relation  $\alpha_0$ , il faut additionner les tailles des tableaux de références contenus dans chaque brin. La taille de ce tableau est fonction du niveau d'insertion du brin. On note que  $\frac{7}{8}$  des brins sont insérés au niveau de résolution maximum et n'ont donc qu'une seule liaison  $\alpha_0$ ,  $\frac{7}{8}$  des autres brins (c'est-à-dire  $\frac{7}{64}$ ) ont deux liens ... ; pour  $i$  entre 1 et  $k$  il y a  $|D| \times \frac{7}{8^i}$  brins dans le tableau à  $i$  éléments. Les brins décrivant le niveau 0 ont  $k+1$  éléments dans leur tableau. Le nombre total d'éléments dans les tableaux de la relation  $\alpha_0$  de tous les brins

est donc :

$$h_0 \times (k + 1) + h_0 \times 7 \sum_{i=1}^k i \times 8^{k-i}$$

Au total, le nombre de références est donc :

$$3 \times h_0 \times 8^k + h_0 \times (k + 1) + h_0 \times 7 \sum_{i=1}^k i \times 8^{k-i}$$

La somme de l'expression précédente peut-être identifiée à la série entière :  $\sum_{n \geq 0} i \times x^i = \frac{x}{(1-x)^2}$ , définie pour  $|x| < 1$ . En négligeant les termes de la série tels que  $i > k$ , on obtient (avec  $x = \frac{1}{2}$ ) :

$$\sum_{i=1}^k i \times 8^{k-i} \simeq 8^k \times \frac{\frac{1}{8}}{(1 - \frac{1}{8})^2} = 8^k \times \frac{8}{49}$$

On substitue :

$$3 \times h_0 \times 8^k + \left( h_0 \times (k + 1) + h_0 \times 8^k \times \frac{8}{7} \right)$$

En négligeant le terme  $k + 1$  par rapport au terme  $8^k$ , cette équation se simplifie en :

$$\left( 3 + \frac{8}{7} \right) \times h_0 \times 8^k = \frac{29}{7} \times h_0 \times 8^k$$

Les racines de la forêt d'octree stockent 8 références pour les informations hiérarchiques vers les fils. Dans la forêt d'octree, un nœud nécessite 9 références pour les informations hiérarchiques, c'est-à-dire 8 pour les fils et 1 pour le parent. Comme nous l'avons exposé dans la section précédente, chaque racine et chaque nœud de la forêt d'octree stocke également 56 références pour les informations topologiques et géométriques. Pour résumer, nous obtenons 64 références pour chaque racine et 65 références par nœud.

Soit  $f_0$  le nombre de volumes au niveau 0 (c'est à dire la racine de l'arbre) et  $k$  le niveau de résolution maximum. Comme le nombre de volumes est multiplié par un facteur 8 à chaque niveau de subdivision, le nombre total de références stockés est :

$$f_0 \times 64 + 65 \times f_0 \times \sum_{l=1}^k 8^l$$

La somme s'écrit sous la forme :  $\sum_{i=0}^n x^i = \frac{x^{n+1}-1}{x-1}$ . Nous pouvons l'exprimer de la manière suivante (avec  $x = 8$ ) :

$$\sum_{l=1}^m 8^l \simeq \frac{8^{m+1} - 1}{7} - 1$$

Comme nous savons que  $f_0 = \frac{h_0}{24}$ , notre équation se simplifie donc en :

$$\frac{65}{21} \times h_0 \times 8^k$$

Le ratio entre les deux structures de données est de :

$$\frac{\frac{29}{7} \times h_0 \times 8^k}{\frac{65}{21} \times h_0 \times 8^k}$$

ou bien encore :

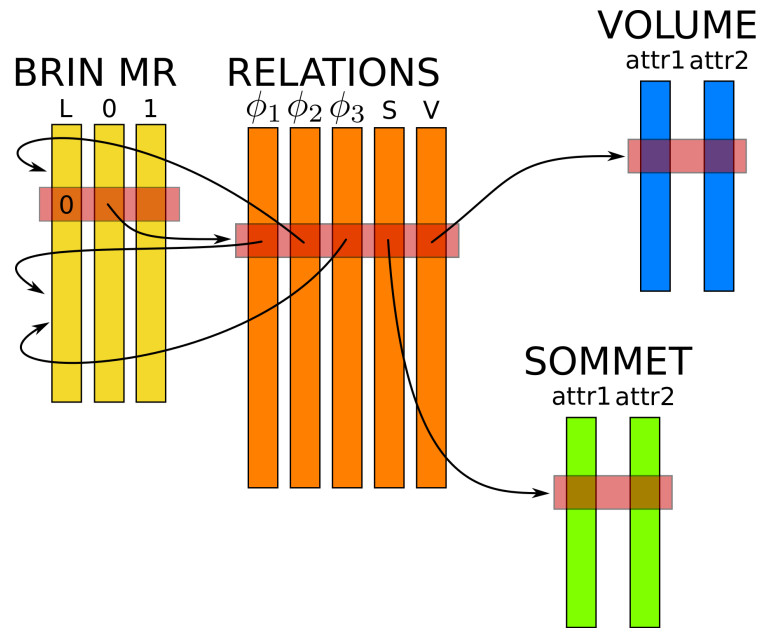
$$\frac{87}{65} \simeq 1,3$$

Le modèle des cartes combinatoires multirésolutions dans sa version primale est donc au final seulement 30% plus coûteux en terme d'espace que la forêt d'arbres octals. Ce surcoût de notre modèle est lié à l'information topologique proposée par le modèle et non à son extension multirésolution. Il est largement compensé par la généricité des polyèdres représentables ainsi que par l'exhaustivité des parcours de voisinage disponibles.

#### 4.2.4 Mise en œuvre et analyse de la performance

Le but de cette partie du travail est de continuer à bénéficier de tous les mécanismes existants sans négliger les objectifs initiaux que sont la généricité et l'efficacité. La mise en œuvre de la représentation multirésolution explicite se veut une extension de la mise en œuvre monorésolution présentée précédemment. C'est donc pourquoi, nous ne basons pas notre mise en œuvre sur celle de la complexité théorique détaillée dans le cas d'une représentation primale. Cette mise en œuvre permet de représenter des cartes combinatoires multirésolutions en dimension  $n$ .

Dans le cadre multirésolution, chaque brin dispose de relations potentiellement différentes pour chaque niveau de résolution entre son niveau d'insertion



**FIGURE 4.4** : Les conteneurs d'une carte combinatoire multirésolution de dimension 3 ayant des attributs de sommet et de volume. Les conteneurs de brins multirésolutions stockent des indices vers les relations. Les conteneurs de relations stockent les relations  $\phi_1$ ,  $\phi_2$  et  $\phi_3$  vers d'autres brins multirésolutions ainsi que des liens S et V vers les sommets et volumes plongés auxquels un brin se rapporte étant donné un niveau.

dans la carte et le niveau maximum courant de la carte. Un brin multirésolution doit donc stocker son niveau d'insertion et ses relations pour chaque niveau où il existe.

Un nouveau conteneur fait donc son apparition : le conteneur de brins multirésolutions. Ce nouveau conteneur possède un premier attribut stockant pour chaque brin le niveau d'insertion. Les autres attributs stockent des indices vers les relations d'un brin pour chaque niveau de résolution où il existe. Un brin est désormais un indice de ce nouveau conteneur. Une ligne du conteneur de relations qui représentait un brin dans le cas monorésolution représente désormais les relations et les indices de plongement pour un brin à un niveau de résolution. Cette représentation fait donc l'usage d'une indirection en mémoire de plus que la version monorésolution. Ainsi, comme dans la définition, les brins sont réutilisés d'un niveau au suivant.

La figure 4.4 illustre les conteneurs d'une carte combinatoire multirésolution de dimension 3 à deux niveaux de résolution avec un plongement de sommet et de volume. Le conteneur de brins multirésolutions a un attribut L stockant pour chaque brin  $d$  son niveau d'insertion ainsi que  $i$  attributs, numérotés de 0 à  $i - 1$ , stockant un indice dans le conteneur de relations pour chaque niveau



entre le niveau d'insertion du brin et le niveau maximum courant  $i$  de la carte. Dans notre exemple, ce sont les attributs 0 et 1 pour une carte à deux niveaux de résolution. Le conteneur de relations a un attribut  $\phi_1$ ,  $\phi_2$  et  $\phi_3$  stockant pour chaque brin  $d$  les indices dans le conteneur de brins multirésolutions de  $\phi_1(d)$ ,  $\phi_2(d)$  et  $\phi_3(d)$ . Comme dans le cas monorésolution, il a également les attributs S et V stockant, pour chaque brin, les indices des sommets et des volumes dans les conteneurs d'attributs de sommets et de volumes qui lui sont associés.

La création d'un nouveau niveau s'effectue en deux étapes. La première étape consiste à dupliquer les brins multirésolutions d'un niveau  $i$  au niveau  $i+1$  en ajoutant un nouvel attribut de niveau au conteneur de brins multirésolutions comme copie de l'attribut de niveau précédent. Dans le cas où tous les brins de ce nouveau niveau voient leurs relations modifiées, comme c'est le cas lors d'une subdivision régulière, les relations des brins multirésolutions précédemment dupliqués sont à leur tour dupliqués. Dans le cas où tous les brins ne sont pas modifiés, comme c'est le cas lors d'une subdivision adaptative, seules les relations modifiées sont dupliquées lors de l'application des différentes opérations modifiant la connectivité. Ainsi, les relations ne sont présentes qu'une seule fois, c'est-à-dire sans duplication.

Nous illustrons le cas adaptatif par la figure 4.5 représentant une carte multirésolution où certaines cellules ont trois niveaux de résolution. Le brin  $d1$  est un brin inséré dans la carte de niveau 0. Un algorithme de subdivision modifie les relations de ce brin au niveau 1, celles-ci sont donc dupliquées. Puis, les relations ne sont pas modifiées au niveau 2. L'indice des relations de ce brin au niveau 1 est donc recopié pour le niveau 2. Le brin  $d2$  est inséré dans la carte de niveau 1. Il n'a donc pas de relation au niveau 0. La cellule à laquelle appartient ce brin n'est pas modifiée au niveau 2. Les relations restent donc inchangées. L'indice des relations de ce brin au niveau 1 est donc également recopié au niveau 2.

La suppression d'un niveau correspond à une simplification du maillage. Cette opération n'est possible que pour le niveau maximum courant afin d'appliquer correctement la définition du modèle multirésolution. Elle s'effectue assez simplement en supprimant une colonne du conteneur de brins multirésolutions et toutes les relations de brins dont le niveau d'insertion correspondant au niveau que l'on veut supprimer.

La taille du conteneur de brins multirésolutions est égale au nombre de brins du niveau maximal. La taille du conteneur de relations est égale au nombre de relations distinctes que prennent les fonctions  $\phi_1$ ,  $\phi_2$  et  $\phi_3$ .

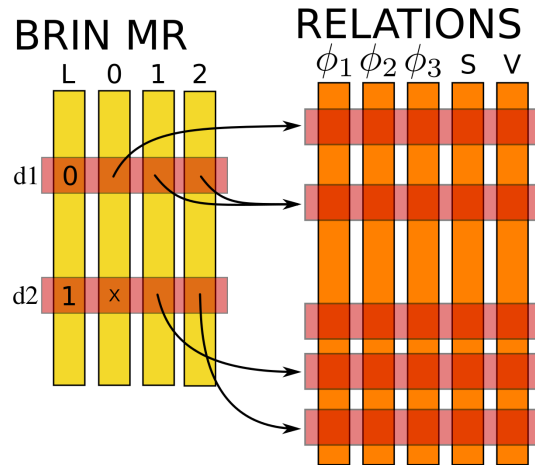


FIGURE 4.5 : Illustration de la duplication des relations pour les brins  $d1$  et  $d2$  dans un cas de subdivision adaptative.

Nous allons maintenant procéder à l'évaluation pratique de notre mise en œuvre. Le coût du changement de niveau courant est négligeable car comparable à l'accès à un élément d'un tableau. Nous avançons que, dans notre représentation multirésolution explicite, chaque niveau représente une carte où les relations sont directement accessibles. Chaque niveau modélise donc une carte combinatoire « monorésolution ». Afin d'évaluer notre représentation multirésolution explicite, et surtout le surcoût engendré par l'indirection liée au conteneur de brins multirésolutions, nous avons donc effectué une analyse comparative des performances en utilisant la procédure d'analyse utilisée dans le cadre monorésolution. Nous comparons donc un maillage chargé grâce à un carte monorésolution et ce même maillage chargé dans notre carte multirésolution au niveau de résolution 0.

Les résultats obtenus sont présentés dans la figure 4.6. Globalement, la structure multirésolution explicite (CGoGN MR) obtient de moins bonnes performances que son homologue monorésolution (CGoGN). Les tests montrent que CGoGN MR est en moyenne 45% plus coûteux. Ces résultats s'expliquent assez naturellement par l'ajout du conteneur de brins multirésolutions, ainsi que par l'indirection vers le conteneur de relations à chaque accès à un brin. En pratique, le parcours et la modification d'un niveau de résolution n'est donc pas tout à fait équivalent à une carte monorésolution. Notons que la structure multirésolution bénéficie néanmoins des mécanismes de la structure monorésolution tels que le parcours en mode « Quick », qui permet de profiter d'un gain de performance. Il faut également mettre ces résultats en perspective face à des structures de données ne proposant pas directement d'extension multirésolution. En effet, les performances de CGoGN MR sont tout de même meilleures que celle de CGAL LCC ou en encore de OVM lors de l'application

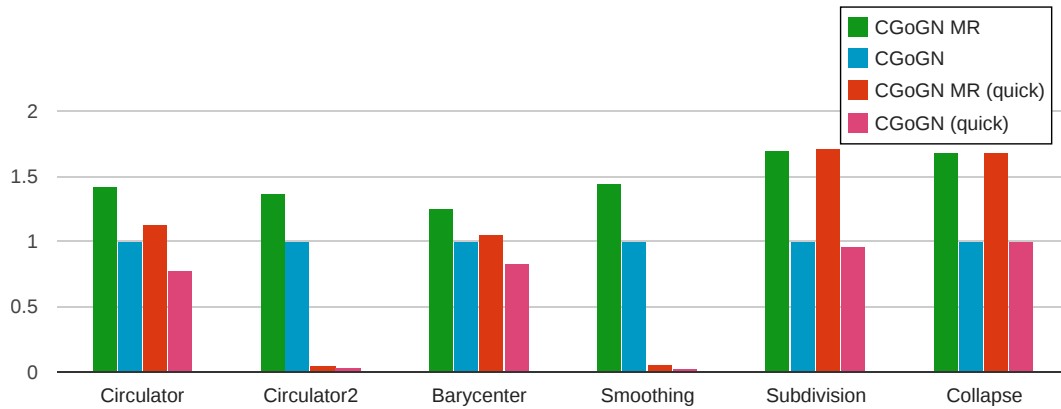


FIGURE 4.6 : Performances temporelles. Les valeurs sont exprimées relativement aux performances de CGoGN (monorésolution).

d'opérations modifiant la connectivité.

### 4.3 Représentation multirésolution implicite

Dans cette section, nous donnons le principe de la représentation implicite ou compacte d'une carte combinatoire multirésolution. Nous comparerons ensuite la complexité théorique en temps et en espace avec la représentation explicite du modèle multirésolution. Pour finir, nous illustrerons notre mise en œuvre et nous ferons une analyse comparative des performances avec la représentation explicite.

#### 4.3.1 Principe

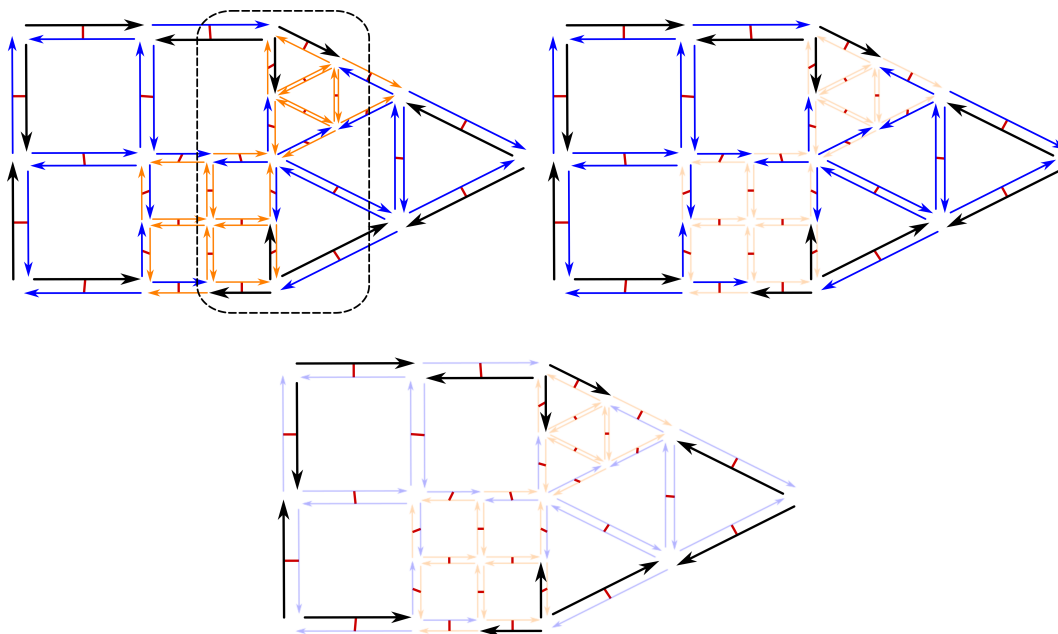
L'objectif est de proposer une représentation compacte de la hiérarchie de maillages générée par un processus de subdivision primale. Ce modèle supporte une subdivision régulière ou adaptative du maillage avec les différents raffinements primaux tels que nous les avons vus dans le chapitre précédent. Il est encodé grâce à une structure de données monorésolution pour garantir sa compacité. Enfin, nous définissons ce modèle dans le formalisme dual des cartes combinatoires qui nous permet, de surcroît, de le définir de manière constructive en plusieurs dimensions.

