



HAL
open science

Définition d'un modèle unifié pour la simulation physique adaptative avec changements topologiques

Elsa Fléchon Fléchon

► **To cite this version:**

Elsa Fléchon Fléchon. Définition d'un modèle unifié pour la simulation physique adaptative avec changements topologiques. Autre [cs.OH]. Université Claude Bernard - Lyon I, 2014. Français. NNT : 2014LYO10284 . tel-01107197v2

HAL Id: tel-01107197

<https://theses.hal.science/tel-01107197v2>

Submitted on 9 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ CLAUDE BERNARD LYON 1

N° attribué par la bibliothèque :
284-2014



THÈSE

pour obtenir le grade de **Docteur de l'UCBL**

Spécialité : **Informatique**

préparée au **Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS)**

dans le cadre de l'École Doctorale **InfoMaths**

présentée et soutenue publiquement par

Elsa FLÉCHON

le 09-12-14

Titre:

Définition d'un modèle unifié pour la simulation physique adaptative avec changements topologiques

*sous la direction de Fabrice JAILLET et de Florence ZARA
avec la collaboration de Guillaume DAMIAND*

Jury :

M.	David CAZIER, PU - Université de Strasbourg	Rapporteur
M.	Philippe MESEURE, PU - Université de Poitiers	Rapporteur
M.	Stéphane COTIN, DR HDR - INRIA Lille	Examineur
M.	Benjamin GILLES, CR - UMR CNRS 5506 Montpellier	Examineur
M.	Guillaume DAMIAND, CR HDR - UMR CNRS 5205 Villeurbanne	Invité
M.	Emmanuel PROMAYON, MCU HDR - Université Grenoble 1	Invité
M.	Fabrice JAILLET, MCU HDR - Université Lyon 1	Directeur doctoral
Mme.	Florence ZARA, MCU - Université Lyon 1	Co-Directrice doctorale

UNIVERSITÉ CLAUDE BERNARD - LYON 1

Président de l'Université

Vice-président du Conseil d'Administration
Vice-président du Conseil des Études et de la Vie Universitaire
Vice-président du Conseil Scientifique
Directeur Général des Services

M. François-Noël GILLY

M. le Professeur Hamda BEN HADID
M. le Professeur Philippe LALLE

M. le Professeur Germain GILLET
M. Alain HELLEU

COMPOSANTES SANTE

Faculté de Médecine Lyon Est - Claude Bernard
Faculté de Médecine et de Maïeutique Lyon Sud - Charles Mérieux
Faculté d'Odontologie
Institut des Sciences Pharmaceutiques et Biologiques
Institut des Sciences et Techniques de la Réadaptation
Département de formation et Centre de Recherche en Biologie Humaine

Directeur : M. le Professeur J. ETIENNE
Directeur : Mme la Professeure C. BURILLON

Directeur : M. le Professeur D. BOURGEOIS
Directeur : Mme la Professeure C. VINCIGUERRA
Directeur : M. le Professeur Y. MATILLON
Directeur : Mme. la Professeure A-M. SCHOTT

COMPOSANTES ET DÉPARTEMENTS DE SCIENCES ET TECHNOLOGIE

Faculté des Sciences et Technologies
Département Biologie
Département Chimie Biochimie
Département GEP
Département Informatique
Département Mathématiques
Département Mécanique
Département Physique
UFR Sciences et Techniques des Activités Physiques et Sportives
Observatoire des Sciences de l'Univers de Lyon
Polytech Lyon
École Supérieure de Chimie Physique Électronique
Institut Universitaire de Technologie de Lyon 1
École Supérieure du Professorat et de l'Éducation
Institut de Science Financière et d'Assurances

Directeur : M. F. DE MARCHI
Directeur : M. le Professeur F. FLEURY
Directeur : Mme Caroline FELIX
Directeur : M. Hassan HAMMOURI
Directeur : M. le Professeur S. AKKOUCHE
Directeur : M. le Professeur Georges TOMANOV
Directeur : M. le Professeur H. BEN HADID
Directeur : M. Jean-Claude PLENET
Directeur : M. Y. VANPOULLE

Directeur : M. B. GUIDERDONI
Directeur : M. P. FOURNIER
Directeur : M. G. PIGNAULT
Directeur : M. le Professeur C. VITON
Directeur : M. le Professeur A. MOUGNIOTTE
Directeur : M. N. LEBOISNE

Remerciements

Je tiens tout d'abord à remercier l'ensemble des membres de mon jury, et plus particulièrement mes rapporteurs, pour tout l'intérêt qu'ils ont porté à mes travaux.

Je poursuis avec le remerciement de mes encadrants officiels. Mon directeur de doctorat, Fabrice Jaillet, qui a porté administrativement mon doctorat. Ma co-directrice de doctorat, Florence Zara, que je remercie pour m'avoir fortement encouragée à accepter ce sujet de thèse passionnant ainsi que pour toute la sympathie qu'elle a eu à mon égard. Je les remercie également pour m'avoir laissée une très grande autonomie dans mon travail de recherche, mais aussi pour avoir permis la création du projet transversal TopoSim. Ce projet a été un grand tournant dans mon doctorat, en me donnant la chance de travailler avec le « grand ambassadeur des cartes combinatoires », Guillaume Damiant. Cette page de remerciement est enfin l'occasion de lui exprimer toute ma reconnaissance. Je tiens à le remercier pour toute son expertise, pour m'avoir initiée aux mystères des cartes combinatoires, pour nos envolées algorithmiques, pour ses conseils avisés, pour son immense soutien et pour sa grande gentillesse. Sans lui, cette thèse était loin d'être trop « fas » :p.

Je remercie pour leur gentillesse, tous les membres du LIRIS que j'ai cotoyés pendant ces trois dernières années, mes co-bureaux et les personnes avec lesquelles j'ai sympathisé notamment lors de conférences. Je tiens à remercier, plus particulièrement, les personnes qui m'ont accompagnée pendant mon doctorat. Tout d'abord, Grégoire, mon « binôme », avec qui j'ai effectué bras dessus, bras dessous, une bonne partie de ma scolarité dans le supérieur. Je remercie Aurélien pour m'avoir convaincue de faire un doctorat ! XD. Je remercie également Camille pour avoir été un mentor dans la réalisation de mon doctorat. Merci à Jean-David, pour tous nos debriefing de fin de journée ;) . Je remercie également Jérémy et David, pour toutes nos sorties course à pied, pour les utepils et pour être au bout du tchat :p. Enfin, je remercie mes étudiants, j'ai pris un réel plaisir à leur transmettre ma passion pour la programmation. L'enseignement fut pour moi une belle expérience.

Je termine par le remerciement de mes proches. Je remercie ma maman, pour son soutien en m'ayant prodigué tout au long de ma scolarité « le bisou de la chance », mon papa, pour tous ses conseils visant à forger mon mental, et ma soeurette adorée, pour toute l'attention qu'elle me porte. Je vous remercie pour tout votre soutien et votre amour. Pour le bonheur qu'il m'a apporté ces dix dernières années, de tout mon coeur, merci mon chéri. Je remercie également ma belle-famille pour leur immense gentillesse à mon égard. Enfin, merci à tous mes amis notamment les TS, ainsi que ceux que j'ai enrôlé dans Clash pour combattre à mes côtés :p.



Table des matières

1	Introduction	5
1.1	La simulation physique	6
1.2	L'informatique pour le médical	8
1.2.1	La chirurgie mini-invasive	8
1.2.2	Les simulateurs médicaux	9
1.3	La simulation du comportement des objets déformables	10
1.4	Verrous scientifiques	11
1.5	Organisation du manuscrit	12
1.6	Publications	13
2	Pré-requis fondés sur la littérature	15
2.1	Objets déformables	16
2.1.1	Détermination des paramètres rhéologiques des matériaux	17
2.1.2	Génération de maillages	18
2.1.3	La Mécanique des Milieux Continus	20
2.1.4	Les schémas d'intégration	28
2.2	Modèles pour les objets déformables	29
2.2.1	Modèles non-physiques ou géométriques	30
2.2.2	Modèles physiques	30
2.2.3	Conclusion sur la simulation physique d'objets déformables.	45
2.3	Modèle topologique	45
2.3.1	Quelques notions de topologie	47
2.3.2	Structure de données topologiques pour les maillages surfaciques	50
2.3.3	Structure de données nD	55
2.3.4	Travaux relatifs aux structures de données topologiques	70
2.3.5	Conclusion sur les structures de données topologiques.	76
2.4	Changements topologiques	77
2.4.1	Multi-résolution	78
2.4.2	Découpe	81
2.5	Conclusion	83
3	Modèle unifié alliant topologie et physique	85
3.1	Description du modèle unifié	86
3.2	Initialisation du modèle unifié	88
3.2.1	Masses	91
3.2.2	Ressorts	93

3.3	Simulation physique	96
3.4	Changement topologique de la découpe	97
3.4.1	La partie topologique du processus de découpe	97
3.4.2	La partie physique du processus de découpe	100
3.5	Conclusion	103
4	Raffinement local	105
4.1	Notions et concepts de base	107
4.1.1	Niveau de subdivision	107
4.1.2	Éléments hiérarchiques	110
4.1.3	Niveau de subdivision des éléments hiérarchiques	113
4.1.4	Brin hiérarchique	114
4.1.5	Jonction T	115
4.1.6	Éléments non conformes	116
4.2	Algorithmes de base	117
4.2.1	Récupérer le prochain brin hiérarchique	117
4.2.2	Tester si une face est simple	117
4.2.3	Trouver le prochain brin hiérarchique d'une face	118
4.2.4	Calcul du niveau de subdivision d'un élément	120
4.2.5	Trouver un brin hiérarchique d'un élément	123
4.3	Subdivision régulière d'un quadrilatère en quatre sous-quadrilatères	124
4.3.1	Présentation intuitive	124
4.3.2	Subdivision d'un quadrilatère inscrit dans un modèle localement raffiné	125
4.3.3	Initialisation et mise à jour des informations physiques	128
4.4	Subdivision régulière d'un hexaèdre en huit sous-hexaèdres	134
4.4.1	Présentation intuitive	134
4.4.2	Subdivision d'un hexaèdre inscrit dans un modèle localement raffiné	138
4.4.3	Initialisation et mise à jour des informations physiques	145
4.5	Découpe d'un objet 2D localement raffiné	161
4.5.1	Mise à jour des extrémités de projection	163
4.5.2	Enlever la propriété de jonction T à un sommet	164
4.5.3	Supprimer une jonction T	164
4.6	Découpe d'un objet 3D localement raffiné	165
4.6.1	Mise à jour des extrémités de projection	166
4.6.2	Enlever la propriété de jonction T à un sommet	169
4.6.3	Supprimer une jonction T	170
4.7	Conclusion	171
5	Résultats préliminaires	173
5.1	Découpe	173
5.2	Découpe et raffinement	176
5.3	Méthode adaptative	177
5.4	Conclusion	183
6	Conclusion	187
7	Annexes	189
7.1	Annexe A	189
7.2	Annexe B	191

Bibliographie

191

Table des figures

1.1	Des exemples d'application de la simulation numérique.	7
1.2	Simulateur de la chirurgie de la cataracte [CADC11].	8
1.3	Les robots chirurgicaux.	9
1.4	Les applications de la simulation du comportement des organes.	10
2.1	Tests mécaniques permettant la détermination des propriétés mécaniques d'un matériau.	17
2.2	Génération de maillages à partir d'images médicales.	18
2.3	Génération d'un maillage volumique à partir d'une surface polygonale.	19
2.4	La relation entre la déformation et la contrainte.	22
2.5	Le tenseur des contraintes.	23
2.6	Test d'étirement permettant la caractérisation des paramètres rhéologiques du module de Young et du coefficient de Poisson.	25
2.7	Exemple du modèle FFD.	30
2.8	Exemple du modèle FFD.	30
2.9	Un des résultats obtenus avec le modèle physique établi par Terzopoulos [TPBF87].	31
2.10	Illustration du comportement d'un modèle éléments finis linéaires, n'étant pas invariant en rotation.	32
2.11	Exemple de simulation de découpe du foie.	34
2.12	Exemple de SMR.	35
2.13	Exemple des premiers travaux employant un SMR pour l'animation faciale.	38
2.14	Exemple de l'utilisation d'un SMR pour la simulation de tissu.	39
2.15	Schéma explicitant la paramétrisation du SMR en fonction des paramètres physiques des matériaux.	40
2.16	Exemple de SMR.	41
2.17	Illustration de la méthode de Van Gelder.	42
2.18	Schémas expliquant la méthode de Baudet <i>et al.</i> [BBJ ⁺ 09b], permettant la paramétrisation d'un SMR en fonction des paramètres physiques des matériaux que sont le module de Young et le coefficient de Poisson.	43
2.19	Le modèle CSG étendu aux surfaces implicites.	45
2.20	Un exemple de surface de Bézier.	46
2.21	La notion de <i>i</i> -cellule et de relation d'incidence et d'adjacence.	47
2.22	La définition mathématique d'une variété 2D.	48
2.23	La définition pour un maillage d'une variété 2D par des contre-exemples.	48
2.24	La notion mathématique de quasi-variété pour un objet 3D.	49

2.25	La définition pour un maillage 3D, d'une quasi-variété par des contre-exemples.	49
2.26	Un exemple d'objet 2D non-orientable.	50
2.27	Un exemple de maillage surfacique.	51
2.28	Un exemple d'une structure de données fondée sur les faces.	51
2.29	Un exemple de graphe d'incidence.	52
2.30	Un exemple de la structure de données de l'arête ailée.	53
2.31	Un exemple de la structure de données des demi-arêtes.	54
2.32	La notion d'involution et de permutation pour la structure de données des demi-arêtes.	54
2.33	Les brins composant une face incidente à deux volumes.	56
2.34	Confrontation d'un objet 2D subdivisé en cellules et sa représentation avec une carte combinatoire.	59
2.35	Confrontation d'un objet 3D subdivisé en cellules et sa représentation avec une carte combinatoire.	59
2.36	Le parcours des brins incidents à un sommet (<i>resp.</i> arête).	60
2.37	Le parcours des brins incidents à un sommet (<i>resp.</i> arête, face) pour un modèle 3D.	61
2.38	Un exemple de 3-décousure.	63
2.39	Un exemple de l'opérateur d'insertion d'un sommet dans une arête incidente à deux quadrilatères.	63
2.40	Un exemple de l'opérateur d'insertion d'un sommet (<i>resp.</i> arête) dans une arête (<i>resp.</i> face) incidente à deux hexaèdres.	64
2.41	Un exemple de l'opérateur de suppression d'un sommet et d'une arête incidente à deux faces.	65
2.42	Un exemple de l'opérateur de suppression d'une face incidente à deux volumes et d'une arête incidente à deux faces.	66
2.43	Un exemple de l'opérateur de suppression d'une arête incidente à deux faces et d'un sommet incident à deux arêtes, dans le cadre d'un maillage 3D.	67
2.44	Un objet 2D représenté avec une carte combinatoire puis avec une carte généralisée.	68
2.45	L'impossibilité de représenter un objet 2D non-orientable avec une carte combinatoire.	69
2.46	La possibilité de représenter des objets 2D non-orientables avec une carte généralisée.	70
2.47	Exemple de la définition d'un nouveau modèle topologique.	70
2.48	Exemple de l'utilisation de la structure de données des demi-arêtes pour la simplification de maillage.	71
2.49	Exemple de l'utilisation d'une carte combinatoire 2D pour la segmentation d'images.	71
2.50	Exemple de l'utilisation de cartes combinatoires permettant la réalisation d'un modèle unifié de bâtiments pour la simulation acoustique et énergétique.	72
2.51	Exemple de l'utilisation de cartes généralisées permettant la simulation d'évolutions géomorphologiques.	73
2.52	Exemple de l'utilisation de cartes combinatoires pour la simulation de changements topologiques de type fracture, durant une animation.	74
2.53	Exemple de carte multi-résolution à deux niveaux consécutifs.	75
2.54	Exemple de maillage obtenu par une subdivision adaptative.	76
2.55	Résultats des travaux de Hutchinson <i>et al.</i> [HPH96] et de Villard <i>et al.</i> [VB05] permettant le raffinement local d'un maillage SMR 2D.	78

TABLE DES FIGURES

2.56	Résultats des travaux de Debunne <i>et al.</i> [DDCB01].	80
2.57	Illustration des différentes techniques permettant la découpe d'un maillage triangulaire. Cette illustration est présentée dans l'état de l'art de Wu <i>et al.</i> [WWD14].	81
2.58	Résultats et modèle issus des travaux de Dick <i>et al.</i> [DGW11] permettant la découpe virtuelle d'objets déformables.	82
2.59	Résultats des travaux de Steinemann <i>et al.</i> [SHGS06] permettant la découpe virtuelle d'objets déformables.	83
3.1	Le processus de construction de notre modèle allant des informations géométriques au modèle LCC+SMR.	86
3.2	Un quadrilatère et un hexaèdre représenté par notre modèle LCC+SMR.	89
3.3	Exemple d'un fichier stockant les informations géométriques et topologiques nécessaires à l'élaboration de notre modèle unifié et le maillage indexé correspondant.	90
3.4	Calcul de la masse associée à un sommet, pour le cas 2D.	91
3.5	Calcul de la masse associée à un sommet, pour le cas 3D.	92
3.6	Calcul de la raideur d'un ressort, pour le cas 2D.	93
3.7	Calcul de la raideur d'un ressort, dans le cas 3D.	95
3.8	Le processus de la simulation physique.	96
3.9	Un exemple du processus de découpe, pour le cas 2D.	98
3.10	Un exemple du processus de découpe, pour le cas 3D.	99
4.1	La notion de niveau de subdivision pour un modèle 2D.	108
4.2	La notion de niveau de subdivision pour un modèle 3D.	109
4.3	La notion d'arête hiérarchique pour un modèle 2D.	111
4.4	La notion d'arête et de face hiérarchiques pour un modèle 3D.	112
4.5	La définition de brin hiérarchique pour une face incidente à deux volumes n'ayant pas le même niveau de subdivision.	113
4.6	La notion de brin hiérarchique pour un modèle 2D.	115
4.7	La notion de jonction T bivalente.	117
4.8	Le parcours de la bordure de la face hiérarchique d'un volume étant plus subdivisée que ce dernier.	119
4.9	Le processus de subdivision régulière d'un quadrilatère en quatre sous-quadrilatères.	124
4.10	La subdivision d'un quadrilatère incident à deux hexaèdres, en quatre sous-quadrilatères.	126
4.11	La subdivision d'un quadrilatère adjacent à un quadrilatère ayant un niveau de subdivision plus grand.	126
4.12	La création d'une jonction T suite à la subdivision d'un quadrilatère qui avait initialement le même niveau de subdivision qu'un quadrilatère adjacent.	129
4.13	La mise à jour des masses et des ressorts, suite au processus de subdivision d'un quadrilatère.	130
4.14	Le comportement erroné induit par la création de jonctions T et la nécessité de contraindre la position de ses dernières, pour le cas 2D.	133
4.15	Le processus de subdivision d'un hexaèdre.	135
4.16	Le processus de subdivision d'un hexaèdre avec la représentation d'une partie des brins.	136
4.17	Le marquage hiérarchique d'une face simple.	137

4.18	Les premières étapes du processus de subdivision d'un volume, présentant des arêtes et des faces hiérarchiques subdivisées en amont par des volumes adjacents.	140
4.19	L'initialisation des extrémités de projection d'une jonction T nouvellement créée, pour le cas 3D.	148
4.20	L'initialisation des extrémités de projection d'une jonction T nouvellement créée dans une face hiérarchique qui a déjà un niveau de subdivision plus grand que son volume associé, pour le cas 3D.	149
4.21	La présentation de la notion de jonction T bivalente et mise à jour des extrémités de projection d'une jonction T bivalente, pour le cas 3D.	150
4.22	La modification de l'état de jonctions T relatives à la face hiérarchique ayant un niveau de subdivision plus grand que son volume associé en cours de subdivision, pour le cas 3D.	152
4.23	L'évolution de l'état de jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D.	154
4.24	La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D.	155
4.25	La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D.	156
4.26	La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D.	157
4.27	Le comportement erroné induit par la création de jonctions T et la nécessité de contraindre la position de ses dernières, pour le cas 3D.	158
4.28	La suppression de jonction T suite au processus de découpe, pour le cas 2D.	162
4.29	La nécessité de mettre à jour les extrémités de projection des jonctions T, pour le cas 2D.	163
4.30	Une 3-décousure avec la représentation d'une partie des brins.	165
4.31	La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D.	167
4.32	La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D.	168
4.33	La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D.	169
4.34	La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D.	170
5.1	Résultats illustrant le processus de découpe.	174
5.2	Résultats illustrant le processus de découpe interactif.	174
5.3	Résultats illustrant le processus de découpe.	174
5.4	Résultats illustrant le processus de découpe.	175
5.5	Résultats illustrant le processus de découpe.	175
5.6	Résultats illustrant le processus de découpe.	176
5.7	Résultats illustrant le processus de raffinement et de découpe, pour un maillage surfacique.	176
5.8	Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique.	177
5.9	Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique.	177

TABLE DES FIGURES

5.10	Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique.	178
5.11	Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique.	178
5.12	Résultats illustrant le processus de découpe et de raffinement, pour un maillage volumique représentant un lapin.	179
5.13	Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique.	181
5.14	Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique.	181
5.15	Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique.	182
5.16	Résultats illustrant le processus de subdivision adaptative, pour un maillage volumique.	183
5.17	Courbes de précision de notre méthode adaptative, pour un maillage surfacique.	184
5.18	Courbes de précision de notre méthode adaptative, pour un maillage volumique.	185

Liste des Algorithmes

1	Boucle de simulation	96
2	Mise à jour des ressorts diagonaux	101
3	Récupérer le prochain brin hiérarchique d'une face simple	118
4	Tester si une face est simple	118
5	Le prochain brin hiérarchique d'une face	119
6	Calculer le niveau de subdivision d'une face simple lorsque ses brins hiérarchiques sont correctement marquées	120
7	Calculer le niveau de subdivision d'une face simple lorsque ses brins hiérarchiques ne sont pas correctement marqués	121
8	Niveau de subdivision d'un élément	122
9	Obtenir un brin hiérarchique d'un élément	123
10	Récupérer le brin partant du sommet situé au milieu d'une arête hiérarchique donnée	128
11	Marquer hiérarchiquement les brins d'une face hiérarchique d'un volume en cours de subdivision, ayant un niveau de subdivision plus important que son volume	143
12	Récupérer le prochain brin d'une face hiérarchique en fonction d'un niveau de subdivision	144
13	Constituer un ensemble de brins afin d'insérer une face	144
14	Mettre à jour les extrémités de projection des jonctions T à partir d'une jonction T frontalière	160
15	Mettre à jour les extrémités de projection d'une jonction T	161
16	Mettre à jour les extrémités de projection des jonctions T centrées incluses dans une face hiérarchique	161
17	Scinder une arête en deux	189
18	Scinder une face en deux	190
19	Subdiviser une face en quatre sous-faces	190
20	Tester si une face contient un brin hiérarchique à un volume	191
21	Marquer hiérarchiquement les brins incidents à une face hiérarchique simple	191
22	Récupérer le prochain brin d'une face hiérarchique et simple en fonction d'un niveau de subdivision	192
23	Subdiviser une face en deux	192
24	Subdiviser un volume en deux (fonction intermédiaire)	192
25	Subdiviser un volume en deux	193

26	Subdiviser un volume en quatre	193
27	Subdiviser un volume en huit sous-volumes	194

Résumé

Ce manuscrit présente les travaux réalisés pendant mon doctorat que j'ai effectué au LIRIS (Laboratoire d'Informatique en Image et Systèmes d'information) au sein de l'équipe SAARA (Simulation, Analyse, Animation pour la Réalité Augmentée). Ces travaux s'inscrivent dans le domaine de la simulation physique. Nous nous sommes intéressés à l'élaboration d'un modèle unifié couplant un modèle topologique et un modèle physique, pour la simulation physique d'objets déformables, tout en réalisant pendant cette simulation des changements topologiques comme la découpe ou la subdivision d'éléments.

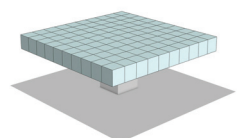
Nous avons fait le choix des cartes combinatoires et plus particulièrement celui des complexes cellulaires linéaires, comme modèle topologique de notre modèle unifié. Ils permettent de représenter des objets surfaciques comme volumiques, par une décomposition en cellules. Ils ont pour principal intérêt d'offrir un accès à l'ensemble des relations d'incidence et d'adjacence entre les cellules de l'objet modélisé, facilitant ainsi la gestion des changements topologiques. De plus, leur définition mathématique permet de garantir la validité topologique du modèle, au cours des changements topologiques.

Nous avons fait le choix d'un système masses-ressorts pour le modèle physique de notre modèle unifié. L'avantage de ce dernier réside dans la simplicité de ses équations, son implémentation intuitive, son interactivité et sa facilité à gérer les changements topologiques. La paramétrisation des coefficients de raideur de notre système masses-ressorts est permise par une méthode analytique proposée par Baudet *et al.*. Leur méthode permet la formulation des coefficients de raideur des ressorts en fonction des paramètres rhéologiques des matériaux que sont le module de Young et le coefficient de Poisson. Par contre, l'utilisation de leur méthode implique des modèles 2D composés de quadrilatères et des modèles 3D constitués d'hexaèdres.

Les modèles topologiques exclusivement 2D sont largement employés dans le domaine de la simulation physique. Ils permettent de stocker l'ensemble des informations géométriques et topologiques du modèle simulé, tout en permettant de faciliter l'accès à ces informations. C'est donc naturellement que nous nous sommes intéressés à l'emploi d'un modèle topologique plus complexe pour la simulation physique d'objets 3D. Enfin, la définition d'un modèle unifié évite la redondance d'informations et facilite la mise à jour de ces dernières.

Ces travaux montrent les bénéfices de l'emploi d'un modèle topologique pour la simulation physique, au travers de l'implémentation de changements topologiques comme la découpe en 2D, selon les arêtes, et en 3D, selon les faces du modèle, ainsi que la subdivision de quadrilatères ou d'hexaèdres. Nous détaillons dans ce manuscrit comment réaliser ces changements topologiques d'un point de vue topologique, puis d'un point de vue physique avec la mise à jour des informations embarquées au niveau du modèle topologique.

Enfin, l'opération de subdivision locale des quadrilatères ou des hexaèdres du modèle, que



nous avons implémentée, nous permet de proposer une méthode adaptative. Une méthode adaptative repose sur le constat que plus un maillage est fin, plus la simulation du modèle sera précise, mais en contrepartie elle sera plus gourmande en temps de calcul. Elle permet alors un compromis entre la précision et le temps de simulation, avec un maillage constitué d'éléments de différents niveaux de résolution. L'objectif étant de raffiner ou de dé-raffiner localement les éléments du maillage, en fonction d'un critère s'appuyant sur les besoins en précision de la simulation de manière locale. Notre méthode adaptative consiste à débiter avec un modèle contenant peu d'éléments, et à raffiner ces derniers, selon un critère géométrique.

Abstract

This manuscript presents the work made during my PhD in LIRIS laboratory (Laboratoire d'Informatique en Image et Systèmes d'information - Computer science in image and information system laboratory) within SAARA team (Simulation, Analysis, Animation for augmented reality). This work is in the field of physical simulation. We are interested in developing a unified model coupling a topological model and a physical model, for the physical simulation of deformable objects, while performing during this simulation, topological changes such as cutting or refinement.

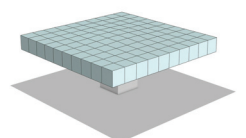
We made the choice of combinatorial maps and more particularly linear cell complexes for the topological model of our unified model. The main advantage is that they can represent surface as volume objects. The representation of these objects consists in a decomposition into cells. The advantage of linear cellular complex is that it provides an access to all the relationships of incidence and the adjacency between cells of the modeled object, facilitating the management of topology changes. Finally, their mathematical definition ensures the validity of the model when topological changes occur.

We choose a mass-spring system for the physical model of our unified model. The main advantage is the simplicity of its equations, its intuitive implementation, its interactivity and its ability to handle topological changes. Moreover, the parameterization of the stiffness coefficients of our mass-spring system is given by an analytical method proposed by Baudet *et al.*. Indeed, the formulation of stiffness coefficients of the springs based on rheological parameters of material, Young's modulus and Poisson's ratio. However, the use of their approach involves 2D models composed of quads and 3D models made of hexahedra.

The topological models valid only 2D are widely used in the field of physical simulation. They can store all geometric and topological information of the simulated model, while allowing an easy access to this information. So naturally we are interested in the use of a more complex topological model for the physical simulation of 3D objects. In addition, the definition of a unified model avoids redundancy of information and facilitates their updating.

This work shows the benefits of using a topological model for physical simulation, through the implementation of topological changes such as cutting in 2D, along the edges, and in 3D, along faces of the model, and also quads or hexahedra refinement. We explain in this manuscript how to perform these topological changes in a topological regard, then a physical regard with updating information embedded in the topological model.

Finally, the operation of local subdivision of quads or hexahedra we propose allows us to design an adaptive method. An adaptive method is based on the analysis that more mesh is finer, the simulation model will be more accurate, but in return it will be more time-cost calculation. So, it allows a trade-off between accuracy and simulation time-cost with a mesh composed of elements with different levels of resolution. The aim is to locally refined or



coarsed elements of the mesh according to a criterion based on the accuracy requirements of the simulation. Our adaptive method starts with a mesh that contains few elements, and refines them according to a geometric criterion.

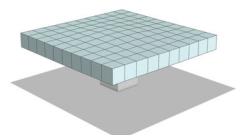
Chapitre 1

*« En une Dimension, un Point ne produirait-il pas en se mouvant une Ligne dotée de deux points terminaux ? En deux Dimensions, une ligne ne produirait-elle pas en se mouvant un Carré doté de quatre points terminaux. En Trois Dimensions, un Carré ne produirait-il pas en se mouvant - et ne m'a-t-il pas été donné à moi-même de le contempler ? - un Cube, cet être béni doté de huit points terminaux ? Et en quatre Dimensions, un Cube ne produirait-il pas en se mouvant - hélas pour l'Analogie, hélas pour le Progrès de la Vérité s'il n'en était pas ainsi - quelque Organisation encore plus divine dotée de seize points terminaux ? »
(Flatland - Edwin Abbott Abbott)*

Introduction

Sommaire

1.1	La simulation physique	6
1.2	L'informatique pour le médical	8
1.2.1	La chirurgie mini-invasive	8
1.2.2	Les simulateurs médicaux	9
1.3	La simulation du comportement des objets déformables	10
1.4	Verrous scientifiques	11
1.5	Organisation du manuscrit	12
1.6	Publications	13



La recherche scientifique a entamé un grand tournant avec le domaine de la simulation physique. La simulation physique devient incontournable pour de nombreuses problématiques scientifiques, en permettant la modélisation et la simulation de phénomènes complexes. Elle intervient en complément des approches théoriques et expérimentales. Les applications sont nombreuses et diverses comme dans le domaine du climat, des sciences de la vie, de l'astrophysique, de l'aéronautique, du spatial ou encore de l'énergie. La simulation physique est présente dans les cycles de conception et de production au niveau des industriels. Enfin, elle prend part à l'élaboration des politiques énergétiques, de prévention des risques naturels ou encore de la défense [sdLR12].

1.1 La simulation physique

La modélisation et la simulation numérique permettent de remplacer certaines expérimentations qui peuvent être difficiles à mettre en place en terme de coût et de temps. Leur objectif est de pouvoir reproduire un phénomène à des fins d'étude, d'analyse ou encore de conception d'un produit. En effet, la simulation numérique permet de reproduire le comportement d'un produit soumis à un phénomène complexe. Grâce à une modélisation virtuelle du produit en question, on peut ainsi analyser son comportement, le tester, déterminer ses limitations, quantifier les risques de son utilisation et enfin, l'optimiser. Cette étape, en amont de la conception, devient indispensable dans certaines applications. Afin d'illustrer ces propos, nous présentons ici quelques exemples de domaines faisant appel à la simulation numérique.

Dans ce sens, le logiciel SPH-flow réalisé en partenariat avec l'Ecole centrale de Nantes et la start-up HydrOcean (www.sph-flow.com), permet une simulation numérique des écoulements de fluides. Elle utilise la méthode SPH (Smoothed Particle Hydrodynamics) [Gé06] permettant de résoudre les équations de la mécanique des fluides de Navier-Stokes. Ce modèle consiste en une technique lagrangienne où un fluide est discrétisé en un ensemble fini de particules. Une particule est un élément discret régi par un ensemble de propriétés comme sa position ou sa masse et est dotée d'une zone d'influence décrivant son interaction avec ses particules voisines. Le logiciel SPH-flow couplé à l'utilisation de supercalculateur (parallélisation du calcul sur des milliers de processeur), rend possible une simulation précise d'écoulement de fluide comme par exemple la simulation d'un fluide en aquaplaning (*cf.* la figure 1.1(a)). La simulation numérique permet ainsi d'étudier le phénomène de l'aquaplaning et de proposer un pneumatique minimisant les risques causés par ce phénomène. Différents pneumatiques peuvent ainsi être modélisés, tout en faisant varier leurs propriétés physiques, leur structure, leurs constituants ou encore leur forme. Chaque type de pneumatique peut ainsi être simulé en amont de la conception afin d'être testé, analysé, amélioré, ou encore permettre la validation de leur conformité.

Nous pouvons également prendre l'exemple du projet ITER (International Thermonuclear Experimental Reactor). Ce projet a pour but la réalisation d'un réacteur expérimental de fusion nucléaire pour démontrer sa viabilité comme source d'énergie du futur. La fusion nucléaire consiste à fabriquer à partir de deux noyaux légers, un noyau atomique plus lourd. Cette réaction engendre une quantité d'énergie gigantesque qui se produit de manière naturelle dans les étoiles. Dans le cas de ITER, des atomes de deutérium et de tritium sont fusionnés pour obtenir des atomes d'hélium. Pour que cette fusion se produise, les particules doivent être chauffées à des températures extrêmes de l'ordre de la centaine de millions de degrés. A ces températures, la matière est sous une forme dite d'un plasma. Un **tokamak** (*cf.* la figure 1.1(b)) est une enceinte torique permettant de confiner un plasma à l'aide de champs magnétiques intenses créant les conditions nécessaires à la fusion thermodynamique.

Afin d'étudier le phénomène des pertes de chaleur dues à de minuscules tourbillons turbulents sur la bordure extérieure du plasma, des simulations ont été réalisées grâce à des supercalculateurs sur des maillages de 34 milliards de points correspondant en une discrétisation du plasma en particules. Cette simulation a permis de mettre en évidence des processus physiques extrêmement complexes et riches, de les comprendre, de proposer des solutions au problème de ces turbulences et même de découvrir de nouveaux phénomènes inattendus.

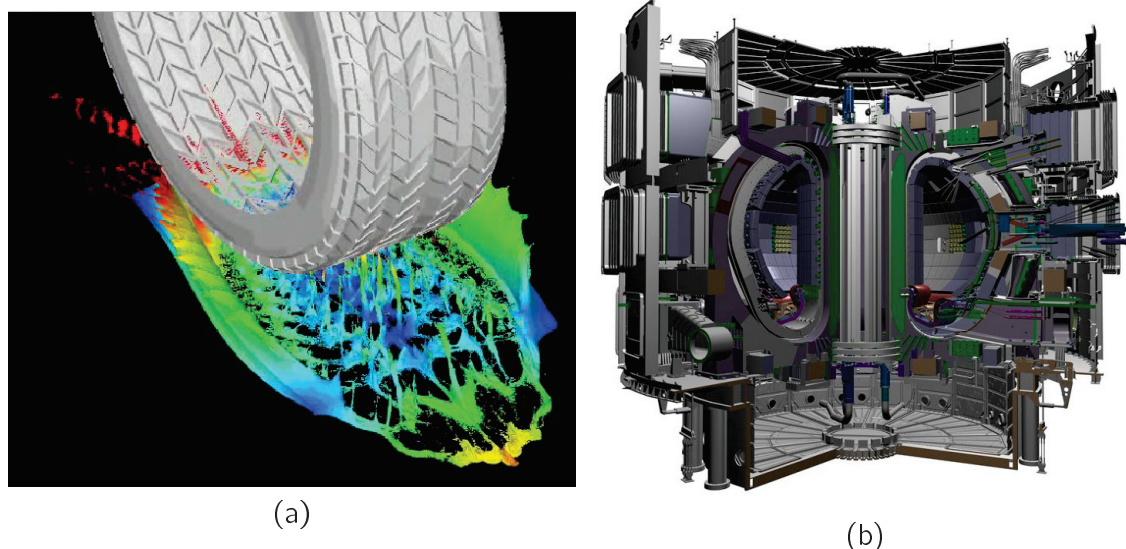
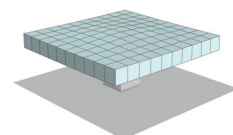


Figure 1.1 – (a) La simulation numérique d'un aquaplaning avec le logiciel SPH-flow (www.sph-flow.com). (b) Un tokamak (www.iter.org).

Ce ne sont que deux exemples parmi tant d'autres. La simulation numérique prend également une part de plus en plus importante dans le domaine médical. Nous pouvons citer l'exemple de la fondation *HelpMeSee* (www.helpmeseesee.org) ayant pour but de réunir les fonds nécessaires à la réalisation d'actes chirurgicaux pour soigner la cataracte dans les pays en développement. En effet, 22 millions d'enfants et d'adultes sont aujourd'hui aveugles à cause d'une pathologie de la cataracte non traitée. La cataracte se traduit par une opacification partielle ou totale du cristallin. Elle est la première cause de cécité dans les pays en voie de développement. Elle est favorisée par la dénutrition, la déshydratation, le tabagisme ou encore le diabète. Une chirurgie de 15 minutes pour un enfant et 5 minutes pour un adulte leur permettrait de recouvrir la vue. Malheureusement, le coût d'une telle opération ne leur permet pas d'en bénéficier. Fondée en 2010, *HelpMeSee* réalise à bas coût, des opérations chirurgicales de type MSIC (Manual Small Incision Cataract Surgery). MSIC est un nouveau protocole chirurgical s'opposant à l'approche conventionnelle de la phacoémulsification. Elle permet de réaliser rapidement (en 10 min) et pour seulement 50\$ une chirurgie de la cataracte. Elle consiste en l'extraction de la cataracte défectueuse pour la remplacer par un implant artificiel. Or, cette chirurgie nécessite une formation des chirurgiens. Ils sont actuellement que trop peu à être formés : 100 alors que 30000 seraient nécessaires. N'ayant ainsi pas assez de chirurgiens formés à cette technique, et afin de permettre aux apprenants de s'entraîner à cette technique hors patient, l'équipe SHACRA (INRIA Lille) et la société InSimo ont, dans ce sens, réalisé un simulateur pour l'apprentissage du protocole MSIC modélisant un œil virtuel. Le simulateur permet à l'apprenant de réaliser une chirurgie réaliste de la cataracte (cf. la figure 1.2). Enfin, il faut noter que 90% de la formation sera réalisée sur le simulateur avec un ordre de grandeur de 250 sessions d'entraînement.



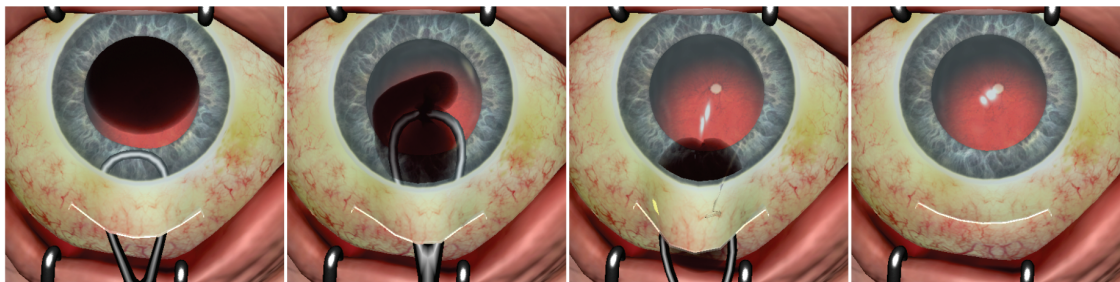


Figure 1.2 – Simulateur de la chirurgie de la cataracte [CADC11].

1.2 L'informatique pour le médical

L'évolution des nouvelles technologies et de l'informatique a permis de grandes avancées dans le domaine de la médecine. Dans ce sens, le domaine de la **surgétique** est un domaine de recherche récemment défini comme transdisciplinaire entre l'informatique, la robotique ainsi que l'imagerie pour la pratique médico-chirurgicale. Depuis 1985 (date des premières expériences dans le domaine de la surgétique [CDL+01]), l'évolution de l'informatique a permis l'avènement de nouvelles techniques chirurgicales basées ordinateur. L'assistance de l'ordinateur pour la chirurgie a pour but de préparer en amont la procédure d'une opération chirurgicale, de guider et d'effectuer des gestes chirurgicaux, de permettre la formation des futurs chirurgiens, de s'entraîner sur des cas rares et/ou difficiles, de valider de nouveaux protocoles chirurgicaux ou encore de concevoir de nouveaux instruments chirurgicaux.

L'amélioration de l'imagerie médicale avec l'IRM (Imagerie par Résonance Magnétique) et l'ensemble des travaux sur des algorithmes d'analyse et de traitement des images résultantes en est un premier exemple. L'IRM aide au diagnostic médical en produisant des vues 2D et maintenant 3D d'une partie du corps humain. En effet, elle permet d'étudier avec une grande précision les tissus mous comme le cerveau, les muscles, le cœur ainsi que les tumeurs de manière non invasive.

La chirurgie a débuté un grand tournant avec la chirurgie minimalement-invasive. Cette dernière est un deuxième exemple des avancées spectaculaires qui ont été réalisées en alliant les nouvelles technologies et l'informatique au médical.

1.2.1 La chirurgie mini-invasive

La chirurgie mini-invasive se définit par opposition à la chirurgie traditionnelle dite « à ciel ouvert ». Selon l'intervention chirurgicale à effectuer, la chirurgie traditionnelle nécessitant de larges incisions peut se soustraire à la chirurgie mini-invasive qui permet ainsi de limiter le traumatisme opératoire de cette dernière et une plus grande précision et efficacité des gestes réalisés. Cependant, la chirurgie mini-invasive, bien qu'elle soit actuellement largement employée, limite fortement la perception visuelle et tactile du chirurgien ou encore sa gestuelle par rapport à la chirurgie traditionnelle. Elle provoque de lourds changements dans les procédures opératoires en limitant l'espace opératoire et en nécessitant une coordination mains-yeux du chirurgien. L'objectif de la surgétique est d'améliorer les opérations minimalement invasives en redonnant au chirurgien toute sa dextérité et en augmentant ses capacités de perception, grâce aux avancés dans l'informatique, l'analyse et le traitement d'images ou encore de la robotique. Par exemple, grâce à la réalité augmentée, le chirurgien peut visualiser l'emplacement d'une tumeur pendant l'opération afin de mieux pouvoir la traiter.

Nous pouvons prendre comme premier exemple de l'impact de l'évolution de l'image-

rie médicale sur la médecine, celui de l'endoscopie. L'endoscopie est une technologie mini-invasive de diagnostic et d'intervention chirurgicale. Un endoscope est en général constitué d'un tube télescopique flexible avec à son extrémité une source de lumière et une caméra fixée sur un système optique reliée à un moniteur externe. Ce dernier permet de visualiser le parcours de l'endoscope à l'intérieur d'un conduit naturel ou d'une cavité par incision à l'aide de trocart. Dans le cas d'une intervention chirurgicale, l'endoscope peut également être muni d'un ou plusieurs canaux opérateurs pour l'emploi de différents types d'instruments permettant, en prenant l'exemple de la coloscopie, d'insuffler de l'air pour décoller les parois d'un conduit naturel, d'aspirer une zone à traiter pour pouvoir placer des ligateurs élastiques afin de réséquer des lésions coliques ou encore de mettre en place des clips en cas de perforation.

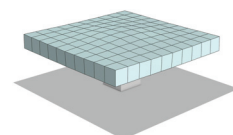
Les robots chirurgicaux (cf. la figure 1.3) sont issus du développement de la robotique et de l'informatique. Ils nécessitent la réalisation d'un modèle virtuel du patient réalisé à l'aide des technologies de l'imagerie (IRM, CT-Scan, PET...). Ce modèle permet au robot, suite à une planification, d'assister lors d'interventions minimalement invasives les gestes médico-chirurgicaux des professionnels en les rendant plus précis et sûrs. Ils ont également pour objectif d'améliorer les conditions opératoires du chirurgien, par exemple en réduisant le temps total de l'intervention ou encore la fatigue du chirurgien en lui permettant d'être assis plutôt que debout.



Figure 1.3 – Les robots chirurgicaux. (a) Le robot chirurgical Da Vinci pour la chirurgie mini-invasive (www.davincisurgery.com). (b) Le robot neuromate de Renishaw pour l'assistance à la neuro-chirurgie (www.renishaw.com/en/10712.aspx)

1.2.2 Les simulateurs médicaux

Les simulateurs médico-chirurgicaux (cf. la figure 1.4(a)) sont une technologie récente permettant de reproduire le déroulement d'une intervention médico-chirurgicale minimalement invasive. Ils ont pour but de former ou d'entraîner à la coordination avancée de la main et de l'œil mais également aux gestes complexes car contraints, spécifiques à ce type d'intervention. Ils peuvent mettre en place divers scénarii pour la formation, ou encore générer des cas rares ou complexes pour les utilisateurs confirmés. Les simulateurs médico-chirurgicaux permettent également, dans un objectif de planification pré-opératoire, de pouvoir tester et/ou d'optimiser un protocole opératoire en prédisant le résultat de gestes chirurgicaux.



Hervé Delingette dans [Del99] présente les différentes versions de ces derniers. Les simulateurs médico-chirurgicaux dits de première génération permettent de se déplacer dans une représentation virtuelle du patient générée à partir d'images médicales. Cette version a pour objectif d'améliorer le diagnostic de certaines pathologies. Elle est également utilisée pour la planification opératoire. Cependant, elle ne permet quasiment aucune interaction avec les organes géométriques. Les simulateurs de deuxième génération consistent en une amélioration de la première version, prenant en compte les propriétés biomécaniques des organes considérés afin de simuler leur comportement ainsi que les différentes interactions auxquels ils sont soumis, notamment avec des instruments chirurgicaux. La réalité virtuelle permet à l'utilisateur de pouvoir suivre le déroulement de l'intervention ; couplée à un système haptique, d'avoir un retour visuel et tactile de ses interactions sur les organes. Cette dernière version nécessite une modélisation des organes et la simulation réaliste de leur comportement ainsi que de leurs interactions en temps interactif.

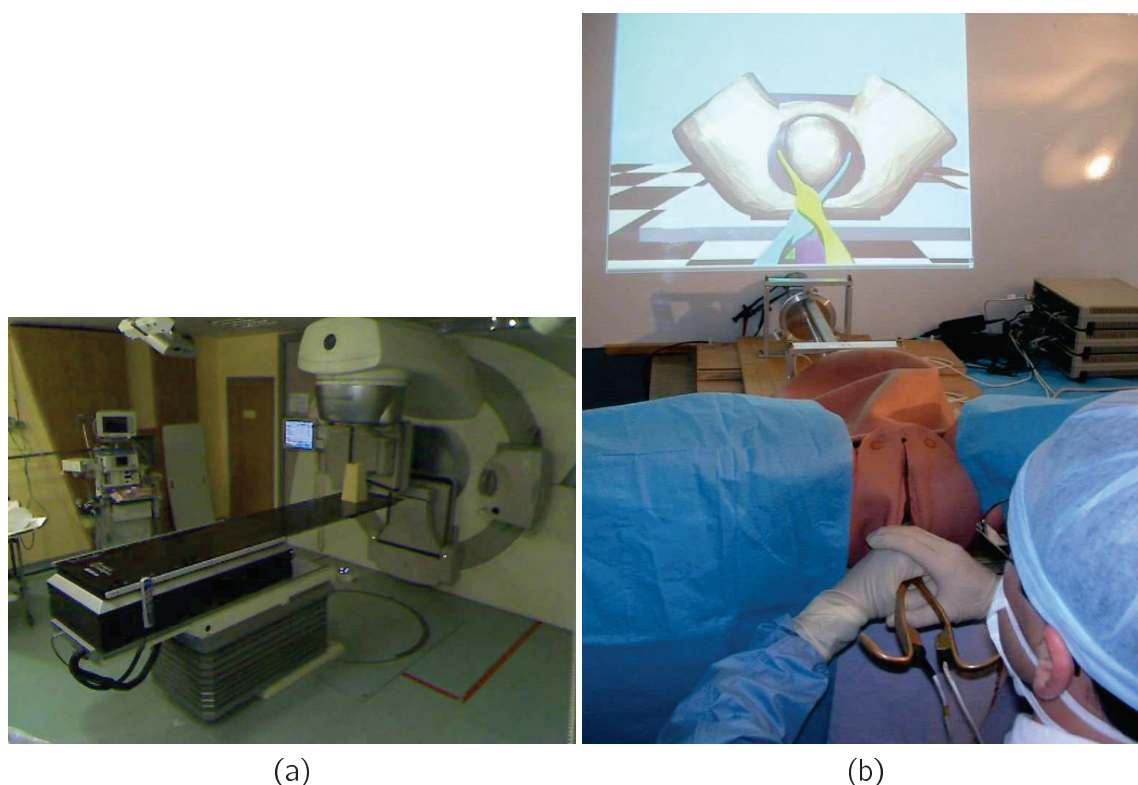


Figure 1.4 – Les applications de la simulation du comportement des organes. (a) Salle de traitement en radiothérapie. (b) BirthSim : simulateur pour l'aide à l'apprentissage du geste médical de l'accouchement [Mor07].

1.3 La simulation du comportement des objets déformables

La modélisation et la simulation des organes sont plébiscités pour le développement des technologies médicales du futur comme par exemple l'hadronthérapie (cf. la figure 1.4(a)). Dans ce sens, l'équipe SAARA (Simulation, Analyse, Animation pour la Réalité Augmentée) participe au projet européen ETOILE (Espace de Traitement Oncologique par Ions Légers dans le cadre Européen) qui a pour vocation l'élaboration d'un complexe médical dédié au traitement de tumeurs cancéreuses par faisceaux d'ion carbone. Ils travaillent, entre autres,

sur la réalisation d'un modèle physique permettant de simuler le mouvement des poumons lors de la respiration. Ces travaux ont pour but d'obtenir une connaissance précise du mouvement de la tumeur du patient traité pour que l'irradiation soit la plus efficace possible.

Enfin, la modélisation et la simulation des organes sont des problématiques prenant place dans le projet SAGA (Simulateurs pour l'Apprentissage des Gestes de l'Accouchement), de l'équipe SAARA. Ce projet a pour objectif l'élaboration d'un simulateur médical reproduisant de manière réaliste les sensations ressenties par la sage-femme, lors d'un accouchement (cf. la figure 1.4(b)). Il a pour objectif la formation du personnel médical, et se présente en deux parties avec un modèle numérique et un dispositif mécanique. Le modèle numérique vise à simuler virtuellement les contacts et les efforts mis en jeu au moment de l'accouchement. Il permet également la visualisation interactive des déformations des éléments anatomiques modélisés comme l'abdomen, l'utérus, le placenta, le bassin osseux, le plancher pelvien et le fœtus. Le dispositif mécanique prend en compte les mouvements effectués par l'utilisateur et commande une interface haptique en intégrant les efforts calculés par la simulation numérique.

1.4 Verrous scientifiques

Les simulateurs ayant pour objectif l'aide à l'apprentissage des gestes médicaux-chirurgicaux, nécessitent des simulations physiques d'objets déformables, interactives, ainsi que la possibilité de réaliser des changements topologiques sur les modèles virtuels d'organes, au cours de la simulation.

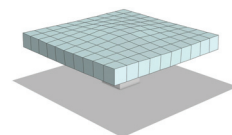
Dans ce sens, nous nous sommes intéressés à l'élaboration d'un modèle unifié couplant un modèle topologique et un modèle physique, pour la simulation physique d'objets déformables, tout en réalisant pendant cette simulation des changements topologiques comme la découpe ou la subdivision d'éléments.

Nous avons fait le choix des cartes combinatoires et plus particulièrement celui des complexes cellulaires linéaires, comme modèle topologique de notre modèle unifié. Ils permettent de représenter des objets surfaciques comme volumiques, par une décomposition en cellules. Ils ont pour principal intérêt d'offrir un accès à l'ensemble des relations d'incidence et d'adjacence entre les cellules de l'objet modélisé, facilitant ainsi la gestion des changements topologiques. De plus, leur définition mathématique permet de garantir la validité topologique du modèle, au cours des changements topologiques.

Nous avons fait le choix d'un système masses-ressorts pour le modèle physique de notre modèle unifié. L'avantage de ce dernier réside dans la simplicité de ses équations, son implémentation intuitive, son interactivité et sa facilité à gérer les changements topologiques. La paramétrisation des coefficients de raideur de notre système masses-ressorts est permise par une méthode analytique proposée par Baudet *et al.*. Leur méthode permet la formulation des coefficients de raideur des ressorts en fonction des paramètres rhéologiques des matériaux que sont le module de Young et le coefficient de Poisson. Par contre, l'utilisation de leur méthode implique des modèles 2D composés de quadrilatères et des modèles 3D constitués d'hexaèdres.

Les modèles topologiques exclusivement 2D sont largement employés dans le domaine de la simulation physique. Ils permettent de stocker l'ensemble des informations géométriques et topologiques du modèle simulé, tout en permettant de faciliter l'accès à ces informations. C'est donc naturellement que nous nous sommes intéressés à l'emploi d'un modèle topologique plus complexe pour la simulation physique d'objets 3D. Enfin, la définition d'un modèle unifié évite la redondance d'informations et facilite la mise à jour de ces dernières.

Ces travaux montrent les bénéfices de l'emploi d'un modèle topologique pour la simulation



physique, au travers de l'implémentation de changements topologiques comme la découpe en 2D, selon les arêtes, et en 3D, selon les faces du modèle, ainsi que la subdivision de quadrilatères ou d'hexaèdres. Nous détaillons dans ce manuscrit comment réaliser ces changements topologiques d'un point de vue topologique, puis d'un point de vue physique avec la mise à jour des informations embarquées au niveau du modèle topologique.

Enfin, l'opération de subdivision locale des quadrilatères ou des hexaèdres du modèle, que nous avons implémentée, nous permet de proposer une méthode adaptative. Une méthode adaptative repose sur le constat que plus un maillage est fin, plus la simulation du modèle sera précise, mais en contrepartie elle sera plus gourmande en temps de calcul. Elle permet alors un compromis entre la précision et le temps de simulation, avec un maillage constitué d'éléments de différents niveaux de résolution. L'objectif étant de raffiner ou de dé-raffiner localement les éléments du maillage, en fonction d'un critère s'appuyant sur les besoins en précision de la simulation de manière locale. Notre méthode adaptative consiste à débiter avec un modèle contenant peu d'éléments, et à raffiner ces derniers, selon un critère géométrique.

1.5 Organisation du manuscrit

Nous avons décomposé ce manuscrit en quatre chapitres :

Pré-requis fondés sur la littérature

Le chapitre 2 de ce manuscrit présente les pré-requis nécessaires à la compréhension de nos contributions ainsi que de nos choix.

Dans un premier temps, nous introduisons tout d'abord ce qu'est un objet déformable et comment le modéliser. Puis, nous présentons brièvement la mécanique des milieux continus, permettant de définir physiquement le comportement d'objets déformables. Enfin, nous explicitons la notion de schéma d'intégration permettant la simulation numérique d'objets déformables.

Dans un second temps, nous présentons les différents modèles existants pour l'animation d'objets déformables. Cette partie permet d'introduire les pré-requis nécessaires à l'animation ou la simulation physique d'objets déformables. C'est également l'occasion de présenter notre choix de modèle physique.

Dans un troisième temps, nous explicitons ce qu'est un modèle topologique, avec les différents modèles existants. Nous présentons quelques applications nécessitant l'emploi d'un modèle topologique. Enfin, cette partie expose notre choix de modèle topologique.

Dans un dernier temps, nous présentons un bref état de l'art sur les changements topologiques que sont les méthodes adaptatives et la découpe. En effet, le modèle unifié défini dans nos travaux montre son intérêt dans la réalisation de changements topologiques, dont nous en avons réalisé une implémentation basique. Cette partie permet d'aborder la littérature répondant à la problématique des modèles multi-résolutions avec les méthodes adaptatives, et celle de la découpe.

Modèle unifié alliant topologie et physique

Le chapitre 3 détaille nos contributions avec tout d'abord, la définition de notre modèle unifié. Nous présentons ensuite sa construction, et l'initialisation de ses paramètres physiques permettant la simulation physique d'objets déformables. Enfin, nous détaillons l'opération permettant sa découpe.

Raffinement local

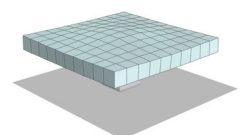
Le chapitre 4 développe les modifications apportées à notre modèle unifié pour gérer le changement topologique consistant, en 2D, en la subdivision d'un quadrilatère en quatre sous-quadrilatères et en 3D, en la subdivision d'un hexaèdre en huit sous-hexaèdres. Enfin, nous expliquons les modifications apportées au processus de découpe pour gérer un modèle localement raffiné.

Résultats préliminaires

Le dernier chapitre de ce manuscrit présente quelques résultats préliminaires de notre méthode. Nous montrons notamment les capacités de notre modèle unifié à gérer des changements topologiques comme la découpe selon les arêtes, en 2D, et les faces, en 3D, ainsi que le raffinement de quadrilatères et d'hexaèdres avec les résultats de notre méthode adaptative.

1.6 Publications

- Nos travaux ont amené à une première publication en 2013, à la conférence internationale WSCG (International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision), intitulée : A generic topological framework for physical simulation. Cette publication présente la définition et l'intérêt de notre modèle unifié au travers du changement topologique de la découpe.
- Puis, l'adaptation de notre modèle unifié pour la mise en place d'une méthode unifiée avec le raffinement local des éléments, a permis une deuxième publication en 2014, à la conférence internationale VRIPHYS (Workshop on Virtual Reality Interaction and Physical Simulation), intitulée : A Unified Topological-Physical Model for Adaptive Refinement.



Chapitre 2

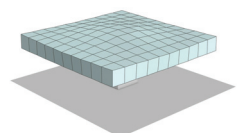
« Les savants ont calculé que les chances d'exister d'un phénomène aussi manifestement absurde sont de une sur un million. Mais les magiciens, eux, ont calculé que les chances uniques sur un million se réalisent neuf fois sur dix. »

(Les Annales du Disque-Monde, Mortimer - Terry Pratchett)

Pré-requis fondés sur la littérature

Sommaire

2.1 Objets déformables	16
2.1.1 Détermination des paramètres rhéologiques des matériaux	17
2.1.2 Génération de maillages	18
2.1.3 La Mécanique des Milieux Continus	20
2.1.4 Les schémas d'intégration	28
2.2 Modèles pour les objets déformables	29
2.2.1 Modèles non-physiques ou géométriques	30
2.2.2 Modèles physiques	30
2.2.3 Conclusion sur la simulation physique d'objets déformables.	45
2.3 Modèle topologique	45
2.3.1 Quelques notions de topologie	47
2.3.2 Structure de données topologiques pour les maillages surfaciques	50
2.3.3 Structure de données nD	55
2.3.4 Travaux relatifs aux structures de données topologiques	70
2.3.5 Conclusion sur les structures de données topologiques.	76
2.4 Changements topologiques	77
2.4.1 Multi-résolution	78
2.4.2 Découpe	81
2.5 Conclusion	83



Dans la communauté d'informatique graphique, la simulation physique d'objets déformables est un domaine rassemblant de nombreuses et diverses problématiques. Nous pouvons citer celle de la définition de nouvelles lois de comportement avec l'identification des paramètres physiques des matériaux, la segmentation d'images issues de scanner CT ou d'IRM, la génération de maillages adaptés à la simulation physique, la gestion des collisions entre objets déformables ou encore l'optimisation et la parallélisation de l'implémentation d'un modèle physique dans une optique temps-réel.

Dans cette partie, nous présenterons certaines des problématiques inhérentes à la simulation physique d'objets déformables.

La simulation physique d'un objet déformable réel nécessite en premier lieu, la conception de son modèle virtuel. Pour cela, nous nous intéresserons tout d'abord, à la détermination des **paramètres physiques** du matériau constituant l'objet déformable réel. Ces paramètres participent à la précision et au réalisme de la simulation du comportement de l'objet considéré. Puis, nous parlerons de l'élaboration du modèle géométrique avec la **génération de maillages** effectuée à partir de données issues de l'imagerie médicale. Cette étape permet la discrétisation d'un objet continu à simuler, afin de pouvoir le représenter virtuellement. Pour simuler un objet déformable, les mécaniciens ont élaboré la **mécanique des milieux continus**, formalisant le comportement d'un objet déformable. Nous exposerons cette théorie qui amènera ensuite à l'introduction de la **méthode des éléments finis**. Cette dernière permet de simuler le comportement d'un objet déformable en se fondant sur la mécanique des milieux continus. Enfin, dans le cas d'une résolution dynamique, nous présenterons les différents types de **schémas d'intégration**, permettant le calcul des nouvelles positions de l'objet modélisé et simulé, au cours du temps.

Nous exposerons ensuite quelques modèles de la littérature permettant la simulation d'objets déformables. Nous présenterons plus particulièrement les modèles dits **physiques** avec notamment la description du **système masses-ressorts**, qui est le modèle physique que nous avons employé dans nos travaux. De plus, nous introduirons également les différentes méthodes de la littérature permettant son paramétrage. Enfin, nous justifierons le choix de ce modèle physique.

Nous expliciterons ensuite la notion de modèle topologique, avec la présentation de modèles topologiques exclusivement 2D amenant à l'introduction de modèles topologiques nD . Nous définirons plus particulièrement le modèle topologique des cartes combinatoires, que nous avons employé dans nos travaux. Nous présenterons les caractéristiques de ce modèle et ses avantages pour la simulation physique. Enfin, nous introduirons quelques travaux relatifs aux modèles topologiques.

Enfin, nous présenterons brièvement la littérature sur les changements topologiques que sont la découpe et la multi-résolution. En effet, dans nos travaux, nous avons implémenté une opération de découpe et une méthode adaptative basique.

2.1 Objets déformables

Un objet **indéformable** ne peut subir que des translations ou rotations (six degrés de liberté), contrairement à un objet déformable. Plus précisément, la distance entre chaque paire de points d'un objet indéformable ne varie pas au cours du temps. Alors qu'un objet **déformable** est, quant à lui, soumis à des déformations élastiques, plastiques ou induisant une rupture dans l'objet.

2.1.1 Détermination des paramètres rhéologiques des matériaux

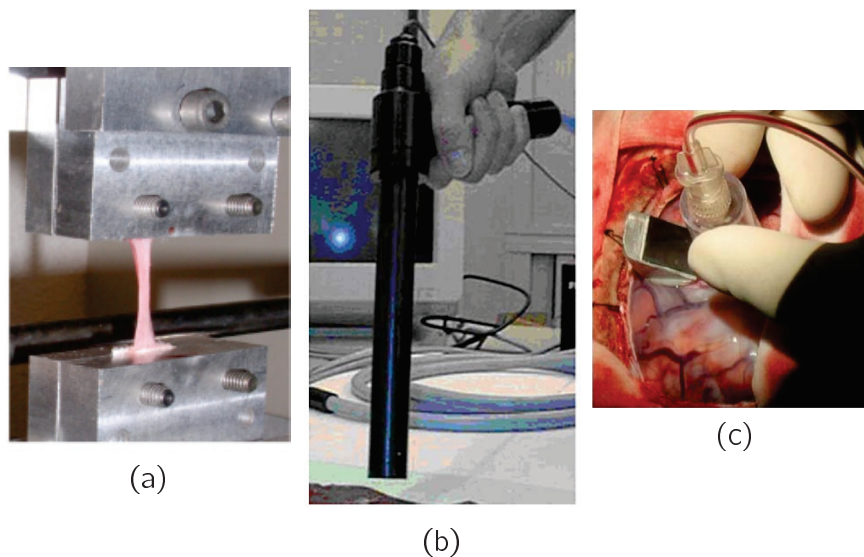
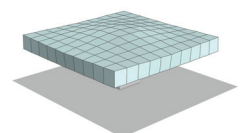


Figure 2.1 – Tests mécaniques permettant la détermination des propriétés mécaniques d'un matériau. (a) Tests mécaniques réalisés *ex vivo* sur un prélèvement organique [FB03]. (b) Un instrument par aspiration permettant de récupérer *in vivo*, les propriétés mécaniques du col de l'utérus [MNB⁺06]. (c) Un instrument par aspiration permettant de mesurer, par exemple, l'élasticité du cerveau pendant une neuro-chirurgie [SCB⁺, SPP10]. La déformation du tissu aspiré est visualisé grâce à un miroir relié à une caméra externe.

La simulation d'un objet déformable réel nécessite d'une part l'élaboration d'une représentation géométrique et topologique de cet objet, et d'autre part la récupération de ses propriétés mécaniques permettant de retranscrire son comportement au niveau de modèle physique employé.

La récupération des paramètres mécaniques d'un objet réel est un domaine de recherche à part entière. Lors de la récupération de ces paramètres, l'objet peut se situer dans différents environnements : **in vivo**, **ex vivo** et **in vitro**. L'environnement *in vivo* implique que les tests effectués sur la partie organique à tester se fasse dans l'organisme vivant l'incluant. *Ex vivo* consiste en un environnement artificiel accueillant la partie de l'organisme vivant à tester. Enfin, dans un environnement *in vitro*, les tests sont réalisés sur la partie organique, en dehors de l'organisme vivant auquel elle appartenait (*cf.* la figure 2.1). Nous précisons que les tests effectués dans un environnement *in vitro* ou *ex vivo* sont plus faciles à réaliser. Cependant, les valeurs obtenues sont souvent inexactes. De plus, il est souvent difficile pour des raisons techniques ou éthiques de faire des expérimentations *in vivo*. Enfin, des données autres que relatives aux paramètres mécaniques, et dépendant de conditions particulières permises seulement par un environnement *in vivo*, peuvent être requises pour la simulation physique de l'organe considéré, comme par exemple la récupération de contraction utérine.

Un **protocole** détermine la manière dont les paramètres mécaniques d'un matériau sont récupérés. Notons qu'il est difficile d'obtenir la valeur exacte de ces paramètres. En effet, les valeurs obtenues peuvent varier d'un facteur mille selon le protocole employé et selon le technicien réalisant le protocole. De plus, les valeurs diffèrent même entre les différents essais réalisés par une même personne. Enfin, nous précisons que les paramètres mécaniques d'un organe varient en fonction de l'âge du patient ou encore en fonction de la pathologie relative à la partie organique testée. Par exemple, la rigidité de l'utérus augmente fortement avec l'âge de la femme testée.



2.1.2 Génération de maillages

La simulation d'un objet réel nécessite au préalable, la génération d'une représentation surfacique ou volumique de cet objet. Il existe de nombreuses méthodes permettant la création d'un maillage à partir d'un objet réel. Elles sont relatives aux types de données souhaités en entrée et en sortie, ainsi qu'au temps nécessaire à la génération du maillage ou encore à la qualité du maillage obtenu.



Figure 2.2 – Génération de maillages à partir d'images médicales. A gauche : Des coupes parallèles issues d'images scanner. A droite : La segmentation de ces images [Nes08] (Données issues de l'IRCAD).

Nous présentons ci-dessous, un exemple de pipeline permettant la génération d'un maillage pour la simulation physique d'un objet réel de type foie [Fre03].

- Le processus prend en entrée une acquisition ou numérisation d'un buste humain grâce à des outils d'imagerie médicale comme l'**IRM (Imagerie à Résonance Magnétique)** ou le **Scanner CT (Computer Tomography)** permettant de collecter de larges volumes de données (cf. la figure 2.2). Les données médicales issues de cette étape peuvent être des **séries de coupes parallèles**, des **ensembles de voxels** ou des **nuages de points**. Les données médicales se présentant sous la forme d'une série de coupes parallèles, consistent en un ensemble de plans régulièrement espacés. Chaque plan correspond à une image, soit un tableau 2D de pixels. Un scalaire indiquant la densité de l'organe est associé à chacun de ces pixels. Dans la suite de la description du pipeline, nous prenons l'exemple de données en entrée de type : série de coupes parallèles.
- Ces données brutes sont ensuite segmentées, filtrées ou encore améliorées, afin d'extraire la zone d'intérêt comme un organe particulier. La figure 2.2 montre la segmentation d'un foie à partir d'une série de coupes parallèles. La segmentation est une étape complexe souvent manuelle où certains traitements permettent l'emploi d'algorithmes automatiques. Dans notre exemple, elle consiste en la réalisation d'un traitement d'images, permettant la détection des contours bidimensionnels sur chacune des différentes coupes.
- Les données brutes filtrées sur l'organe en question sont ensuite traitées en vue de générer le maillage associé. Des algorithmes de reconstruction sont alors employés afin de générer une représentation polygonale (maillage surfacique), de l'organe sélectionné. Une des manières intuitives de reconstruire cette triangulation est de construire des triangles à partir des points sur les contours ainsi définis, entre deux coupes adjacentes. Enfin, notons que des critères géométriques minimisant les angles des triangles peuvent par exemple être utilisés pour contrôler la qualité de la triangulation.

- Notons que l'étape de création du maillage surfacique nécessite souvent des processus d'extrapolation des parties manquantes afin d'obtenir des maillages fermés ou des processus de recalage lorsque les données ne proviennent pas de la même acquisition.
- Le maillage surfacique, émanant de données brutes très volumineuses, doit alors être drastiquement épuré du fait de son trop grand nombre d'éléments. Dans cette optique, des algorithmes de simplification à base de suppression ou de bascule d'arêtes sont employés. Ces derniers permettent également de corriger les différents artefacts géométriques inhérents au processus de reconstruction (cf. la méthode de Delaunay).
- Enfin, un processus d'adaptation de maillage peut être employé afin d'optimiser le maillage obtenu dans un objectif de simulation physique. Il consiste en l'élaboration d'estimateurs d'erreur géométrique ou physique. Un processus transcrit ensuite l'erreur calculée en modifiant localement le maillage. Un ensemble d'itérations est alors effectué à partir du maillage modifié, jusqu'à l'obtention d'un maillage optimal pour la simulation physique. Nous citons par exemple les travaux de Frey *et al.* [FB03] proposant une estimation d'erreur géométrique *a posteriori* reposant sur le Hessien de la surface à remailler.
- Une dernière étape pourrait consister en la génération d'un maillage volumique grâce à une génération de tétraèdres à partir du maillage surfacique obtenu. Bern *et al.* [BP00] présentent un état de l'art détaillé sur la génération de maillage pour la méthode des éléments finis.

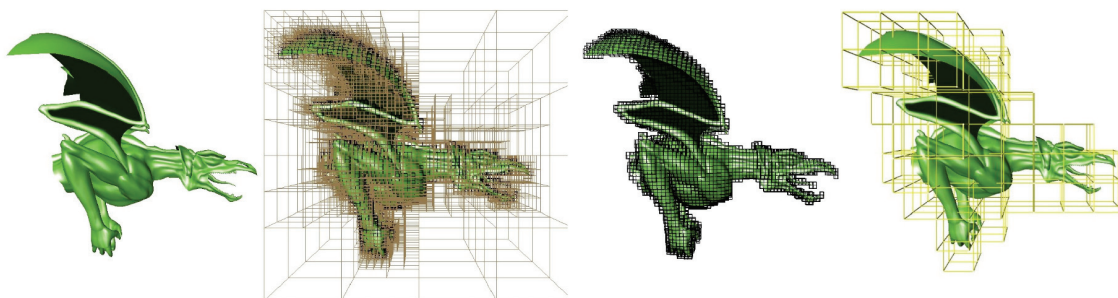
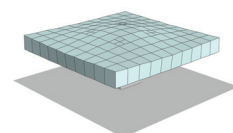


Figure 2.3 – Génération d'un maillage volumique à partir d'une surface polygonale. De gauche à droite : Une surface polygonale représentant un dragon, L'arbre octal associé, L'arbre octal où seulement les éléments les plus fins sont représentés et L'arbre octal où seulement les éléments d'un niveau de division intermédiaire sont représentés [Nes08].

Les travaux de Nesme *et al.* [NFP06, Nes08] proposent également une méthode de génération de maillages pour la simulation physique. Pour générer un maillage volumique régulier à partir de la surface polygonale d'un objet réel à animer, ils commencent par récupérer la boîte englobante de cette surface (cf. la figure 2.3). Ils construisent ensuite un octree à partir de cette boîte englobante. Cette dernière est ainsi subdivisée régulièrement en huit sous-éléments. Ils procèdent alors de manière récursive sur les sous-éléments se trouvant à cheval sur la surface ou à l'intérieur. La récursion s'arrête pour un niveau de subdivision donné. Une fois le maillage volumique construit, ils appliquent la mécanique sur ce dernier, la surface est alors déformée par interpolation linéaire. La surface suit ainsi les déformations des voxels l'englobant.

Afin de conclure cette partie sur la modélisation géométrique et biomécanique d'un objet déformable de type organique, nous citons les travaux de Buttin *et al.* [BZS⁺13]. Ces derniers proposent une modélisation biomécanique des interactions entre le fœtus et les principaux



organes de la parturiente impliqués durant l'accouchement que sont l'utérus, l'abdomen et le bassin. Le modèle géométrique des différents organes est issu de données scanners. Pour réaliser cette modélisation biomécanique, différents choix ont été effectués, comme entre autres, les organes à modéliser, la modélisation des contraintes utérines et des forces d'expulsions ou encore la loi de comportement. Pour cette dernière, ils ont choisi la **loi de Hooke linéaire et isotrope** pour la modélisation des os du bassin et la **loi de Neo-Hooke hyper-élastique** pour le fœtus, l'abdomen et l'utérus. Enfin, les propriétés mécaniques des différents organes ont été récupérées dans la littérature.

Notons que l'acquisition des données médicales ne permet pas l'obtention de toutes les informations nécessaires à la construction du modèle biomécanique [RLGB⁺13]. En effet, dans le cas de la modélisation du système pelvien chez la femme, il est nécessaire, pour chaque composant du système pelvien, d'acquiescer par imagerie les données médicales relatives au composant considéré, puis de segmenter ces données et enfin de reconstruire le maillage associé à ce composant. Les différents maillages ainsi obtenus doivent alors être assemblés en les positionnant spatialement les uns par rapport aux autres, afin de reconstituer virtuellement le système pelvien. Le modèle géométrique du système pelvien ainsi constitué, il faut alors déterminer les paramètres mécaniques des différents composants permettant la définition de leur comportement. Une fois le modèle biomécanique du système pelvien construit, une première simulation est effectuée. Dans notre illustration par cet exemple, cette simulation permet de se rendre compte d'une mobilité incorrecte se traduisant par la vessie passant au travers d'un autre organe. Il faut alors se tourner vers l'anatomie pour récupérer les différentes informations manquant à ce modèle comme par exemple la mise en place de ligaments qui n'ont pas été identifiés lors de l'acquisition numérique. Or, il n'y a pas de consensus médical sur la description anatomique du système pelvien et plus particulièrement sur le nombre et la position de ces ligaments. Une solution consiste alors à utiliser l'analyse d'images afin de quantifier l'intensité des déplacements des organes. Cette analyse des déplacements est ensuite mise en parallèle avec la simulation du système pelvien. Des ligaments sont ainsi rajoutés au modèle en fonction de la réalité IRM (donnant l'intensité des déplacements). Le modèle ainsi obtenu permet un comportement lors de la simulation en adéquation avec la réalité.

2.1.3 La Mécanique des Milieux Continus

Dans cette partie, nous présentons de manière brève et sommaire, la mécanique des milieux continus. Pour cela, nous nous plaçons dans un espace tridimensionnel. Les trois axes de cet espace sont notés : x , y et z .

À l'échelle nanoscopique, la matière est un agrégat de molécules. Elle est perçue comme étant discontinue. Par contre, à l'échelle macroscopique, elle se présente comme étant continue. Un objet est constitué d'un nombre infini de molécules. Il est donc impossible de décrire le milieu à l'échelle d'une molécule. C'est pour cette raison que l'on fait l'hypothèse des milieux continus. Cette dernière considère des **paquets de molécules ou de particules**. La taille de ces paquets est assez importante, par rapport à la distance inter-moléculaire, pour que puissent être négligées les fluctuations des molécules que le paquet renferme. De plus, les propriétés physiques telles que la masse volumique ou la vitesse, sont considérées comme variant de manière continue d'un paquet de particules du milieu continu à un autre.

La **mécanique des milieux continus** est considérée comme étant un modèle mathématique permettant de simplifier la définition des propriétés physiques en se plaçant à l'échelle de ces paquets de particules. À chaque paquet de particules est associé des grandeurs physiques, représentées sous forme d'un **tenseur**. Par exemple, la masse d'un paquet de particules est

un scalaire, représentée par un tenseur d'ordre 0. La force appliquée à un paquet de particules est un vecteur qui est représentée par un tenseur d'ordre 1. Enfin, les déformations et les contraintes appliquées à un paquet de particules sont représentées par des tenseurs d'ordre 2. Les tenseurs d'ordre 2 sont représentés en 3D par des matrices 3×3 . Enfin, chaque grandeur physique, représentée sous forme de tenseur au niveau de chaque paquet de particules du milieu continu, correspond à un **champ continu** au niveau de ce milieu.

Il existe deux approches pour décrire le milieu continu : l'approche **lagrangienne** et l'approche **eulérienne**. L'approche lagrangienne consiste à suivre le mouvement d'un paquet de particules. Contrairement à l'approche eulérienne se définissant comme un point d'observation fixe au cours du temps, où l'on regarde les grandeurs physiques des paquets de particules passant devant ce point. Pour les objets déformables, l'approche lagrangienne est préférée alors que celle eulérienne est privilégiée pour les fluides. De manière plus précise, dans l'approche lagrangienne, l'observateur suit chaque paquet de particules après l'avoir identifié par sa position initiale. La **position** au temps t d'un paquet de particules est notée $\mathbf{X}(t)$. La **vitesse** d'un paquet de particules au temps t est notée $\dot{\mathbf{X}}(t)$. Elle correspond à la dérivée en fonction du temps, de sa position au temps t . Son **accélération** au temps t est notée $\ddot{\mathbf{X}}(t)$. Elle est définie comme la dérivée en fonction du temps, de sa vitesse au temps t et comme la dérivée seconde en fonction du temps, de sa position au temps t .

On nommera par la suite, un **objet**, un milieu continu constitué d'un ensemble de paquets de particules. La notion de déformation d'un objet naît de la comparaison entre une configuration dite « initiale » et une configuration qualifiée de « finale » [CNM07]. Plus précisément, deux configurations d'un même objet aux temps t_1 et t_2 avec $t_1 \leq t_2$, sont considérées. Si entre ces deux configurations, il n'y a pas eu conservation des distances et des angles entre les paquets de particules, alors l'objet a été déformé entre t_1 et t_2 . Sous l'action de forces, un objet déformable se déforme. Les positions au temps t_1 des paquets de particules constituant l'objet déformable, sont ainsi différentes des positions de ces mêmes paquets de particules au temps t_2 .

Nous détaillons ci-dessous les différentes étapes du pipeline de la mécanique des milieux continus. Ce dernier consiste tout d'abord à évaluer les **déplacements** des différents paquets de particules composant l'objet considéré. Puis, le **champ de déformation** de l'objet est évalué à partir des déplacements juste obtenus. Le **champ de contrainte** de l'objet est ensuite déterminé en fonction de son champ de déformation. Enfin, les **forces** appliquées à l'objet sont calculées en fonction du champ de contrainte de ce dernier.

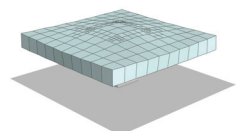
Le déplacement

Le **déplacement** \mathbf{U} d'un paquet de particules p , est défini comme le vecteur entre la position de p au temps t_1 et sa position au temps t_2 , respectivement notées $x(t_1)$ et $x(t_2)$. Le déplacement entre les temps t_1 et t_2 du paquet de particules p , noté $\mathbf{U}(p)|_{t_1-t_2}$, est ainsi formulé de la manière suivante : $\mathbf{U}(p)|_{t_1-t_2} = x(t_2) - x(t_1)$.

Du déplacement à la déformation : Le tenseur de déformation

Comme précisé précédemment, la **déformation** est la description de la manière dont se déforme un objet. Elle est décrite par une matrice de neuf éléments. Le **tenseur de déformation** : $\bar{\epsilon}$ est représenté de la manière suivante :

$$\bar{\epsilon} = \begin{bmatrix} \epsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \epsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \epsilon_z \end{bmatrix}$$



Les valeurs du tenseur rendent compte d'un étirement (compression ou élongation) ou d'un cisaillement, dans les différentes directions. ϵ_i ($i \in \{x, y, z\}$) correspond à un étirement selon l'un des trois axes i du repère tridimensionnel. γ_{ij} représente un cisaillement angulaire entre les axes i et j ($j \in \{x, y, z\}$ et $i \neq j$) du repère tridimensionnel. Le tenseur de déformation est défini en tout paquet de particules de l'objet, on parle alors de **champ de déformation**.

La **formulation de Green-Lagrange** permet pour chaque paquet de particules, de déterminer les neuf valeurs du tenseur de déformation en fonction du déplacement de ces dernières. Le **tenseur de Green-Lagrange** : $\overline{\overline{\epsilon}}_{GL}$ est formulé de la manière suivante :

$$\overline{\overline{\epsilon}}_{GL} = \frac{1}{2}(\nabla \mathbf{U} + \nabla \mathbf{U}^T + \nabla \mathbf{U}^T \cdot \nabla \mathbf{U})$$

où $\nabla \mathbf{U}$ est le gradient du déplacement, c'est-à-dire la somme des dérivées partielles de chaque composante de \mathbf{U} en fonction de x , y et z . b^T correspond à la transposée de b .

Notons que la linéarisation du tenseur de Green-Lagrange donne le **tenseur de déformation de Cauchy**, dont la formulation est la suivante :

$$\overline{\overline{\epsilon}}_C = \frac{1}{2}(\nabla \mathbf{U} + \nabla \mathbf{U}^T)$$

Ce tenseur est souvent employé car il est moins coûteux en terme de calcul que le tenseur de Green-Lagrange, du fait de son hypothèse de petites déformations et de petits déplacements rotationnels. Par contre, il n'est pas invariant en rotation, provoquant par exemple, lors d'un test de la poutre encastree soumise à la gravité, un gonflement de son extrémité. Afin de résoudre ce problème, une solution consiste à employer une méthode dite **corotationnelle** [NPF05, Nes08].

De la déformation à la contrainte : Les lois de comportement

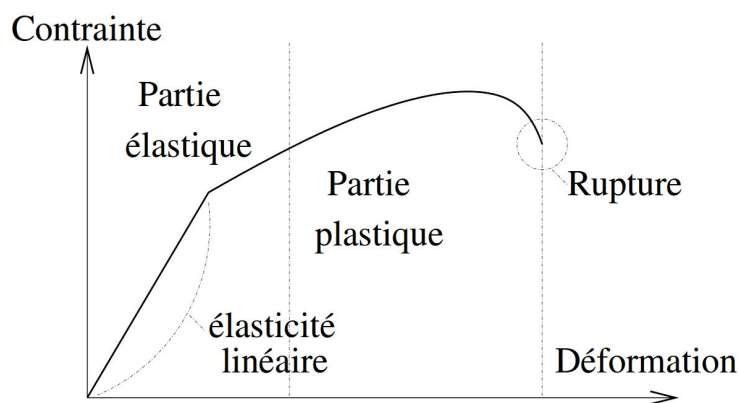


Figure 2.4 – La relation entre la déformation et la contrainte [Vil11].

La réponse d'un objet soumis à des sollicitations extérieures dépend du matériau le constituant. On appelle **loi de comportement** f d'un matériau, la relation liant le tenseur de déformation au tenseur des contraintes :

$$\overline{\overline{\epsilon}} = f(\overline{\overline{\sigma}})$$

Il existe de nombreuses lois de comportement. En effet, chaque matériau ne se comporte pas de la même manière lorsqu'il est soumis à des forces extérieures. Un matériau peut avoir

un comportement dit **élastique** (cf. la partie élastique sur la figure 2.4). Dans ce cas, le matériau est déformé sous l'action de forces, puis regagne son état initial lorsque ces forces disparaissent. Plus précisément, chaque paquet de particules rejoint son état initial en passant par les mêmes positions, que celles ayant amenées jusqu'à l'état déformé.

La loi de comportement la plus simple décrit un matériau **élastique linéaire** (cf. la partie élasticité linéaire de la courbe présentée dans la figure 2.4). Elle consiste en un sous-ensemble de la loi de comportement élastique. Les contraintes sont alors proportionnelles aux déformations. Notons qu'il n'existe pas de matériau répondant à un comportement exactement linéaire. Enfin, la figure 2.4 montre que cette linéarité n'est valable que pour de petites déformations. En effet, on considère qu'au delà de 10%, un matériau n'est plus linéaire. On est alors dans le cas de grandes déformations. Si l'on conserve une loi linéaire pour de grandes déformations, le comportement du matériau simulé sera d'autant plus erroné que le pourcentage de déformation sera élevé.

La **loi de Hooke** est une loi de comportement pour les matériaux élastiques, linéaires, homogènes et isotropes. La propriété d'**isotropie** implique une invariance des propriétés physiques du matériau en fonction de la direction. Cette propriété permet de drastiquement simplifier le nombre de paramètres caractérisant le matériau à seulement deux. Ces deux paramètres sont les **coefficients de Lamé**, notés λ et μ . La loi de Hooke est formalisée de la manière suivante :

$$\bar{\sigma} = \lambda \text{tr}(\bar{\epsilon})I + 2\mu\bar{\epsilon}$$

où $\text{tr}(A)$ est la trace de la matrice carrée A , et I est la matrice identité.

De la contrainte à la force : Le tenseur des contraintes

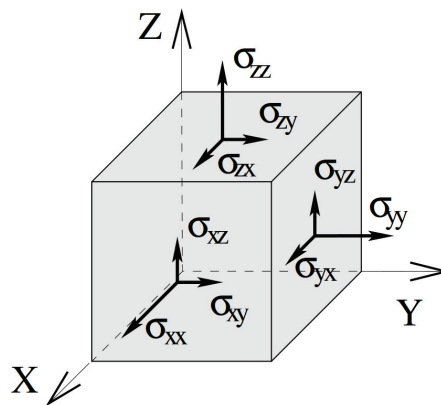
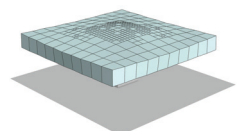


Figure 2.5 – Le tenseur des contraintes. La contrainte σ_{ij} ($i, j \in \{x, y, z\}$) est appliquée sur une facette ayant pour normale l'axe i et pour direction l'axe j [Vil11].

L'objectif est maintenant de calculer les déformations induites par les différentes forces appliquées au niveau de ces paquets de particules comme la gravité, ou dues à des collisions avec d'autres objets ou encore engendrées par des interactions utilisateurs. La **contrainte** est la manière dont on représente les forces qui agissent à l'intérieur d'un objet, lorsque ce dernier est déformé. Le **tenseur des contraintes** : $\bar{\sigma}$ est représenté de la manière suivante :



$$\overline{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix}$$

Dans la figure 2.5, un paquet de particules est représenté par un cube. Pour chaque facette d'un paquet de particules, des contraintes tangentielles et une contrainte perpendiculaire sont définies. Tout comme le champ de déformation, la définition d'un **tenseur des contraintes** en chaque paquet de particules, définit un **champ des contraintes**. Il existe plusieurs tenseurs des contraintes, le plus classique étant le **tenseur des contraintes de Cauchy**. Le tenseur des contraintes de Cauchy, noté $\overline{\sigma}_C$ est formulé de la manière suivante :

$$\mathbf{F} = \overline{\sigma}_C \mathbf{N} dS$$

où \mathbf{F} correspond au vecteur force s'exerçant sur un paquet de particules. \mathbf{N} est la normale de la facette considérée et dS est son aire.

Enfin, nous présentons deux modèles non-linéaires. Le **modèle de Mooney-Rivlin** consiste en une relation non-linéaire entre les déplacements et les déformations ainsi que entre les déformations et les contraintes. Le **modèle de Saint Venant-Kirchhoff** décrit, quant à lui, une relation linéaire entre la déformation et la contrainte, mais une relation non-linéaire entre les déplacements et les déformations.

Les paramètres rhéologiques du module de Young et du coefficient de Poisson

Le **module de Young** et le **coefficient de Poisson** sont des paramètres rhéologiques plus usités que les coefficients de Lamé, du fait qu'ils soient plus facilement interprétables expérimentalement. Il existe une correspondance entre le binôme module de Young et coefficient de Poisson, et les coefficients de Lamé.

Pour présenter le module de Young et le coefficient de Poisson, nous considérons l'exemple de la figure 2.6. Nous considérons alors une barre 3D de dimensions initiales : X_0 , Y_0 et Z_0 et dont la base est fixée (face grise dans la figure 2.6). Une force notée $F_{\text{étirement}}$, ayant pour direction l'axe Y et étant perpendiculaire à la face du dessus de la barre (face rouge dans la figure 2.6) est appliquée sur cette dernière. Suite à l'application de cette force, la barre est déformée avec une elongation de η selon l'axe Y , un rétrécissement de 2ϵ selon l'axe X et un rétrécissement de 2ζ selon Z .

Ce test d'étirement permet de définir le module de Young et le coefficient de Poisson de cette barre. En effet, le module de Young, noté E , est défini comme le rapport entre la contrainte et la déformation. La déformation est caractérisée par l'allongement relatif selon Y . Le module de Young est formalisé de la manière suivante :

$$E = \frac{F_{\text{étirement}}/S_0}{\eta/Y_0}$$

où S_0 est l'aire de la face soumise à la force d'étirement (face rouge dans la figure 2.6).

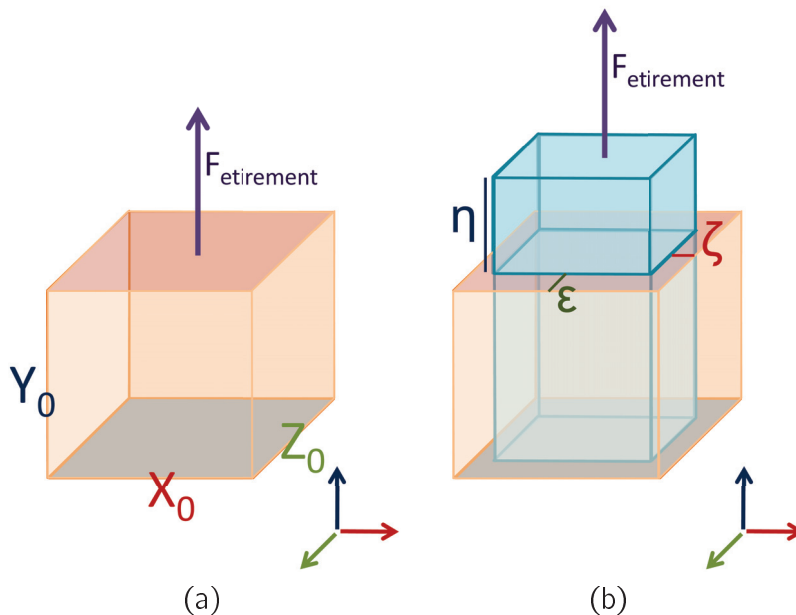


Figure 2.6 – Test d’étirement permettant la caractérisation des paramètres rhéologiques du module de Young et du coefficient de Poisson. (a) Une barre de dimension X_0 , Y_0 et Z_0 , dont sa face grise est fixée et sa face rouge est soumise à une force : $F_{\text{étirement}}$. (b) Le même modèle que (a) après l’application de $F_{\text{étirement}}$. La barre s’est étirée de η selon l’axe Y et s’est rétrécie de 2ζ selon l’axe X et de 2ϵ selon l’axe Z .

Le coefficient de Poisson est quant à lui déterminé en fonction des rétrécissements engendrés selon les axes X et Z . La formulation du coefficient de Poisson est la suivante :

$$\nu_x = \frac{2\zeta/Y_0}{\eta/X_0}$$

ou

$$\nu_z = \frac{2\epsilon/Y_0}{\eta/Z_0}$$

L’équation de mouvement

L’équation de mouvement a pour formulation :

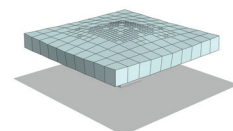
$$\rho \dot{\mathbf{U}} = \nabla \bar{\bar{\sigma}} + \rho \mathbf{F}$$

où ρ est la masse volumique de l’objet, $\bar{\bar{\sigma}}$ est le tenseur des contraintes et \mathbf{F} correspond à une action mécanique exercée sur le solide par les systèmes qui lui sont extérieurs. \mathbf{F} inclut classiquement une action de pesanteur.

Cette équation consiste en la généralisation de la seconde loi de Newton, énonçant le principe fondamental de la dynamique, formulé de la manière suivante :

$$\mathbf{F} = M \ddot{\mathbf{U}}$$

où \mathbf{F} est la somme des forces appliquées à l’objet, M et $\ddot{\mathbf{U}}$ correspondent respectivement à la masse et à l’accélération de l’objet.



Système d'équation aux dérivées partielles

La relation entre le déplacement et la déformation, la loi de comportement, la définition de la contrainte et l'équation de mouvement amènent à un système d'équations aux dérivées partielles. Or, il est souvent impossible de trouver une solution analytique à un problème de la mécanique des milieux continus. C'est pour cette raison que des méthodes numériques, comme la **méthode des éléments finis**, sont employées. La méthode des éléments finis permet d'approximer la solution d'une équation aux dérivées partielles.

Pour résoudre numériquement une équation aux dérivées partielles, appelée **formulation forte**, on commence par définir les conditions aux limites de type **Dirichlet** et **Neumann**. La condition de Dirichlet impose des valeurs que la solution de l'équation aux dérivées partielles doit vérifier au niveau des limites de l'objet. La condition de Neumann spécifie les valeurs de dérivée que la solution doit vérifier au niveau des limites de l'objet. Dans le cas d'une poutre en traction, la condition de Dirichlet permet l'encastrement d'une extrémité de la poutre et la condition de Neumann impose une force de traction à son autre extrémité.

Puis une fois les conditions aux limites spécifiées, l'équation aux dérivées partielles est transformée en une formulation intégrale équivalente, suite à l'application du principe des travaux virtuels. Cette étape consiste en la multiplication de l'équation aux dérivées partielles par une **fonction test**. Cette fonction test peut correspondre par exemple aux déplacements virtuels. Cette multiplication est suivie de l'intégration sur l'objet considéré. Le terme de plus haut degré de la formulation intégrale ainsi obtenue, est ensuite intégrée par partie (formule de Green). Les termes de bords générés sont alors calculés en fonction des conditions de Dirichlet et de Neumann préalablement définies. On obtient ainsi une **formulation faible** de l'équation aux dérivées partielles. Enfin, une **formulation variationnelle** peut être déduite de cette formulation faible. Notons que l'approximation par éléments finis nécessite l'écriture de la formulation faible ou variationnelle du problème posé par une équation aux dérivées partielles (cours de Marc Buffat, professeur à l'UFR de Mécanique de l'université Claude Bernard Lyon 1).

La résolution avec la méthode des éléments finis

Les termes de la formulation faible ou variationnelle sont des intégrales. L'intérêt de ces formulations est alors qu'elles peuvent être transcrites en somme d'intégrales sur un ensemble d'**éléments**. La méthode des éléments finis a donc pour première étape, la discrétisation de l'objet considéré en un nombre fini d'éléments, définissant alors un maillage. Notons que la méthode des éléments finis permet un maillage irrégulier. Les éléments sont généralement de forme simple comme des triangles en 2D ou des tétraèdres, hexaèdres en 3D. Les sommets des éléments sont appelés **nœuds**. Notons que dans le but d'augmenter le nombre de degrés de liberté, des points remarquables comme le centre de gravité de l'élément ou encore les milieux des arêtes de ce dernier, peuvent être considérés comme étant des nœuds. Nous précisons que l'ajout de degrés de liberté permet une amélioration de la précision des résultats.

Des fonctions d'interpolations polynomiales (linéaires ou quadratiques), appelées **fonctions de forme**, définissent alors le déplacement en tout point d'un élément en fonction des déplacements de ses nœuds. En conséquence, déterminer les déplacements nodaux de chacun des éléments permet d'approximer le champ de déplacement de l'objet entier. Afin de mieux se représenter la notion de fonction de forme, nous précisons qu'elle vaut 1 pour un nœud de l'élément considéré et 0 pour les autres nœuds de cet élément.

Après la discrétisation de l'objet et la définition des fonctions de forme, la méthode des éléments finis transcrit la formulation faible sur chaque élément de l'objet. Une équation est

ainsi définie en chaque élément de l'objet. Ces dernières sont ensuite assemblées donnant un système global qui est ensuite résolu.

Notons que pour une **résolution statique**, si l'on choisit un tenseur de déformation linéaire comme le tenseur de Cauchy et une loi de comportement linéaire comme la loi de Hooke, alors l'application de la méthode des éléments finis sur la formulation faible ou variationnelle du problème, se ramène à un système matriciel linéaire de type :

$$\mathbf{F} = \mathbf{K}\mathbf{U}$$

où K est la **matrice de raideur**.

En effet, dans ce cas, les forces sont proportionnelles aux déplacements, puisque les déplacements sont proportionnels aux déformations et que les déformations sont proportionnelles aux contraintes.

Une résolution statique est indépendante du temps contrairement à une résolution dynamique. Dans le cas d'une résolution statique, les effets dus notamment à l'inertie ou à la viscoélasticité sont ignorés. Le système global est résolu grâce à une **méthode directe** comme par exemple en inversant la matrice K ou en utilisant une **méthode itérative** comme celle du **gradient conjugué**. Dans le cas d'une résolution dynamique, le système global est résolu pas à pas en utilisant un **schéma d'intégration numérique**. Notons que la formulation forte décrivant un problème n'est pas la même suivant si elle est définie en statique ou en dynamique. Le choix d'une résolution statique ou dynamique dépend de l'application visée. En effet, une résolution dynamique permet d'interagir au cours de la simulation, par exemple, en imposant une force sur une partie du maillage ou en changeant la topologie de ce dernier par une découpe. Enfin, il existe un troisième type de résolution : la **résolution quasi-statique**. Cette dernière est bien plus proche de la résolution statique que dynamique. En effet, elle permet l'obtention d'une suite d'états d'équilibre du système.

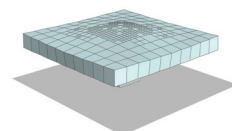
Pour une **résolution dynamique**, l'application de la méthode des éléments finis sur la formulation faible ou variationnelle du problème donne la formulation suivante :

$$\mathbf{F} = \mathbf{M}\ddot{\mathbf{U}} + \mathbf{D}\dot{\mathbf{U}} + \mathbf{K}\mathbf{U} \tag{2.1}$$

où M et D sont respectivement la **matrice de masse** et la matrice d'**amortissement**.

Notons que la propriété de linéarité est définie au niveau du matériau mais également au niveau de la géométrie de l'objet. La matrice de raideur K dépend des matrices B et D . La matrice B transcrit la relation entre les déplacements et les déformations. La propriété de linéarité géométrique de l'objet dépend de B . La matrice D correspond à la loi de comportement du matériau en liant les déformations aux contraintes. La propriété de linéarité du matériau dépend de D . Pour un système non-linéaire, B et/ou D dépendent du déplacement \mathbf{U} . K dépend alors de \mathbf{U} et change en fonction des déplacements.

Pour conclure, nous avons vu que la méthode des éléments finis permet de résoudre numériquement un problème de la mécanique des solides, se formalisant avec une équation aux dérivées partielles. Mais notons que cette méthode permet de résoudre une grande variété de problèmes d'ingénierie se formalisant avec une équation aux dérivées partielles comme, par exemple, la thermodynamique ou l'acoustique. Enfin, nous avons présenté comment le modèle physique fondé sur la méthode des éléments finis découle de la mécanique des milieux continus. Un tel modèle physique est qualifié de **modèle continu**. Un modèle continu implique la discrétisation d'une formulation continue fondée sur la mécanique des milieux continus.



2.1.4 Les schémas d'intégration

Les schémas d'intégration sont employés afin de résoudre dynamiquement le système (cf. l'équation 2.1) décrivant un objet à simuler. Ils ont pour objectif de calculer les nouvelles positions des particules discrétisant l'objet considéré, au pas de temps suivant. Nous nommons \mathbf{X}_t : l'état du système au temps t , et h : le pas de **temps d'intégration**. Notons que le pas de temps d'un schéma d'intégration est une valeur critique de ce dernier. En effet, si il est trop petit, la simulation ne variera que très peu à chaque itération du schéma d'intégration, c'est-à-dire que les temps de calcul seront alors trop longs. Par contre, si il est trop grand, la solution approximée par le schéma d'intégration s'éloignera beaucoup trop de la solution exacte, c'est-à-dire qu'elle divergera.

Les schémas d'intégration résolvent une **équation différentielle ordinaire**. Cette dernière est formalisée de la manière suivante :

$$\dot{\mathbf{X}} = \mathbf{F}(\mathbf{X})$$

où $\dot{\mathbf{X}}$ correspond à la dérivée de \mathbf{X} par rapport au temps, et \mathbf{F} est une fonction connue.

Si une solution analytique à l'équation différentielle ordinaire ne peut pas être déterminée, on procède à une intégration numérique de cette équation. Une intégration numérique consiste à calculer une variation de \mathbf{X} , notée $\Delta\mathbf{X}$, sur un intervalle de temps h , à partir de la valeur de \mathbf{X}_t .

Il existe deux types de schémas d'intégration, le schéma d'intégration **explicite** et le schéma d'intégration **implicite**. Le schéma d'intégration explicite est la méthode de résolution numérique la plus simple. D'un point de vue géométrique, ce schéma consiste à calculer la tangente à la solution de l'équation différentielle ordinaire, au temps t afin d'obtenir une approximation de cette dernière au temps $t + h$. La région de stabilité de ce schéma est moindre en comparaison du schéma d'intégration implicite (cf. l'**équation de test de Dahlquist** [Sch61]). De plus, la convergence du schéma d'intégration explicite nécessite des pas de temps petits au regard de ceux que peut employer le schéma d'intégration implicite, sans que ce dernier ne perde pour autant en stabilité [BW98]. Par contre, ce dernier implique une quantité de calculs et une complexité bien plus importante que le schéma explicite.

Dans le cas d'un schéma d'intégration explicite, $\dot{\mathbf{X}} = \frac{\mathbf{X}_{t+h} - \mathbf{X}_t}{h}$. En remplaçant $\dot{\mathbf{X}}$ par cette expression dans l'équation différentielle ordinaire au temps t , on obtient :

$$\mathbf{X}_{t+h} = \mathbf{X}_t + h\mathbf{F}(\mathbf{X}_t)$$

La valeur de \mathbf{X}_t au pas de temps suivant : \mathbf{X}_{t+h} , est obtenue en fonction de l'application de \mathbf{F} sur l'état du système au temps t . Contrairement au schéma d'intégration explicite, le schéma d'intégration implicite, calcule la valeur de \mathbf{X}_{t+h} en fonction de l'application de \mathbf{F} sur l'état du système, cette fois-ci, au temps $t + h$. Dans le cas d'un schéma d'intégration implicite, on obtient ainsi :

$$\mathbf{X}_{t+h} = \mathbf{X}_t + h\mathbf{F}(\mathbf{X}_{t+h})$$

Le calcul de cette expression nécessite la transformation de cette dernière en un système linéaire de type $Ax = b$ où A est une matrice, x est le vecteur à déterminer et b est un vecteur. Ce système est résolu par une méthode directe ou itérative comme celle du gradient conjugué. Les méthodes directes permettent d'obtenir une solution exacte en un nombre fini d'opérations. Contrairement aux méthodes directes, les méthodes itératives tentent de converger vers une solution la plus proche possible de la solution exacte. Si A est de très grande taille, une méthode itérative sera préférée car elle nécessitera moins de mémoire

qu'une méthode directe et pourra générer une solution en un nombre relativement petit d'itérations [Zar03].

Exemples

Il existe différents schémas d'intégrations explicites comme la méthode d'**Euler**, de **Störmer-Verlet** (appelée également **leapfrog**) ou encore de **Runge-Kutta d'ordre 2 et 4**.

Nous détaillons ci-après la méthode d'Euler et le schéma d'intégration implicite, pour une équation différentielle ordinaire du second ordre [Mar06] :

$$\ddot{\mathbf{X}} = \mathbf{F}(\mathbf{X}, \dot{\mathbf{X}})$$

Cette expression correspond à la célèbre loi de Newton : $\mathbf{F} = m\mathbf{A}$.

Nous considérons ici, \mathbf{X} comme étant la position d'une particule discrétisant l'objet à simuler, \mathbf{V} comme étant sa vitesse avec $\mathbf{V} = \dot{\mathbf{X}}$.

Cette équation différentielle ordinaire du second ordre amène au système d'équations différentielles du premier ordre suivant :

$$\begin{cases} \dot{\mathbf{X}} = \mathbf{V} \\ \dot{\mathbf{V}} = \mathbf{F}(\mathbf{X}, \mathbf{V}) \end{cases} \quad (2.2)$$

Pour la méthode d'Euler, ce système devient :

$$\begin{cases} \mathbf{X}_{t+h} = \mathbf{X}_t + h\mathbf{V}_t \\ \mathbf{V}_{t+h} = \mathbf{V}_t + h\mathbf{F}(\mathbf{V}_t, \mathbf{X}_t) \end{cases} \quad (2.3)$$

Pour un schéma d'intégration implicite, le schéma donné dans l'équation (2.2), devient :

$$\begin{cases} \mathbf{X}_{t+h} = \mathbf{X}_t + h\mathbf{V}_{t+h} \\ \mathbf{V}_{t+h} = \mathbf{V}_t + h\mathbf{F}(\mathbf{V}_{t+h}, \mathbf{X}_{t+h}) \end{cases} \quad (2.4)$$

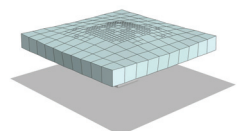
Notons que le schéma **symplectique** est un schéma d'intégration explicite où \mathbf{V}_{t+h} est déterminé de la même manière que pour la méthode d'Euler mais \mathbf{X}_{t+h} est calculé en fonction de \mathbf{V}_{t+h} et non en fonction de \mathbf{X}_t .

2.2 Modèles pour les objets déformables

Chaque modèle est caractérisé par sa **précision** (capacité à reproduire de manière réaliste le comportement de l'objet à simuler), sa **robustesse** (la stabilité de la simulation permise par le modèle), son **réalisme** (critère reposant sur le rendu visuel de la simulation proposée par le modèle) et son **interactivité** (permet de lier un retour haptique avec la simulation) ou d'autres critères dépendant de l'application visée. Nous pouvons également rajouter les propriétés liées à la facilité d'implémentation du modèle ou de parallélisation sur GPU ou encore sa capacité à gérer les changements topologiques. Enfin, notons que le principal enjeu d'une simulation visant le temps réel est de faire un compromis entre le critère de précision et d'interactivité de la simulation.

Deux catégories de modèle existent pour déformer des objets. La première catégorie rassemble tous les modèles dits « **non-physiques** » ou « **géométriques** ». Cette première catégorie est à l'origine de la seconde, groupant les modèles dits « **physiques** ».

Dans cette partie, nous introduisons brièvement les modèles de la première catégorie avant de nous attarder plus longuement sur les modèles physiques. En effet, nous avons fait le choix de l'emploi d'un modèle physique pour les travaux présentés dans ce manuscrit.



2.2.1 Modèles non-physiques ou géométriques

Les modèles non-physiques ou géométriques reposent sur la géométrie et non sur la physique de l'objet à simuler, comme l'indique leur dénomination. Nous prenons l'exemple du modèle des **FFD** appartenant à cette catégorie.

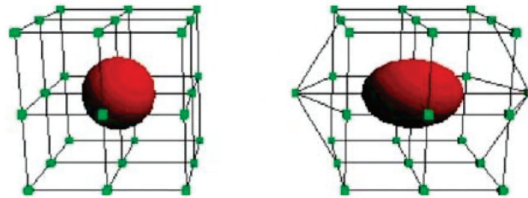


Figure 2.7 – Exemple du modèle FFD. A gauche : La grille de contrôle et l'objet englobé par cette dernière. A droite : L'objet englobé est déformé suite aux déplacements des points de la grille de contrôle [DN08].

FFD est l'acronyme pour **Free Form Deformation**. Sederberg *et al.* [SP86] introduisent ce modèle comme permettant la déformation locale ou globale d'un modèle géométrique. Le modèle FFD consiste en l'association des points appartenant à l'objet à déformer avec des points de contrôle mobiles (*cf.* les figures 2.7 et 2.8). Les points de l'objet sont déformés selon le déplacement de points de contrôle, en évaluant un **produit tensoriel trivariable du polynôme de Bernstein** (**trivariate tensor product Bernstein polynomial** en anglais). Chaque point de contrôle influence ainsi sur les points de l'objet en fonction d'un poids défini par le produit tensoriel trivariable du polynôme de Bernstein. Enfin, l'utilisateur peut définir la trajectoire des points de contrôle en fonction du temps et l'objet sera déformé en conséquence.

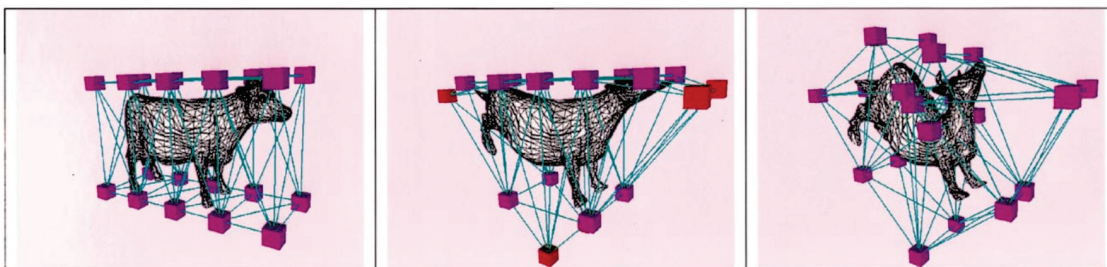


Figure 2.8 – Exemple du modèle FFD. La première figure : La grille de contrôle et une modélisation d'un objet représentant une vache, englobé par cette dernière. La deuxième et troisième figure : Le maillage représentant une vache est déformé suite aux déplacements des points de la grille de contrôle [HML00].

Afin que la déformation de l'objet semble réaliste, les modèles non-physiques doivent intégrer des contraintes, afin par exemple de conserver le volume de l'objet considéré [HML00].

2.2.2 Modèles physiques

L'inconvénient des modèles non-physiques est qu'ils reposent sur des lois de déformations empiriques. De plus, il est difficile d'intégrer les paramètres physiques des matériaux dans leur formulation. C'est pour cette raison que des modèles physiques, se fondant sur la théorie de la mécanique, sont alors apparus. En effet, nous avons vu précédemment que la mécanique des milieux continus permet de formaliser le comportement d'objets déformables. L'objectif

est d'analyser et de comprendre ce formalisme afin de le transcrire au niveau de l'ordinateur, au travers d'algorithmes, et plus généralement en élaborant des modèles physiques. Nous avons montré que la méthode des éléments finis permettait de résoudre les systèmes issus de la mécanique des milieux continus et plus précisément la mécanique du solide. Le modèle fondé sur la méthode des éléments finis est alors un modèle dit **physique**.

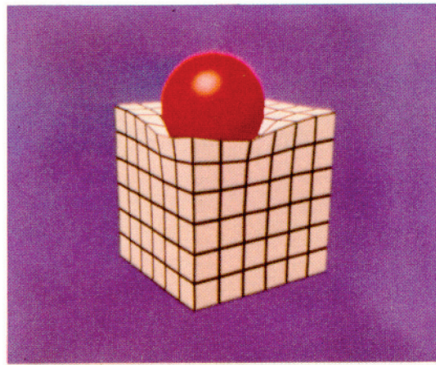


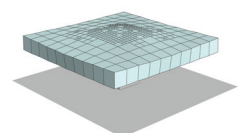
Figure 2.9 – Un des résultats obtenus avec le modèle physique établi par Terzopoulos [TPBF87].
Un objet rigide sphérique sur un objet déformable cubique [TPBF87].

Terzopoulos [TPBF87] est le premier à avoir élaboré, en informatique graphique, un modèle physique. Il a réussi à faire le lien entre le domaine de la mécanique formalisant les lois physiques de déformation, et l'informatique graphique (cf. la figure 2.9). Dans cet article, il s'oppose aux modèles (que nous nommons modèles non-physiques), qu'il définit comme passifs car les objets modélisés par ces derniers ne peuvent pas interagir entre eux ou réagir à des forces extérieures. Ils sont soumis à des trajectoires prédéfinies et une animation réaliste avec ces derniers nécessite un véritable savoir-faire. Les modèles qu'il définit comme actifs (que nous nommons modèles physiques), sont fondés sur des principes physiques et réagissent physiquement à des forces comme la gravité, les contraintes, les paramètres environnementaux comme la viscosité du milieu dans lequel se situe l'objet à simuler ou encore des obstacles. Le modèle physique qu'il propose consiste en une discrétisation en éléments finis de l'objet à simuler. Ce modèle sera ensuite déformé en s'appuyant sur une version simplifiée de la théorie de l'élasticité. Notons que les résultats obtenus étaient déjà très réalistes.

Dans le domaine de la simulation physique de nombreux travaux proposent des méthodes efficaces pour modéliser et simuler physiquement des objets déformables, comme en témoigne l'état de l'art de Nealen *et al.* [NMK⁺06]. Les modèles physiques sont classés en deux catégories, les modèles dits **continus** et les modèles dits **discrets**. Un modèle continu implique la discrétisation d'une formulation continue fondée sur la mécanique des milieux continus. Contrairement au modèle continu, un modèle discret implique la discrétisation du problème en lui-même.

Modèles continus

Le modèle physique fondé sur la méthode des éléments finis (abrégée par la suite **MEF**) que nous avons détaillé précédemment et le modèle physique de Terzopoulos présenté ci-dessus, sont des modèles dits continus. En effet, un modèle continu se définit comme reposant sur les équations de la mécanique des milieux continus. Ces dernières sont alors discrétisées au niveau des différents éléments constituant l'objet à simuler, afin de permettre sa simulation.



MEF Nous revenons ici plus en détails sur le modèle fondé sur la MEF. Ce modèle est le plus utilisé des modèles continus. Nous avons vu précédemment qu'il s'appuie sur la mécanique des milieux continus, pour simuler le comportement des objets déformables. Il intègre ainsi naturellement les propriétés physiques des matériaux à simuler. Il permet alors des simulations très précises et réalistes. Il est ainsi beaucoup utilisé dans le domaine médical pour la simulation d'actes chirurgicaux [CDC⁺96]. Par contre, son principal inconvénient concerne ses temps de calculs. En effet, ses calculs coûteux engendrent de faibles performances, nécessitant des optimisations afin de permettre son emploi pour des applications temps-réel.

D'autres modèles continus ont alors été élaborés pour pallier cette limitation d'interactivité, comme les modèles fondés sur la **méthode des différences finies** [TPBF87], la **méthode des volumes finis** [TBHF03], la **méthode des éléments frontières** [JP99] ou encore la méthode des **masses-tenseurs** [CDA99]. Mais ces dernières méthodes ne sont pas aussi précises ou robustes que celle de la MEF. Le choix d'un modèle continu plutôt qu'un autre, relève de l'application visée ou des propriétés du problème comme sa linéarité, la complexité des contours de l'objet ou encore la dimension du maillage [Mar06].

Les temps de calculs de la méthode des éléments finis sont d'autant plus importants que la taille du système à résoudre est grande. Dans ce sens, des travaux sont apparus permettant de rendre interactif cette méthode. Les méthodes d'optimisation pour des éléments finis linéaires avec une résolution quasi-statique, peuvent consister, par exemple, en des méthodes pré-calculant la **décomposition LU** de la matrice de raideur ou encore en des méthodes de **condensation**. Dans ce sens, les travaux de Cotin et Bro-Nielsen [BN96, BNC96, Cot97] distinguent les nœuds se trouvant sur la surface de l'objet, des nœuds internes à l'objet. La technique de condensation consiste à supprimer les nœuds internes du système d'équations linéaires. En effet, ils considèrent que seuls les nœuds à la surface présentent un intérêt. Plus précisément, seuls les nœuds surfaciques sont visibles et permettent de rendre compte des interactions avec l'objet virtuel. Ils obtiennent ainsi un système linéaire de taille moindre, tout en conservant le même comportement volumique. Notons que ces optimisations ne sont pas compatibles avec la réalisation de changements topologiques du maillage. Ces derniers seraient en effet trop coûteux pour être réalisés en temps réel, du fait de la reconstruction de la matrice de raideur suite à sa modification.

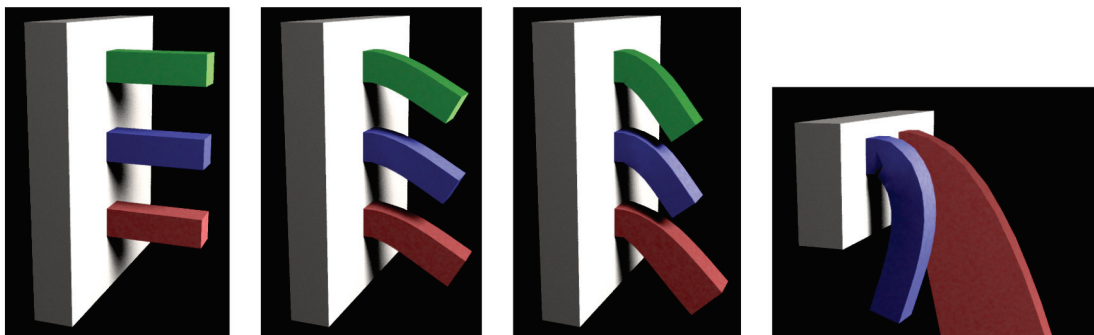


Figure 2.10 – Illustration du comportement d'un modèle éléments finis linéaires, n'étant pas invariant en rotation. La barre verte correspond à un comportement non-linéaire, la barre bleue à la méthode proposée par Müller et al. [MDM⁺ 02] et la barre rouge à un comportement linéaire. On observe le grossissement de la barre rouge qui est due au fait que les éléments finis linéaires ne sont pas invariants en rotation. Notons que le modèle de Müller et al. permet de gérer cette invariance en rotation inhérente à la linéarité du modèle en évitant l'emploi d'une méthode non-linéaire.

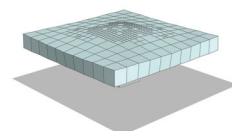
Nous avons décrit précédemment la non-linéarité géométrique et la non-linéarité au niveau du matériau. Un modèle éléments finis linéaire est préféré à un modèle non-linéaire. En

effet, la propriété de linéarité permet d'importantes simplifications du système. Par exemple, la raideur ne change pas au cours de la simulation. Nous précisons que ces raccourcis induisent des approximations du comportement de l'objet simulé. Le modèle linéaire permet des temps interactifs grâce aux nombreux travaux proposant des optimisations de calculs. Un modèle non-linéaire induit, quant à lui, une formulation bien plus complexe. De plus, cette non-linéarité implique que la matrice de raideur K est dépendante du déplacement \mathbf{U} . Dans ce cas, la raideur du système change en fonction des forces appliquées sur ce système. Dans le contexte d'une résolution dynamique, la matrice de raideur doit être recalculée à chaque itération entraînant un surcoût considérable. Par contre, il permet de simuler des matériaux ayant un comportement **hyperélastique** (élastique non linéaire) comme les organes. Il peut permettre également de grands déplacements et gère l'invariance en rotation. Cependant, il reste très coûteux et de nombreux travaux ont proposé des méthodes complexes d'optimisation. C'est pour cette raison que l'on préfère souvent améliorer le modèle linéaire plutôt que d'employer un modèle non linéaire. Par exemple, l'hypothèse de linéarité ne permet que de petits déplacements en rotation. La solution consiste en une méthode corotationnelle, que nous avons mentionnée précédemment. Elle permet de garder un système linéaire tout en gérant l'invariance en rotation (*cf.* la figure 2.10).

La MEF n'est pas adaptée pour gérer les changements topologiques de manière interactive. Nous décrivons plus précisément cet inconvénient. Une des étapes de la MEF consiste en l'assemblage de la matrice globale K . Plus précisément, cette étape nécessite le calcul préalable de l'ensemble des matrices pour chaque élément que l'on va ensuite placer dans la matrice globale. Lors d'un changement topologique, si l'on supprime un élément, il faut enlever tout ce qui relève de cet élément dans la matrice globale. Une ligne peut être supprimée et des coefficients peuvent être mis à 0. Mais selon les méthodes de résolution employées, il faut garantir que la matrice reste inversible ou qu'elle garde des caractéristiques comme par exemple la présence de bandes creuses, permettant d'employer des algorithmes spécifiques d'optimisation. Il faut alors stocker une correspondance entre un élément et toutes les parties lui correspondant dans la matrice. Lors de changements topologiques comme un raffinement local, des éléments sont rajoutés au système. Ce rajout peut nécessiter une reconstruction de la matrice. En conclusion, des problématiques de changements topologiques nécessitent au mieux de stocker de nombreuses informations relatives à la position des informations relatives à un élément et au pire une reconstruction de la matrice, qui est au final souvent nécessaire. Nous rappelons que cette reconstruction est une étape coûteuse et ne permet pas de garantir l'interactivité de la simulation.

XFEM Afin de gérer les changements topologiques, une extension de la méthode des éléments finis appelée **XFEM** a été proposée par Jeřábková *et al.* [JK09]. Cette extension permet la découpe du maillage en modélisant les discontinuités, sans avoir besoin d'effectuer un remaillage de l'objet, évitant ainsi un surcoût conséquent. Les fonctions de forme utilisées dans la version classique de la MEF, se voient rajouter un terme additionnel permettant de gérer les discontinuités. La méthode présentée dans cet article en constitue une première approche.

La méthode des masses-tenseurs Cotin, Delingette *et al.* [CDA99, Cot97] définissent la méthode des masses-tenseurs afin de pallier les problèmes de gestion des changements topologiques du modèle fondé sur la MEF. Dans cet article, ils présentent l'optimisation d'un modèle des éléments finis linéaire en résolution quasi-statique, en pré-calculant une partie des déformations du modèle. Cependant, l'approche présentée interdit le changement



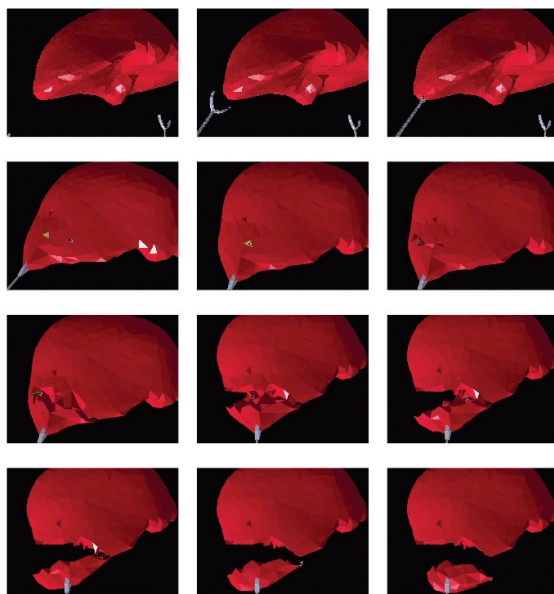


Figure 2.11 – Exemple de simulation de découpe du foie. Modèle hybride employant un modèle éléments finis linéaire et un modèle masses-tenseurs, permettant la simulation interactive de la découpe d'un foie [CDA99].

de la topologie du maillage en temps réel. En effet, un changement topologique induit la modification de la matrice de raideur et nécessite de recommencer la phase de pré-calcul. Notons que cette phase peut prendre entre quelques minutes et plusieurs heures. De plus, sans cette phase, il n'est pas possible de résoudre correctement le système et en temps-réel. Ils définissent alors une alternative à la MEF, la méthode des masses-tenseurs. Enfin, ils combinent leur modèle des éléments finis linéaire pré-calculé avec un modèle masses-tenseurs, en un modèle alors hybride (cf. la figure 2.11).

Nous détaillons ici plus précisément la méthode des masses-tenseurs (abrégée **MMT**). Cette méthode est définie comme étant un modèle continu s'inspirant du **modèle fondé sur le système masses-ressorts**. Elle est très proche de la MEF du fait qu'elle s'appuie sur les équations de la mécanique des milieux continus. Par contre, elle ne nécessite pas l'assemblage de la matrice globale de rigidité K et le calcul des forces se fait au niveau des nœuds du maillage comme pour le système masses-ressorts. Nous rappelons que pour la MEF, le calcul des forces est réalisé au niveau des éléments. Dans l'article de Cotin *et al.* [CDA99], ils commencent par discrétiser l'objet à simuler en tétraèdres. Puis, ils définissent les fonctions de forme permettant d'interpoler le déplacement de tout point du tétraèdre selon les déplacements des quatre nœuds de ce dernier. Notons que la masse de l'objet est concentrée sur les nœuds du maillage, tout comme les effets d'amortissement, selon la technique du **mass lumping**. Cette technique consiste à rendre diagonale la matrice de masses. Les masses des différents nœuds du maillage se situent ainsi sur la diagonale de cette matrice. Dans la MEF, les matrices de masse et d'amortissement sont creuses alors que dans la MMT, elles sont diagonales. Cette simplification permet à chaque instant t , de calculer les forces élastiques exercées sur chacun des nœuds du maillage. Puis, pour chaque sommet, les équations du mouvement sont intégrées pour trouver la nouvelle position du sommet à l'instant $t + 1$ [Pic01]. Plus précisément, la force élastique exercée sur un nœud i du maillage, est formalisée par les tenseurs de rigidité des éléments incidents à i et des tenseurs de rigidité des nœuds voisins. Lors d'un changement topologique, il suffit alors de mettre à jour les contributions des éléments incidents et celles des nœuds voisins.

Modèles discrets

Un modèle discret est un modèle physique car il découle des lois de la physique. Tout comme un modèle continu, un modèle discret implique la discrétisation spatiale de l'objet à simuler. Par contre, les lois physiques décrivant la déformation des objets, sont également discrétisées.

Un **système de particules** est un modèle discret, où l'objet à simuler a été discrétisé en un ensemble de particules. Chaque particule interagit avec les particules de son voisinage, par exemple, en définissant un ressort linéaire entre chacune d'entre elles, ou selon le potentiel de **Lennard-Jones**. Ce potentiel prend en paramètre la distance à l'équilibre entre deux particules, et la distance courante. Si deux particules sont trop proches, ce potentiel les éloigne l'une de l'autre en appliquant une force de répulsion. Si par contre, elles sont trop éloignées l'une de l'autre, elles subissent une force d'attraction. Notons que la force de répulsion est beaucoup plus importante que celle d'attraction. Nous pouvons retrouver des similarités entre ce potentiel et l'interaction forte au niveau des atomes.

Le modèle discret de type **système masses-ressorts** (abrégé par la suite **SMR**) est une spécification du système de particules. Dans nos travaux, nous avons fait le choix du SMR pour simuler le comportement des objets déformables. C'est pour cette raison que nous le détaillons ci-dessous, de manière plus précise.

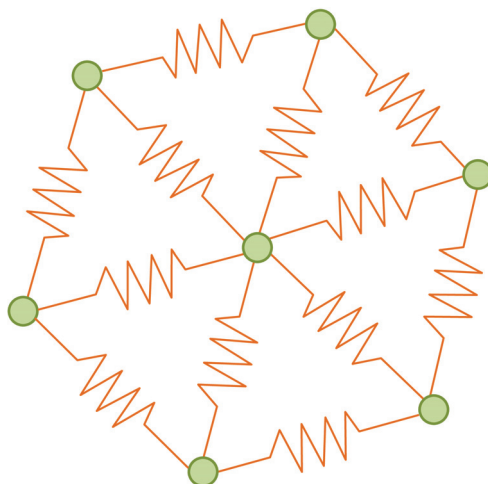
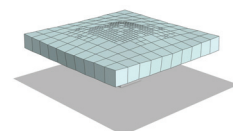


Figure 2.12 – Exemple de SMR. Un SMR 2D où les particules sont représentées en vert et les ressorts en orange.

SMR Définition : Le SMR consiste en la discrétisation d'un objet réel en un ensemble de **particules** aussi appelées **masses**, connectées ensemble par des **ressorts** (cf. la figure 2.12). Chaque particule est ainsi influencée par son voisinage proche. Notons que les ressorts suivent généralement la topologie du maillage.

Le SMR est apprécié du fait de son implémentation intuitive et de son faible coût en temps de calcul. Son interactivité lui vaut d'être employé pour des applications nécessitant d'être en temps-réel. Il est par exemple très utilisé dans les jeux vidéos. De plus, il est également plus approprié pour l'implémentation sur GPU ou encore la gestion des changements topologiques dans le maillage, que la MEF. Enfin, il gère facilement les larges déformations en comparaison à la MEF linéaire. Par contre, il ne permet pas des simulations aussi précises que celles générées par la MEF. Une des raisons est que, contrairement à cette dernière, il n'intègre



pas naturellement les paramètres physiques des matériaux dans sa formulation. De plus, il est difficile de paramétrer un SMR en fonction de ces paramètres. Le choix des coefficients de raideur des ressorts du SMR sont alors souvent attribués de manière arbitraire. Enfin, le comportement de la simulation du modèle représentant l'objet à simuler est dépendant du maillage en lui-même, c'est-à-dire de sa résolution et de sa topologie. Pour conclure, son manque de précision le dessert pour être employé dans des applications nécessitant une exactitude du comportement, comme par exemple les planifications chirurgicales.

Nous décrivons ici de manière plus précise le pipeline du SMR, de sa construction à sa résolution.

- Comme nous l'avons spécifié précédemment, il consiste tout d'abord en la discrétisation en particules et ressorts de l'objet à simuler. Cette étape permet d'obtenir un maillage surfacique ou volumique de l'objet à simuler.
- Puis, la masse de l'objet est répartie sur l'ensemble des particules du système, de manière uniforme ou non.
- Les coefficients de raideur de l'ensemble des ressorts du système sont initialisés. Nous rappelons que la détermination de ces coefficients est une étape difficile, qui a suscité de nombreux travaux.
- Une fois le modèle établi, on passe à l'étape de résolution.
 - A chaque itération, les forces dues aux ressorts sont calculées au temps $t + h$, à partir de l'état de leurs extrémités au temps t . Pour chaque ressort, la force ainsi obtenue est accumulée au niveau des particules se trouvant aux extrémités du ressort en question.

Nous considérons un ressort ayant pour extrémités les particules i et j . Ces dernières ont respectivement comme masse m_i et m_j , comme position au temps t : $\mathbf{P}_i(t)$ et $\mathbf{P}_j(t)$, comme vitesse au temps t , $\mathbf{V}_i(t)$ et $\mathbf{V}_j(t)$ et comme accélération au temps t , $\mathbf{A}_i(t)$ et $\mathbf{A}_j(t)$. Le coefficient de raideur de ce ressort est k_{ij} et sa distance au repos entre ses deux extrémités est l_{ij} . Enfin, le coefficient d'amortissement de ce ressort est γ_{ij} .

La force du ressort est composée d'une partie élastique et d'une partie correspondant à l'amortissement du ressort. La force du ressort \mathbf{F}_{ij} au temps t , est formulée de la manière suivante :

$$\mathbf{F}_{ij}(t) = \mathbf{F}_{ij}^e(t) + \mathbf{F}_{ij}^v(t) \quad (2.5)$$

où \mathbf{F}_{ij}^e est la partie correspondant à la force élastique du ressort, et \mathbf{F}_{ij}^v est la partie représentant la force d'amortissement du ressort.

La force élastique du ressort est le plus souvent linéaire, elle est ainsi proportionnelle à la variation de longueur entre ses deux extrémités. La formulation de la force élastique du ressort est alors la suivante :

$$\begin{cases} \mathbf{F}_{ij}^e(t) &= k_{ij} (d_{ij} - l_{ij}) \mathbf{U}_{ij}(t) \\ \mathbf{F}_{ji}^e(t) &= -\mathbf{F}_{ij}^e(t) \end{cases}$$

où $d_{ij} = \|\mathbf{P}_j(t) - \mathbf{P}_i(t)\|$, et $\mathbf{U}_{ij}(t)$ est le vecteur de direction normalisé entre les deux particules i et j . Ce dernier est formalisé par :

$$\mathbf{U}_{ij}(t) = \frac{\mathbf{P}_j(t) - \mathbf{P}_i(t)}{\|\mathbf{P}_j(t) - \mathbf{P}_i(t)\|}$$

Nous précisons que \mathbf{F}_{ij}^e correspond à la force du ressort appliquée sur son extrémité i et \mathbf{F}_{ji}^e , à la force du ressort appliquée sur son autre extrémité j . En effet, un ressort vérifie le principe d'action-réaction, inférant que la somme des forces induites par un ressort doit être nulle. Enfin, notons qu'il existe d'autres formulations de la force élastique d'un ressort. Nous citons par exemple les travaux de Teschner *et al.* [TGG00], proposant une formulation permettant d'adapter le SMR à la simulation d'objets déformables ayant une loi de comportement non-linéaire.

La partie de la force du ressort correspondant à son amortissement permet de simuler l'énergie dissipée lors de la déformation du ressort. En effet, elle permet de prendre en compte le fait que les objets simulés ne sont pas parfaitement élastiques [NNUF04]. La force d'amortissement du ressort est définie par :

$$\mathbf{F}_{ij}^v(t) = \gamma_{ij} [(\mathbf{V}_j(t) - \mathbf{V}_i(t)) \cdot \mathbf{U}_{ij}(t)] \mathbf{U}_{ij}(t)$$

Notons que cette force d'amortissement a pour direction celle du ressort. De plus, le coefficient d'amortissement du ressort peut être défini de la manière suivante [BL05] :

$$\gamma_{ij} = 2\sqrt{\frac{m_i + m_j}{2}} k_{ij}$$

En conclusion, nous précisons que la représentation des forces est simple et le calcul de ces dernières en chaque sommet, permettent une résolution très rapide.

- Suite à l'accumulation des forces induites par les ressorts au niveau des particules, nous procédons de même avec les forces externes comme la gravité ou les interactions avec l'utilisateur.
- Puis, le mouvement de chaque particule i est déterminé en fonction de la deuxième loi de Newton. En effet, son accélération au temps t , est obtenue grâce à cette loi :

$$m_i \frac{d^2}{dt^2} \mathbf{P}_i(t) = \mathbf{F}_i(t) \quad (2.6)$$

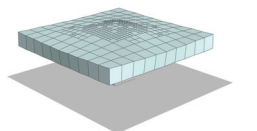
où \mathbf{F}_i est la somme des forces appliquées à la particule i . Comme indiqué précédemment, \mathbf{F}_i inclut l'ensemble des forces des ressorts ayant pour extrémité i , la gravité, ainsi que des éventuelles forces d'interactions avec l'utilisateur ou encore celles issues de collisions avec d'autres objets de type déformable ou rigide.

- Les nouvelles vitesse et position sont ensuite déterminées en fonction de l'accélération grâce à l'emploi d'un schéma d'intégration explicite ou implicite (*cf.* la sous-section 2.1.4). Notons que la résolution d'un tel système oblige l'emploi d'un schéma d'intégration. Nous considérons un schéma d'intégration explicite symplectique. La vitesse de la particule i au temps $t + h$ est ainsi obtenue en fonction de l'accélération de cette dernière au temps t , grâce à la formulation suivante :

$$\mathbf{V}_i(t + h) = \mathbf{V}_i(t) + h\mathbf{A}_i(t) \quad (2.7)$$

- Enfin, la position de i au temps $t + h$ est calculée en fonction de sa vitesse au temps $t + h$, conformément à la définition du schéma symplectique.

$$\mathbf{X}_i(t + h) = \mathbf{X}_i(t) + h\mathbf{V}_i(t + h) \quad (2.8)$$



Pour finir, nous avons vu précédemment comment se définissait un SMR. Cependant, la notion de ressort peut être généralisée. En effet, des énergies de déformation peuvent être employées afin de modéliser les forces élastiques comme dans les travaux [BHW94, BW98, THMG04]. Cette énergie est minimale quand l'objet simulé est au repos. Elle agit pour la préservation des distances, angles, aires ou encore volumes. Notons que la notion de force élastique présentée permettait seulement d'agir sur la conservation de la distance. La force appliquée à une particule est alors calculée en dérivant cette énergie par rapport à sa position.

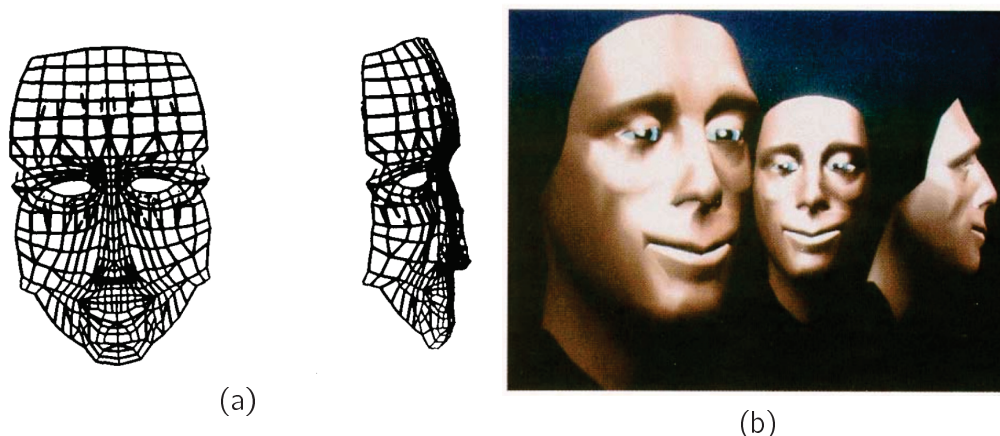


Figure 2.13 – Exemple des premiers travaux employant un SMR pour l'animation faciale. (a) Résultats obtenus par Platt et al. [PB81] correspondant à une modélisation faciale. (b) Résultats obtenus par Waters et al. [Wat87] correspondant à l'expression faciale du bonheur.

Applications et améliorations Le SMR est un modèle physique très usité en animation. Les premiers travaux employant un SMR datent de 1981. Ils avaient pour vocation l'animation faciale. Nous citons les travaux de Platt, Waters *et al.* [PB81, Wat87] qui ont proposé un modèle du visage, pouvant être animé en résolution statique et permettant de réaliser différentes expressions du visage (*cf.* la figure 2.13). Puis, des travaux sont apparus permettant de modéliser, avec une résolution dynamique, les différents niveaux de la peau : les tissus cutanés, les tissus sub-cutanés et les muscles, ou encore la peau, la graisse et les muscles [CHP89, TW90, WT91, LTW95, KHS01]. Les SMR ont également été utilisés pour simuler le déplacement de serpents, de vers ou encore de poissons [Mil88, TT94]. De plus, ils ont également été employés pour réaliser des simulations de chirurgie [CEO⁺93, KKH⁺97, Kü00, MSVCS04, ZGHX05]. Les SMR sont plus adaptés à la simulation de tissus, que les modèles fondés sur la MEF [BHW94, BFA02, BW98] (*cf.* la figure 2.14(a)). Enfin, ils se prêtent très bien à la simulation de cheveux [SLF08] (*cf.* la figure 2.14(b)).

Provot [Pro95] propose une amélioration du SMR classique permettant la simulation du comportement non-élastique des tissus. Lorsqu'un tissu est suspendu en fixant un nombre réduit de particules le composant, la forte concentration de contraintes entraîne une déformation non réaliste. Une solution consiste alors à augmenter la raideur des ressorts. Cependant, cette solution entraîne une diminution drastique des performances. En effet, dans la formulation du pas de temps, ce dernier croît proportionnellement à la raideur [Bat82]. Provot propose alors de contraindre les ressorts à une longueur maximale de déformation.

Un ressort est un élément 1D, de ce fait, il n'est pas particulièrement adapté à la simulation d'objets volumiques. Dans ce sens, Promayon *et al.* [PBP96, Pro97] proposent une méthode à « mémoire de forme » permettant de contraindre le volume d'un objet déformable comme constant au cours de la simulation. L'objet déformable est décrit seulement par son

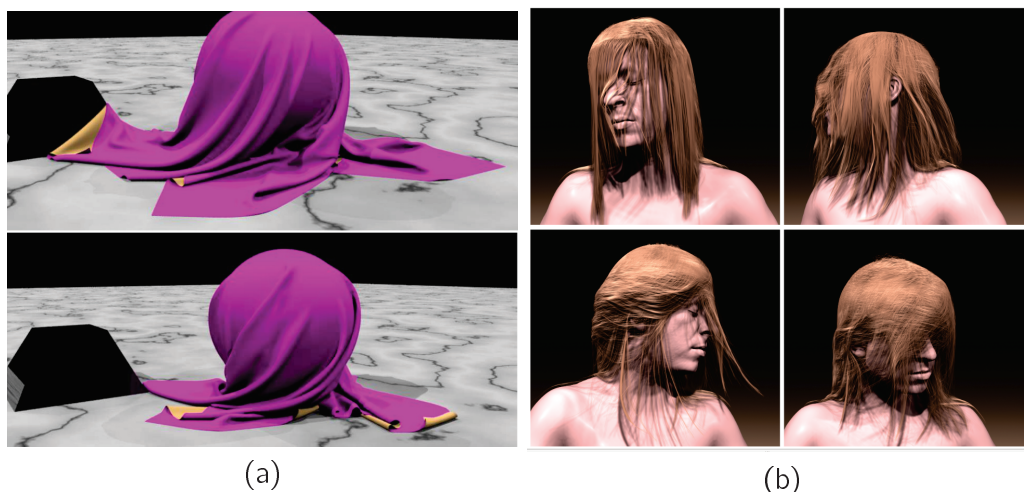


Figure 2.14 – Exemple de l'utilisation d'un SMR pour la simulation de tissu. (a) Résultats des travaux de Bridson et al. [BFA02]. Simulation des frictions entre une sphère rotative et un morceau de tissu s'enroulant autour, créant une structure complexe de plis. (b) Résultats des travaux de Selle et al. [SLF08]. Simulation de dix mille cheveux raides composés de cinquante segments chacun, soit un million de particules, fixés sur la tête d'un personnage tournant la tête de droite à gauche.

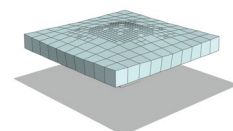
contour, composé d'un ensemble de particules. Cette méthode permet ainsi de modéliser des objets incompressibles comme par exemple l'abdomen. Cette méthode a notamment été employée dans les travaux de Marchal [Mar06] œuvrant pour la simulation de procédures médicales pour le diagnostic et le traitement du cancer de la prostate. Un modèle physique discret est employé avec une formulation reposant sur un principe de mémoire de forme. Enfin, Marchal a proposé une démarche expérimentale afin de valider sa méthode et la comparer aux autres méthodes de simulation d'objets déformables ou encore avec des données réelles.

Enfin, Liu *et al.* [LBOK13] proposent une méthode permettant d'accélérer la résolution numérique d'un SMR avec un schéma d'intégration implicite. Notons que cette méthode s'applique pour un SMR ayant des ressorts gouvernés par la loi de comportement de Hooke, c'est-à-dire que ces derniers sont linéaires. Leur méthode permet ainsi une grande valeur de pas de temps.

Paramétrisation

Comme indiqué précédemment, la précision en terme de comportement physique reste le point faible des SMR. Un SMR n'inclut pas naturellement les paramètres physiques des matériaux contrairement aux modèles fondés sur la MEF. Il est donc difficile de reproduire le comportement réel d'un matériau donné, même si les SMR permettent de produire des animations visuellement convaincantes. Cependant, des méthodes tentent de lier les coefficients de raideur des ressorts aux propriétés mécaniques de l'objet à simuler afin d'améliorer la précision du SMR.

Nous abordons maintenant la difficulté de la paramétrisation d'un SMR (*cf.* la figure 2.15), comme la détermination des masses, des coefficients de raideur ou d'amortissement des ressorts ou encore la configuration topologique du maillage. Dans ce sens, quelques travaux ont proposé des méthodes afin de transcrire les paramètres physiques des matériaux au niveau du SMR. Ces dernières peuvent être classées en deux catégories, « **les méthodes orientées données** » et « **les méthodes analytiques** » [LSH07]. La première catégorie inclut l'ensemble des méthodes qui déduisent les paramètres du SMR en se calquant sur des données réelles ou venant de modèles fondés sur la MEF. La deuxième catégorie regroupe les approches formalisant des expressions analytiques, qui permettront de calculer les coefficients de raideur. Le



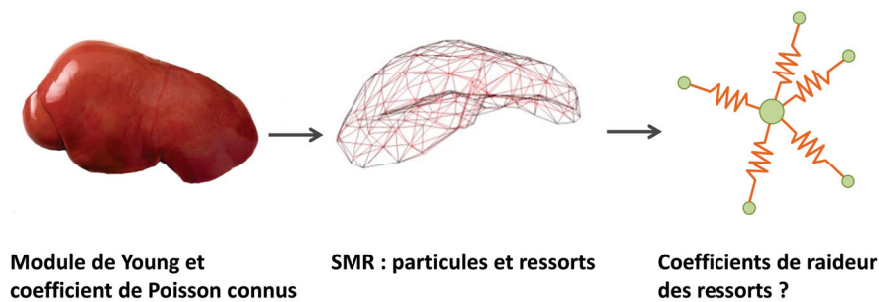


Figure 2.15 – Schéma explicitant la paramétrisation du SMR en fonction des paramètres physiques des matériaux. De gauche à droite : Un objet réel caractérisé par un module de Young et un coefficient de Poisson, La discrétisation de cet objet en un maillage où les nœuds sont les particules du SMR et les liens sont les ressorts du SMR et La question du paramétrage du SMR en fonction des paramètres physiques des matériaux.

SMR employé dans les travaux de ce manuscrit est paramétré par une méthode appartenant à cette deuxième catégorie.

- **Les méthodes orientées données.** Les paramètres du SMR sont souvent déterminés de manière empirique, en essayant d’approcher un comportement visuellement réaliste, en observant des objets réels. Des méthodes plus élaborées, estiment les paramètres optimaux d’un SMR, en essayant de minimiser l’erreur obtenue entre la simulation réalisée avec le SMR et celle d’un modèle de référence. Ce modèle de référence peut être fondé sur la MEF ou encore consister en des mesures de déformations effectuées sur des objets réels. Ces méthodes utilisent des techniques comme le **recuit simulé** ou l’**algorithme génétique**.

Dans ce sens, Deussen *et al.* [DKT95] proposent de disposer de manière homogène les particules constituant le SRM. Pour cela, ils emploient un algorithme itératif s’appuyant sur un diagramme de **Voronoi** où les particules sont placées à chaque itération sur les centres de gravité de ce diagramme. Nous rappelons qu’un diagramme de Voronoi consiste en un partitionnement de l’espace. Il est construit à partir d’un échantillonnage de la surface considérée, où chaque cellule d’un échantillon est définie par l’ensemble des points qui sont plus proches de l’échantillon considéré que tout autre échantillon. Puis, ils proposent une méthode permettant d’assigner la masse des particules en s’appuyant sur les moments d’inertie. Ils proposent une **triangulation de Delaunay** (dual du diagramme de Voronoi) à partir des positions des particules afin de définir les ressorts du SMR. Les coefficients de raideur des ressorts sont ensuite déterminés en comparant les déplacements des particules du SMR avec plusieurs tests simples de référence possédant une solution analytique. Un algorithme de recuit-simulé est alors employé comme méthode d’optimisation pour minimiser la différence entre les déplacements comparés, en modifiant à chaque itération les coefficients de raideur des ressorts.

Bianchi *et al.* [BSSH03, BSSH04] tentent d’approximer le comportement d’un modèle de référence, en ajustant les paramètres du SMR. Leur méthode consiste à définir simultanément les coefficients de raideur des ressorts et la topologie du maillage. Pour cela, ils définissent une fonction de coût correspondant à la mesure de la différence entre le comportement du modèle de référence et celui du modèle apprenant. Elle correspond au calcul d’une distance euclidienne entre la position d’un point du modèle

apprenant et son correspondant au niveau du modèle référent. Ils utilisent alors un **algorithme d'optimisation génétique**. Cet algorithme débute avec différentes configurations topologiques, puis différents événements surviennent comme la « sélection », le « brassage génétique » ou encore la « mutation », jusqu'à la convergence vers une configuration topologique. Notons qu'il peut converger plus ou moins rapidement vers les paramètres optimaux du SMR dans le meilleur des cas, ou vers des paramètres sous-optimaux, ou encore dans le pire des cas, ne pas converger du tout. Enfin, ils montrent également que simuler le comportement d'un matériau élastique linéaire avec un SMR n'ayant qu'une unique valeur de coefficient de raideur, donne de mauvais résultats.

Zerbato *et al.* [ZGF07] orientent leur paramétrisation pour la simulation d'organes. En effet, leur méthode s'appuie sur les informations extraites d'images DICOM issues de scanner CT. Elles représentent les densités locales de l'organe à simuler. Ces dernières permettent ainsi de définir la masse de chaque particule du maillage sur laquelle repose le SMR. Un algorithme génétique est ensuite employé afin de déterminer les coefficients de raideur des ressorts du système en minimisant les différences entre les mesures de déformations faites sur des tissus organiques réels lors de l'application de contraintes, et le comportement du modèle virtuel soumis aux mêmes forces. Pour initialiser cet algorithme, les coefficients de raideur et d'amortissement des ressorts sont définis en fonction du module de Young de l'organe simulé, en utilisant la formulation proposée par Van Gelder [Van98].