Rappelons que la construction d'une hiérarchie débute avec un maillage grossier représenté par la carte combinatoire  $M^0 = (D^0, \phi_1^0, \dots, \phi_n^0)$ . À chaque étape de subdivision, les opérations de raffinement ajoutent de nouveaux brins

à la carte et modifient leurs relations définissant successivement les cartes combinatoires  $M^i = (D^i, \phi_1^i, \dots, \phi_n^i)$ . Le principe de base de cette représentation implicite est de ne stocker que le maillage le plus fin  $M^k = (D^k, \phi_1^k, \dots, \phi_n^k)$ . Étant donné ce maillage fin, le défi est de retrouver les versions intermédiaires  $M^l$  de la carte en utilisant le moins d'informations additionnelles possible. Autrement dit, nous devons reconstruire pour chaque niveau  $i$  l'ensemble de brins  $D^i$  et les relations  $\phi_1^i, \dots, \phi_n^i$  entre ces brins.

### Ensembles de brins

Le processus de raffinement ajoute de nouveaux brins à chaque étape de subdivision. Pour chaque niveau  $i$ , la carte est alors définie par  $D^i$  l'ensemble de tous les brins insérés du niveau 0 au niveau  $i$ . Pour pouvoir être capable de construire chaque  $D^i$  comme un sous ensemble de  $D^k$ , les brins sont étiquetés par leur niveau d'insertion. L'ensemble  $D^i$  est alors défini par l'ensemble des brins dont le niveau d'insertion est inférieur ou égale à  $i$ . La figure 4.7 illustre ce principe sur un exemple en dimension 2.



**FIGURE 4.7 :** La carte en haut à gauche a été subdivisée deux fois de manière adaptative. Le niveau d'insertion des brins est illustré par des couleurs : noir, bleu, orange pour les brins insérés respectivement au niveau 0, 1 puis 2. L'ensemble de brins  $D^2$  définissant la carte de niveau 2 est montré par la figure en haut à gauche. Les sous-ensembles de brins  $D^1$  et  $D^0$  définissant les cartes de niveau 1 et 0 sont mis en évidence par les figures en haut à droite et celle du bas.

La quantité d'informations requise pour encoder ce niveau d'insertion dépend du nombre maximal de niveau de résolution désiré.

## Relations entre ces brins

Nous savons à présent comment définir  $D^i$  pour chaque niveau de résolution. Nous devons maintenant reconstruire les relations  $\phi_1^i, \dots, \phi_n^i$  entre ces brins. Nous commençons par expliquer notre méthode avec une carte de dimension 2 afin de bien montrer les mécanismes mis en jeu. Nous montrons ensuite que notre méthode s'étend en dimension 3.

### Cas de la dimension 2

La première relation à reconstruire est la relation  $\phi_1^i$  qui définit des cycles de brins dans  $D^i$ . La figure 4.8 montre un détail de la partie encadrée de la figure 4.7. Dans cet exemple, nous tentons de traverser la carte au niveau 0. Les brins de  $D^0$  sont mis en évidence en noir gras. Débutant par le brin  $d1$ , le brin suivant dans cette face au niveau 0 est le brin  $d2$ . Le processus qui reconstruit la relation  $\phi_1^0$  doit « sauter » au-dessus de plusieurs brins pour aller de  $d1$  à  $d2$ .

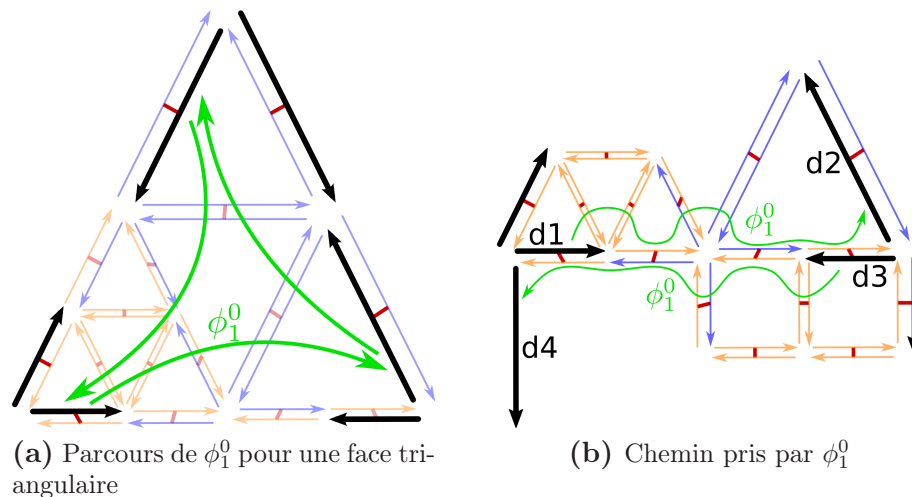
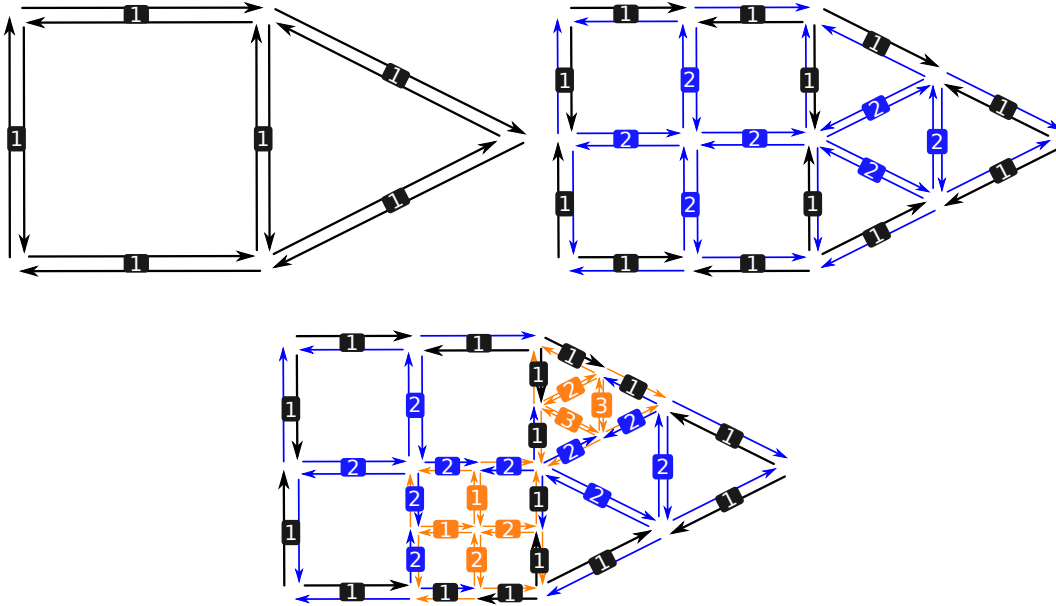


FIGURE 4.8 : La relation  $\phi_1^i$  est définie par un chemin dans  $D^k$  entre deux brins de  $D^i$  qui se suivaient l'un l'autre dans une face avant les étapes de subdivision.

L'idée fondamentale consiste à traverser tous les sommets qui ont été successivement connectés à l'arête de niveau 0 par le processus de raffinement jusqu'à trouver un brin du niveau considéré. Pour chaque sommet intermédiaire,

un certain nombre d'arêtes incidentes doivent être « sautées » pour retourner sur l'arête d'origine et continuer le chemin. Le nombre d'arêtes à sauter dépend de l'algorithme de raffinement utilisé dans cette face. Considérons à nouveau la figure 4.8. Pour la face comprenant le brin  $d1$ , deux nouvelles arêtes incidentes à chaque nouveau sommet ont été obtenues par une quadrisection de faces triangulaires. Pour la face comprenant le brin  $d3$ , une seule nouvelle arête incidente à chaque nouveau sommet a été obtenue par une quadrisection de faces polygonales. Il faut noter que dans notre exemple, les degrés des sommets intermédiaires (respectivement 4, 8 et 3) ne sont pas constants. Ces degrés dépendent du niveau de raffinement et du type de raffinement utilisé dans une face. Cette information du nombre d'arêtes insérées ne suffit donc pas. Ainsi, afin que l'algorithme sache où stopper le processus de saut d'arête, les arêtes d'origines sont identifiées par des étiquettes.



**FIGURE 4.9 :** Les étiquettes d'arête sont utilisées comme un guide pour reconstruire les relations  $\phi_1^i$ . Seulement trois étiquettes sont nécessaires pour distinguer une arête d'origine d'une arête qui lui est incidente après une étape de raffinement.

Une première approche consiste à associer une unique étiquette d'arête à chaque arête du maillage grossier. Puis, à chaque étape de raffinement, les étiquettes des arêtes de niveau  $i$  sont copiées sur les deux arêtes de niveau  $i + 1$  résultant du raffinement d'arête. De nouvelles étiquettes sont données aux nouvelles arêtes insérées au niveau  $i + 1$ . Toutes ces étiquettes peuvent être utilisées pour reconstruire avec succès les relations  $\phi_1^i$  pour chaque niveau de résolution  $i$ . La procédure du « saut » consiste alors simplement à tourner autour d'un sommet intermédiaire jusqu'à retrouver l'arête avec la bonne étiquette. Cependant, cette approche a une limite forte : le nombre total d'arêtes

identifiables de manière unique est limité au nombre d'étiquettes disponibles. Notre but étant d'être compact, il n'est pas possible de dépenser autant de mémoire pour ces étiquettes d'arêtes.

Pour que l'algorithme de reconstruction de  $\phi_1^i$  fonctionne, l'arête d'origine doit seulement être distinguée de celles qui sont incidentes aux sommets intermédiaires ajoutés par les raffinements successifs. L'étiquette de chaque arête doit donc seulement être différente des étiquettes des arêtes qui sont adjacentes aux sommets insérés par le raffinement. Il n'y a donc pas besoin d'une unicité globale des étiquettes. Il s'en suit que dans le cas d'un raffinement qui procède à une quadrangulation de faces triangulaires, trois étiquettes sont suffisantes. En effet, chaque nouvelle arête insérée au niveau  $i + 1$  est incidente à deux arêtes existantes de niveau inférieur desquelles elle doit être distinguée. Dans le cas d'une quadrangulation de faces polygonales, deux étiquettes sont suffisantes car chaque nouvelle arête insérée au niveau  $i + 1$  est incidente à une seule arête de niveau inférieur. La figure 4.9 montre les étiquettes d'arêtes utilisées pour notre exemple. Au niveau 0, toutes les étiquettes sont initialisées à la même valeur car, à ce niveau, il n'y a pas encore besoin de la phase de reconstruction pour parcourir les cellules. Puis, à chaque étape de raffinement, l'étiquette d'une nouvelle arête est choisie comme la plus petite étiquette différente de celles des arêtes qui lui sont incidentes par un sommet.

L'algorithme pour calculer  $\phi_1^i(d)$  est donc le suivant :

---

**Algorithme 3** Algorithme calculant  $\phi_1^i$  en dimension 2.

---

```

1: fonction  $\phi_1^i(i,d)$ 
2:    $finished \leftarrow false$ 
3:    $l \leftarrow edgeLabel(d)$ 
4:    $it \leftarrow d$ 
5:   répéter
6:      $it \leftarrow \phi_1^k(it)$ 
7:     si  $insertionLevel(it) \leq i$  alors
8:        $finished \leftarrow true$                                      ▷ Brin de  $D^i$  trouvé
9:     sinon
10:      tant que  $edgeLabel(it) \neq l$  faire                               ▷ Saute les arêtes
11:         $it \leftarrow \phi_1^k(\phi_2^k(it))$                                    ▷ autour du sommet
12:      fin tant que
13:    fin si
14:  jusqu'à  $!finished$ 
15:  renvoyer  $it$ 
16: fin fonction

```

---

Les relations  $\phi_2^i$  peuvent être définies à partir des relations  $\phi_1^i$ . En effet,

pour un brin  $d$  donné,  $\phi_2(d)$  est le brin qui représente la même arête que  $d$  mais dans la face adjacente. Il représente également le sommet suivant dans la face du brin  $d$ . Dans l'exemple de la figure 4.10 nous avons  $\phi_2^0(d1) = d3$ . Aller du brin  $d1$  au brin  $d3$  peut se faire en allant d'abord de  $d_1$  au brin  $d_2 = \phi_1^0(d1)$  qui représente le sommet suivant dans la face de  $d1$ , puis en allant de  $d_2$  à  $d3 = \phi_2^k(\phi_{-1}^k(d2))$  qui représente la face précédente (suivant le sens trigonométrique) autour de ce sommet. Pour tout niveau  $i$  et pour tout brin  $d \in D^i$ , nous avons :  $\phi_2^i(d) = \phi_2^k(\phi_{-1}^k(\phi_1^k(d)))$ .

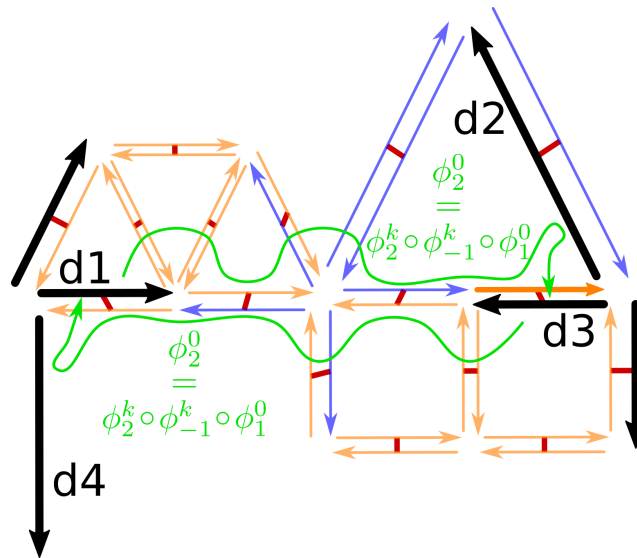


FIGURE 4.10 : La relation  $\phi_2^i$  est définie en utilisant la relation  $\phi_1^i$  puis un tour en sens trigonométrique autour du sommet atteint.

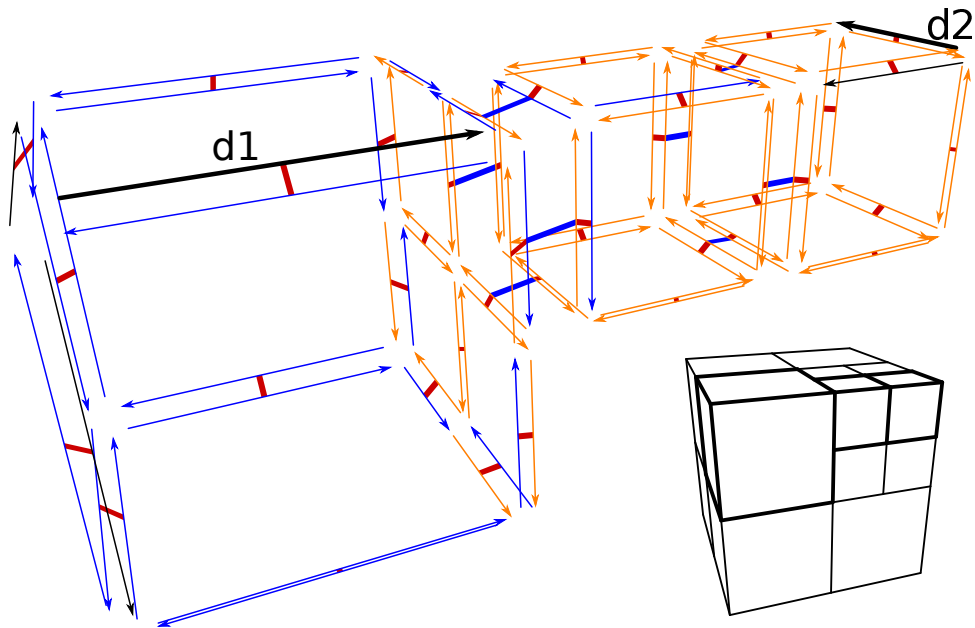
Grâce à  $D^i$ ,  $\phi_1^i$  et  $\phi_2^i$ , nous avons défini complètement tout niveau de résolution intermédiaire comme une carte  $M^i = (D^i, \phi_1^i, \phi_2^i)$ . Toutes les cellules de ces maillages peuvent être traversées comme dans une carte combinatoire classique.

### Cas de la dimension 3

De manière similaire au cas 2D, la première relation à reconstruire est  $\phi_1^i$ . La figure 4.11 montre le détail d'un maillage volumique subdivisé de manière adaptative.