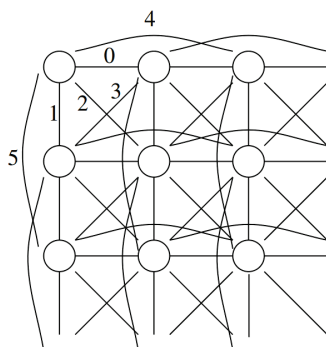
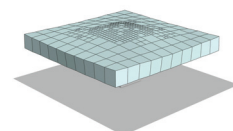


Figure 2.16 – Exemple de SMR. Le modèle SMR employé par Louchet *et al.* [LPC95] pour simuler des tissus élastiques non-linéaires.

Louchet *et al.* [LPC95] utilisent également un algorithme génétique où ils se restreignent à cinq paramètres qui sont spécifiques à la simulation de tissus (*cf.* la figure 2.16). En effet, il existe différents types de ressorts pour simuler un tissu, comme les ressorts « structurels » se positionnant sur les arêtes d'un quadrilatère (*cf.* les ressorts 0 et 1 de la figure 2.16), les ressorts de « cisaillement » (*cf.* les ressorts 2 et 3) correspondant aux ressorts diagonaux internes au quadrilatère, ou encore les ressorts de « flexion » (*cf.* les ressorts 4 et 5).

Pour conclure, ces méthodes orientées données ne sont pas adaptées pour des maillages de grande taille car le nombre de paramètres pour l'algorithme d'optimisation ne doit pas être trop important. De plus, la paramétrisation obtenue est dépendante d'un maillage et de tests qui peuvent être de type étirement, cisaillement, compression, ... En effet, il est difficile d'obtenir une paramétrisation unique pour un ensemble de tests ou même de considérer tous les tests. Du coup, la paramétrisation doit être recommencée à chaque nouveau maillage ou encore à chaque nouveau test. Ces méthodes sont alors réhivitoires pour la simulation de changements topologiques.



— **Les méthodes analytiques.**

Quelques travaux cherchent l'obtention d'une formulation analytique des coefficients de raideur d'un SMR en s'appuyant sur la mécanique des milieux continus ou encore en comparant la matrice de raideur du SMR avec celle de la MEF.

Van Gelder [Van98] définit une formulation reliant les raideurs des ressorts d'un maillage triangulaire aux paramètres rhéologiques du matériau de l'objet simulé. Il se repose sur les lois de la mécanique des milieux continus pour un matériau élastique isotrope. Pour le cas 2D, il obtient la formulation du coefficient de raideur d'une arête de longueur au repos c . Cette arête est incidente aux deux triangles T_1 et T_2 . T_i ($i \in \{1, 2\}$) a pour aire $|T_i|$ et se compose des arêtes c , a_i , b_i . La formulation du coefficient de raideur est la suivante :

$$k_c = \sum_{i=1}^n \frac{E}{1+\nu} \frac{|T_i|}{c^2} + \frac{E\nu}{1-\nu^2} \frac{a_i^2 + b_i^2 - c^2}{8|T_i|}, \quad (2.9)$$

où E est le module de Young, et ν est le coefficient de Poisson de l'objet à simuler.

En 3D, il procède de la même manière en considérant alors le volume des tétraèdres contenant l'arête dont il souhaite calculer le coefficient de raideur. Il a par ailleurs également montré qu'assigner le même coefficient de raideur à tous les ressorts donne de très mauvais résultats pour simuler un objet élastique et homogène.

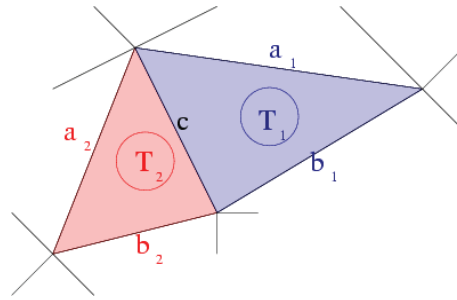


Figure 2.17 – Illustration de la méthode de Van Gelder. Modèle 2D constitué de deux triangles incidents à la même arête de longueur c dont on souhaite formuler le coefficient de raideur [Van98].

Pendant, cette formulation est très restrictive sur le choix du coefficient de Poisson puisque ce dernier est fixé à nul. De plus, Baudet [Bau06, BBJ⁺09b] montre que le modèle de Van Gelder, bien qu'étant largement utilisé, est erroné. Pour cela, il démontre de manière théorique et expérimentale que ce dernier ne permet pas d'obtenir des animations en cohérence avec le coefficient de Poisson et le module de Young imposés. En effet, le module de Young mesuré lors de la stabilisation d'une expérience de traction dévie de 25% par rapport au module de Young imposé.

Par contre, Lloyd *et al.* [LSH07] montrent que pour certains cas particuliers, il est possible de retrouver les coefficients de raideur des ressorts en fonction des paramètres issus de la mécanique des milieux continus, permettant une équivalence avec la MEF. Pour cela, ils comparent la matrice de raideur du modèle fondé sur la MEF et celle du SMR. Cette comparaison aboutit à l'obtention d'un système d'équations, dont la résolution permet d'obtenir les coefficients de raideur des ressorts. Par contre, pour faciliter la résolution de ce système, ils fixent $\nu = 0.3$, valeur pour laquelle l'erreur moyenne est la plus faible. Notons que dans le cas d'un maillage quadrangulaire, un degré de liberté doit être rajouté afin d'augmenter le nombre de paramètres du système d'équations, permettant ainsi d'obtenir une solution exacte à ce système. Le degré

de liberté additionnel est permis par le pré-étirement des ressorts induisant que la longueur initiale de ces derniers dépend d'une variable qui ajoute ainsi un degré de liberté. Dans le cas d'un maillage tétraédrique, différentes solutions sont mises en place pour améliorer la similarité entre les SMR et les MEF, comme le fait que le coefficient de Poisson soit fixé à 0.25, l'emploi d'une fonction de minimisation pour la comparaison des matrices de raideurs, ou encore l'ajout d'une force permettant de conserver le volume de l'objet simulé afin de pouvoir augmenter le degré de liberté du système d'équations correspondant.

Delingette [Del08] établit une connexion formelle entre les paramètres de raideur et la mécanique des milieux continus pour des objets élastiques 1D et 2D de type membrane, en introduisant des ressorts bi-quadratiques.

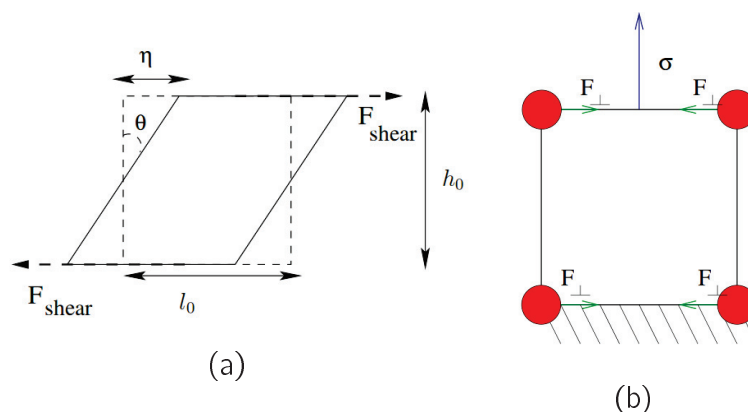
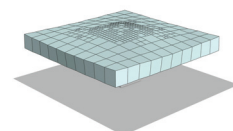


Figure 2.18 – Schémas expliquant la méthode de Baudet et al. [BBJ⁺09b], permettant la paramétrisation d'un SMR en fonction des paramètres physiques des matériaux que sont le module de Young et le coefficient de Poisson. (a) Le test de cisaillement. (b) Les forces de correction.

Baudet et al. [BBJ⁺09b] proposent la paramétrisation des coefficients de raideur des ressorts à partir des paramètres physiques des matériaux : E (module de Young) et ν (coefficient de Poisson), en s'appuyant sur le lagrangien du SMR. Leur méthode s'applique uniquement à des rectangles en 2D et des pavés en 3D car elle se repose sur la symétrie de leur forme. Cette méthode est décrite de manière plus précise que les autres car nous l'employons dans les travaux présentés dans ce manuscrit. Leur méthode est la suivante :

1. En 2D, ils considèrent un rectangle de dimension $l_0 \times h_0$, composé de quatre ressorts de raideur k_{l_0} et k_{h_0} et deux ressorts diagonaux internes au rectangle, de raideur k_d .
2. Afin de déterminer k_d , ils effectuent un test de cisaillement consistant à appliquer deux forces opposées sur deux arêtes parallèles du rectangle (cf. F_{shear} dans la figure 2.18(a)). La déformation du rectangle due au cisaillement entraîne une déviation d'un angle θ et d'un déplacement η .
3. Ils formalisent alors la déformation des ressorts diagonaux, puis posent le lagrangien relatif au test de cisaillement. Ce dernier correspond à la somme des énergies potentielles. Notons que les énergies potentielles des ressorts associés aux arêtes l_0 et h_0 sont négligées, en raison de l'hypothèse d'un angle de cisaillement presque nul. Cette hypothèse est indispensable à la construction du système d'équations qui aboutira à la formulation des raideurs.



4. Le lagrangien est ensuite minimisé en fonction du déplacement η , permettant d'obtenir la formulation de ce déplacement.
5. Pour finir, ils déduisent la formulation de k_d en fonction de E , de ν , des formulations de η et du module de cisaillement qui est fonction de E et ν .
6. Ils déterminent ensuite les autres ressorts : k_{l_0} et k_{h_0} en effectuant cette fois-ci deux tests d'étirement respectivement latéral et longitudinal.
7. Cependant, le système d'équations issu de ces deux tests est sur-contraint, restreignant alors la solution pour un coefficient de Poisson égal à 0.3. Ils ajoutent ainsi des forces de correction (cf. F_{\perp} dans la figure 2.18(b)) qui sont orthogonales à la force d'étirement (cf. σ dans la figure 2.18(b)). Notons que les forces de correction peuvent s'interpréter comme une compensation de l'effet induit par ν .
8. Ils posent ensuite de la même manière que précédemment, le lagrangien des tests d'étirement incluant l'énergie potentielle relative aux forces de correction ainsi que le coefficient de raideur k_d calculé en amont.
9. Le test d'étirement induit une élongation de η dans la direction de la force d'étirement et un rétrécissement de 2σ dans la direction orthogonale à cette force. Le lagrangien est ainsi minimisé en fonction de ses deux déformations, permettant leur formulation.
10. Les raideurs k_{l_0} et k_{h_0} sont ainsi formulées en fonction des formulations des déformations, de E et ν , ainsi que des forces de correction : $F_{\perp h_0}$ et $F_{\perp l_0}$ correspondant respectivement aux tests longitudinal et latéral.
11. Ils étendent ensuite la méthode en 3D en définissant un pavé avec douze ressorts situés le long de ses arêtes et quatre ressorts diagonaux internes au pavé.
12. Ils procèdent ensuite de la même manière que pour le 2D en commençant par formuler les raideurs des ressorts diagonaux avec un test de cisaillement, puis en formulant les autres raideurs avec des tests d'étirement et l'ajout de forces de correction.

Les formulations des raideurs déduites de ces travaux, sont présentées et utilisées dans la suite de ce manuscrit.

Natsupakpong *et al.* [NCÇ10] présentent une approche similaire à celle de Lloyd. En effet, ils cherchent également à minimiser la différence entre la matrice de raideur d'un SMR et celle d'un MEF pour un même objet, afin de déterminer les coefficients de raideur des ressorts. Ils procèdent ensuite à une optimisation permettant de minimiser l'erreur entre ces deux matrices.

San-Vicente-Otamendi [SVO11] propose dans sa thèse une méthode permettant également de déterminer les coefficients de raideurs des ressorts du SRM en fonction d'une MEF linéaire. Pour cela, il commence par calculer analytiquement la matrice de raideur d'un cube, en utilisant la MEF. Puis, il linéarise les équations du SMR afin d'obtenir la matrice de raideur linéarisée associée, de manière similaire à Lloyd. Ces deux matrices de raideur sont ensuite comparées et une méthode de minimisation est employée afin de déterminer les coefficients de raideurs. Enfin, une fois que les paramètres ont été déterminés pour un cube, l'ensemble de ces cubes sont assemblés pour former l'objet déformable. Contrairement à Lloyd, leur méthode ne nécessite pas de fixer le coefficient de Poisson.

Enfin, Sala *et al.* [STF⁺11] proposent une méthode permettant l'initialisation automatique des masses et des raideurs d'un SMR appliqué à un maillage tétraédrique. De

plus, le coefficient d'amortissement des ressort est déterminé grâce à l'approche de Rayleigh [CD03]. Enfin, la paramétrisation des ressorts emploie également la méthode de Lloyd.

2.2.3 Conclusion sur la simulation physique d'objets déformables.

Nous avons fait le choix, pour la simulation physique d'objets déformables, d'un modèle physique fondé sur un SMR. Les raisons sont les suivantes : la simplicité de ses équations, son implémentation intuitive, son interactivité et sa facilité à gérer les changements topologiques. La paramétrisation de notre SMR se fait avec une méthode analytique proposée par Baudet *et al.* Cette dernière nous donne une formulation des coefficients de raideur des ressorts en fonction des paramètres physiques des matériaux que sont le modèle de Young et le coefficient de Poisson. Elle est valable en 2D comme en 3D pour des rectangles et des pavés. Le SMR était notre premier choix et les travaux présentés dans ce manuscrit reposent sur ce dernier. Cependant, le modèle masse-tenseur est également adapté aux changements topologiques et permet une plus grande précision de la simulation en comparaison au SMR. Or, sa mise en place étant plus complexe, il a été intégré tardivement à notre méthode. Enfin, la résolution numérique choisie est de type explicite symplectique.

2.3 Modèle topologique

Pour modéliser géométriquement un solide, des méthodes basées CSG (Constructive Solid Geometry) ou B-Rep (Boundary Representation) sont généralement employées.

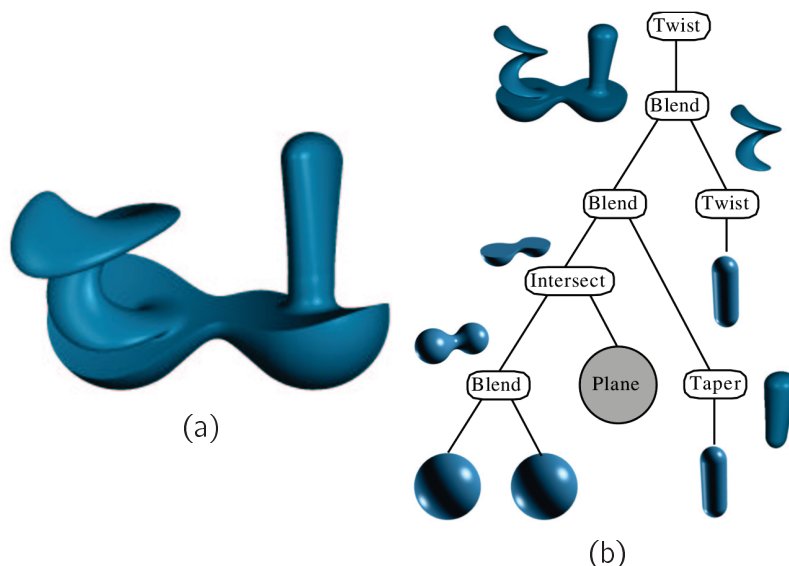
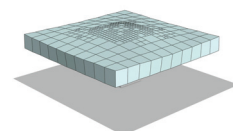


Figure 2.19 – Le modèle CSG étendu aux surfaces implicites. (a) Une surface implicite générée par une succession d'opérations booléennes et de déformations. (b) L'arbre des opérations ayant permis la construction de l'objet présenté dans (a) (schéma issu de [WGG99]). Wyvill *et al.* sont à l'origine de la définition des surfaces implicites [WMW86, VvO96].

Les modèles CSG consistent en la construction d'un modèle virtuel d'un objet réel en combinant des formes primitives par application d'opérations booléennes (*cf.* la figure 2.19). Requicha *et al.* sont à l'origine de la définition du modèle CSG [Voe77, RA77, RA78].



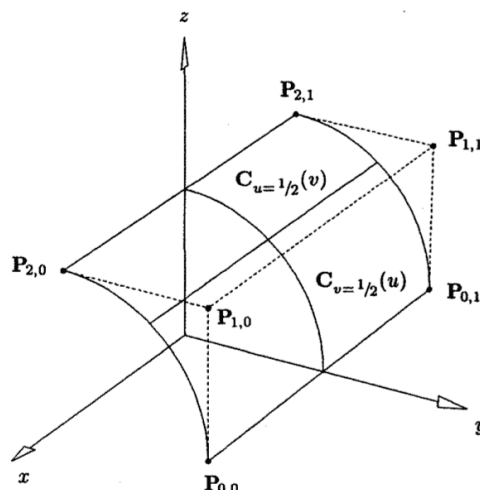


Figure 2.20 – Un exemple de surface de Bézier. Un patch d'une surface cylindrique générée par une surface rationnelle de Bézier [PT97].

Les modèles B-Rep, quant à eux, modélisent un objet réel par un ensemble de surfaces paramétriques ou polygonales. Une modélisation par surfaces paramétriques est un agencement de carreaux géométriques restreints pouvant être définis par des surfaces B-splines, de Béziers ou des NURBS [Pou92] (cf. la figure 2.20). Les surfaces polygonales ou maillages consistent, quant à eux, en une décomposition en cellules d'un objet réel. Ces cellules peuvent avoir une géométrie 2D en étant de type polygonal (triangles, quadrilatères, ...) ou une géométrie 3D avec des polyèdres (tétraèdres, pyramides, prismes, cubes, ...). Chacune de ces cellules est composée de points caractérisés par des coordonnées géométriques. La représentation par maillage est décomposée en deux catégories qui sont les complexes simpliciaux et les complexes cellulaires. Un simplexe est défini comme étant la généralisation d'un triangle qui est un objet 2D, à une dimension quelconque. Par exemple, un tétraèdre est la généralisation d'un triangle en 3D. Cette définition nous amène à celle des complexes simpliciaux. Les complexes simpliciaux consistent en la décomposition d'un objet réel en des cellules de type simplexe. Contrairement aux complexes simpliciaux, les complexes cellulaires sont un ensemble de cellules de topologie quelconque. Les complexes simpliciaux ou cellulaires définissent la topologie d'un objet en établissant les relations d'incidence et d'adjacence entre l'ensemble des différents éléments du modèle comme les sommets, arêtes, triangles (*resp.* faces) et plus généralement un simplexe (*resp.* cellule) de dimension quelconque. Les complexes simpliciaux présentent cependant des limitations par rapport aux complexes cellulaires. En effet, lorsqu'un simplexe est scindé en deux parties, il est possible d'obtenir deux éléments ne correspondant pas à des simplexes. Par exemple, un triangle peut être scindé en un sous-triangle et un sous-trapèze. Des méthodes coûteuses et complexes sont alors employées afin de garantir des éléments de type simplexe au cours des changements topologiques comme des algorithmes de triangulation.

Dans la suite de ce manuscrit, nous nous intéresserons aux modèles B-Rep et plus particulièrement aux complexes cellulaires. Comme nous l'avons mentionné précédemment, un maillage est une généralisation du concept de complexe cellulaire. Il est considéré comme un ensemble d'informations géométriques et topologiques, permettant de modéliser un objet réel. Nous emploierons par la suite le terme de maillage pour qualifier les différentes modélisations géométriques et topologiques d'objets, présentées dans ce manuscrit. Nous étudierons également les structures de données topologiques permettant de représenter un maillage. Ces

dernières stockent l'ensemble des informations géométriques et topologiques d'un maillage. Les informations topologiques d'un maillage sont retrouvées grâce aux relations entre les différents éléments composant un maillage, définies au niveau de la structure de données topologique. Ces relations permettent de parcourir ou encore de modifier le maillage. Chaque structure de données topologique est ainsi caractérisée par les informations qu'elle stocke, la manière de les stocker, la définition des opérations permettant de la parcourir ainsi que les algorithmes gérant sa modification et leur efficacité.

De nombreuses structures de données topologiques ont été définies. Nous présenterons par la suite, les plus usitées d'entre elles. Pour cela, il est nécessaire d'explicitier au préalable quelques notions et terminologies relatives au domaine de la topologie.

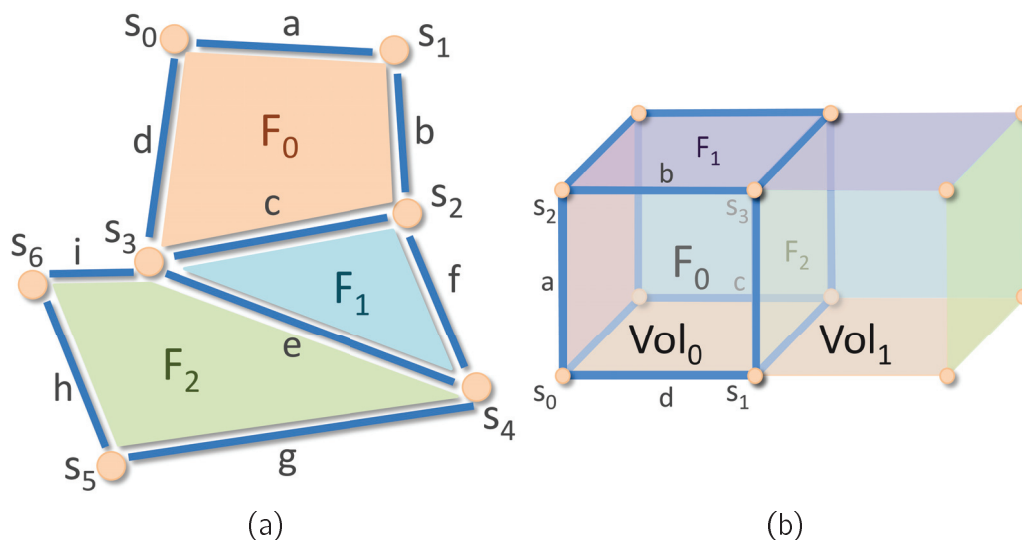


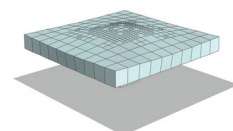
Figure 2.21 – La notion de i -cellule et de relation d'incidence et d'adjacence. (a) Un maillage 2D composé de trois faces (2-cellules) : F_0, F_1 et F_2 , neuf arêtes (1-cellules) labellisées de a à i et de sept sommets (0-cellules) labellisés de s_0 à s_6 . (b) Un maillage 3D composé de deux volumes (3-cellules) : Vol_0 et Vol_1 , de onze faces (2-cellules) dont F_0, F_1 et F_2 , de vingt arêtes (1-cellules) dont a, b, c et d et de douze sommets (0-cellules) dont s_0, s_1, s_2 et s_3 .

2.3.1 Quelques notions de topologie

Une structure de données topologique représente un objet par une décomposition en cellules (sommets, arêtes, faces, volumes, ...). De manière plus formelle, une **i -cellule** est une cellule de dimension i . En 3D, une 0-cellule correspond à un sommet, une 1-cellule à une arête, une 2-cellule à une face et une 3-cellule à un volume. Chaque cellule de dimension i ($i \geq 1$) est bornée par des cellules de dimension $i - 1$, ces dernières constituent sa frontière. Par exemple, une face (2-cellule) est bornée par des arêtes (1-cellules).

La figure 2.21 illustre ce propos. En effet, dans la figure 2.21(a), la 2-cellule F_1 est bornée par les 1-cellules : c, f et e . La 1-cellule d est bornée par les 0-cellules s_0 et s_3 . Dans la figure 2.21(b), la 3-cellule Vol_0 est bornée par six 2-cellules dont F_0, F_1 et F_2 . La 2-cellule F_0 est bornée par les 1-cellules : a, b, c et d . Enfin, la 1-cellule b est bornée par les 0-cellules s_2 et s_3 .

Deux cellules sont dites **incidentes** si une cellule appartient à la frontière de l'autre. Par exemple, les sommets (0-cellules) d'un volume (3-cellule) sont incidents au volume. Deux



cellules sont **adjacentes** si elles sont de même dimension i ($i \geq 1$) et si elles sont incidentes à une même cellule de dimension $i - 1$.

Dans la figure 2.21(a), la 1-cellule e appartient à la frontière de la 2-cellule F_1 . e et F_1 sont donc incidentes. e et F_2 étant également incidentes, les 2-cellules F_1 et F_2 sont ainsi adjacentes. La 0-cellule s_3 est incidente aux 1-cellules : d , c et e . Ces dernières 1-cellules sont ainsi adjacentes entre elles. Dans la figure 2.21(b), la 2-cellule F_2 appartient à la frontière de Vol_0 et de Vol_1 . F_2 et Vol_0 sont donc incidents, tout comme F_2 et Vol_1 . Vol_0 et Vol_1 sont ainsi adjacents selon F_2 . La 1-cellule b est incidente aux 2-cellules : F_0 et F_1 . F_0 et F_1 sont ainsi adjacentes. Enfin, la 0-cellule s_2 est entre autre incidente aux 1-cellules a et b . Ces dernières sont donc adjacentes entre elles.

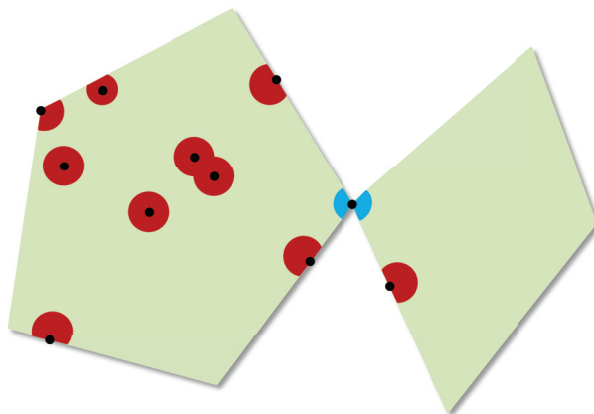


Figure 2.22 – La définition mathématique d'une variété 2D. Une surface qui n'est pas, par définition mathématique, une variété 2D. En effet, chaque voisinage des points de la surface (en noir) est homéomorphe à un disque (en rouge) sauf le point à l'intersection des deux faces vertes.

Une surface est définie mathématiquement comme étant une **variété 2D** (2D manifold en anglais) que l'on abrégera 2-variété par la suite, si chaque point de cette dernière est homéomorphe à un disque ou un demi-disque au niveau de sa frontière. Ce concept est illustré par la figure 2.22.

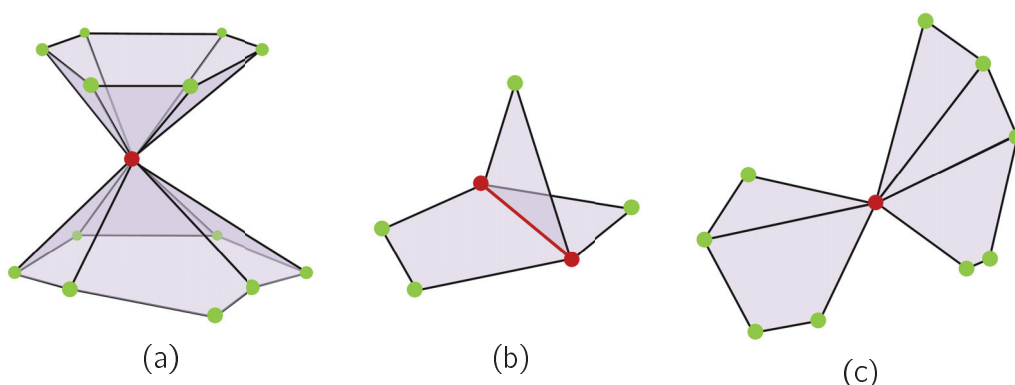


Figure 2.23 – La définition pour un maillage d'une variété 2D par des contre-exemples. (a) Deux surfaces reliées par un sommet en rouge. Ce dernier n'est pas une variété. (b) L'arête en rouge n'est pas une variété car elle possède plus de deux faces incidentes. (c) Le sommet en rouge n'est pas une variété car il relie deux éventails de faces [Bot05].

Nous spécifions maintenant cette définition à un maillage. Pour cela, nous définissons tout d'abord la notion de non-variété pour une arête et un sommet. Une arête n'est pas

une variété, si elle est incidente à plus de deux faces (cf. la figure 2.23(b)), c'est-à-dire si elle appartient à la bordure de plus de deux faces. Un sommet est une non-variété, si il est incident à plus d'un éventail de faces comme montré dans la figure 2.23(c). Un maillage est ainsi une 2-variété si toutes ses arêtes sont des variétés ainsi que tous ses sommets et qu'il ne comporte pas d'intersections [BKP⁺10].

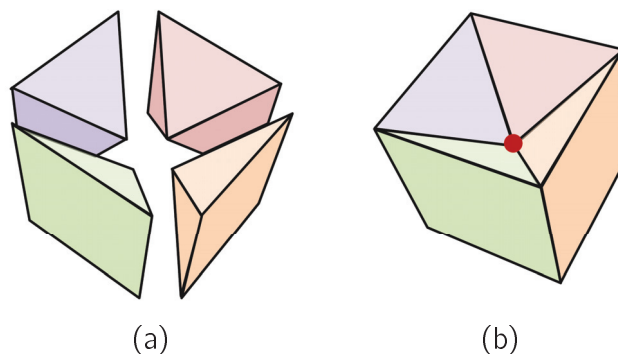


Figure 2.24 – La notion mathématique de quasi-variété pour un objet 3D. Un objet 3D ayant la propriété de quasi-variété. En effet, l'objet 3D présenté dans (b) est constitué de quatre pyramides (comme montré dans (a)), qui sont cousues deux à deux selon leurs faces.

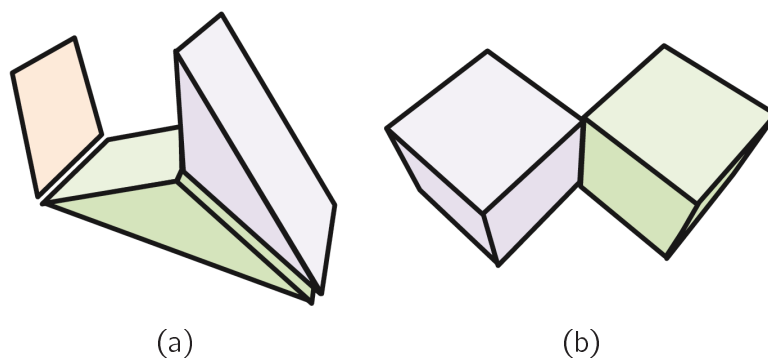
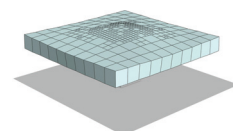


Figure 2.25 – La définition pour un maillage 3D, d'une quasi-variété par des contre-exemples. (a) Un objet 3D qui n'est pas une quasi-variété car il contient une 2-cellule orange isolée. (b) Un objet 3D n'étant pas une quasi-variété car les deux 3-cellules le composant partagent une arête mais n'ont pas de face commune.

La notion combinatoire de variété n'a pas été caractérisée en 3D. C'est pour cette raison que la notion de **quasi-variété** a été définie pour un maillage de dimension supérieure à 2. La figure 2.24 présente un objet 3D ayant une particularité similaire à celle de l'objet 2D présenté dans la figure 2.22(a). Cet objet 3D n'est pas une variété puisque tous les points le composant, ne sont pas homéomorphiques à une boule. En effet, le point rouge représenté dans la figure 2.24(b) n'est pas homéomorphe à une boule. Par contre, c'est une quasi-variété car toutes ses 3-cellules (resp. n -cellules pour un objet de dimension n) prises séparément peuvent être cousues selon une 2-cellule (resp. $(n - 1)$ -cellule) et chaque 2-cellule (resp. $(n - 1)$ -cellule) est incidente à au plus deux 3-cellules (resp. n -cellules). Enfin, la figure 2.25 montre deux exemples d'objets 3D n'étant pas des quasi-variétés [Dam12].

Un **plongement** consiste à associer une géométrie à une représentation topologique d'un objet réel.

Un objet est dit **orientable**, si plongé dans un espace euclidien, on peut distinguer un intérieur d'un extérieur. Un ruban classique est un exemple d'objet orientable, contrairement



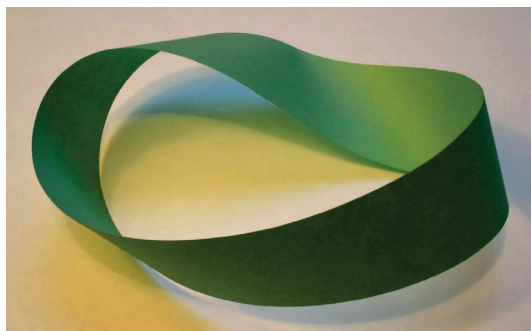


Figure 2.26 – Un exemple d'objet 2D non-orientable. Ruban de Möbius.

au ruban de Möbius qui est un objet non-orientable (cf. la figure 2.26).

La présentation des structures de données topologiques les plus employées, commence par celles gérant exclusivement des maillages surfaciques. Puis, nous introduirons celles qui sont valables pour des maillages de dimension quelconque.

2.3.2 Structure de données topologiques pour les maillages surfaciques

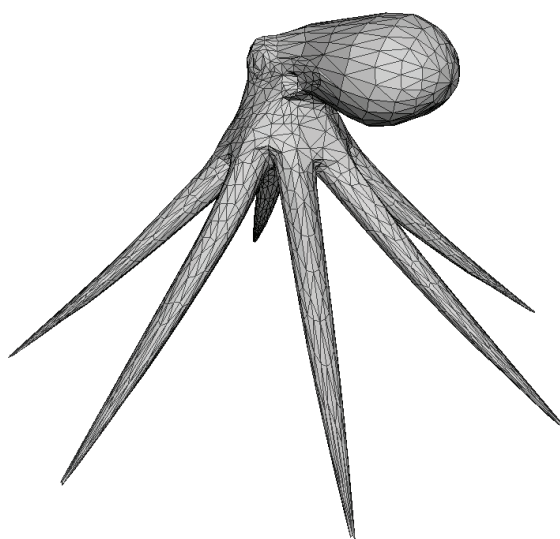


Figure 2.27 – Un exemple de maillage surfacique. Un objet 2D représentant un octopus est décomposé en un ensemble de triangles.

Un maillage surfacique modélise un objet réel par une décomposition de ce dernier en cellules ayant une topologie 2D (cf. la figure 2.27). Notons qu'un maillage surfacique peut représenter un objet volumique fermé comme par exemple une sphère. En effet, un maillage constitué d'éléments 2D peut être plongé dans \mathbb{R}^3 .

Nous allons maintenant présenter différentes structures de données topologiques permettant de stocker exclusivement des maillages surfaciques, avec dans un premier temps deux structures de données fondées sur les sommets et les faces, puis différentes structures de données centrées sur les arêtes. Ces dernières, en s'appuyant sur l'arête, facilitent la recherche d'informations de connectivité. En effet, l'arête est l'élément topologique à la jonction de deux faces adjacentes.

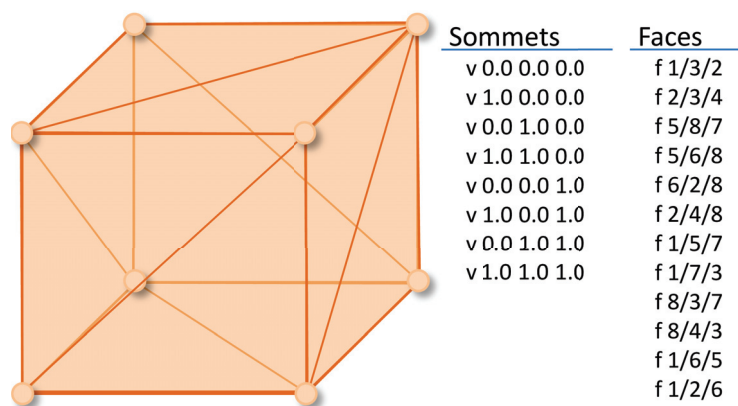


Figure 2.28 – Un exemple d'une structure de données fondée sur les faces. Un hexaèdre triangulé représenté avec une structure de données fondée sur les faces. Cette dernière stocke les coordonnées géométriques des sommets du maillage. De plus, chaque face du maillage est stockée avec les indices des sommets la composant.

Structure de données classique

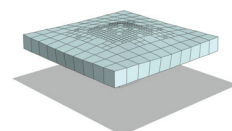
En informatique graphique, la structure de données la plus utilisée et également la plus simple est une structure fondée sur les sommets et les faces. Elle stocke toutes les coordonnées géométriques des sommets du maillage auxquels correspondent des indices. Chaque face est ensuite définie par un ensemble d'indices correspondant aux sommets composant la face en question. Cette structure de données basique est classique pour afficher un maillage mais ne contient aucune information relative à la connectivité du maillage. Par exemple, pour déterminer les faces adjacentes d'une face (les faces adjacentes sont les faces ayant une arête en commun avec la face en question), il faut parcourir toutes les faces du maillage et chercher les correspondances de sommets en s'appuyant sur leurs coordonnées géométriques. Notons que cette structure de données est celle du format .OBJ (cf. l'exemple montré dans la figure 2.28). Enfin, des variantes proposent de stocker les faces incidentes à chaque sommet ou encore l'ensemble des faces adjacentes à chaque face.

Graphe d'incidence

Un **graphe d'incidence** est un modèle topologique cellulaire non orienté. Il consiste en un graphe ayant pour nœuds les cellules du maillage, et une arête entre chaque paire de cellules incidentes entre elles. Les faces ont pour fils les arêtes qui leur sont incidentes. Enfin, les arêtes ont pour fils les différents sommets qui leur sont incidents. Un exemple de graphe d'incidence est donné dans la figure 2.29. Le maillage représenté par le graphe d'incidence possède deux faces F_0 et F_1 , six arêtes (a, \dots, f) et cinq sommets (A, \dots, E). Le lien reliant F_1 à e dans le graphe, indique que l'arête e est incidente à la face F_1 , elle fait partie de la bordure de cette face. Le sommet D est incident à f , il constitue une extrémité de cette arête, ils sont ainsi reliés dans le graphe d'incidence.

Structure de données de l'arête ailée

La **structure de données de l'arête ailée** (**winged-edge data structure** en anglais) est un exemple de structure de données centrée arête [Bau72, Bau75]. C'est un modèle cellulaire orienté. En effet, la structure est composée de cellules de différentes dimensions : sommets,



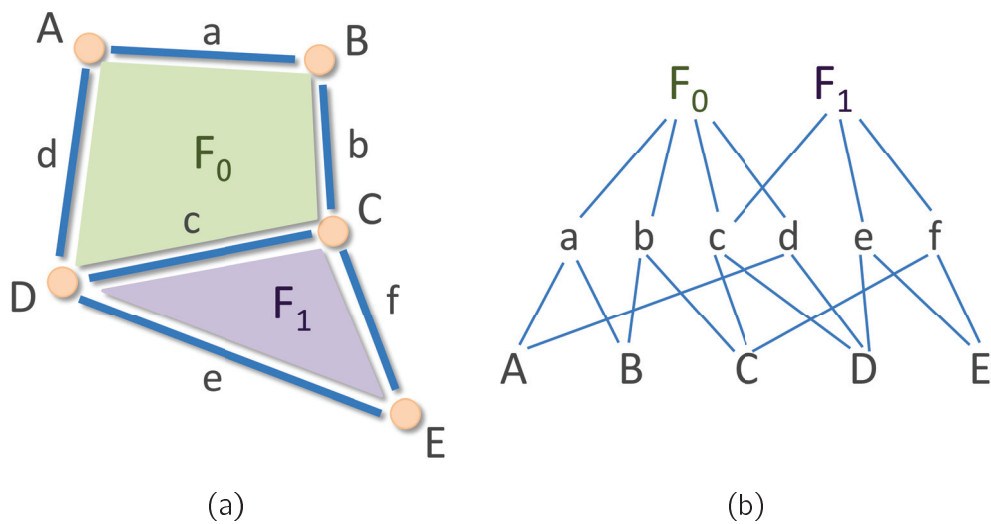


Figure 2.29 – Un exemple de graphe d'incidence. (a) Un objet 2D constitué de deux faces F_0 et F_1 , de six arêtes labellisées de a à f et de cinq sommets labellisés de A à E. (b) Le graphe d'incidence associé au modèle présenté dans (a).

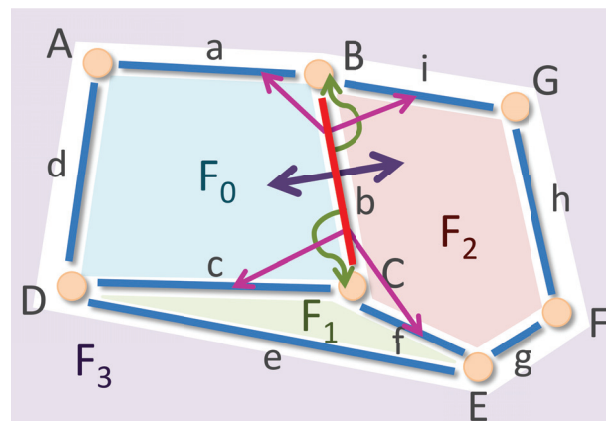


Figure 2.30 – Un exemple de la structure de données de l'arête ailée. Un objet 2D constitué de quatre faces : F_0 , F_1 , F_2 et F_3 , de neuf arêtes ailées labellisées de a à i et de sept sommets labellisés de A à G. L'arête ailée b (en rouge) est représentée avec l'ensemble de ses pointeurs. Les flèches vertes représentent ses pointeurs sur ses deux extrémités B et C. Les flèches violettes correspondent à ses pointeurs sur ses deux faces incidentes F_0 et F_2 . Enfin, les flèches fuchsias sont ses pointeurs sur les arêtes ailées précédentes et suivantes de F_0 et F_2 .

arêtes et faces (de formes quelconques). De plus, ses arêtes sont orientées contrairement au graphe d'incidence présenté précédemment. L'élément principal de cette structure de donnée est nommé « arête ailée ». Cette dernière consiste en une arête orientée ayant huit pointeurs sur les différentes cellules l'entourant. Chaque arête ailée pointe sur ses deux sommets incidents, sur ses deux faces incidentes et enfin sur les quatre arêtes précédentes et suivantes appartenant à la bordure de ses deux faces incidentes. Nous précisons que l'arête ailée différencie sa face incidente gauche de sa face incidente droite. Cette caractéristique correspond au fait que les arêtes de la structure de données, soient orientées.

La figure 2.30 présente un exemple de cette structure de données. L'arête ailée b (de couleur rouge) pointe sur ses deux extrémités B et C . Cette relation est représentée par les deux flèches vertes sur le schéma. b pointe sur ses deux faces incidentes F_0 : sa face gauche et F_2 : sa face droite (les flèches violettes sur la figure). Nous précisons que b est l'arête séparant F_0 et F_2 . Enfin, les flèches fuchsias correspondent aux pointeurs sur les arêtes précédentes et suivantes de F_0 et de F_2 .

En plus de la liste de l'ensemble des arêtes ailées, cette structure de données stocke également une liste de tous les sommets et de toutes les faces composant l'objet représenté. Des informations peuvent être associées aux sommets et aux faces, comme par exemple une coordonnée géométrique pour le sommet et une couleur pour la face. Chaque sommet et chaque face contient, en plus des informations éventuellement choisies par l'utilisateur, un pointeur sur une de ses arêtes ailées incidentes.

Un des inconvénients de la structure de données de l'arête ailée est la présence de deux cycles d'arêtes orientées autour de la même arête. En effet, il existe un cycle pour la face de gauche et un autre pour la face de droite impliquant la définition des quatre pointeurs sur les arêtes précédentes et suivantes. Ceci amène à des difficultés pour le parcours du maillage et surtout pour sa modification. Par exemple, afin de parcourir la bordure d'une face, il faut choisir pour chaque arête ailée parcourue le bon pointeur parmi les quatre stockés. Nous précisons que ces pointeurs font référence à ceux ciblant les arêtes ailées précédentes ou suivantes de la face de gauche ou de droite de l'arête ailée en question.

Structure de données des demi-arêtes

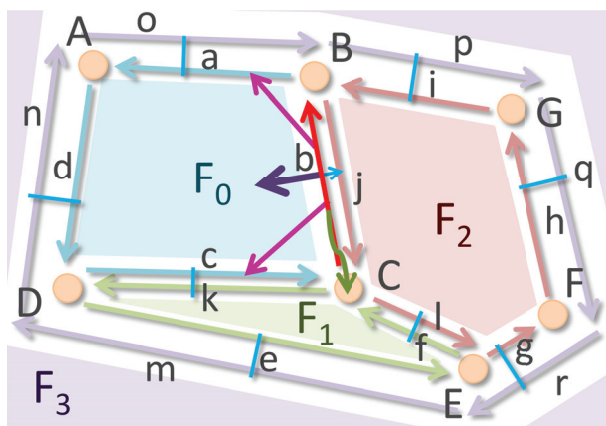
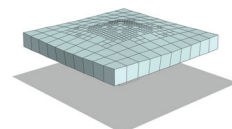


Figure 2.31 – Un exemple de la structure de données des demi-arêtes. Un objet 2D constitué de quatre faces : F_0 , F_1 , F_2 et F_3 , de dix-huit demi-arêtes labellisées de a à r et de sept sommets labellisés de A à G . La demi-arête b (en rouge) est représentée avec l'ensemble de ses pointeurs. La verte représente son pointeur sur son extrémité C . La flèche violette correspond à son pointeur sur sa face incidente F_0 . Enfin, les flèches fuchsias sont ses pointeurs sur les demi-arêtes précédentes et suivantes, incidentes à F_0 . Les liens bleus relient deux demi-arêtes opposées.



La structure de données des demi-arêtes (**half-edge data structure** en anglais) est également une structure de données centrée arête [Män87]. Elle correspond à une amélioration de la structure de données de l'arête ailée en scindant chaque arête ailée orientée en deux demi-arêtes orientées et opposées. En effet, cette distinction permet d'avoir un seul sommet et une seule face par demi-arête. Chaque demi-arête est donc maintenant incidente à un seul cycle d'arêtes orientées. Une demi-arête d est donc une moitié d'arête orientée avec quatre pointeurs. Il y a un premier pointeur sur son sommet incident, deux autres sur la demi-arête précédente et suivante appartenant à la même face que d et enfin un dernier pointeur sur la demi-arête appartenant à la même arête que d mais pas à la même face.

La figure 2.31 montre un exemple de cette structure. La demi-arête b (en rouge) appartient à une seule face F_0 (contrairement à l'arête ailée). b pointe sur son sommet C , sur l'arête suivante a et l'arête précédente c appartenant à la même face incidente F_0 et enfin sur l'arête opposée j .

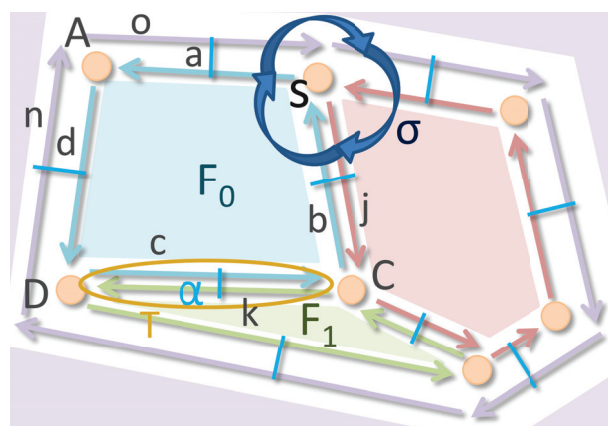


Figure 2.32 – La notion d'involution et de permutation pour la structure de données des demi-arêtes. Un objet 2D constitué de quatre faces, de dix-huit demi-arêtes et de sept sommets. Sa labellisation est similaire à celle de la figure 2.31 mais pour des raisons de clarté, seuls certains éléments possèdent un label. Le lien bleu correspond à une involution α et la flèche bleu foncé à une permutation σ . Les flèches bleu foncé sont seulement représentées autour du sommet s . L'arête T est composée des deux demi-arêtes c et k .

De manière plus formelle, une structure de données des demi-arêtes \mathcal{S} est un ensemble de demi-arêtes A , d'une involution α et d'une permutation σ [Dam10]. L'involution α permet à partir d'une demi-arête de récupérer son opposé appartenant à la même arête et au même volume mais pas à la même face. C'est une involution car si on applique deux fois α sur une demi-arête a , on obtient a . α correspond aux liens bleus dans la figure 2.32. La permutation σ permet quant à elle de donner le successeur d'une demi-arête autour d'un sommet. σ est une permutation car à partir d'une demi-arête a , il faut appliquer σ plusieurs fois (pas forcément 2 fois) pour retrouver a . Une carte combinatoire est ainsi notée $\mathcal{S} = (A, \alpha, \sigma)$. Une structure de données des demi-arêtes a pour nombre de demi-arêtes, le double de son nombre d'arêtes.

Enfin, il faut noter que les sommets ne sont pas explicitement représentés dans une structure de données des demi-arêtes. En effet, un sommet s est défini comme l'ensemble des demi-arêtes obtenues à partir d'une demi-arête a incidente à s , en appliquant σ jusqu'à retrouver a . La figure 2.32 illustre ce propos avec les trois flèches bleues passant d'une demi-arête incidente au sommet s à une autre. Nous rappelons qu'une demi-arête incidente à un sommet est une demi-arête partant du sommet en question.

Un cycle de demi-arêtes délimite une face. On définit la notion de face (par exemple

la face F_0 dans figure 2.32) comme étant un cycle de demi-arêtes (dans notre exemple, $\{a, b, c, d\}$). Afin d'obtenir l'ensemble D des demi-arêtes d'un cycle, à partir d'une demi-arête a avec $a \in D$, on applique l'involution α puis la permutation σ . On reproduit ce processus jusqu'à retomber sur a . La composition $\sigma \circ \alpha$ donc également implicitement à une permutation.

Nous expliquons la composition $\sigma \circ \alpha$ avec la figure 2.32. On note D , l'ensemble des demi-arêtes formant le cycle définissant F_0 . Pour parcourir la face F_0 , on part de la demi-arête a qui est incidente à F_0 . En appliquant α à a , on obtient o . Puis, la permutation σ donne à partir de o , d qui est une demi-arête appartenant à D . On procède de la même manière pour d en se conformant à l'orientation donnée par la structure de données (orientation des demi-arêtes) et la permutation σ (orientation donnée par les flèches bleues). On obtient ainsi $c = \sigma \circ \alpha(d)$ et finalement $b = \sigma \circ \alpha(c)$.

2.3.3 Structure de données nD

Carte combinatoire

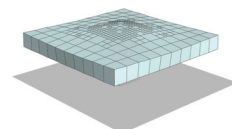
La présentation **des cartes combinatoires** faite dans cette partie s'appuie sur la documentation CGAL [Dam12].

Le modèle des cartes combinatoires fait partie des rares modèles complètement génériques par rapport à la dimension des objets représentés [Lie89, Lie91, Lie94]. Il consiste en une structure de données combinatoire permettant la description d'objets orientés nD subdivisés en cellules de topologie quelconque et de différentes dimensions (sommet, arête, face, volume, ...).

La carte combinatoire est un modèle centré arête. Elle est définie à partir d'un élément basique nommé **brin**, et un ensemble de **pointeurs** entre ces brins. Elle correspond à une généralisation de la structure de données des demi-arêtes en toute dimension. En effet, la notion des demi-arêtes se retrouve dans la définition des brins des cartes combinatoires. Un brin correspond à une notion abstraite pouvant être définie comme une partie d'une arête orientée (1-cellule), tout en étant une partie des cellules incidentes de dimension i ($i \in \{0, 2, 3, \dots, n\}$). Grâce à l'ensemble des pointeurs définis entre les brins, toutes les relations d'incidence et d'adjacence entre les cellules de différentes dimensions sont représentées. Enfin, des contraintes définies sur ces pointeurs permettent de garantir la validité topologique des objets ainsi décrits.

La définition formelle faite pour les structures de données des demi-arêtes correspond à celle des cartes combinatoires 2D. En effet, la relation d'adjacence entre les arêtes est représentée par une permutation, qui à partir d'un brin donne le brin suivant appartenant à la même 2-cellule en respectant une orientation donnée. Cette permutation est notée β_1 car elle met en relation des arêtes qui sont des cellules de dimension 1 : les 1-cellules. La relation d'adjacence entre les faces est représentée par une involution β_2 car mettant en relation des cellules de dimension 2 : les 2-cellules. Par rapport aux notions présentées pour la structure de données des demi-arêtes, β_1 correspond à la permutation : $\sigma \circ \alpha$ et β_2 à l'involution α . Cette définition est étendue pour une dimension quelconque. Les β_i (avec $i \in \{0, \dots, n\}$ et n : la dimension de la carte combinatoire) permettent ainsi la généralisation de cette première définition en dimension quelconque. Nous précisons que la structure de données des demi-arêtes définit une permutation et une involution, alors qu'une carte combinatoire nD , définit une permutation et $n - 2$ involutions.

La définition formelle des cartes combinatoires est la suivante [Lie91, Dam12] :



Définition 1

Soit $n \geq 0$. Une n -carte combinatoire (carte combinatoire de dimension n) est une algèbre $\mathcal{C} = (B, \beta_1, \dots, \beta_n)$ où

1. B est un ensemble fini de brins ;
2. β_1 est une permutation sur B ;
3. $\forall i : 2 \leq i \leq n : \beta_i$ est une involution sur B ;
4. $\forall i, j : 1 \leq i \leq i+2 \leq j \leq n : \beta_i \circ \beta_j$ est une involution.

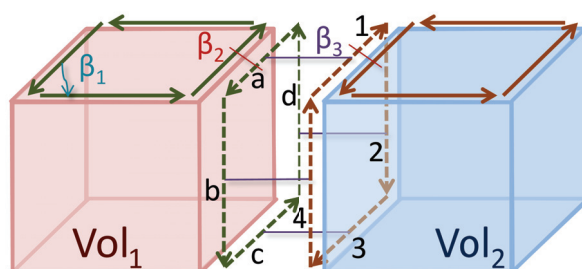


Figure 2.33 – Les brins composant une face incidente à deux volumes. Deux 3-cellules adjacentes : Vol_1 et Vol_2 où les brins de la 2-cellule commune sont représentés en pointillé (numérotés de a à d et de 1 à 4). Les deux volumes sont reliés deux à deux par β_3 . Les traits bleus (resp. rouges, violets) correspondent aux liaisons β_1 (resp. β_2 , β_3).

Les β_i sont des involutions sauf β_1 qui est une permutation. On retrouve dans cette définition, l'ensemble des concepts présentés précédemment pour la structure de données des demi-arêtes. La dernière ligne de la définition pose des contraintes pour les objets de dimension supérieure ou égale à 3. Par exemple, pour $i = 1$ et $j = 3$, la contrainte ajoutée est que $\beta_1 \circ \beta_3$ doit être une involution. Cette contrainte permet de garantir la validité topologique des cartes combinatoires, c'est-à-dire qu'elles représentent des quasi-variétés.

La figure 2.33 permet d'illustrer cette définition. En appliquant l'involution β_3 (lien violet) sur le brin a , on obtient le brin 1. La permutation β_1 sur 1 donne 2. De plus, $\beta_3(2) = d$ puis $\beta_1(d) = a$. Nous retombons sur le brin de départ a montrant la propriété d'involution de $\beta_1 \circ \beta_3$. Cette contrainte, formalisée par la dernière ligne de la définition, exige que lorsque deux brins appartenant à deux cycles de brins différents (les cycles formés des brins violets et gris dans la figure 2.33) sont en relation par β_3 (liens violets), alors tous les autres brins des deux cycles doivent être en relation deux à deux par β_3 .

Pour terminer, notons que cette définition de base des cartes combinatoires décrivant la notion de brins, pointeurs et contraintes, permet de représenter la topologie des objets (décomposition en cellules et relations d'incidence et d'adjacence entre elles). Cependant, cette description ne contient pas d'informations supplémentaires. Or, des informations relatives par exemple à la géométrie, la couleur des sommets ou encore l'aire des faces de l'objet peuvent être utiles. Dans ce sens, les cartes combinatoires permettent d'associer n'importe quel type d'information à n'importe quel type de cellule. De manière plus formelle, les informations relatives à une i -cellule sont stockées dans un i -attribut. Cet i -attribut est associé à tous les brins appartenant à une même i -cellule. Ce procédé permet d'accéder directement depuis un brin à tous les i -attributs définis.

Vocabulaire Nous définissons dans cette partie, quelques notions et termes relatifs aux cartes combinatoires.

Nous notons β_0 , la permutation qui, à partir d'un brin b , donne le brin précédent appartenant à la même face et au même volume, mais pas à la même arête : $\beta_0 = \beta_1^{-1}$. De plus, la notation β_{ij} équivaut à $\beta_j \circ \beta_i$.

Le **degré** d'une i -cellule avec $0 \leq i \leq n$, correspond au nombre de $(i + 1)$ -cellules incidentes à cette dernière.

La **i -couture** permet de mettre en relation deux brins par β_i . Si deux brins b_1 et b_2 sont tels que $\beta_i(b_1) = b_2$ alors b_1 est dit **i -cousu** à b_2 . Si β_i est une involution et b_1 est i -cousu à b_2 alors b_2 est i -cousu à b_1 .

Une **orbite** est un ensemble de brins atteignables en appliquant des β_i donnés. La notion d'orbite permet de définir celle d'une i -cellule de manière plus précise (par rapport à la première définition faite dans la sous-section 2.3.1). Une i -cellule est représentée de manière implicite par un ensemble de brins obtenus grâce à la notion d'orbite. Chaque i -cellule correspond à une orbite spécifique. La définition formelle d'une i -cellule est la suivante :

Soit une n -carte \mathcal{C} telle que $\mathcal{C} = (B, \beta_1, \dots, \beta_n)$, un brin $b \in B$ et $i \in \{0, \dots, n\}$. La i -cellule incidente à b , notée $c_i(b)$ est :

- Si $i = 0$ alors la i -cellule correspond à l'orbite $\langle \beta_{02}, \dots, \beta_{0n} \rangle (b)$;
- Si $i > 0$ alors la i -cellule correspond à l'orbite comprenant l'ensemble des β_j ($j \in \{1, \dots, n\}$ et $j \neq i$) : $\langle \beta_1, \dots, \beta_{i-1}, \beta_{i+1}, \dots, \beta_n \rangle (b)$.

La dernière définition montre que pour parcourir l'ensemble des brins d'une i -cellule, il suffit d'enlever de l'orbite la définissant l'involution, β_i . En effet, cette dernière permet de changer de i -cellule.

La différenciation pour les 0-cellules est due à l'hétérogénéité de la définition des cartes combinatoires. En effet, β_1 est une permutation alors que tous les autres β_i sont des involutions.

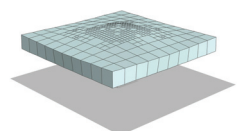
La définition de i -cellule est illustrée à l'aide de la figure 2.32. Prenons pour exemple la 0-cellule s . Le brin a est incident à s , c'est-à-dire a est un brin partant de s . D'après la définition de i -cellule, s correspond à l'orbite $\langle \beta_{02} \rangle (a)$. En effet, en appliquant β_{02} sur a , nous obtenons j . j est bien un brin incident à s . En appliquant une nouvelle fois β_{02} sur j , on obtient p , le troisième brin incident à la 0-cellule s .

Pour la 1-cellule T ayant pour brin incident c , elle correspond d'après la définition à l'orbite $\langle \beta_2 \rangle (c)$. La 1-cellule T est donc définie par les brins c et $k = \beta_2(c)$.

Enfin, pour la 2-cellule F_0 ayant pour brin incident d , elle correspond, d'après la définition, à l'orbite $\langle \beta_1 \rangle (d)$. Si on applique β_1 sur d , on obtient c puis successivement b et enfin a . Nous obtenons ainsi l'ensemble des brins définissant notre 2-cellule. Notons que l'emploi de β_2 , nous ferait changer de 2-cellule.

Objets ouverts La définition de la structure de données de l'arête ailée et celle des demi-arêtes, ainsi que la définition des cartes combinatoires introduite précédemment n'est valable que pour des modèles dits **fermés**. En effet, chaque arête ailée est incidente à deux faces, chaque demi-arête est liée par une involution à sa demi-arête opposée et chaque brin est lié par β_2 à son brin opposé. Ces modèles nécessitent alors la présence d'une n -cellule « complémentaire » au modèle.

Dans les figures 2.30 et 2.31, la face F_3 est une face « complémentaire » au modèle représenté. Cette face externe englobe toutes les autres faces du modèle. Pour le cas de la structure des demi-arêtes, chaque demi-arête est reliée à une arête opposée par un lien bleu dans la figure 2.31.



Pour lever cette limitation, la définition des cartes combinatoires a été étendue pour des objets ouverts. Cette extension permet de représenter des objets orientables ouverts, en utilisant le concept de **brins libres**. La définition d'un brin libre nécessite l'introduction d'un élément spécifique : \emptyset . Un brin peut ainsi être mis en relation avec \emptyset par β_i pour être défini comme i -libre. De manière plus formelle, un brin b d'une carte combinatoire est dit i -libre ($i \in \{0, \dots, n\}$), si $\beta_i(b) = \emptyset$. La figure 2.32 représente indifféremment une structure de données des demi-arêtes ou une carte combinatoire 2D non-étendue. Nous faisons ici l'hypothèse que cette dernière correspond à une carte combinatoire 2D non-étendue. Avec l'extension de la définition des cartes combinatoires, la face F_3 peut ne plus à exister et les brins a, e, f, g, l, d sont maintenant 2-libres.

Lorsqu'un brin est i -cousu avec \emptyset , β_i n'est plus une permutation ou une involution sur B , mais ce que nous définissons comme une **permutation partielle** ou une **involution partielle**. Enfin, nous précisons que la robustesse de la définition d'une involution ou d'une permutation partielle f implique que son application sur un brin libre donne un brin libre, et la restriction de f aux brins non libres est une involution ou permutation.

La définition formelle de l'extension des cartes combinatoires est la suivante :

Définition 2

Soit $n \geq 0$. Une n -carte combinatoire ouverte (carte combinatoire ouverte de dimension n) est une algèbre $\mathcal{C} = (B, \beta_1, \dots, \beta_n)$ où :

1. B est un ensemble fini de brins ;
2. β_1 est une permutation partielle sur B ;
3. $\forall i : 2 \leq i \leq n : \beta_i$ est une involution partielle sur B ;
4. $\forall i, j : 1 \leq i \leq i + 2 \leq j \leq n$ et $j \geq 3 : \beta_i \circ \beta_j$ est une involution partielle.

Cette extension des cartes combinatoires est la version que nous appellerons cartes combinatoires et qui est employée dans nos contributions.

Exemples L'aspect formel des cartes combinatoires étant défini, une présentation plus intuitive et pratique va être donnée dans cette partie.

La figure 2.34 confronte la représentation d'un objet 2D subdivisé en cellules (a) avec la carte combinatoire correspondante (b). L'objet 2D est composé de trois faces (F_1, F_2 et F_3), de neuf arêtes et de sept sommets. La carte combinatoire 2D associée est constituée de douze brins. Les faces F_1 et F_2 sont adjacentes le long de a_1 . L'arête a_1 est donc incidente à F_1 et F_2 . Le sommet s_1 est incident à l'arête a_1 et donc aux faces F_1 et F_2 par transitivité.

Un objet 3D composé de trois volumes (Vol_1, Vol_2 et Vol_3), de quinze faces, de vingt-sept arêtes et de huit sommets, est montré dans la figure 2.35(a). La carte combinatoire 3D associée est représentée dans figure 2.35(b). Une face sépare chaque couple de volumes comme la face F_4 pour les volumes Vol_1 et Vol_2 . La face en question (dans notre exemple F_4) est incidente aux deux volumes (dans notre exemple Vol_1 et Vol_2) qu'elle sépare, et ces volumes sont donc adjacents le long de cette face. L'arête a_4 est incidente à l'ensemble des faces séparant les trois volumes et est donc par transitivité incidente à ces volumes.

Nous rappelons qu'en partant d'un brin b , $\beta_1(b)$ donne le prochain brin appartenant à la même face (*resp.* et au même volume en 3D). De la même manière, $\beta_0(b)$ donne le brin précédent à b . $\beta_2(b)$ permet d'obtenir le brin appartenant à la même arête (*resp.* et le même volume en 3D) mais pas à la même face. Pour le cas 3D, $\beta_3(b)$ donne le brin appartenant à la même arête et à la même face mais pas au même volume.

Chaque brin est représenté par un segment orienté dans les figures 2.34(b) et 2.35(b).

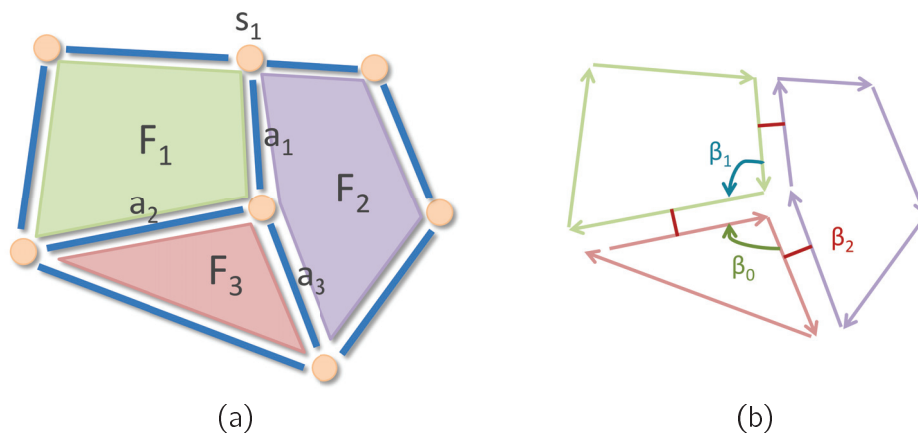


Figure 2.34 – Confrontation d'un objet 2D subdivisé en cellules et sa représentation avec une carte combinatoire. (a) Un objet 2D composé de trois 2-cellules (F_1 , F_2 et F_3), de neuf 1-cellules (dont a_1 , a_2 et a_3) et de sept 0-cellules (dont s_1). (b) Une carte combinatoire 2D composée de douze brins.

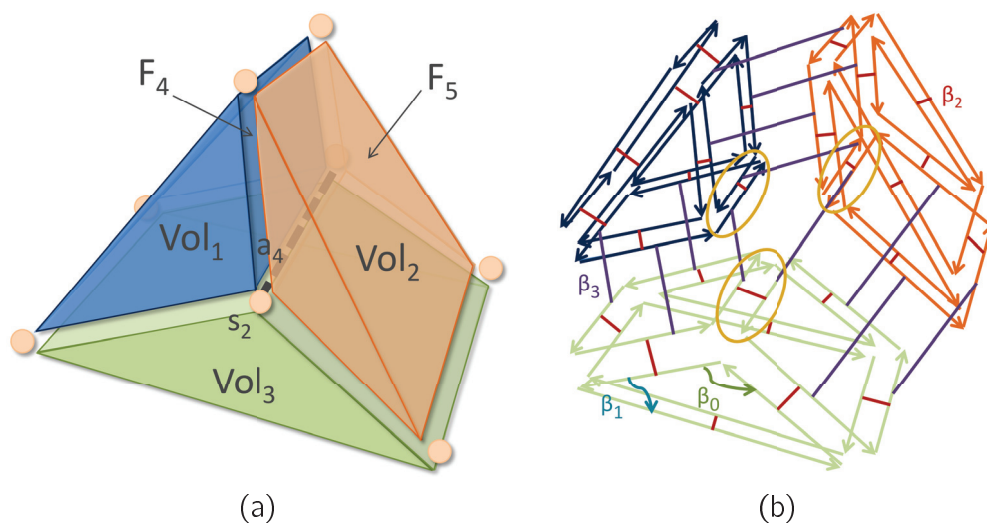
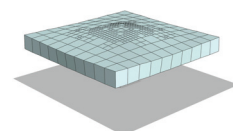


Figure 2.35 – Confrontation d'un objet 3D subdivisé en cellules et sa représentation avec une carte combinatoire. (a) Un objet 3D composé de trois 3-cellules (Vol_1 , Vol_2 et Vol_3), de quinze 2-cellules (dont F_4 et F_5), de vingt-sept 1-cellules (dont a_4) et de huit 0-cellules (dont s_2). (b) Une carte combinatoire 3D contenant cinquante-quatre brins (dix-huit pour chaque 3-cellule). Les ovals jaunes correspondent à la 1-cellule a_4 présentée dans (a).



— **Cas 2D.** Dans la 2.34(b), les trois 2-cellules : F_1 , F_2 et F_3 sont cousues entre elles par des β_2 . Nous explicitons ci-après comment parcourir les brins composant les cellules de différentes dimensions du modèle.

— La 2-cellule (face) F_1 est décrite avec quatre brins. Ces brins sont reliés entre eux par des β_1 . Pour parcourir tous les brins d'une 2-cellule, il faut partir d'un brin b incident à la 2-cellule en question et appliquer successivement β_1 jusqu'à retomber sur le brin de départ b .

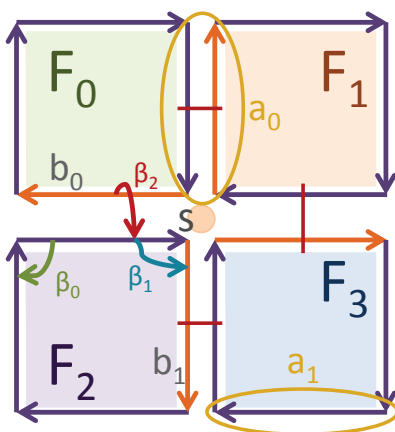


Figure 2.36 – Le parcours des brins incidents à un sommet (resp. arête). Une carte combinatoire 2D constituée de quatre quadrilatères : F_i ($i \in \{0, \dots, 3\}$), cousus ensemble par β_2 . Chaque ovale jaune correspond à une 1-cellule. Les brins orange sont les brins incidents à la 0-cellule s . L'opérateur β_{21} permet de passer d'un brin incident à s à un autre (par exemple, $b_1 = \beta_{21}(b_0)$). Les liens en rouge correspondent aux liaisons β_2 .

— Pour récupérer l'ensemble des brins composant une 1-cellule, il faut récupérer un brin b incident à cette 1-cellule. Puis, si b n'est pas 2-libre, nous appliquons β_2 afin de récupérer le deuxième brin incident à la 1-cellule en question.

Dans la figure 2.36, l'arête a_0 est composée des deux brins cerclés de jaune dont un est incident à F_0 et l'autre à F_1 . Ces derniers sont reliés entre eux par une liaison β_2 . Par contre, l'arête a_1 n'est composée que d'un seul brin car elle n'est incidente qu'à F_3 . Ce brin est par définition, 2-libre.

— Enfin, pour récupérer l'ensemble des brins incidents à un sommet s , il faut récupérer un brin b partant de s . Puis, si b n'est pas 2-libre, nous appliquons $\beta_{21}(b)$ afin d'obtenir un deuxième brin incident à s . Nous parcourons ainsi tous les brins incidents à s , jusqu'à retomber sur b . Par contre, si nous tombons au cours de ce parcours sur un brin libre, il nous faut alors appliquer β_{02} sur le brin ayant amené à ce brin libre, et ceci jusqu'à retomber sur un autre brin libre.

Dans la figure 2.36, les brins orange correspondent aux brins incidents à la 0-cellule s . Pour les parcourir, il faut partir d'un premier brin incident à s . Ce brin consiste en une partie de cette 0-cellule. Nous faisons le choix du brin b_0 qui n'est pas 2-libre. Pour récupérer un autre brin partant de s , nous appliquons alors $\beta_{21}(b_0)$ pour obtenir le brin b_1 qui consiste en un autre brin incident à s . Notons qu'une 0-cellule a un brin incident par face incidente.

— **Cas 3D.** Dans la figure 2.35(b), les trois 3-cellules : Vol_1 , Vol_2 et Vol_3 sont cousues entre elles par des β_3 . Comme précédemment, nous indiquons ci-dessous la procédure

permettant de parcourir les cellules de différentes dimensions du modèle.

- Pour parcourir l'ensemble des brins composant une 3-cellule notée Vol , il faut tout d'abord récupérer un brin b incident à cette 3-cellule. Puis, nous appliquons β_1 afin de parcourir l'ensemble des brins incidents à Vol , composant la face incidente à b . Pour chacun des brins parcourus, nous recommençons cette procédure avec son β_2 et ainsi de suite, jusqu'à ce que nous ayons parcouru l'ensemble des brins de cette 3-cellule. Notons, qu'il faut gérer les redondances. En effet, un brin peut être parcouru plusieurs fois.

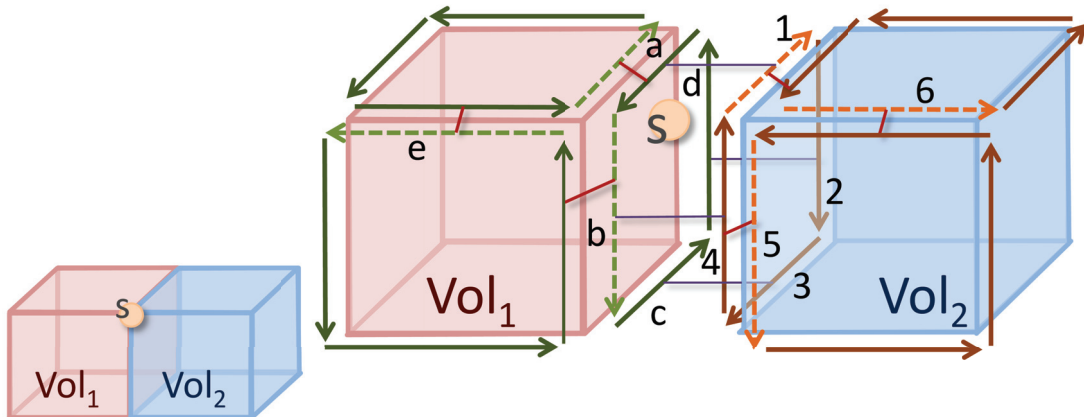


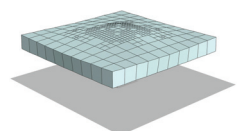
Figure 2.37 – Le parcours des brins incidents à un sommet (resp. arête, face) pour un modèle 3D. Une carte combinatoire 3D constituée de deux hexaèdres adjacents : Vol_1 et Vol_2 cousus ensemble par β_3 . La figure montre l'ensemble des brins incidents à la 2-cellule commune à Vol_1 et Vol_2 . Les brins en pointillé correspondent à l'ensemble des brins incidents à la 0-cellule s . Les traits rouges (resp. violets) représentent les liaisons β_2 (resp. β_3).

- Pour récupérer l'ensemble des brins composant une 2-cellule à partir d'un brin b incident à cette dernière, il faut appliquer β_1 jusqu'à retomber sur b . Puis, si b n'est pas 3-libre, nous recommençons cette procédure avec son β_3 .