L'idée fondamentale est donc similaire au cas 2D : passer à travers tous les sommets incidents à une arête de niveau 0 jusqu'à trouver un brin du niveau considéré. Pour chacun de ces sommets intermédiaires, certaines arêtes incidentes à ces sommets doivent être sautées pour retourner sur l'arête d'origine





**FIGURE 4.11** : Détail d'une 3-carte subdivisée trois fois de manière adaptative. Les brins de couleurs noir, bleu, orange représentent respectivement les niveaux 0, 1 et 2. Les brins  $d1$  et  $d2$  appartiennent donc à la même face de niveau 0 et pour être plus précis  $\phi_1^0(d1) = d2$

et continuer le chemin. De plus, les raffinements en dimension 3 insèrent également de nouvelles faces incidentes aux nouvelles arêtes. Ainsi, pour chaque arête à sauter, un certain nombre de faces incidentes doivent également être sautées pour retourner sur la face d'origine et continuer le chemin. Le nombre d'arêtes et de faces à sauter dépendent du raffinement utilisé dans le volume. Afin que l'algorithme sache où s'arrêter dans les processus de saut d'arêtes et de saut de faces, les arêtes et faces d'origines doivent être identifiées en utilisant des étiquettes. Par exemple sur la figure 4.11, dans le volume de  $d1$  le raffinement a ajouté une seule arête à chaque nouveau sommet (quadrisection de faces polygonales) et une seule face entre chaque nouvelle arête (raffinement hexaédrique).

Notre objectif étant d'être le plus compacte possible, nous étendons à la dimension 3 la procédure d'étiquetage utilisée en dimension 2. Comme nous l'avons vu au chapitre précédent, le raffinement primal de volume commence par un raffinement de surface appliqué aux faces des volumes. Les étiquettes d'arêtes sont donc traitées de la même manière que dans le cas 2D. De plus, les arêtes intérieures, incidentes au sommet central créé par le raffinement primal, peuvent prendre n'importe quelle étiquette. Elles ne font pas partie des faces du volume subdivisé et ne sont pas utiles pour la reconstruction du  $\phi_1^i$  des brins de ces faces.

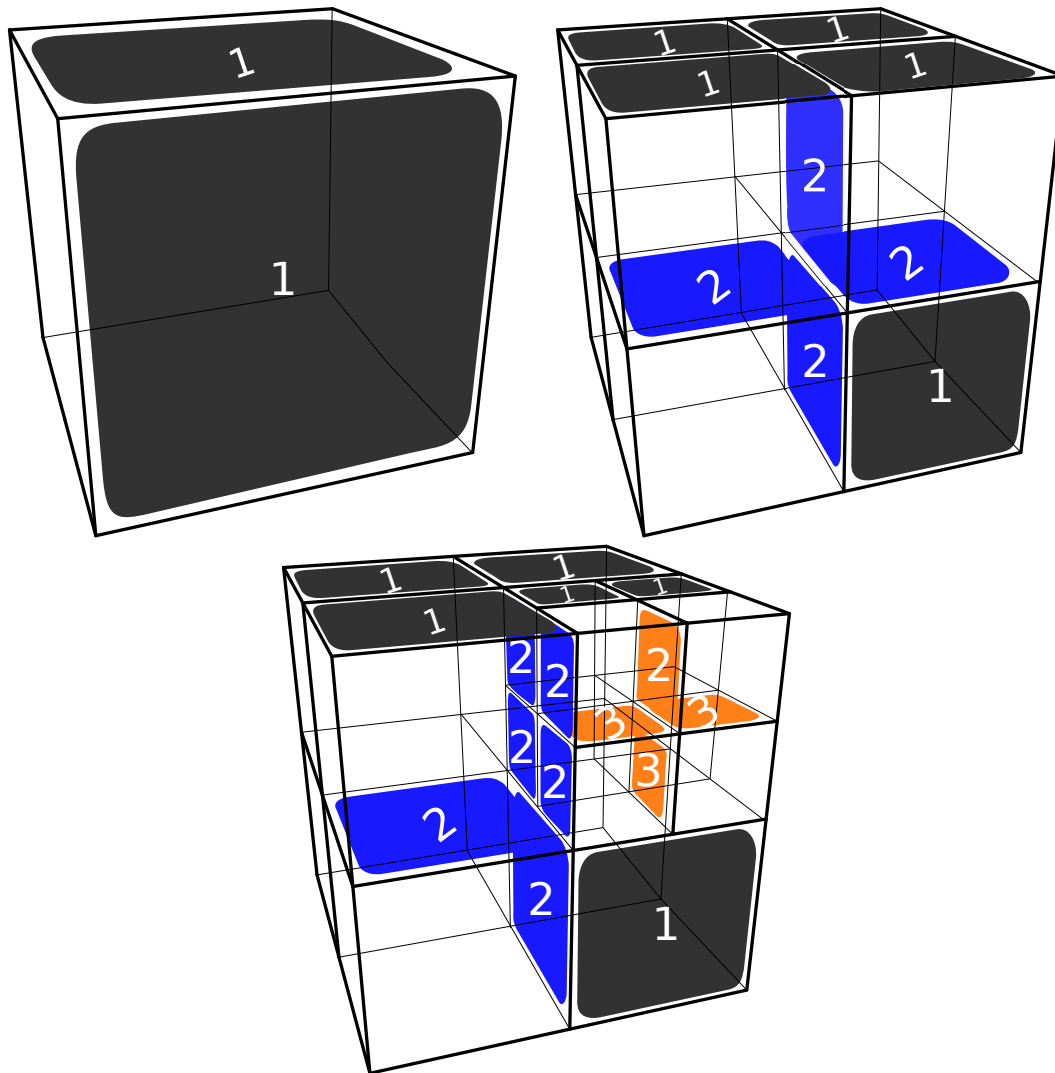
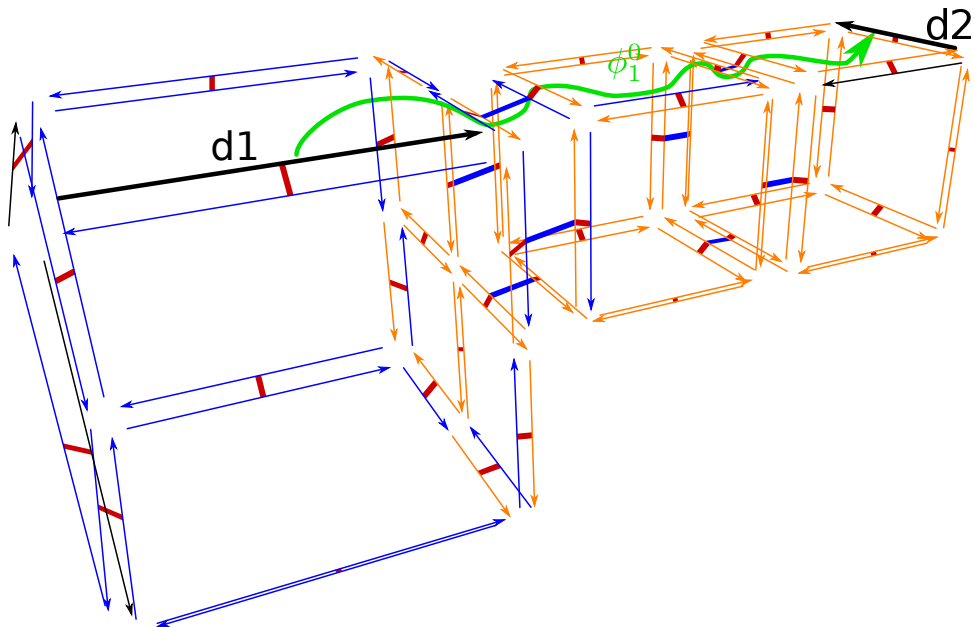


FIGURE 4.12 : Certaines étiquettes de faces sont utilisées en tant que guide pour reconstruire les relations  $\phi_1^i$ . Dans le cas général, seulement quatre étiquettes différentes sont nécessaires pour être capable de distinguer une face d'origine de toutes les faces de niveau supérieur qui lui sont connectées durant les étapes de raffinement.

Les faces d'origines doivent également être distinguées de celles qui sont ajoutées par les étapes de raffinements successifs. Ainsi, l'étiquette de chaque face doit seulement différer des étiquettes des faces qui lui sont incidentes et qui ont été ajoutées pendant le raffinement. Il s'en suit que dans le cas d'un raffinement tétraédrique/octaédrique, quatre étiquettes sont suffisantes. En effet, chaque nouvelle face ajoutée par le raffinement est adjacente à trois faces existantes dont elle doit être distinguée. Dans le cas d'un raffinement polyédrique quelconque, trois étiquettes suffisent car chaque nouvelle face est adjacente à seulement deux faces existantes. Au niveau 0, les étiquettes sont initialisées

à la même valeur car à ce niveau il n'y a pas encore besoin de la phase de reconstruction pour parcourir les cellules. À chaque étape de raffinement l'étiquette d'une nouvelle face est choisie comme la plus petite étiquette différente des étiquettes des faces adjacentes. La figure 4.12 illustre l'étiquetage des faces utilisé dans notre exemple.

Pour obtenir l'algorithme de  $\phi_1^i(d)$  en dimension 3, la ligne 11 de l'algorithme en dimension 2 qui permet de tourner autour d'un sommet doit être remplacée par son pendant volumique permettant de tourner autour de faces comme illustré par la figure 4.13.



**FIGURE 4.13** : En dimension 3, la relation  $\phi_1^i$  est également définie par un chemin dans  $D^k$  entre deux brins de  $D^i$  qui se suivent dans une face avant le raffinement. Nous illustrons ici  $\phi_1^0$  entre les brins  $d1$  et  $d2$  pour une carte à 3 niveaux de résolution.

L'algorithme 4 permet de calculer  $\phi_1^i(d)$ .

La relation  $\phi_2^i$  peut être définie à partir de la relation  $\phi_1^i$  et reste la même que dans le cas 2D. Dans l'exemple de la figure 4.14,  $\phi_2^0(d1) = d3$ . Aller de  $d1$  à  $d3$  peut se faire en allant d'abord de  $d1$  à  $d2 = \phi_1^0(d1)$  qui représente le prochain sommet dans la face de  $d1$ , puis par aller de  $d2$  à  $d3 = \phi_2^k(\phi_{-1}^k(d2))$  qui représente la face précédente autour de ce sommet (dans le sens trigonométrique). Pour tout niveau  $i$  et pour tout brin  $d \in D^i$ , nous avons  $\phi_2^i(d) = \phi_2^k(\phi_{-1}^k(\phi_1^i(d)))$ .

La relation  $\phi_3^i$  peut être définie de manière similaire à  $\phi_2^i$ , à partir de la relation  $\phi_1^i$ . Dans l'exemple de la figure 4.15,  $\phi_3^0(d1) = d3$ . Aller de  $d1$  à  $d3$  peut

---

**Algorithme 4** Algorithme calculant  $\phi_1^i$  en dimension 3.

---

```

1: fonction  $\text{phi}_1(i,d)$ 
2:    $\text{finished} \leftarrow \text{false}$ 
3:    $l \leftarrow \text{edgeLabel}(d)$ 
4:    $it \leftarrow d$ 
5:   répéter
6:      $it \leftarrow \phi_1^k(it)$ 
7:     si  $\text{insertionLevel}(it) \leq i$  alors
8:        $\text{finished} \leftarrow \text{true}$                                 ▷ Brin de  $D^i$  trouvé
9:     sinon
10:      tant que  $\text{edgeLabel}(it) \neq l$  faire                        ▷ Saute les arêtes
11:         $l2 \leftarrow \text{faceLabel}(it)$                                ▷ autour du sommet
12:         $it2 \leftarrow \phi_2^k(it)$ 
13:        tant que  $\text{faceLabel}(it2) \neq l2$  faire                    ▷ Saute les faces
14:           $it2 \leftarrow \phi_2^k(\phi_3^k(it2))$                        ▷ autour de l'arête
15:        fin tant que
16:         $it \leftarrow \phi_1^k(it2)$ 
17:      fin tant que
18:    fin si
19:  jusqu'à  $!\text{finished}$ 
20:  renvoyer  $it$ 
21: fin fonction

```

---

se faire en allant d'abord de  $d1$  à  $d2 = \phi_1^0(d1)$  qui représente le prochain sommet dans la face de  $d1$ , puis par aller de  $d2$  à  $d3 = \phi_3^k(\phi_{-1}^k(d2))$  qui représente la face du volume adjacent autour de ce sommet. Pour tout niveau  $i$  et pour tout brin  $d \in D^i$ , nous avons  $\phi_3^i(d) = \phi_3^k(\phi_{-1}^k(\phi_1^i(d)))$ .

Grâce à  $D^i$ ,  $\phi_1^i$ ,  $\phi_2^i$ ,  $\phi_3^i$ , nous avons complètement défini chaque niveau de résolution intermédiaire comme étant une carte  $M^i = (\phi_1^i, \phi_2^i, \phi_3^i)$ . Toutes les cellules de ces maillages sont complètement définies pour chaque niveau et peuvent être traversées comme dans une carte combinatoire classique.

### 4.3.2 Complexité temporelle

Rappelons que dans le cas d'une représentation multirésolution explicite, chaque maillage de la hiérarchie représente une 3-carte directement accessible. Les requêtes d'adjacences sont donc exécutées en temps constant. La complexité des algorithmes plus évolués, comme traverser les arêtes voisines, est linéaire en la taille du voisinage considéré.

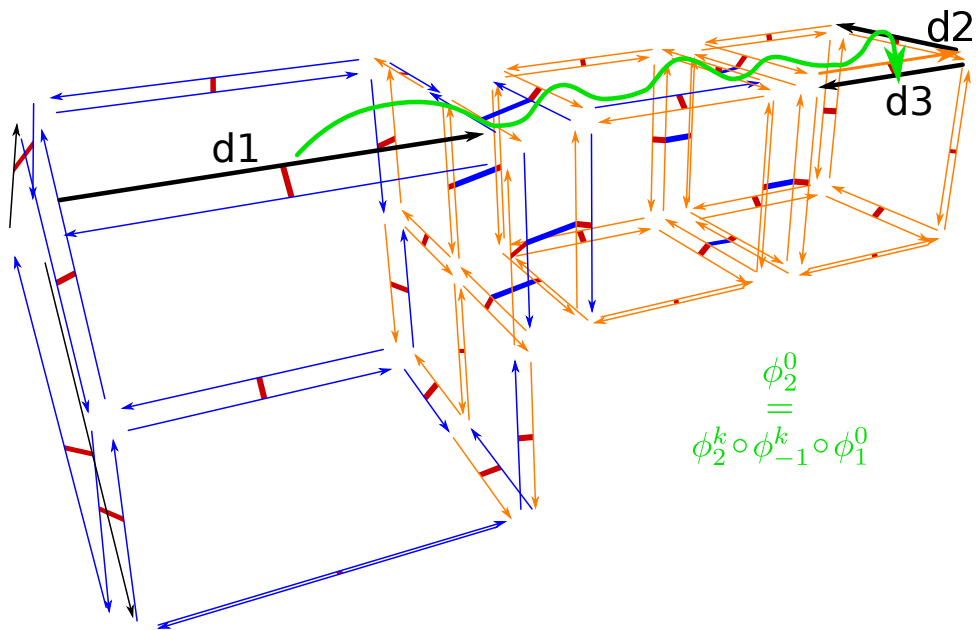


FIGURE 4.14 : La relation  $\phi_2^i$  est définie en utilisant la relation  $\phi_1^i$  ainsi qu'un tour dans le sens trigonométrique autour du sommet atteint. Nous illustrons ici  $\phi_2^0$  entre les brins  $d1$  et  $d3$  pour une carte à 3 niveaux de résolution.

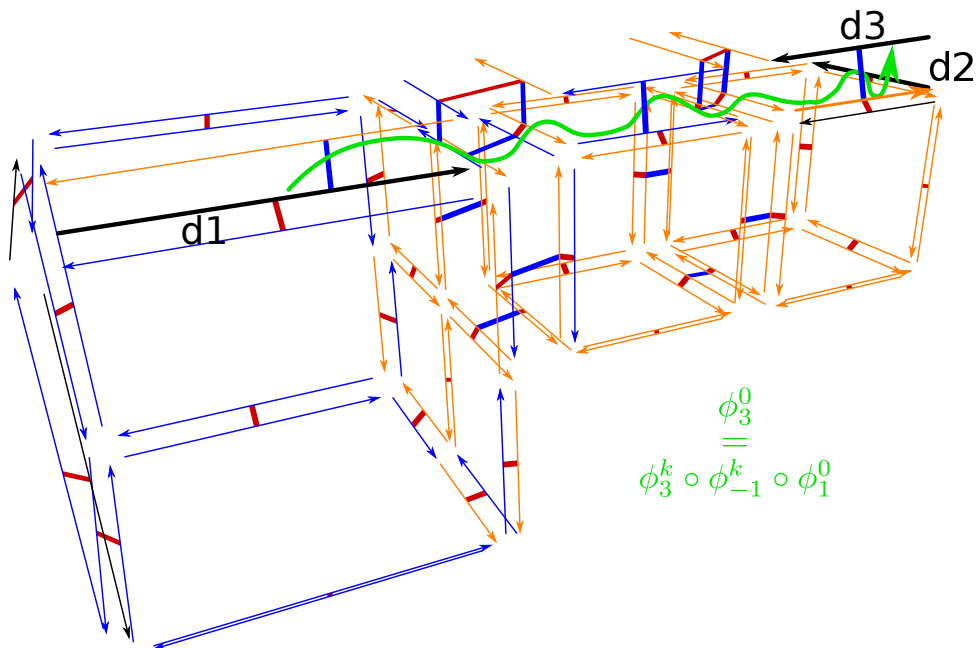


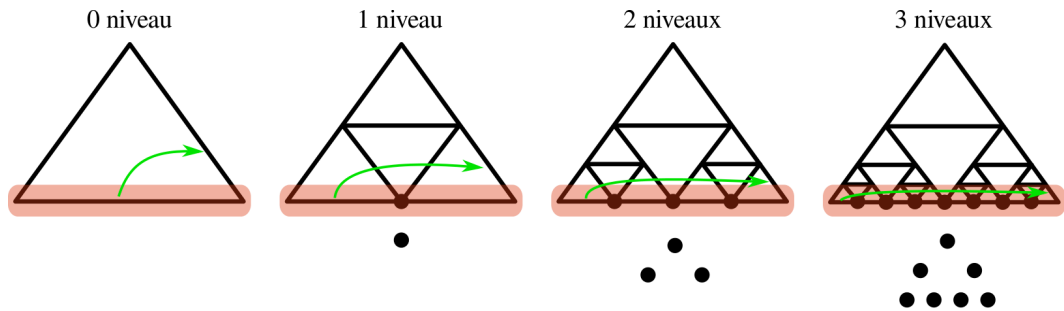
FIGURE 4.15 : La relation  $\phi_3^i$  est définie en utilisant la relation  $\phi_1^i$  ainsi qu'un tour dans le sens trigonométrique autour du sommet atteint pour finir dans la face du volume adjacent. Nous illustrons ici  $\phi_3^0$  entre les brins  $d1$  et  $d3$  pour une carte à 3 niveaux de résolution.

Dans le cas d'une représentation multirésolution implicite, toute la complexité se situe dans l'opération  $\phi_1^i$ , car c'est elle qui permet de « sauter » les brins des niveaux inférieurs. Étant donné un niveau que l'on cherche à parcourir, la complexité de cette opération est dépendante du niveau maximum de subdivision du maillage et du raffinement utilisé. Soit une carte à  $n$  niveaux. Étant donné un brin de niveau  $n$ , la complexité de l'opération  $\phi_1^n$  est constante car l'opération est effectuée au niveau maximum de la carte. On utilise donc la relation stockée. Étant donné un brin de niveau  $i$  avec  $0 \leq i < n$ , la complexité de l'opération  $\phi_1^i$  est multipliée par la longueur moyenne du chemin de brins entre le niveau du brin de départ et le niveau considéré.

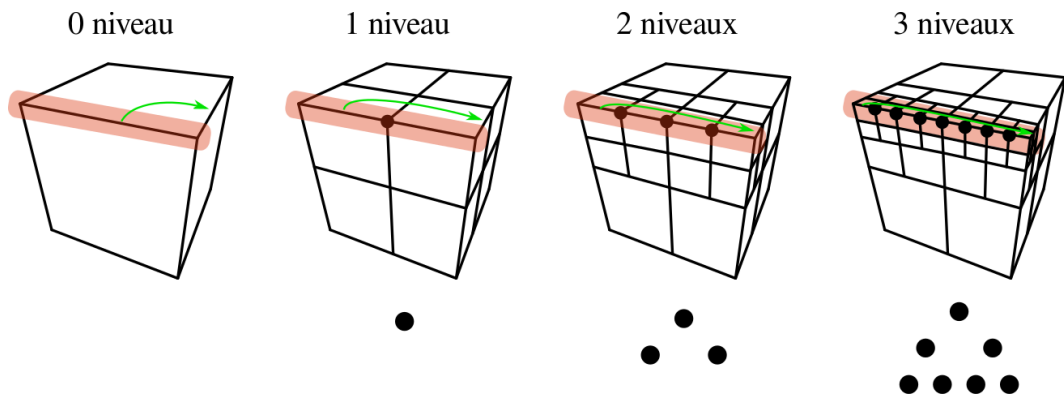
Pour évaluer l'évolution de la longueur de ce chemin, nous nous plaçons dans le cadre d'une hiérarchie construite de manière régulière par une quadrisection de faces triangulaires en 2D et un raffinement volumique par quadrisection de faces polygonales. Dans ce cas régulier, les opérations de raffinement 2D et 3D ajoutent à chaque étape un nouveau sommet à une arête. Le chemin de brins traverse donc tous les sommets de l'arête dont le niveau est inférieur au niveau considéré. Ceci implique que sa longueur est corrélée au nombre de sommets ajoutés à l'arête.

La figure 4.16a illustre ce principe en 2D pour un maillage triangulaire à 3 niveaux de résolutions. L'arborescence de sommets ajoutés à l'arête forme un arbre binaire plein dans lequel chaque sommet est soit un sommet interne avec exactement deux fils (c'est-à-dire un sommet ajouté à des niveaux de raffinements antérieurs), soit une feuille (c'est-à-dire un sommet ajouté au niveau maximum). La flèche verte représente la relation  $\phi_1$  entre deux brins de deux arêtes de niveau 0. La figure 4.16b illustre ce principe en 3D pour un maillage hexaédrique à 3 niveaux de résolutions. L'arborescence de sommets ajoutés à l'arête forme également un arbre binaire plein. La flèche verte représente la relation  $\phi_1$  entre deux brins de deux arêtes de niveau 0.

On constate dans ces exemples que le processus de raffinement ajoute exactement  $2^i$  sommets à l'arête avec  $i$  le niveau de subdivision du maillage, soit exactement le nombre de nœuds à un niveau  $i$  dans un arbre binaire plein. La complexité de cette opération est donc de l'ordre du nombre maximum de nœuds d'un arbre binaire pour un niveau donné. Étant donné un arbre de niveau  $n$ , ou encore un maillage à  $n$  niveaux de résolutions, la longueur du chemin de brins au niveau  $i$  est de  $2^i - 1$  avec  $i \geq 1$  et où  $i$  est la hauteur courante de l'arbre ou encore le niveau de résolution courant. La complexité de l'opération  $\phi_1^i$  est donc de l'ordre du nombre maximum de nœuds d'un arbre binaire entre la racine et une hauteur donnée ou plus exactement exponentielle en le niveau de résolution parcouru.



(a) Arborescence formée par l'ajout de nouveaux sommets à chaque niveau de résolution pour un raffinement de faces triangulaires.



(b) Arborescence formée par l'ajout de nouveaux sommets à chaque niveau de résolution pour un raffinement d'hexaèdre.

FIGURE 4.16 : Illustration de l'arborescence de sommets générée par les raffinements 2D et 3D.

### 4.3.3 Complexité spatiale

Le but de cette représentation implicite est d'éviter le coût de l'information hiérarchique, ceci afin d'être aussi compact que possible en comparaison d'une formulation plus explicite de la hiérarchie. Comme nous l'avons expliqué dans la section 4.3.1, il est nécessaire de stocker deux types d'informations pour être capable de définir et traverser les différents niveaux de résolution : le niveau d'insertion d'un brin et des étiquettes pour les arêtes et pour les faces (en dimension 3 uniquement).

En dimension 2, la quadrisection de faces triangulaires exige trois étiquettes différentes alors que la quadrisection de faces polygonales seulement deux. Il s'en suit que dans le cas général deux bits sont suffisants pour stocker les étiquettes d'arêtes. Lorsque seule la quadrisection de faces polygonales est utilisée, un seul bit suffit pour stocker les étiquettes d'arête.

En dimension 3, le raffinement tétraédrique/octaédrique et le raffinement mixte nécessitent trois étiquettes d'arête différentes et quatre étiquettes de face différentes. Le raffinement polyédrique basé sur une quadrisection de faces polygonales nécessite, quant à lui, uniquement deux étiquettes d'arête différentes ainsi que trois étiquettes de face différentes. Dans le premier cas, deux bits sont nécessaires pour les étiquettes d'arête ainsi que deux bits pour les étiquettes de face. Dans le second cas, un bit suffit pour les étiquettes d'arête et deux bits pour les étiquettes de face.

Le tableau 4.1 résume ces informations et montre le nombre de bits restant disponibles pour encoder le niveau d'insertion des brins, si l'on choisi de stocker toutes ces informations additionnelles dans un octet par brin seulement. Il est à noter que dans le pire cas (raffinement tétraédrique/octaédrique ou mixte de maillages volumiques), quatre bits sont laissés libre pour encoder le niveau d'insertion. Ceci permet d'encoder 16 niveaux de résolution.

| Dimension | Information d'un Brin | Raffinement |                 |
|-----------|-----------------------|-------------|-----------------|
|           |                       | Quad        | Tri or Tri/Quad |
| 2-cartes  | Étiquette d'arête     | 1 bit       | 2 bits          |
|           | Niveau d'insertion    | 7 bits      | 6 bits          |
| 3-cartes  | Étiquette d'arête     | 1 bit       | 2 bits          |
|           | Étiquette de face     | 2 bits      | 2 bits          |
|           | Niveau d'insertion    | 5 bits      | 4 bits          |

**TABLE 4.1** : Résumé du coût mémoire d'une hiérarchie compacte de 2-cartes et 3-cartes sur une base de 1 octet par brin.

Le coût mémoire d'une 2-carte multirésolution et d'un quadtree, deux structures hiérarchiques explicites pour les maillages surfaciques multirésolutions, ont été étudiés dans [68] pour le cas du raffinement régulier d'un maillage triangulaire. Le nombre d'indices stockés par une 2-carte multirésolution est de :

$$\frac{33}{9} \times (h \times 4^m)$$

où  $h$  est le nombre de brins au niveau 0 et  $m$  le nombre de niveaux de résolution. Si chaque indice est stocké dans quatre octets, le coût mémoire en octet est de :

$$4 \times \frac{33}{9} \times (h \times 4^m) \simeq 15 \times (h \times 4^m)$$

Dans ces travaux, il a également été montré que le coût d'un quadtree est presque équivalent au coût de la 2-carte multirésolution dans ce cas.

Notre structure compacte encode un maillage équivalent dans une 2-carte



monorésolution dont le nombre de brins est :

$$h \times 4^m$$

Pour chaque brin, elle stocke deux indices ( $\phi_1$  et  $\phi_2$ ) ainsi qu'un octet additionnel pour les informations hiérarchiques. Il est résulte un coût mémoire en octets de :

$$9 \times (h \times 4^m)$$

Le ratio entre les deux structures est alors de :

$$\frac{9}{15} = 0,6$$

En d'autres termes, cette structure compacte utilise 40% de mémoire en moins que les structures équivalentes formulées de manière explicite.

Nous avons calculé le coût d'une 3-carte multirésolution et d'un octree dans le cas d'un raffinement régulier d'un maillage hexaédrique dans la section précédente. Le nombre d'indices stockés par une 3-carte multirésolution est de :

$$\frac{29}{7} \times (h \times 8^m)$$

où, comme précédemment,  $h$  est le nombre de brins au niveau 0 et  $m$  le nombre de niveaux de résolution. Si chaque indice est stocké sur quatre octets, le coût mémoire en octet est de :

$$4 \times \frac{29}{7} \times (h \times 8^m) \simeq 17 \times (h \times 8^m)$$

Nous avons montré dans la section 4.2.3 qu'un octree est 30% moins coûteux qu'une 3-carte multirésolution dans ce cas.

Notre structure compacte encode un maillage équivalent dans une 3-carte monorésolution dont le nombre de brins est :

$$h \times 8^m$$

Pour chaque brin, elle stocke trois indices ( $\phi_1$ ,  $\phi_2$  et  $\phi_3$ ) ainsi qu'un octet additionnel pour les informations hiérarchiques. Il est résulte un coût mémoire en octet de :

$$13 \times (h \times 8^m)$$

Le ratio entre les deux structures est alors de :

$$\frac{13}{17} = 0,76$$

En d'autres termes, cette structure compacte utilise 24% de mémoire en moins que la structure de 3-carte multirésolution explicite. Ceci implique également

que notre structure compacte est presque équivalente à une forêt d'octree dans ce cas.

Lors de l'examen de la complexité spatiale d'une 3-carte multirésolution, dans la section 4.2.3, nous avons conclu que le coût de la structure monorésolution impactait fortement le coût global de la représentation explicite. Il est donc normal ici que la différence de coût entre les deux représentations soit plus faible que dans le cas 2D. Cependant, l'encodage des informations hiérarchiques permet à cette représentation d'être équivalente à une forêt d'octree, tout en conservant les avantages d'un modèle à base de cartes combinatoires : la généralité des polyèdres représentables, l'exhaustivité des parcours de voisinage, la polyvalence des raffinements et la disponibilité de toutes les cellules à chaque niveau de la hiérarchie.

#### 4.3.4 Mise en œuvre et analyse de la performance

La mise en œuvre de la représentation multirésolution implicite se fait grâce à la mise en œuvre monorésolution présentée à la section 4.1.1. Nous ajoutons simplement un attribut supplémentaire au conteneur de brins pour gérer la partie hiérarchique.

Dans le chapitre 1, nous avons défini une carte plongée comme étant une carte combinatoire possédant des attributs associés aux cellules, par l'intermédiaire de fonctions qui associent à tous les brins d'une orbite un indice commun. Cet indice permet de retrouver les valeurs des attributs associés à une orbite. Dans le cas d'une représentation explicite, où chaque niveau est représenté par une carte combinatoire, cette association est naturellement présente à chaque niveau. Cependant, dans le cas d'une représentation compacte, où les cellules des niveaux intermédiaires sont implicitement représentées, il n'est pas possible de directement associer des attributs aux cellules des différents niveaux.

#### Gestion d'attributs

Pour résoudre cette carence, nous fournissons également un mécanisme permettant d'associer des attributs à chaque cellule du maillage pour chaque niveau intermédiaire de la hiérarchie. Les propriétés du raffinement primal impliquent une gestion différente des attributs de sommet de ceux des autres cellules.

### Attributs de sommet

Le raffinement primal laisse la topologie des cellules sommets inchangée. Aucun nouveau brin n'est inséré dans un sommet existant pour former une ou plusieurs nouvelles cellules. Cela signifie que l'ensemble des brins définissant un sommet ne change jamais du niveau d'insertion du sommet au niveau maximum de la hiérarchie. Un sommet est alors associé à un seul indice pour tous les niveaux existants. Les valeurs des attributs de ce sommet sont donc les mêmes à chaque niveau.

Nous proposons de maintenir un tableau d'indices par sommet afin d'associer un indice différent – et donc différentes valeurs pour les attributs – à un sommet à chaque niveau où il est présent. La taille de ce tableau est de  $(k - il)$ , où  $k$  est le niveau maximum courant et  $il$  est le niveau d'insertion du sommet. Les valeurs des attributs d'un sommet de niveau  $i$  sont associées à l'indice stocké dans la  $(i - il)$ -ième entrée de ce tableau. Ce mécanisme est illustré par la figure 4.17. Dans la figure 4.17a, les sommets A et B ont un unique indice de plongement associé pour tous les niveaux où ils existent. Dans la figure 4.17b, le sommet A possède un indice de sommet au niveau 0 (en noir) et un indice de sommet au niveau 1 (en bleu). Le sommet B possède, quant à lui, un indice de sommet au niveau 0, 1 et 2 respectivement en noir, bleu et orange.