La figure 2.37 illustre ce propos avec la face incidente aux volumes Vol_1 et Vol_2 , que nous nommons F . Cette dernière est constituée des brins a à d et 1 à 4. Pour parcourir l'ensemble des brins incidents à F , nous prenons l'un d'entre eux : par exemple, le brin a . Puis, comme mentionné précédemment, nous parcourons les brins b , c et d en appliquant l'opérateur β_1 . Notons que les brins de a à d sont incidents au même volume Vol_1 . Comme a n'est pas 3-libre, nous prenons le brin $1 = \beta_3(d)$, incident au volume adjacent Vol_2 . Nous récupérons ensuite à partir de ce dernier, les brins 2, 3 et 4, de la même manière que précédemment.

- Pour parcourir les brins composant une 1-cellule a , nous commençons par récupérer un brin b incident à a . La solution proposée consiste à récupérer un deuxième brin incident à a en appliquant β_2 sur b . Puis, pour chacun de ces deux brins incidents, nous appliquons β_3 à condition qu'il ne soit pas 3-libre. Nous procédons alors de la même manière sur les brins obtenus en gérant les redondances.

La figure 2.35(b) montre la décomposition de la 1-cellule a_4 (cf. la 2.35(a)) en six brins cerclés de jaune dans (b). Pour parcourir l'ensemble des brins composant a_4 , il faut partir d'un brin b incident à cette arête. Puis, grâce au pointeur β_2 appliqué sur b , on obtient un autre brin composant également a_4 . Cet autre brin appartient



donc à la même arête et au même volume que b mais pas à la même face. Le pointeur β_3 permet quand à lui d'obtenir un autre brin appartenant également à la même arête et à la même face que b mais pas au même volume.

- Enfin, nous présentons maintenant comment récupérer l'ensemble des brins incidents à une 0-cellule s . Cette explication est illustrée par la figure 2.37. Nous commençons tout d'abord par récupérer un brin partant de s . Dans notre exemple, nous prenons un des brins représentés en pointillés. En effet, ces derniers correspondent à l'ensemble des brins incidents à s . Nous faisons le choix du brin b . Puis, nous appliquons alors l'opérateur β_{21} sur b afin d'obtenir un autre brin également incident à s et appartenant au même volume que b mais pas à la même arête ni à la même face. Dans cet exemple, nous obtenons $e = \beta_{21}(b)$. Le processus est répété jusqu'à obtenir tous les brins incidents à s pour un volume incident donné (dans notre exemple Vol_1). Nous procédons ensuite de la même manière pour l'ensemble des volumes incidents à s . Pour cela, il convient d'appliquer β_{32} à partir de chaque brin partant de s et incident au volume de départ, à condition que ce dernier ne soit pas 3-libre. Dans notre exemple, nous passons au volume Vol_2 , en appliquant β_{32} à b , nous donnant le brin 5. Puis de 5, nous récupérons 6 grâce à l'opérateur β_{21} et ainsi de suite.

Opérations de découure, d'insertion et de suppression Nous finissons cette présentation des cartes combinatoires en précisant un aspect de leur définition relatif aux opérations de découure, d'insertion et de suppression. L'introduction de ces opérations est nécessaire à la compréhension des contributions explicitées dans la suite de ce manuscrit.

L'opération de découure L'opération de i -découure ($i \in \{2, 3, \dots, n\}$), permet de découure deux i -cellules cousues selon une $(i - 1)$ -cellule commune. Elle consiste en la suppression des liens β_i entre tous les brins appartenant à la $(i - 1)$ -cellule commune. Suite à l'application de cette opération, la $(i - 1)$ -cellule initiale est divisée en deux $(i - 1)$ -cellules et n'est plus commune à deux i -cellules [Dam10].

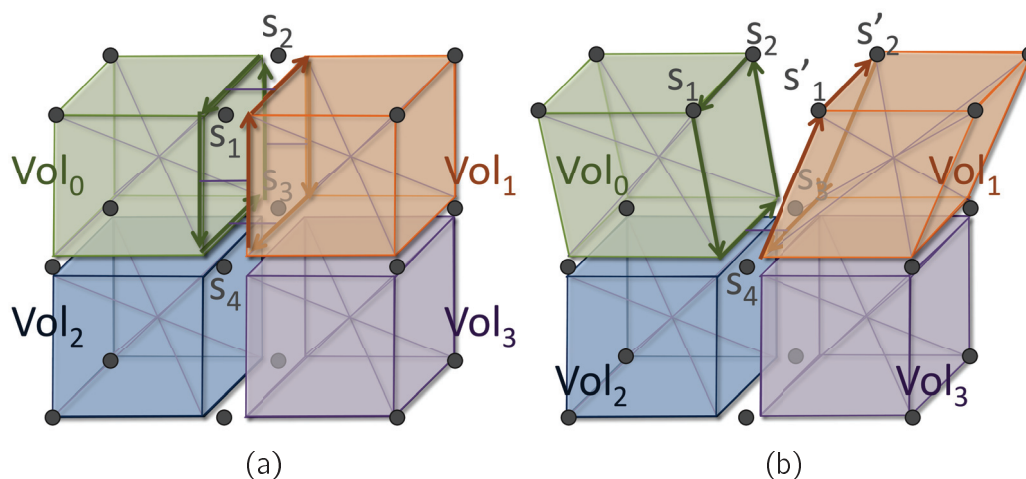


Figure 2.38 – Un exemple de 3-découure. (a) Un modèle 3D composé de 4 hexaèdres : Vol_0 , Vol_1 , Vol_2 et Vol_3 , cousus ensemble par β_3 . La face commune à Vol_0 et Vol_1 est décrite par les sommets : s_1 , s_2 , s_3 et s_4 . (b) Le modèle de (a) après une opération de 3-découure entre les volumes Vol_0 et Vol_1 . Les sommets : s_1 , s_2 , s_3 et s_4 ont été scindés en s'_1 , s'_2 , s'_3 et s'_4 . Les liaisons β_3 entre les brins incidents à Vol_0 et Vol_1 , ont été supprimées.

La figure 2.38 illustre une opération de 3-décousure. La figure 2.38(a) présente un objet 3D composé de quatre hexaèdres : Vol_0 , Vol_1 , Vol_2 et Vol_3 . La face F commune aux volumes Vol_0 et Vol_1 est décrite par les sommets (s_1, s_2, s_3, s_4) . Seuls les brins incidents à F sont représentés. L'opérateur de 3-décousure va décoller les 3-cellules Vol_0 et Vol_1 selon la 2-cellule F . Suite à cette opération, F est scindée en deux. La figure 2.38(b) présente la configuration du modèle 3D après l'opération. Les sommets s_1 et s_2 ont ainsi été dupliqués ainsi que les arêtes (s_4, s_1) , (s_1, s_2) et (s_2, s_3) .

Il faut cependant noter que les sommets s_3 et s_4 et l'arête (s_3, s_4) ne sont pas dupliqués. En effet, ils sont toujours connectés ensemble par les deux 3-cellules Vol_2 et Vol_3 . Ainsi, quatre β_3 ont été supprimés pendant cette 3-décousure.

L'opération d'insertion Lorsqu'un sommet est inséré dans une arête provoquant son scindement en deux parties, l'ensemble des brins incidents à l'arête en question sont scindés. Cet aspect est également valable en dimension supérieure avec par exemple, l'insertion d'une arête dans une face, provoquant son scindement en deux parties. Si cette face est partagée par deux volumes incidents alors les deux cycles de brins incidents aux deux volumes incidents sont scindés de manière symétrique par cette insertion d'arête. Cette propriété permet de garantir la validité topologique de la carte combinatoire suite à une insertion.

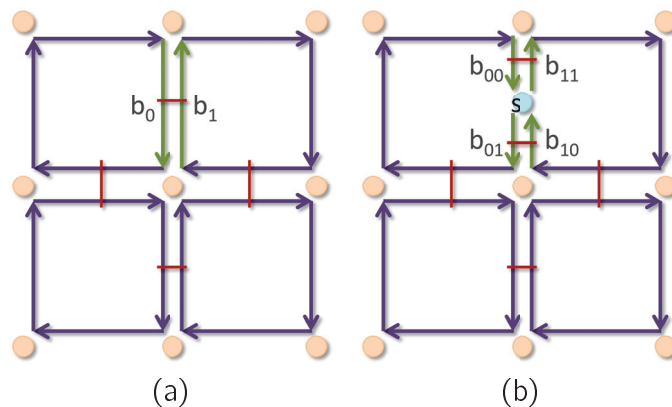
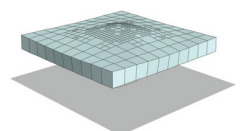


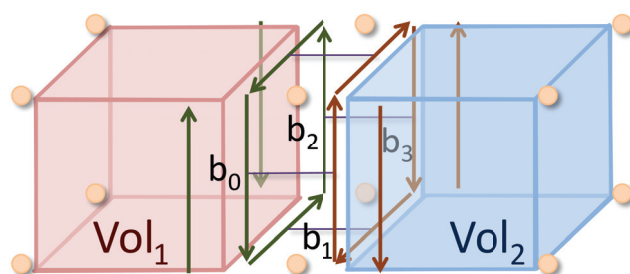
Figure 2.39 – Un exemple de l'opérateur d'insertion d'un sommet dans une arête incidente à deux quadrilatères. (a) Un objet 2D constitué de quatre quadrilatères cousus ensemble par β_2 . (b) L'objet 2D après l'insertion du sommet s dans l'arête incidente aux brins b_0 et b_1 . Le brin b_0 a été scindé en b_{00} et b_{01} ; et le brin b_1 en b_{10} et b_{11} .

La figure 2.39 illustre l'insertion d'un sommet dans une arête pour un modèle 2D et la figure 2.40 explicite l'insertion d'un sommet dans une arête et d'une arête dans une face pour un modèle 3D. Elles montrent que tous les brins incidents à l'élément scindé sont scindés de manière symétrique.

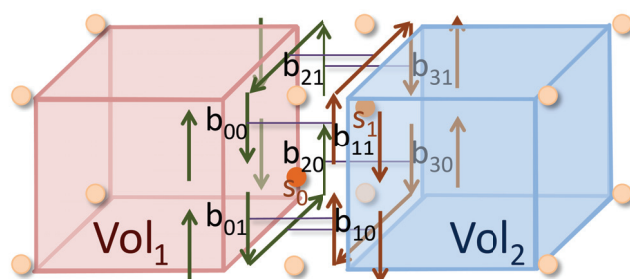
L'opération de suppression De manière intuitive, l'opération de suppression appliquée à une i -cellule consiste en la suppression de cette dernière, puis en la fusion entre elles des deux $(i + 1)$ -cellules qui lui étaient incidentes. Pour une carte combinatoire de dimension n , l'opération de suppression est définie pour les cellules de dimensions 0 à $n - 1$. Notons que l'opération de suppression est l'opération inverse de celle de l'insertion.

La figure 2.41 illustre, pour le cas 2D, le fonctionnement de l'opérateur de suppression. La figure 2.41(a) correspond à un modèle 2D constitué de deux quadrilatères adjacents F_1 et F_2 . Dans la figure 2.41(b), ces deux derniers vont être fusionnés suite à la suppression de

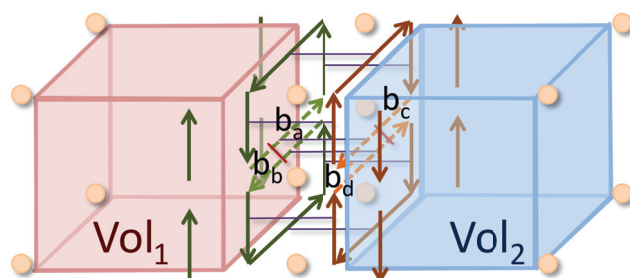




(a)



(b)



(c)

Figure 2.40 – Un exemple de l'opérateur d'insertion d'un sommet (resp. arête) dans une arête (resp. face) incidente à deux hexaèdres. (a) Un objet 3D constitué de deux hexaèdres adjacents : Vol_1 et Vol_2 . (b) L'objet 3D après l'insertion du sommet s_0 dans l'arête incidente aux brins b_0 et b_1 et l'insertion du sommet s_1 dans l'arête incidente aux brins b_2 et b_3 . Le brin b_0 a été scindé en b_{00} et b_{01} ; le brin b_1 en b_{10} et b_{11} ; le brin b_2 en b_{20} et b_{21} ; et le brin b_3 en b_{30} et b_{31} . (c) L'objet 3D après l'insertion d'une arête dans la face F commune aux deux volumes Vol_1 et Vol_2 . L'insertion de l'arête provoque le scindement transversal de F . Elle engendre la création de quatre brins : b_a , b_b , b_c et b_d . Les deux cycles de brins générés incidents à Vol_1 (resp. Vol_2) sont reliés par β_2 au niveau des brins b_a et b_b (resp. b_c et b_d). Enfin, l'ensemble des brins composant les deux nouvelles sous-faces et incidents à Vol_1 , sont reliés deux à deux par β_3 , avec l'ensemble des brins composant les deux nouvelles sous-faces et incidents à Vol_2 .

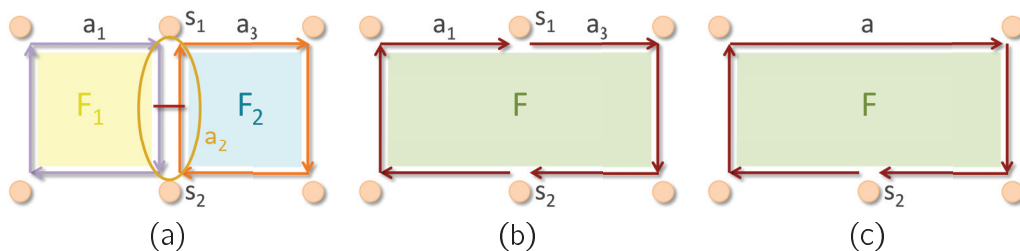


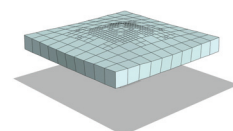
Figure 2.41 – Un exemple de l’opérateur de suppression d’un sommet et d’une arête incidente à deux faces. (a) Un objet 2D constitué de deux éléments adjacents F_1 et F_2 , cousus ensemble par β_2 . (b) L’objet 2D après la suppression de l’arête a_2 . Les deux faces F_1 et F_2 sont alors fusionnées en une seule même face F . (c) L’objet 2D après la suppression du sommet s_1 . Les deux arêtes a_1 et a_3 sont fusionnées en la même arête a .

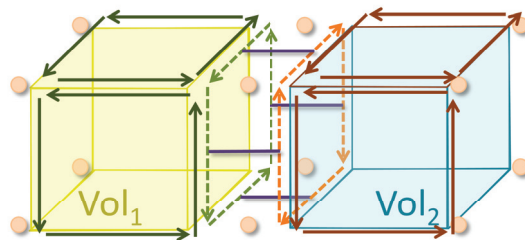
leur arête commune a_2 . Enfin, la figure 2.41(c) montre la fusion de deux arêtes suite à la suppression du sommet s_1 qui leur était commun.

Dans la figure 2.41(a), le sommet s_1 est incident aux arêtes a_1 , a_2 et a_3 . s_1 est donc de degré 3. L’arête a_1 est incidente à la face F_1 et est donc de degré 1, contrairement à l’arête a_2 qui est de degré 2.

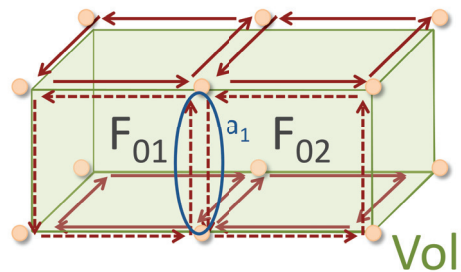
Notons que la suppression n’est possible que pour des cellules de degré inférieur ou égal à deux. Dans le modèle présenté dans la figure 2.41(a), comme le sommet s_1 est de degré 3, il n’est pas possible de le supprimer. Par contre, suite à la suppression de a_2 (cf. figure 2.41(b)), le sommet s_1 devient alors de degré 2. Il peut ainsi être supprimé.

La figure 2.42 explicite l’opérateur de suppression pour un modèle 3D. La figure 2.42(a) correspond à un modèle 3D constitué de deux hexaèdres adjacents. Dans la figure 2.42(b), ces deux derniers vont être fusionnés suite à la suppression de leur face commune. Enfin, la figure 2.42(c) montre la fusion de deux faces suite à la suppression de l’arête qui leur était commune. Nous ne pouvons pas supprimer le sommet s_2 car il est de degré 3. Dans la figure 2.43(a), on supprime alors a_4 permettant ainsi la suppression de s_2 (cf. les figures 2.43(b) et (c)).

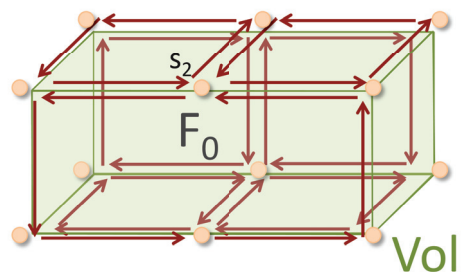




(a)

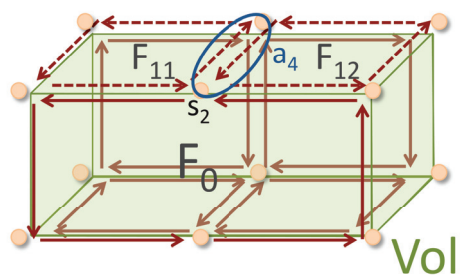


(b)

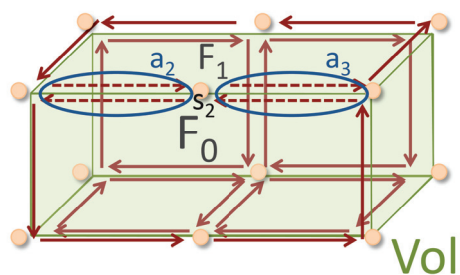


(c)

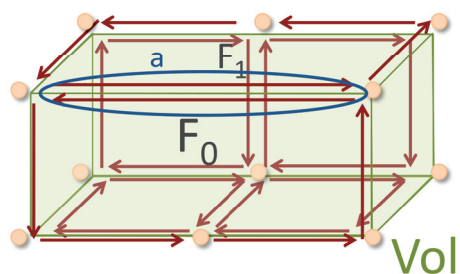
Figure 2.42 – Un exemple de l'opérateur de suppression d'une face incidente à deux volumes et d'une arête incidente à deux faces. (a) Un objet 3D constitué de deux éléments adjacents : Vol_1 et Vol_2 cousus ensemble par β_3 . (b) L'objet 3D après la suppression de la face F , commune aux deux hexaèdres. Les brins incidents à cette face sont représentés en pointillés. Les deux hexaèdres : Vol_1 et Vol_2 sont alors fusionnés en un seul même volume Vol . (c) L'objet 3D après la suppression de l'arête a_1 . Les deux faces F_{01} et F_{02} sont fusionnées en une seule même face F_0 .



(a)

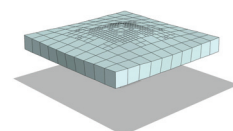


(b)



(c)

Figure 2.43 – Un exemple de l'opérateur de suppression d'un sommet incident à deux arêtes et d'une arête incidente à deux faces, dans le cadre d'un maillage 3D. (a) Un objet 3D correspondant à celui de la figure 2.42(c). (b) L'objet 3D après la suppression de l'arête a_4 . Les deux faces : F_{11} et F_{12} sont alors fusionnées en une seule même face F_1 . (c) L'objet 3D après la suppression du sommet s_2 . Les deux arêtes a_2 et a_3 sont fusionnées en une seule même arête a .



Conclusion sur les cartes combinatoires Pour conclure cette présentation des cartes combinatoires, nous rappelons que leurs intérêts principaux sont :

- Leur définition par la notion de brins et de pointeurs β_i , permettant de représenter des objets de dimension quelconque.
- La définition des contraintes garantissant la validité topologique des objets représentés.
- L'existence de nombreuses opérations de modifications, qui nous ont été utiles dans nos travaux.

Complexes cellulaires linéaires

Les **complexes linéaires cellulaires (Linear Cell Complexes en anglais)**, que nous abrégons dans la suite de ce manuscrit **LCC**, sont une extension des cartes combinatoires permettant de les plonger dans un espace géométrique. En effet, de nombreuses applications requièrent une description géométrique des objets, en plus de leur représentation topologique.

Ce plongement est réalisé de manière **linéaire**, c'est-à-dire que la géométrie d'un sommet correspond à une coordonnée nD . Une arête correspond alors à un segment, une face à une portion de plan et ainsi de suite pour les dimensions supérieures. Dans le cas d'un plongement linéaire, toutes les i -cellules ($\forall i : 1 \leq i \leq n$) sont implicitement plongées et déduites du plongement des sommets. Ce ne serait pas le cas pour un plongement non linéaire, où le plongement des sommets ne suffirait pas à déduire le plongement des autres cellules. Notons que chaque i -cellule topologique doit être plongée dans un objet géométrique ouvert de dimension i , car le plongement du bord de la i -cellule topologique est représenté par le plongement de ses $(i - 1)$ -cellules incidentes.

Cette distinction entre la topologie et la géométrie est un des points forts des cartes combinatoires. En effet, selon les algorithmes, certains traitements seront effectués en utilisant uniquement la topologie, sans considération de la géométrie correspondante (par exemple, l'insertion d'arête ou le calcul de propriétés topologiques comme la caractéristique d'Euler-Poincaré), et à l'inverse, en employant uniquement la géométrie, sans modifier la carte combinatoire (comme par exemple la translation d'un objet). Enfin, des traitements employant la topologie et la géométrie vont bénéficier de cette séparation par une simplification des accès aux informations et des modifications.

Cartes généralisées

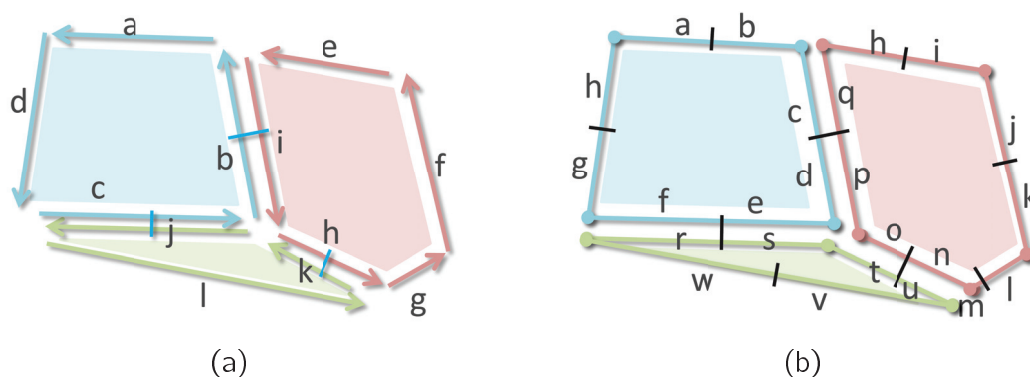


Figure 2.44 – Un objet 2D représenté avec une carte combinatoire puis avec une carte généralisée. (a) La carte combinatoire. (b) La carte généralisée.

Le modèle topologique **des cartes généralisées** est une extension de celui des cartes combinatoires. Il permet de représenter des objets étant des quasi-variétés orientables ou non-orientables avec ou sans bord [Lie89, Lie91, Lie94, Dam10]. Leur principal avantage réside dans leur définition qui a pour propriété d'être homogène pour toutes les dimensions contrairement aux cartes combinatoires (β_1 est une permutation alors que tous les autres β_i sont des involutions). Cette homogénéité simplifie les définitions, l'écriture des algorithmes et rend la définition des cartes généralisées directement valide pour les objets avec ou sans bords.

La définition formelle des cartes généralisées est la suivante :

Définition 3

Soit $n \geq 0$. Une nG -carte (carte généralisée de dimension n) est une algèbre $\mathcal{G} = (B, \alpha_0, \dots, \alpha_n)$ où :

1. B est un ensemble fini de brins ;
2. $\forall i : 0 \leq i \leq n : \alpha_i$ est une involution sur B ;
3. $\forall i \in \{0, \dots, n-2\}, \forall j \in \{i+2, \dots, n\} : \alpha_i \circ \alpha_j$ est une involution.

Ainsi, contrairement aux cartes combinatoires, tous les α_i sont des involutions. La figure 2.44 montre un objet 2D représenté par une carte combinatoire et une carte généralisée, permettant ainsi la comparaison entre les deux modèles. Un brin d'une carte généralisée est représenté par un segment ayant pour extrémité un disque et un petit segment orthogonal. Deux brins reliés par α_0 sont alignés et sont séparés par un petit segment perpendiculaire. Par exemple, dans la figure 2.44(b), $\alpha_0(a) = b$. Deux brins reliés par α_1 sont séparés par un disque, par exemple, $\alpha_1(b) = c$. Enfin, deux brins reliés par α_2 sont parallèles et partagent le même petit segment orthogonal, par exemple, $\alpha_2(c) = q$.

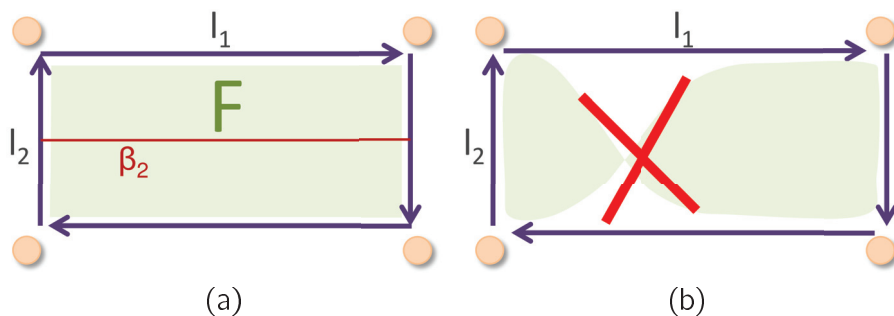
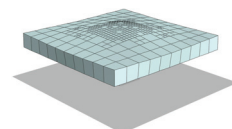


Figure 2.45 – L'impossibilité de représenter un objet 2D non-orientable avec une carte combinatoire. (a) Une carte combinatoire 2D représentant un ruban classique. (b) Le modèle interdisant le vrillement.

Nous ne détaillerons pas ce modèle plus en détails, mais notons sa capacité à représenter des objets non orientés comme le ruban de Möbius en 2D. Considérons un rectangle de dimension l_1 et l_2 . Pour obtenir un ruban de Möbius, un côté du rectangle de longueur l_1 doit être collé avec l'autre côté de longueur l_1 , en effectuant une torsion. Les cartes combinatoires représentent ce rectangle avec 4 brins de même orientation. Pour obtenir un ruban de Möbius, il faut vriller le brin ayant une géométrie de longueur l_1 . Or, les cartes combinatoires interdisent cette opération (cf. la figure 2.45). Contrairement aux cartes combinatoires, les cartes généralisées permettent de couder de deux manières différentes : avec ou sans torsion (cf. la figure 2.46).



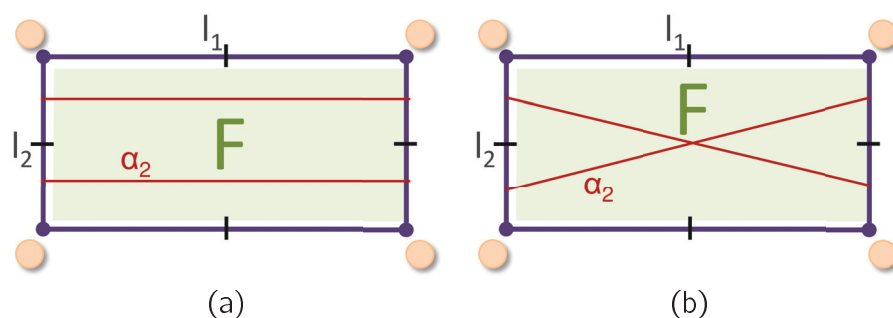


Figure 2.46 – La possibilité de représenter des objets 2D non-orientables avec une carte généralisée. (a) Une carte généralisée 2D représentant un ruban classique. (b) Une carte généralisée 2D représentant un ruban de Möbius.

2.3.4 Travaux relatifs aux structures de données topologiques

De nombreux travaux relevant du domaine graphique ou de la génération d'images, emploient une structure de données topologique sous-jacente.

Nouvelles structure de données topologiques

Régulièrement, de nouveaux travaux présentent l'élaboration de structure de données topologiques. Ces nouvelles structures répondent le plus souvent à des applications précises.

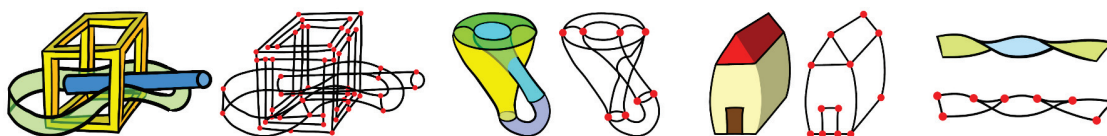


Figure 2.47 – Exemple de la définition d'un nouveau modèle topologique. De gauche à droite : trois résultats obtenus avec leurs nouveaux outils de dessin vectoriel, accompagné d'une représentation de la topologie sous-jacente [DRVDP14].

Nous citons par exemple le récent article de Dalstein *et al.* [DRVDP14] (*cf.* la figure 2.47). Ils proposent une nouvelle structure de données topologique surfacique : VGC (Vector Graphics Complexes). Cette dernière permet d'améliorer la conception de dessins vectoriels, en gérant le partage d'arête entre deux faces, les jointures multiples et le chevauchement. Ils définissent également différents opérateurs d'édition topologique comme coller des sommets ou des arêtes, découper des arêtes ou des faces ainsi que les opérations inverses.

Exemple de travaux employant une structure de données topologique surfacique

Parmi les travaux utilisant une structure de données topologique, nous pouvons citer ceux de Kobbelt *et al.* [KCS98] (*cf.* la figure 2.48). Ces derniers proposent de simplifier des maillages surfaciques très détaillés, qui sont lourds en mémoire. Pour cela, ils modifient de manière incrémentale le maillage d'origine en appliquant de simples modifications topologiques. Entre un maillage obtenu à l'étape i et le maillage $i + 1$ obtenu après modification topologique, ils calculent une mesure de distance soumise à un seuil de tolérance ainsi qu'un critère permettant d'évaluer la qualité du maillage $i + 1$ reposant sur la courbure locale ou encore sur la géométrie (angles et longueurs) des triangles composant le maillage.

Braquelaire et Brun *et al.* [BMD03, BD99] proposent une méthode employant une structure de données topologique afin de segmenter une image en plusieurs ensembles connectés

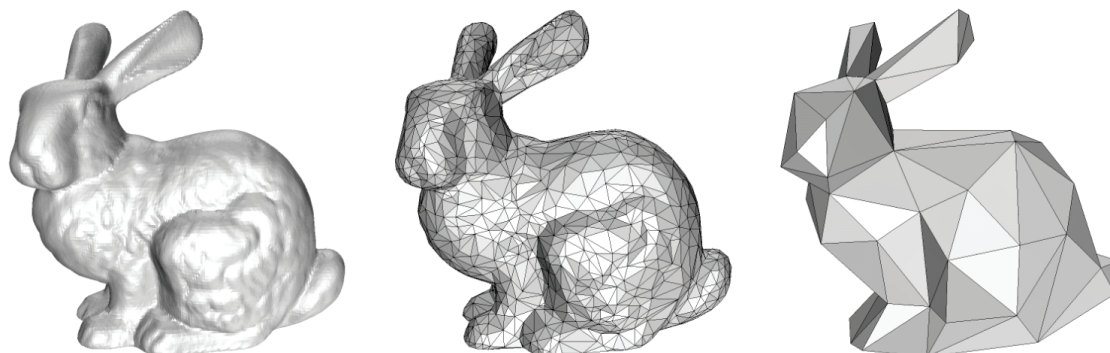


Figure 2.48 – Exemple de l'utilisation de la structure de données des demi-arêtes pour la simplification de maillage. De gauche à droite : Le modèle de référence, une première simplification du maillage et une simplification beaucoup plus importante du maillage [KCS98].

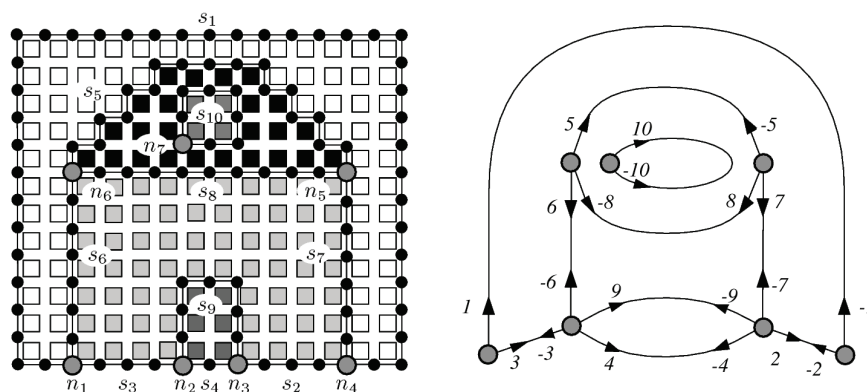
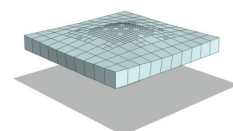


Figure 2.49 – Exemple de l'utilisation d'une carte combinatoire 2D pour la segmentation d'images. A gauche : Une image segmentée en cinq régions. Les disques noirs correspondent aux points de la bordure des différentes régions, ils sont reliés par de courts segments. Les disques gris sont les nœuds de chaque région. A droite : Les deux cartes combinatoires associées à la figure de gauche [BMD03].



de pixels. L'objectif est de modifier la partition de l'image jusqu'à satisfaction d'un critère d'homogénéité donné. Ils utilisent une approche mixant une méthode haut-bas (top-down en anglais) et bas-haut (bottom-up en anglais) consistant respectivement à débiter avec des régions grossières et à les diviser itérativement en plus petites régions, ou à débiter avec de petites régions qui vont être fusionnées jusqu'à satisfaction d'un critère. Leur méthode consiste grossièrement à placer des disques noirs entre chaque pixel du pourtour de l'image à segmenter (cf. la figure 2.49). Si un de ces disques se trouve entre deux pixels n'ayant pas la même caractéristique (dépendant du critère donné) alors il sera un nœud. L'ensemble de ces nœuds permet de définir une première carte combinatoire. Puis, à partir de ces nœuds, des disques noirs vont être créés à la frontière des deux régions que séparait le premier nœud. Si une autre région est rencontrée alors un nouveau nœud est créé. La carte combinatoire est ainsi modifiée itérativement en rajoutant des nœuds et des arêtes entre ces derniers, jusqu'à obtenir une segmentation en accord avec le critère donné.

Exemple de travaux employant une structure de données topologique 3D.

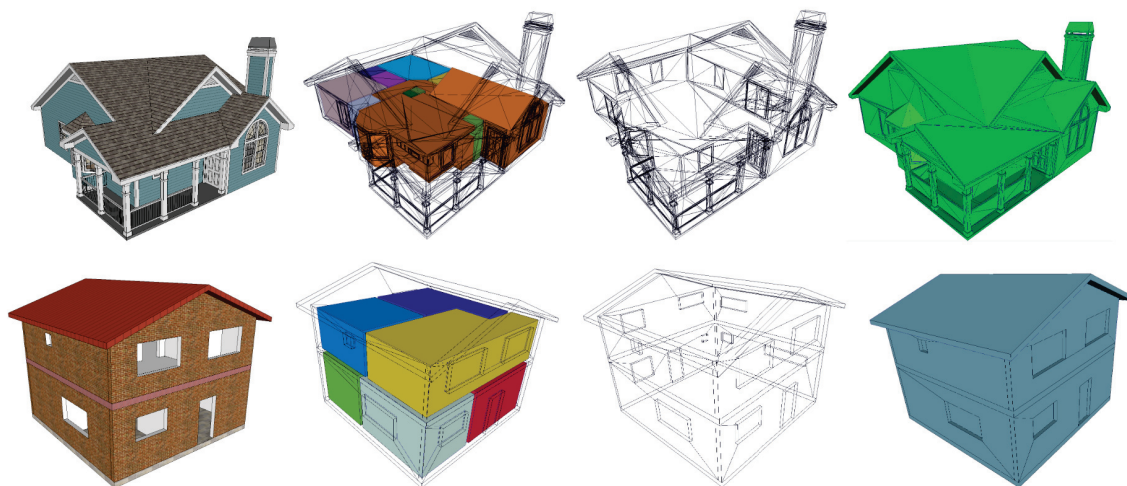


Figure 2.50 – Exemple de l'utilisation de cartes combinatoires permettant la réalisation d'un modèle unifié de bâtiments pour la simulation acoustique et énergétique. De gauche à droite : Le modèle original texturé, les volumes intérieurs, l'enveloppe extérieure représentée en filaire et une simplification de l'enveloppe extérieure [DDV14].

Les travaux de Diakité *et al.* [DDV14] sont un exemple d'utilisation de carte combinatoire (cf. la figure 2.50). Ils proposent une méthode prenant en entrée seulement la géométrie d'un bâtiment 3D et donnant en sortie la topologie intérieure et extérieure de ce dernier. Elle permet ainsi d'extraire différents niveaux de détails de cette description topologique. Avec les cartes combinatoires, ils proposent un modèle commun pour représenter les bâtiments selon les besoins des acteurs de ce domaine comme les architectes ou encore les ingénieurs. En effet, ces derniers n'ont pas les mêmes attentes vis à vis d'une modélisation d'un bâtiment. Les architectes créent des modèles de bâtiments avec beaucoup de détails visuels, contrairement aux modèles requis par les ingénieurs souhaitant, par exemple, faire des simulations acoustiques ou énergétiques. Les cartes combinatoires leur permettent entre autres d'avoir accès à toutes les relations d'adjacence et d'incidence entre les différentes cellules construites composant le modèle du bâtiment, de les taguer comme étant des murs, des sols, des plafonds, ... mais également de procéder à des simplifications de la représentation topologique obtenue.

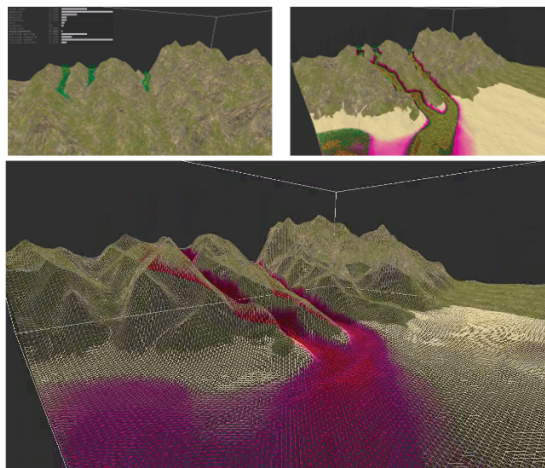


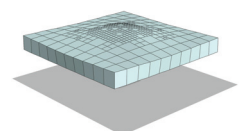
Figure 2.51 – Exemple de l'utilisation de cartes généralisées permettant la simulation d'évolution géomorphologiques. De gauche à droite et de haut en bas : Un ruisseau simulé par un modèle physique de fluide qui débute sur une partie élevée en hauteur d'un terrain représenté par une carte généralisée. Le ruisseau dévale la pente en rentrant en collision avec le terrain, entraînant alors son érosion. Les zones érodées sont représentées en violet [Bé13].

Bézin *et al.* [Bé13,BCS+11,BPC+11] emploient les cartes généralisées 3D afin de simuler les évolutions géomorphologiques (l'érosion hydraulique et la sédimentation) d'un terrain en 3D (*cf.* la figure 2.51). Ce dernier est subdivisé en différents volumes. Ils emploient également un modèle physique de fluide. Ils détectent les collisions entre une particule de fluide et les triangles du terrain. La collision avec un triangle entraîne l'érosion de ce dernier dépendant d'un coefficient d'érosion, du plan d'inclinaison du triangle et de la vitesse de la particule collisionnée. L'érosion se traduit par un déplacement vers le bas des sommets du maillage du terrain. Pour cela, ils définissent différentes opérations de modifications topologiques et géométriques complexes, garantissant la cohérence topologique grâce au modèle topologique des cartes généralisées. Ces opérations topologiques gèrent la déformation de maillages volumiques soumis à des événements générant des incohérences topologiques et géométriques. Ces opérations permettent par exemple, d'élargir le lit d'une rivière en creusant plusieurs strates de volumes ou encore de créer une arche au milieu de plusieurs strates de volumes.

Exemple de travaux employant une structure de données topologique pour la simulation physique.

Dans ce manuscrit, nous nous intéressons à l'emploi d'une structure de données topologique pour la simulation physique. C'est pour cette raison que nous détaillons dans cette partie, les différents travaux œuvrant dans ce sens.

Nous citons l'article de Meseure *et al.* [MDS10a] qui proposent un couplage du système masses-ressorts et d'une carte généralisée, pour la simulation d'objets déformables. L'objet à simuler est modélisé par un ensemble de triangles ou de tétraèdres en 3D. Pour la structure de données topologique, ils ont recours à la librairie MOKA [mok]. Ils présentent l'association des informations relatives au modèle physique à la carte généralisée, les opérations topologiques permettant la découpe d'arêtes ou de faces et la mise à jour des informations mécaniques suite à des changements topologiques. Nous précisons que les changements topologiques surviennent au cours de la simulation. Malheureusement, leur méthode est quatre fois plus coûteuse en temps qu'avec l'emploi d'une structure de données topologique naïve. Cet article présente néanmoins un premier couplage d'un modèle physique avec un modèle topologique



complexe.

Dans l'article [MDS10b], Meseure *et al.* présentent l'intérêt de structures de données topologique complexes comme les cartes généralisées, pour l'animation physique. Contrairement aux structures de données topologiques naïves, les cartes généralisées permettent de contrôler la topologie du modèle, en garantissant la propriété de quasi-variété des maillages volumiques. Plus précisément, ils évitent la création de faces pendantes ou de volumes seulement reliés à un sommet ou une arête. Les changements topologiques sont alors gérés de manière robuste, grâce aux définitions mathématiques à la base des cartes généralisées. Or, l'emploi d'une telle structure de données est présentée comme lourde en mémoire et coûteuse en temps de calcul. Malheureusement, pour optimiser les temps de calcul, ils doivent mettre en place une structure additionnelle contenant toutes les informations associées à chaque cellule d'une dimension donnée ainsi qu'un pointeur sur le brin incident à la cellule correspondante. Cette solution améliore la vitesse d'exécution de leur méthode mais entraîne la gestion d'une structure de données complexe et coûteuse au niveau de ses opérations de mise à jour. En effet, cette structure additionnelle doit mettre à jour ses données après chaque changement topologique effectué sur la carte généralisée.

Dans le même sens, Darles *et al.* [DKS⁺11] ont montré l'intérêt du couplage des cartes généralisées avec un modèle physique de type **masses-interactions**. Les cartes généralisées permettent de modéliser la topologie du modèle. Le modèle physique modélise quant à lui le mouvement. Pour la partie relative au modèle physique, ils emploient le framework CORDIS-ANIMA [COR]. Le modèle masses-interactions est un modèle physique **sans maillage**, implémenté dans CORDIS-ANIMA. Le modèle physique de ce framework permet de représenter des phénomènes physiques en construisant un réseau interconnectant des composants appelés **MAT** et **LIA**. Les composants MAT modélisent l'inertie du phénomène physique. Les composants LIA correspondent, quant à eux, aux interactions entre les composants MAT. Dans cet article, ils associent les composants MAT avec les cellules de la carte généralisée. Ils gèrent la modification du modèle topologique en fonction de l'état du modèle physique. Ils illustrent leur méthode au travers des exemples de changements topologiques que sont le déchirement, la fracture ainsi que la fusion d'un objet. Il est difficile d'appréhender la manière dont ils gèrent la mise à jour des paramètres de leur modèle physique et de l'association avec la carte généralisée, en fonction des changements topologiques.

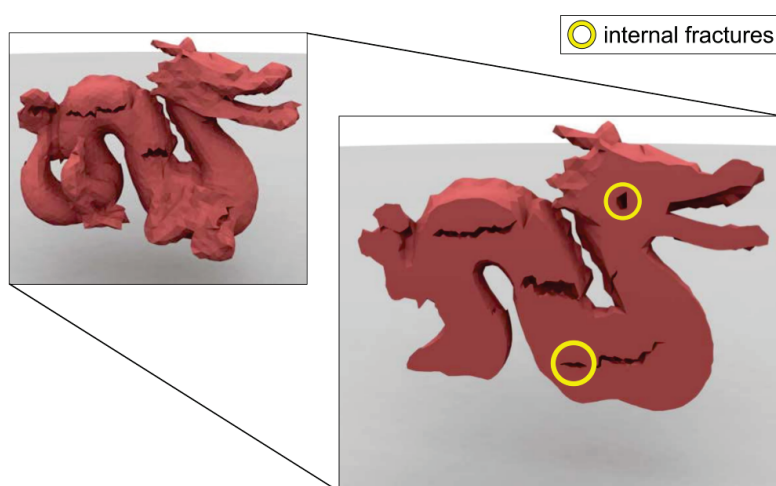


Figure 2.52 – Exemple de l'utilisation de cartes combinatoires pour la simulation de changements topologiques de type fracture, durant une animation. Un maillage volumique tétraédrique représentant un dragon. Nous observons la présence de fractures notamment internes [JAD⁺12b].

De la même manière, Jund *et al.* [JAD⁺12a, JAD⁺12b] ont également proposé d'associer un modèle particulière, permettant des mouvements basés points, avec une carte combinatoire (*cf.* la figure 2.52). Ils emploient le framework CORDIS-ANIMA [COR], pour générer les mouvements et la librairie CGoGN [CGo], pour son implémentation des cartes combinatoires. Leur méthode permet d'associer n'importe quelle forme volumique à n'importe quel mouvement basé points et de contrôler les changements topologiques. Leur méthode consiste tout d'abord à associer les points (fonctions d'évolution 3D) du système particulière, aux cellules de la carte combinatoire. Chaque sommet de la carte combinatoire est influencé par un ensemble de points. Le mouvement est ensuite appliqué au modèle, provoquant le déplacement des sommets de la carte combinatoire. Enfin, au cours de la simulation, des changements topologiques peuvent survenir selon des critères géométriques, entraînant par exemple la séparation de cellules topologiques. L'association entre le modèle physique et le modèle topologique doit alors être mise à jour. Notons également que la carte combinatoire permet la visualisation du modèle. Comme le présente cet article, le changement topologique peut être compliqué à gérer suivant le positionnement des zones d'influence des points par rapport aux cellules topologiques.

Exemple de travaux employant une structure de données topologique pour la multi-résolution.

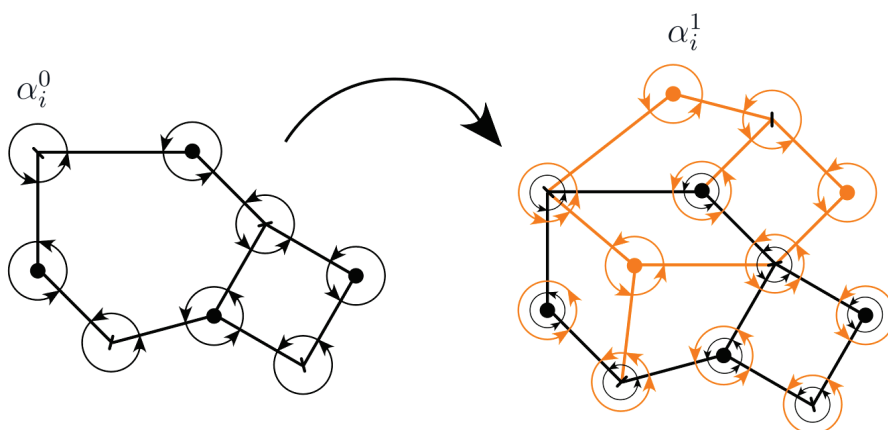
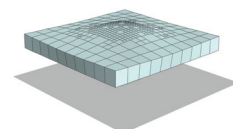


Figure 2.53 – Exemple de carte multi-résolution à deux niveaux consécutifs. A gauche : Carte au niveau 0, les brins sont liés par des permutations α_i^0 . A droite : Carte au niveau 1 obtenue à partir de la carte au niveau 0 et de l'ajout de nouveaux brins. L'ensemble des nouveaux brins sont liés par des permutations α_i^1 , tout comme les brins préexistants qui sont en plus reliés par α_i^0 [Kra08].

Dans ce sens, Kraemer [Kra08] présente dans sa thèse l'élaboration d'un nouveau modèle de représentation **multi-résolution** à partir des cartes combinatoires. Cette extension multi-résolution des cartes combinatoires permet la représentation d'un objet comme une hiérarchique de cartes combinatoires où chacune d'entre elles correspond à un niveau de résolution. Le modèle multi-résolution proposé possède la même genericité que le modèle classique des cartes combinatoires. C'est-à-dire qu'il est générique par rapport à la dimension et au type de l'objet représenté (orientable, variété, ...) ou encore au type des cellules dont le modèle est composé. Plus précisément, ce modèle consiste en l'élaboration d'une carte combinatoire initiale qui se verra rajouter par la suite des brins pour former un nouveau niveau de résolution (*cf.* la figure 2.53). L'application d'opérateurs comme par exemple, l'insertion d'un sommet dans une arête ou encore la subdivision d'une face en deux sous-faces entraîne l'ajout de ces brins. Le modèle multi-résolution consiste grossièrement en une imbrication



d'ensembles de brins, les uns dans les autres. L'ajout de brins à un ensemble de brins de niveau n donne un nouvel ensemble de brins de niveau $n + 1$. Les brins ajoutés sont reliés aux brins préexistants du niveau inférieur. En effet, les brins préexistants se voient attribuer des permutations de niveau $n + 1$ en plus de leurs permutations n déjà présentes. Enfin, Kraemer présente dans sa thèse différentes applications de son modèle multi-résolution comme les **surfaces de subdivision multi-résolution** (cf. la figure 2.54) ou encore la **compression de maillage**. Une surface de subdivision permet de lisser un maillage en subdivisant les éléments le composant tout en modifiant les coordonnées géométriques de ses points. Il existe différents type de schémas de subdivision comme par exemple celui de Catmull-Clark, de Loop ou encore la subdivision $\sqrt{3}$. Untereiner *et al.* [UCB13, Unt13] emploient ce modèle multi-résolution pour l'élaboration d'une méthode **adaptative** utilisée dans le cadre du développement d'un simulateur electro-thermique. Ils associent un modèle FEM au modèle multi-résolution mais ces derniers restent distincts. De plus, les changements topologiques n'interviennent pas pendant la simulation.

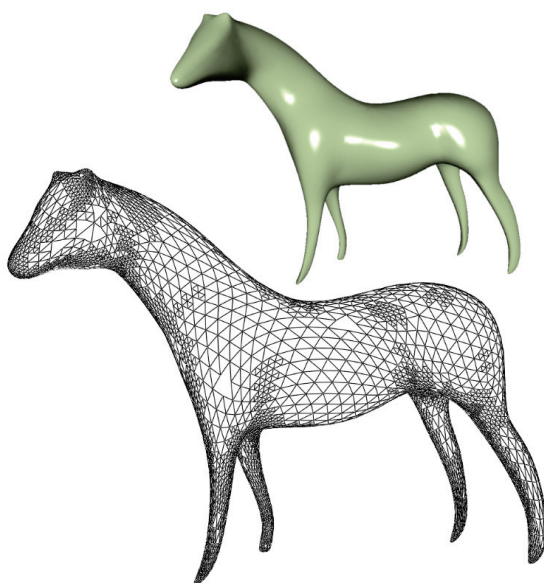


Figure 2.54 – Exemple de maillage obtenu par une subdivision adaptative. Maillage obtenu par subdivision d'un maillage triangulaire initial, de façon adaptative en utilisant le schéma de subdivision de Loop et une estimation de la courbure de la surface comme critère de subdivision [Kra08].

2.3.5 Conclusion sur les structures de données topologiques.

Les travaux présentés dans ce manuscrit se sont intéressés à employer une structure de données topologique pour la simulation physique d'objet déformables. Une structure de données topologique permet de stocker les informations géométriques et topologiques de la modélisation d'un objet à simuler.

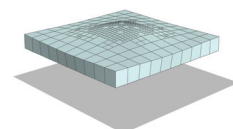
La structure de données topologique des cartes combinatoires et celle des cartes généralisées, permettent de représenter des objets surfaciques comme volumiques, contrairement aux structures de données topologiques exclusivement surfaciques comme la structure de données de l'arête ailée ou des demi-arêtes, qui ont été présentées précédemment. Ces structures de données topologiques nD représentent les objets par une décomposition en cellules de topologie quelconque. De plus, elles donnent accès à l'ensemble des relations d'incidence et d'adjacence entre les cellules de l'objet modélisé. Enfin, elles facilitent les changements to-

pologiques du modèle, tout en garantissant la validité topologique de ce dernier. Dans ce sens, nous citons l'article de Forest *et al.* [FDA05]. Les travaux présentés dans ce dernier, proposent une méthode de découpe qui consiste en un raffinement suivi d'une suppression des éléments se trouvant au plus près du chemin de découpe. Ils définissent un maillage tétraédrique comme étant une variété, si en tout point du maillage, ce dernier a une épaisseur strictement supérieure à zéro. Ils nomment **singularité topologique**, l'endroit du maillage ayant une épaisseur nulle. Ils présentent la nécessité de garantir la propriété de variété d'un maillage pour différentes raisons que sont par exemple le rendu ou la simulation avec une MEF. Ils proposent un algorithme complexe et lourd permettant d'éviter la génération de singularités topologiques. Ce dernier montre la nécessité de l'emploi d'une structure de données topologique pour garantir la validité topologique du maillage volumique au cours de changements topologiques comme la suppressions d'éléments.

Nous revenons maintenant sur notre choix de structure de données topologique. Contrairement aux cartes combinatoires, une carte généralisée permet de représenter des objets non orientables. De plus, la définition des cartes généralisées est homogène pour toutes les dimensions. Cette propriété permet de simplifier les opérations topologiques, comme par exemple, l'insertion d'un sommet dans une arête. Nous avons fait le choix des cartes combinatoires et plus particulièrement des complexes linéaires cellulaires, au détriment des cartes généralisées. En effet, les objets déformables que nous manipulons sont exclusivement orientables. De plus, les cartes généralisées représentent ces derniers avec deux fois plus de brins qu'une carte combinatoire. Les cartes généralisées sont alors plus lourdes en mémoire et ses algorithmes de parcours sont plus coûteux en temps de calcul. La complexité de l'implémentation des opérations topologiques des cartes combinatoires, nous a été épargnée par l'emploi de la librairie CGAL. Enfin, les complexes linéaires cellulaires nous permettent d'associer une géométrie à la modélisation de l'objet à simuler, mais également d'embarquer toutes les informations nécessaires notamment à la simulation physique.

2.4 Changements topologiques

Nous avons précisé que dans nos travaux, nous employons un modèle topologique exhaustif de type carte combinatoire, permettant de gérer les changements topologiques survenant dans un maillage de dimension quelconque, tout en garantissant sa validité topologique. Plus précisément, nos travaux présentent comment construire un modèle unifié alliant ce modèle topologique et un SMR. Puis, nous montrons que notre modèle unifié est particulièrement adapté à la gestion de changements topologiques comme ceux survenant suite à des opérations de découpe ou de raffinement de maillage. Nous présentons alors la manière dont sont gérés ces changements topologiques au niveau de notre modèle, avec entre autre la mise à jour des informations relatives au SMR. Les opérations de découpe ou de raffinement que nous proposons ne sont pas aussi élaborées que celles présentes dans la littérature. En effet, l'objectif est de démontrer par des exemples simples de découpe et de raffinement, l'intérêt de notre modèle unifié. Notre opération de découpe s'effectue le long des arêtes de nos quadrilatères ou hexaèdres. De plus, notre raffinement consiste simplement en la subdivision d'un quadrilatère en quatre sous-quadrilatères et, pour le cas 3D, la subdivision d'un hexaèdre en huit sous-hexaèdres. Nos travaux permettent également une méthode adaptative selon un critère géométrique en considérant le pourcentage de déformation des ressorts diagonaux internes à un quadrilatère ou un hexaèdre, par rapport à sa longueur initiale.



2.4.1 Multi-résolution

Nous avons vu qu'une discrétisation de l'objet réel est nécessaire à l'élaboration de son modèle virtuel. En général, plus cette discrétisation est fine, plus le modèle virtuel correspond à l'objet réel. La simulation du comportement de l'objet réel à partir de ce modèle virtuel sera alors d'autant plus précise. Par contre, plus la discrétisation est fine, plus la simulation sera coûteuse en temps. En effet, nous avons présenté auparavant la construction d'un système MEF. Ce dernier est d'autant plus coûteux à résoudre que sa taille est grande. Pour le SMR, plus il y a d'éléments, plus il y a de ressorts et de particules. En effet, il faut parcourir tous les ressorts pour accumuler leur force, puis pour chaque particule, sa vitesse doit être intégrée pour déterminer sa nouvelle position. En conclusion, il faut balancer entre une discrétisation trop fine, engendrant des temps de calculs trop importants et une discrétisation trop grossière, induisant une simulation du comportement peu précise. Ce postulat est le fait des applications temps-réel. Une solution classique à cette problématique, est la **multi-résolution**. Cette dernière concerne tous les modèles comportant différents niveaux de discrétisation, que l'on nomme également niveau de résolution. Les zones dites **raffinées**, ont un niveau de résolution plus élevé que les autres zones. Une méthode **adaptative** consiste en le raffinement et/ou le dé-raffinement au cours de la simulation du modèle multi-résolution en fonction d'un critère donné. Le raffinement d'un élément dépend d'un critère qui peut être par exemple d'ordre géométrique en considérant l'élongation des arêtes de l'élément, ou d'ordre physique en calculant l'ensemble des forces auxquelles l'élément est soumis, ou encore d'ordre matériel en déterminant les capacités de calcul disponible. Notons également que dans le domaine du rendu, les éléments d'un modèle sont plus ou moins raffinés selon leur distance à la caméra. Cette technique nommée **niveau de détail** (abrégiée **LOD** de l'acronyme anglais **Level Of Detail**), permet d'améliorer la fluidité du rendu en n'affichant qu'un nombre réduit de polygones pour les objets se situant loin de la caméra. Au contraire, un objet proche de la caméra sera constitué d'un plus grand nombre de polygones permettant une visualisation précise de ses détails. Il existe un grand nombre de travaux répondant à cette problématique de modèles multi-résolutions, nous en présentons ici quelques uns.

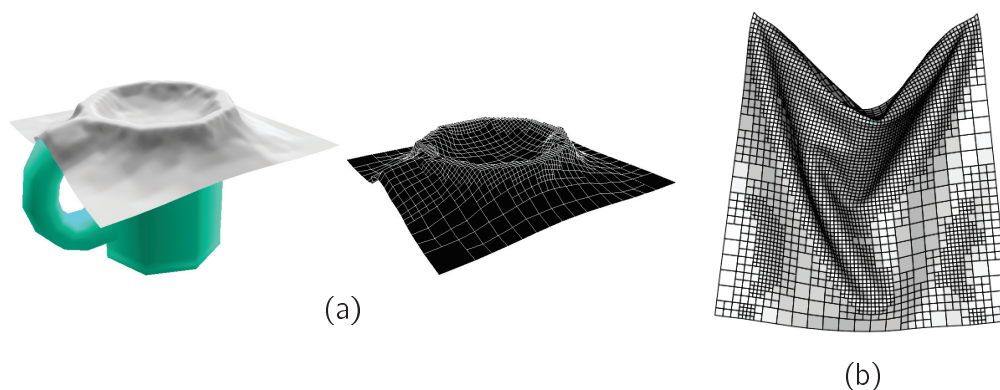
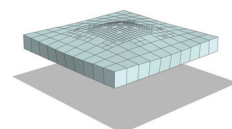


Figure 2.55 – Résultats des travaux de Hutchinson et al. [HPH96] et de Villard et al. [VB05] permettant le raffinement local d'un maillage SMR 2D. (a) Résultat des travaux de Hutchinson et al. d'un maillage SMR 2D représentant un tissu, se déposant sur une tasse en se raffinant en fonction d'un critère d'angle. (b) Résultat des travaux de Villard et al. [VB05] d'un maillage SMR 2D représentant un tissu.

Dans ce sens, Hutchinson et al. [HPH96] proposent une méthode permettant de raffiner localement le maillage 2D d'un SMR, pour améliorer le réalisme de l'animation pour un coût en calcul peu élevé (cf. la figure 2.55(a)). Le maillage consiste en une grille régulière où une

particule est reliée par des ressorts à ses voisins selon une 8-connexité. Ils précisent que la résolution du maillage dans le cas d'un SMR est primordiale. En effet, si le maillage est trop grossier, l'animation ne donne pas de résultats réalistes du fait d'une trop grande approximation du modèle. Par contre, si le maillage est trop fin, l'animation permet des résultats plus réalistes mais pour des temps de calculs trop importants. Ils proposent, en réponse, une méthode permettant de raffiner le maillage seulement dans les régions nécessitant un plus grand réalisme. Plus précisément, le maillage est raffiné aux endroits où l'angle entre deux ressorts, de même direction et joignant une même masse, présentent un angle supérieur à un seuil fixé. L'objectif étant de détecter une imprécision afin de l'éviter en ajoutant des masses et des ressorts dans la zone où se situe cette imprécision. Initialement, ils commencent avec un maillage consistant en une grille régulière de points, ayant un niveau de résolution grossier de valeur 0. Puis, lors de la détection d'une imprécision, ils génèrent de nouvelles masses et de nouveaux ressorts pour un niveau de résolution 1. Ils définissent alors des particules « actives » correspondant à toutes les particules des différents niveaux de subdivision, dont ils calculent les nouvelles positions, contrairement aux particules « inactives » qui ne rentrent pas dans le calcul de l'animation. Il existe également un dernier type de particules situées aux interfaces entre deux zones de niveau de résolution différents. Leur déplacement est interpolé linéairement en fonction des particules actives avoisinantes. Les nouvelles masses ont la même masse que toutes les autres masses du niveau de résolution 0. Il est difficile avec un SMR de simuler le même comportement physique quelque soit le niveau de subdivision du maillage. Pour compenser le fait qu'ils gardent la même masse pour toutes les nouvelles particules, ils doublent la raideur des ressorts à chaque niveau de résolution afin de garantir le même comportement entre les différents niveaux de résolution. Enfin, le pas de temps des niveaux de résolution fine est plus petit que celui des niveaux de résolution grossière, afin de garantir la stabilité.

Dans le même contexte, Villard *et al.* [VB05] proposent une méthode pour l'animation de tissus avec un modèle présentant différents types de ressorts (*cf.* la figure 2.55(b)). Chaque type de ressorts permet de modéliser un comportement différent du tissu selon qu'il soit étiré, cisailé ou encore torsionné ou vrillé. Ils utilisent pour cela un modèle physique SMR consistant en une méthode améliorée de celle de Provot [Pro95]. Ils présentent l'inconvénient d'un maillage grossier pour représenter les plis du tissu. Ils proposent alors une méthode adaptative débutant avec un maillage quadrangulaire régulier et grossier, qui est raffiné localement en fonction d'un critère géométrique reposant sur la courbure calculée en un nœud du maillage. Pour leur méthode de raffinement, ils définissent deux types de nœud, les nœuds « actifs » et les nœuds « virtuels ». Les nœuds « actifs » sont les nœuds pour lesquels le maillage est conforme, c'est-à-dire que si l'on ne considère pas la bordure, ils sont incidents à quatre éléments. Ils participent alors à la simulation du tissu. Les nœuds « virtuels » représentent quant à eux tous les autres nœuds. Leur présence est nécessaire pour définir la topologie du maillage. Ils sont en relation avec trois nœuds actifs sur lesquels leurs contributions physiques sont transmises. Ces nœuds virtuels correspondent à des **jonctions en T**. Lorsque le critère de raffinement est satisfait en un nœud du maillage, les quatre éléments incidents à ce nœud seront alors raffinés en quatre sous-éléments. En ce qui concerne les paramètres physiques, la masse est initialement distribuée de manière uniforme sur l'ensemble des nœuds qui sont alors tous actifs. Puis, lorsque le maillage est raffiné, la masse des nœuds concernés par le raffinement est mise à jour. Le calcul de la masse est fonction des aires des éléments incidents au nœud en cours de mise à jour. Plus précisément, pour un nœud ayant quatre éléments incidents, huit triangles sont construits avec les quatre barycentres des éléments incidents et les quatre nœuds les plus proches du nœud considéré. Les aires de ces huit triangles permettent de calculer la nouvelle masse. Enfin, lors du raffinement d'un élément



incident, la raideur d'un ressort incident scindé en deux parties voit la raideur de ses deux parties doublée, tout comme les nouveaux ressorts créés.

Comme indiqué précédemment, il est difficile avec un SMR de garantir le même comportement physique indépendamment du niveau de résolution. Nous avons que Hutchinson *et al.* doublent la raideur de leurs ressorts, tout en gardant une même masse pour leurs nouvelles particules. Villard *et al.* initialisent ou mettent à jour les masses de leurs particules en fonction de l'aire des éléments qui sont incidents à la particule considérée. De plus, ils ajustent les forces d'élasticité en fonction du niveau de résolution. Dans ce sens, nous citons deux autres travaux :

- Ganovelliet *al.* [GCMS00] proposent une méthode de raffinement adaptatif d'un SMR tétraédrique. Le raffinement est effectué en fonction d'un chemin de découpe réalisé par l'utilisateur. Ils s'appuient sur la formulation des coefficients de raideur de Van Gelder [Van98] dans le but de préserver la cohérence des propriétés mécaniques à tous les niveaux de résolution.
- Choi *et al.* [CHCK02] présentent une méthode adaptative utilisant la structure topologique de l'arête ailée et un SMR. Lors d'un raffinement local, ils emploient des ondelettes de surfaces comme méthode de subdivision pour passer d'un niveau de résolution à un autre. Enfin, ils appliquent des filtres, afin de déterminer les masses des nouveaux nœuds, sur les nœuds préexistants, afin de préserver la masse globale.

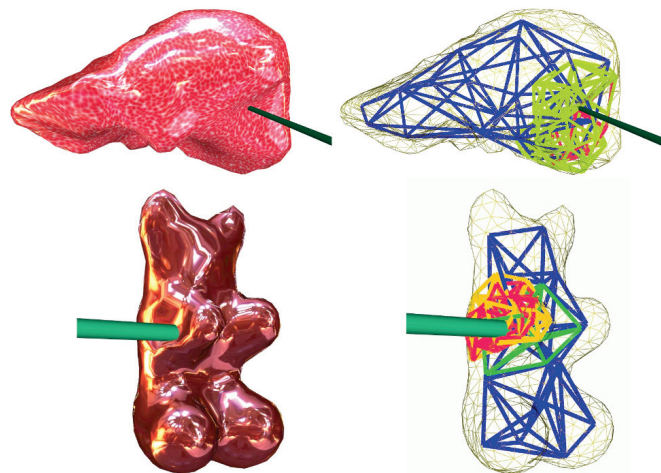


Figure 2.56 – Résultats des travaux de Debunne et al. [DDCB01]. Modèles tétraédriques raffinés localement.

Debunne *et al.* [DDCB01] présentent une méthode adaptative permettant l'animation temps-réel d'objets déformables **visco-élastiques** (cf. la figure 2.56). Un objet visco-élastique comporte une partie solide élastique et une partie fluide visqueux. Ils proposent une technique automatique permettant une adaptation en temps et en espace du modèle. Ce dernier consiste en un ensemble de tétraèdres plus ou moins raffinés non-emboîtés. Plus précisément, il consiste en un ensemble de maillages indépendants du même objet, par niveau de résolution. Chaque maillage est construit par un échantillonnage quasi-uniforme de la surface 3D de l'objet, suivi de la tétraédrisation de l'intérieur de la surface ainsi définie. Ils définissent alors la notion de nœud « parent » et « enfant ». Un nœud parent d'un niveau de résolution n a des nœuds enfants dans le niveau de résolution $n+1$ ($n+1$ étant plus fin que n). Les nœuds enfants appartiennent à la cellule Voronoï du nœud parent dans le niveau $n+1$. Quand on raffine un élément, les nœuds parents constituant l'élément en question, sont remplacés par

les nœuds enfants du niveau de résolution plus fin, et inversement pour le dé-raffinement, où les nœuds enfants sont substitués à un nœud parent. Avant le raffinement, les nœuds parents étaient « actifs », maintenant ce sont les nœuds enfants correspondants qui le sont. Des nœuds d'« interface » sont définis au niveau de la frontière entre les nœuds actifs appartenant à des maillages de différentes résolutions. Les nœuds d'interface sont interpolés linéairement en fonction du tétraèdre actif auquel ils appartiennent. Dans cet article, ils ont également étudié les vibrations générées par la cohabitation de plusieurs niveaux de résolution. En effet, si les régions de différentes résolutions commencent à vibrer à des fréquences et des amplitudes différentes, des interférences au niveau des interfaces de ces régions peuvent rapidement engendrer l'instabilité du système tout entier. La plupart des méthodes choisissent d'augmenter la viscosité pour pallier ces vibrations. Debonne *et al.* montrent que le SMR ne permet pas les mêmes amplitudes et fréquences pour des résolutions différentes, tout comme l'emploi du tenseur de déformation de Cauchy dans une méthode de type MEF linéaire. Ils optent pour le tenseur de déformation de Green-Lagrange avec une méthode de type MEF. Enfin, leur critère de raffinement s'appuie sur le calcul de la courbure de déplacement. En conclusion, le nombre de nœuds est plus important dans les régions nécessitant une plus grande précision du fait d'une courbure de déplacement des nœuds trop importante.

2.4.2 Découpe

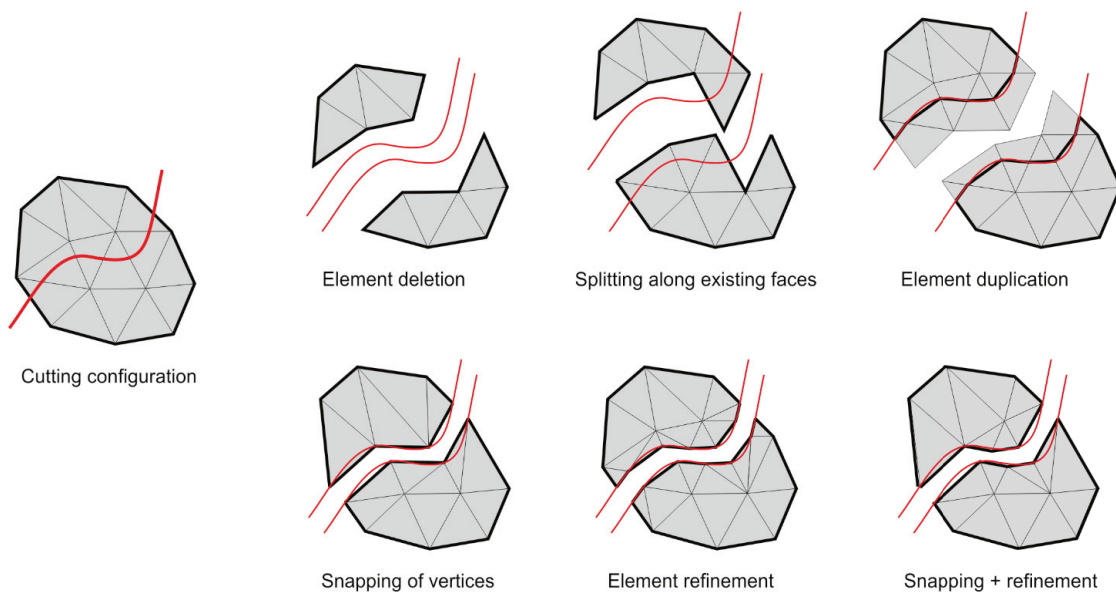
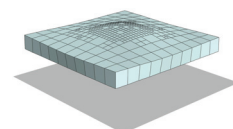


Figure 2.57 – Illustration des différentes techniques permettant la découpe d'un maillage triangulaire. Cette illustration est présentée dans l'état de l'art de Wu et al. [WWD14]. A gauche : Le maillage triangulaire initial et la représentation d'un chemin de découpe en rouge. De gauche à droite et de haut en bas : La suppression des éléments se trouvant sur le chemin de découpe. La séparation des arêtes se trouvant le long du chemin de découpe. La duplication des éléments se trouvant sur le chemin de découpe suivie de la séparation des arêtes se trouvant le long du chemin de découpe. La projection des points du maillage se trouvant au plus près du chemin de découpe, sur ce dernier, suivie de la séparation des arêtes se situant le long du chemin de découpe. Le raffinement des éléments se trouvant sur le chemin de découpe suivi de la séparation des arêtes. Une méthode alliant raffinement et recalage des points puis séparation des arêtes.

Nous faisons, dans cette partie, une brève présentation de la problématique de la découpe. Récemment, Wu *et al.* [WWD14] ont publié un état de l'art sur la découpe virtuelle



d'objets déformables (cf. la figure 2.57). Ils introduisent la problématique de la découpe virtuelle comme un domaine de recherche actif, particulièrement présent dans la réalisation de simulateurs chirurgicaux virtuels. Cet état de l'art présente les travaux ayant contribué à la problématique de la découpe virtuelle d'objets déformables. Les méthodes présentées peuvent être valables pour des maillages volumiques tétraédriques, hexaédriques ou encore polyhédriques. Elles proposent différentes techniques de découpe comme par exemple, la suppression des éléments se trouvant sur le chemin de découpe ou encore le raffinement des éléments se trouvant sur le chemin de découpe suivi de la séparation des arêtes (cf. la figure 2.57). Notons que la problématique de la multi-résolution est un domaine d'application de la découpe virtuelle. Enfin, ce rapport présente les différents modèles physiques utilisés pour la découpe virtuelle comme la MEF classique, polyhédrique, composite et étendue ainsi que les méthodes sans maillage.

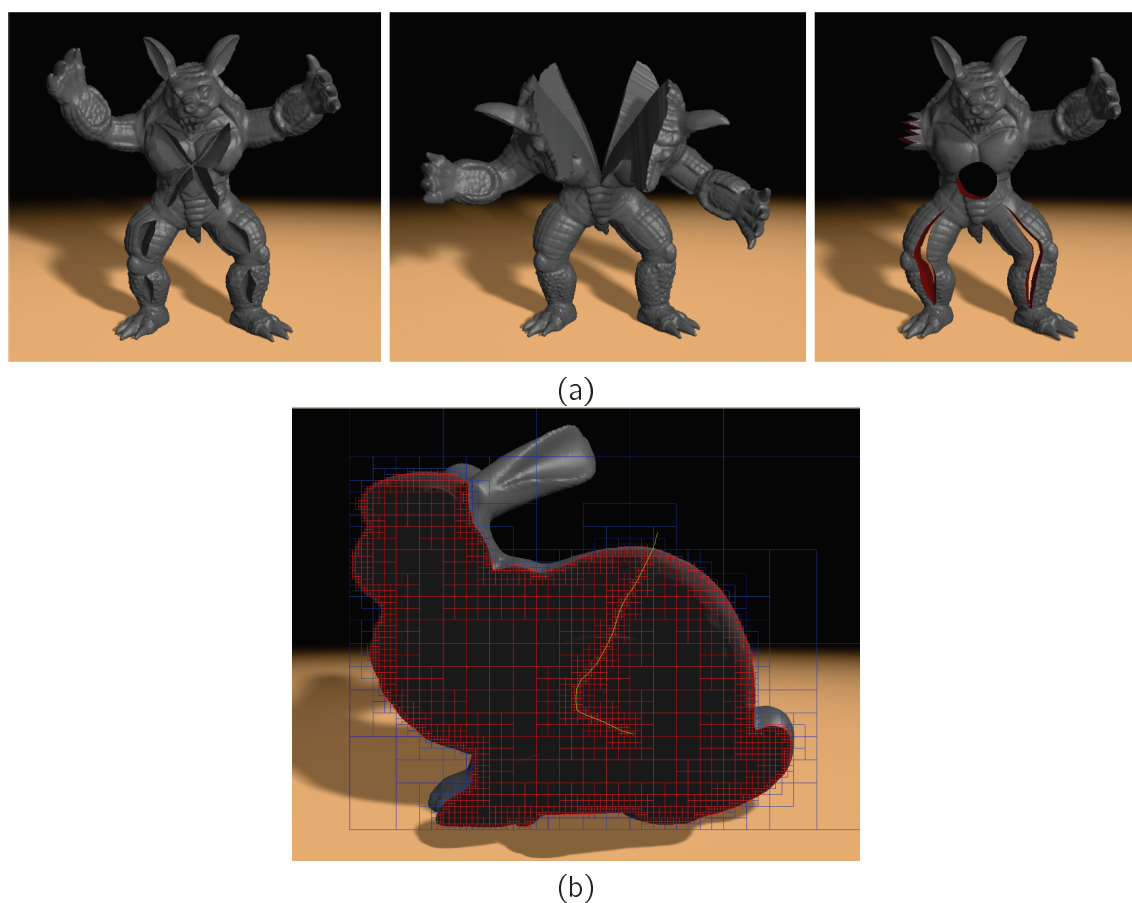


Figure 2.58 – Résultats et modèle issus des travaux de Dick et al. [DGW11] permettant la découpe virtuelle d'objets déformables. Un modèle représentant initialement l'Armadillo de Stanford. (a) Ce modèle suite à la simulation et l'application de différentes coupes, permises par une méthode adaptative appliquée à un modèle hexaédrique. (b) L'arbre octal dont on observe son raffinement le long de la surface de l'objet 3D représente un lapin, ainsi que le long du chemin de découpe.

Nous citons les travaux de Dick *et al.* [DGW11] pour l'emploi d'un maillage hexaédrique. En effet, beaucoup de méthodes sont valables sur des maillages tétraédriques. Or ces derniers présentent l'inconvénient de produire des éléments dégénérés lors, par exemple, de technique de découpe projetant les sommets du maillage sur le chemin de découpe ou lors de méthode de raffinement quand le tétraèdre est subdivisé en sous-tétraèdres. Nous précisons qu'un élément dégénéré provoque des instabilités numériques et doit à tout pris être évité.

Dick *et al.* [DGW11] présentent une méthode adaptative appliquée à un maillage MEF hexaédrique avec une formulation corotationnelle, permettant la simulation de découpe d'objets déformables élastiques linéaires (*cf.* la figure 2.58(a)). Ils emploient pour cela, un schéma d'intégration multi-résolution. La simulation débute avec une grille hexaédrique grossière, puis cette dernière est raffinée localement en fonction de l'interaction avec un objet de découpe. Ils emploient pour cela un arbre octal (*cf.* la figure 2.58(b)). Lorsque l'arbre octal est assez fin, la découpe est modélisée par la séparation des éléments le long de cette dernière.

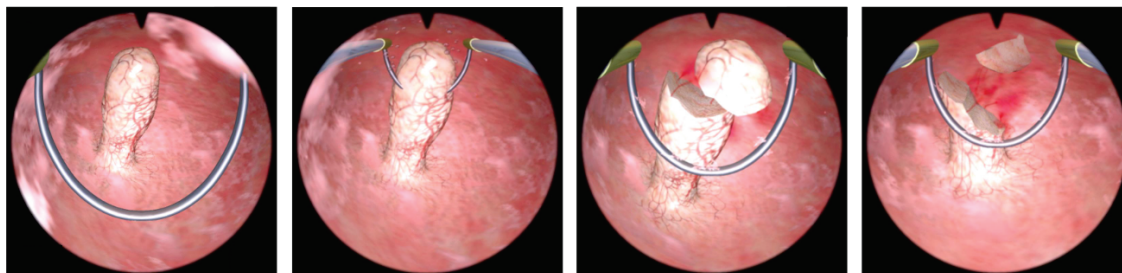


Figure 2.59 – Résultats des travaux de Steinemann *et al.* [SHGS06] permettant la découpe virtuelle d'objets déformables. Résultat obtenu représentant l'ablation d'un polype dans le cadre d'un simulateur chirurgical.

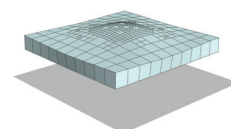
Enfin, nous citons Steinemann *et al.* [SHGS06] pour l'emploi d'un SMR. Ils proposent une méthode de découpe virtuelle appliquée à un maillage tétraédrique (*cf.* la figure 2.59). Pour cela, ils emploient une technique raffinant un tétraèdre en sous-tétraèdres selon des patterns de subdivision et projetant les particules sur le chemin de découpe, en évitant de produire des tétraèdres dégénérés.

2.5 Conclusion

Nos travaux consistent en l'élaboration d'un nouveau modèle unifié pour la simulation physique d'objets déformables tout en permettant des changements topologiques au cours de cette simulation. Ce nouveau modèle unifié est constitué d'un modèle topologique embarquant un modèle physique.

Le modèle topologique employé est celui des LCC qui sont issus des cartes combinatoires et permettent donc de représenter des objets orientables de dimension quelconque par une subdivision en cellules de n'importe quelle topologie. Ils permettent alors de représenter des maillages 2D mixant aussi bien des triangles que des quadrilatères, et des maillages 3D composés aussi bien de tétraèdres que d'hexaèdres ou encore de prismes. En plus de sa généralité, ce modèle topologique permet, par sa définition mathématique, de garantir toutes les relations d'incidence et d'adjacence entre les cellules composant la représentation de l'objet modélisé. Cette définition mathématique rigoureuse permet également de garantir la validité topologique du modèle suite à des changements topologiques. Nous précisons que les opérations de parcours et de modifications définies par le modèle sont également génériques au regard de la dimension des objets modélisés ou encore de la topologie des cellules. Enfin, ce modèle topologique est doté d'un mécanisme d'attributs permettant l'association d'informations au niveau de ses cellules.

Nous avons étudié l'intérêt de l'emploi d'un modèle topologique nD pour la simulation d'objets déformables soumis à des changements topologiques. Bien que des modèles topologiques exclusivement 2D soient communément employés dans le cadre d'une simulation d'objets surfaciques, très peu de travaux ont étendu cette association aux objets volumiques.



Nous nous sommes alors tout naturellement intéressés à l'emploi d'un modèle topologique plus complexe, pour simuler le comportement d'objets déformables volumiques. De plus, les objets déformables que nous manipulons sont orientables. Nous employons alors le modèle topologique des LCC, dont une implémentation est proposée dans la librairie CGAL (Computational Geometry Algorithms Library) [CGA]. L'utilisation de cette librairie implique l'utilisation de la version de la définition des cartes combinatoires valable pour des objets ouverts. Nous précisons que cette version permet un modèle réduit en terme de brins mais complexifie les opérations par l'introduction du brin libre. Enfin, nous rappelons que les travaux présentés dans la littérature distinguent le modèle topologique et le modèle physique. Nos travaux présentent quant à eux un modèle unifié. L'unification du modèle topologique et physique évite les redondances d'informations et facilite la mise à jour de ces dernières.

Le modèle physique que nous avons embarqué dans le modèle topologique des LCC consiste en un système masses-ressorts (SMR). Nous avons fait le choix de ce modèle physique pour la simplicité de ses équations permettant une implémentation intuitive et un faible coût en terme de calcul. Ce dernier point permet son emploi dans des applications interactives. Il permet, en effet, des interactions temps-réel avec la simulation physique. De plus, l'application des forces au niveau des particules facilite les changements topologiques ainsi que la mise à jour nécessaire des paramètres physiques. Le paramétrage de notre SMR utilise les travaux de Baudet *et al.* [Bau06, BBJ⁺09b] proposant une méthode analytique. Cette dernière permet une formulation des coefficients de raideur en fonction des paramètres rhéologiques des matériaux comme le module de Young ou le coefficient de Poisson. Cependant, cette formulation n'est définie que pour des quadrilatères en 2D et des hexaèdres en 3D. Du fait de cette formulation des raideurs, la méthode présentée dans ce manuscrit est alors limitée à l'emploi de ces deux formes. Enfin, nous précisons que notre méthode est générique quant au type du modèle physique employé. Ainsi, un modèle physique s'appuyant sur la méthode des masses-tenseurs est en cours d'intégration [Fau14].

Notre modèle unifié (LCC+SMR) permet ainsi une grande souplesse dans la gestion des changements topologiques tout en garantissant la validité topologique du modèle. Les changements topologiques que nous proposons dans ces travaux sont réalisés de manière répétée au cours de la simulation. Nous avons implémenté différents changements topologiques comme celui de la découpe ou du raffinement.

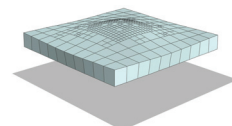
Chapitre 3

« Le Bagage s'arrêta à quelques pas du mage et, au bout d'un moment, rentra ses jambes. Rincevent ne lui voyait pas d'yeux mais il était sûr que le coffre le regardait. Et qu'il attendait. « Va coucher », dit le mage d'une voix faible. Le Bagage refusa de bouger, mais son couvercle s'ouvrit en grinçant et relâcha le cadavre du voleur. »
(Les Annales du Disque-Monde, La huitième couleur - Terry Pratchett)

Modèle unifié alliant topologie et physique

Sommaire

3.1	Description du modèle unifié	86
3.2	Initialisation du modèle unifié	88
3.2.1	Masses	91
3.2.2	Ressorts	93
3.3	Simulation physique	96
3.4	Changement topologique de la découpe	97
3.4.1	La partie topologique du processus de découpe	97
3.4.2	La partie physique du processus de découpe	100
3.5	Conclusion	103



Nous présentons dans cette partie nos contributions, qui consistent en la création d'un **nouveau modèle unifié** pour la simulation physique d'objets déformables tout en permettant des changements topologiques au cours de cette simulation. Ce nouveau modèle unifié est constitué du modèle topologique des LCC embarquant le modèle physique du SMR.

Dans ce chapitre, nous détaillerons tout d'abord, la construction de notre modèle unifié et l'initialisation de ses paramètres physiques. Dans un second temps, nous présenterons la simulation physique d'objets déformables permise par notre modèle. Enfin, nous décrirons le changement topologique de la découpe, réalisé en 2D, selon les arêtes d'une face, ou en 3D, selon les faces d'un volume. Cette opération de découpe sera décrite, tout d'abord avec un regard topologique, puis un regard physique avec la mise à jour des paramètres du SMR. Cette opération de découpe amène à l'opération de perçage et de suppression d'**éléments**. Nous noterons par la suite : élément, une face en 2D et un volume en 3D.

3.1 Description du modèle unifié

Nous présentons dans cette partie, la construction de notre modèle unifié : LCC+SMR. Ce modèle consiste en le modèle topologique des LCC embarquant le modèle physique du SMR.

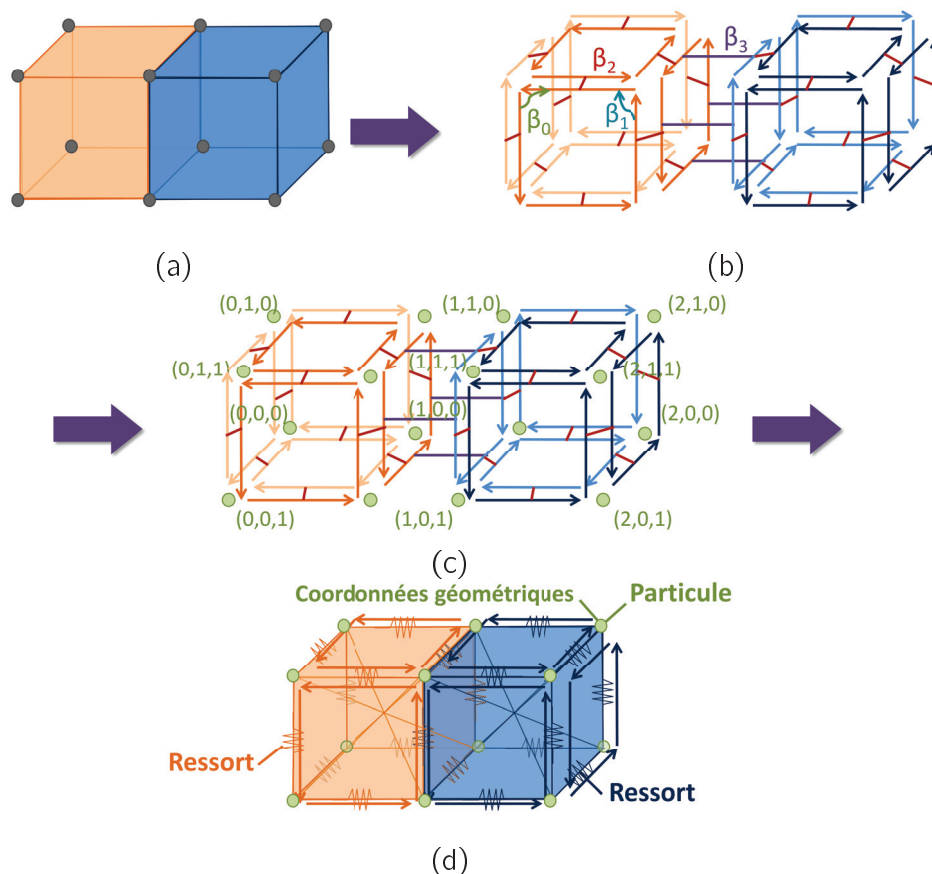


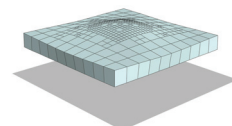
Figure 3.1 – Le processus de construction de notre LCC+SMR. (a) Un objet 3D représenté par un maillage composé de deux hexaèdres adjacents. (b) La carte combinatoire 3D correspondante. (c) La LCC avec l'association des coordonnées géométriques 3D aux sommets. (d) Notre modèle unifié qui associe à la LCC 3D présentée en (c) les informations nécessaires à la simulation physique. Pour des raisons de lisibilité, les relations β_2 et β_3 ne sont pas représentées sur le schéma (d).

- La figure 3.1(a) montre que la construction de notre LCC+SMR prend en entrée un maillage constitué de quadrilatères en 2D ou d'hexaèdres en 3D. Dans cet exemple, les données géométriques et topologiques en entrée correspondent à deux hexaèdres adjacents. Nous rappelons que la propriété générique des cartes combinatoires permet des cellules de topologie quelconque. La restriction aux cellules de type quadrilatère ou hexaèdre n'étant alors que d'ordre physique, notre méthode garde cette généricité pour la construction d'une LCC ayant des cellules de formes quelconques. De cette manière, nous pouvons facilement étendre cette généricité grâce à l'emploi d'un modèle physique permettant la simulation de cellules de topologie quelconque.
- La figure 3.1(b) représente la carte combinatoire 3D issue des données géométriques et topologiques, présentées dans la figure 3.1(a). Le maillage 3D se retrouve ainsi décomposé en deux 3-cellules, onze 2-cellules, vingt 1-cellules et douze 0-cellules.
- La carte combinatoire 3D est ensuite plongée dans \mathbb{R}^3 . Plus précisément, les sommets du modèle topologique sont attribués de coordonnées géométriques 3D, grâce au mécanisme d'attribut des cartes combinatoire. Cette étape permet d'associer une géométrie à une carte combinatoire. Ce plongement définit alors une LCC, représentée dans la figure 3.1(c).
- La dernière étape consiste à associer des attributs stockant les informations nécessaires à la simulation physique, aux différentes cellules de la LCC. La figure 3.1(d) correspond au modèle LCC+SMR ainsi construit.

Nous détaillons plus précisément cette dernière étape. Les figures 3.2(a) et (b) correspondent respectivement à un exemple 2D et 3D de notre LCC+SMR. Ces modèles sont obtenus suite aux différentes étapes présentées dans le processus (*cf.* la figure 3.1). Pour le cas 2D, le modèle correspond à un quadrilatère qui a été décomposé en une 2-cellule, quatre 1-cellules et quatre 0-cellules. Pour le cas 3D, le modèle est un hexaèdre qui a été décomposé en une 3-cellule, six 2-cellules, douze 1-cellules et huit 0-cellules. Le SMR que nous avons implémenté dans ces travaux, est valable pour des maillages constitués de quadrilatères en 2D et d'hexaèdres en 3D. Ce SMR distingue deux types de ressorts, ceux disposés sur les arêtes de chaque élément du maillage (quadrilatère en 2D et hexaèdre en 3D), et ceux situés à l'intérieur de ce dernier, que nous nommerons par la suite **ressorts diagonaux**. Les figures 3.2(a) et (b) illustrent la disposition des ressorts pour respectivement un élément 2D et 3D. En 2D, nous observons un ressort sur chacune des quatre arêtes du quadrilatère ainsi que deux ressorts diagonaux internes à l'élément. En 3D, nous notons douze ressorts disposés sur chacune des arêtes de l'hexaèdre ainsi que les arêtes de l'élément et de ressorts diagonaux internes à l'élément. Cette disposition est issue des travaux de Baudet *et al.* [Bau06, BBJ⁺09b].

Lors de la dernière étape du processus (*cf.* la figure 3.1), les informations relatives au SMR sont associées au LCC grâce au mécanisme des attributs. Les structures de données **Particule** et **Ressort** sont ainsi respectivement associées aux 0-cellules et aux 1-cellules de la LCC. L'association des paramètres du SMR avec la LCC se fait de la manière suivante :

- Chaque 0-cellule embarque une particule du SMR, grâce à son association avec l'attribut **Particule**. La structure de données **Particule** stocke la **masse**, l'**accélération** et la **vitesse** de la particule en question ainsi que la **somme de l'ensemble des forces** qui lui sont appliquées. Elle stocke également des informations relatives aux **contraintes** pouvant s'exercer sur la particule, en indiquant si elle est fixée selon un des axes du repère. Ces contraintes permettent notamment la réalisation de tests d'étirement, de cisaillement ou encore de flexion. Enfin, elle référence l'ensemble des **ressorts diagonaux**



ayant pour extrémité cette particule. Cette information est nécessaire lors de l'accumulation des forces internes dues aux ressorts diagonaux. Notons que la position de la particule est déjà associée par défaut à la 0-cellule en question (*cf.* la figure 3.1(c)).

- Chaque 1-cellule embarque un ressort du SMR, grâce à son association avec l'attribut `Ressort`. La structure de données `Ressorts` stocke le **coefficient de raideur**, le **coefficient d'amortissement** ainsi que la **longueur initiale** du ressort en question. La longueur initiale, notée L_0 par la suite, correspond à la distance euclidienne entre les particules extrémités du ressort. Nous précisons que L_0 est une constante. Enfin, la structure de données `Ressort` stocke également les **particules extrémités** du ressort. Notons que cette dernière information ne nécessite pas d'être stockée lorsque le ressort est associé à un élément topologique comme dans le cas présent. En effet, les extrémités du ressort peuvent être directement récupérées en accédant depuis les 0-cellules extrémités de la 1-cellule en question, aux particules correspondantes. Par contre, ce n'est plus le cas pour des ressorts de type « flexion » comme présenté dans les travaux de Louchet *et al.* [LPC95], ou encore des ressorts diagonaux internes à un élément qu'ils nomment ressorts de type « cisaillement ».
- Pour finir, deux (*resp.* quatre) ressorts diagonaux sont définis à l'intérieur d'un élément de type quadrilatère (*resp.* hexaèdre). Les ressorts diagonaux ne sont pas portés par un élément topologique comme peuvent l'être les particules aux 0-cellules ou les ressorts aux 1-cellules. Ces derniers sont stockés de manière indépendante dans un conteneur. Ils sont référencés par les particules situées à leurs extrémités. Enfin, un ressort diagonal stocke un brin incident à la 2-cellule ou 3-cellule auquel il appartient, et sa longueur initiale nommée également L_0 , correspondant au calcul de la racine carrée de la somme des carrés des dimensions de l'élément incident. Nous précisons une nouvelle fois que L_0 est une constante.

Nous terminons cette partie en revenant sur l'aspect permettant de représenter des objets ouverts, du modèle topologique employé (*cf.* la partie 2.3.3). Nous rappelons que cette propriété inclut le concept de brin libre. Les brins situés sur la bordure du modèle sont alors 2-libres dans le cas d'un modèle 2D et 3-libres dans le cas d'un modèle 3D. Ceci implique de modifier les opérations de parcours des brins incidents à un sommet en 2D, et un sommet et une arête en 3D, par rapport à celles définies dans la version des cartes combinatoires valables pour des objets fermés. En effet, comme présenté dans la partie 2.3.3, dans le cas 2D, pour parcourir, l'ensemble des brins incidents à un sommet donné, nous appliquons successivement l'opérateur β_{21} depuis un brin partant de ce sommet jusqu'à retomber sur ce premier brin. Par contre, si nous tombons sur un brin libre, alors le parcours inverse : β_{02} doit être réalisé afin de garantir le parcours de l'ensemble des brins incidents au sommet. Afin de simplifier la présentation de nos travaux, nous nommons **parcours(invol-permu)**, la procédure qui à partir d'une succession de permutations et/ou d'involutions : **invol-permu**, gère l'aspect de la représentation d'objets ouverts, c'est-à-dire la rencontre de brins libres pendant un parcours. Par exemple, dans l'exemple précédent, la procédure `parcours(β_{21})` permet de parcourir l'ensemble des brins incidents à un sommet en partant d'un brin incident à ce dernier, puis si un brin libre est rencontré, la procédure se chargera de faire le parcours inverse en appliquant β_{02} (*cf.* la partie 2.3.3). Nous précisons que cette procédure gère les redondances, c'est-à-dire qu'elle ignore les brins déjà parcourus.

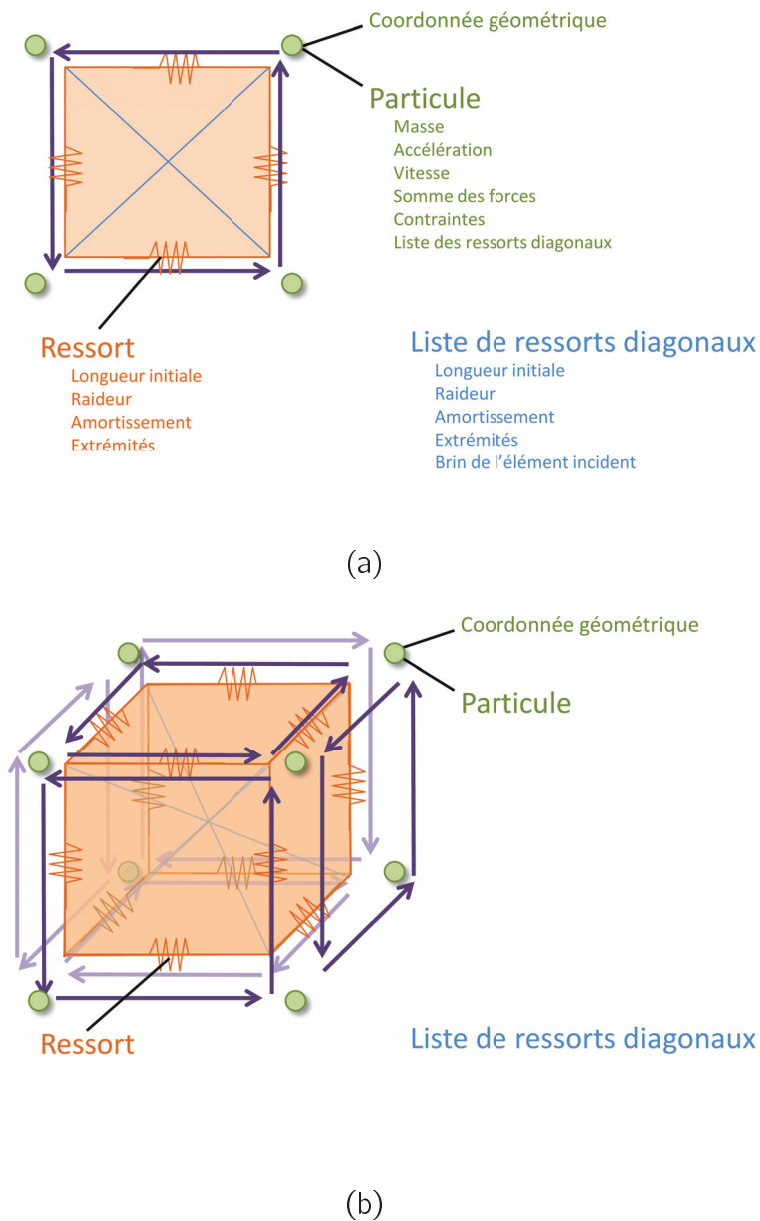
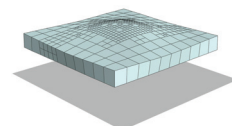


Figure 3.2 – Un quadrilatère et un hexaèdre représenté par notre modèle LCC+SMR. (a) Un quadrilatère représenté par notre modèle LCC+SMR. Il est composé de quatre particules et de quatre ressorts du SMR, associés respectivement aux 0-cellules et aux 1-cellules de la LCC. Par défaut, une coordonnée géométrique est également associée à la 0-cellule. Une structure de données Particule rassemblant les propriétés physiques d'une particule du SMR, est associée à chaque 0-cellule de la LCC. Une structure de données Ressort rassemblant les propriétés physiques d'un ressort du SMR, est associée à chaque 1-cellule de la LCC. Enfin, un conteneur stocke les deux ressorts diagonaux du système ainsi que leurs propriétés physiques. (b) Un hexaèdre représenté par notre modèle LCC+SMR. Il est composé de huit particules et de douze ressorts du SMR associés respectivement aux 0-cellules et aux 1-cellules de la LCC. De la même manière que pour (a) des structures de données stockent les propriétés physiques des éléments du SMR. Pour des raisons de lisibilité, les relations β_2 et β_3 ne sont pas représentées sur le schéma (b). Enfin, nous avons représenté pour le cas 2D et 3D, la liste de l'ensemble des ressorts diagonaux du système.



3.2 Initialisation du modèle unifié

Pour effectuer la simulation d'un objet déformable, notre méthode débute par un pré-traitement ayant pour objectif la construction de notre modèle unifié, puis cette étape est suivie de la simulation physique de ce modèle. Nous venons de décrire notre modèle unifié et les différentes étapes permettant son obtention. Dans cette partie, nous revenons plus en détails sur ce pré-traitement en présentant la manière dont sont stockées les informations géométriques et topologiques en entrée du processus, puis comment la carte combinatoire est construite à partir de ces dernières. Enfin, nous détaillerons le calcul des informations physiques relatives au SMR.

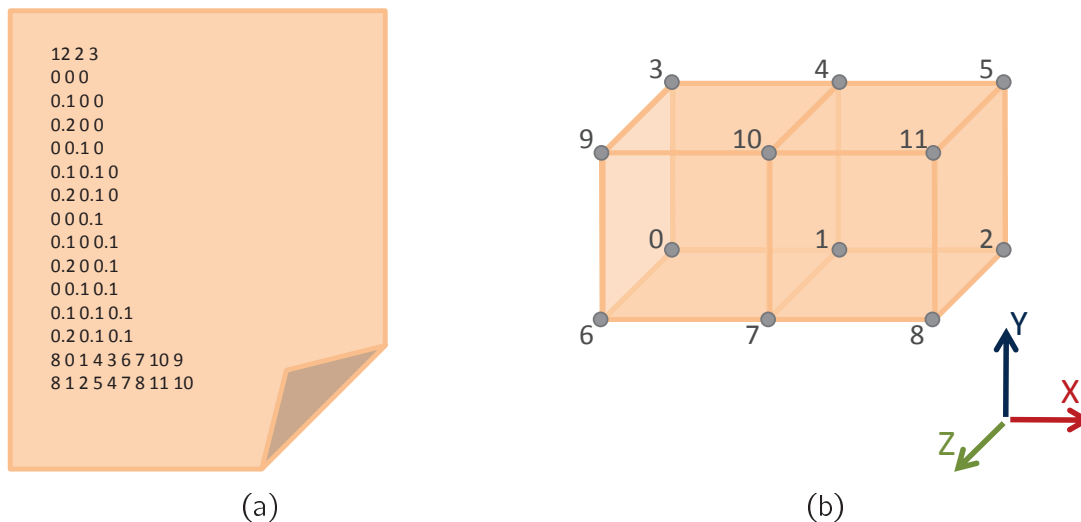


Figure 3.3 – (a) Exemple d'un fichier stockant les informations géométriques et topologiques nécessaires à l'élaboration de notre modèle unifié. Ces dernières correspondent à deux hexaèdres adjacents de dimension $0.1 \times 0.1 \times 0.1$. (b) Maillage correspondant au fichier présenté dans (a) présentant l'indexation des sommets.

Les informations géométriques et topologiques sont issues d'un fichier correspondant à des maillages 2D ou 3D. La première ligne de ce fichier comporte le nombre de sommets suivi du nombre d'éléments constituant le maillage, et enfin la dimension de ce dernier. Il liste ensuite de manière classique les coordonnées géométriques 3D des sommets du maillage. Puis, il décrit chaque élément du maillage par une ligne composée du nombre de ses sommets suivi de l'ensemble des index des sommets le constituant. Nous pouvons définir de cette manière des maillages 2D triangulaires, quadrangulaires mais également constitués de faces de forme quelconque. Des maillages mixtes peuvent également être construits en rassemblant les différents types de faces précédemment mentionnés. En 3D, des maillages tétraédriques peuvent être définis tout comme des maillages hexaédriques ou encore des maillages mixant des tétraèdres et des hexaèdres.

La figure 3.3(a) présente le fichier correspondant au maillage en entrée du processus (cf. la figure 3.1(a)). La première ligne du fichier indique que le maillage est constitué de douze sommets, de deux volumes et qu'il est volumique. Puis, nous observons douze lignes dont chacune correspond à une coordonnée géométrique d'un sommet. Notons que ces coordonnées géométriques sont rangées selon l'axe X, Y puis Z. Enfin, nous avons deux lignes pour chacun des deux volumes du maillage. Le premier chiffre indique le nombre de sommets constituant l'élément considéré. Dans notre exemple, nous avons deux hexaèdres, les deux lignes commencent donc par un 8. Pour chaque ligne, les chiffres qui font suite au 8 corres-

pondent aux index des sommets composant l'hexaèdre en question. La figure 3.3(b) montre comment les index sont répartis au niveau des sommets des deux hexaèdres. Nous précisons que pour des maillages réguliers composés de quadrilatères ou d'hexaèdres, ces fichiers sont générés en spécifiant le nombre d'éléments selon X , Y et Z ainsi que les dimensions de ces derniers.

- Pour construire une LCC à partir d'un maillage 2D, nous employons la librairie CGAL. Cette dernière propose les outils nécessaires à la construction d'une LCC surfacique à partir du nombre de sommets et de faces du maillage. Puis, toutes les coordonnées géométriques des sommets sont récupérées. Une face est ensuite créée pour chaque ensemble d'index. Enfin, le tout forme une surface, en liant par β_2 les faces partageant une arête.
- Pour construire une LCC à partir d'un maillage 3D, nous employons également la librairie CGAL. Elle permet la création de tétraèdres et d'hexaèdres à partir des coordonnées géométriques des sommets les constituant. Une fois l'ensemble des éléments générés, les éléments partageant une face commune sont reliés par β_3 .

Nous expliquons maintenant comment calculer les informations physiques relatives au SMR. Nous rappelons que ces informations sont associées à la LCC. Nous détaillerons tout d'abord le calcul des masses des particules composant le SMR, puis nous poursuivrons avec le calcul des coefficients de raideur des ressorts du SMR.

3.2.1 Masses

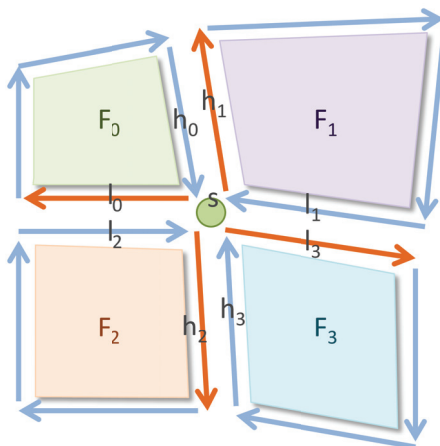
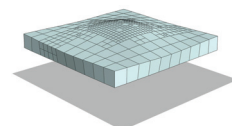


Figure 3.4 – Calcul de la masse associée à un sommet, pour le cas 2D. Un modèle 2D constitué de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . Les faces F_i ont pour dimensions l_i et h_i avec $i \in \{0, 1, 2, 3\}$. Les brins orange correspondent aux brins incidents au sommet s .

Cas 2D

La figure 3.4 présente un modèle 2D constitué de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . Dans cette figure, les brins orange sont les brins incidents au



sommet s , appartenant chacun à un de ses éléments incidents : F_0, F_1, F_2 et F_3 . Pour récupérer l'ensemble de ces brins, nous appliquons $\text{parcours}(\beta_{21})$ sur un des brins partant de s . La masse de s est calculée par accumulation des contributions de chacun de ses éléments incidents. La formulation de la masse de s pour un élément incident rectangulaire est :

$$m_s = \frac{1}{4} \sum_{e \in F_s} \rho_e \mathcal{A}_e$$

où m_s est la masse de la particule s , F_s est l'ensemble des faces incidentes à s , ρ_e est la densité surfacique de l'élément e et \mathcal{A}_e est l'aire de l'élément e (soit la hauteur h_e × la largeur l_e , pour une face carrée ou rectangulaire). Pour obtenir les dimensions h_e et l_e de l'élément e , nous récupérons la longueur stockée dans L_0 au niveau du ressort associé à un brin incident à l'élément e et le L_0 du ressort associée au β_1 de ce brin incident. Nous faisons ici l'hypothèse d'un élément rectangulaire mais nous pouvons également calculer l'aire de la face en fonction des aires des deux triangles inscrits dans le quadrilatère.

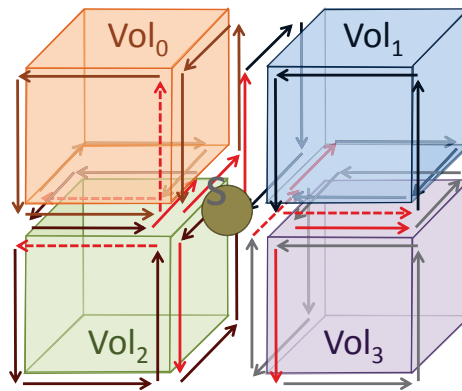


Figure 3.5 – Calcul de la masse associée à un sommet, pour le cas 3D. Un modèle 3D constitué de quatre hexaèdres : Vol_0, Vol_1, Vol_2 et Vol_3 , cousus ensemble par β_3 . Les brins en rouge correspondent aux brins incidents à s . Les brins rouges en pointillés représentent un ensemble de brins répondant à la contrainte d'un brin incident à s par volume incident à s . Ce schéma a pour objectif d'illustrer la procédure du calcul de la masse de s pour le cas 3D.

Cas 3D

La figure 3.5 présente un modèle 3D constitué de quatre hexaèdres : Vol_0, Vol_1, Vol_2 et Vol_3 , cousus ensemble par β_3 . Dans cette figure, les brins incidents à la particule s sont représentés en rouge. Dans le cadre d'un hexaèdre, un sommet possède trois brins incidents par volume incident. Pour récupérer les trois brins partant de s et incidents au volume Vol , nous commençons par récupérer un premier brin b incident à s et Vol . Puis, nous appliquons successivement $b = \beta_{21}(b)$, jusqu'à retomber sur le brin incident de départ. Notons que ce parcours est similaire à celui présenté pour le cas 2D, sauf que nous ne pouvons pas ici tomber sur un brin libre. En effet, dans le cadre d'un modèle 3D, nous ne permettons que des brins 3-libres.

La masse d'une particule s est obtenue par l'accumulation des contributions de chacun de ses volumes incidents. Pour calculer cette masse, il nous faut alors récupérer un brin par volume incident à s . Nous expliquons maintenant comment procéder en considérant le parcours précédent : β_{21} . Dans la figure 3.5, ils sont au nombre de quatre car s est incident

à quatre volumes et sont représentés par des pointillés. Nous précisons que ces derniers correspondent à un des ensembles de brins répondant à l'assertion : « un brin incident à s par volume incident à s ».

Une solution consiste alors à récupérer un brin b partant de s . Ce brin représente la contribution du volume qui lui est incident. Puis, nous appliquons β_{32} sur les trois brins obtenus par β_{21} . Nous marquons comme traité ces trois brins. Nous recommençons la procédure sur un brin qui n'est pas marqué comme traité. Ce dernier a été obtenu à partir de β_{32} . Il représentera alors la contribution de son volume incident. Nous procédons ainsi jusqu'à avoir obtenu un brin partant de s par volume incident.

La formulation de la masse de s pour un élément incident hexaédrique est :

$$m_s = \frac{1}{8} \sum_{e \in Vol_s} \rho_e \mathcal{V}_e$$

où m_s est la masse de la particule s , Vol_s est l'ensemble des volumes incidents à s , \mathcal{V}_e et ρ_e sont respectivement le volume et la densité volumique de l'élément e . Pour calculer le volume d'un pavé Vol , il nous faut récupérer ses dimensions. Pour cela, nous prenons la longueur stockée dans L_0 au niveau du ressort associé à un brin b_0 incident à Vol ainsi que le L_0 du ressort associé au brin $b_1 = \beta_{21}(b_0)$ et enfin le L_0 du ressort associé au brin $b_2 = \beta_{21}(b_1)$. Nous faisons ici l'hypothèse d'un élément de type pavé mais nous pouvons également calculer le volume en décomposant en tétraèdres l'hexaèdre dont nous souhaitons calculer le volume. Il nous suffirait alors de sommer les volumes des différents tétraèdres sous-jacents.

3.2.2 Ressorts

Cas 2D

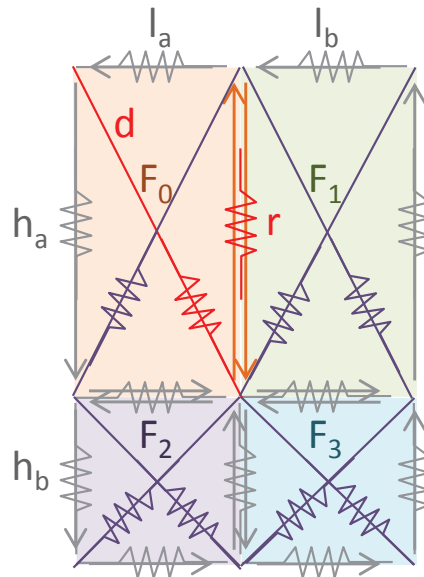
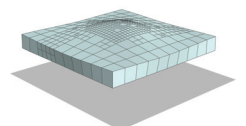


Figure 3.6 – Calcul de la raideur d'un ressort, pour le cas 2D. Un modèle 2D constitué de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . Les dimensions de F_0 , F_1 , F_2 et F_3 sont respectivement l_a, h_a , l_b, h_a , l_a, h_b et l_b, h_b . Les brins orange correspondent aux brins incidents à l'arête associée au ressort r .



- Pour calculer la raideur d'un ressort porté par une arête du maillage, nous parcourons l'ensemble de ses brins incidents. Leur nombre est compris entre un et deux.

Dans la figure 3.6, le ressort r est partagé par les 2-cellules : F_0 et F_1 et est porté par les deux brins orange. Nous rappelons que les coefficients de raideur des ressorts sont obtenus par une méthode analytique, en fonction des paramètres rhéologiques des matériaux que sont le module de Young et le coefficient de Poisson [BBJ⁺09a]. Le coefficient de raideur d'un ressort de longueur i par rapport à une 2-cellule F donnée, est formulé de la manière suivante :

$$k_F(i, j) = \frac{E(j^2(3\nu + 2) - i^2)}{4ij(1 + \nu)}$$

où E et ν sont respectivement le module de Young et le coefficient de Poisson de l'objet (ou de l'élément F pour un objet hétérogène). Ainsi, les deux raideurs d'un élément 2D F de dimension l_X et l_Y sont données par $k_F(l_X, l_Y)$ et $k_F(l_Y, l_X)$.

Dans la figure 3.6, la formulation du coefficient de raideur du ressort r est la suivante :

$$k_r = k_{F_0}(L_0(r), L_0(s_0)) + k_{F_1}(L_0(r), L_0(s_1))$$

$L_0(r)$ correspond à la longueur initiale stockée au niveau du ressort r . Elle a été initialisée à h_0 . s_0 correspond au ressort associé au β_1 du brin incident à r et à F_0 . $L_0(s_0)$ a été initialisé à l_a . s_1 correspond, quant à lui, au ressort associé au β_1 du brin incident à r et à F_1 . $L_0(s_1)$ a été initialisé à l_b .

$$k_r = \frac{E(l_a^2(3\nu + 2) - h_a^2)}{4l_a h_a(1 + \nu)} + \frac{E(l_b^2(3\nu + 2) - h_a^2)}{4l_b h_a(1 + \nu)}$$

- Le coefficient de raideur du ressort diagonal d (représenté en vert dans la figure 3.6) est interne à l'élément 2D F_0 de dimension l_a et h_a , est formulé de la manière suivante :

$$k_d = \frac{E(l_a^2 + h_a^2)}{4l_a h_a(1 + \nu)}$$

Notons que la formulation de Baudet *et al.*, implique que deux ressorts diagonaux internes à un élément ont le même coefficient de raideur.

Cas 3D

Un volume Vol de type pavé est caractérisé par trois longueurs initiales. Ses longueurs correspondent à celles de ses arêtes pour chaque direction dans le repère 3D. Ces dernières seront notées $\{X_{Vol}, Y_{Vol}, Z_{Vol}\}$ (cf. X_{Vol_3} , Y_{Vol_3} et Z_{Vol_3} dans la figure 3.7(b)). Nous les retrouvons dans les L_0 stockées au niveau des ressorts associés aux trois brins obtenus lors de l'application de β_{21} sur un brin incident à Vol . La détermination des longueurs de trois arêtes orthogonales entre elles, par rapport à un volume donné, est nécessaire au calcul du coefficient de raideur des ressorts incidents à ce volume.

- Pour calculer la raideur du ressort r , nous devons récupérer un brin incident à r par volume incident à r . Pour cela, nous parcourons l'ensemble des brins qui lui sont incidents (brins rouges dans la figure 3.7). Puis, nous appliquons $parcours(\beta_{23})$ sur un brin incident à r . Lors de ce parcours, nous gardons un brin sur deux. Chaque brin sélectionné permettra de déterminer la contribution de chaque volume incident à r , au calcul du coefficient de raideur de ce dernier. Le coefficient de raideur d'un ressort,

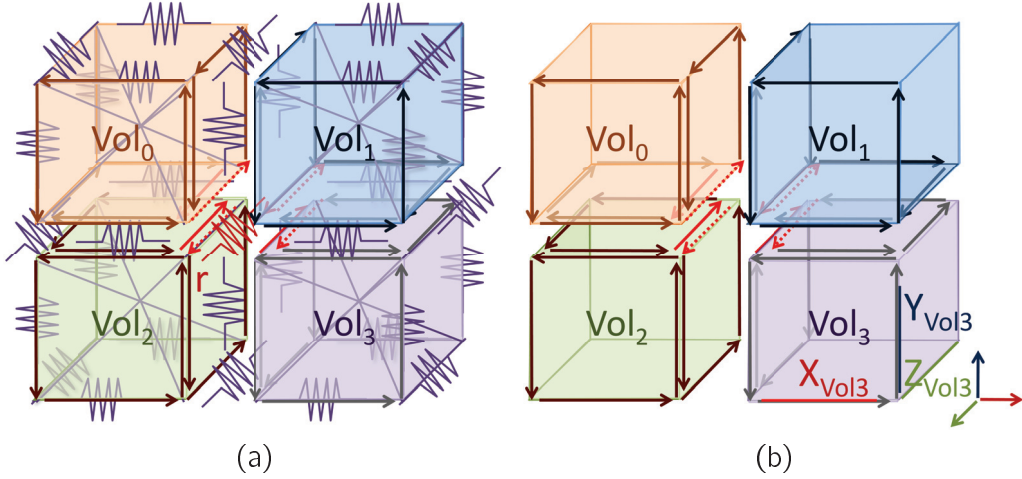


Figure 3.7 – Calcul de la raideur d'un ressort, dans le cas 3D. Un modèle 3D constitué de quatre hexaèdres : Vol_0 , Vol_1 , Vol_2 et Vol_3 , cousus ensemble par β_3 . Les brins en rouge correspondent aux brins incidents à l'arête associée au ressort r . Les brins rouges en pointillés représentent un ensemble de brins répondant à la contrainte : « Un brin incident à r par volume incident à r ». (a) Le modèle où sont représentés l'ensemble des ressorts (en violet) du système. Nous observons également l'ensemble des brins décrivant les différentes 2-cellules incidentes au ressort r ainsi que ceux des 2-cellules situées au premier plan. (b) Le modèle de (a) mais sans les ressorts, pour plus de lisibilité.

porté par une arête du modèle de longueur i , par rapport à un volume Vol est formulé de la manière suivante :

$$k_{Vol}(i, j, k) = \frac{E (6 j^2 k^2 (1 + \nu) + (\nu(j^2 + k^2) - i^2) (i^2 + j^2 + k^2))}{24 (1 + \nu) i j k}$$

où E et ν sont respectivement le module de Young et le coefficient de Poisson de l'objet (ou de l'élément Vol pour un objet hétérogène). Les coefficients de raideur des trois ressorts associés aux arêtes principales d'un volume Vol de dimensions $\{X_{Vol}, Y_{Vol}, Z_{Vol}\}$ sont données par $k_{Vol}(X_{Vol}, Y_{Vol}, Z_{Vol})$, $k_{Vol}(Y_{Vol}, X_{Vol}, Z_{Vol})$ et $k_{Vol}(Z_{Vol}, X_{Vol}, Y_{Vol})$. Dans la figure 3.7, la formulation du coefficient de raideur du ressort r est la suivante :

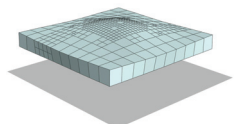
$$\begin{aligned} k_r &= k_{Vol_0}(Z_{Vol_0}, X_{Vol_0}, Y_{Vol_0}) \\ &+ k_{Vol_1}(Z_{Vol_1}, X_{Vol_1}, Y_{Vol_1}) \\ &+ k_{Vol_2}(Z_{Vol_2}, X_{Vol_2}, Y_{Vol_2}) \\ &+ k_{Vol_3}(Z_{Vol_3}, X_{Vol_3}, Y_{Vol_3}) \end{aligned}$$

où le volume Vol_i a pour dimension X_{Vol_i} , Y_{Vol_i} et Z_{Vol_i} avec $i \in \{0, \dots, 3\}$. Enfin, nous précisons que $L_0(r) = Z_{Vol_i}$.

- Le coefficient de raideur d'un ressort diagonal d interne à volume Vol_i est formulé de la manière suivante :

$$k_d = \frac{E (X_{Vol_i}^2 + Y_{Vol_i}^2 + Z_{Vol_i}^2)^2}{24 (1 + \nu) X_{Vol_i} Y_{Vol_i} Z_{Vol_i}}$$

Notons que, tout comme pour le cas 2D, la formulation de Baudet *et al.*, implique que quatre ressorts diagonaux internes à un élément, ont le même coefficient de raideur.



3.3 Simulation physique

Une fois que notre modèle unifié a été construit à partir d'un maillage initial surfacique ou volumique, et que les informations physiques ont été calculées, nous procédons au processus de la simulation physique. Ce processus a été explicité dans la partie 2.2.2. Nous rappelons qu'il consiste en une boucle de simulation où, à chaque itération, les nouvelles positions de l'ensemble des particules du SMR sont calculées. Pour cela, nous employons un schéma d'intégration d'Euler explicite symplectique. La boucle de simulation est décrite par l'algorithme 1.

Algorithme 1 : Boucle de simulation

Données : Un SMR.

Résultat : Calculer les nouvelles positions des particules du SMR.

Pour tous *Les ressorts du SMR* **faire**

┌ Calculer la force du ressort;

└ Accumuler cette force sur les particules extrémités du ressort;

Accumuler les forces externes induites par une interaction de l'utilisateur, au niveau des particules du SMR;

Pour tous *Les particules du SMR* **faire**

┌ Accumuler la gravité;

└ Calculer l'accélération de la particule en fonction du principe de la dynamique;

Intégrer cette accélération pour obtenir la nouvelle vitesse de la particule;

Multiplier cette nouvelle vitesse par un coefficient d'amortissement pour simuler la viscosité du milieu;

└ Intégrer cette nouvelle vitesse pour obtenir la nouvelle position de la particule;

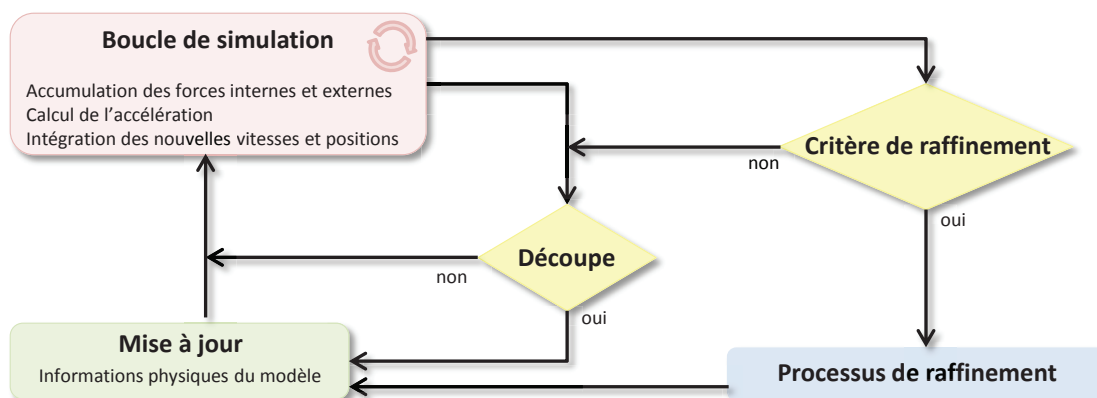


Figure 3.8 – Le processus de la simulation physique.

Au cours du processus de simulation, des changements topologiques comme celui de la découpe ou du raffinement, peuvent apparaître. Ils sont suivis de la mise à jour des informations physiques. Ce propos est illustré par la figure 3.8.

3.4 Changement topologique de la découpe

Nous rappelons que l'un des intérêts de notre modèle unifié est de faciliter la réalisation de changements topologiques dans le maillage, au cours de la simulation. En effet, la partie topologique apporte toutes les informations d'incidence et d'adjacence entre les cellules, pour des maillages surfaciques comme volumiques. Ceci permet de faciliter la réalisation de changements topologiques tout en garantissant la validité topologique du maillage. Elle facilite également la mise à jour des informations physiques qui lui sont associées. Enfin, la partie physique est appropriée à la réalisation interactive de changements topologiques au cours de la simulation, avec l'application des forces et la répartition des masses au niveau des particules. Le modèle unifié permet quant à lui d'éviter la redondance des informations et facilite la mise à jour des informations associées au modèle topologique après la réalisation de changements topologiques.

Nous illustrons cet aspect par l'implémentation d'une opération de découpe le long des arêtes du maillage en 2D, et le long des faces en 3D. Cette opération intervient au cours de la simulation et de manière répétée. Le processus de découpe présenté, consiste en une opération topologique de décousure suivie d'une mise à jour des informations associées au modèle topologique.

3.4.1 La partie topologique du processus de découpe

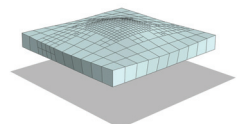
Cette première étape du processus de découpe repose sur la définition de l'opérateur de décousure faite dans la partie relative à l'introduction des cartes combinatoires (cf. la partie 2.3.3). Le processus de découpe prend en paramètre un brin. En 2D, si ce brin n'est pas 2-libre, alors il est 2-décousu. L'arête incidente est ainsi scindée en deux. Les deux faces qui étaient initialement incidentes à cette arête, ne sont plus adjacentes. En 3D, si ce brin n'est pas 3-libre, alors il est 3-décousu ainsi que les trois autres brins composant la face incidente à ce brin. La face incidente est alors scindée en deux. Les deux volumes qui étaient initialement incidents à cette face, ne sont plus adjacents.

Lors de la suppression de liaisons β_2 , en 2D, ou de liaisons β_3 , en 3D, des sommets ou des arêtes peuvent être scindés. La scission d'un sommet (*resp.* d'une arête) e en un sommet (*resp.* une arête) e' , entraîne la copie de l'attribut stocké par e , au niveau de e' . e' est donc associé à un attribut renfermant les mêmes informations que celles relatives à l'attribut de e . Enfin, ces informations peuvent nécessiter une mise à jour.

Les schémas 3.9 et 3.10 présentent respectivement un exemple de 2-décousure et de 3-décousure.

- La figure 3.9(a) débute le processus de découpe avec un modèle 2D composé de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . Les brins b_a et b_b sont alors reliés par β_2 . De plus, les faces F_0 et F_1 sont adjacentes selon l'arête constituée de ces deux brins. Enfin, cette arête porte le ressort r . La figure 3.9(b) montre le modèle présenté dans (a), suite à l'application de la première partie du processus de découpe sur le brin b_a . La liaison β_2 reliant les brins b_a et b_b est alors supprimée. Les faces F_0 et F_1 ne sont alors plus adjacentes entre elles.

De plus, la suppression de la liaison β_2 entre les brins b_a et b_b , provoque la scission du sommet s en le sommet s' . L'attribut stocké par s est alors copié au niveau de s' . De plus, l'arête incidente à b_a , portant le ressort r , est scindée en deux. Sa première partie incidente à F_0 , porte le ressort r . Sa deuxième partie, incidente à F_1 , porte la copie r' du ressort r . Les informations stockées au niveau des attributs scindés et de leur copie seront mises à jour lors de la deuxième partie du processus de découpe. Enfin, nous



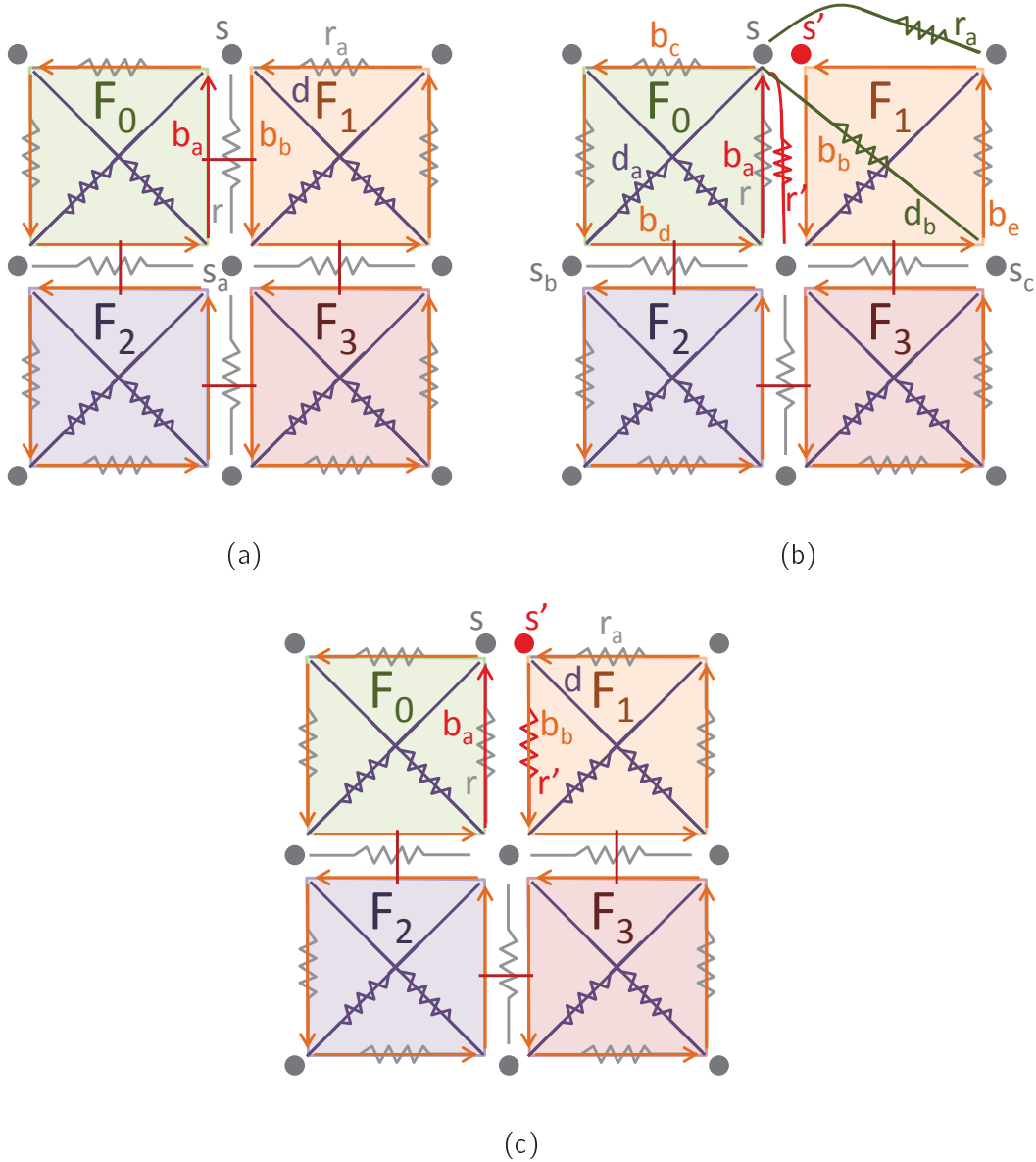


Figure 3.9 – Un exemple du processus de découpe, pour le cas 2D. (a) Un modèle 2D constitué de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . Le brin b_a est lié par β_2 à b_b . (b) Le modèle de (a) après l'application de la première partie du processus de découpe, consistant en la suppression de la liaison β_2 entre b_a et b_b , la scission du sommet s en un sommet s' et la scission du ressort r en un ressort r' . Notons que les ressorts r_a et r' et le ressort diagonal d_b ont pour extrémité le sommet s alors qu'ils devraient avoir pour extrémité le sommet s' . (c) Le modèle de (b) après l'application de la deuxième partie du processus de découpe, correspondant à la mise à jour des extrémités des ressorts en les reliant au nouveau sommet s' .

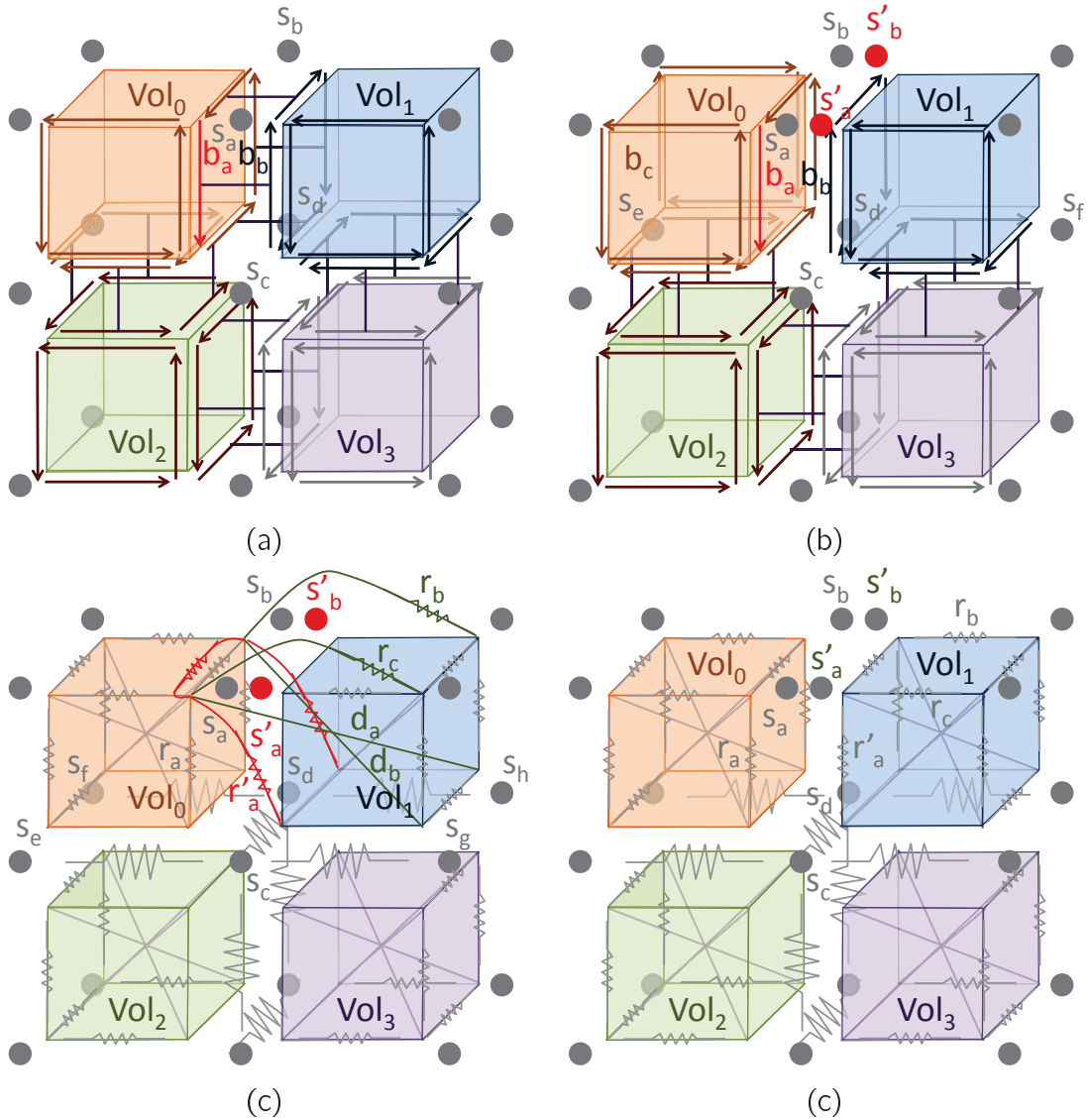
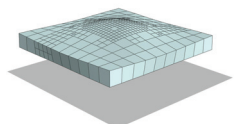


Figure 3.10 – Un exemple du processus de découpe, pour le cas 3D. (a) Un modèle 3D constitué de quatre hexaèdres : Vol₀, Vol₁, Vol₂ et Vol₃, cousus ensemble par β_3 . Le brin b_a est lié par β_3 à b_b . (b) Le modèle de (a) après l'application de la première partie du processus de découpe, correspondant à la suppression des liaisons β_3 entre les huit brins incidents à la face commune à Vol₀ et Vol₁. Les sommets s_a et s_b sont scindés en respectivement les sommets s'_a et s'_b . Pour des raisons de lisibilité, les ressorts du modèle ne sont pas représentés dans (a) et (b). (c) Le modèle de (b) montrant la scission du ressort r_a en le ressort r'_a . Les nouveaux ressorts sont représentés en rouge. Notons que trois ressorts sont scindés lors de ce processus de découpe. Les ressorts en rouge et en vert ont actuellement pour extrémité les sommets s_a et s_b . (d) Le modèle de (c), après l'application de la deuxième partie du processus de découpe, correspondant à la mise à jour des extrémités de ces derniers ressorts. Ils ont maintenant comme extrémités s'_a et s'_b au lieu de s_a et s_b . Pour des raisons de lisibilité, les brins ne sont plus représentés dans (c) et dans (d), mais ils représentent les ressorts du modèle contrairement à (a) et (b).



précisons que le sommet s_a n'est pas scindé car tous ses brins incidents sont accessibles successivement par application de β_{12} ou β_{02} , correspondant à la définition de l'orbite de la 0-cellule s_a .

- La figure 3.10 débute le processus de découpe avec un modèle 3D composé de quatre hexaèdres : Vol_0 , Vol_1 , Vol_2 et Vol_3 , cousus ensemble par β_3 . Les brins b_a et b_b sont alors reliés par β_3 . De plus, les hexaèdres Vol_0 et Vol_1 sont adjacents selon la face incidente à ces deux brins. La figure 3.10(b) montre le modèle présenté dans (a), suite à l'application de la première partie du processus de découpe sur le brin b_a . Quatre liaisons β_3 , dont celle reliant les brins b_a et b_b , sont alors supprimées. Les hexaèdres Vol_0 et Vol_1 ne sont alors plus adjacents entre eux.

De plus, la suppression des quatre liaisons β_3 , provoque la scission des sommets s_a et s_b en respectivement les sommets s'_a et s'_b . L'attribut stocké par s_a (*resp.* s_b) est alors copié au niveau de s'_a (*resp.* s'_b). De plus, la figure 3.10(c) montre que trois ressorts ont été créés. Parmi ces trois ressorts, le ressort r'_a correspond à une copie du ressort r_a . Les informations stockées au niveau des attributs scindés et de leur copie seront mises à jour lors de la deuxième partie du processus de découpe. Enfin, d'une manière similaire au 2D, les sommets s_c et s_d ne sont pas scindés car l'ensemble des brins qui leur sont incidents correspondent à l'orbite de la 0-cellule associée.

3.4.2 La partie physique du processus de découpe

Une opération topologique de 2-décousure ou de 3-décousure consiste en un changement topologique du modèle où des sommets et des arêtes peuvent être dupliqués. La duplication d'un sommet ou d'une arête entraîne alors la création de nouvelles particules et de nouveaux ressorts. Les informations physiques stockées au niveau des attributs Particule et Ressort doivent ainsi être mises à jour.

Scission d'un sommet

Nous rappelons que lorsqu'un sommet s est scindé en deux sommets s et s' , l'attribut associé au sommet s est copié. Cette copie est associée au sommet s' . Les informations physiques stockées au niveau des attributs de s et de s' doivent alors être mises à jour.

Masse Nous rappelons que la structure de données Particule stocke une masse. De plus, le calcul de cette dernière dépend des contributions des éléments incidents au sommet associé. Or, lorsque ce sommet est scindé, il est incident à moins d'éléments. Nous devons alors recalculer la masse de la particule en question. Nous procédons exactement de la même manière que pour l'initialisation (*cf.* la sous-section 3.2.1).

Dans la figure 3.9, le sommet s était initialement incident aux faces F_0 et F_1 . Sa masse dépendait alors de ces deux faces, tout comme celle de s' , puisque cette dernière a été copiée de s . Nous recalculons alors la masse de s en fonction de F_0 , et celle de s' , en fonction de F_1 .

Dans la figure 3.10, le sommet s_a (*resp.* s_b) était initialement incident aux hexaèdres Vol_0 et Vol_1 . Sa masse dépendait alors de ces deux hexaèdres, tout comme celle de s'_a (*resp.* s'_b), puisque cette dernière a été copiée de s_a (*resp.* s_b). Nous recalculons alors la masse de s_a (*resp.* s_b) en fonction de Vol_0 , et celle de s'_a (*resp.* s'_b), en fonction de Vol_1 .

Ressorts Les figures 3.9(b) et 3.10(c) illustrent la nécessité de mettre à jour les extrémités des ressorts dont les arêtes associées sont incidentes à un sommet dupliqué.

- Dans la figure 3.9(b), les ressorts r' et r_a doivent être mis à jour afin d'avoir comme extrémité le sommet s' et non plus le sommet s . Pour cela, nous parcourons l'ensemble des brins ayant pour extrémité s' , afin de mettre à jour les extrémités des ressorts qui leur sont associés. Nous appliquons alors $\text{parcours}(\beta_{21})$ où chaque brin parcouru voit ses extrémités mises à jour. Ce parcours est identique à celui permettant de récupérer l'ensemble des brins incidents à un sommet. Nous l'avons présenté dans le cadre du calcul de la masse d'une particule associée à un sommet en question (cf. la sous-section 3.2.1).
- Dans la figure 3.10(c), les cinq ressorts rouges et verts, comprenant entre autre r'_a , r_b et r_c , ont pour extrémité les sommets s_a et s_b , alors qu'ils devraient avoir comme extrémité les sommets s'_a et s'_b . Nous devons alors mettre à jour leurs extrémités. Pour cela, nous parcourons l'ensemble des brins ayant pour extrémité s'_a (resp. s'_b), afin de mettre à jour les extrémités des ressorts qui leur sont associés. Le parcours employé est le même que celui présenté dans le cadre du calcul de la masse d'une particule associée à un sommet en question (cf. la sous-section 3.2.1).

Ressorts diagonaux Nous devons ensuite traiter les ressorts diagonaux ayant pour extrémité un sommet ayant été scindé. Nous rappelons que la structure de données Particule stocke les ressorts diagonaux ayant pour extrémité le sommet associé. Les listes de ressorts diagonaux du sommet scindé et de sa copie doivent alors également être mises à jour. Le processus de mise à jour des ressorts diagonaux est décrit par l'algorithme 2.

Algorithme 2 : Mise à jour des ressorts diagonaux

Données : Un sommet s .

Résultat : La liste des ressorts diagonaux associés au sommet s , est mise à jour.

$Opp \leftarrow$ un ensemble de sommets;

Pour tous Les éléments e incidents à s faire

 Récupérer le sommet s_1 diagonalement opposé à s par rapport à e ;

 Stocker s_1 dans Opp ;

Pour tous Les ressorts diagonaux d stockés dans la liste de s faire

Si Une des extrémités de $d \in Opp$ **Alors**

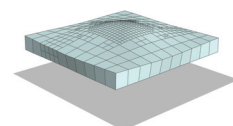
 Affecter l'autre extrémité de d à s ;

Sinon

 Supprimer d de la liste de s ;

- Dans la figure 3.9(b), le ressort diagonal d_b doit être mis à jour afin d'avoir comme extrémité le sommet s' et non plus le sommet s . De plus, la Particule associée au sommet s , ne doit plus stocker dans sa liste des ressorts diagonaux, le ressort diagonal d_b . Par contre, la Particule associée au sommet s' doit stocker le ressort diagonal d_b , dont son extrémité correspondant au sommet s aura été préalablement remplacée par le sommet s' .

Pour cela, nous appliquons l'algorithme 2 sur s . Nous commençons par récupérer l'ensemble des sommets diagonalement opposés au sommet s , pour chacune de ses faces incidentes. Il nous faut tout d'abord, récupérer l'ensemble des faces incidentes au sommet s , en employant $\text{parcours}(\beta_{21})$. Dans notre exemple, le sommet s n'est incident



qu'à une seule face : F_0 . Puis, nous appliquons ensuite β_{11} sur le brin b_c récupéré lors du précédent parcours. Nous obtenons alors le brin b_d , partant du sommet s_b . s_b est le sommet diagonalement opposé à s , par rapport à la face F_0 . s_b est alors stocké dans *Opp*. Puis, nous parcourons l'ensemble des ressorts diagonaux stockés dans la liste de s : d_a et d_b . d_a ayant pour extrémité un sommet appartenant à *Opp* : s_b . Ce ressort diagonal est conservé. Par contre, d_b ayant pour extrémité un sommet n'appartenant pas à *Opp* : s_c , il est supprimé de la liste des ressorts diagonaux de s . Enfin, nous procédons de la même manière pour s' . Le ressort diagonal d_a est supprimé de la liste de s' . Par contre, le ressort diagonal d_b est conservé et son extrémité s est remplacée par s' . Notons que le conteneur stockant l'ensemble des ressorts diagonaux ne perd, ni ne gagne de ressorts diagonaux.

- Dans la figure 3.10(c), le ressort diagonal d_a (resp. d_b) doit être mis à jour afin d'avoir comme extrémité le sommet s'_a (resp. s'_b) et non plus le sommet s_a (resp. s_b). De plus, la Particule associée au sommet s_a (resp. s_b), ne doit plus stocker dans sa liste des ressorts diagonaux, le ressort diagonal d_a (resp. d_b). Par contre, la Particule associée au sommet s'_a (resp. s'_b) doit stocker le ressort diagonal d_a (resp. d_b), dont son extrémité correspondant au sommet s_a (resp. s_b) aura été préalablement remplacée par le sommet s'_a (resp. s'_b).

Pour cela, comme présenté en 2D, nous appliquons l'algorithme 2 sur s_a . Nous commençons par récupérer l'ensemble des sommets diagonalement opposés au sommet s_a , pour chacune de ses hexaèdres incidents. Il nous faut tout d'abord récupérer l'ensemble des hexaèdres incidents au sommet s . Nous employons alors le même parcours que celui présenté dans le cadre du calcul de la masse d'une particule associée à un sommet en question (cf. la sous-section 3.2.1). Dans notre exemple, le sommet s_a n'est incident qu'à un seul hexaèdre : Vol_0 . Puis, nous appliquons ensuite β_{00200} sur le brin b_a récupéré lors du précédent parcours. Nous obtenons alors le brin b_c , partant du sommet s_e (cf. la figure 3.10(b)). s_e est le sommet diagonalement opposé à s_a , par rapport à l'hexaèdre Vol_0 . s_e est alors stocké dans *Opp*. Puis, nous parcourons l'ensemble des ressorts diagonaux stockés dans la liste de s_a . d_a ayant pour extrémité, un sommet n'appartenant pas à *Opp* : s_b , il est supprimé de la liste des ressorts diagonaux de s_a . Puis, nous procédons de la même manière pour s'_a . Le ressort diagonal d_a est conservé dans la liste des ressorts diagonaux de s'_a et son extrémité s_a est remplacée par s'_a . Par contre, l'autre ressort diagonal interne à Vol_0 est supprimé de sa liste. Enfin, nous procédons de la même manière pour s_b et s'_b . Notons que le conteneur stockant l'ensemble des ressorts diagonaux ne perd, ni ne gagne de ressorts diagonaux.

Nous précisons que les autres informations que sont la position, la vitesse, l'accélération, la somme des forces ainsi que celles relatives aux contraintes appliquées à la particule, ne nécessitent pas d'être mises à jour.

Scission d'une arête

Nous rappelons que lorsqu'une arête a est scindée en deux arêtes a et a' , l'attribut associé à l'arête a est copié. Cette copie est associée à l'arête a' . Les informations physiques stockées au niveau des attributs de a et de a' doivent alors être mises à jour.

- Dans la figure 3.9(b), le ressort r est scindé en le ressort r' . Le ressort r' doit alors être mis à jour afin d'avoir comme extrémité le sommet s' et non plus le sommet s . Or, cette mise à jour a été effectuée lors du scission de s .