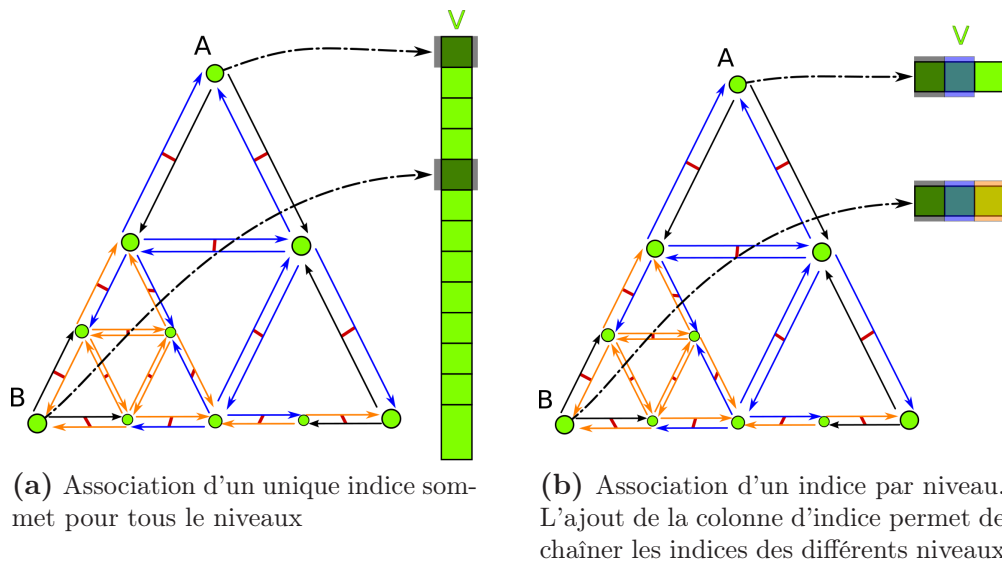
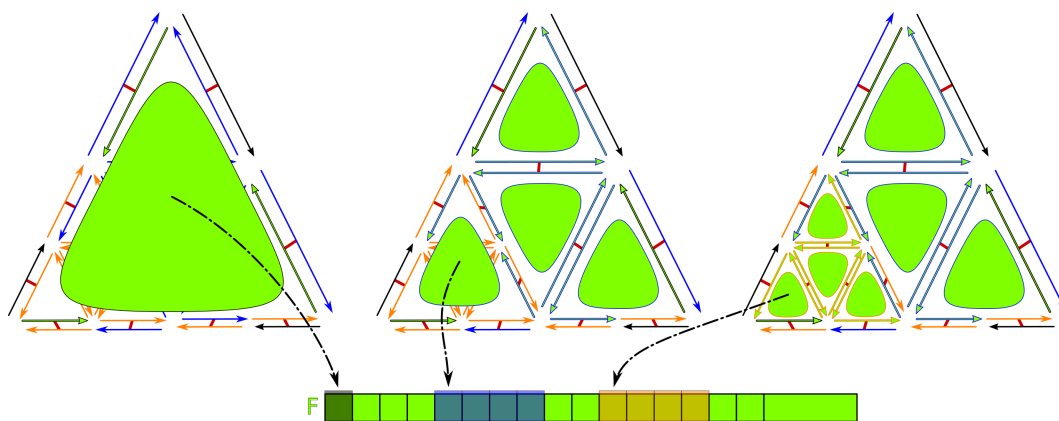


FIGURE 4.17 : Illustration des deux mécanismes pour les attributs de sommet

L'accès au bon indice est effectué en temps constant. Cependant, cette fonctionnalité induit un surcoût de mémoire dû au tableau d'indices à chaque sommet. Il est possible de la désactiver si l'application n'a besoin que d'un unique ensemble de valeurs par sommet pour tous les niveaux où il est présent



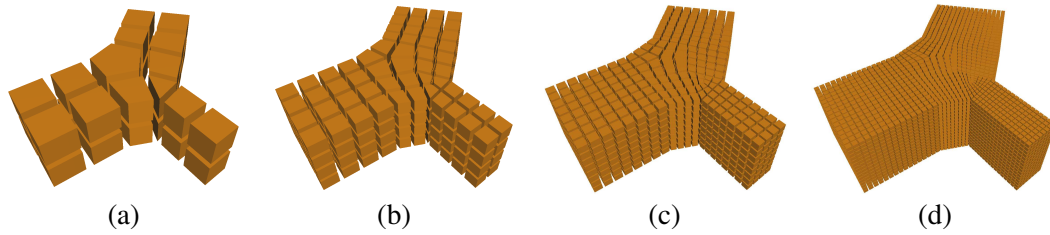
**FIGURE 4.18** : Illustration du mécanisme pour un plongement de face lors du raffinement d'une face triangulaire. Les brins possédant une information de plongement pour un niveau donné sont représentés sur fond vert pour le plongement de face et à bord de couleur en fonction de leur niveau. Sur la figure de gauche, les brins noir/vert ont un plongement de face de niveau 0. Sur la figure du milieu, les brins bleu/vert ont un plongement de face de niveau 1. Pour finir, sur la figure de droite, les brins orange/vert ont un plongement de face de niveau 2.

dans le maillage, ce qui est le cas dans les applications multirésolutions où le calcul se fait en place au sein d'un même niveau, et laisse sur les sommets des niveaux supérieurs un coefficient de détail.

### Attributs des autres cellules

Toutes les cellules de dimensions supérieures (arêtes, faces, volumes) sont découpées durant une étape de raffinement primal. L'ensemble des brins définissant une de ces cellules est composé d'un mélange de brins de différents niveaux. Dans ce contexte, nous proposons un mécanisme qui n'entraîne aucun coût mémoire supplémentaire.

Lorsqu'une cellule est raffinée, un nouvel indice est demandé pour chaque nouvelle cellule. Ces nouveaux indices sont associés uniquement aux nouveaux brins de ces cellules, c'est-à-dire ceux insérés à ce niveau. Les indices des vieux brins (ceux des niveaux précédents) sont laissés inchangés. Certaines nouvelles cellules sont composées uniquement de nouveaux brins qui sont tous associés au nouvel indice. C'est le cas des arêtes intérieures pour une quadrangulation de faces triangulaires, ou encore les faces et les arêtes intérieures d'un polyèdre raffiné. Les valeurs des attributs d'une cellule d'un niveau  $i$  sont associées à l'indice stocké au brin le plus récent de la cellule. Nous illustrons le cas d'une face triangulaire dans la figure 4.18. La face possède un indice de face au niveau



**FIGURE 4.19** : Maillages utilisés pour l'évaluation multirésolution. Ces maillages ont respectivement : (a) 75 sommets, 164 arêtes, 118 faces et 28 volumes ; (b) 385 sommets, 968 arêtes, 808 faces et 224 volumes ; (c) 2385 sommets, 6512 arêtes, 5920 faces et 1792 volumes ; (d) 16609 sommets, 47456 arêtes, 45184 faces et 14336 volumes

0, chaque face de niveau 1 possède un indice de face au niveau 1. Enfin, seule les quatre faces du bas ont un indice au niveau 2.

Ce mécanisme n'induit aucun coût mémoire supplémentaire. Cependant, l'accès au bon indice est effectué en temps linéaire en nombre de brin de la cellule.

## Analyse de la performance

Nous procédons maintenant à l'évaluation pratique de notre mise en œuvre multirésolution implicite par rapport à notre mise en œuvre explicite. Nous utilisons la séquence de maillage illustrée par la figure 4.19.

Les algorithmes de construction de la hiérarchie s'exécutent au niveau le plus fin, c'est-à-dire au niveau où les parcours ne sont pas implicites. De plus, les opérations d'étiquetages sont restreintes à un voisinage local. Ainsi, le coût engendré par ces petites opérations est négligeable en comparaison de celui des opérations de modification de la connectivité des algorithmes de raffinement. Nous pouvons donc affirmer que dans ce cas le coût est équivalent à celui d'une structure monorésolution et donc bien meilleur que pour la représentation multirésolution explicite comme l'illustre la figure 4.6.

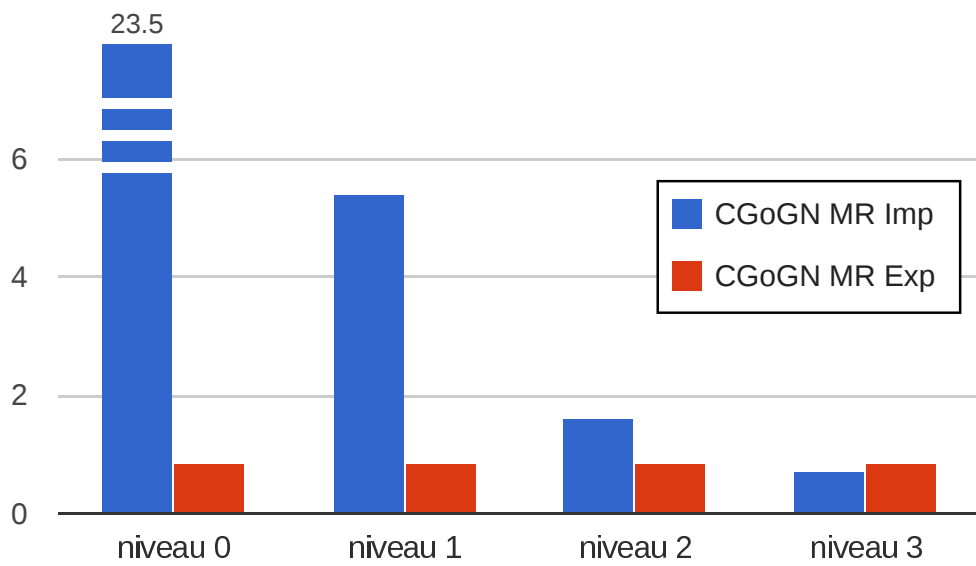
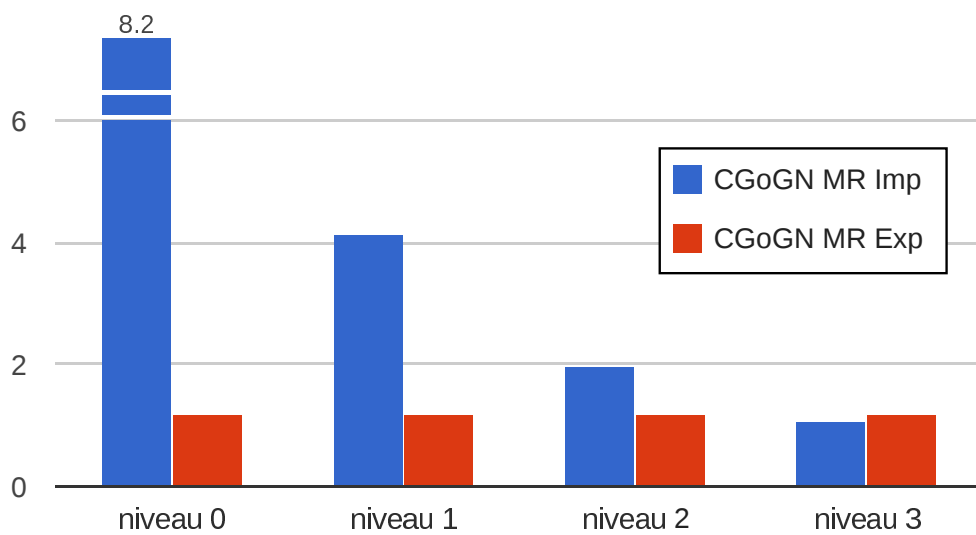
Nous évaluons uniquement les performances des algorithmes de parcours de cellules. Dans ce contexte, nous pouvons déjà faire deux hypothèses. Le niveau le plus fin profitera des performances de la mise en œuvre monorésolution car les mécanismes permettant de gérer la hiérarchie auront une très faible incidence. À l'inverse, le coût du parcours du niveau le plus grossier devrait être l'image du coût théorique calculé précédemment.

Nous reprenons à nouveau notre procédure de test des sections précédentes que nous effectuons à différents niveaux (quatre au total) d'une hiérarchie construite par un raffinement primal produisant des hexaèdres. Seulement, cette fois nous procéderons à un seul parcours global par test. Le test *Circulator* est donc scindé en deux nouveaux tests :

- *CirculatorA* : pour chaque sommet, énumérer ses volumes incidents.
- *CirculatorB* : pour chaque volume énumérer ses sommets incidents.

Séparer tous ces parcours nous permet ensuite de diviser les résultats par le nombre de cellules correspondantes à chaque niveau afin d'avoir des résultats homogènes pour tous les niveaux.

Les résultats obtenus sont présentés dans les figures 4.20 et 4.21. Nous pouvons constater que nos hypothèses de départ se vérifient. Les parcours entre le niveau 0 et le niveau 2 sont beaucoup moins performants dans le cas implicite (bleu) que dans le cas explicite (rouge). On notera cependant que le parcours au niveau maximum est un peu plus rapide pour la structure multirésolution implicite car il n'y a pas l'indirection vers le conteneur de relations à chaque accès à un brin comme dans le cas explicite.

(a) Test *Barycenter*(b) Test *Smoothing***FIGURE 4.20** : Performances temporelles. Les valeurs sont exprimées relativement aux performances de CGoGN MR.

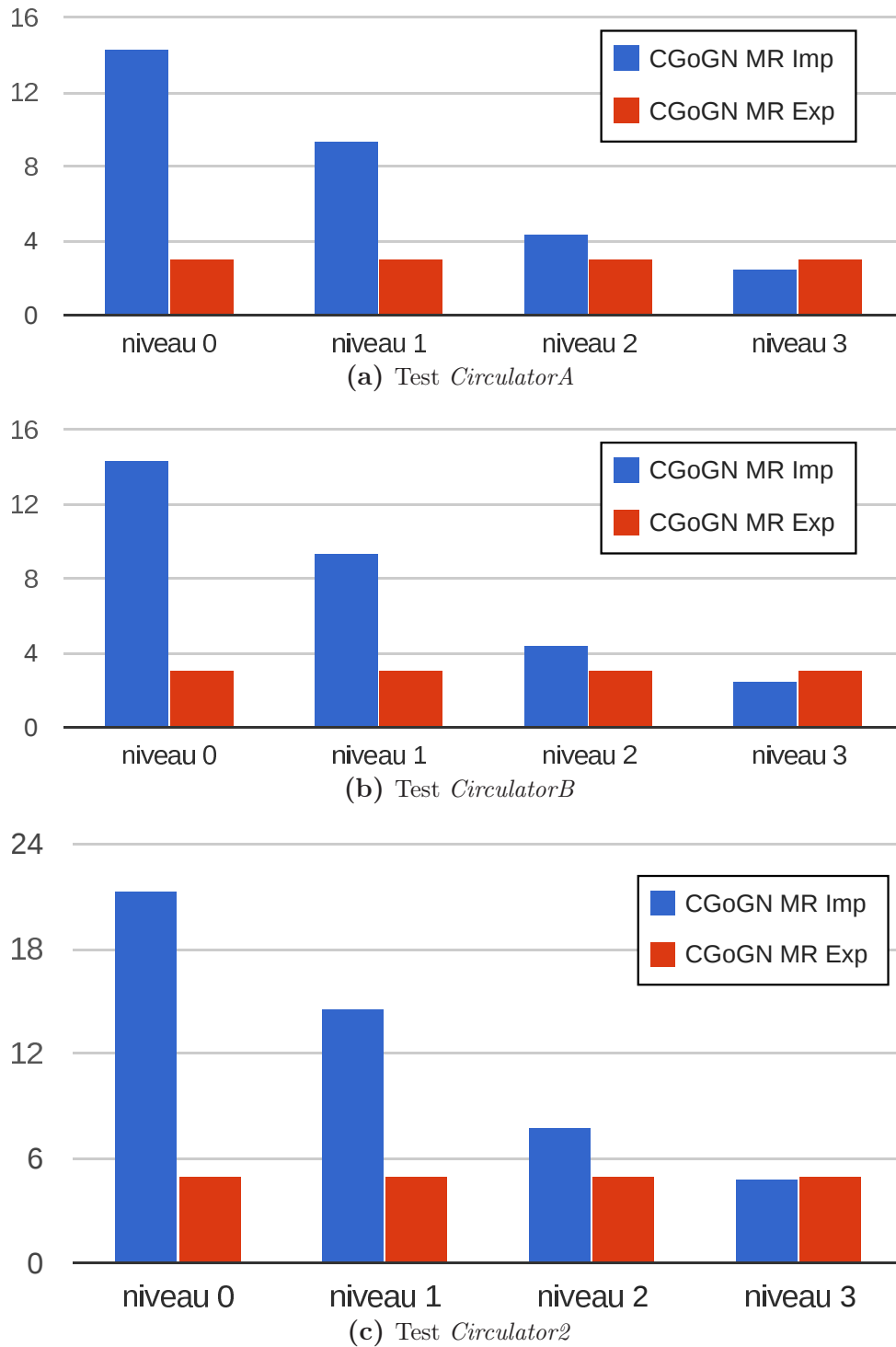


FIGURE 4.21 : Performances temporelles pour les tests de parcours. Les valeurs sont exprimées relativement aux performances de CGoGN MR.



## 4.4 Conclusion

Ce chapitre a permis de montrer comment le cadre formel des cartes combinatoires multirésolutions peut prendre corps en différentes représentations. Les deux représentations présentées suivent la définition présentée au chapitre 2. Ainsi, chaque niveau de la hiérarchie est une carte, mais représentée de manière différente : soit explicitement, soit implicitement. Ceci confère l'avantage de garder la généralité des éléments du domaine et de la dimension représentable, propre aux cartes combinatoires. Enfin, il est également possible d'attacher des attributs à toutes les cellules (sommet, arête, face, volume, etc.) de la hiérarchie. Cependant, ces deux représentations ont chacune leurs avantages et leurs inconvénients.

La première représentation explicite chaque niveau de la hiérarchie. Elle a un coût constant des requêtes de voisinage pour chacun des niveaux de résolutions. Elle supporte tout type de raffinement ( $\sqrt{3}$ , primal, etc.) de la topologie. Enfin, elle permet également d'autres types de mode de construction de la hiérarchie. Elle a cependant le désavantage d'être plus coûteuse en mémoire que d'autres structures plus spécifiques.

La seconde représentation explicite uniquement le niveau le plus fin de la hiérarchie. Les autres niveaux sont reconstruits de manière implicite à la demande. Elle a l'avantage d'avoir un coût mémoire très faible pour une structure représentant une connectivité complète à chaque niveau de la hiérarchie. Elle est cependant contrainte aux raffinements primaux et entraîne un surcoût des requêtes de voisinage.



# CONCLUSION

---

Les travaux de recherche présentés dans les chapitres précédents avaient un double objectif. Il s'agissait tout d'abord de concevoir des algorithmes de raffinement volumique suffisamment génériques pour être capables de supporter des maillages volumiques quelconques. Il s'agissait ensuite d'offrir différentes représentations pour une hiérarchie de maillages construit à partir de ces raffinements. Nos travaux se sont appuyés sur le modèle des cartes combinatoires déjà utilisé en modélisation géométrique, et notamment dans le cadre des surfaces de subdivision multirésolutions.

Nous résumons ici les différentes contributions, avant de présenter les perspectives à plus ou moins longs termes relatives à ces travaux.

## 1 Rappel des contributions

Notre première contribution consiste en une extension multirésolution du modèle des cartes combinatoires en dimension quelconque, dérivée du cadre général des hypercartes multirésolutions. Ce modèle définit une carte multirésolution comme un ensemble de cartes imbriquées les unes dans les autres. Ainsi, chaque niveau de résolution correspond à un ensemble de brins progressivement enrichi. Nous avons également conçu et développé un ensemble

d'opérations topologiques de bas niveau, nécessaires à la génération de hiérarchies de maillages multirésolutions construits par des subdivisions de cellules.

Dans le cadre des volumes de subdivision multirésolutions, nous avons produit plusieurs algorithmes de raffinement d'un maillage grossier de départ. Nous avons illustré leur généricité et leur flexibilité pour générer des hiérarchies aussi bien par subdivision régulière que adaptative.

Pour notre dernière contribution, nous avons proposé et détaillé deux représentations des cartes combinatoires multirésolutions. Pour ces deux représentations, nous avons évalué le coût mémoire et la complexité des parcours. Au final, nous avons montré que les deux modèles proposés présentent des avantages de plusieurs ordres.

Revenons sur les points forts de nos contributions.

## 1.1 Généricité des méthodes

Les structures de données multirésolutions de la littérature et leurs algorithmes de raffinement sont étroitement liées les unes aux autres. Il faut donc développer autant de structures de données que de schémas de subdivision et inversement. Par exemple, les structures à base d'arbres octals ne permettent que la subdivision en cube.

Nos algorithmes de raffinement sont exprimés dans un cadre qui les rend adaptés à tout type de maillage. Cette propriété leur confère une grande souplesse pour la conception ou le mélange de schémas de subdivision. Elle nous a permis, par exemple, d'utiliser simultanément la quadrisection de faces polygonales et triangulaires sur des polyèdres afin de maintenir des polyèdres spécifiques (prisme, pyramide, etc.). Elle nous autorise également le mélange d'opérations de subdivision et de « flip » de tétraèdres pour des schémas tels que le  $\sqrt{3}$  volumique.

Nos algorithmes de raffinement de volumes sont décrits en utilisant des opérations topologiques de bas niveau. Cette caractéristique nous assure de maîtriser la combinatoire des objets manipulés notamment dans le cas de polyèdres quelconques. De plus elle nous permet de minimiser le nombre de brins modifiés lors des changements de relations topologiques.

Enfin, la formalisation de l'extension multirésolution, pour un modèle mathématique indépendant d'une implémentation sous forme de structures de

données concrètes, nous donne accès à une certaine genericité. Nous avons ainsi obtenu deux structures de données supportant les mêmes modèles et opérations mais avec des caractéristiques différentes (en terme de coût).

## 1.2 Préservation de la cohérence topologique

Les structures multirésolutions de la littérature ne représentent pas l'ensemble des cellules (sommets, arêtes, faces, volumes, etc.) de la subdivision. Ainsi, dans le cas tridimensionnel, seuls les cubes sont représentés pour des maillages hexaédriques (octree) ou encore les arêtes, les tétraèdres et les octaèdres pour des maillages tétraédriques/octaédriques (hiérarchie de tétraèdres/octaèdres). Les cartes multirésolutions ont la faculté de représenter des maillages polyédriques quelconques, en même temps que toutes les cellules de la subdivision, et que l'ensemble des relations topologiques liant ces cellules. Cette propriété leur confère la capacité de maintenir la consistance du maillage pour tous les niveaux de résolution et pour tout schéma de subdivision.

Dans le cadre d'un raffinement adaptatif, la non représentation de certaines cellules par les structures ad hoc engendre des trous, communément appelés jonctions en « T », dans la topologie des maillages représentés. Ces trous engendrent des problèmes de conformité aux frontières des cellules subdivisées à des niveaux différents. Les solutions proposées consistent généralement en un ajustement spécifique de la méthode et limitent donc les possibilités de ladite méthode comme dans le cas tétraédrique/octaédrique.

Au sein des cartes combinatoires multirésolutions, les opérations de raffinement sont décomposées en une suite d'opérations agissant sur les cellules de chaque dimension. Cette gestion constructive, additionnée au calcul du niveau de subdivision d'une cellule, permet de faire évoluer le degré des cellules de manière cohérente avec le voisinage sans ajout de cellules temporaires ou spécifiques. Elle offre, en outre, la garantie de la représentation d'une variété (ou quasi-variété) orientable de dimension 3 tout au long du cycle de vie de la représentation multirésolution.

## 1.3 Efficacité des méthodes

Nous avons expérimenté des algorithmes de traitement de la géométrie sur nos modèles comme l'analyse multirésolution et la subdivision, par des schémas variés. Pour ces applications, le calcul de la position des sommets dépend de la

position des sommets voisins et nécessité de nombreuses requêtes de voisinage, et par extension, de nombreux parcours de la topologie.

Les structures arborescentes répondent à ces requêtes en un temps dépendant de leur profondeur. Les cartes combinatoires multirésolutions explicites représentent chaque maillage de la hiérarchie par une carte directement accessible à tout moment. Cette formulation permet de répondre aux requêtes de voisinage en temps optimal quel que soit le niveau de résolution considéré.

Pour des applications pour lesquelles peu de niveaux de résolution sont nécessaires, la complexité des opérations de reconstruction de la représentation multirésolution implicite reste faible. Le coût de reconstruction peut alors être amorti par le coût du traitement appliqué, tout en gardant l'efficacité du stockage de la hiérarchie.

#### 1.4 Flexibilité des méthodes

Nous avons privilégié une approche combinatoire et formelle basée sur un modèle défini mathématiquement. Ce modèle s'exprime sous la forme d'involutions et non de pointeurs dans une structure de données. Cette caractéristique nous laisse une certaine liberté pour mettre en œuvre ce modèle en une structure de données. Cette flexibilité concerne aussi bien les cartes combinatoires que leur extension multirésolution. Ainsi, il est tout à fait envisageable de proposer des implantations, différentes de celles que nous avons proposées, et notamment dédiées à des utilisations spécifiques par exemple pour des approches « hors mémoire » (out-of-core), parallèles, etc.

Une mise en œuvre classique des cartes combinatoires consiste en une liste de brins possédant des pointeurs pour chaque relation du modèle vers d'autres brins de la liste. Nous avons choisi de nous appuyer sur la mise en œuvre mono-résolution performante, basée sur des tableaux, disponible dans CGoGN [70]. À partir de là, nous avons proposé une mise en œuvre pour chacune de nos représentations multirésolutions.

La représentation multirésolution explicite offre un accès efficace à chaque niveau de résolution, quelque soit leur nombre. Cette structure de données est particulièrement adaptée aux applications nécessitant une utilisation intensive des voisinages comme pour les outils de traitement de la géométrie.

La représentation multirésolution implicite permet des subdivisions et simplifications, régulières ou adaptatives, des cellules, sans devoir maintenir une

structure de données hiérarchique ad hoc. Elle offre un contrôle sur le niveau de subdivision des cellules dans une hiérarchie de maillages en minimisant le coût mémoire. Cette structure de données convient davantage aux applications où la résolution doit être adaptée suivant un critère donné.

## 2 Perspectives

Les perspectives à ces travaux sont nombreuses. Nous les présentons en les regroupant autour de trois axes : l'intégration d'autres méthodes de raffinement, l'évolution du modèle de représentation et les applications pouvant tirer parti de nos méthodes.

Concernant les volumes de subdivision multirésolutions, il existe d'autres schémas de subdivision et notamment le schéma  $\sqrt[3]{2}$  [99]. Ce raffinement est naturellement adaptatif car la décomposition s'opère en gardant une classe de polyèdres réguliers (hexaèdres, pyramides, tétraèdres) et permet ainsi d'éviter l'apparition des jonctions en T. Ce raffinement peut également être mis en œuvre par l'intermédiaire du modèle des cartes combinatoires de dimension 3 car les opérations de raffinement peuvent également s'exprimer à l'aide de nos opérations topologiques de bas niveau. À l'image de notre algorithme de raffinement mixte mélangeant le raffinement de faces quadrangulaires et de faces triangulaires produisant une classe de polyèdres réguliers (hexaèdres, pyramides, prismes, octaèdres, tétraèdres), il doit être possible de mélanger ces deux raffinements de volumes sur un même objet volumique en faisant l'hypothèse de rendre compatible les zones frontières. Ce mélange permettrait de varier les algorithmes de subdivision au sein d'un même objet et d'obtenir une subdivision adaptative sans jonction en T, tout en gardant des polyèdres de même forme que ceux qui les ont générés. Caractériser la limite du processus de subdivision géométrique dans les zones frontières peut se révéler difficile. Cependant, préserver la forme des polyèdres pendant le raffinement est une caractéristique importante pour certaines méthodes numériques (éléments finis), ou encore pour mieux contrôler le comportement de particules (pas exemple leur circulation) dans des subdivisions volumiques multirésolutions non limitées à la subdivision d'un cube.

Une autre voie à explorer concerne les schémas de subdivision volumique duaux. Le raffinement pour ce type de schéma est moins trivial à expliciter. En effet, le raffinement ne se limite pas à la découpe de cellules existantes mais fait intervenir la création de nouvelles cellules. Par exemple, un sommet de degré  $n$  se transforme en un polyèdre à  $n$  faces. Ces nouvelles cellules ont donc une combinatoire complexe dans le cas quelconque. Cependant, l'utilisation

des cartes combinatoires permet de maîtriser la combinatoire de ces nouveaux éléments à insérer.

Nous avons travaillé sur la représentation d'une hiérarchie dans un cas très général représentant des polyèdres quelconques. Nous avons vu que le modèle des cartes combinatoires, à travers la notion de brin, rend disponible toutes les relations entre les cellules de chaque dimension. Ce modèle sera donc toujours plus coûteux que des structures de données n'explicitant pas toutes ces relations et plus encore vis à vis de structures de données spécifiques à certains types de polyèdres. Cependant, en contraignant le domaine de représentation du modèle, il est possible de minimiser cet écart en encodant une ou plusieurs relations pour chaque brin, ou le brin (et ses relations) lui-même. Nous pourrions donc obtenir des structures de données hiérarchiques encore moins coûteuses en mémoire que celles que nous proposons au prix d'un surcoût en temps lié à la reconstruction des relations du modèle. L'avantage de formuler toutes ces représentations dans un même cadre formalisé réside dans le fait de pouvoir garder la validité des algorithmes sur les cartes et notamment nos algorithmes de raffinement.

Une autre évolution de notre modèle consiste à élargir le domaine de cette étude à l'approche « fin vers grossier ». Envisager des techniques multirésolutions telles que le filtrage ou la compression de maillages ayant une connectivité quelconque nécessite également, à l'instar des volumes de subdivision, l'usage intensif de parcours de voisinage. Ce type de méthodes vise à obtenir une hiérarchie de maillages progressifs construits à partir de maillages tétraédriques irréguliers. L'utilisation de structures classiques amène à déplacer un front dans une structure à base de graphes et ainsi à effectuer un certain nombre d'opérations topologiques. Encore une fois, l'utilisation d'un modèle à base de cartes combinatoires permettrait d'avoir les relations de chaque niveau de résolution directement accessibles.

Il est également possible de faire évoluer le modèle des cartes combinatoires multirésolutions lui-même. Dans notre définition donnée au chapitre 2, les brins sont inclus dans des ensembles imbriqués les uns dans les autres de manière linéaire. Il est tout à fait possible d'envisager une structure arborescente pour remplacer cette structure linéaire. Chaque nœud de l'arbre représenterait un ensemble de brins contenant les brins des nœuds parents. Ainsi, chaque nœud correspondrait à une version du maillage pour une résolution donnée. Cette structuration permettrait l'accès à des hiérarchies différentes issues d'un même maillage initial.

Par ailleurs, ces ensembles de brins imbriqués sont construits par l'ajout successif de nouveaux brins. Une autre évolution de notre modèle multirésol-



lution consiste à autoriser la suppression de brins afin de créer une hiérarchie où des brins apparaissent et disparaissent. Une manière de gérer cette nouvelle fonctionnalité consiste à ajouter à chaque brin une information concernant son niveau de suppression. Cette modification permet de représenter l'évolution dans le temps de maillages avec, comme application, l'analyse spatio-temporelle de séquences de maillages quelconques.

Enfin, une autre perspective consiste à utiliser notre modèle formel dans le cadre théorique de la preuve d'algorithmes en s'inscrivant dans la lignée des travaux de Dufourd [11]. Ici, il s'agit d'utiliser les méthodes de preuves formelles pour prouver des propriétés sur les algorithmes agissant sur des séquences de maillages. Ces travaux prolongent notre travail sur la définition en dimension  $n$  du raffinement primal.

En terme d'applications, nous visons l'intégration de nos algorithmes de subdivision multirésolutions à des méthodes de remaillages dynamiques dans le cadre d'applications de simulation physique ou encore la compression de maillages volumiques. Nous souhaitons également apporter nos méthodes hiérarchiques et multirésolutions au rendu volumique par grille utilisant un échantillonnage différent du cubique cartésien [104].

Plus largement, nos méthodes robustes et définies en dimension  $n$  se posent comme un socle solide afin de se donner des outils d'analyse multirésolution de gros volumes de données hétérogènes. En effet, le phénomène des gros volumes de données, également appelé « Big Data », pose de nombreuses questions. Les principales concernent la visualisation de ces données et l'utilisation de métaphores d'interaction efficaces. Cependant, comme nous l'avons vu au fil de ce travail, la représentation des données est une question importante afin d'envisager d'autres applications tel que l'analyse, la compression ou encore la manipulation à différentes échelles.

# Table des figures

|      |  |    |
|------|--|----|
| 1    | Exemple de subdivision d'un maillage volumique polyédrique. . . . .  | 5  |
| 1.1  | La structure octet-truss. . . . .                                    | 11 |
| 1.2  | Exemple d'un 2-cube et d'un 3-cube. . . . .                          | 12 |
| 1.3  | Exemple de subdivision d'un 2-cube et d'un 3-cube. . . . .           | 13 |
| 1.4  | Décomposition cellulaire d'un objet tridimensionnel. . . . .         | 16 |
| 1.5  | Graphe d'incidence d'un objet en 3D. . . . .                         | 17 |
| 1.6  | Construction itérative de cell-tuples. . . . .                       | 18 |
| 1.7  | Exemple d'une quasi-variété de dimension 3. . . . .                  | 19 |
| 1.8  | Illustration d'une 3-g-carte. . . . .                                | 20 |
| 1.9  | Construction d'une 3-carte à partir d'une 3-g-carte. . . . .         | 22 |
| 1.10 | Orbites duales dans une 3-carte. . . . .                             | 23 |
| 1.11 | Orbites primales dans une 3-carte. . . . .                           | 24 |
| 1.12 | Opération de contraction d'arête et d'éclatement de sommet. . . . .  | 29 |
| 1.13 | Raffinement d'un octree. . . . .                                     | 30 |
| 1.14 | Raffinement adaptatif d'un octree. . . . .                           | 31 |
| 1.15 | Les trois types de pavages respectivement CC, FCC et BCC. . . . .    | 32 |
| 1.16 | Raffinement régulier de simplexe. . . . .                            | 33 |
| 1.17 | Opération de bisection de simplexe en 2D et 3D. . . . .              | 34 |
| 1.18 | Configuration d'arêtes en 2D et 3D. . . . .                          | 34 |
| 1.19 | Classes de diamants en 2D et 3D. . . . .                             | 35 |
| 1.20 | Arbres de diamants et de simplexes en 2D. . . . .                    | 36 |
| 1.21 | Raffinement d'un octaèdre. . . . .                                   | 37 |
| 2.1  | Ensembles de brins dans une hypercarte multirésolution. . . . .      | 45 |
| 2.2  | Indexation des relations dans une hypercarte de dimension 3. . . . . | 46 |
| 2.3  | Opérations élémentaires duales de couture et découteure. . . . .     | 52 |
| 2.4  | Découpe et découpe inverse d'une arête en 1D. . . . .                | 54 |
| 2.5  | Découpe et découpe inverse d'une arête en 2D. . . . .                | 54 |
| 2.6  | Découpe et découpe inverse d'une arête en 3D. . . . .                | 55 |
| 2.7  | Insertion et suppression d'un sommet dans une arête en 3D. . . . .   | 56 |
| 2.8  | Découpe de face en 2D. . . . .                                       | 56 |
| 2.9  | Découpe de face en 3D. . . . .                                       | 57 |
| 2.10 | Insertion d'une arête dans une face en 3D. . . . .                   | 57 |
| 2.11 | Découpe de volume. . . . .   | 58 |