Nous devons mettre à jour la raideur des ressorts r et r' . En effet, ces derniers sont maintenant incidents à moins de faces. Pour calculer la nouvelle raideur de r (*resp.* r'), nous parcourons l'ensemble des faces incidentes à r (*resp.* r'). Nous procédons exactement de la même manière que pour l'initialisation (*cf.* la sous-section 3.2.2).

- Dans la figure 3.10(c), les trois ressorts rouges, comprenant entre autre le ressort r'_a , ont pour extrémité les sommets s_a ou s_b , alors qu'ils devraient avoir comme extrémité respectivement les sommets s'_a ou s'_b . Nous devons alors mettre à jour leurs extrémités. Or, cette mise à jour a été effectuée lors du scission de s_a et de s_b .

Nous devons mettre à jour la raideur de ces trois ressorts rouges. En effet, ces derniers sont maintenant incidents à moins d'hexaèdres. Pour calculer la nouvelle raideur de ces trois ressorts, nous parcourons pour chacun d'entre eux, l'ensemble des hexaèdres qui leurs sont incidents. Nous procédons exactement de la même manière que pour l'initialisation (*cf.* la sous-section 3.2.2).

Nous précisons que les autres informations que sont la longueur initiale et l'amortissement ne nécessitent pas d'être mises à jour.

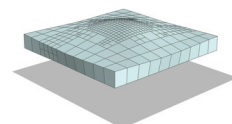
3.5 Conclusion

Dans ce chapitre, nous avons décrit la construction de notre modèle unifié LCC+SMR, prenant en entrée un maillage constitué de quadrilatères en 2D et d'hexaèdres en 3D. A partir de ces données géométriques et topologiques, nous créons une carte combinatoire. Puis, nous associons une géométrie à cette dernière, définissant un LCC. Enfin, la dernière étape de la construction consiste en l'association des attributs relatifs au SMR au LCC.

Pour le modèle physique, nous avons fait le choix d'un système masses-ressorts, où les ressorts sont positionnés sur les arêtes des éléments du maillage ainsi que de manière interne et diagonale aux éléments. La masse des particules est calculée en fonction de l'aire ou du volume des éléments qui lui sont incidents, ainsi que de leur densité surfacique ou leur densité volumique dans le cas d'un objet hétérogène. Enfin, les raideurs sont déterminées grâce à une formulation obtenue depuis la méthode analytique proposée par Baudet *et al.*

Une fois la construction de notre modèle unifié et l'initialisation de ses paramètres physiques effectuées, nous procédons à la simulation de notre modèle, où les forces internes induites des ressorts et les forces externes comme la gravité ou les interactions utilisateurs, sont accumulées à chaque itération au niveau des particules du système. Une fois cette accumulation réalisée, pour chaque particule, son accélération est calculée ainsi que sa nouvelle vitesse et sa nouvelle position grâce à un schéma d'intégration d'Euler explicite symplectique.

Enfin, nous décrivons le changement topologique de la découpe, composé d'une opération topologique et d'une mise à jour des informations physiques embarquées au niveau du modèle topologique. Ce changement topologique est réalisé selon les arêtes en 2D et les faces en 3D.



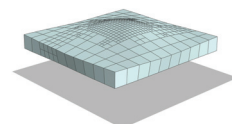
Chapitre 4

« ...Ah bon ? » fit tristement Morty. Il s'était trompé, il y avait bien une lumière au bout du tunnel, et c'était un lance-flammes. »
(Les Annales du Disque-Monde, Mortimer - Terry Pratchett)

Raffinement local

Sommaire

4.1	Notions et concepts de base	107
4.1.1	Niveau de subdivision	107
4.1.2	Éléments hiérarchiques	110
4.1.3	Niveau de subdivision des éléments hiérarchiques	113
4.1.4	Brin hiérarchique	114
4.1.5	Jonction T	115
4.1.6	Éléments non conformes	116
4.2	Algorithmes de base	117
4.2.1	Récupérer le prochain brin hiérarchique	117
4.2.2	Tester si une face est simple	117
4.2.3	Trouver le prochain brin hiérarchique d'une face	118
4.2.4	Calcul du niveau de subdivision d'un élément	120
4.2.5	Trouver un brin hiérarchique d'un élément	123
4.3	Subdivision régulière d'un quadrilatère en quatre sous-quadrilatères	124
4.3.1	Présentation intuitive	124
4.3.2	Subdivision d'un quadrilatère inscrit dans un modèle localement raffiné	125
4.3.3	Initialisation et mise à jour des informations physiques	128
4.4	Subdivision régulière d'un hexaèdre en huit sous-hexaèdres	134
4.4.1	Présentation intuitive	134
4.4.2	Subdivision d'un hexaèdre inscrit dans un modèle localement raffiné	138
4.4.3	Initialisation et mise à jour des informations physiques	145
4.5	Découpe d'un objet 2D localement raffiné	161
4.5.1	Mise à jour des extrémités de projection	163
4.5.2	Enlever la propriété de jonction T à un sommet	164
4.5.3	Supprimer une jonction T	164
4.6	Découpe d'un objet 3D localement raffiné	165
4.6.1	Mise à jour des extrémités de projection	166
4.6.2	Enlever la propriété de jonction T à un sommet	169



4.6.3	Supprimer une jonction T	170
4.7	Conclusion	171

La présentation que nous avons faite de notre modèle dans le chapitre précédent, n'inclut pas la gestion du changement topologique induit par le raffinement d'un élément. En effet, chaque quadrilatère était alors toujours composé de quatre arêtes, et chaque hexaèdre de douze arêtes et de six faces. Or, le raffinement d'un élément du maillage peut invalider cette propriété. En effet, en 2D, le côté d'un quadrilatère peut alors être composé de plusieurs arêtes, et en 3D, une face peut alors être constituée de plusieurs sous-faces. De plus, en 3D, une face ou sous-face peut également avoir ses côtés composés de plusieurs sous-arêtes. Ce constat nécessite ainsi la définition de nouvelles notions permettant le raffinement local du maillage. Dans ce sens, nous allons tout d'abord détailler les nouvelles informations ajoutées à notre modèle. Puis, nous présenterons les algorithmes de base permettant le parcours de notre modèle localement raffiné. Dans un second temps, nous expliciterons le processus de subdivision d'un quadrilatère en quatre sous-quadrilatères et d'un hexaèdre en huit sous-hexaèdres. Pour cela, nous ferons tout d'abord une présentation intuitive du processus de base, puis nous approfondirons cette première présentation pour expliciter le processus de subdivision d'un élément inscrit dans un modèle localement raffiné. Cette dernière partie plus technique, sera suivie de l'explication de l'initialisation et de la mise à jour des informations physiques du système. Cette étape nécessite la gestion des jonctions T dont leur création est inhérente à notre processus de subdivision. Enfin, nous détaillerons la modification du processus de découpe pour un modèle localement raffiné, avec la gestion des jonctions T.

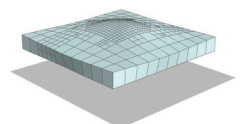
Nous rappelons que plus la résolution d'un maillage est grande, plus la simulation physique associée est précise mais aussi plus coûteuse en temps. Pour cela, un compromis est réalisé entre la précision et le temps par l'emploi d'un maillage multi-résolution présentant des éléments de différents niveaux de subdivision qui ont été raffinés en fonction de critères. Une solution classique consiste à débiter la simulation avec un maillage relativement grossier dont les éléments qui le nécessitent seront raffinés au moment opportun. Cette solution est permise par la définition de nos opérations de raffinement d'un quadrilatère et d'un hexaèdre. Les éléments du maillage sont ainsi raffinés selon un critère d'ordre géométrique reposant sur la déformation des ressorts diagonaux.

4.1 Notions et concepts de base

L'ensemble des informations stockées au niveau des structures de données Particule et Ressort, qui ont été présentées, ne suffisent pas à gérer le changement topologique du raffinement local. Des informations doivent ainsi être ajoutées. Dans ce sens, nous définissons dans cette partie les concepts de niveau de subdivision, de face hiérarchique, d'arête hiérarchique, de sommet hiérarchique, de brin hiérarchique, de jonction T frontalière, de jonction T centrée, d'élément (face ou volume) non conforme et enfin d'élément non conforme relativement à une jonction T.

4.1.1 Niveau de subdivision

Chaque arête a stocke son **niveau de subdivision** noté I_a . Tous les brins incidents à a ont alors accès à I_a . Lors de l'initialisation, les niveaux de subdivision associés à chaque brin prennent la valeur de 1. En effet, nous faisons l'hypothèse que l'objet initial n'a pas été préalablement localement subdivisé. Au cours du processus de raffinement, si une arête est scindée en deux parties, alors le niveau de subdivision des deux arêtes résultantes est doublé. De plus, comme pour les arêtes, chaque élément E (face en 2D ou volume en 3D) stocke un niveau de subdivision noté I_E . Il vaut initialement 1 et sa valeur est également liée au



nombre de fois qu'il a été subdivisé. Notons que le niveau de subdivision des sous-éléments issus du raffinement d'un élément, est le double du niveau de subdivision de cet élément initial.

Les figures 4.1 et 4.2 illustrent le concept de niveau de subdivision, respectivement pour les cas 2D et 3D.

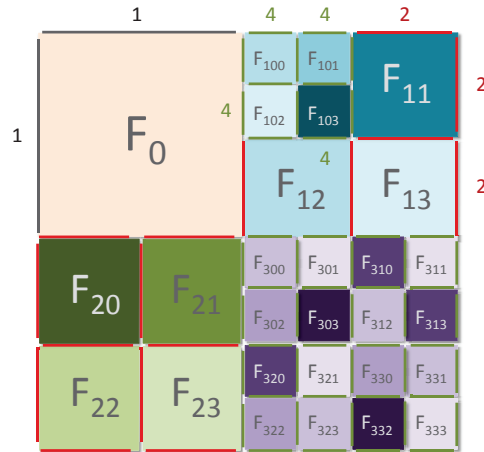


Figure 4.1 – La notion de niveau de subdivision pour un modèle 2D. Un modèle 2D initialement composé de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . La figure montre les niveaux de subdivision des différentes arêtes du modèle, suite à une première subdivision des quadrilatères : F_1 , F_2 et F_3 , puis une deuxième subdivision des quadrilatères : F_{10} et F_{3i} avec $i \in \{0, \dots, 3\}$. Les arêtes de niveau de subdivision 1, 2 et 4 sont représentées respectivement en noir, rouge et vert.

- **Cas 2D.** La figure 4.1 montre un modèle 2D initialement constitué de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 . F_1 a été subdivisé en $\{F_{10}, F_{11}, F_{12}$ et $F_{13}\}$. F_{10} a été subdivisé en $\{F_{100}, F_{101}, F_{102}$ et $F_{103}\}$. F_2 a été subdivisé en $\{F_{20}, F_{21}, F_{22}$ et $F_{23}\}$. Enfin, F_3 a été subdivisé en $\{F_{30}, F_{31}, F_{32}$ et $F_{33}\}$. Ces derniers ont également été subdivisés respectivement en $\{F_{300}, F_{301}, F_{302}$ et $F_{303}\}$, $\{F_{310}, F_{311}, F_{312}$ et $F_{313}\}$, $\{F_{320}, F_{321}, F_{322}$ et $F_{323}\}$ et $\{F_{330}, F_{331}, F_{332}$ et $F_{333}\}$.

Cette figure montre le niveau de subdivision de chacune des arêtes et faces du modèle. Par exemple, la face F_0 n'a jamais été subdivisée, son niveau de subdivision vaut donc 1. Par contre, les faces : F_{20} , F_{21} , F_{22} et F_{23} sont issues de la subdivision de F_2 . Or, F_2 ayant eu initialement un niveau de subdivision égal à 1, ces quatre sous-faces ont alors un niveau de subdivision égal à 2. Il en va de même pour les faces F_{11} , F_{12} et F_{13} . Enfin, les autres faces sont issues d'une deuxième subdivision, elles ont donc un niveau de subdivision égal à 4. Par exemple, les faces F_{100} , F_{101} , F_{102} et F_{103} ont un niveau de subdivision égal à 4, car elles sont issues de la subdivision de F_{10} qui avait un niveau de subdivision égal à 2.

Le côté gauche et le côté haut de F_0 ont un niveau de subdivision égal à 1, car ils n'ont pas été subdivisés. Par contre, son côté bas a été subdivisé en deux sous-arêtes, suite à la subdivision de F_2 . Ces dernières ont alors un niveau de subdivision égal à 2. Son côté droit a également été subdivisé une première fois en deux sous-arêtes, conformément à la subdivision de F_1 . Ces dernières ont alors chacune un niveau de subdivision égal à 2. Puis, une des ces sous-arêtes a été subdivisée une deuxième fois en deux sous-arêtes, suite à la subdivision de la sous-faces F_{10} de F_1 . Ces dernières ont alors un niveau de subdivision égal à 4.

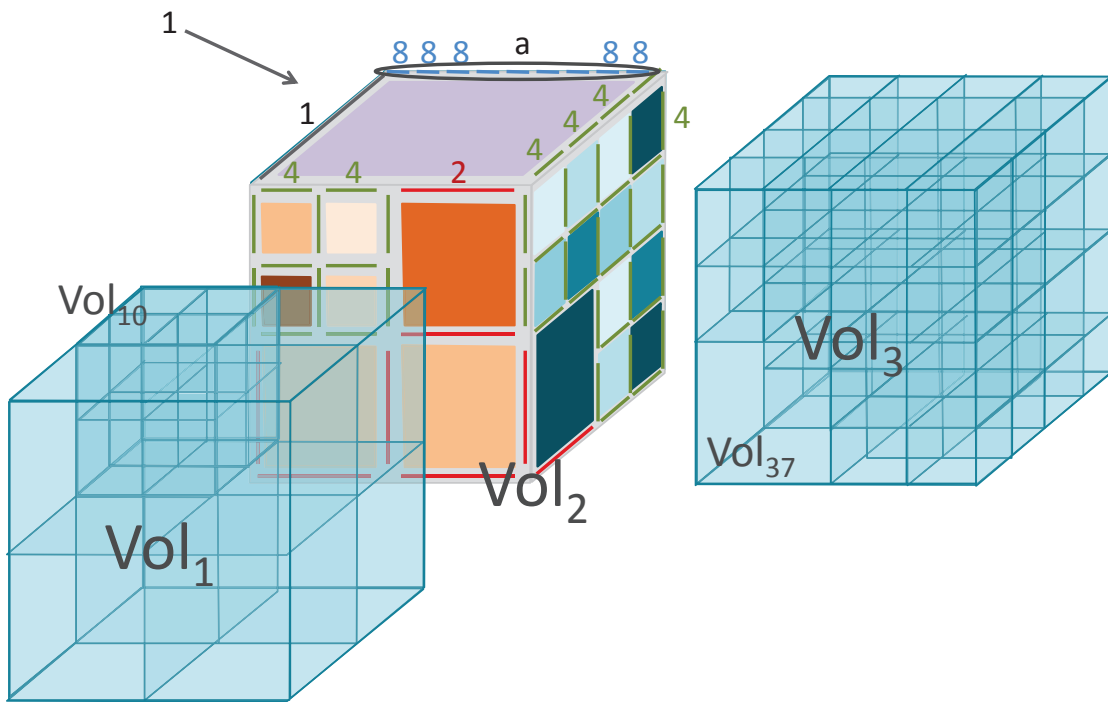
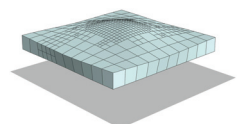


Figure 4.2 – La notion de niveau de subdivision pour un modèle 3D. Un modèle 3D initialement composé de trois hexaédres : Vol_1 , Vol_2 et Vol_3 , cousus ensemble par β_3 . Vol_2 n'a pas été subdivisé et a donc un niveau de subdivision égal à 1, contrairement aux autres volumes adjacents correspondant aux sous-volumes de Vol_1 et Vol_3 . Ces derniers ont un niveau de subdivision supérieur à Vol_2 . Le schéma montre les niveaux de subdivision de quelques arêtes composant les côtés de Vol_2 . Les arêtes de niveau de subdivision 1, 2, 4 et 8 sont représentées respectivement en noir, rouge, vert et bleu.



- **Cas 3D.** La figure 4.2 permet d'expliciter la notion de niveau de subdivision pour un modèle 3D. Ce dernier était initialement composé de trois volumes : Vol_1 , Vol_2 et Vol_3 . Puis, Vol_1 a été subdivisé une première fois en huit sous-volumes, suivi par la seconde subdivision d'un de ses sous-volumes Vol_{10} . Les sept sous-volumes de Vol_1 ont alors un niveau de subdivision égal à 2, et les huit sous-volumes de Vol_{10} ont un niveau de subdivision égal à 4. Vol_3 a également été subdivisé en huit sous-volumes. Puis, l'ensemble de ses huit sous-volumes ont été subdivisés une nouvelle fois sauf : Vol_{37} . Le sous-volume de Vol_3 a un niveau de subdivision égal à 2, et l'ensemble des autres sous-volumes issus de Vol_3 , ont un niveau de subdivision égal à 4. Enfin, Vol_2 n'a jamais été subdivisé, impliquant un niveau de subdivision : $l_{Vol_2} = 1$. Le modèle est alors composé de soixante-treize volumes : $(7 + 8)$ pour Vol_1 + 1 pour Vol_2 + $(7 * 8 + 1)$ pour Vol_3 .

La face avant de Vol_2 a été décomposée en sept sous-faces conformément aux différentes étapes de subdivision de Vol_1 . La face droite de Vol_2 est décomposée en treize sous-faces suite à la subdivision de Vol_3 . Enfin la configuration de la face du haut de Vol_2 indique qu'il n'y a pas de volume à sa gauche ou qu'il n'a pas été subdivisé. La même hypothèse est faite pour le volume du haut de Vol_2 . Par contre, son arête a , décomposée en huit sous-arêtes, indique que le volume situé au-dessus du volume derrière Vol_2 , a ses volumes incidents à a qui ont été subdivisés trois fois.

Enfin, l'arête du haut de la face avant de Vol_2 avait initialement un niveau de subdivision égal à 1. Puis, elle a été scindée en deux, suite à une première subdivision de Vol_1 . Les deux sous-arêtes résultantes ont alors un niveau de subdivision égal à 2. Une de ces sous-arêtes a ensuite été une nouvelle fois scindée en deux, suite à la seconde subdivision de Vol_{10} . Les deux parties résultantes ont alors un niveau de subdivision égal à 4.

4.1.2 Éléments hiérarchiques

Nous venons de définir le niveau de subdivision d'une arête, d'une face ou encore d'un volume pour le cas 3D. Dans la figure 4.2, nous avons constaté que le côté bas et droit de F_0 , avaient été subdivisés, lors de la subdivision de faces adjacentes à F_0 : F_1 , F_{10} et F_2 . Ces côtés sont maintenant constitués de plusieurs sous-arêtes, alors que la face F_0 n'a, quant à elle, pas été subdivisée. Ce constat, qui est également présent pour le cas 3D, montre la nécessité de pouvoir distinguer les différents côtés d'un élément quel que soit le niveau de subdivision des éléments qui lui sont adjacents. Nous définissons alors dans cette partie, les concepts de face, arête et sommet hiérarchiques.

- Une **face hiérarchique** (*resp.* **arête hiérarchique**) **d'un élément** correspond au côté de cet élément. Elle correspond à l'ensemble des faces (*resp.* arêtes) issues de la subdivision d'une même face (*resp.* arête) d'un élément. Si la face (*resp.* l'arête) n'a pas été subdivisée, la face (*resp.* l'arête) hiérarchique correspond à cette face (*resp.* arête). Le concept de face hiérarchique existe seulement en 3D, tandis que celui d'arête hiérarchique est défini en 2D comme en 3D. Enfin, nous nommons **face simple**, une face hiérarchique qui n'a pas été subdivisée en plusieurs sous-faces. Par opposition, nous appelons **face complexe**, une face hiérarchique qui a été subdivisée en plusieurs sous-faces. Nous précisons que la notion de face simple vs. face complexe est seulement définie pour le cas 3D.

Arête hiérarchique, pour le cas 2D. La figure 4.3 montre un modèle 2D constitué initialement de deux quadrilatères adjacents F_0 et F_1 . F_1 a été subdivisé en F_{1i} avec

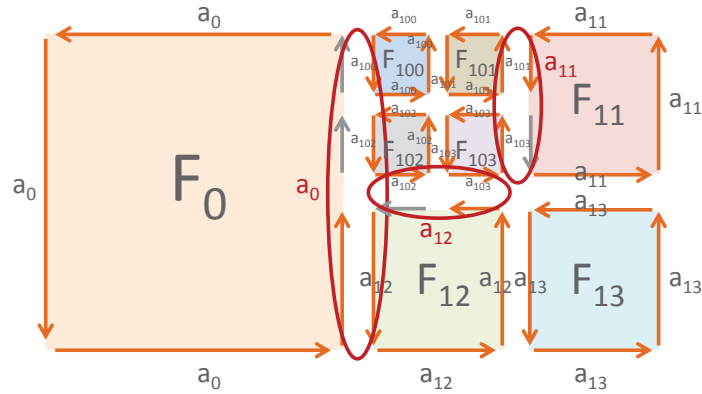


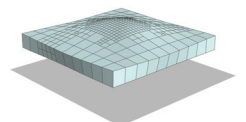
Figure 4.3 – La notion d’arête hiérarchique pour un modèle 2D. Un modèle 2D initialement composé de deux quadrilatères : F_0 et F_1 cousus ensemble par β_2 . Le schéma représente l’ensemble des brins du modèle. Les a_i sont des arêtes hiérarchiques de F_i avec $i \in \{0, 100, 101, 102, 103, 11, 12, 13\}$. Les arêtes hiérarchiques entourées de rouge sont les seules constituées de plusieurs sous-arêtes. Les brins orange correspondent aux brins hiérarchiques.

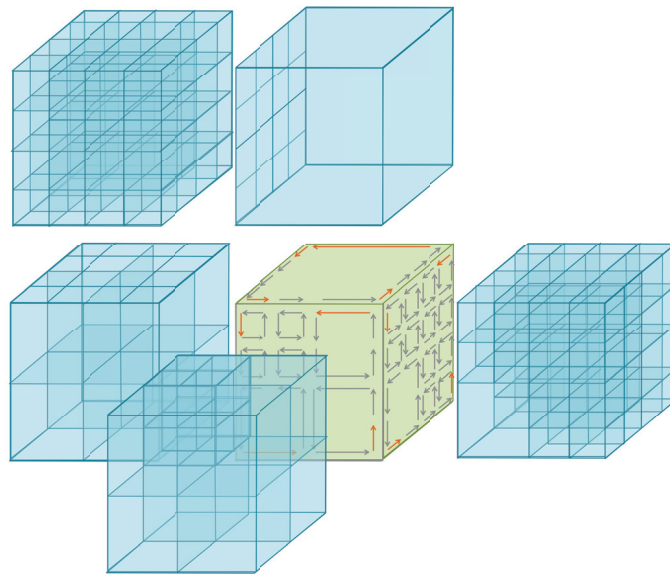
$i \in \{0, \dots, 3\}$. Puis F_{10} a été subdivisé en F_{10i} . Les a_n labellisent les arêtes hiérarchiques des éléments F_n avec $n \in \{0, 11, 12, 13, 100, 101, 102, 103\}$. Nous illustrons le concept d’arête hiérarchique avec l’exemple suivant. Le côté gauche de la face F_0 correspond à l’arête hiérarchique a_0 , qui est constituée d’une seule arête. Ce n’est par contre pas le cas pour son côté droit, correspondant à une arête hiérarchique constituée de trois sous-arêtes : a_{100} , a_{102} et a_{12} . Cette subdivision en trois sous-arêtes est due à la subdivision de F_1 suivie de celle de F_{10} .

Arête et face hiérarchiques, pour le cas 3D. La figure 4.4 illustre le concept de face et d’arête hiérarchiques pour le cas 3D. Elle représente un modèle 3D initialement composé de six volumes. Nous nommons Vol l’hexaèdre vert, dont le schéma représente trois de ses six faces hiérarchiques : F_0 , F_1 et F_2 . Ces dernières ont été subdivisées suite à la subdivision des volumes adjacents à Vol , représentés en bleu. Les a_0 , a_1 et a_2 sont les arêtes hiérarchiques de Vol et également, par définition, celles des trois faces hiérarchiques de Vol représentées. La face hiérarchique F_0 est constituée de sept sous-faces. L’arête hiérarchique a_0 est composée de trois sous-arêtes et a_1 et a_2 ont été subdivisée en quatre sous-arêtes. De manière récursive, les sous-faces des faces hiérarchiques peuvent également avoir des arêtes hiérarchiques composées de sous-arêtes. Ce cas est illustré par la sous-face F_{01} de F_0 .

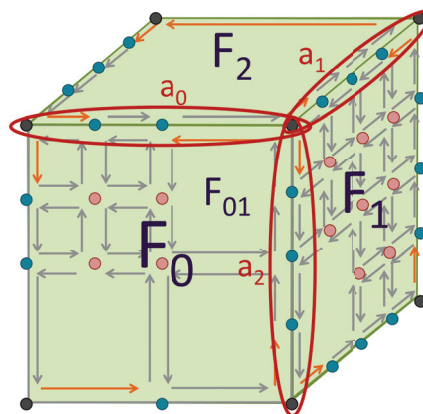
Nous revenons plus en détails sur le concept de face hiérarchique à un volume donné. Nous définissons pour cela, la notation $F_{i|Vol_j}$, correspondant à l’ensemble des brins de F_i incidents au volume Vol_j . Dans la figure 4.5, $F_{0|Vol_0}$ est la face hiérarchique de Vol_0 . Elle a été subdivisée en quatre sous-faces : $F_{00|Vol_0}$, $F_{01|Vol_0}$, $F_{02|Vol_0}$ et $F_{03|Vol_0}$, suite à la subdivision de Vol_1 en huit sous-volumes. $F_{00|Vol_{10}}$ (resp. $F_{01|Vol_{11}}$, $F_{02|Vol_{12}}$ et $F_{03|Vol_{13}}$) est alors une face hiérarchique du sous-volume Vol_{10} (resp. Vol_{11} , Vol_{12} et Vol_{13}). Notons que les faces hiérarchiques $F_{00|Vol_{10}}$, $F_{01|Vol_{11}}$, $F_{02|Vol_{12}}$ et $F_{03|Vol_{13}}$, ne sont constituées que d’une seule face.

- Un **sommet hiérarchique** est une extrémité d’une arête hiérarchique. En 2D, un quadrilatère a toujours quatre sommets hiérarchiques, et un hexaèdre possède toujours huit sommets hiérarchiques. Notons que dans le cas 3D, les faces hiérarchiques d’un volume possèdent également quatre sommets hiérarchiques.





(a)



(b)

Figure 4.4 – La notion d'arête et de face hiérarchique pour un modèle 3D. (a) Un modèle 3D composé de six hexaèdres cousus ensemble par β_3 . (b) Zoom du volume vert Vol, qui n'a pas été subdivisé. Cependant, ses faces ont été subdivisées car elles sont incidentes à des volumes qui l'ont été. Les a_0 , a_1 et a_2 sont des arêtes hiérarchiques de Vol ainsi que de ses faces hiérarchiques : F_0 , F_1 et F_2 . Les brins orange représentent les brins hiérarchiques de Vol contrairement aux autres brins qui sont gris. Les sommets bleus correspondent aux jonctions T frontalières et les sommets roses aux jonctions T centrées.

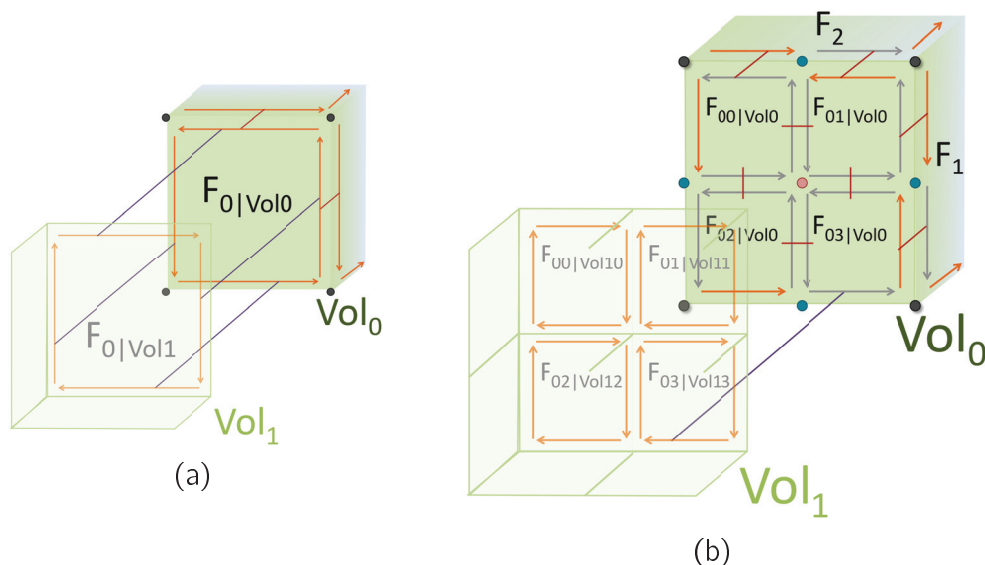


Figure 4.5 – La définition de brin hiérarchique pour une face incidente à deux volumes n’ayant pas le même niveau de subdivision. (a) Un modèle 3D composé de deux hexaédres : Vol_0 et Vol_1 , cousus ensemble par β_3 . Ils sont incidents à une même face F_0 . (b) Le modèle de (a) après la subdivision de Vol_1 en Vol_{1i} avec $i \in \{0, \dots, 7\}$. Il représente l’ensemble des brins incidents à : $F_{0|Vol_0}$, $F_{00|Vol_{10}}$, $F_{01|Vol_{11}}$, $F_{02|Vol_{12}}$ et $F_{03|Vol_{13}}$. Les liens rouges (resp. violets) correspondent aux liaisons β_2 (resp. β_3) et les brins orange sont hiérarchiques. Pour des raisons de lisibilité, nous ne représentons pas toutes les liaisons β_3 .

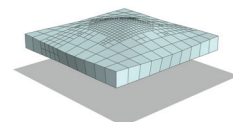
Cas 2D. Dans la figure 4.6, le quadrilatère F_0 contient sept sommets dont quatre sont hiérarchiques. Ces derniers sont représentés en noir sur la figure. Ils correspondent aux extrémités des arêtes hiérarchiques de F_0 , qui sont : a_{00} , a_{01} , a_{02} et a_{03} . Les quadrilatères F_{20} et F_{312} sont quant à eux composés de quatre sommets qui sont tous hiérarchiques. En effet, les éléments qui leur sont adjacents n’ont pas un niveau de subdivision plus grand qu’eux.

Cas 3D. Dans la figure 4.4(b), seuls huit sommets (représentés en noir) parmi l’ensemble de ceux composant Vol , sont hiérarchiques. Ils correspondent aux différentes extrémités des arêtes hiérarchiques de Vol . Nous rappelons que a_0 , a_1 et a_2 sont trois des douze arêtes hiérarchiques de Vol .

Pour conclure la définition de ce concept d’élément hiérarchique, nous précisons que la notion d’arête et de sommet hiérarchique est relative à un élément donné, tout comme la notion de face hiérarchique pour le cas 3D.

4.1.3 Niveau de subdivision des éléments hiérarchiques

En 2D, le niveau de subdivision d’une face est égal au niveau de subdivision de ses arêtes hiérarchiques. De plus, en 3D, le niveau de subdivision d’un volume est égal au niveau de subdivision de ses faces hiérarchiques, dont chacune d’entre elles a un niveau de subdivision égal à celui de ses arêtes hiérarchiques. Par transitivité, le niveau de subdivision d’un volume est égal à celui de ses arêtes hiérarchiques. Le calcul du niveau de subdivision d’une arête hiérarchique consiste à additionner l’inverse des niveaux de subdivision des sous-arêtes la composant. Nous précisons que cette somme s’effectue sur des formes fractionnaires, c’est-à-dire que nous ne calculons pas les inverses des niveaux de subdivision pour les additionner. Réaliser un calcul en conservant les formes fractionnaires permet de garantir la précision



numérique de notre résultat. C'est également pour cette raison que nous avons choisi des niveaux de subdivision valant initialement 1 et qui sont doublés à chaque subdivision. Ce choix permet également des optimisations par décalage de bits. Notons, que lorsque nous additionnons deux formes fractionnaires, la forme fractionnaire résultante est simplifiée pour éviter de dépasser la taille des entiers non signés. En effet, les comparaisons entre formes fractionnaires se font à dénominateur égal, en comparant les numérateurs.

- **Cas 2D.** La somme des inverses des niveaux de subdivision de chaque sous-arête d'un côté donne le niveau de subdivision de l'arête hiérarchique et donc de la face. Par exemple, dans la figure 4.1, nous faisons l'hypothèse que la face F_0 a un niveau de subdivision égal à 1. Son côté droit est composé de deux sous-arêtes ayant un niveau de subdivision égal à 4 et d'une sous-arête ayant un niveau de subdivision égal à 2. En additionnant l'inverse des niveaux de subdivision de ces trois sous-arêtes, nous obtenons $1 (\frac{1}{4} + \frac{1}{4} + \frac{1}{2})$, correspondant bien au niveau de subdivision de F_0 . Le niveau de subdivision de F_0 est donc bien égal au niveau de subdivision de ses arêtes hiérarchiques.
- **Cas 3D.** De manière similaire au cas 2D, la somme des inverses des niveaux de subdivision de chaque sous-arête issue d'une même arête initiale d'un volume, donne le niveau de subdivision de ce volume. Par exemple, dans la figure 4.2, nous considérons l'arête du haut de la face avant de Vol_2 , notée a_1 . En additionnant l'inverse des niveaux de subdivision des trois sous-arêtes issue de a_1 , nous obtenons $1 (\frac{1}{4} + \frac{1}{4} + \frac{1}{2})$, correspondant bien au niveau de subdivision du volume Vol_2 . Nous obtenons le même résultat en additionnant l'inverse des niveaux de subdivision des huit sous-arêtes composant a . Le niveau de subdivision de Vol_2 est donc bien égal au niveau de subdivision de ses arêtes hiérarchiques mais également à celui de ses faces hiérarchiques.

4.1.4 Brin hiérarchique

Après avoir défini la notion de sommet, d'arête et de face hiérarchiques d'un élément, permettant d'identifier les différents côtés de ce dernier, nous présentons dans cette partie, la notion de **brin hiérarchique d'un élément**. Un brin hiérarchique est le premier brin d'une arête hiérarchique d'un élément donné, incident à cet élément. En 2D, cet élément est une face et en 3D, une face hiérarchique d'un volume. Notons qu'un brin incident à un élément et partant d'un sommet hiérarchique de cet élément, est hiérarchique à cet élément. A l'initialisation, tous les brins sont marqués comme étant hiérarchiques. En effet, comme nous l'avons vu précédemment, une arête hiérarchique qui n'a jamais été subdivisée ne possède qu'un seul brin incident à l'élément. Cependant, lorsqu'une arête hiérarchique est subdivisée, les nouveaux brins la composant ne seront pas marqués comme étant hiérarchiques, par contre le brin hiérarchique initial restera marqué. Ce marquage permet d'identifier le début d'une arête hiérarchique sans devoir se référer à la géométrie du modèle, et permet également de retrouver facilement les sommets hiérarchiques.

- **Cas 2D.** Dans la figure 4.6, chaque face est constituée de quatre arêtes hiérarchiques et donc de quatre brins hiérarchiques. Sur ce schéma, les brins orange correspondent aux brins hiérarchiques du modèle contrairement aux brins gris. La face F_0 est ainsi constituée de quatre arêtes hiérarchiques : a_{00} , a_{01} , a_{02} et a_{03} . Chacune de ces arêtes comprend un brin hiérarchique à F_0 , qui sont respectivement : a_{00} , a_{010} , a_{020} et a_{03} . L'arête hiérarchique a_{01} de F_0 , était initialement incidente à deux brins, un brin incident et hiérarchique à F_0 et un brin incident et hiérarchique à F_2 . Lors de la subdivision de F_2 , a_{01} a été subdivisée en deux sous-arêtes : a_{010} et a_{011} . La partie de a_{010} incidente à F_0 ,

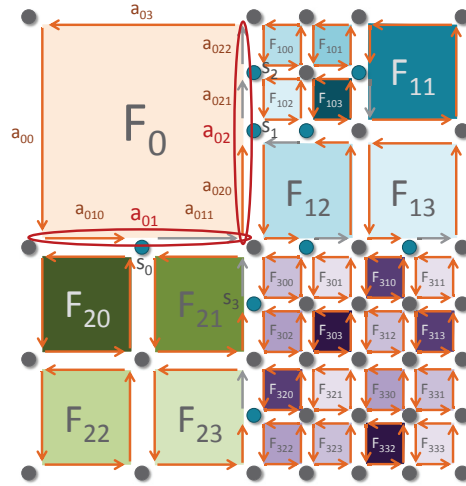


Figure 4.6 – La notion de brin hiérarchique pour un modèle 2D. Un modèle 2D initialement composé de quatre quadrilatères cousus ensemble par β_2 . Les brins orange sont des brins hiérarchiques contrairement aux brins gris. Les sommets bleus représentent les jonctions T frontalières.

notée $a_{010}|_{F_0}$, est restée hiérarchique et $a_{011}|_{F_0}$, ayant été créée lors de la subdivision, n'est pas hiérarchique pour F_0 . Par contre, $a_{011}|_{F_{21}}$, est quant à elle bien hiérarchique.

- **Cas 3D.** Dans la figure 4.4, les brins orange correspondent aux brins hiérarchiques de Vol et de ses faces hiérarchiques, contrairement aux brins gris. Vol possède vingt-quatre brins hiérarchiques, soit quatre brins hiérarchique par face hiérarchique. Nous constatons que les brins hiérarchiques et incidents à Vol , partent d'un sommet hiérarchique de Vol . De plus, ils se situent au début de chaque arête hiérarchique de Vol , relativement à une face hiérarchique de Vol .

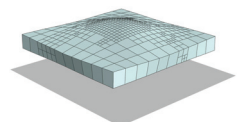
4.1.5 Jonction T

- Une **jonction T frontalière** consiste en un sommet situé à l'intérieur d'une arête hiérarchique d'un élément donné. Un élément est une face en 2D et une face hiérarchique ou un volume en 3D. Une jonction T frontalière n'est donc en aucun cas une extrémité d'une arête hiérarchique de l'élément en question. Chaque jonction T frontalière stocke dans l'attribut qui lui est associé, les deux extrémités de l'arête hiérarchique de plus faible niveau de subdivision dans laquelle elle est incluse.

La création d'une jonction T frontalière intervient pendant le processus de subdivision. En 2D (*resp.* 3D), si une arête hiérarchique est scindée (ou rescindée) et qu'elle est commune à deux faces (*resp.* au moins deux volumes) alors le nouveau sommet sera une jonction T frontalière.

Cas 2D. Dans la figure 4.6 les sommets bleus sont des jonctions T frontalières. Le sommet labellisé s_0 est une jonction T frontalière par rapport à la face F_0 . En effet, elle se situe à l'intérieur de son arête hiérarchique a_{01} . Par contre, s_0 n'est pas une jonction T frontalière pour la face F_{20} ni pour la face F_{21} . De la même manière s_3 est une jonction T frontalière pour la face F_{21} mais pas au regard des faces F_{300} et F_{302} .

Cas 3D. Dans la figure 4.4(b), les sommets bleus sont également les jonctions T frontalières relativement à Vol . Ces jonctions T frontalières ont été créées lors de la subdivision des volumes adjacents à Vol .



- Une **jonction T centrée** est seulement définie pour le cas 3D. Elle consiste en un sommet situé à l'intérieur d'une face hiérarchique, contrairement aux jonctions T frontalières qui sont situées sur les bordures des faces hiérarchiques d'un volume. Chaque jonction T centrée stocke en son attribut les quatre sommets hiérarchiques de la face hiérarchique de plus faible niveau de subdivision qui l'inclut.

Si une face hiérarchique est subdivisée (ou re-subdivisée) et qu'elle est commune à deux volumes, alors le nouveau sommet créé au milieu de la face sera une jonction T centrée pour le volume ayant le niveau de subdivision le plus faible. Nous reviendrons plus en détails sur le processus de subdivision amenant à la création de ces jonctions T dans les sections suivantes.

Dans la figure 4.4(b), les sommets en rose sont des jonctions T centrées relativement à Vol . Elles sont bien situées à l'intérieur des faces hiérarchiques de Vol . La face hiérarchique F_0 est constituée de quatre sommets hiérarchiques, huit jonctions T frontalières et de quatre jonctions T centrées. La face hiérarchique F_1 est composée de quatre sommets hiérarchiques, de douze jonctions T frontalières et de huit jonctions T centrées. Enfin, la face hiérarchique F_2 est constituée de quatre sommets hiérarchiques et de huit jonctions T frontalières.

- Une **jonction T bivalente** consiste en un sommet qui est à la fois une jonction T frontalière pour un volume et une jonction T centrée pour un autre volume. Une jonction T centrée est incidente à deux volumes, alors qu'une jonction T bivalente est incidente à trois volumes. Deux de ces trois volumes peuvent avoir un niveau de subdivision égal, par contre le troisième volume a un niveau de subdivision inférieur aux deux autres. Une jonction T bivalente est traitée de la même manière qu'une jonction T frontalière. Elle stocke ainsi dans l'attribut qui lui est associé, les deux extrémités de l'arête hiérarchique, incidente non pas au volume de plus faible niveau de subdivision, mais au volume de niveau de subdivision intermédiaire. Par exemple, dans la figure 4.7, les sommets bivalents sont indiqués par une flèche. La jonction T bivalente s_1 est frontalière pour Vol_{0011} et centrée pour Vol_1 . Comme $l_{Vol_1} < l_{Vol_{0011}} < l_{Vol_{0111}}$, s_1 stocke les deux extrémités de l'arête hiérarchique de Vol_{0011} . De plus, la jonction T bivalente s_2 est frontalière pour Vol_{0110} et centrée pour Vol_1 . Comme $l_{Vol_1} < l_{Vol_{0110}} < l_{Vol_{0111}}$, s_2 stocke les deux extrémités de l'arête hiérarchique de Vol_{0110} .

4.1.6 Éléments non conformes

Une face (*resp.* volume) est dite **non conforme** si elle possède au moins une jonction T. Une face (*resp.* volume) est dite **non conforme relativement à** une jonction T si un brin incident à la face (*resp.* volume) et partant de la jonction T en question, n'est pas un brin hiérarchique. Nous précisons qu'une face (*resp.* volume) non conforme relativement à une jonction T, voit tous les brins qui lui sont incidents et partant de cette jonction T comme étant non hiérarchiques.

- **Cas 2D.** Dans la figure 4.6, la face F_0 est non conforme relativement aux jonctions T frontalières s_0 , s_1 et s_2 . Nous observons que les brins de F_0 incidents aux jonctions T sus-mentionnées ne sont pas hiérarchiques. Nous prenons également l'exemple du quadrilatère F_{21} qui est non conforme relativement à la jonction T frontalière s_3 .
- **Cas 3D.** Dans la figure 4.4(b), Vol est non conforme relativement à tous les sommets bleus et roses. Tous les brins incidents à ces sommets et qui sont également incidents à Vol ne sont pas hiérarchiques (de couleur grise).

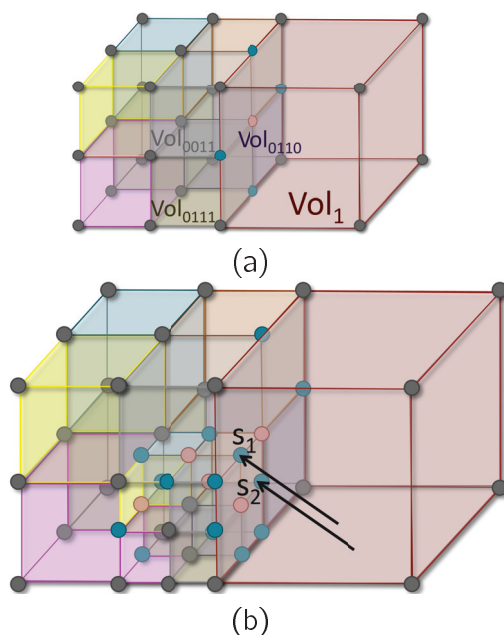


Figure 4.7 – La notion de jonction T bivalente. (a) Un modèle 3D composé de neuf hexaèdres dont Vol_{0111} et Vol_1 , cousus ensemble par β_3 . (b) Le modèle de (a) après la subdivision de Vol_{0111} . Les sommets bleus (resp. roses) correspondent aux jonctions T frontalières (resp. centrées). Les flèches pointent les deux jonctions T bivalentes du modèle. La jonction T bivalente s_1 est frontalière pour Vol_{0011} et centrée pour Vol_1 . La jonction T bivalente s_2 est frontalière pour Vol_{0110} et centrée pour Vol_1 .

4.2 Algorithmes de base

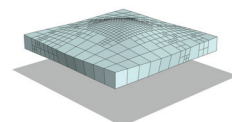
Nous présentons ci-dessous des algorithmes de base, utilisant les différents concepts explicités précédemment, afin de parcourir le bord d'une face (resp. face hiérarchique en 3D); de passer d'un brin hiérarchique d'une face (resp. face hiérarchique) au suivant; de récupérer un brin hiérarchique d'un élément donné et de calculer le niveau de subdivision d'un élément. Ces algorithmes nous seront utiles par la suite, pour expliquer notre méthode de subdivision d'un élément.

4.2.1 Récupérer le prochain brin hiérarchique

L'algorithme 3 permet de récupérer à partir d'un brin hiérarchique d'une face (resp. face hiérarchique), le prochain brin hiérarchique de cette même face (resp. face hiérarchique). Pour cela, il prend en paramètre un brin b_0 quelconque d'une face F dite simple. Puis, il parcourt, à partir de b_0 , la bordure de F en itérant sur les brins la composant grâce à β_1 . Enfin, il retourne le premier brin hiérarchique de F rencontré. Il permet ainsi de récupérer un des brins hiérarchiques d'une face simple ou encore de parcourir les différents côtés d'une face simple en allant d'un brin hiérarchique à un autre.

4.2.2 Tester si une face est simple

Il est très facile de tester si une face F est simple ou non, comme nous pouvons le voir dans l'algorithme 4. En effet, une face est simple si elle possède quatre sommets hiérarchiques. Cet algorithme procède pour cela de la manière suivante : il parcourt la bordure d'une face F à tester à partir d'un de ses brins incidents, tout en comptant le nombre de brins hiérarchiques



Algorithme 3 : Récupérer le prochain brin hiérarchique d'une face simple

Données : b_0 : un brin d'une face simple F .
Résultat : Le prochain brin b_1 hiérarchique après b_0 .
 $b_1 \leftarrow b_0$;
Répéter
 | $b_1 \leftarrow \beta_1(b_1)$;
jusqu'à b_1 est un brin hiérarchique;
Retourner b_1 ;

rencontrés. Si ce nombre vaut quatre, alors la face n'a pas été décomposée en sous-faces et est donc simple, c'est-à-dire que tous les brins de sa bordure sont accessibles par β_1 .

Algorithme 4 : Tester si une face est simple

Données : b : un brin d'une face F .
Résultat : Vrai si F est simple, faux sinon.
 $compteur \leftarrow 0$;
 $b_{debut} \leftarrow b$;
Répéter
 | **Si** b est un brin hiérarchique **Alors**
 | | $compteur \leftarrow compteur + 1$;
 | $b \leftarrow \beta_1(b)$;
jusqu'à $b \neq b_{debut}$;
Retourner $compteur = 4$;

4.2.3 Trouver le prochain brin hiérarchique d'une face

L'algorithme 5 a le même objectif que l'algorithme 3. Cependant, il est valable pour une face simple comme pour une face complexe. Par contre, dans le cas d'une face complexe, l'algorithme doit partir d'un brin appartenant à la bordure de cette dernière et non d'un brin quelconque comme c'est le cas pour une face simple. Nous détaillons ci-dessous les différentes étapes de l'algorithme en nous appuyant sur la figure 4.8. Cette dernière correspond à un zoom de la face hiérarchique F_0 de Vol présenté dans la figure 4.4(b). Les étapes sont les suivantes :

- Nous appelons l'algorithme 4, pour déterminer si F est simple ou non.
- Si F est simple, nous appliquons simplement l'algorithme 3.
- Sinon, F est une face complexe. Dans ce cas, la pré-condition oblige à ce que b_0 appartienne à la bordure de F . En effet, la procédure pour retrouver un brin hiérarchique de F à partir de n'importe quel brin appartenant à l'une de ses sous-faces, est plus difficile et plus coûteuse que la méthode présentée dans l'algorithme 3.
- Le parcours de la bordure d'une face complexe comme celle présentée dans la figure 4.8 nécessite l'emploi de β_1 mais également de β_{121} . En effet, considérons l'arête hiérarchique a de Vol , il nous faut alors parcourir toutes les sous-arêtes composant a depuis son brin hiérarchique b_0 . Chaque sous-arête correspond à une partie de l'arête hiérarchique d'une sous-face de la face hiérarchique. Pour parcourir l'arête hiérarchique d'une sous-face, nous employons β_1 jusqu'à tomber sur un brin dont le β_3 est marqué comme

Algorithme 5 : Le prochain brin hiérarchique d'une face

Données : b_0 : un brin d'une face F simple ou appartenant à la bordure d'une face F complexe.

Résultat : Le prochain brin b_1 hiérarchique de F après b_0 .

Si $est-simple(b_0)$ **Alors**

$b_1 \leftarrow \text{prochain-brin-hierarchique-face-simple}(b_0)$;

Sinon

 // Cas uniquement pour le 3D : face complexe

$b_1 \leftarrow b_0$;

Tant que $\beta_1(b_1)$ *n'est pas un brin hiérarchique* **faire**

Tant que $\beta_3(b_1)$ *n'est pas un brin hiérarchique* **faire**

$b_1 \leftarrow \beta_1(b_1)$;

Si $\beta_1(b_1)$ *est un brin hiérarchique* **Alors**

Retourner $\beta_1(b_1)$;

$b_1 \leftarrow \beta_{121}(b_1)$;

Retourner b_1 ;

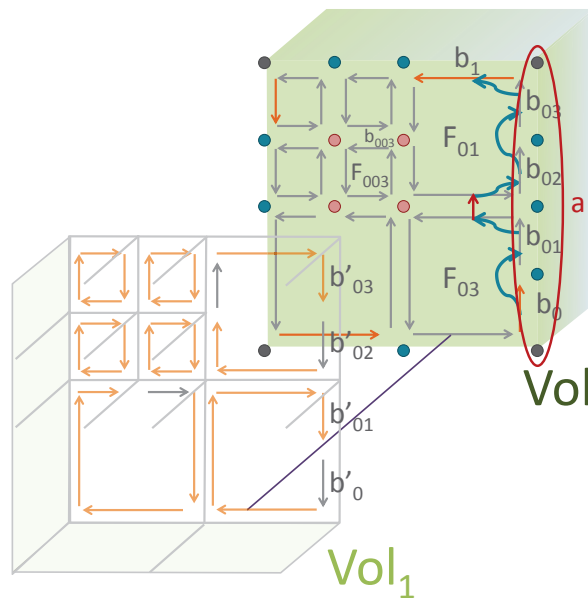
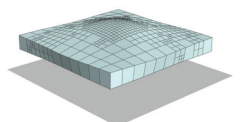


Figure 4.8 – Le parcours de la bordure de la face hiérarchique d'un volume étant plus subdivisée que ce dernier. Zoom sur la face hiérarchique F_0 de la figure 4.4(b). Par transparence, la partie de la face F_0 incidente au volume Vol_1 adjacent à Vol , est représentée. Tous les brins représentés sont liés deux à deux par β_3 . Les traits gris représentés au niveau de Vol_1 , délimitent les sept sous-volumes de Vol_1 adjacents à Vol . Le trait violet correspond à une relation β_3 , les autres n'étant pas représentées. Seule une partie de ces derniers est représentée.



hiérarchique. En effet, les sous-faces composant la face hiérarchique parcourue, sont hiérarchiques pour les volumes adjacents à Vol issus de Vol_1 . Les brins hiérarchiques aux sous-volumes de Vol_1 adjacents à Vol , permettent d'identifier les différents côtés des sous-faces de la face hiérarchique que nous parcourons. Enfin, pour passer d'une sous-arête d'une sous-face à une autre, nous utilisons β_{121} . En effet, ce parcours permet de rester sur l'arête hiérarchique parcourue.

Nous reprenons maintenant l'explication avec un exemple concret. La face hiérarchique F_0 de Vol présentée dans la figure 4.8 a été subdivisée. Supposons que l'algorithme prend en paramètre le brin hiérarchique b_0 de l'arête hiérarchique a de Vol . La deuxième partie de l'algorithme correspond au cas d'une face complexe. $b_{01} = \beta_1(b_0)$ n'est pas un brin hiérarchique de Vol . De plus, $b'_{01} = \beta_3(b_0)$ n'étant pas un brin hiérarchique du volume adjacent, nous nous déplaçons sur le brin suivant b_{01} par β_1 . Par contre, $b'_{01} = \beta_3(b_{01})$ est un brin hiérarchique du volume adjacent et comme $\beta_1(b_{01})$ n'est pas un brin hiérarchique de F , alors nous appliquons $\beta_{121}(b_{01})$, donnant b_{02} . Nous procédons de la même manière jusqu'au brin b_{03} . $b'_{03} = \beta_3(b_{03})$ est un brin hiérarchique du volume adjacent. Or, $b_1 = \beta_1(b_{03})$ est un brin hiérarchique de Vol . L'algorithme se termine donc en retournant le brin b_1 .

4.2.4 Calcul du niveau de subdivision d'un élément

Nous explicitons ci-dessous comment calculer le niveau de subdivision d'un élément. Nous rappelons que le niveau de subdivision d'un élément (face en 2D et volume en 3D) est égal à celui de ses arêtes hiérarchiques ainsi que de ses faces hiérarchiques en 3D.

Pour calculer le niveau de subdivision d'une face, deux possibilités s'offrent à nous. La première, décrite par l'algorithme 6, s'applique lorsque les brins sont correctement marqués comme hiérarchiques. Or, ce n'est pas le cas lors de certaines étapes du processus de subdivision. Nous employons alors la seconde méthode présentée par l'algorithme 7. En effet, la deuxième méthode ne s'appuie pas sur les brins hiérarchiques contrairement à la première.

Algorithme 6 : Calculer le niveau de subdivision d'une face simple lorsque ses brins hiérarchiques sont correctement marquées

Données : F : une face simple.

Résultat : Le niveau de subdivision I_F de F .

$b \leftarrow$ un brin incident à F ;

$I_F \leftarrow 0$;

Tant que b n'est pas un brin hiérarchique de F **faire**

$b \leftarrow \beta_1(b)$;

Répéter

$I_F \leftarrow I_F + \frac{1}{b}$;

$b \leftarrow \beta_1(b)$;

jusqu'à b est un brin hiérarchique;

Retourner $\frac{1}{I_F}$;

Les algorithmes 6 et 7 consistent à itérer, par β_1 , sur l'ensemble des brins constituant la bordure de la face simple F dont nous souhaitons déterminer le niveau de subdivision. Pour chaque brin parcouru, nous accumulons l'inverse de son niveau de subdivision. Nous rappelons

Algorithme 7 : Calculer le niveau de subdivision d'une face simple lorsque ses brins hiérarchiques ne sont pas correctement marqués

Données : F : une face simple.

Résultat : Le niveau de subdivision l_F de F .

$b \leftarrow$ un brin incident à F ;

$b_0 \leftarrow b$;

$l_F \leftarrow 0$;

Répéter

| $l_F \leftarrow l_F + \frac{1}{l_b}$;

| $b \leftarrow \beta_1(b)$;

jusqu'à $b_0 \neq b$;

$l_F \leftarrow \frac{l_F}{4}$ // 4 est le nombre d'arêtes hiérarchiques de F

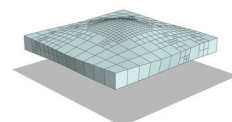
Retourner $\frac{1}{l_F}$;

que nos calculs relatifs au niveau de subdivision, s'effectuent exclusivement sur des formes fractionnaires.

- Pour l'algorithme 6, nous nous plaçons sur un brin b hiérarchique de F et nous itérons jusqu'à tomber sur le brin hiérarchique suivant. Nous obtenons ainsi le niveau de subdivision de l'arête hiérarchique ayant comme brin hiérarchique b , et donc par définition, nous obtenons également le niveau de subdivision de F .
- Pour l'algorithme 7, nous nous plaçons sur un brin b de F et nous itérons jusqu'à retomber sur ce même brin. Nous obtenons ainsi, en accumulant sur chaque brin parcouru, la somme des niveaux de subdivision des quatre arêtes hiérarchiques de F . Nous divisons donc le résultat obtenu par quatre pour avoir le niveau de subdivision d'une seule arête hiérarchique et donc de F .

Nous présentons maintenant la méthode de calcul du niveau de subdivision l_{Vol} d'un volume Vol . Nous rappelons que si une face hiérarchique de Vol a été subdivisée suite à la subdivision de volumes adjacents, alors elle correspond par définition à une face complexe. Le parcours des brins composant sa bordure nécessite donc un processus plus complexe et plus coûteux que celui permettant le parcours des brins composant la bordure d'une face simple qui emploie simplement l'opérateur β_1 . Les différentes étapes de cette méthode sont décrites par l'algorithme 8. Nous précisons que cet algorithme est générique à une face comme à un volume. Nous détaillons ci-dessous les étapes de la procédure en s'appuyant sur la figure 4.8 :

- Nous commençons par récupérer un brin hiérarchique b du volume Vol . Pour cela, nous appelons l'algorithme 9 sur un brin du volume dont nous souhaitons récupérer le niveau de subdivision.
- Si la face F , incidente à b , est simple, nous utilisons simplement la procédure présentée dans l'algorithme 6 pour obtenir l_F . Notons que si F est une face d'un volume, alors elle est hiérarchique et le niveau de subdivision obtenu correspond non seulement à son niveau de subdivision mais également à celui du volume dont elle est hiérarchique.
- Si F n'est pas simple, alors nous devons parcourir la bordure de la face hiérarchique F_h contenant la sous-face F , de la même manière que la procédure présentée dans l'algorithme 5. Pour cela, nous partons du brin hiérarchique b , obtenu dans l'étape précédente, et nous itérons sur chaque brin b' appartenant à l'arête hiérarchique incidente à b , tout en accumulant l'inverse de leur niveau de subdivision : $l_{b'}$. L'algorithme s'arrête



Algorithme 8 : Niveau de subdivision d'un élément**Données** : E : un élément.**Résultat** : Le niveau de subdivision l_E de E . $b \leftarrow$ un brin de E ; $b \leftarrow$ obtenir-brin-hierarchique-element(b); $b' \leftarrow b$; $F \leftarrow$ la face contenant b ;**Si** *est-simple*(b) **Alors** **Retourner** *niveau-subdivision-face-simple-version1*(F); // En 2D, $E = F$ **Sinon**

// Cas uniquement pour le 3D : face complexe

 $l_E \leftarrow 0$; **Tant que** $\beta_1(b')$ n'est pas un brin hiérarchique **faire** $l_E \leftarrow l_E + \frac{1}{l_{b'}}$; **Tant que** $\beta_3(b')$ n'est pas un brin hiérarchique **faire** $b' \leftarrow \beta_1(b')$; $l_E \leftarrow l_E + \frac{1}{l_{b'}}$; **Si** $\beta_1(b')$ est un brin hiérarchique **Alors** **Retourner** $\frac{1}{l_E}$; $b' \leftarrow \beta_{121}(b')$;**Retourner** $\frac{1}{l_E}$;

lorsque le brin parcouru est un brin hiérarchique de F_h . Nous obtenons ainsi le niveau de subdivision de cette arête hiérarchique de F_h et donc celui de l_{Vol} .

Nous illustrons cet algorithme par un exemple présenté dans la figure 4.8. Nous souhaitons calculer le niveau de subdivision de Vol . L'algorithme commence alors par récupérer un brin b incident à Vol . Nous prenons comme exemple, le brin b_{003} de Vol . La face F_{003} associée n'étant pas simple, nous appelons l'algorithme 9. Cet algorithme parcourt l'ensemble des brins de Vol et retourne par exemple le brin hiérarchique b_0 de Vol . F_{03} n'étant pas simple, nous passons directement à la deuxième partie de l'algorithme. $b_{01} = \beta_1(b_0)$ n'est pas un brin hiérarchique de Vol . Nous accumulons alors $\frac{1}{l_{b_0}} = \frac{1}{4}$ à l_{Vol} . Puis, $b'_{01} = \beta_3(b_0)$ n'étant pas un brin hiérarchique de Vol_1 , nous nous déplaçons sur le brin suivant : b_{01} par β_1 et nous accumulons $\frac{1}{l_{b_{01}}} = \frac{1}{4}$. Par contre, $b'_{01} = \beta_3(b_{01})$ est un brin hiérarchique de Vol_1 et comme $\beta_1(b_{01})$ n'est pas un brin hiérarchique de Vol , nous appliquons alors $\beta_{121}(b_{01})$, nous donnant b_{02} . Nous procédons comme ceci jusqu'au brin b_{03} . L'algorithme se termine ainsi après avoir accumulé $\frac{1}{l_{b_{03}}} = \frac{1}{4}$ et retourné la valeur de $l_{Vol} = 1$.

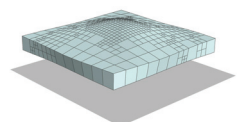
4.2.5 Trouver un brin hiérarchique d'un élément

Algorithme 9 : Obtenir un brin hiérarchique d'un élément

Données : b : un brin de l'élément E .
Résultat : Un brin hiérarchique de E .
 $b' \leftarrow \beta_1(b)$;
Si b' est un brin hiérarchique **Alors**
 | **Retourner** b' ;
Si *est-simple*(b) **Alors**
 | **Répéter**
 | | $b' \leftarrow \beta_1(b')$;
 | **jusqu'à** b' est un brin hiérarchique;
Sinon
 | **Pour tous** $b' \in c_3(b)$ **faire**
 | | **Si** b' est un brin hiérarchique de E **Alors**
 | | | **Retourner** b' ;
Retourner b' ;

L'algorithme 9 prend en paramètre un brin b quelconque d'un élément et retourne un brin hiérarchique de cet élément. Si b n'est pas un brin hiérarchique alors l'algorithme teste si la face F incidente à b est simple. Si c'est le cas, alors l'algorithme itère sur les brins de la bordure de F , jusqu'à tomber sur un brin hiérarchique. Par contre, si F est une face complexe, nous ne pouvons pas nous contenter d'itérer avec β_1 sur la bordure de F jusqu'à trouver un brin hiérarchique de Vol . Nous devons alors parcourir tous les brins de Vol à la recherche d'un brin hiérarchique. Notons que l'ensemble des brins d'un volume incident à b , est donné par l'orbite $\langle \beta_1, \beta_2 \rangle (b)$, que nous nommons $c_3(b)$ dans l'algorithme.

Nous considérons l'exemple de la figure 4.8. Nous souhaitons alors récupérer un brin hiérarchique de Vol à partir du brin b_{003} . La sous-face F_{003} de la face hiérarchique F_0 , incidente à b_{003} , ne possède pas de brin hiérarchique. L'algorithme doit alors parcourir tous les brins de Vol , jusqu'à trouver un brin hiérarchique. Enfin, notons que les sous-faces comme F_{01} ou F_{03} , contiennent dans leur bordure un brin hiérarchique de Vol . Une optimisation est alors



réalisée en testant si F est simple.

Cette partie a présenté les concepts et les algorithmes de base, nécessaires à la réalisation de notre méthode de subdivision. Dans les prochaines sections, nous présenterons, le processus de subdivision d'une face avec la mise à jour des informations physiques pour le cas 2D. Puis, nous détaillerons le processus de subdivision d'un volume avec la mise à jour des informations physiques pour le cas 3D. Enfin, nous expliciterons les différentes modifications apportées au processus de découpe pour qu'il puisse gérer des modèles subdivisés localement.

4.3 Subdivision régulière d'un quadrilatère en quatre sous-quadrilatères

Cette partie a pour objectif de présenter notre processus de subdivision pour, en 2D, une face, et en 3D, une face hiérarchique d'un volume donné. Nous expliciterons ce processus dans un premier temps sous l'angle topologique, puis dans un second temps sous l'angle physique.

4.3.1 Présentation intuitive

La figure 4.9 montre un modèle 2D initialement composé de deux quadrilatères adjacents F_0 et F_1 . Ces derniers ont le même niveau de subdivision égal à 1. Ils sont donc composés de quatre brins chacun. La figure illustre les différentes étapes de subdivision régulière du quadrilatère F_0 en quatre sous-quadrilatères. Le processus de subdivision s'appuie sur les opérations d'insertion des cartes combinatoires (cf. la partie 2.3.3). Il est composé de quatre étapes :

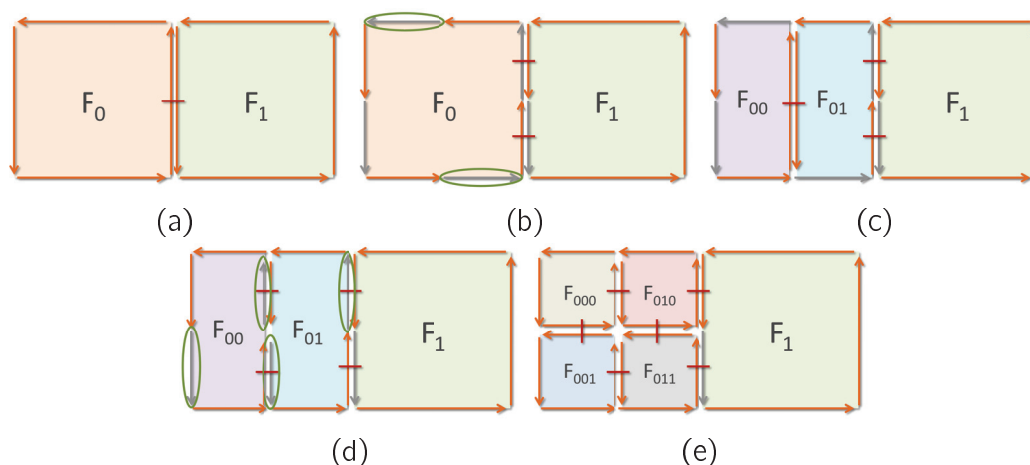


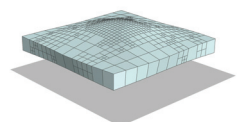
Figure 4.9 – Le processus de subdivision d'un quadrilatère. (a) Un modèle 2D composé de deux quadrilatères adjacents : F_0 et F_1 , cousus ensemble par β_2 . (b) Le modèle de (a) après l'insertion d'un sommet au milieu de chacune des quatre arêtes hiérarchiques de F_0 . (c) Le modèle de (b) après l'insertion d'une arête au milieu de la face F_0 . F_0 a été scindée en deux sous-faces : F_{00} et F_{01} . (d) Le modèle de (c) après l'insertion d'un sommet au milieu de la nouvelle arête. (e) Le modèle de (d) après l'insertion d'une arête au milieu de chacune des deux nouvelles sous-faces : F_{00} et F_{01} . A la fin du processus, le quadrilatère F_0 a été subdivisé en quatre sous-quadrilatères : F_{000} , F_{001} , F_{010} et F_{011} . Les liens rouges représentent les relations β_2 . Les brins orange correspondent aux brins hiérarchiques contrairement aux brins gris.

1. Un nouveau sommet est inséré au milieu de chacune des quatre arêtes hiérarchiques de F_0 . Cette étape consiste à insérer une 0-cellule dans chaque 1-cellule incidente à F_0 . La scission en deux parties d'un brin hiérarchique crée un deuxième brin. Le premier brin reste hiérarchique contrairement au deuxième qui n'est pas un brin hiérarchique de F_0 . Les deux brins résultants voient leur niveau de subdivision doublé par rapport au brin initial. Cette étape crée cinq nouveaux brins non hiérarchiques (représentés en gris dans la figure 4.9(b)). En effet, la scission de la 1-cellule commune à F_0 et F_1 génère deux nouveaux brins. Enfin, un nouvel attribut Particule est associé à chacune des quatre nouvelles 0-cellules ainsi qu'un nouvel attribut Ressort à chacune des quatre nouvelles 1-cellules. L'initialisation et la mise à jour des informations relatives au modèle physique sont explicitées dans la sous-section 4.3.3.
2. Une nouvelle 1-cellule est insérée au milieu de la 2-cellule F_0 entre les brins entourés de vert dans la figure 4.9(c). Ces derniers ont pour origine les sommets nouvellement créés. Ils sont situés au milieu de deux arêtes hiérarchiques opposées. L'insertion de la nouvelle 1-cellule crée deux nouveaux brins hiérarchiques ayant un niveau de subdivision égal à $2|_{F_0}$. F_0 est ainsi scindée en deux sous-2-cellules F_{00} et F_{01} . Un nouveau Ressort est associé à la nouvelle 1-cellule.
3. Un nouveau sommet est inséré au milieu de la nouvelle 1-cellule précédemment créée (cf. la figure 4.9(d)). Une nouvelle Particule est associée à ce nouveau sommet et un nouveau Ressort est associé à la nouvelle arête.
4. F_{00} et F_{01} , récemment créés, vont être également scindés en deux de la même manière que F_0 à l'étape 2. Deux 1-cellules vont être insérées dans ces deux dernières faces en utilisant deux brins opposés appartenant à la même 2-cellule (cf. les brins entourés de vert dans la figure 4.9(d)). Les deux nouvelles 1-cellules ont un niveau de subdivision initialisé à $2|_{F_0}$. L'ensemble des quatre brins les composant sont des brins hiérarchiques. F_0 a ainsi été subdivisée en quatre sous-2-cellules : F_{000} , F_{010} , F_{001} et F_{011} (cf. la figure 4.9(e)).

La figure 4.10 illustre la subdivision régulière d'un quadrilatère incident à deux hexaèdres, en quatre sous-quadrilatères. La figure 4.10(a) montre un modèle 3D composé de deux hexaèdres adjacents Vol_0 et Vol_1 . Ces derniers ont pour face commune F_0 où $F_{0|Vol_0}$ correspond à la partie incidente à Vol_0 et $F_{0|Vol_1}$ celle incidente à Vol_1 . Vol_0 et Vol_1 ont le même niveau de subdivision égal à 1. Ils sont alors composés de vingt-quatre brins chacun. La figure 4.10(b) représente le modèle de la figure 4.10(a) après la subdivision du quadrilatère F_0 en quatre sous-quadrilatères. La définition des cartes combinatoires implique que la subdivision de $F_{0|Vol_0}$ entraîne de manière identique celle de $F_{0|Vol_1}$. Le processus de subdivision employé pour subdiviser F_0 en quatre sous-quadrilatères est exactement le même que celui présenté ci-dessus pour le cas 2D.

4.3.2 Subdivision d'un quadrilatère inscrit dans un modèle localement raffiné

Cependant, le processus de subdivision qui vient d'être présenté implique que les éléments adjacents à l'élément E à subdiviser, aient un niveau de subdivision inférieur ou égal à E . Plus précisément, en 2D, les arêtes hiérarchiques de la face F à subdiviser n'ont pas été en amont subdivisées par les faces adjacentes à F . En 3D, les arêtes hiérarchiques de la face hiérarchique F du volume Vol considéré, ne doivent pas avoir été subdivisées en amont, par les autres faces hiérarchiques de Vol qui sont adjacentes à F , ni par les autres volumes



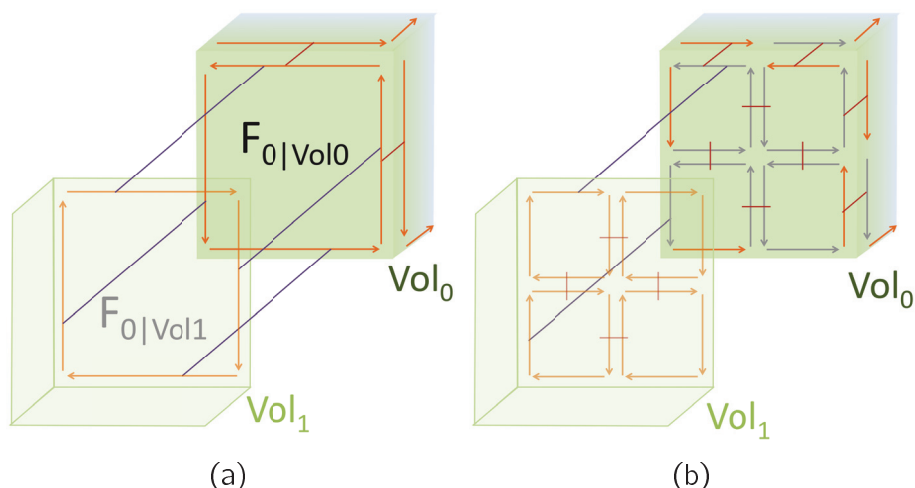


Figure 4.10 – La subdivision d'un quadrilatère incident à deux hexaèdres, en quatre sous-quadrilatères. (a) Un modèle 3D composé de deux hexaèdres adjacents : Vol_0 et Vol_1 , cousus ensemble par β_3 . F_0 est composée de $F_{0|Vol_0}$ et $F_{0|Vol_1}$. Par transparence, la partie de la face F_0 incidente à Vol_1 : $F_{0|Vol_1}$ est représentée. Tous les brins incidents à $F_{0|Vol_0}$ sont liés deux à deux avec les brins incidents à $F_{0|Vol_1}$ par β_3 . (b) Le modèle de (a) après l'application du processus permettant la subdivision régulière de F_0 en quatre sous-quadrilatères. Les liens rouges (resp. violets) représentent les relations β_2 (resp. β_3). Seule une partie des liaisons β_3 est représentée dans (b). Les brins orange correspondent aux brins hiérarchiques.

adjacents à Vol . De plus, les faces hiérarchiques d'un volume Vol à subdiviser, ne doivent pas avoir été subdivisées préalablement par les volumes adjacents à Vol .

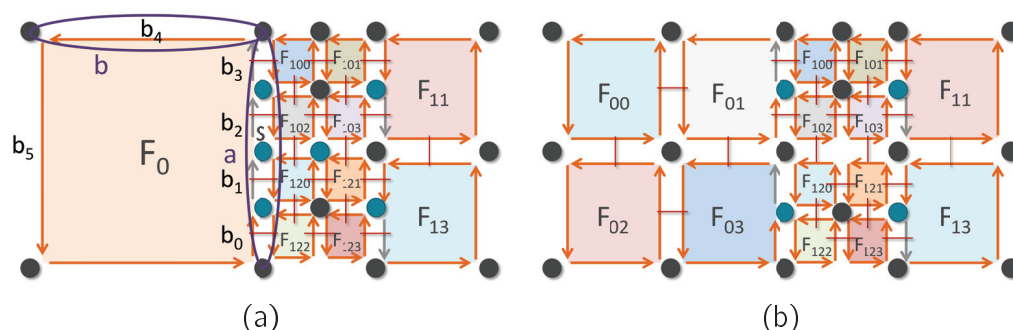


Figure 4.11 – La subdivision d'un quadrilatère adjacent à un quadrilatère ayant un niveau de subdivision plus grand. Un modèle 2D initialement composé de deux quadrilatères cousus ensemble par β_2 : F_0 et F_1 . (a) Le modèle initial obtenu après que F_1 puis F_{10} et F_{12} ont été subdivisés. (b) Le modèle de (a) après la subdivision de F_0 . Le schéma présente une subdivision régulière d'un quadrilatère (F_0) adjacent à un quadrilatère (F_1) ayant un niveau de subdivision plus important. Les sommets bleus correspondent à des jonctions T .