---

|      |  |     |
|------|--|-----|
| 2.12 | Insertion d'une face dans un volume. . . . .                             | 59  |
| 2.13 | Fusion de volumes. . . . .   | 59  |
| 2.14 | Fusion de deux volumes le long de leur face commune en primal. . . . .   | 60  |
| 3.1  | Schéma de Catmull-Clark. . . . .   | 65  |
| 3.2  | Schéma de MacCracken-Joy. . . . .  | 65  |
| 3.3  | Raffinements primaux en 2D. . . . .                                      | 66  |
| 3.4  | Raffinement dual en 2D. . . . .  | 68  |
| 3.5  | Exemple de maillage obtenu par subdivision régulière. . . . .            | 73  |
| 3.6  | Quadrisection de faces triangulaires et polygonales. . . . .             | 74  |
| 3.7  | Quadrisection d'un hexaèdre. . . . .                                     | 75  |
| 3.8  | Quadrisection de faces quadrangulaires au sein d'une 3-carte MR. . . . . | 75  |
| 3.9  | Quadrisection d'hexaèdres au sein d'une 3-carte MR. . . . .              | 76  |
| 3.10 | Illustration de volumes de type-n. . . . .                               | 77  |
| 3.11 | Quadrisection de tétraèdres. . . . .                                     | 78  |
| 3.12 | Quadrisection d'octaèdres. . . . .                                       | 79  |
| 3.13 | Quadrisection de faces triangulaires au sein d'une 3-carte MR. . . . .   | 79  |
| 3.14 | Quadrisection de tétraèdres au sein d'une 3-carte MR. . . . .            | 80  |
| 3.15 | Quadrisection d'octaèdres au sein d'une 3-carte MR. . . . .              | 81  |
| 3.16 | Quadrisection de prismes. . . . .  | 82  |
| 3.17 | Subdivision mixte au sein d'une 3-carte MR. . . . .                      | 83  |
| 3.18 | Relations à reconstruire dans une face subdivisée. . . . .               | 84  |
| 3.19 | Étapes de subdivision d'une face. . . . .                                | 85  |
| 3.20 | Permutation de faces autour d'une arête. . . . .                         | 86  |
| 3.21 | Subdivision topologique. . . . .   | 87  |
| 3.22 | Exemple de subdivision adaptative. . . . .                               | 88  |
| 3.23 | Subdivision adaptative d'un hexaèdre dans un maillage . . . . .          | 89  |
| 3.24 | Différents niveaux d'une face. . . . .                                   | 90  |
| 3.25 | Brin d'une face dans une 3-carte MR. . . . .                             | 91  |
| 3.26 | Brin d'une arête dans une 3-carte MR. . . . .                            | 91  |
| 3.27 | Brin d'un volume dans une 3-carte MR. . . . .                            | 92  |
| 3.28 | Calcul du niveau d'une face. . . . .                                     | 95  |
| 3.29 | Exemple de raffinement adaptatif. . . . .                                | 97  |
| 3.30 | Masques du bord pour MacCracken-Joy. . . . .                             | 98  |
| 3.31 | Masque pour un sommet de sommet. . . . .                                 | 99  |
| 3.32 | Masque pour un sommet d'arête. . . . .                                   | 100 |
| 3.33 | Masque pour un sommet de face. . . . .                                   | 101 |
| 3.34 | Masque pour un sommet de volume. . . . .                                 | 101 |
| 3.35 | Masque pour un sommet impair de volume. . . . .                          | 103 |
| 3.36 | Masque pour un sommet impair de face. . . . .                            | 103 |
| 3.37 | Masque pour un sommet impair d'arête. . . . .                            | 103 |
| 3.38 | Masque pour un sommet pair de sommet. . . . .                            | 104 |
| 3.39 | Masque pour un sommet pair d'arête. . . . .                              | 105 |

|      |  |     |
|------|--|-----|
| 3.40 | Masque pour un sommet pair de face. . . . .                          | 106 |
| 3.41 | Diagramme de flux des données. . . . .                               | 107 |
| 3.42 | Illustration du pipeline de l'application élethro-thermique. . . . . | 108 |
| 3.43 | Schéma $\sqrt{3}$ au sein d'une 3-carte MR. . . . .                  | 110 |
| 3.44 | Opération de split 1-4 au sein d'une 3-carte MR. . . . .             | 112 |
| 3.45 | Opération de swap 2-3 au sein d'une 3-carte MR. . . . .              | 113 |
| 3.46 | Opération de transformation face/arête dans une 3-carte MR. . . . .  | 113 |
| 3.47 | Opération de split 1-3 au sein d'une 3-carte MR. . . . .             | 114 |
| 3.48 | Opération de swap 2-2 au sein d'une 3-carte MR. . . . .              | 115 |
| 3.49 | Opération de swap 4-4 au sein d'une 3-carte MR. . . . .              | 115 |
| 3.50 | Opération de swap généralisé 3-2 au sein d'une 3-carte MR. . . . .   | 116 |
| 3.51 | Raffinement $\sqrt{3}$ adaptatif à l'intérieur. . . . .              | 119 |
| 3.52 | Raffinement $\sqrt{3}$ adaptatif sur le bord. . . . .                | 120 |
| 3.53 | Masques de subdivision des sommets du bord. . . . .                  | 121 |
| 3.54 | Masques de subdivision des sommets de l'intérieur. . . . .           | 121 |
|      |  |     |
| 4.1  | Implantation monorésolution. . . . .                                 | 126 |
| 4.2  | Maillages d'évaluation. . . . .                                      | 127 |
| 4.3  | Performances monorésolution. . . . .                                 | 128 |
| 4.4  | Implantation multirésolution explicite. . . . .                      | 136 |
| 4.5  | Duplication des brins. . . . .                                       | 138 |
| 4.6  | Performances temporelles de la multirésolution explicite. . . . .    | 139 |
| 4.7  | Représentation implicite en 2D. . . . .                              | 140 |
| 4.8  | Relation $\phi_1^i$ en 2D. . . . .                                   | 141 |
| 4.9  | Étiquettes d'arêtes. . . . .   | 142 |
| 4.10 | Relation $\phi_2^i$ en 2D. . . . .                                   | 144 |
| 4.11 | Représentation implicite en 3D. . . . .                              | 145 |
| 4.12 | Étiquettes de face. . . . .  | 146 |
| 4.13 | Relation $\phi_1^i$ en 3D. . . . .                                   | 147 |
| 4.14 | Relation $\phi_2^i$ en 3D. . . . .                                   | 149 |
| 4.15 | Relation $\phi_3^i$ en 3D. . . . .                                   | 149 |
| 4.16 | Complexité temporelle implicite. . . . .                             | 151 |
| 4.17 | Attributs de sommet implicites. . . . .                              | 155 |
| 4.18 | Attributs de cellule implicites. . . . .                             | 156 |
| 4.19 | Maillages d'évaluation multirésolution. . . . .                      | 157 |
| 4.20 | Performances temporelles. . . . .                                    | 159 |
| 4.21 | Performances temporelles. . . . .                                    | 160 |

# BIBLIOGRAPHIE

---

- [1] P. ALLIEZ et C. GOTSMAN : Recent advances in compression of 3d meshes. *In* N. A. DODGSON, M. S. FLOATER et M. A. SABIN, édés : *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, p. 3–26. Springer Berlin Heidelberg, 2005. [70]
- [2] M. BABA-ALI, G. DAMIAND, X. SKAPIN et D. MARCHEIX : Insertion and expansion operations for  $n$ -dimensional generalized maps. *In* D. COEUR-JOLLY, I. SIVIGNON, L. TOUGNE et F. DUPONT, édés : *Discrete Geometry for Computer Imagery*, vol. 4992 de *Lecture Notes in Computer Science*, p. 141–152. Springer Berlin Heidelberg, 2008. [39]
- [3] C. L. BAJAJ et L. C. KARLPALEM : Volume subdivision based hexahedral finite element meshing of domains with interior 2-manifold boundaries. *In Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, AFRIGRAPH '06, p. 127–136, New York, NY, USA, 2006. ACM. [70]
- [4] C. L. BAJAJ, S. SCHAEFER, J. WARREN et G. XU : A subdivision scheme for hexahedral meshes. *The Visual Computer*, 18(5-6):343–356, août 2002. [12, 65]
- [5] M. BERTRAM : Biorthogonal wavelets for subdivision volumes. *In Proceedings of the seventh ACM symposium on Solid modeling and applications*, SMA '02, p. 72–82, New York, NY, USA, 2002. ACM. [70, 102,

- 122]
- [6] Y. BERTRAND : Topofil : un modeleur interactif d'objets 3d à base topologique. *Technique et Science Informatiques (TSI)*, 17(4):443–484, avr. 1998. [125]
  - [7] G.-P. BONNEAU, G. ELBER, S. HAHMANN et B. SAUVAGE : Multiresolution Analysis. In L. DE FLORIANI et M. SPAGNUOLO, édés : *Shape Analysis and Structuring*, Mathematics and Visualization, p. 83–114. Springer Berlin Heidelberg, 2008. [70]
  - [8] M. BOTSCH, S. STEINBERG, S. BISCHOFF et L. KOBBELT : Openmesh - a generic and efficient polygon mesh data structure. In *Proceedings of the OpenSG Symposium*, 2002. [26]
  - [9] E. BRISSON : Representing geometric structures in  $d$  dimensions: topology and order. In *Proceedings of the 5th annual symposium on Computational geometry*, SCG '89, p. 218–227, New York, NY, USA, 1989. ACM. [17, 19]
  - [10] E. BRISSON : Representing geometric structures in  $d$  dimensions: Topology and order. *Discrete & Computational Geometry*, 9(1):387–426, 1993. [17]
  - [11] C. BRUN, J.-F. DUFOURD et N. MAGAUD : Designing and proving correct a convex hull algorithm with hypermaps in coq. *Computational Geometry: Theory and Applications*, 45(8):436–457, oct. 2012. [25, 169]
  - [12] L. BRUN et W. G. KROPATSCH : Combinatorial pyramids. In *IEEE International Conference on Image Processing*, vol. 2, p. 33–37. IEEE Computer Society, sept. 2003. [39]
  - [13] L. BRUN et W. G. KROPATSCH : Receptive fields within the combinatorial pyramid framework. *Graphical Models*, 65(1–3):23–42, 2003. [39]
  - [14] L. BRUN et W. G. KROPATSCH : Contains and inside relationships within combinatorial pyramids. *Pattern Recognition*, 39(4):515–526, 2006. [39]
  - [15] D. BURKHART, B. HAMANN et G. UMLAUF : Adaptive and feature-preserving subdivision for high-quality tetrahedral meshes. *Computer Graphics Forum*, 29(1):117–127, mars 2010. [4, 65, 109, 110, 120, 122]
  - [16] D. BURKHART, B. HAMANN et G. UMLAUF : Adaptive tetrahedral subdivision for finite element analysis. In *Computer Graphics International, Singapore*, 2010. short paper. [69, 109]

- [17] D. BURKHART, B. HAMANN et G. UMLAUF : Iso-geometric finite element analysis based on catmull-clark subdivision solids. *Computer Graphics Forum*, 29(5):1575–1584, juil. 2010. [69]
- [18] C. BURSTEDDE, L. C. WILCOX et O. GHATTAS : p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, mai 2011. [31]
- [19] S. CAPELL, S. GREEN, B. CURLESS, T. DUCHAMP et Z. POPOVIĆ : A multiresolution framework for dynamic deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, SCA '02, p. 41–47, New York, NY, USA, 2002. ACM. [28, 70]
- [20] E. CATMULL et J. CLARK : Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 10(6):350–355, nov. 1978. [65, 98]
- [21] D. CAZIER : *Construction de systèmes de réécriture pour les opérations booléennes en modélisation géométrique*. Thèse de doctorat, Université de Strasbourg, Strasbourg, FR, nov. 1997. [24, 53]
- [22] D. CAZIER et J.-F. DUFOURD : A formal specification of geometric refinements. *The Visual Computer*, 15(6):279–301, 1999. [24]
- [23] CGAL : Computational Geometry Algorithms Library. <http://www.cgal.org>. [26]
- [24] CGoGN : Combinatorial and Geometric modeling with Generic N-dimensional maps. <http://cgogn.unistra.fr>. [27, 124, 125]
- [25] G. M. CHAIKIN : An algorithm for high-speed curve generation. *Computer Graphics and Image Processing*, 3(4):346–349, déc. 1974. [65]
- [26] Y.-S. CHANG, K. T. MCDONNELL et H. QIN : A new solid subdivision scheme based on box splines. In *Proceedings of the seventh ACM symposium on solid modeling and applications*, SMA '02, p. 226–233, New York, NY, USA, 2002. ACM. [3, 65]
- [27] Y.-S. CHANG, K. T. MCDONNELL et H. QIN : An interpolatory subdivision for volumetric models over simplicial complexes. In *Proceedings of the Shape Modeling International 2003*, SMI '03, p. 143–152, Washington, DC, USA, 2003. IEEE Computer Society. [65]
- [28] P. CIGNONI, L. DE FLORIANI, P. MAGILLO, E. PUPPO et R. SCOPIGNO : Selective refinement queries for volume visualization of unstructured te-

- trahedral meshes. *IEEE Transactions on Visualization and Computer Graphics*, 10(1):29–45, jan. 2004. [28, 38]
- [29] A. COHEN : *Numerical Analysis of Wavelet Methods*, vol. 32 de *Studies in Mathematics and its Applications*. Elsevier Science, 2003. [70]
- [30] R. CORI : *Un code pour les graphes planaires et ses applications*. Num. 27 de Astérisque. Société mathématique de France, juin 1975. [26]
- [31] R. CORI : Hypermaps and indecomposable permutations. *European Journal of Combinatorics*, 30(2):540–541, fév. 2009. [26]
- [32] R. CORI et A. MACHI : Maps, hypermaps and their automorphisms: a survey i. In *Expositiones Mathematicae*, vol. 10, p. 403–427, 1992. [26]
- [33] F. H. CROOM : *Basic Concepts of Algebraic Topology*. Undergraduate Texts in Mathematics. Springer, 1978. [2]
- [34] G. DAMIAND : Combinatorial maps. In CGAL EDITORIAL BOARD, éd. : *CGAL User and Reference Manual*. 2012. 4.0 Edition. [27, 125, 127]
- [35] G. DAMIAND : Linear cell complex. In CGAL EDITORIAL BOARD, éd. : *CGAL User and Reference Manual*. 2012. 4.0 Edition. [125, 127]
- [36] G. DAMIAND et P. LIENHARDT : Removal and contraction for  $n$ -dimensional generalized maps. In I. NYSTRÖM, G. Sanniti di BAJA et S. SVENSSON, édés : *Discrete Geometry for Computer Imagery*, vol. 2886 de *Lecture Notes in Computer Science*, p. 408–419. Springer Berlin Heidelberg, 2003. [39]
- [37] E. DANOVARO et L. DE FLORIANI : Half-edge multi-tessellation: A compact representations for multiresolution tetrahedral meshes. In *1st International Symposium on 3D Data Processing Visualization and Transmission*, p. 494–499. IEEE Computer Society, 2002. [39]
- [38] E. DANOVARO, L. DE FLORIANI, P. MAGILLO et N. SOKOLOVSKY : Data structures for encoding lod models generated through half-edge collapse. Rap. tech. DISI-TR-01-06, Department of Computer and Information Science, University of Genova, Genova (Italy), 2003. [39]
- [39] E. DANOVARO, L. FLORIANI, P. MAGILLO et E. PUPPO : Data structures for 3d multi-tessellations: an overview. In F. H. POST, G. M. NIELSON et G.-P. BONNEAU, édés : *Data Visualization*, vol. 713 de *The Springer International Series in Engineering and Computer Science*, p. 239–255. Kluwer Academic Publishers, 2003. [38]



- [40] L. DE FLORIANI, L. KOBELT et E. PUPPO : A survey on data structures for level-of-detail models. In N. A. DODGSON, M. S. FLOATER et M. A. SABIN, édés : *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, p. 49–74. Springer Berlin Heidelberg, 2005. [29]
- [41] L. DE FLORIANI et E. PUPPO : Hierarchical triangulation for multiresolution surface description. *ACM Transactions on Graphics*, 14(4):363–411, oct. 1995. [38]
- [42] L. DE FLORIANI, E. PUPPO et M. PAOLA : A formal approach to multiresolution hypersurface modeling. In R. R. KLEIN, W. STRASSER et R. RAU, édés : *Geometric Modeling: Theory and Practice*, Focus on Computer Graphics, p. 302–323. Springer Berlin Heidelberg, 1997. [38]
- [43] D. P. DOBKIN et M. J. LASZLO : Primitives for the manipulation of three-dimensional subdivisions. In *Proceedings of 3rd ACM Symposium on Computational Geometry*, p. 86–99, New York, NY, USA, 1987. ACM. [27]
- [44] D. DOO et M. SABIN : Analysis of the behaviour of recursive subdivision surfaces near extraordinary points. *Computer-Aided Design*, 10(6):356–360, nov. 1978. [65]
- [45] N. DYN, D. LEVINE et J. A. GREGORY : A butterfly subdivision scheme for surface interpolation with tension control. *ACM Transactions on Graphics*, 9(2):160–169, avr. 1990. [65]
- [46] H. EDELSBRUNNER : *Algorithms in Combinatorial Geometry*. European Association for Theoretical Computer Science: EATCS monographs on theoretical computer science. Springer-Verlag, Inc., New York, NY, USA, 1987. [16]
- [47] J. R. EDMONDS : A combinatorial representation for polyhedral surfaces. *Notices American Mathematical Society*, 7, 1960. [2]
- [48] J. EL-SANA et A. VARSHNEY : Generalized view-dependent simplification. *Computer Graphics Forum*, 18(3):83–94, déc. 1999. [38]
- [49] X. FENG, Y. WANG, Y. WENG et Y. TONG : Compact combinatorial maps: A volume mesh data structure. *Graphical Models*, 75(3):149–156, mai 2012. [27]
- [50] S. FOUREY et L. BRUN : Connecting walks and connecting dart sequences for n-D combinatorial pyramids. In P. WIEDERHOLD et R. P. BARNEVA, édés : *Progress in Combinatorial Image Analysis (Internatio-*