Dans ce sens, le processus de subdivision d'un élément E (face en 2D et face hiérarchique d'un volume en 3D) doit être modifié pour gérer le cas où l'élément E est moins subdivisé qu'un élément adjacent. L'exemple présenté dans la figure 4.11 illustre ce cas. La figure 4.11(a) présente un modèle 2D initialement composé de deux quadrilatères adjacents F_0 et F_1 , après la subdivision de F_1 en F_{1i} avec $i \in \{0, \dots, 3\}$, suivi de la subdivision de F_{10} en F_{10i} et celle de F_{12} en F_{12i} . La figure 4.11(b) présente le modèle de (a) après la subdivision de F_0 .

Nous énonçons ici les deux modifications à apporter au processus de subdivision pour

gérer le cas présenté ci-dessus :

- La première étape du processus de subdivision consiste en l'insertion de nouveaux sommets au milieu de chacune des arêtes hiérarchiques de l'élément E à subdiviser. Si une des arêtes hiérarchiques a de E s'avère avoir déjà été subdivisée lors de la subdivision d'un élément adjacent à E et incident à a , alors le sommet devant être inséré est déjà présent. Dans ce cas, aucun nouveau sommet ne devra être inséré dans a pendant cette étape du processus de subdivision.

La figure 4.11 illustre ce propos avec le sommet s correspondant au sommet qui ne sera pas inséré pendant le processus de subdivision de F_0 . En effet, l'arête hiérarchique a de F_0 a déjà été subdivisée lors de la subdivision de F_1 , ce qui a entraîné la création du sommet s scindant alors a en deux parties.

Ce constat implique de tester si un brin hiérarchique b doit être subdivisé ou non. Pour cela, nous testons si $\beta_1(b)$ est un brin hiérarchique. Si $\beta_1(b)$ n'est pas un brin hiérarchique, alors b doit être subdivisé, sinon le test indique qu'il a déjà été subdivisé. En effet, lorsqu'une arête hiérarchique est scindée, les nouveaux brins créés ne sont pas hiérarchiques. Ceci implique alors que si le β_1 d'un brin hiérarchique n'est pas hiérarchique, alors l'arête hiérarchique incidente a été scindée.

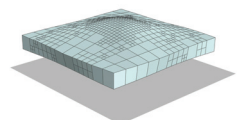
Sur la figure 4.11, a est composée de quatre brins. Le brin hiérarchique associé à a est b_0 . $b_1 = \beta_1(b_0)$ étant un brin hiérarchique, b_0 ne doit pas être subdivisé. Par contre, les trois autres arêtes hiérarchiques de F_0 sont scindées conformément au processus de subdivision détaillé ci-dessus. En effet, nous pouvons prendre l'exemple de l'arête hiérarchique b de F_0 . Comme $\beta_1(b_4)$ est un brin hiérarchique, alors b_4 sera subdivisé.

- Une deuxième modification du processus de subdivision intervient lorsque l'on scinde une face en deux, et plus précisément lorsque l'on récupère un brin partant du sommet situé au milieu d'une arête hiérarchique.

Dans les figures 4.9(b) et 4.9(d), ces brins correspondent aux brins entourés de vert. Pour les récupérer, il suffit de prendre le β_1 des brins hiérarchiques. Or, cette méthode n'est plus valable lorsque l'arête hiérarchique considérée a été scindée au moins deux fois en amont. Nous expliquons ici la méthode permettant de récupérer le brin partant du sommet situé au milieu d'une arête hiérarchique a ; ce brin est nommé b_{milieu} . Cette méthode s'appuie sur les niveaux de subdivision des sous-arêtes d'une arête hiérarchique, afin de récupérer le b_{milieu} de cette dernière. Il suffit alors d'accumuler les niveaux de subdivision, depuis le brin hiérarchique de cette arête hiérarchique, jusqu'à obtenir une valeur supérieure à la moitié du niveau de subdivision de l'arête hiérarchique, pour récupérer son b_{milieu} . Les différentes étapes de cette méthode sont décrites par l'algorithme 10 avec la figure 4.11.

1. Nous commençons par récupérer le niveau de subdivision de a grâce à l'algorithme 8. Nous prenons la moitié de l_a afin de considérer la moitié de l'arête hiérarchique, que nous stockons dans v . Dans la figure 4.11, nous considérons que $l_{F_0} = l_a = l_b = 1$, $l_{b_0} = l_{b_1} = l_{b_2} = l_{b_3} = 4$ et $l_{b_4} = 1$. L'arête hiérarchique considérée est a et b_0 est son brin hiérarchique. Enfin, $v = \frac{1}{2}$.
2. Comme F est une face simple, nous pouvons parcourir sa bordure avec β_1 . Nous accumulons alors l'inverse des niveaux de subdivision de chaque brin parcouru, depuis le brin hiérarchique b_0 de a , jusqu'à obtenir une valeur supérieure à la moitié de $\frac{1}{l_a}$.

Dans l'exemple, nous débutons avec b_0 en accumulant $\frac{1}{l_{b_0}} = \frac{1}{4}$ à *cumule*. Comme, $cumule \leq \frac{1}{2}$, nous passons au brin b_1 et nous accumulons $\frac{1}{l_{b_1}}$, donnant $cumule =$



Algorithme 10 : Récupérer le brin partant du sommet situé au milieu d'une arête hiérarchique donnée

Données : E : un élément.

a : une arête hiérarchique de E .

b_{debut} : le brin hiérarchique de a .

Résultat : Retourne b_{milieu} .

$l_a \leftarrow \text{niveau-subdivision-élément}(E)$;

// l_a : le niveau de subdivision de a , $l_a = l_E$

$v \leftarrow \frac{l_a}{2}$;

$b \leftarrow b_{debut}$;

$cumule \leftarrow 0$;

Tant que $cumule \leq \frac{1}{v}$ **faire**

$cumule \leftarrow cumule + \frac{1}{l_b}$;

$b \leftarrow \beta_1(b)$;

Retourner b ;

$\frac{1}{2}$. Comme, $cumule \leq \frac{1}{2}$, nous passons maintenant au brin b_2 et nous accumulons $\frac{1}{l_{b_2}}$, donnant alors $cumule = \frac{3}{4}$. $cumule$ étant alors supérieur à $\frac{1}{2}$, nous retournons b_2 . b_2 est bien le brin partant du sommet s situé au milieu de a .

Pour conclure cette partie, nous avons présenté de manière intuitive comment subdiviser topologiquement un quadrilatère quel que soit le niveau de subdivision de ses quadrilatères adjacents. L'algorithme complet permettant la subdivision d'un quadrilatère en quatre sous-quadrilatères, est présenté dans la section 7.1.

4.3.3 Initialisation et mise à jour des informations physiques

Lors du processus de subdivision d'un quadrilatère, l'insertion de nouveaux sommets et arêtes implique la création de nouvelles Particules et Ressorts. Ces Particules et Ressorts doivent ainsi être initialisés ou mis à jour pour conserver les propriétés physiques de l'objet. Or, malgré des similitudes dans le processus de mise à jour des informations physiques entre la subdivision d'une face, pour le cas 2D et la subdivision d'une face hiérarchique pour le cas 3D, nous préférons, dans un objectif de clarté de l'explication, détailler tout d'abord le cas 2D, puis expliciter le cas 3D en s'appuyant sur le cas 2D.

Dans cette partie, nous nommerons F , la face venant d'être subdivisée.

Masses

Les sommets créés pendant le processus de subdivision doivent, entre autre, initialiser leur masse. Cette dernière est stockée dans la Particule qui leur est associée. De plus, l'ensemble des sommets incidents à F , doivent également mettre à jour leur masse. En effet, leur masse était fonction de l'aire de F , alors que suite à la subdivision de cette dernière, elle est fonction non plus de l'aire de F mais de l'aire d'une sous-face de F . Dans la figure 4.13(b), les particules qui doivent initialiser leur masse sont rouges et les particules qui doivent mettre à jour leur masse sont vertes.

Pour initialiser ou mettre à jour la masse d'un sommet, nous procédons de la même manière que pour l'initialisation (cf. la sous-section 3.2.1). Nous rappelons que le calcul de la masse d'une particule est fonction des aires de ses faces incidentes. Or, pour calculer l'aire

d'une face F_a , dans un modèle localement raffiné, nous devons récupérer les longueurs initiales L_0 stockées au niveau des ressorts appartenant à deux arêtes hiérarchiques successives de F_a . Puis, nous divisons ces longueurs par le niveau de subdivision de F_a . De cette manière, nous considérons les dimensions d'une face, issue d'une subdivision, par rapport à celles de sa face mère initiale. Par exemple, dans la figure 4.12(a) où $l_{F_1} = 1$, nous posons les dimensions de F_1 comme étant : $longueur(F_1) = 2$ et $largeur(F_1) = 1$. Dans la figure 4.12(b), F_{11} , une des quatre sous-faces de F_1 , aura alors comme dimension : $longueur(F_{11}) = \frac{longueur(F_1)}{l_{F_{11}}} = \frac{2}{2} = 1$ et $largeur(F_{11}) = \frac{largeur(F_0)}{l_{F_{11}}} = \frac{1}{2}$.

Cas des jonctions T Une procédure particulière doit être mise en place pour le calcul de la masse des jonctions T. Nous rappelons tout d'abord que lors de la deuxième étape du processus de subdivision (cf. la sous-section 4.3), si un sommet est inséré dans une arête hiérarchique incidente à deux éléments, alors le sommet en question sera une jonction T (cf. la partie 4.3.3). La Particule associée stocke alors l'information de sa propriété de jonction T.

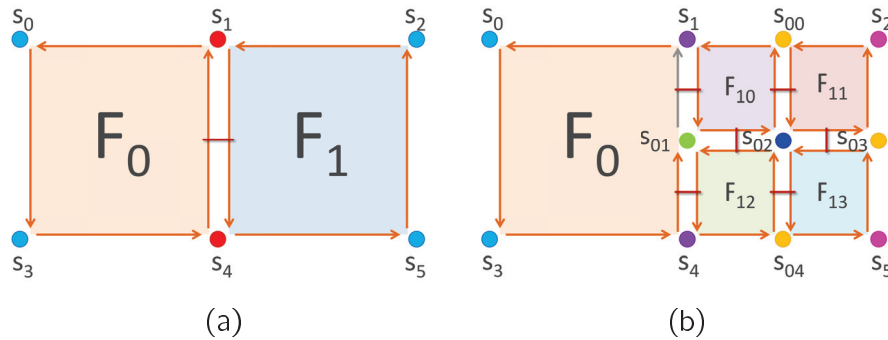
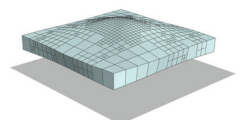


Figure 4.12 – La création d'une jonction T suite à la subdivision d'un quadrilatère qui avait initialement le même niveau de subdivision qu'un quadrilatère adjacent. (a) Un modèle 2D composé de deux quadrilatères adjacents F_0 et F_1 , cousus ensemble par β_2 . (b) Le modèle de (a) après la subdivision régulière de F_1 en quatre sous-quadrilatères. Cette dernière provoque la création de la jonction T s_{01} . Si l'on fait l'hypothèse que F_0 et F_1 ont la même densité surfacique alors toutes les particules de même couleur ont la même masse.

La figure 4.12 montre la création d'une jonction T suite à la subdivision d'un quadrilatère, qui avait le même niveau de subdivision qu'un quadrilatère adjacent. Le modèle présenté dans la figure 4.12(a) est composé de deux quadrilatères adjacents F_0 et F_1 . La figure 4.12(b) montre l'apparition de la jonction T s_{01} suite à la subdivision de F_1 en F_{1i} avec $i \in \{0, \dots, 3\}$. Ceci est dû au fait que ces sous-faces de F_1 ont alors un niveau de subdivision supérieur à F_0 . De plus, F_0 est maintenant constituée de cinq sommets : s_0, s_1, s_{01}, s_4 et s_3 , et est non conforme relativement à s_{01} .

La masse d'une jonction T est calculée en accumulant les contributions des éléments incidents qui lui sont conformes. Pour cela, nous parcourons l'ensemble des brins incidents à cette jonction T. Tous les brins hiérarchiques sont alors incidents à un élément qui contribuera au calcul de la masse de cette jonction T. En effet, nous rappelons que si un brin partant d'une jonction T est non hiérarchique, alors l'élément incident à ce brin est non conforme à cette jonction T. Dans figure 4.12(b), le brin incident à F_0 et partant de s_{01} , n'est pas un brin hiérarchique ; F_0 ne contribuera donc pas au calcul de la masse de s_{01} .

Exemple Nous faisons l'hypothèse que dans la figure 4.12, F_0 et F_1 ont la même densité surfacique. Les sommets de même couleur ont alors la même masse. Dans la figure 4.12(a),



s_0 et s_3 ont pour masse $\frac{\rho \cdot \mathcal{A}_{F_0}}{4}$. s_2 et s_5 ont pour masse $\frac{\rho \cdot \mathcal{A}_{F_1}}{4}$ et s_1 et s_4 ont pour masse $\frac{\rho \cdot \mathcal{A}_{F_0}}{4} + \frac{\rho \cdot \mathcal{A}_{F_1}}{4}$ (cf. la sous-section 3.2.1). Dans la figure 4.12(b), les masses de s_0 et s_3 restent inchangées. Par contre, s_2 (resp. s_5) a pour masse $\frac{\rho \cdot \mathcal{A}_{F_{11}}}{4}$ (resp. $\frac{\rho \cdot \mathcal{A}_{F_{13}}}{4}$) avec $\mathcal{A}_{F_{11}}$ (resp. $\mathcal{A}_{F_{13}}$) dépendant de la longueur et largeur de F_{11} (resp. F_{13}). Les dimensions de F_{11} (resp. F_{13}) correspondent aux dimensions de F_1 divisées par le niveau de subdivision de F_{11} (resp. F_{13}), soit 2 dans notre exemple. Nous obtenons ainsi $\mathcal{A}_{F_{11}} = \frac{\mathcal{A}_{F_1}}{4}$ (resp. $\mathcal{A}_{F_{13}} = \frac{\mathcal{A}_{F_1}}{4}$) et s_2 (resp. s_5) a ainsi pour masse $\frac{\rho \cdot \mathcal{A}_{F_1}}{16}$. De la même manière, s_{00} , s_{03} et s_{04} ont pour masse $2 \frac{\rho \cdot \mathcal{A}_{F_1}}{16}$, s_{02} a pour masse $4 \frac{\rho \cdot \mathcal{A}_{F_1}}{16}$. s_1 et s_4 ont pour masse $\frac{\rho \cdot \mathcal{A}_{F_0}}{4} + \frac{\rho \cdot \mathcal{A}_{F_1}}{16}$. Enfin, la jonction T s_{01} a pour masse $2 \frac{\rho \cdot \mathcal{A}_{F_1}}{16}$ (F_0 étant non conforme relativement à s_{01} , F_0 ne contribue pas au calcul de sa masse).

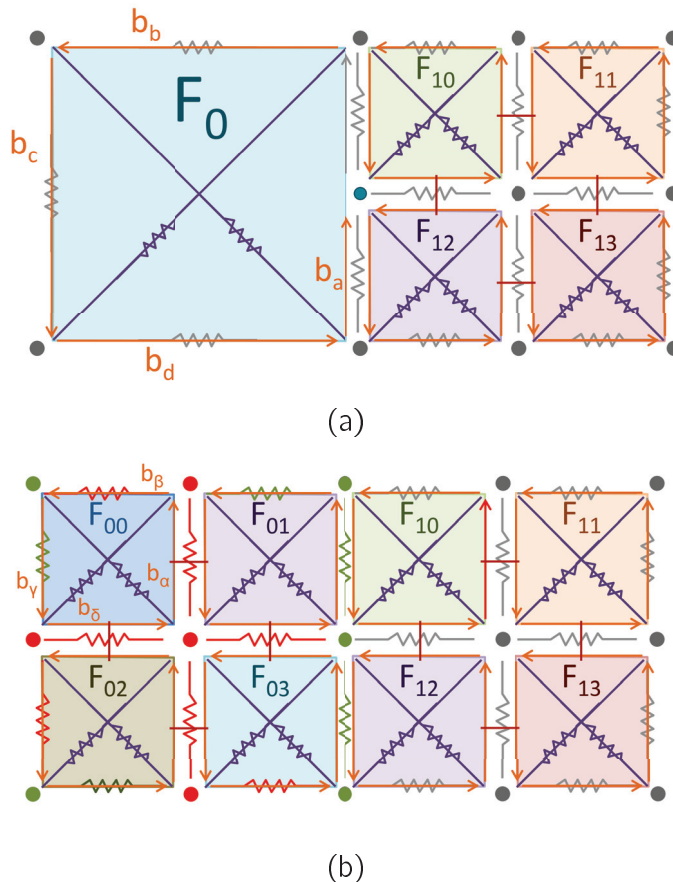


Figure 4.13 – La mise à jour des masses et des ressorts, suite au processus de subdivision d'un quadrilatère. (a) Un modèle 2D initialement composé de deux quadrilatères adjacents F_0 et F_1 , cousus ensemble par β_2 . Le quadrilatère F_1 a été subdivisé en quatre sous-quadrilatères : F_{10} , F_{11} , F_{12} et F_{13} . (b) Le modèle de (a) après la subdivision régulière du quadrilatère F_0 en quatre sous-quadrilatères : F_{00} , F_{01} , F_{02} et F_{03} . Les ressorts rouges sont les ressorts créés pendant le processus de subdivision, et doivent être initialisés. Les ressorts verts sont les ressorts qui doivent être mis à jour. Le même code couleur est appliqué aux particules.

Ressorts diagonaux

Les ressorts diagonaux internes à la face F sont supprimés avant sa subdivision. Puis, une fois le processus de subdivision réalisé, nous créons deux nouveaux ressorts diagonaux

pour les quatre sous-faces nouvellement créées.

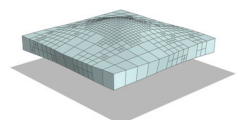
- **Suppression.** Pour supprimer un ressort diagonal, nous devons récupérer les deux sommets situés aux extrémités de ce dernier. La procédure à suivre sera expliquée avec la figure 4.13(a). Nous commençons par récupérer un brin hiérarchique b_a de F_0 . Ensuite, nous parcourons la bordure de F_0 à partir de b_a , pour récupérer les quatre brins hiérarchiques consécutifs : b_a , b_b , b_c et b_d . Les deux sommets associés aux brins b_a et b_c sont les extrémités du premier ressort diagonal de F_0 à supprimer, et les deux sommets associés aux brins b_b et b_d , les extrémités du deuxième ressort diagonal de F_0 à supprimer.
- **Création.** Afin de créer les ressorts diagonaux pour chaque nouvelle sous-face issue de la subdivision de F_0 , nous appliquons la procédure décrite précédemment, en récupérant les quatre brins hiérarchiques de la sous-face F_{00} : b_α , b_β , b_γ et b_δ , puis en créant les deux nouveaux ressorts diagonaux de cette sous-face entre les sommets des brins b_α et b_γ , et entre les sommets des brins b_β et b_δ (cf. la figure 4.13(b)). Notons que lors de la création des ressorts diagonaux, leurs **extrémités** sont initialisées avec les sommets ayant permis leur création. De plus, un ressort diagonal stocke un **brin de la face à laquelle il appartient**. Nous initialisons cette information lors de la création du ressort diagonal. Dans notre exemple, nous lui affectons le brin b_β . Par ailleurs, les Particules stockant l'ensemble des ressorts diagonaux auxquels elles sont reliées, nous rajoutons le nouveau ressort diagonal dans les **listes de ses Particules extrémités**. Enfin, la **raideur** des ressorts diagonaux doit être initialisée. Nous procédons pour cela, de la même manière que lors de l'initialisation du modèle (cf. la sous-section 3.2.2). La seule différence est que le calcul des dimensions de l'élément contenant le ressort diagonal en question, se fait comme expliqué précédemment pour la mise à jour des masses, c'est-à-dire en divisant les longueurs initiales stockées dans les ressorts appartenant à deux arêtes hiérarchiques successives de cet élément, par le niveau de subdivision de l'élément.

Nous précisons que la **longueur initiale** du ressort diagonal d inclus dans le quadrilatère F , nécessaire au calcul de sa force, est égale à $\sqrt{(X^2 + Y^2)}$ où $X = \frac{L_0(r)}{l_F}$ et $Y = \frac{L_0(s)}{l_F}$, avec r et s , les ressorts associés à deux arêtes hiérarchiques successives de F .

Ressorts non diagonaux

Les Ressorts créés lors du processus de subdivision doivent être initialisés. Les autres Ressorts qui sont également associés aux arêtes incidentes aux nouvelles sous-faces de F , doivent être mis à jour. Dans la figure 4.13(b), les ressorts qui doivent être initialisés sont rouges et les ressorts qui doivent être mis à jour sont verts.

Les Ressorts doivent initialiser ou mettre à jour leurs **extrémités** en récupérant celles d'un de leur brin incident. Ils doivent également initialiser ou mettre à jour leur **raideur**. Pour cela, nous procédons de la même manière que pour l'initialisation (cf. la sous-section 3.2.2), sauf pour récupérer les dimensions des éléments incidents qui sont récupérées au niveau des longueurs initiales des ressorts associés à deux arêtes hiérarchiques successives de l'élément en question, pondérées par le niveau de subdivision de l'élément (comme présenté pour le calcul de la masse et de la raideur des ressorts diagonaux). La raideur obtenue est relative à une arête hiérarchique d'une face F . Or, une arête hiérarchique pouvant être décomposée en sous-arêtes, nous pondérons la raideur obtenue par rapport au niveau de subdivision l_a de la sous-arête associée au ressort en question et l_F . La raideur $k_{F(i,j)}|_a$ d'un ressort associé



à l'arête a (de niveau de subdivision l_a) d'une face F (de niveau de subdivision l_F) est donc formulée de la manière suivante :

$$k_{F(i,j)|a} = k_F(i,j) \frac{l_a}{l_F}$$

où, $k_F(i,j)$ correspond à la raideur de l'arête hiérarchique contenant l'arête a . i et j sont les dimensions de la face F .

Enfin, de la même manière que pour l'initialisation, la raideur d'un ressort est déterminée en accumulant les raideurs calculées pour chaque face incidente.

Nous précisons que la **longueur initiale** du ressort r , nécessaire au calcul de sa force, est égale à $\frac{L_0(r)}{l_r}$.

Vitesses et accélérations

Les nouvelles Particules sont créées avec une vitesse et une accélération nulle. Pour éviter de freiner le système, nous initialisons la vitesse et l'accélération de ces dernières en fonction de leurs voisines. Plus précisément, lorsqu'un sommet est inséré dans une arête, nous récupérons la moyenne des vitesses (*resp.* accélérations) des deux extrémités de l'arête allant être scindée. Pour la particule créée au milieu de la face en cours de subdivision, nous calculons la moyenne des vitesses (*resp.* accélérations) des quatre sommets hiérarchiques de la face.

Contraintes

Les particules peuvent être contraintes selon l'axe X et/ou Y et/ou Z . Si la nouvelle particule est insérée au milieu d'une arête où ses deux extrémités étaient contraintes alors par défaut, elle sera elle-aussi contrainte de la même manière.

Jonction T

L'explication de cette partie s'appuie sur l'exemple présenté dans les figures 4.12 et 4.14. Elles montrent l'apparition de la jonction T s_{01} suite à la subdivision de F_1 en F_{1j} , avec $j \in \{0, \dots, 3\}$. De manière plus précise, elle est apparue lors de la première étape du processus de subdivision, consistant à scinder en deux parties chaque arête hiérarchique de F_1 . L'arête hiérarchique scindée par s_{01} était alors commune à deux éléments (F_0 et F_1) de même niveau de subdivision. Enfin, s_{01} se voit attribuer ses deux extrémités de projection correspondant aux extrémités de l'arête hiérarchique qu'elle scinde : s_1 et s_4 .

Pour gérer les jonctions T, nous avons fait le choix que ces dernières n'existent pas physiquement pour les éléments qui leur sont non conformes. Par exemple, la jonction T s_{01} n'existe pas physiquement pour F_0 . Nous présentons ci-dessous, les différents traitements appliqués aux jonctions T, en considérant la jonction T s_{01} de notre exemple :

- F_0 ne contribue pas au calcul de la masse de s_{01} (comme expliqué ci-dessus pour l'initialisation ou la mise à jour des masses).
- Comme le montre la figure 4.14, s_{01} n'étant « retenue » par aucun ressort dans le quadrilatère F_0 , elle « rentre » dans ce dernier.

Afin de corriger ce comportement erroné, à chaque itération de la boucle de simulation, s_{01} sera projetée orthogonalement sur le segment donné par ses extrémités, soit le segment $[s_1, s_4]$. Cette projection donne le point p . A chaque itération, nous calculerons ainsi p , puis nous l'affecterons à la position de s_{01} .

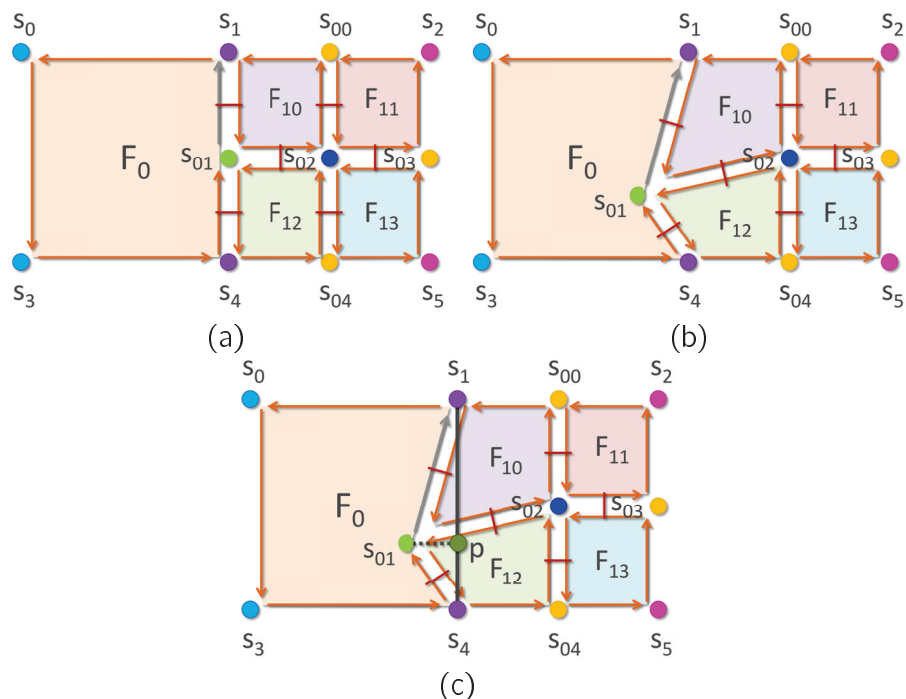


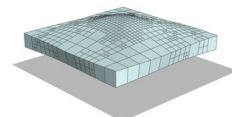
Figure 4.14 – Le comportement erroné induit par la création de jonctions T et la nécessité de contraindre la position de ses dernières, pour le cas 2D. (a) Le modèle de la figure 4.12(b). (b) s_{01} n'étant pas « retenue » par F_0 , elle « rentre » dans cette dernière. (c) La projection orthogonale p de s_{01} sur l'arête (s_1, s_4) . s_{01} sera contrainte sur p .

- La somme des forces appliquées à la jonction T s_{01} ne va pas être intégrée pour calculer sa nouvelle position puisque cette dernière va être directement affectée à celle de p . Or, s_{01} existant physiquement pour les éléments F_{10} et F_{12} , cette jonction T contribue à la somme des forces de ces éléments. Nous avons donc fait le choix de distribuer sa force au niveau des particules des éléments incidents, après l'accumulation des autres forces comme les forces internes dues aux ressorts ou encore la gravité.

Pour cela, nous parcourons tous les éléments incidents à une jonction T . Dans notre exemple, la jonction T s_{01} a comme éléments incidents : F_0 , F_{10} et F_{12} . Pour chaque brin b incident à la jonction T , si b est hiérarchique, alors la face F incidente à b est conforme relativement à cette jonction T . Dans ce cas, nous parcourons tous les sommets hiérarchiques de F et nous accumulons pour chacun d'entre eux, un sixième de la force de la jonction T en question. En effet, une jonction T est toujours incidente à trois quadrilatères et ces derniers ont trois particules chacun en plus de la jonction T , donnant un total de six particules. Le sixième de la force de s_{01} n'est pas directement accumulé à la force des sommets hiérarchiques de F . En effet, ces derniers peuvent également être des jonctions T pour un autre élément. Le sixième de la force de s_{01} est donc accumulé dans une autre variable stockée dans la Particule associée au sommet hiérarchique de F en question. Cette variable sera accumulée à la somme des forces du sommet hiérarchique considéré, à la fin de la distribution de la force de l'ensemble des jonctions T du modèle.

Dans notre exemple, nous accumulons un sixième de la force de s_{01} au niveau des sommets hiérarchiques de F_{10} : $\{s_1, s_{00}$ et $s_{02}\}$ ainsi que ceux de F_{12} : $\{s_{02}, s_{04}$ et $s_4\}$ reçoit ainsi deux sixièmes de la force de s_{01} .

Enfin, si le quadrilatère F_0 est amené à être subdivisé, alors s_{01} ne sera plus une jonction



T. Nous utilisons l'algorithme 10 depuis le brin hiérarchique de F_0 partant de s_4 pour retrouver le brin partant de s_{01} et donc la jonction T s_{04} . La vitesse (*resp.* accélération) de la jonction T a été initialisée à NULL. Or, lorsqu'elle perd son statut de jonction T, la particule a toujours une vitesse (*resp.* accélération) nulle. Pour éviter de freiner le système et d'induire ainsi une erreur dans le comportement, nous parcourons l'ensemble de ses extrémités de projection pour calculer la moyenne de leur vitesse (*resp.* accélération). Puis, nous affectons cette moyenne à la vitesse (*resp.* accélération) de la particule en question.

4.4 Subdivision régulière d'un hexaèdre en huit sous-hexaèdres

La partie précédente explicitait le processus de subdivision d'un quadrilatère en quatre sous-quadrilatères, avec tout d'abord son étape de subdivision topologique, puis sa mise à jour des informations physiques nécessaires à la simulation physique pour le cas 2D.

Nous présentons maintenant le processus de subdivision d'un hexaèdre en huit sous-hexaèdres. L'explication de ce processus s'appuiera sur la partie précédente. Nous procéderons d'ailleurs de la même manière que pour cette dernière, en présentant tout d'abord la partie topologique du processus, puis dans un deuxième temps sa partie physique.

4.4.1 Présentation intuitive

Dans cette partie, nous allons décrire les différentes étapes du processus de subdivision topologique d'un hexaèdre avec les figures 4.15 et 4.16. Ces figures illustrent les différentes étapes de subdivision d'un modèle 3D, initialement composé d'un seul hexaèdre que nous nommerons Vol (*cf.* les figures 4.15(a) et 4.16(a)). Ce dernier va ainsi être subdivisé régulièrement en huit sous-hexaèdres. Les différentes étapes sont présentées de manière schématique dans la figure 4.15, et de façon plus détaillées dans la figure 4.16 avec la représentation d'une partie des brins constituant le modèle. Initialement, la figure 4.16(a) représente douze des vingt-quatre brins de Vol . Nous observons que l'ensemble des brins incidents à Vol sont hiérarchiques à Vol . Enfin, nous notons l_{Vol} le niveau de subdivision de Vol . Le processus de subdivision s'appuie sur les opérations d'insertion des cartes combinatoires [BADSM08]. Ce processus est composé de six étapes, décrites ci-dessous :

1. Conformément au processus de subdivision d'un quadrilatère présenté dans l'algorithme 19, chacune des six faces hiérarchiques de Vol est subdivisée en quatre sous-faces. Le niveau de subdivision de l'ensemble des arêtes incidentes à Vol , prend alors comme valeur le double de l_{Vol} . Les points rouges sur les figures 4.15(b), 4.16(b) et 4.15(c) correspondent aux nouveaux sommets créés pendant cette étape.

Notons que lorsque la première face hiérarchique F_0 est subdivisée en F_{0i} avec $i \in \{0, \dots, 3\}$, ses quatre arêtes hiérarchiques sont scindées en deux (*cf.* la première étape du processus explicitée dans la sous-section 4.3). Ceci implique que les quatre faces hiérarchiques adjacentes à F_0 n'auront pas à subdiviser l'arête hiérarchique qu'elles ont en commun avec F_0 (*cf.* 4.16(b) et 4.16(c)).

2. Nous insérons ensuite une face F_a au milieu de Vol , coupant de manière transversale ce dernier en deux hexaèdres : Vol_0 et Vol_1 . Puis, nous marquons hiérarchiquement les brins de F_a correspondant au début d'un de ses côtés. F_a est représentée en rouge dans la figure 4.15(c). Elle interface les deux nouveaux volumes.

De manière plus précise, F_a est insérée selon une liste de brins \mathcal{L} formant un cycle au milieu des quatre faces hiérarchiques latérales de Vol (*cf.* les brins en pointillés de la figure 4.16(c)). Pour construire \mathcal{L} , nous commençons par récupérer un brin b qui est

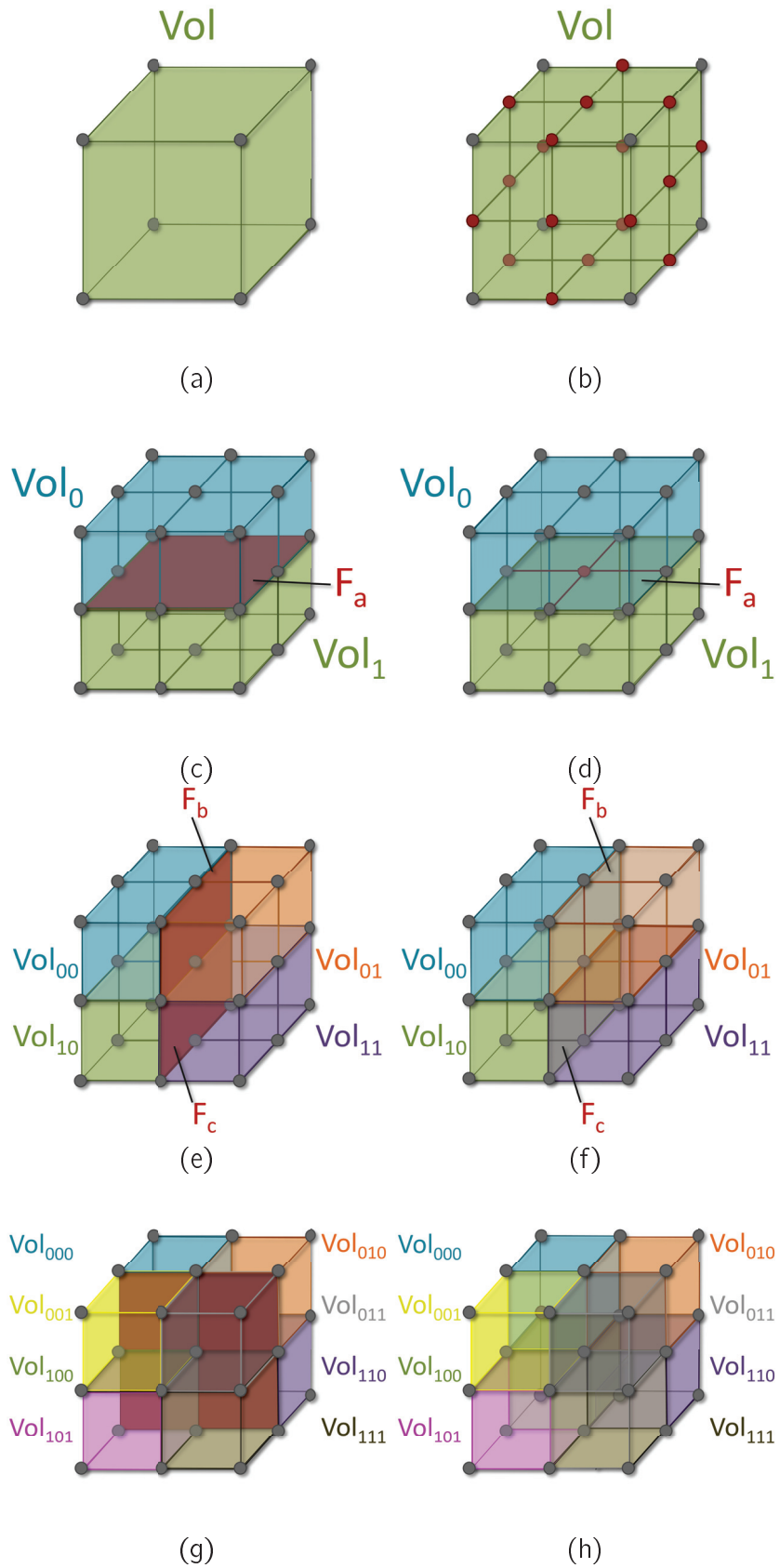
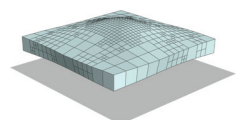


Figure 4.15 – Le processus de subdivision d'un hexaédre. (a) Un modèle 3D composé d'un seul hexaédre Vol. Le schéma montre les différentes étapes du processus permettant la subdivision régulière de Vol en huit sous-hexaédres.



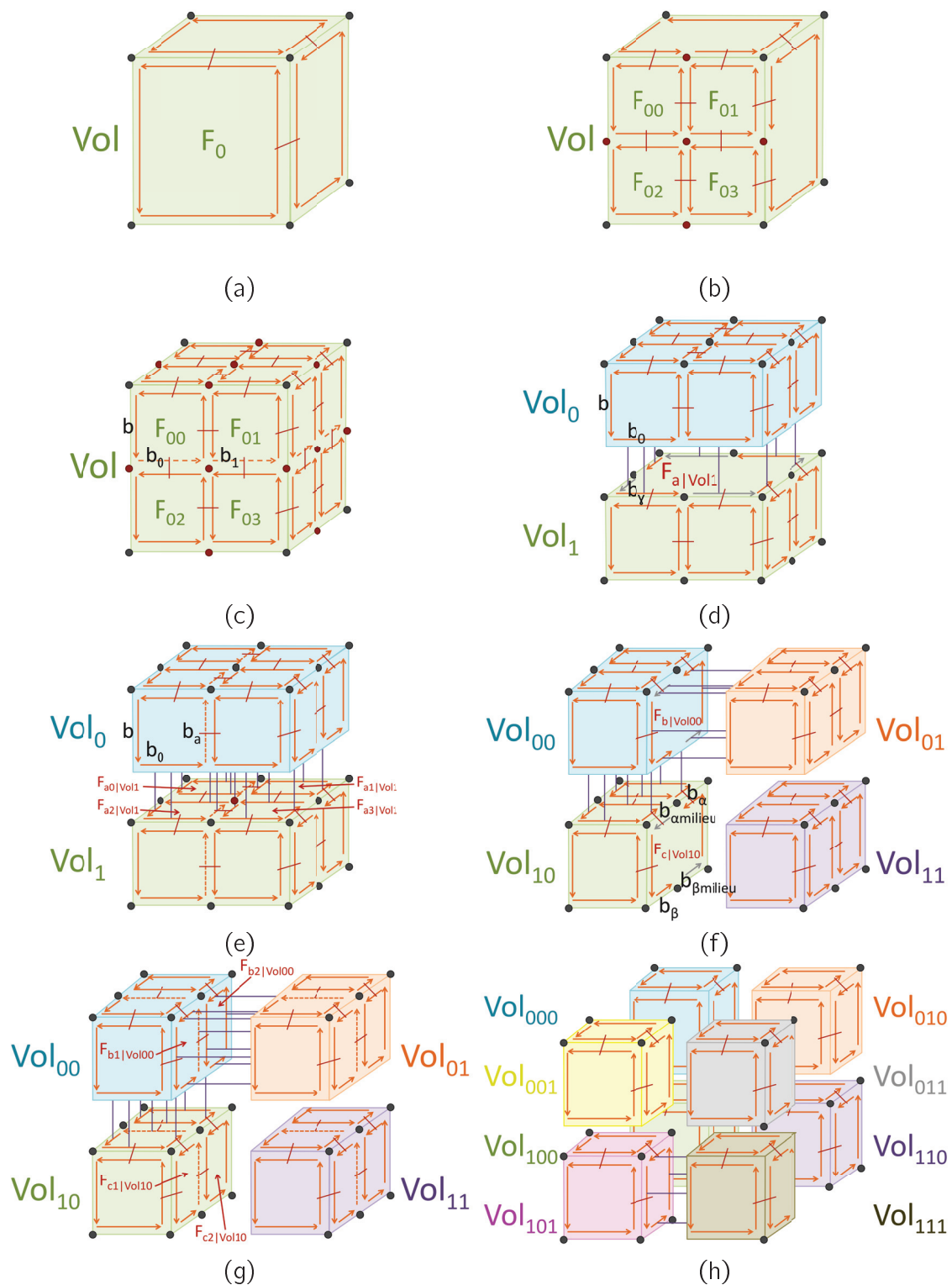


Figure 4.16 – Le processus de subdivision d'un volume avec la représentation d'une partie des brins. Tout comme la figure 4.15, ce schéma décrit les différentes étapes du processus de subdivision d'un hexaèdre Vol, mais de manière plus détaillée en représentant une partie des brins composant le modèle. Les brins orange correspondent aux brins hiérarchiques de Vol. Les liens rouges (resp. violets) correspondent aux liaisons β_2 (resp. β_3 , seulement partiellement représentées).

hiérarchique à la version initiale de Vol . Nous prenons ensuite le β_1 de b , donnant b_0 . b_0 correspond alors au premier brin de \mathcal{L} . Puis, nous appliquons l'opérateur b_{121} sur b_0 , donnant b_1 . Nous stockons b_1 dans \mathcal{L} . Nous appliquons ainsi b_{121} tout en stockant les brins parcourus, jusqu'à retrouver b_0 . Notons que la procédure est plus complexe si les faces hiérarchiques de Vol ont été subdivisées plusieurs fois par des volumes adjacents à Vol . Nous reviendrons sur cette remarque un peu plus loin dans le manuscrit.

F_a est constituée de deux cycles de brins, dont l'un est incident à Vol_0 et l'autre à Vol_1 : $F_{a|Vol_0}$ et $F_{a|Vol_1}$. Ces deux cycles de brins sont reliés deux à deux par β_3 (cf. 4.16(d)). Ils consistent respectivement en une face hiérarchique de Vol_0 et de Vol_1 . Nous devons marquer comme hiérarchique les quatre brins de ces deux faces hiérarchiques.

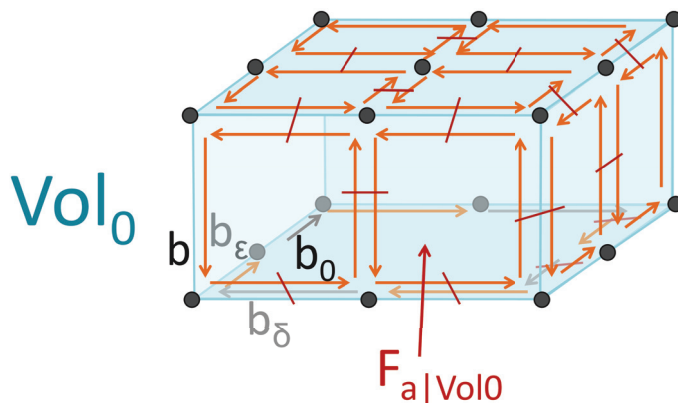


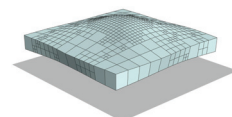
Figure 4.17 – Le marquage hiérarchique d'une face simple. Un zoom sur le sous-volume Vol_0 de la figure 4.16(d), où les brins de $F_{a|Vol_0}$ sont représentés.

Nous détaillons ici la procédure permettant de marquer comme hiérarchique les quatre brins de $F_{a|Vol_0}$, en considérant la figure 4.16(d). Ces quatre brins correspondent chacun au premier brin d'un des côtés de $F_{a|Vol_0}$. Pour les marquer, nous commençons par récupérer le premier brin inséré dans \mathcal{L} : b_0 . Nous appliquons β_2 , pour récupérer un brin incident à $F_{a|Vol_0}$: b_δ (cf. la figure 4.17). Notons que le brin b_δ se situe à la fin de l'arête hiérarchique de $F_{a|Vol_0}$, nous appliquons alors β_1 pour obtenir le premier brin b_ϵ incident à $F_{a|Vol_0}$ que nous marquerons comme hiérarchique. Il nous faut maintenant récupérer le prochain brin $\beta_{11}(b_\epsilon)$ à marquer hiérarchiquement. Enfin, nous procédons de la même manière pour les deux derniers brins.

Pour récupérer le premier brin hiérarchique b_γ de $F_{a|Vol_1}$ (cf. la figure 4.16(d)), nous avons avant l'insertion de F_a , stocké le β_2 de b_0 . Il suffit maintenant de prendre le β_2 de ce brin pour obtenir le premier brin hiérarchique de $F_{a|Vol_1}$. Pour marquer hiérarchiquement $F_{a|Vol_1}$, nous procédons de la même manière que pour $F_{a|Vol_0}$.

3. L'étape suivante consiste à scinder F_a en quatre sous-faces, en suivant encore une fois le processus de subdivision décrit dans l'algorithme 19. Le nouveau sommet inséré est représenté en rouge dans les figures 4.15(d) et 4.16(e).
4. Puis, une nouvelle face F_b (resp. F_c) est insérée dans le nouvel hexaèdre Vol_0 (resp. Vol_1). Elle coupe de manière sagittale l'hexaèdre Vol_0 (resp. Vol_1) en deux sous-hexaèdres : Vol_{00} et Vol_{01} (resp. Vol_{10} et Vol_{11}). Les deux nouvelles faces insérées sont représentées en rouge dans la figure 4.15(e).

Le processus employé est similaire à celui décrit dans la deuxième étape. En effet, nous commençons par décrire un cycle de brins qui permettra l'insertion de la nouvelle face. Notons que le premier brin b_a inséré dans le cycle \mathcal{L}_0 qui scindera Vol_0 , correspond au



β_1 de b_0 (cf. la figure 4.16(e)). Nous précisons ceci afin de montrer l'importance de se positionner sur le bon brin avant chacune des étapes.

Suite à l'insertion de la face F_{bc} (correspondant indifféremment à F_b ou F_c), nous marquons hiérarchiquement les brins composant chacune des deux parties de F_{bc} : $F_{b|Vol_{00}}$, $F_{b|Vol_{01}}$, $F_{c|Vol_{10}}$ et $F_{b|Vol_{11}}$ (cf. la figure 4.16(f)). Cette étape diffère quelque peu de la procédure précédente. En effet, F_{bc} étant insérée dans la moitié supérieure ou inférieure de l'hexaèdre initial Vol , sa longueur et sa largeur ne sont pas constituées du même nombre de brins. En effet, F_a était constituée de deux brins par côté. Or, les faces F_{bc} considérées ici, ont bien deux brins dans le sens de la longueur mais seulement un dans le sens de la largeur. C'est pour cette raison que nous appliquons β_{11} pour parcourir sa longueur, et que nous appliquons β_1 pour parcourir sa largeur.

5. F_b et F_c sont ensuite scindées en deux dans le sens de la longueur. Nous pouvons voir dans la figure 4.15(f), les deux arêtes verticales créées au milieu de F_b et F_c représentées en rouge.

Pour réaliser cette étape, nous commençons par nous placer sur un des deux brins hiérarchiques b_α de F_{bc} correspondant à une longueur (cf. la figure 4.16(f)). Puis nous récupérons le brin $b_{\alpha milieu}$ au milieu de cette longueur en appliquant $\beta_1(b_\alpha)$. Nous nous plaçons ensuite sur le deuxième brin b_β hiérarchique de F_{bc} après b_α . b_β correspond au brin hiérarchique de la longueur opposée à celle du brin b_α . Nous récupérons alors de la même manière que pour la première longueur, le brin $b_{\beta milieu}$ ($b_{\beta milieu} = \beta_1(b_\beta)$). $b_{\alpha milieu}$ et $b_{\beta milieu}$ permettent ainsi d'insérer l'arête qui scindera F_{bc} en deux. Le niveau de subdivision des deux nouvelles arêtes prend comme valeur : $2 l_{Vol}$.

Pour finir, nous marquons comme hiérarchique les quatre brins incidents à la nouvelle arête insérée dans F_b ainsi que pour chacun d'entre eux, leur β_1 . Nous procédons de la même manière pour les quatre brins incidents à la nouvelle arête de F_c (cf. la figure 4.16(g)).

6. Enfin, une nouvelle face est insérée au milieu de chacun des quatre nouveaux hexaèdres Vol_{00} , Vol_{01} , Vol_{10} et Vol_{11} . Elle coupe de manière coronale chacun de ces hexaèdres en deux sous-hexaèdres. Vol_n est ainsi scindé en Vol_{n0} et Vol_{n1} avec $n \in \{00, 01, 10, 11\}$. Les nouvelles faces sont représentées en rouge dans la figure 4.15(g).

Nous procédons de la même manière que pour les étapes (3) et (5) en créant un cycle de brins permettant d'insérer la face en question, puis de marquer hiérarchiquement tous les brins qui lui sont incidents (cf. la figure 4.16(g)).

7. Les figures 4.15(h) et 4.16(h) correspondent au résultat obtenu après le processus de subdivision. Vol est alors divisé en huit sous-hexaèdres : Vol_{000} , Vol_{001} , Vol_{010} , Vol_{011} , Vol_{100} , Vol_{101} , Vol_{110} et Vol_{111} .

4.4.2 Subdivision d'un hexaèdre inscrit dans un modèle localement raffiné

La présentation du processus de subdivision d'un volume qui vient d'être faite, est valable seulement si tous les volumes adjacents au volume Vol à subdiviser ont un niveau de subdivision inférieur ou égal à Vol . Plus précisément, les arêtes et faces hiérarchiques de Vol ne doivent pas avoir été subdivisées par des volumes qui lui sont adjacents. Afin de s'affranchir de cette limitation, le processus de subdivision d'un volume doit être modifié en conséquence.

La figure 4.18 représente une partie d'un modèle 3D constitué de plusieurs hexaèdres adjacents, centrée sur l'hexaèdre $Vol+$. $Vol+$ contient des arêtes et faces hiérarchiques subdivisées par des volumes adjacents. Nous allons présenter les différentes modifications à

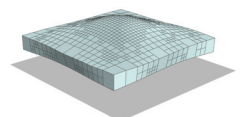
apporter au processus de subdivision pour qu'il puisse permettre la subdivision d'un volume, inscrit dans un modèle 3D localement raffiné, rassemblant les mêmes contraintes que $Vol+$, de niveau de subdivision l_{Vol+} . Notons que contrairement à Vol , les brins incidents à $Vol+$ ne sont pas tous hiérarchiques. Nous reprenons ici les différentes étapes du processus présenté ci-dessus afin de détailler les différentes modifications qui ont été apportées à ce dernier.

1. Nous rappelons que la première étape consiste à subdiviser les six faces hiérarchiques de $Vol+$. Afin de réaliser cette première étape, nous récupérons en amont l'ensemble des faces \mathcal{F} incidentes à $Vol+$. Nous précisons que \mathcal{F} comprend l'ensemble des faces hiérarchiques de $Vol+$ si elles n'ont pas été subdivisées, sinon à l'ensemble des sous-faces de chaque face hiérarchique subdivisée. Pour cela, nous devons tester si nous pouvons appliquer le processus de subdivision sur la face $F \in \mathcal{F}$. Nous listons ci-après les deux raisons empêchant F de suivre le processus de subdivision :
 - Si F ne possède pas de brin hiérarchique de $Vol+$ (cf. la sous-face F_{003} de la figure 4.18(a)), alors nous passons à une autre face de \mathcal{F} .
 - Sinon, nous testons si F a déjà été subdivisée ou non. Pour cela, nous récupérons un brin b incident à F et hiérarchique de $Vol+$, et nous parcourons la bordure de F avec β_1 à partir de b . Si nous ne trouvons pas quatre brins hiérarchiques de $Vol+$, alors ceci implique que F a déjà été subdivisée (c'est le cas de $F_{0|Vol+}$ dans la figure 4.18(a)). Nous stockons alors F dans \mathcal{F}_∞ pour marquer hiérarchiquement ses brins et nous passons à une autre face de $Vol+$.

Si F possède quatre brins hiérarchiques de $Vol+$. F est une face hiérarchique de $Vol+$ qui n'a pas été préalablement subdivisée en quatre sous-faces. Dans la figure 4.18(a), la face F_1 fait partie des faces hiérarchiques de $Vol+$ devant être subdivisées. Nous procédons à la subdivision de F en quatre sous-faces. La première étape du processus de subdivision de F (cf. l'algorithme 19) consiste à scinder ses quatre arêtes hiérarchiques. Or, les arêtes hiérarchiques de F ont pu être subdivisées par des volumes ou des faces incidentes. Pour chaque arête hiérarchique de F , nous devons tester si elle doit être scindée. Pour cela, une méthode consiste à vérifier que le niveau de subdivision du brin hiérarchique de l'arête hiérarchique à tester est égal à l_{Vol} . Si c'est le cas, l'arête hiérarchique doit être scindée, sinon une autre arête hiérarchique de F doit être vérifiée. Le processus de subdivision de F se poursuit ensuite de la même manière que celui présenté dans l'algorithme 19.

Si $F_n \in \mathcal{F}_\infty$. F_n est une sous-face d'une face hiérarchique F de $Vol+$, ayant été préalablement subdivisée. Nous devons alors marquer hiérarchiquement F . La figure 4.18 illustre le processus marquant une face hiérarchique, décrit par l'algorithme 11.

Exemple La figure 4.18 nous permet d'expliciter les différentes étapes aboutissant à la subdivision et au marquage hiérarchique des six faces hiérarchiques de $Vol+$. Dans la figure 4.18(a), la face hiérarchique F_0 de $Vol+$, a été préalablement subdivisée au cours d'une première subdivision de Vol_1 , suivie d'une seconde de deux des sous-volumes de Vol_1 adjacents à $Vol+$. La figure 4.18(b) représente le modèle de (a) après l'étape de subdivision des faces hiérarchiques de $Vol+$ n'ayant pas été préalablement subdivisées. Nous faisons l'hypothèse que seules les faces hiérarchiques F_1 et F_2 de $Vol+$ sont concernées. La figure 4.18(c) correspond au modèle de (b) suite au processus de marquage hiérarchique de la face F_0 , ayant été préalablement subdivisée. Enfin, la figure 4.18(d) présente le modèle de (c) après que toutes les faces hiérarchiques de $Vol+$ aient été subdivisées ou marquées hiérarchiquement. Notons que les différences



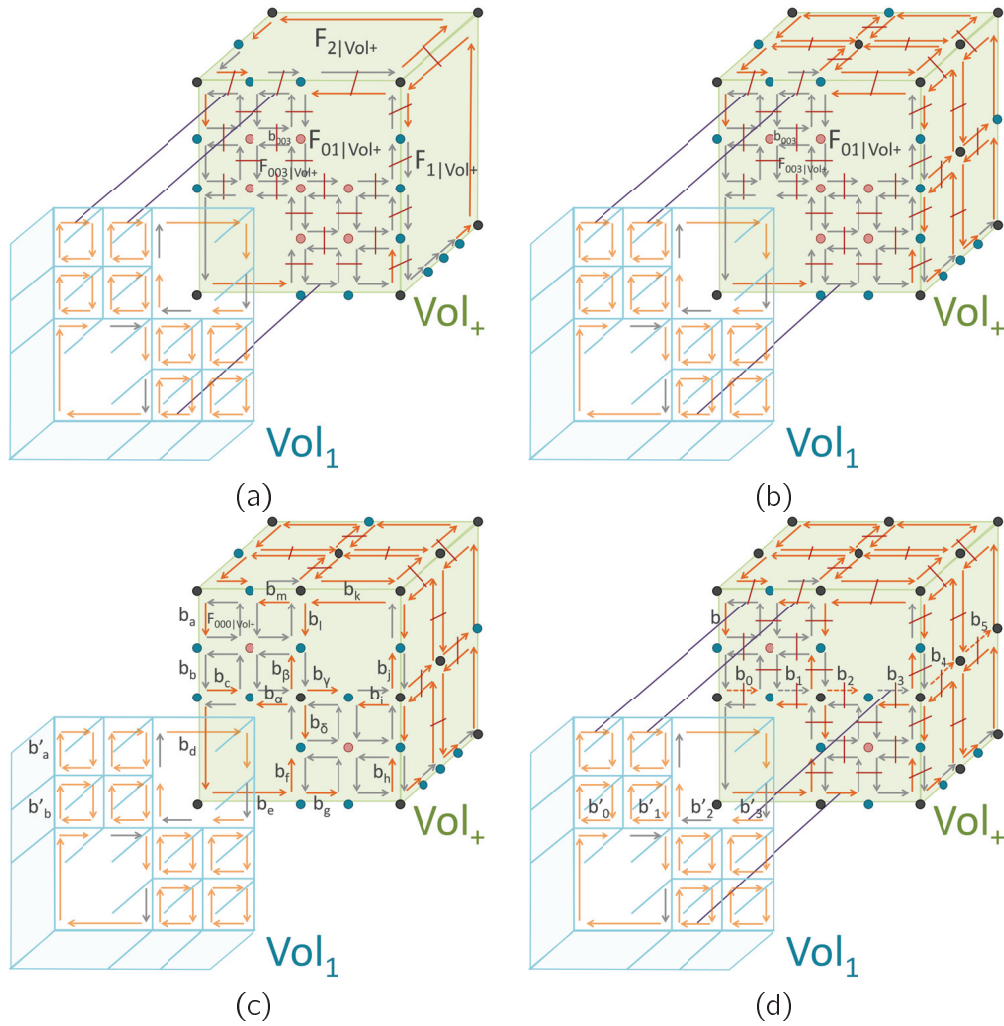
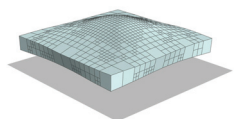


Figure 4.18 – Les premières étapes du processus de subdivision d'un volume, présentant des arêtes et des faces hiérarchiques subdivisées en amont par des volumes adjacents. (a) Une partie d'un modèle 3D composé de plusieurs hexaèdres adjacents, cousus ensemble par β_3 . Vol_1 a été subdivisé une première fois, puis deux de ses sous-volumes adjacents à Vol_+ ont également été subdivisés. La figure représente plus particulièrement l'hexaèdre Vol_+ ainsi que par transparence la partie de l'hexaèdre Vol_1 , adjacente à Vol_+ . (b) Le modèle de (a) après la subdivision des faces hiérarchiques de Vol_+ : F_1 et F_2 . Nous faisons l'hypothèse dans cet exemple, que seules ces deux faces n'ont pas été préalablement subdivisées par un volume adjacent. (c) Le modèle de (b) après avoir marqué les brins hiérarchiques de Vol_+ , incidents à $F_{01|Vol_+}$. Pour des raisons de lisibilité, les liaisons β_2 relatives aux brins de $F_{01|Vol_+}$ ne sont pas représentées. (d) Le modèle de (c) après la subdivision des trois autres faces hiérarchiques de Vol_+ . Nous faisons l'hypothèse que Vol_+ est l'hexaèdre ayant le plus faible niveau de subdivision parmi tous les autres hexaèdres du modèle. Cette hypothèse et la subdivision des dernières faces hiérarchiques entraînent la transformation de jonctions T frontalières en simples sommets. De plus, les brins labellisés de 0 à 5 correspondent au cycle de brins qui permettra d'insérer une face scindant Vol_+ en deux sous-volumes. Enfin, pour les quatre figures du schéma, les brins orange correspondent aux brins hiérarchiques de Vol_+ ou de Vol_1 . Les liens en rouge (resp. violet) sont les relations β_2 (resp. β_3 , partiellement représentées). Les traits bleus représentés au niveau de Vol_1 , délimitent les dix sous-volumes de Vol_1 adjacents à Vol_+ . Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées).

entre les figures 4.18(c) et (d), résident dans les jonctions T frontalières qui ont été transformées en simples sommets.

Le processus de subdivision ou de marquage hiérarchique des faces hiérarchiques de Vol_+ , commence par passer en revue l'ensemble des faces hiérarchiques de Vol_+ ou des sous-faces des faces hiérarchiques de Vol_+ si ces dernières ont été préalablement subdivisées. Suite à cette étape, les faces hiérarchiques F_1 et F_2 de Vol_+ ont été subdivisées en quatre sous-faces, comme expliqué précédemment. De plus, les sous-faces $F_{000|Vol_+}$, $F_{01|Vol_+}$, $F_{02|Vol_+}$ et $F_{033|Vol_+}$ ont été stockées dans \mathcal{F}_∞ , car elles possèdent un brin hiérarchique de Vol_+ . Ces sous-faces vont ensuite être traitées dans le but de marquer correctement leur face mère $F_0|Vol_+$. Le traitement est décrit par l'algorithme 11. A la fin de ce processus, toutes les faces hiérarchiques de Vol_+ n'ayant pas été préalablement subdivisées par des volumes adjacents ont été subdivisées, et ses autres faces hiérarchiques de Vol_+ ont quant à elles été marquées hiérarchiquement de la même manière que $F_0|Vol_+$.

2. Nous avons vu précédemment que l'insertion de la face F_a au milieu de Vol nécessite la construction d'un cycle de brins.
 - Nous partions pour cela d'un brin b qui était hiérarchique à la version initiale de Vol . Récupérer le brin b était alors trivial, puisque tous les brins de Vol étaient hiérarchiques. Or, ce n'est plus vrai pour Vol_+ , où nous devons employer l'algorithme 9 pour récupérer b . Ce dernier peut être contraint de parcourir tous les brins de Vol_+ à la recherche d'un brin hiérarchique.
 - Une fois le brin hiérarchique b récupéré, le premier brin b_0 de \mathcal{L} était $\beta_1(b)$. Or, ce n'est plus vrai pour Vol_+ . Pour récupérer le brin b_0 à partir du brin hiérarchique b , nous employons maintenant l'algorithme 5 (cf. la figure 4.18(d)). En effet, l'arête hiérarchique contenant le brin b , pouvant être scindée, nous devons récupérer le prochain brin hiérarchique de la face incidente à b et non plus simplement le $\beta_1(b)$.
 - Pour constituer l'ensemble des brins \mathcal{L} permettant l'insertion de F_a , nous appliquons successivement β_{121} . Or, nous ne pouvons plus nous contenter de cette méthode dans le cas de Vol_+ et nous appliquons maintenant l'algorithme 13.
 - Une fois F_a insérée selon \mathcal{L} , nous marquons hiérarchiquement quatre des huit brins de $F_a|Vol_0$ (resp. $F_a|Vol_1$), à partir de b_ϵ (resp. b_γ), en employant successivement β_{11} . Or, nous ne pouvons plus procéder de cette manière dans le cas de Vol_+ , pour passer de b_ϵ (resp. b_γ), au prochain brin de $F_a|Vol_0$ (resp. $F_a|Vol_1$) à marquer comme hiérarchique, nous utilisons maintenant l'algorithme 22 avec comme paramètres b_ϵ (resp. b_γ) et l_{Vol_+} . Nous précisons que cet algorithme consiste en une version simplifiée de l'algorithme 12, uniquement valable pour les faces simples.
3. Cette étape est identique à celle pour Vol .
4. — Le premier brin b_a inséré dans le cycle \mathcal{L}_0 qui scindera Vol_0 (resp. Vol_0), ne correspond pas à $\beta_1(b_0)$ mais à prochain-brin-hierarchique-face(b_0) (cf. l'algorithme 5).
 - Pour marquer hiérarchiquement quatre des six brins de F_{bc} , nous parcourons sa longueur en appliquant β_{11} et sa largeur en utilisant β_1 . Or, dans le cas de Vol_+ , nous parcourons la longueur (resp. largeur) de F_{bc} , en employant l'algorithme 22, avec comme paramètres le brin hiérarchique associé à cette longueur (resp. largeur) et l_{Vol_+} (resp. $2 l_{Vol_+}$). En effet, en accumulant les inverses des niveaux de subdivision des brins constituant sa longueur, nous obtenons $\frac{1}{l_{Vol_+}}$, alors que si l'on considère sa largeur, nous obtenons $\frac{1}{2 l_{Vol_+}}$.



5. Pour récupérer le brin $b_{\alpha\text{milieu}}$ à partir de b_{α} , nous n'appliquons plus simplement β_1 mais l'algorithme 10. De plus, pour obtenir b_{β} à partir de b_{α} , nous appliquons deux fois l'algorithme 5 ($b_{\beta} = \text{prochain-brin-hierarchique-face}(\text{prochain-brin-hierarchique-face}(b_{\alpha}))$).
6. Enfin, pour insérer une nouvelle face au milieu des nouveaux hexaèdres : Vol_{00} , Vol_{01} , Vol_{10} et Vol_{11} , les modifications faites pour gérer le cas de $Vol+$ sont les mêmes que celles présentées pour les étapes (3) et (5). Plus précisément, nous employons l'algorithme 5 au lieu de β_1 . De plus, pour parcourir la frontière d'une face hiérarchique, nous devons utiliser la méthode présentée dans les algorithmes 5, 8, 12 et 13. Enfin, l'algorithme 12, permet en s'appuyant sur le niveau de subdivision, de parcourir les différents côtés d'une face.

Marquer hiérarchiquement une face hiérarchique, préalablement subdivisée, d'un volume en cours de subdivision. Nous décrivons ici l'algorithme 11 en considérant les figures 4.18(b) et (c). Cet algorithme consiste à marquer hiérarchiquement une face hiérarchique F , ayant été préalablement subdivisée, d'un volume en cours de subdivision. Cet algorithme prend en paramètre un brin hiérarchique b incident à une sous-face F_n de F . Dans notre exemple, l'algorithme va marquer la face hiérarchique $F_{0|Vol+}$, en parcourant tout d'abord sa bordure à partir du brin hiérarchique b_a , afin de marquer hiérarchiquement les brins : $b_{\#}$ avec $\# \in \{a, c \text{ à } m\}$. Puis, il marque les brins internes de $F_{0|Vol+}$, qui sont b_i avec $i \in \{\alpha, \beta, \gamma, \delta\}$. Nous précisons que quelle que soit la configuration de la face hiérarchique à marquer hiérarchiquement, ses brins internes sont toujours au nombre de quatre. Enfin, suite à l'application de cet algorithme les brins de $F_{0|Vol+}$ seront marqués comme suite au processus de subdivision d'un quadrilatère en quatre sous-quadrilatères (cf. l'algorithme 19).

Plus précisément, nous appliquons l'algorithme 11 sur un des brins incidents à $F_{000|Vol+}$.

- La première étape a pour objectif de récupérer un brin hiérarchique appartenant à $F_{000|Vol+}$. Si le brin passé en paramètre de l'algorithme n'est pas b_a , alors nous appliquons l'algorithme 3, pour parcourir la frontière de $F_{000|Vol+}$ en employant β_1 jusqu'à tomber sur b_a .
- Puis, nous vérifions que $F_{0|Vol+}$ n'a pas déjà été marquée hiérarchiquement. Pour cela, nous appliquons l'algorithme 12 avec les paramètres b_a et $\frac{l_{Vol+}}{2}$, donnant b_c . b_c correspond au premier brin après b_a à devoir être marqué hiérarchiquement. Si ce dernier est déjà marqué hiérarchiquement, ceci implique que $F_{0|Vol+}$ a déjà été marquée hiérarchiquement et l'algorithme s'arrête. C'est par exemple le cas, si l'algorithme a été auparavant appliqué sur la sous-face $F_{01|Vol+}$. Enfin, notons que si nous étions partis de b_e (resp. b_h , b_k), nous aurions testé si b_f (resp. b_i , b_l) était hiérarchique de $Vol+$.
- Si $F_{0|Vol+}$ n'a pas déjà été marqué hiérarchiquement, nous procédons à la suite de l'algorithme en testant si b_c est un brin hiérarchique. Comme ce n'est pas le cas, nous le marquons comme hiérarchique puis nous allons sur $b_d = \beta_{21}(b_c)$, que nous marquons également comme hiérarchique. Puis, nous appliquons l'algorithme 12 avec les paramètres b_d et $2 l_{Vol+}$. Comme $l_{b_d} = \frac{1}{2}$, l'algorithme retourne le brin b_e . b_e est un brin hiérarchique de $Vol+$, nous récupérons donc $b_f = \text{prochain-brin-subdivision-face}(b_e, 2)$ que nous marquons comme hiérarchique. Nous procédons ainsi jusqu'à retrouver le brin b_a . Nous avons ainsi marqué hiérarchiquement les brins incidents à $F_{0|Vol+}$ en parcourant sa frontière, qui sont : $b_{\#}$ avec $\# \in \{a, c \text{ à } m\}$.
- Nous marquons maintenant les brins b_i avec $i \in \{\alpha, \beta, \gamma, \delta\}$, internes à $F_{0|Vol+}$. Pour cela, nous partons de notre brin hiérarchique de départ : b_a . Nous appliquons alors l'algorithme 12 avec les paramètres b_a et 2, donnant le brin b_c . En appliquant cet même

algorithme avec b_c comme paramètre, nous obtenons b_β que nous marquons alors comme hiérarchique. Enfin, nous marquons hiérarchiquement les trois autres brins internes, en appliquant successivement β_{21} .

Algorithme 11 : Marquer hiérarchiquement les brins d'une face hiérarchique d'un volume en cours de subdivision, ayant un niveau de subdivision plus important que son volume

Données : F_n : une sous-face d'une face hiérarchique F , contenant un brin hiérarchique.
 b : un brin incident à F_n .

Résultat : Les brins incidents à F sont correctement marqués hiérarchiquement.

$begin \leftarrow b$;

Si b n'est pas un brin hiérarchique **Alors**
 $b \leftarrow$ prochain-brin-hierarchique-face-simple(b);

$b \leftarrow$ prochain-brin-subdivision-face(b , $\frac{l_{Vol}}{2}$);

Si b n'est pas un brin hiérarchique **Alors**
 // Parcours de la frontière de F
 Répéter
 Si b n'est pas un brin hiérarchique **Alors**
 marquer-hierarchiquement(b);
 $b \leftarrow \beta_{21}(b)$;
 marquer-hierarchiquement(b);
 $b \leftarrow$ prochain-brin-subdivision-face(b , $\frac{l_{Vol}}{2}$);
 jusqu'à $begin = b$;

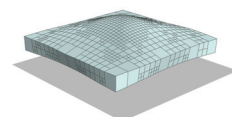
 // Traitement des brins internes à F
 $b \leftarrow$ prochain-brin-subdivision-face(prochain-brin-subdivision-face(b , $\frac{l_{Vol}}{2}$), $\frac{l_{Vol}}{2}$);

Pour i allant de 1 à 4 **faire**
 marquer-hierarchiquement(b);
 $b \leftarrow \beta_{21}(b)$;

Récupérer le prochain brin en fonction d'un niveau de subdivision. Nous détaillons ici l'algorithme 12 en nous appuyant sur la figure 4.18(c). Cet algorithme permet, à partir d'un brin donné, de parcourir la face hiérarchique incidente à b , en accumulant les niveaux de subdivision des brins parcourus, jusqu'à obtenir une valeur donnée. Une fois cette valeur obtenue, l'algorithme retourne le β_1 du brin sur lequel il s'est arrêté. Par exemple, dans la figure 4.18(c), nous faisons l'hypothèse que $Vol+$ n'a jamais été subdivisé, impliquant que $l_{Vol+} = 1$. En appliquant cet algorithme avec comme paramètres b_a et $2 l_{Vol+} = 2$, nous récupérons b_c .

Plus précisément, l'algorithme commence par affecter $\frac{1}{l_{b_a}}$ à $cumule$. $cumule$ vaut alors $\frac{1}{4}$. b'_a étant hiérarchique, nous passons sur le brin $b_b = \beta_{121}(b_a)$. $cumule$ étant inférieur à $\frac{1}{2}$, nous additionnons la valeur de $\frac{1}{l_{b_b}}$ à $cumule$ donnant $\frac{1}{2}$. b'_b étant hiérarchique, et $cumule$ étant égal à $\frac{1}{2}$, l'algorithme retourne alors $b_c = \beta_1(b_b)$.

Comme illustré par cet exemple, l'algorithme 12 permet de récupérer les différents côtés d'une face hiérarchique ou d'une sous-face d'une face hiérarchique, sans s'appuyer sur le marquage hiérarchique des brins, mais sur le niveau de subdivision de ces derniers. Il sert



ainsi, comme nous avons pu le voir précédemment, à marquer hiérarchiquement une face hiérarchique.

Algorithme 12 : Récupérer le prochain brin d'une face hiérarchique en fonction d'un niveau de subdivision

Données : b : un brin hiérarchique et l : un niveau de subdivision.

Résultat : Retourne le prochain brin de la face hiérarchique incidente à b après b , en fonction de l .

$cumule \leftarrow 0$;

Tant que $cumule < \frac{1}{l}$ **faire**

$cumule \leftarrow cumule + \frac{1}{l}$;

Tant que $\beta_3(b)$ n'est pas un brin hiérarchique **faire**

$b \leftarrow \beta_1(b)$;

$cumule \leftarrow cumule + \frac{1}{l}$;

Si $cumule \geq \frac{1}{l}$ **Alors**

Retourner $\beta_1(b)$;

$b_1 \leftarrow \beta_{121}(b_1)$;

Retourner $\beta_1(b)$;

Construire un ensemble de brins permettant l'insertion d'une face. Nous explicitons ici l'algorithme 13, en nous appuyant sur l'exemple de la figure 4.18(d). Cet algorithme prend comme paramètre b_0 , pour constituer l'ensemble de brins \mathcal{L} qui scindera de manière transversale Vol_+ en deux sous-volumes. L'algorithme commence par ajouter b_0 à \mathcal{L} . b_0 étant hiérarchique, nous passons au brin $b_1 = \beta_{121}(b_0)$. Nous ajoutons alors b_1 à \mathcal{L} , puis le brin $b_2 = \beta_{121}(b_1)$. b_2 n'étant pas hiérarchique, nous appliquons β_1 au lieu de β_{121} , permettant d'obtenir b_3 que nous ajoutons à \mathcal{L} . Nous procédons ainsi jusqu'à retomber sur b_0 .

Algorithme 13 : Constituer un ensemble de brins afin d'insérer une face

Données : b_0 : le premier brin à insérer dans le cycle de brins.

Résultat : Retourne un cycle de brins \mathcal{C} .