- nal Workshop on Combinatorial Image Analysis*), p. 109–122, Cancun, Mexico, nov. 2009. Research Publishing Services. [39]
- [51] S. FOUREY et L. BRUN : A first step toward combinatorial pyramids in  $n$ -d spaces. In A. TORSELLO, F. ESCOLANO et L. BRUN, édés : *Graph-Based Representations in Pattern Recognition*, vol. 5534 de *Lecture Notes in Computer Science*, p. 304–313. Springer Berlin Heidelberg, 2009. [39]
- [52] S. FOUREY et L. BRUN : Efficient encoding of  $n$ -D combinatorial pyramids. In *Proceedings of the International Conference on Pattern Recognition (ICPR'2010)*, Istanbul, Turkey, août 2010. [39]
- [53] R. B. FULLER : United States Patent Office #2,986,241 OCTET TRUSS, 1961. [11]
- [54] J. GAIN et D. BECHMANN : A survey of spatial deformation from a user-centered perspective. *ACM Transactions on Graphics*, 27(4):1–21, nov. 2008. [68]
- [55] P.-M. GANDOIN et O. DEVILLERS : Progressive lossless compression of arbitrary simplicial complexes. *ACM Transactions on Graphics*, 21(3):372–379, juil. 2002. [29]
- [56] C. GRASSET-SIMON, G. DAMIAND et P. LIENHARDT : Pyramids of  $n$ -dimensional generalized maps. In L. BRUN et M. VENTO, édés : *Graph-Based Representations in Pattern Recognition*, vol. 3434 de *Lecture Notes in Computer Science*, p. 142–152. Springer Berlin Heidelberg, 2005. [40]
- [57] C. GRASSET-SIMON, G. DAMIAND et P. LIENHARDT : Receptive fields for generalized map pyramids: The notion of generalized orbit. In E. ANDRES, G. DAMIAND et P. LIENHARDT, édés : *Discrete Geometry for Computer Imagery*, vol. 3429 de *Lecture Notes in Computer Science*, p. 56–67. Springer Berlin Heidelberg, 2005. [40]
- [58] C. GRASSET-SIMON, G. DAMIAND et P. LIENHARDT :  $n$ D generalized map pyramids: Definition, representations and basic operations. *Pattern Recognition*, 39(4):527–538, 2006. [40]
- [59] B. GREGORSKI, M. DUCHAINEAU, P. LINDSTROM, V. PASCUCCI et K. I. JOY : Interactive view-dependent rendering of large isosurfaces. In *Proceedings of the IEEE conference on Visualization*, p. 475–484, Washington, DC, USA, 2002. IEEE Computer Society. [36]
- [60] G. GREINER et R. GROSSO : Hierarchical tetrahedral-octahedral subdivision for volume visualization. *The Visual Computer*, 16(6):357–369, oct. 2000. [36]

- [61] R. GROSSO et G. GREINER : Hierarchical meshes for volume data. *In Computer Graphics International*, p. 761–769. IEEE Computer Society, 1998. [36]
- [62] H. HOPPE : Progressive meshes. *In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, p. 99–108, New York, NY, USA, 1996. ACM. [29]
- [63] T. J. HUGHES, J. A. COTTRELL et Y. BAZILEVS : Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39–41):4135–4195, 2005. [69]
- [64] J. INOUE et A. J. STEWART : Multiresolution sphere packing tree: a hierarchical multiresolution 3d data structure. *In Proceedings of the ACM symposium on Solid and physical modeling*, p. 367–373, New York, NY, USA, 2008. ACM. [32]
- [65] T. JUND, P. KRAEMER et D. CAZIER : A unified structure for crowd simulation. *Computer Animation and Virtual Worlds*, 23(3):311–320, déc. 2012. [28]
- [66] M. KIM, A. ENTEZARI et J. PETERS : Box spline reconstruction on the face-centered cubic lattice. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1523–1530, nov. 2008. [11]
- [67] L. KOBBELT :  $\sqrt{3}$ -subdivision. *In Proceedings of the 27th annual conference on computer graphics and interactive techniques*, SIGGRAPH '00, p. 103–112, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. [65, 120, 121]
- [68] P. KRAEMER : *Modèles topologiques pour la multirésolution*. Thèse de doctorat, Université Louis Pasteur - Strasbourg I, nov. 2008. [3, 43, 45, 118, 125, 152]
- [69] P. KRAEMER, D. CAZIER et D. BECHMANN : Extension of half-edges for the representation of multiresolution subdivision surfaces. *The Visual Computer*, 25(2):149–163, fév. 2009. [41]
- [70] P. KRAEMER, L. UNTEREINER, T. JUND, S. THERY et D. CAZIER : CGoGN: N-dimensional meshes with combinatorial maps. *In J. SARRATE et M. STATEN, édés : Proceedings of the 22nd International Meshing Roundtable*, p. 485–503. Springer International Publishing, 2014. [126, 166]

- [71] M. KREMER, D. BOMMES et L. KOBBELT : Openvolumemesh – a versatile index-based data structure for 3d polytopal complexes. *In* X. JIAO et J.-C. WEILL, édés : *Proceedings of the 21st International Meshing Roundtable*, p. 531–548. Springer Berlin Heidelberg, 2013. [27, 127]
- [72] J.-C. KRENCKER : *Développement d'outils et de modèles CAO de haut niveau pour la simulation électrothermique de circuits mixtes en technologie 3D*. Thèse de doctorat, Université de Strasbourg, nov. 2012. [107]
- [73] M. LAGE, T. LEWINER, H. LOPES et L. VELHO : CHF : A scalable topological data structure for tetrahedral meshes. *In 18th Brazilian Symposium on Computer Graphics and Image Processing*, p. 349–356. IEEE, oct. 2005. [27]
- [74] C. L. LAWSON : Properties of n-dimensional triangulations. *Computer Aided Geometric Design*, 3(4):231–246, 1986. [109]
- [75] B. LEHNER, B. HAMANN et G. UMLAUF : Generalized swap operation for tetrahedrizations. *In to be published*, 2010. [117]
- [76] E. LEITNER et S. SELBERHERR : Mixed-element decomposition method for three-dimensional grid adaptation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(7):561–572, juil. 1998. [11]
- [77] B. LÉVY, G. CAUMON, S. CONREAUX et X. CAVIN : Circular incident edge lists: a data structure for rendering complex unstructured grids. *In Proceedings of the conference on Visualization '01*, p. 191–198, Washington, DC, USA, 2001. IEEE Computer Society. [27]
- [78] P. LIENHARDT : Subdivisions of n-dimensional spaces and n-dimensional generalized maps. *In Proceedings of the 5th annual symposium on Computational geometry*, p. 228–236, New York, NY, USA, 1989. ACM. [3, 19]
- [79] P. LIENHARDT : Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82, jan. 1991. [3, 19]
- [80] P. LIENHARDT : N-dimensional generalized combinatorial maps and cellular quasi-manifolds. *International Journal of Computational Geometry and Applications*, 4(3):275–324, sept. 1994. [19]
- [81] L. LINSEN, B. HAMANN et K. I. JOY : Wavelets for adaptively refined  $\sqrt[3]{2}$ -subdivision meshes. *International Journal of Computers and Applications*, 29(3):223–231, juin 2007. [70]

- [82] L. LINSEN, V. PASCUCCHI, M. A. DUCHAINEAU, B. HAMANN et K. I. JOY : Hierarchical representation of time-varying volume data with  $\sqrt[4]{2}$  subdivision and quadrilinear b-spline wavelets. *In Proceedings of Tenth Pacific Conference on Computer Graphics and Applications*, Pacific Graphics 2002, p. 346–356, Washington, DC, USA, 2002. IEEE Computer Society Press. [71]
- [83] C. LOOP : Smooth subdivision surfaces based on triangles. Department of mathematics, University of Utah, Utah, USA, août 1987. [65]
- [84] H. LOPES et G. TAVARES : Structural operators for modeling 3-manifolds. *In Proceedings of the 4th ACM symposium on Solid Modeling and Applications*, p. 10–18, New York, NY, USA, 1997. ACM. [27]
- [85] F. LOSASSO, F. GIBOU et R. FEDKIW : Simulating water and smoke with an octree data structure. *ACM Transactions on Graphics*, 23(3):457–462, août 2004. [28]
- [86] R. MACCRACKEN et K. I. JOY : Free-form deformations with lattices of arbitrary topology. *In Proceedings of the 23rd annual conference on computer graphics and interactive techniques*, SIGGRAPH '96, p. 181–188, New York, NY, USA, 1996. ACM. [3, 65, 77, 98, 122]
- [87] S. MALLAT : *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3 éd., déc. 2008. [70]
- [88] K. T. MCDONNELL, Y.-S. CHANG et H. QIN : Interpolatory, solid subdivision of unstructured hexahedral meshes. *The Visual Computer*, 20(6):418–436, août 2004. [14, 65]
- [89] K. T. MCDONNELL, Y.-S. CHANG et H. QIN : DigitalSculpture: a subdivision-based approach to interactive implicit surface modeling. *Graphical Models*, 67(4):347–369, juil. 2005. [68]
- [90] K. T. MCDONNELL, N. NEOPHYTOU, K. MUELLER et H. QIN : Subdivision volume splatting. *In Proceedings of the 9th Joint Eurographics / IEEE VGTC conference on Visualization*, EUROVIS'07, p. 139–146, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. [69]
- [91] K. T. MCDONNELL et H. QIN : Fem-based subdivision solids for dynamic and haptic interaction. *In Proceedings of the sixth ACM symposium on Solid modeling and applications*, SMA '01, p. 312–313, New York, NY, USA, 2001. ACM. [68]

- 
- [92] K. T. MCDONNELL et H. QIN : Dynamic sculpting and animation of free-form subdivision solids. *The Visual Computer*, 18(2):81–96, mars 2002. [68]
- [93] K. T. MCDONNELL et H. QIN : A novel framework for physically based sculpting and animation of free-form solids. *The Visual Computer*, 23(4): 285–296, mars 2007. [68]
- [94] MOKA : Moka - 3D Topological Modeler. <http://moka-modeller.sourceforge.net/>. [27, 125]
- [95] M. J. MUUSS et L. A. BUTLER : Combinatorial solid geometry, boundary representations, and n-manifold geometry. In D. F. ROGERS et R. EARNSHAW, édés : *State of the Art in Computer Graphics: Visualization and Modeling*, p. 185–223. Springer-Verlag, New York, NY, USA, 1991. [14]
- [96] Y. ONO, B.-Y. CHEN, T. NISHITA et J. FENG : Free-form deformation with automatically generated multiresolution lattices. In *Proceedings of the First International Symposium on Cyber Worlds, CW '02*, p. 472–479, Washington, DC, USA, 2002. IEEE Computer Society. [70]
- [97] R. PAJAROLA et J. ROSSIGNAC : Compressed progressive meshes. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):79–93, jan. 2000. [29, 30]
- [98] V. PASCUCCI : Slow growing subdivision (SGS) in any dimension: Towards removing the curse of dimensionality. *Computer Graphics Forum*, 21(3):451–460, mai 2002. [66]
- [99] V. PASCUCCI : Slow growing volumetric subdivision. In *ACM SIGGRAPH 2002 conference abstracts and applications*, SIGGRAPH '2002, p. 251–251, New York, NY, USA, 2002. ACM. [66, 167]
- [100] S. POPINET : Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics*, 190(2):572–600, 2003. [28]
- [101] J. POPOVIĆ et H. HOPPE : Progressive simplicial complexes. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, p. 217–224, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. [29]
- [102] K. PRUESS et J. GARCIA : A systematic approach to local grid refinement in geothermal reservoir simulation. In *Proceedings of the World*

- Geothermal Congress 2000, Kyushu-Tohoku, Japan, May, 28*, p. 2809–2814. Lawrence Berkeley National Laboratory, 2000. [28]
- [103] E. PUPPO : Variable resolution triangulations. *Computational Geometry*, 11(3–4):219–238, déc. 1998. [38]
- [104] F. QIU : *Lattice Volume Rendering*. Thèse de doctorat, State University of New York at Stony Brook, New York, NY, USA, déc. 2008. [32, 169]
- [105] H. SAMET : *The Design and Analysis of Spatial Data Structures*, vol. 50255 de *Addison-Wesley Series in Computer Science*. Addison-Wesley Publishing Company, 1990. [30]
- [106] S. SCHAEFER, J. P. HAKENBERG et J. WARREN : Smooth subdivision of tetrahedral meshes. *In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on geometry processing, SGP '04*, p. 147–154, New York, NY, USA, 2004. ACM. [13, 65]
- [107] T. W. SEDERBERG et S. R. PARRY : Free-form deformation of solid geometric models. *Computer Graphics (SIGGRAPH '86)*, 20(4):151–160, août 1986. [68]
- [108] J. R. SHEWCHUK : Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. *In Unpublished manuscript*, 2002. [117]
- [109] D. SIEGER et M. BOTSCH : Design, implementation, and evaluation of the surface\_mesh data structure. *In W. R. QUADROS, éd. : Proceedings of the 20th International Meshing Roundtable*, p. 533–550. Springer Berlin Heidelberg, 2012. [26]
- [110] O. G. STAADT et M. H. GROSS : Progressive tetrahedralizations. *In Proceedings of the conference on Visualization '98*, p. 397–402, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press. [29, 30]
- [111] W. SWELDENS : The lifting scheme: a construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29(2):511–546, mars 1998. [70, 96]
- [112] J. TERAN, N. MOLINO, R. FEDKIW et R. BRIDSON : Adaptive physics based tetrahedral mesh generation using level sets. *Engineering with Computers*, 21(1):2–18, nov. 2005. [32]
- [113] I. J. TROTTS, B. HAMANN et K. I. JOY : Simplification of tetrahedral meshes with error bounds. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):224–237, 1999. [29]

- 
- [114] A. VINCE : Combinatorial maps. *Journal of Combinatorial Theory, Series B*, 34(1):1–21, fév. 1983. [2]
- [115] K. J. WEILER : Edge-based data structures for solid modeling in curved-surface environments. *IEEE Computer Graphics & Applications*, 5(1): 21–40, jan. 1985. [26]
- [116] K. J. WEILER : *Topological structures for geometric modeling*. Thèse de doctorat, Rensselaer Polytechnic Institute, Troy, NY, USA, 1986. [14]
- [117] K. WEISS et L. DE FLORIANI : Simplex and diamond hierarchies: Models and applications. *Computer Graphics Forum*, 30(8):2127–2155, 2011. [33, 35]
- [118] K.-Y. WHANG, J.-W. SONG, J.-W. CHANG, J.-Y. KIM, W.-S. CHO, C.-M. PARK et I.-Y. SONG : Octree-r : An adaptive octree for efficient ray tracing. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):343–349, déc. 1995. [30]
- [119] D. ZORIN : Modeling with multiresolution subdivision surfaces. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, p. 30–50, New York, NY, USA, 2006. ACM. [70]
- [120] D. ZORIN et P. SCHRÖDER : A unified framework for primal/dual quadrilateral subdivision schemes. *Computer Aided Geometric Design*, 18(5):429–454, juin 2001. [65]
- [121] D. ZORIN, P. SCHRÖDER, T. DEROSE, L. KOBELT, A. LEVIN et W. SWELDENS : Subdivision for modeling and animation. In *ACM SIGGRAPH 2000 Courses*, SIGGRAPH '00, New York, NY, USA, 2000. ACM. [70]





## Résumé

### Représentation des maillages multirésolutions : application aux volumes de subdivision

Les maillages volumiques sont très répandus en informatique graphique, en visualisation scientifique et en calcul numérique. Des opérations de subdivision, de simplification ou de remaillage sont parfois utilisées afin d'accélérer les traitements sur ces maillages. Afin de maîtriser la complexité de l'objet et des traitements numériques qui lui sont appliqués, une solution consiste alors à le représenter à différentes échelles. Les modèles existants sont conçus pour des approches spécifiques rendant leur utilisation limitée aux applications pour lesquelles ils ont été pensés.

Nos travaux de recherche présentent un nouveau modèle pour la représentation de maillages multirésolutions en dimension quelconque basé sur le formalisme des cartes combinatoires. Nous avons d'abord appliqué notre modèle aux volumes de subdivision multirésolutions. Dans ce cadre, nous présentons plusieurs algorithmes de raffinement d'un maillage grossier initial. Ces algorithmes supportent des hiérarchies obtenues par subdivision régulière et adaptative. Nous proposons ensuite deux représentations, opposés en terme de coût spatial et temporel, pour ce modèle.

**Mots clés :** modélisation géométrique, représentations multirésolution, modèles topologiques, cartes combinatoires, volumes de subdivision

## Abstract

### Representation of multiresolution meshes : an application to subdivision volumes

Volume meshes are widespread in computer graphics, scientific visualization and numerical computation. Subdivision, simplification or remeshing operations are sometimes used to speed up processing of these meshes. A solution to manage the complexity of the object and numerical processing applied to it consist in presenting this object at different scales. Nevertheless, existing models are designed for specific approaches making them limited to applications for which they were designed.

Our research work present a new model for the representation of multiresolution meshes in any dimension based on the combinatorial maps model. We first applied our model to the multiresolution subdivision volumes. In this framework, we present several refinement algorithms of an initial coarse mesh. These algorithms support hierarchies obtained by regular and adaptive subdivision. Finally, we propose two representations, opposed in term of time and space complexity, of this model.

**Keywords:** geometric modeling, multiresolution representations, topological models, combinatorial maps, subdivision volumes