$b \leftarrow b_0$;

Répéter

 ajouter-au-cycle(\mathcal{C} , b);

Tant que $\beta_3(b)$ n'est pas un brin hiérarchique **faire**

$b \leftarrow \beta_1(b)$;

 ajouter-au-cycle(\mathcal{C} , b);

$b \leftarrow \beta_{121}(b)$;

jusqu'à $b = b_0$;

Retourner \mathcal{C} ;

Pour conclure cette partie, nous avons présenté de manière intuitive comment subdiviser topologiquement un hexaèdre quel que soit le niveau de subdivision de ses hexaèdres adjacents. L'algorithme complet permettant la subdivision d'un quadrilatère en quatre sous-quadrilatères, est présenté dans la section 7.2.

4.4.3 Initialisation et mise à jour des informations physiques

Lors de la description du processus de subdivision d'un quadrilatère (*cf.* la section 4.3), nous avons noté que de nouvelles Particules et de nouveaux Ressorts devaient être créés pour être associés aux nouveaux sommets et arêtes insérés au cours du processus. Tout comme le processus de subdivision d'un quadrilatère, le processus de subdivision d'un hexaèdre implique également la création et l'initialisation de nouvelles Particules et Ressorts, ainsi que la mise à jour des autres Particules et Ressorts déjà existants, impliqués dans le processus de subdivision, afin de conserver les propriétés initiales de l'objet.

L'explication de cette partie sera construite de la même manière que celle dédiée à l'initialisation et la mise à jour des informations physiques pour le cas 2D. Nous précisons qu'au cours du processus de subdivision d'un hexaèdre, la création de nouveaux sommets et arêtes est exclusivement réalisée lors de la subdivision d'une face en deux sous-faces (*cf.* l'algorithme 19) et lors de la subdivision d'une face en quatre sous-faces (*cf.* l'algorithme 23). Enfin, par la suite, nous nommerons Vol , le volume venant d'être subdivisé.

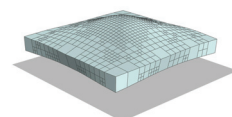
4.4.3.1 Masses

Les sommets créés pendant le processus de subdivision ainsi que les autres sommets incidents à Vol doivent respectivement initialiser ou mettre à jour leur masse. Nous rappelons que cette dernière est stockée dans la Particule qui leur est associée. Cette initialisation ou mise à jour est effectuée à la fin du processus de subdivision du volume. Pour cela, nous parcourons l'ensemble des sommets incidents aux huit nouveaux sous-volumes engendrés par la subdivision de Vol , et recalculons leur masse de manière similaire à l'initialisation (*cf.* la sous-section 3.2.1). Ce calcul nécessite de déterminer le volume des hexaèdres incidents au sommet dont nous souhaitons calculer la masse. Pour cela, nous récupérons les dimensions de l'hexaèdre considéré en prenant les longueurs initiales L_0 stockées au niveau des ressorts associés à trois arêtes hiérarchiques de Vol partant du même sommet. Ces dernières seront ensuite divisées par le niveau de subdivision de Vol . Notons que nous avons procédé de la même manière pour calculer l'aire d'une face. En effet, nous rappelons qu'un élément E issu d'une subdivision a pour dimensions initiales, celles de l'élément initial dont il provient rapportées à son niveau de subdivision l_E . Les dimensions de E ne sont donc pas issues des dimensions déformées au moment de la subdivision de l'élément mère.

Cas des jonction T Enfin, comme pour le cas 2D, une procédure particulière doit être mise en place pour le calcul de la masse des jonctions T. En effet, à la fin du processus de subdivision de Vol , pour calculer la masse d'une particule p , nous parcourons tous ses volumes incidents. Si le brin associé est marqué alors le volume contribue au calcul de la masse de p , sinon il ne contribue pas. Nous détaillerons ci-dessous, le processus de création, de mise à jour et de suppression des jonctions T qui interviennent au cours de la subdivision de Vol .

4.4.3.2 Ressorts diagonaux

La mise à jour des informations physiques se poursuit avec les ressorts diagonaux internes. Au début du processus de subdivision de Vol , ses ressorts diagonaux sont supprimés. Une fois que le processus de subdivision a été réalisé, nous créons quatre nouveaux ressorts diagonaux pour chacun des huit sous-volumes nouvellement créés.



- **Suppression.** Nous commençons par prendre un brin hiérarchique b de Vol . Puis, nous récupérons les quatre sommets hiérarchiques de la face hiérarchique incidente à b , en utilisant l'algorithme 5, que nous abrègerons par la suite par « PBHF ». Nous récupérons ensuite pour chacun de ces quatre sommets hiérarchiques, son sommet diagonalement opposé relativement à Vol .

Plus précisément, pour chacun des quatre sommets hiérarchiques s_n , nous prenons un brin b_n partant de s_n et incident à Vol . Puis, avec l'opération :

$PBHF(PBHF(\beta_2(PBHF(PHBF(b_n))))))$, nous obtenons le sommet diagonalement opposé à s_n dans Vol . Enfin, maintenant que nous avons les deux extrémités du ressort diagonal, nous pouvons procéder à sa suppression. De plus, nous l'enlevons également de la liste des ressorts diagonaux de ses deux particules extrémités tout juste déterminées.

- **Création.** A la fin du processus de subdivision de Vol , nous parcourons ses huit nouveaux sous-volumes Vol_n . Pour chacun d'entre eux, nous procédons comme pour la suppression, en commençant par récupérer un brin hiérarchique de Vol_n , puis en stockant quatre sommets hiérarchiques et enfin en récupérant les sommets hiérarchiques diagonalement opposés. Un nouveau ressort diagonal est alors créé pour chaque paire d'extrémités préalablement déterminées. Enfin, chaque ressort diagonal nouvellement créé, stocke un brin incident à Vol_n et a pour niveau de subdivision $2 l_{Vol}$. Nous calculons ensuite sa raideur de la même manière que pour l'initialisation (cf. la sous-section 3.2.2) sauf pour déterminer les dimensions de Vol .

Nous précisons que la **longueur initiale** du ressort diagonal d inclus dans l'hexaèdre Vol , nécessaire au calcul de sa force, est égale à $\sqrt{X^2 + Y^2 + Z^2}$ où $X = \frac{L_0(r)}{l_{Vol}}$, $Y = \frac{L_0(s)}{l_{Vol}}$ et $Z = \frac{L_0(t)}{l_{Vol}}$, avec r , s et t , les ressorts associés à trois arêtes hiérarchiques de Vol partant du même sommet.

4.4.3.3 Ressorts non diagonaux

L'ensemble des Ressorts associés aux différentes arêtes incidentes aux sous-volumes de Vol doivent être initialisés ou mis à jour. Pour cela, nous procédons de la même manière que pour l'initialisation (cf. la sous-section 3.2.2) sauf pour les dimensions. Par ailleurs, comme nous l'avons vu pour le cas 2D (cf. sous-section 4.3.3), la raideur obtenue est relative à une arête hiérarchique de Vol . Si nous mettons à jour une sous-arête a d'une arête hiérarchique, il faut donc pondérer cette raideur. La raideur $k_{Vol(i,j,k)|_a}$ d'un ressort associé à l'arête a (de niveau de subdivision l_a) de Vol est formulée de la manière suivante :

$$k_{Vol(i,j,k)|_a} = k_{Vol(i,j,k)} \frac{l_a}{l_{Vol}}$$

où $k_{Vol(i,j,k)}$ correspond à la raideur de l'arête hiérarchique contenant l'arête a . i , j et k sont les dimensions du volume Vol et correspondent aux longueurs initiales L_0 des ressorts associés à trois brins hiérarchiques de Vol partant du même sommet, pondérées par le niveau de subdivision de Vol .

Nous précisons que la **longueur initiale** du ressort r , nécessaire au calcul de sa force, est égale à $\frac{L_0(r)}{l_r}$.

4.4.3.4 Vitesses et accélérations

Lors du processus de subdivision d'une face en quatre sous-faces (cf. l'algorithme 19), la vitesse (resp. l'accélération) d'une particule insérée dans une arête hiérarchique a de Vol , est

initialisée avec la moyenne des vitesses (*resp.* accélérations) des deux particules extrémités de a . Le sommet inséré au milieu de la face F en cours de subdivision, aura, quant à lui, pour vitesse (*resp.* accélération), la moyenne des vitesses (*resp.* accélérations) des quatre sommets hiérarchiques de F .

4.4.3.5 Contraintes

Lors du processus de subdivision d'une face en quatre sous-faces (*cf.* l'algorithme 19), si une nouvelle particule p est insérée au milieu d'une arête hiérarchique dont ses deux extrémités étaient contraintes, alors par défaut p sera contrainte de la même manière. Il en est de même pour la particule p_{milieu} insérée au milieu de la face F , si ses quatre sommets hiérarchiques sont contraints.

4.4.3.6 Jonction T

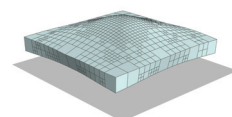
Nous expliquons ici, le processus de création, de mise à jour et de suppression des jonctions T, intervenant durant la subdivision de Vol .

- **Création.** Lors de la première étape du processus de subdivision de Vol consistant à subdiviser les six faces hiérarchiques de Vol , si une de ses faces hiérarchiques F n'a pas été subdivisée en amont par un volume adjacent, alors elle est soumise au processus de subdivision d'une face en quatre nouvelles sous-faces, décrit dans l'algorithme 19.

Lors de la scission de l'arête hiérarchique d'une face hiérarchique en cours de subdivision La première étape de cet algorithme consiste à scinder les quatre arêtes hiérarchiques de F . Si nous insérons un nouveau sommet s dans une des arêtes hiérarchiques de F et que cette dernière est commune à au moins deux hexaèdres, alors s devient une jonction T. Nous précisons que s sera une jonction T pour tous les volumes autres que Vol qui sont incidents à s . Il nous faut maintenant déterminer si s est une jonction T frontalière, centrée ou bivalente. Pour cela, nous commençons par récupérer un brin b partant de s . Nous regardons ensuite les particules s_a et s_b associées respectivement aux brins précédant et suivant b .

- **Création d'une jonction T frontalière.** Si l'un des sommets s_a ou s_b n'est pas une jonction T ou que les particules s_a et s_b sont des jonctions T frontalières, alors s sera une jonction T frontalière. En effet, cette configuration implique forcément que s n'est pas incluse dans une face hiérarchique d'un volume adjacent et est donc une jonction T frontalière. C'est-à-dire que s est incluse dans une arête qui est hiérarchique pour tous les volumes qui lui sont incidents. Ensuite, nous récupérons les deux extrémités de projection de cette nouvelle jonction T frontalière, en utilisant l'algorithme 5. Ces dernières correspondent aux extrémités de l'arête hiérarchique associée à b .

La figure 4.19(a) illustre ce cas. La première étape du processus de subdivision de Vol_1 consiste en la subdivision de la face F_0 commune aux hexaèdres Vol_0 et Vol_1 . L'insertion de s dans l'arête hiérarchique a de F_0 scinde a en deux parties. s étant incident à deux hexaèdres de même niveau de subdivision, nous récupérons indifféremment un brin b partant de s . Les brins précédant et suivant b partent tous deux d'une particule qui n'est pas une jonction T : s_a et s_b . s devient donc une jonction T frontalière avec pour extrémités de projection les sommets marqués d'une croix rose : s_a et s_b , correspondant aux extrémités de a .



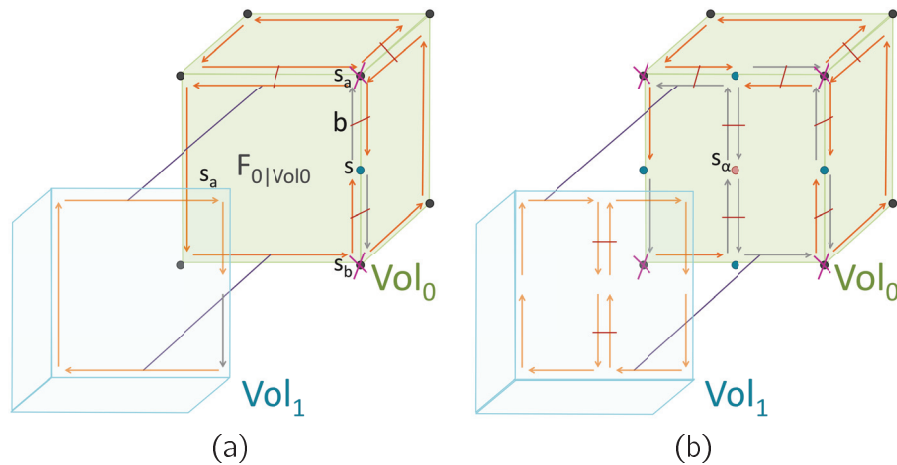


Figure 4.19 – L’initialisation des extrémités de projection d’une jonction T nouvellement créée, pour le cas 3D. (a) Un modèle 3D composé de deux hexaèdres adjacents Vol_0 et Vol_1 , cousus ensemble par β_3 où la figure montre la première étape du processus de subdivision de Vol_1 . La face F_0 commune aux deux hexaèdres est en train d’être subdivisée. Nous pouvons voir qu’une arête hiérarchique de F_0 a été scindée en deux par l’insertion du sommet s . s est par définition une jonction T frontalière. Les deux extrémités de projection de s correspondent aux sommets marqués d’une croix rose. (b) Le modèle de (a) après la subdivision de F_0 en deux sous-faces. Nous pouvons observer quatre jonctions T frontalières et une jonction T centrée s_α . Les croix roses correspondent aux quatre extrémités de projection de s_α . Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées).

La figure 4.20 explicite également ce propos avec un cas plus complexe que le précédent. La figure 4.20(a) montre un modèle 3D composé de deux hexaèdres adjacents : Vol_0 et Vol_1 . Initialement, Vol_1 a été subdivisé une première fois, puis un de ses sous-volumes : Vol_{1000} a également été subdivisé. La figure 4.20(b) présente le modèle de (a) après la subdivision du sous-volume Vol_{1100} . Le sommet s_1 a été créé lors du processus de subdivision de la face F_{03} . Lorsque s_1 a été inséré dans l’arête hiérarchique de F_{03} , il était alors entouré d’une jonction T frontalière s_b et d’un sommet normal s_c . Il est donc devenu une jonction T frontalière avec pour extrémités de projection les sommets entourés de rouge.

- **Création d’une jonction T centrée.** Si l’un des sommets s_a ou s_b est une jonction T centrée et que s n’est incident qu’à deux volumes, alors s sera une jonction T centrée. En effet, cette configuration implique forcément que s est incluse dans une face hiérarchique du volume adjacent et est donc une jonction T centrée. Ensuite, nous affectons les quatre extrémités de projection de cette nouvelle jonction T centrée, avec celles de la jonction T centrée s_a ou s_b .

La figure 4.20 illustre ce propos. En effet, le sommet s_α inséré dans l’arête hiérarchique de F_{03} était entouré des sommets s_a et s_d , qui étaient respectivement une jonction T centrée et une jonction T frontalière. C’est pour cette raison que s_α est devenu une jonction T centrée et ses quatre extrémités de projection correspondent à celles de la jonction T centrée s_a .

Ces dernières ont été calculées suite à l’insertion de s_a lors de la subdivision de F_0 , engendrée par la première subdivision de Vol_1 .

- **Création d’une jonction T bivalente.** Si l’un des sommets s_a ou s_b est une jonction T centrée mais que s est incident à trois volumes, alors s sera une jonction T bivalente, conformément à la définition faite dans la sous-section 4.1.5. Nous

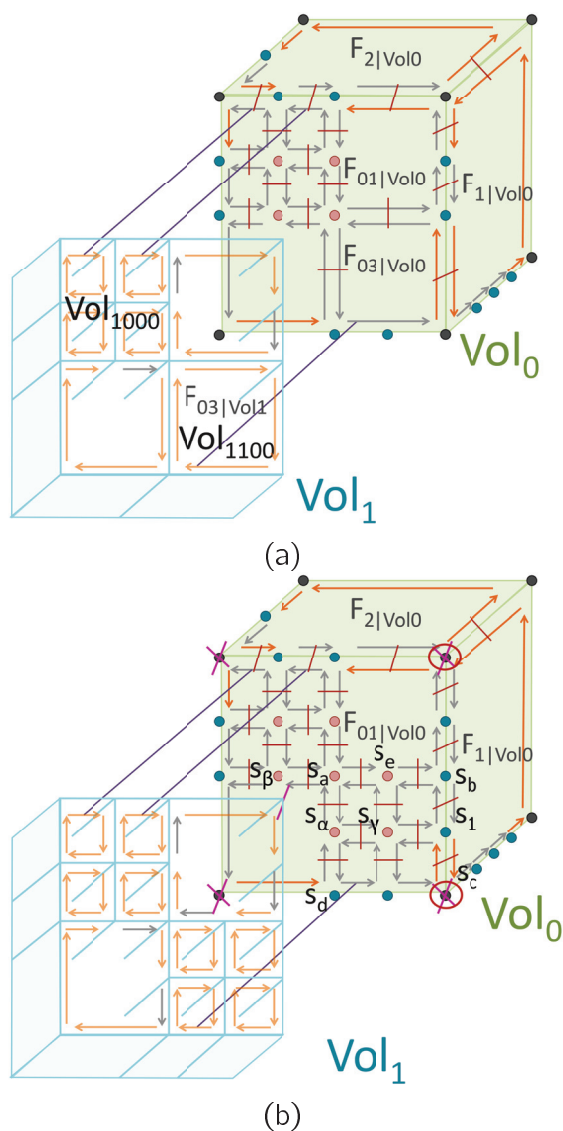
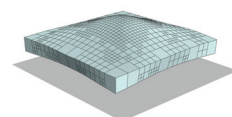


Figure 4.20 – L’initialisation des extrémités de projection d’une jonction T nouvellement créée dans une face hiérarchique qui a déjà un niveau de subdivision plus grand que son volume associé, pour le cas 3D. (a) Un modèle 3D composé de deux hexaèdres adjacents Vol_0 et Vol_1 , cousus ensemble par β_3 . Vol_1 a été subdivisé une première fois en huit sous-volumes. Un de ses sous-volumes Vol_{1000} se trouvant être adjacent à Vol_0 est également subdivisé en huit sous-volumes. (b) Le modèle de (a) après la subdivision du sous-volume Vol_{1100} de Vol_1 . Cette subdivision engendre la création de deux jonctions T frontalières et de trois jonctions T centrées. Les traits bleus représentés au niveau de Vol_1 , délimitent les sept (resp. dix) sous-volumes de Vol_1 adjacents à Vol_0 , pour la figure (a) (resp. (b)). Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées).



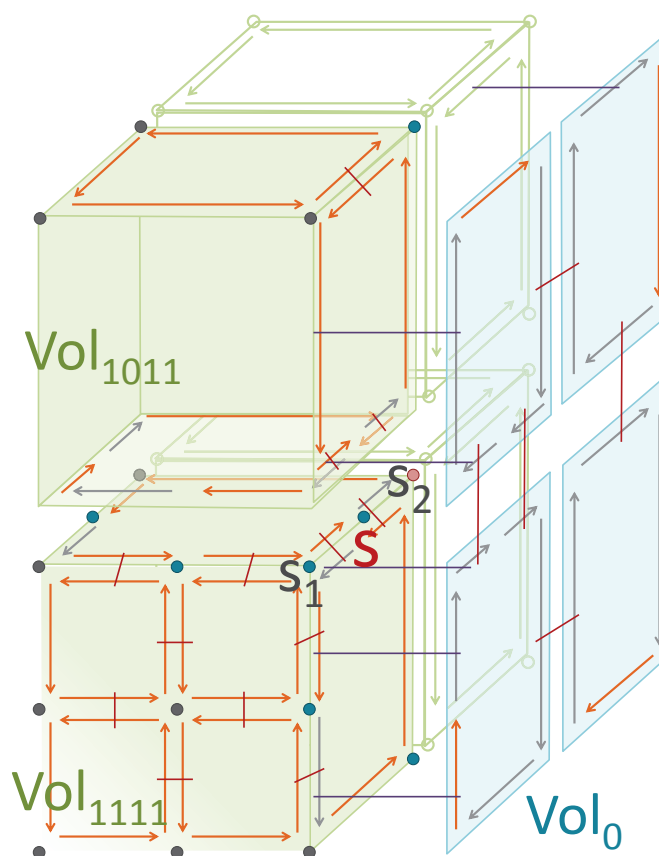


Figure 4.21 – La présentation de la notion de jonction T bivalente et mise à jour des extrémités de projection d'une jonction T bivalente, pour le cas 3D. Un modèle 3D composé de cinq hexaèdres cousus ensemble par β_3 . s correspond à une jonction T bivalente. Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées). Seulement une partie des liaisons β_3 ont été représentées. Enfin, l'ensemble des brins constituant Vol_{1011} et Vol_0 est partiellement représenté.

notons Vol_0 , Vol_1 et Vol_2 , les trois volumes incidents à s avec $l_{Vol_0} \leq l_{Vol_1} < l_{Vol_2}$. Pour récupérer les deux extrémités de projection de s , nous prenons tout d'abord un brin partant de s et incident à Vol_1 , ou Vol_0 si $l_{Vol_0} = l_{Vol_1}$. En effet, nous ne pourrions pas récupérer les extrémités de projection à partir de Vol_0 puisque que l'arête dans laquelle a été insérée s ne compose pas une des arêtes hiérarchiques de Vol_0 .

La figure 4.21 illustre ce cas. Vol_{1111} est en cours de subdivision. Nous pouvons observer qu'une de ses faces hiérarchiques a déjà été subdivisée. De plus, sa face hiérarchique, notée F , adjacente à Vol_0 est en cours de subdivision. La figure montre le modèle après l'insertion de s dans une des arêtes hiérarchiques de F . s est alors une jonction T frontalière pour Vol_{1011} car se trouvant sur sa bordure, mais est également une jonction centrée pour Vol_0 car situé à l'intérieur de sa face hiérarchique. Le sommet s est donc une jonction T bivalente. Nous parcourons ensuite l'arête hiérarchique de Vol_{1011} ou Vol_{1111} , pour récupérer les deux extrémités de projection de s : s_1 et s_2 .

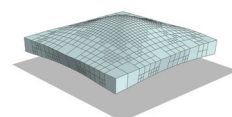
Lors de la scission de la nouvelle arête ayant scindé en deux une face hiérarchique en cours de subdivision

- **Création d'une jonction T centrée.** Une fois que les quatre arêtes hiérarchiques de F ont été scindées, nous insérons une arête a scindant en deux la face hiérarchique F , conformément au processus de subdivision d'une face présenté dans l'algorithme 19. Puis, nous scindons a en deux en insérant un nouveau sommet s_α . Si s_α est incident à deux volumes, alors ce dernier sera, par définition, une jonction T centrée (cf. la sous-section 4.1.5). Nous avons en amont de l'insertion de s_α , récupéré les quatre sommets hiérarchiques de F . Nous comparons alors les niveaux de subdivision de Vol et de l'autre volume incident à s_α . S'ils ont le même niveau de subdivision alors s_α aura comme extrémités de projection les quatre sommets hiérarchiques de F préalablement stockés (cf. le sommet s_α de la figure 4.19(b)). Sinon (cf. le sommet s_γ de la figure 4.20(b)), nous récupérons un brin partant de s_α et incident au volume ayant le plus faible niveau de subdivision. Puis, nous parcourons avec β_1 une des sous-faces incidentes à s_α jusqu'à tomber sur un brin qui n'est pas hiérarchique et qui part d'une autre jonction T centrée s_β (cf. le sommet s_e de la figure 4.20(b)). Nous affectons alors les extrémités de projection de s_α avec celles de s_β .

— Transformation d'une jonction T centrée ou bivalente.

- **Scission des quatre arêtes hiérarchiques d'une face hiérarchique** Nous rappelons que la première étape du processus de subdivision d'une face en quatre sous-faces consiste à insérer un sommet dans chacune des quatre arêtes hiérarchiques de cette face.

Exemple de la transformation d'une jonction T bivalente en centrée Si une de ces quatre arêtes hiérarchiques a déjà été scindée et que le sommet, nommé s , au milieu de cette dernière est une jonction T bivalente, alors il sera transformé en une jonction T uniquement centrée. En effet, s se trouvant sur une arête hiérarchique du volume Vol en cours de subdivision, il est une jonction T frontalière au regard de Vol . Or, suite à la subdivision de Vol , s n'est plus une jonction T pour ce dernier mais reste une jonction T centrée pour le volume incident de plus faible niveau de subdivision.



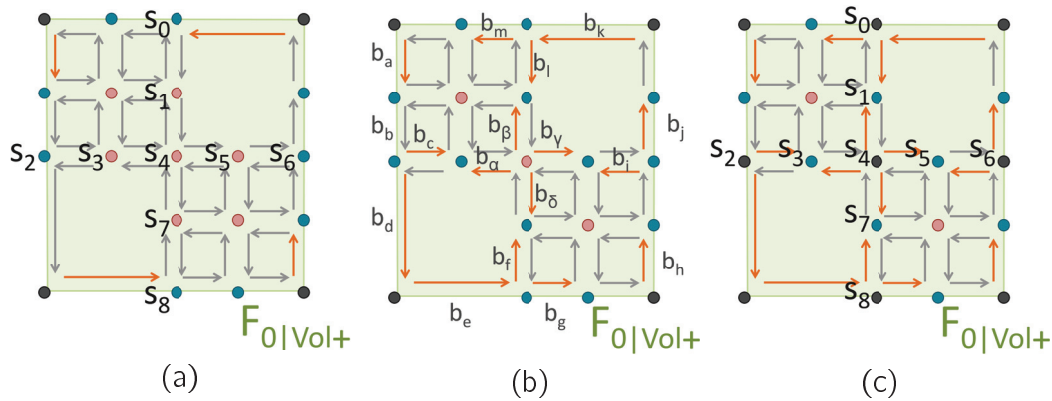


Figure 4.22 – La modification de l'état de jonctions T relatives à la face hiérarchique ayant un niveau de subdivision plus grand que son volume associé en cours de subdivision, pour le cas 3D. (a) Un zoom sur la face $F_{0|Vol_0}$ de la figure 4.20(b). (b) Le modèle de (a) après le traitement visant à marquer correctement une face hiérarchique préalablement subdivisée (cf. l'algorithme 11). Nous pouvons observer que les sommets s_1, s_3, s_5 et s_7 qui étaient des jonctions T centrées dans (a) sont maintenant des jonctions T frontalières. (c) Le modèle de (b) à la fin du processus de subdivision de Vol_+ . Nous constatons que les particules s_0, s_2, s_4, s_6 et s_8 ne sont plus des jonctions T . Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées).

La figure 4.21 permet d'illustrer cette transformation avec la jonction T bivalente s . Si le volume Vol_{1011} venait à être subdivisé, alors sa face hiérarchique F adjacente à Vol_0 serait subdivisée en quatre sous-faces. Le sommet s serait alors transformé en une jonction T uniquement centrée. En effet, nous constatons que s serait uniquement une jonction T centrée pour Vol_0 . Enfin, pour mettre à jour les quatre extrémités de projection de cette jonction T maintenant uniquement centrée, une solution consiste à prendre un brin b partant de cette dernière et incident au volume de niveau de subdivision le plus faible. Puis, une des sous-faces incidentes à b , serait parcourue, jusqu'à tomber sur une jonction T centrée. Les extrémités de projection de cette dernière seraient alors récupérées pour les affecter à celles de s . Dans notre exemple, les quatre extrémités de projection de s seraient affectées à celles de s_2 .

- Marquage d'une face hiérarchique ayant été subdivisée** Après subdivision de l'ensemble des faces hiérarchiques de Vol n'ayant pas été préalablement subdivisées par les volumes adjacents, nous marquons hiérarchiquement les autres faces hiérarchiques de Vol , en utilisant l'algorithme 11. Nous rappelons que cet algorithme commence à marquer hiérarchiquement les brins accessibles depuis la bordure de F , puis les quatre brins internes à F . Ces quatre derniers brins partent du sommet situé au centre de F , que nous nommons s_{milieu} . Ils vont permettre de parcourir les jonctions T centrées situées sur les arêtes hiérarchiques des quatre nouvelles sous-faces de F , ayant pour extrémité s_{milieu} . En effet, ces jonctions T centrées doivent être transformées en jonctions T frontalières, car elles ne vont plus se situer à l'intérieur de F mais sur les arêtes hiérarchiques des sous-volumes en lesquels Vol aura été subdivisé. Enfin, leurs extrémités de projection seront mises à jour à la fin du processus de subdivision de Vol . Notons que s'il s'avère que pendant ce parcours, nous tombons sur des jonctions T bivalentes, alors elles deviendraient de simples jonctions T frontalières. En effet, ces dernières étaient des jonctions T centrées en plus d'être frontalières, à cause de Vol . Or, ce dernier allant être subdivisé, les jonctions T bivalentes se trouvant sur les quatre nouvelles

arêtes internes de sa face hiérarchiques incidente, ne sont plus que frontalières.

Exemple de la transformation d'une jonction T centrée en frontalière La figure 4.22(b) illustre cette transformation. La face hiérarchique F_0 de Vol_+ , présentée dans la figure 4.22(a) doit être marquée hiérarchiquement. Une fois que sa frontière a été parcourue pour marquer hiérarchiquement les brins $b_{\#}$ avec $\# \in \{a, c \text{ à } m\}$, nous traitons les brins b_i avec $i \in \{\alpha, \beta, \gamma, \delta\}$. Puis, nous allons profiter de ce traitement pour transformer les jonctions T centrées s_1, s_3, s_5 et s_7 en jonctions T frontalières. En effet, le volume Vol_+ étant en train d'être subdivisé, ces jonctions T ne se trouveront plus au centre de la face hiérarchique de $F_{0|Vol_+}$ mais sur les arêtes hiérarchiques des faces hiérarchiques de ses sous-volumes de Vol_+ . Pour cela, pour chaque brin b_i avec $i \in \{\alpha, \beta, \gamma, \delta\}$, nous appliquons l'algorithme 5, donnant respectivement les brins b_d, b_k, b_j et b_g , partant respectivement des sommets s_2, s_0, s_6 et s_8 . Toutes les jonctions T centrées traversées pendant cette procédure deviendront des jonctions T frontalières. Leurs extrémités de projection seront mises à jour à la fin du processus de subdivision de Vol_+ .

Exemple de la transformation d'une jonction T bivalente en centrée Les jonctions T bivalentes situées à une extrémité d'une des nouvelles arêtes hiérarchiques partant de s_{milieu} (cf. s_4), deviendront des jonctions T uniquement centrées. Afin de se représenter la position de ces sommets, nous considérons la figure 4.22(b). Dans cet exemple s_{milieu} correspond à s_4 et les extrémités des nouvelles arêtes hiérarchiques partant de s_4 sont : s_0, s_2, s_6 et s_8 . Pour opérer cette transformation, nous procédons de la même manière que pour le cas de la subdivision d'une face dont l'arête hiérarchique a été scindée en deux par une jonction T bivalente. Les figures 4.24 et 4.25 illustrent cette transformation. Dans la figure 4.24, le sommet s_3 est une jonction T bivalente, en étant une jonction T frontalière pour le volume Vol_{1100} et une jonction T centrée pour le volume Vol_0 . Lors de la subdivision de Vol_0 , $F_{0|Vol_0}$ a été marquée hiérarchiquement et le sommet s_3 est devenu une jonction T uniquement frontalière. La figure 4.25 montre l'état du modèle après la subdivision de Vol_0 .

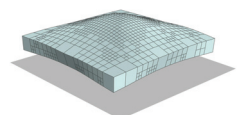
— **Suppression du statut de jonction T d'un sommet ou mise à jour des extrémités de projection des jonctions T.**

A la fin du processus de subdivision de Vol , lors de la mise à jour de la masse d'une particule p , nous parcourons tous les volumes incidents à p . Si un des brins n'est pas marqué comme hiérarchique, alors cela indique que p est encore une jonction T. En effet, si l'ensemble des brins incidents à un sommet sont marqués comme hiérarchiques alors ce dernier n'est pas ou plus de type jonction T. Si p reste une jonction T, alors ses extrémités de projection doivent être mises à jour. Notons que toutes les jonctions T incidentes aux huit nouveaux sous-volumes de Vol et pour lesquelles ces derniers sont non conformes, sont marquées comme devant mettre à jour leurs extrémités de projection. Cette mise à jour est réalisée par l'algorithme 14.

— **Traitement des jonctions T**

Nous finissons cette partie sur les jonctions T, en présentant la manière dont nous les traitons au cours de la simulation. Nous présenterons tout d'abord la contrainte des jonctions T, puis la répartition de leur force sur leurs voisins.

- **Contrainte des jonctions T** Comme pour le cas 2D, une jonction T doit être contrainte pour éviter un comportement erroné. En 3D, à chaque itération de



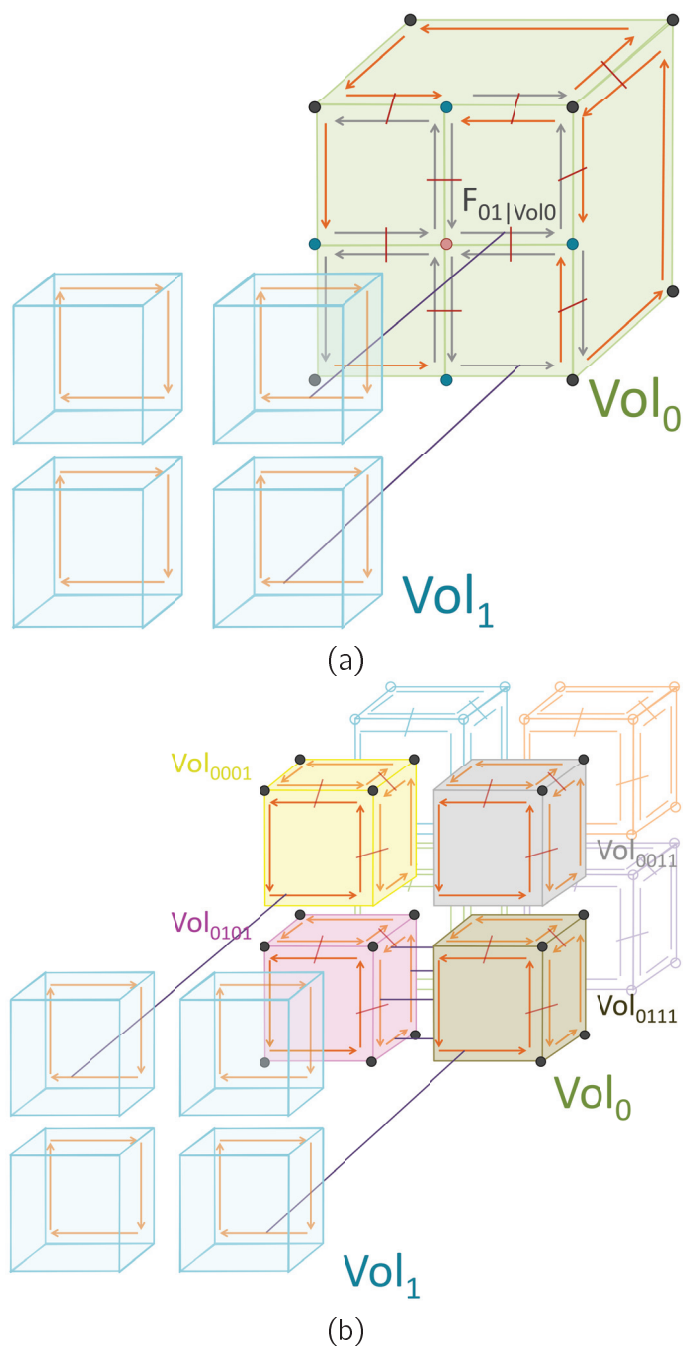


Figure 4.23 – L'évolution de l'état de jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D. (a) Le modèle de la figure 4.19(a) à la fin du processus de subdivision de Vol_1 . (b) Le modèle de (a) après la subdivision de Vol_0 en huit sous-volumes. Les quatre sommets de type jonction T frontalières et le sommet de type jonction T centrée, représentés dans (a), sont maintenant de simples sommets. Nous avons ici fait le choix visuel de scinder les différentes parties de la représentation de Vol_1 selon ses quatre sous-volumes adjacents à Vol_0 . Les particules bleues (resp. roses) correspondent aux jonctions T frontalières (resp. centrées).

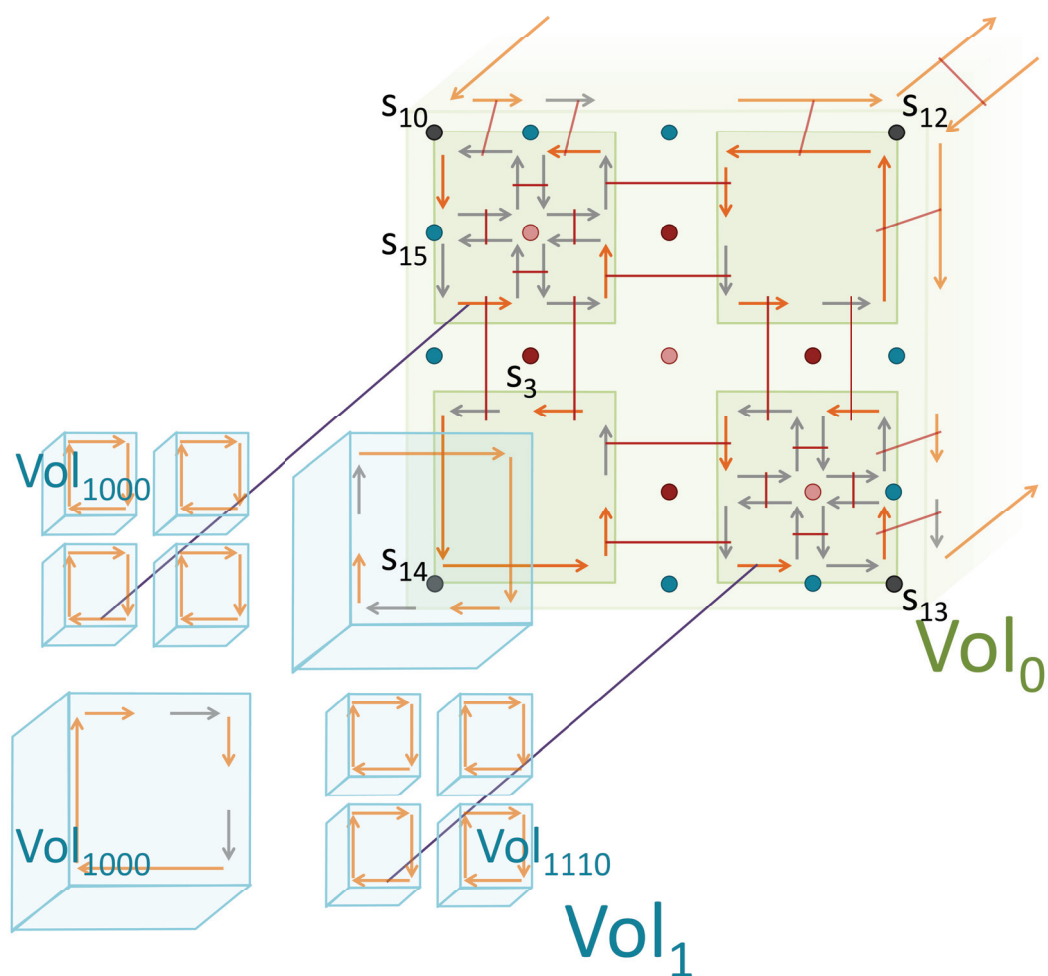
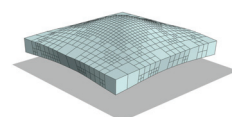


Figure 4.24 – La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D. Un objet 3D initialement composé de deux hexaédres : Vol_0 et Vol_1 , cousus ensemble par β_3 . Vol_1 a été subdivisé en huit sous-volumes, puis deux de ses sous-volumes : Vol_{1000} et Vol_{1110} , ont été subdivisés une nouvelle fois. Les particules bleues (resp. roses, rouges) correspondent aux jonctions T uniquement frontalières (resp. uniquement centrées, bivalentes). Seulement une partie des liaisons β_3 a été représentée.



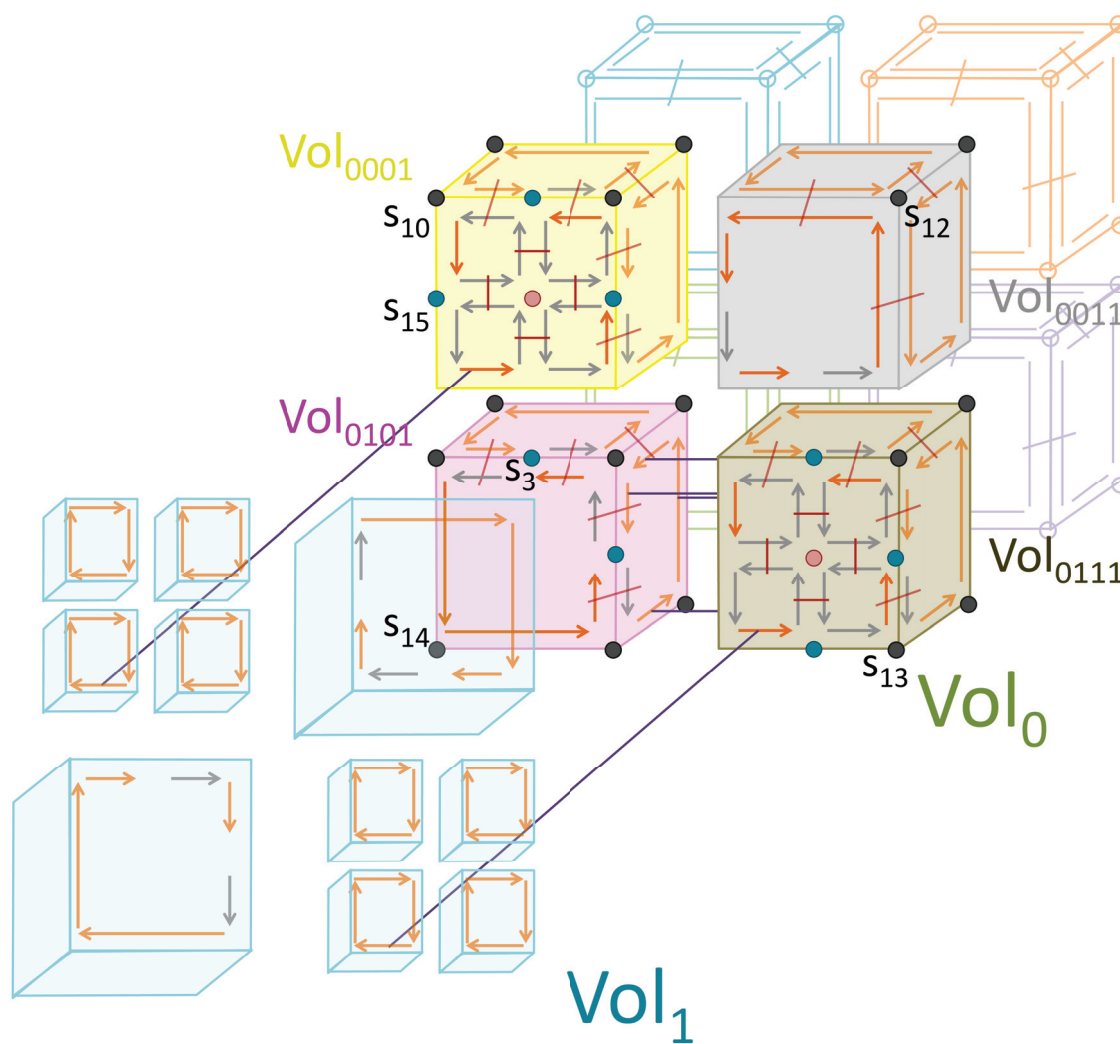


Figure 4.25 – La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D. Le modèle de la figure 4.24 suite à la subdivision de Vol_0 . Nous pouvons voir la transformation des jonctions T bivalentes, en jonctions T frontalières et les jonctions T centrées ou frontalières, en simple sommet.

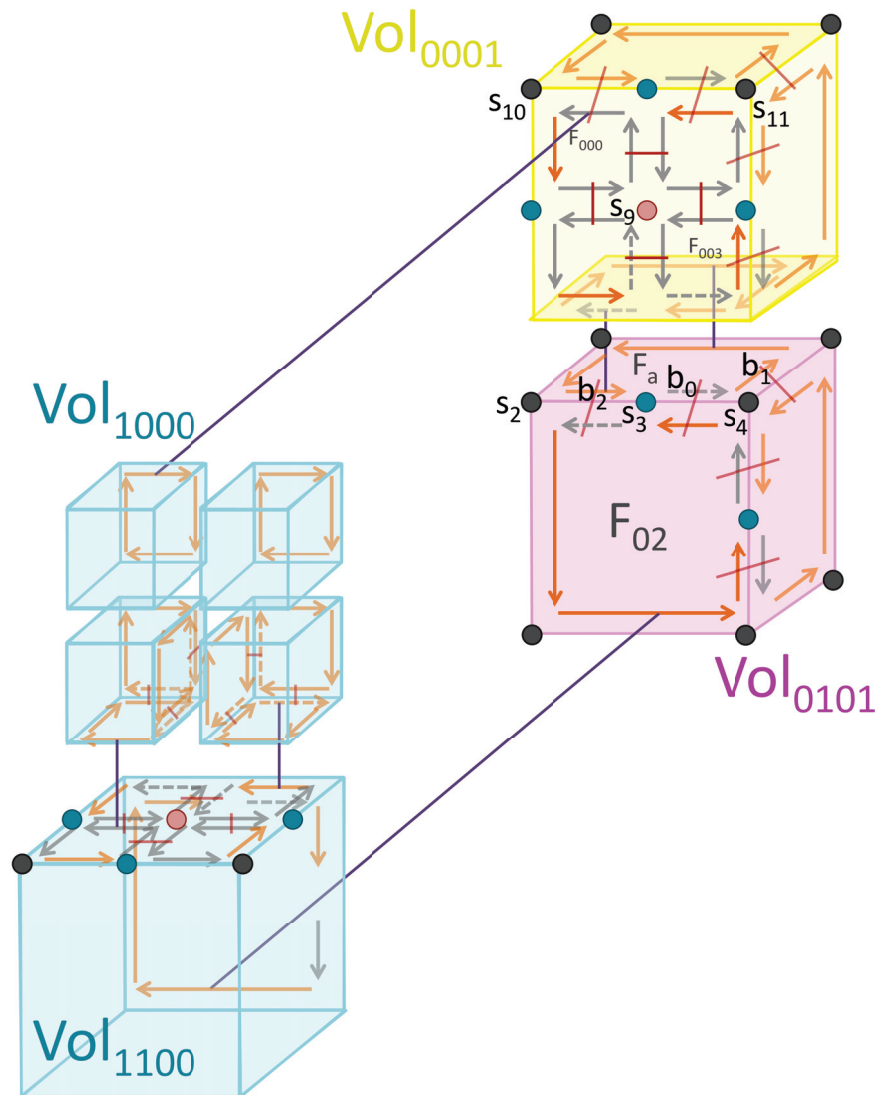
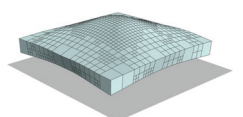


Figure 4.26 – La mise à jour des extrémités de projection des jonctions T qui sont relatives à un volume en cours de subdivision, pour le cas 3D. Un zoom du modèle présenté dans la figure 4.25 où est représenté l'ensemble des brins incidents à s_3 .



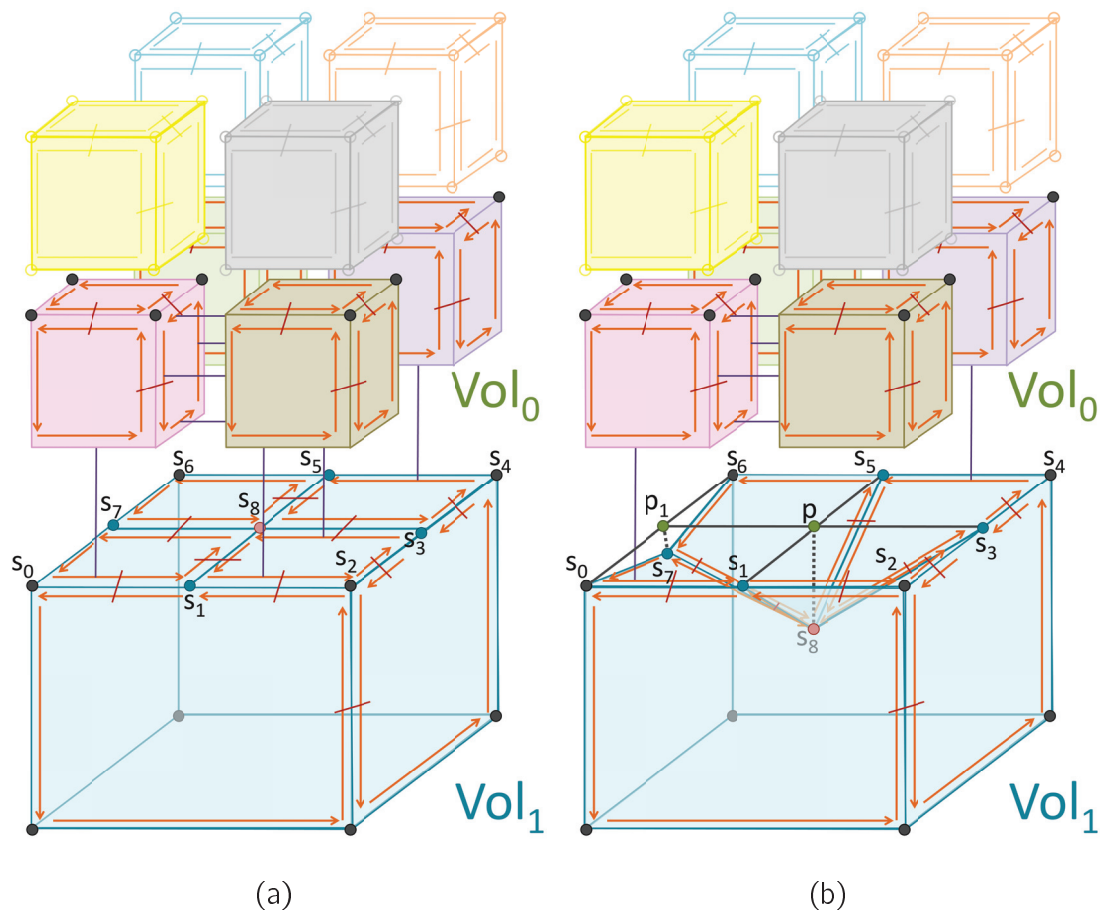


Figure 4.27 – Le comportement erroné induit par la création de jonctions T et la nécessité de contraindre la position de ses demières, pour le cas 3D. (a) Un modèle 3D composé initialement de deux hexaèdres adjacents Vol_0 et Vol_1 , cousus ensemble par β_3 . Vol_0 a été subdivisé en huit sous-volumes Vol_{0n} créant quatre jonctions T frontalières : s_1, s_3, s_5 et s_7 et une jonction T centrée : s_8 . (b) L'affaissement de la jonction T frontalière s_7 et la jonction T centrée s_8 dans Vol_1 . Afin de corriger ce comportement non souhaité, s_7 est projeté orthogonalement sur l'arête formée des sommets s_0 et s_6 , donnant le point p_1 . La position de s_7 est affectée à celle de p_1 . De plus, s_8 est projeté orthogonalement sur la plan formé des sommets : s_0, s_2, s_4 et s_6 , donnant le point p . La position de s_8 est affectée à celle de p . Les traits violets représentent les liaisons β_3 . Pour des question de lisibilité, seulement une partie des brins incidents à Vol_1 est représentée.

la boucle de simulation, chaque jonction T frontalière et bivalente sera ainsi contrainte sur l'arête composée de ses deux extrémités de projection, et chaque jonction T centrée sera quant à elle, projetée sur le plan formé de ses quatre extrémités de projection.

La figure 4.27 illustre le comportement erroné induit par la présence de jonctions T frontalières et centrées et la nécessité de contraindre ces jonctions T. En effet, la position de la jonction T frontalière s_7 doit ainsi être contrainte sur p_1 . p_1 correspond à la projection orthogonale de s_7 sur l'arête formée de ses deux extrémités de projection : s_0 et s_6 . De plus, la position de la jonction T centrée s_8 , doit quant à elle être contrainte sur p . p est défini comme étant la projection orthogonale de s_8 sur le plan composé ses quatre extrémités de projection : s_0 , s_2 , s_4 et s_6 .

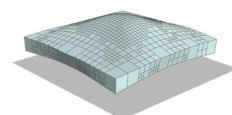
- Répartition de la force des jonctions T** Comme explicité pour le cas 2D, la somme des forces appliquée à chaque jonction T, va être distribuée au niveau des particules constituant son voisinage. Concrètement, pour chaque jonction T j , nous parcourons l'ensemble de ses volumes incidents. Nous stockons alors, parmi ces derniers, ceux étant conformes relativement à j . Puis, pour chacun de ces volumes sélectionnés, nous accumulons au niveau de leurs sommets hiérarchiques autres que j , la force appliquée à j divisée par sept fois le nombre de volumes sélectionnés. En effet, chaque volume sélectionné possède huit sommets hiérarchiques dont j que nous ne comptons pas.

Dans l'exemple présenté dans la figure 4.27, la jonction T centrée s_8 va distribuer la force qui lui est appliquée, aux sommets hiérarchiques des quatre sous-volumes Vol_{0n} qui lui sont incidents. Sa force va donc être répartie sur vingt-huit particules (sept particules pour quatre volumes). De plus, la jonction T frontalière s_7 va répartir sa force sur les quatorze sommets hiérarchiques des deux sous-volumes Vol_{0n} qui lui sont incidents. Dans les deux cas, Vol_1 ne rentre pas dans l'équation puisqu'il est non conforme relativement à s_7 et s_8 .

Mettre à jour les extrémités de projection des jonctions T à partir d'une jonction T frontalière Nous explicitons ici l'algorithme 14 en nous appuyant sur un exemple concret présenté dans les figures 4.24, 4.25 et 4.26. Nous illustrons la nécessité de mettre à jour les extrémités de projection des jonctions T, avec la jonction T frontalière s_{15} et la jonction T centrée s_9 . La jonction T frontalière s_{15} avait pour extrémités de projection les sommets s_{10} et s_{14} , et devrait avoir après la subdivision de Vol_0 : s_{10} et s_2 . La jonction T centrée s_9 , qui avait pour extrémités de projection : s_{10} , s_{12} , s_{13} et s_{14} , et devrait avoir après la subdivision de Vol_0 : s_{10} , s_{11} , s_4 et s_2 .

Les figures 4.24 et 4.25 montrent la transformation de la jonction T bivalente s_3 en une jonction T uniquement frontalière. Les extrémités de projection de s_3 étaient s_2 et s_4 et correspondaient aux extrémités de l'arête hiérarchique de Vol_{1100} . Après la subdivision de Vol_0 , les extrémités de projection restent les mêmes. Par contre, à partir de s_3 , nous allons mettre à jour les jonctions T centrées incluses dans les faces hiérarchiques incidentes à s_3 et aux sous-volumes de Vol_0 .

Pour cela, l'algorithme 14 prend en paramètre un brin partant de s_3 . s_3 étant une jonction T uniquement frontalière, nous parcourons l'ensemble de ses brins incidents (représentés en pointillés dans la figure 4.26). Puis, parmi ces brins, nous en récupérons un qui compose l'arête hiérarchique d'un volume ayant le niveau de subdivision le plus faible. Plus précisément, il faut que ce brin ait pour deuxième extrémité un sommet normal ou une jonction T



frontalière. Vol_{0001} , Vol_{0101} et Vol_{1100} sont les volumes incidents à s_3 ayant le niveau de subdivision le plus faible. Nous faisons l'hypothèse que le brin b_0 est choisi. Nous récupérons ensuite les extrémités de l'arête hiérarchique incluant s_3 , qui sont s_2 (sommet partant du brin prochain-brin-hiérarchique-face(b_0)) et s_4 (brin partant de PBHF(PBHF(PBHF(b_2))), avec PBHF : l'algorithme 5). Ces dernières sont affectées comme extrémités de projection de s_3 . Notons que cette mise à jour n'est pas nécessaire pour une jonction T anciennement bivalente, mais l'est pour une jonction T frontalière, comme par exemple pour s_{15} . Puis, dans un deuxième temps, l'algorithme met à jour les jonctions T centrées incluses dans les faces hiérarchiques incidentes à s_3 et aux nouveaux sous-volumes de Vol_0 . Pour cela, nous récupérons à partir de b_0 , les faces incidentes à s_3 et aux nouveaux sous-volumes, qui sont : F_{003} , F_{02} et F_a . Pour chacune de ses faces, nous récupérons ses quatre sommets hiérarchiques ou ceux de la face hiérarchique qu'elle compose. Nous appelons ensuite l'algorithme 16, avec comme paramètres b_0 et ces quatre sommets hiérarchiques constituant les extrémités de projection, avec lesquelles les jonctions T centrées incluses dans la face hiérarchique considérés, doivent être mises à jour.

Algorithme 14 : Mettre à jour les extrémités de projection des jonctions T à partir d'une jonction T frontalière

Données : b : un brin partant de la jonction T j et incident à une face hiérarchique F .
L'appel à l'algorithme 5 sera abrégé par « PBHF » (Prochain Brin Hiérarchique Face)

Résultat : Les extrémités de projection de j et des jonctions T centrées incluses dans les faces hiérarchiques incidentes à j sont mises à jour.

$\mathcal{B} \leftarrow$ un conteneur de brins;

Si j est une jonction T uniquement frontalière **Alors**

// Récupérer un brin sur la bordure de la face hiérarchique
incidente à j

Pour tous Les brins b_a incidents à j **faire**

Si $\beta_1(b_a)$ (deuxième extrémité de b_a , la première étant j) ne part pas d'une
jonction T centrée et $l_{Vol(b_a)} < \min$ **Alors**

$\min \leftarrow l_{Vol(b_a)}$;

$b_T \leftarrow b_a$;

// Mise à jour des extrémités de projection de j
extremites-projection(j , PBHF(b_T), PBHF(PBHF(PBHF(b_T))));

Pour tous Les brins b_c appartenant à l'arête incidente à b_T **faire**

Si b_c n'est pas 3-libre et que $l_{Vol(b_c)} > l_{Vol(\beta_3(b_c))}$ **Alors**

$b_T \leftarrow \beta_3(b_c)$;

// Récupérer les quatre sommets hiérarchiques de la face
hiérarchique incidente à b_T

Pour i allant de 1 à 4 **faire**

ajouter-au-conteneur(\mathcal{B}, b_T);

$b_T \leftarrow$ PBHF(b_T);

// Mettre à jour les jonctions T centrés incluses dans la face
hiérarchique incidente à b_T

mise-a-jour-extremites-projection-centree(b_T, \mathcal{B});

Algorithme 15 : Mettre à jour les extrémités de projection d'une jonction T

Données : b : un brin partant de la jonction T j .
extremites : les extrémités de projection à affecter à j .
Résultat : Les extrémités de projection de j sont mises à jour avec *extremites*.
affecter-extremites-projection(j , *extremites*);
Pour tous $p \in \textit{extremites}$ **faire**
 └─ incrementer-nombre-jonctions-T-est-extremite(p);

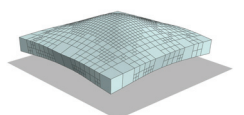
Mettre à jour les extrémités de projection des jonctions T centrées incluses dans une face hiérarchique Nous détaillons ici l'algorithme 16 en nous appuyant sur un exemple concret présenté dans les figures 4.24, 4.25 et 4.26. Cet exemple fait suite à celui présenté pour expliciter l'algorithme 14. Nous rappelons que le brin b_0 partant de s_3 , est incident aux faces : $F_{003|Vol_{0001}}$, $F_{02|Vol_{0101}}$ et F_a . Nous appliquons l'algorithme 16 sur un brin partant de s_3 et incident à $F_{003|Vol_{0001}}$, ainsi que les quatre sommets hiérarchiques de $F_{00|Vol_{0001}}$: s_2 , s_{10} , s_{11} et s_4 . L'algorithme parcourt la bordure de $F_{003|Vol_{0001}}$ à la recherche de jonctions T centrées. Il trouve la jonction T centrée s_9 et met à jour ses extrémités de projection qui ne seront plus : s_{10} , s_{12} , s_{13} et s_{14} , mais les quatre sommets hiérarchiques de $F_{00|Vol_{0001}}$: s_{10} , s_{11} , s_4 et s_2 . L'algorithme est ensuite rappelé pour chaque brin incident à s_9 .

Algorithme 16 : Mettre à jour les extrémités de projection des jonctions T centrées incluses dans une face hiérarchique

Données : b : un brin d'une face hiérarchique F .
extremites : les extrémités de projection à affecter aux jonctions T centrées.
Résultat : Les extrémités de projection des jonctions T centrées incluses dans F , sont mises à jour avec *extremites*.
 $b_0 \leftarrow b$;
Répéter
 └─ $p \leftarrow$ la particule incidente à b_0 ;
 Si p est une jonction T centrée **Alors**
 └─ extremites-projection(p , *extremities*);
 Pour tous Les faces incidentes à p **faire**
 └─ mise-a-jour-extremites-projection-centree(p , *extremites*);
 $b_0 \leftarrow \beta_1(b_0)$;
jusqu'à $b = b_0$;

4.5 Découpe d'un objet 2D localement raffiné

La partie 2.3.3 et la section 3.4 explicitaient le processus de découpe respectivement d'un point de vue topologique et d'un point de vue physique. Plus précisément, nous présentions dans la partie 2.3.3, l'opération de découpe des cartes combinatoires et dans la section 3.4, le processus de découpe de notre modèle unifié, comprenant la découpe topologique de la LCC et la mise à jour des attributs relatifs au modèle physique, mais sans gestion du raffinement local. C'est pour cette raison que dans cette partie, nous revenons sur ce processus de découpe pour l'adapter à un objet ayant été localement raffiné.



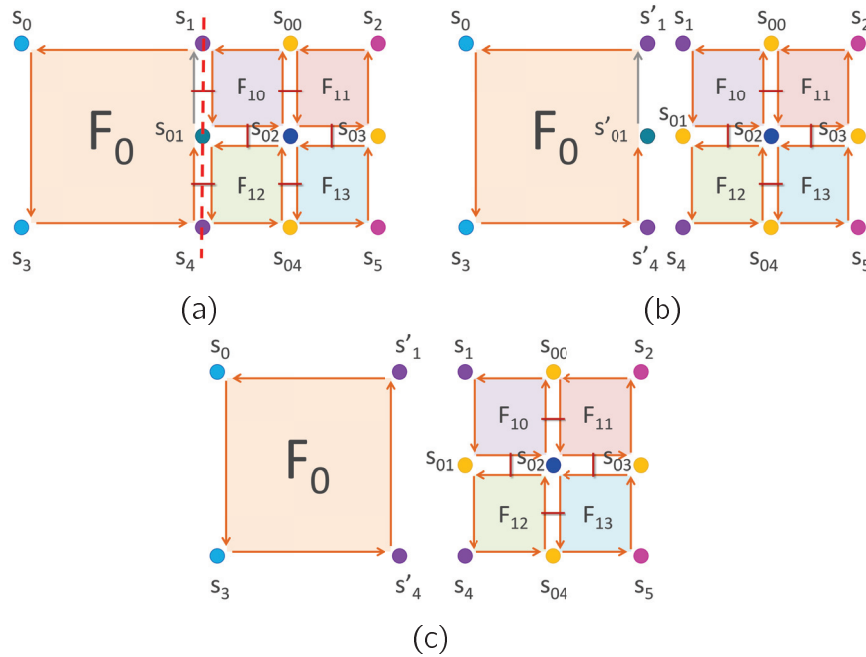


Figure 4.28 – La suppression de jonction T suite au processus de découpe, pour le cas 2D. (a) Le modèle de la figure 4.14. Le trait rouge en pointillé correspond au chemin de découpe. (b) Le modèle de (a) après la découpe selon l'arête hiérarchique droite de F_0 : (s_1, s_4) . Les sommets s_1 , s_{01} et s_4 ont été scindés ainsi que les arêtes (s_1, s_{01}) et (s_{01}, s_4) . Les liaisons β_2 ont ainsi été supprimées entre les éléments $F_0 - F_{10}$ et $F_0 - F_{12}$. (c) Le modèle de (b) après la suppression du sommet s'_{01} , qui n'a plus lieu d'exister par rapport à l'élément F_0 .

Nous rappelons que le processus de découpe prend en paramètre un brin (cf. la section 3.4). En 2D, si ce brin n'est pas 2-libre, alors il est 2-découpu. L'arête incidente est ainsi scindée en deux, et les deux faces qui étaient initialement incidentes à cette arête, ne sont plus adjacentes. Enfin, en 2D, lors de la suppression de liaisons β_2 , des sommets ou des arêtes peuvent être scindés. La scission d'un sommet (resp. d'une arête) e en le sommet (resp. l'arête) e' , entraîne la copie de l'attribut stocké par e , au niveau de e' . e' est donc associé à un attribut renfermant les mêmes informations que celles relatives à l'attribut de e . Notons que ces informations peuvent alors nécessiter une mise à jour des informations physiques. Cette mise à jour inclut celle de la masse des sommets qui ont été scindés, en procédant de la même manière que pour l'initialisation (cf. la sous-section 3.2.1), ainsi que celle de l'ensemble des ressorts ayant pour extrémité un sommet qui a été scindé comme expliqué dans la section 3.4.

Or, le raffinement local du modèle a pu engendrer des jonctions T . Le processus de découpe doit donc gérer ces sommets particuliers. Plus précisément, le processus de découpe peut être amené à mettre à jour les extrémités de projection d'une jonction T , à ôter la propriété de jonction T à un sommet ou encore à supprimer un sommet de type jonction T . Nous explicitons ci-dessous ces trois cas de figure.

La figure 4.28(a) présente un modèle 2D initialement constitué de deux quadrilatères : F_0 et F_1 . Le quadrilatère F_1 a été subdivisé en quatre sous-quadrilatères : F_{10} , F_{11} , F_{12} et F_{13} . Lors de la subdivision de F_1 , la jonction T s_{01} a été créée. La figure 4.28(b) montre le modèle de (a) après un processus de découpe effectué sur un des deux brins incidents à l'arête (s_1, s_{01}) , suivi d'un deuxième processus de découpe appliqué sur un des deux brins incidents à l'arête (s_{01}, s_4) . Les sommets s_1 , s_{01} et s_4 ont été respectivement scindés en les sommets s'_1 , s'_{01} et s'_4 . Les arêtes (s_1, s_{01}) et (s_{01}, s_4) sont respectivement scindées en (s'_1, s'_{01}) et

(s'_{01}, s'_{4}) . Enfin, les quadrilatères incidents aux arêtes en question sont alors 2-décousus deux à deux, en supprimant les liaisons β_2 . F_0 n'est alors plus lié ni avec F_{10} , ni avec F_{12} .

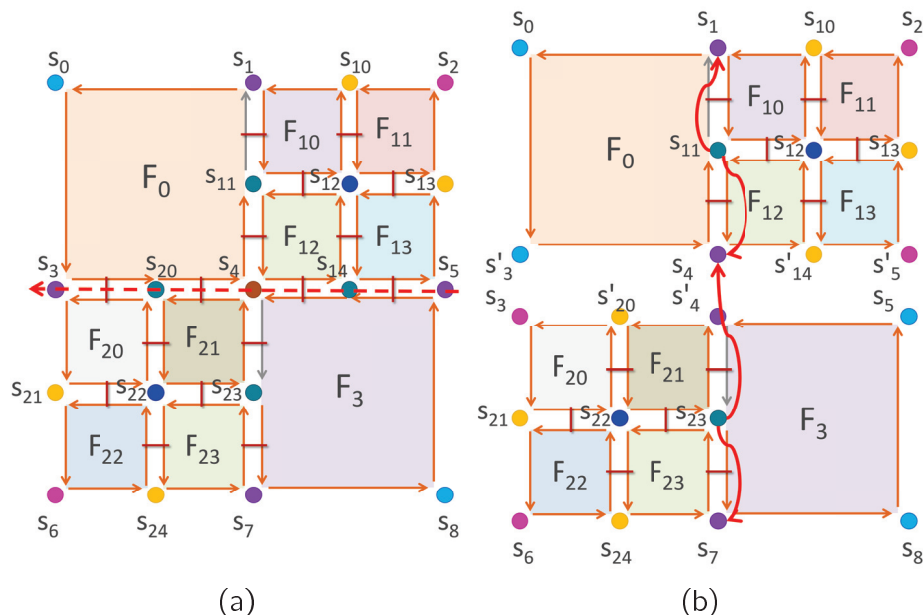


Figure 4.29 – La nécessité de mettre à jour les extrémités de projection des jonctions T , pour le cas 2D. (a) Un modèle 2D composé initialement de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 . F_1 et F_2 ont été subdivisés. Le trait rouge en pointillés correspond au chemin de découpe. (b) Le modèle de (a) après la découpe selon les arêtes (s_5, s_{14}) , (s_{14}, s_4) , (s_4, s_{20}) et (s_{20}, s_3) . Les sommets s_5 , s_{14} , s_4 , s_{20} et s_3 ont été scindés ainsi que les arêtes (s_5, s_{14}) , (s_{14}, s_4) , (s_4, s_{20}) et (s_{20}, s_3) . Les jonctions T : s_{20} et s_{14} ont été supprimées. Les flèches rouges partant de s_{11} et de s_{23} indiquent leurs extrémités de projection. s_{23} pointe de manière erronée sur s_4 . Son extrémité de projection s_4 doit donc être mise à jour pour s'_4 .

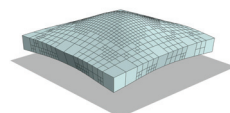
4.5.1 Mise à jour des extrémités de projection

Nous rappelons que la Particule associée à une jonction T stocke les extrémités de projection de cette dernière. Or, si l'une de ses extrémités de projection est scindée, alors nous devons mettre à jour ses extrémités de projection.

La figure 4.29 illustre la nécessité de mettre à jour les extrémités de projection des jonctions T si au moins l'une de ces dernières a été scindée. La figure 4.29(a) montre un modèle 2D initialement composé de quatre quadrilatères : F_0 , F_1 , F_2 et F_3 , cousus ensemble par β_2 , où F_1 et F_2 ont été subdivisés. La figure 4.29(b) représente le modèle de (a) après la découpe selon les arêtes (s_5, s_{14}) , (s_{14}, s_4) , (s_4, s_{20}) et (s_{20}, s_3) . Les sommets s_5 , s_{14} , s_4 , s_{20} et s_3 ont été scindés ainsi que les arêtes (s_5, s_{14}) , (s_{14}, s_4) , (s_4, s_{20}) et (s_{20}, s_3) . Dans la figure 4.29(a), les jonctions T : s_{11} et s_{23} ont respectivement comme extrémités de projection, $\{s_1, s_4\}$ et $\{s_4, s_7\}$. Or, pendant les processus de découpe, le sommet s_4 a été scindé en s'_4 . s_{11} et s_{23} doivent donc mettre à jour leurs extrémités de projection. En effet, s_{23} a comme extrémité de projection s_4 et s_7 , alors qu'elle devrait avoir s'_4 et s_7 .

Pour faciliter la mise à jour des extrémités de projection d'une jonction T , deux informations sont rajoutées au niveau de la Particule :

- Une Particule stocke l'information indiquant si elle est une extrémité de projection pour une jonction T .



- Une Particule stocke l'information du nombre de jonctions T pour lesquelles elle est une extrémité de projection. Lorsqu'un sommet de type jonction T est supprimé ou n'est plus de type jonction T, alors ses extrémités de projection sont décrémentées. Si le nombre de jonctions T pour lesquelles elle est une extrémité de projection devient nul alors elle n'est plus considérée comme une extrémité de projection.

Nous revenons maintenant à notre processus de mise à jour des extrémités de projection d'une jonction T. Nous posons le sommet s comme étant une extrémité de projection. Si s vient à être scindé alors toutes les jonctions T ayant ce dernier comme extrémité de projection mettent à jour leurs extrémités de projection. Pour cela, nous procédons de la manière suivante, lorsqu'un sommet s est scindé, nous testons s'il est une extrémité de projection. Si c'est le cas, nous parcourons alors tous les sommets appartenant aux faces incidentes à s . Lors de ce parcours, pour chaque sommet de type jonction T rencontré, nous testons à la fin du processus de découpe, s'il est encore de type jonction T ou s'il doit être supprimé. Nous détaillerons ces cas dans le prochain point. Si le sommet reste de type jonction T (comme s_{11} et s_{23} dans la figure 4.29), alors ses extrémités sont mises à jour. Pour cela, nous récupérons le brin incident à cette jonction T qui n'est pas hiérarchique (dans notre exemple, ce brin est incident à F_0 pour s_{11} et incident à F_3 pour s_{23}). Ce brin ainsi obtenu, nous permet de récupérer les deux extrémités de l'arête hiérarchique associée au volume ayant le niveau de subdivision le plus faible, contenant notre jonction T (dans notre exemple, $\{s_1, s_4\}$ pour s_{11} et $\{s'_4, s_7\}$ pour s_{23}).

4.5.2 Enlever la propriété de jonction T à un sommet

Lorsqu'un sommet s de type jonction T est scindé, l'un des deux sommets résultant n'est plus une jonction T et l'autre est supprimé. Pour chaque sommet s_a résultant d'un scission, nous parcourons l'ensemble de ses brins incidents. Si l'un d'entre eux n'est pas hiérarchique, alors s_a reste une jonction T, sinon s_a n'est plus une jonction T. Si s_a n'est plus une jonction T, pour chacune de ses deux extrémités de projection, nous décrémentons leur nombre de jonctions T pour lesquelles elles sont une extrémité de projection. Enfin, nous affectons sa vitesse et son accélération avec la moyenne de celles de ses extrémités de projection.

Dans la figure 4.29, s_{20} est un sommet de type jonction T. Lors du processus de découpe, s_{20} est scindé en s'_{20} . Les brins incidents à s'_{20} sont alors tous hiérarchiques. s'_{20} n'est pas une jonction T au regard des éléments F_{20} et F_{21} . Elle perd ainsi sa propriété de jonction T qui lui avait été attribuée par copie de s_{10} .

4.5.3 Supprimer une jonction T

Suite à la scission d'un sommet s de type jonction T, nous testons si les sommets résultants sont supprimables. Nous rappelons qu'un sommet est supprimable s'il est de degré inférieur ou égal à deux (cf. la partie 2.3.3). Sa suppression entraîne la fusion des deux arêtes qui lui étaient incidentes en une seule même arête. Enfin, nous mettons à jour le ressort associé à la nouvelle arête : ses extrémités, son niveau de subdivision, sa raideur ainsi que sa longueur initiale. Son niveau de subdivision a pour valeur la moitié du niveau de subdivision d'une des deux arêtes initiales dont il est issu.

La figure 4.29 illustre cette explication. Le sommet s_{20} de type jonction T est scindé lors du processus de découpe en s'_{20} . Nous avons vu précédemment que le sommet s'_{20} perd sa qualité de jonction T. Ce dernier n'est pas supprimable en raison de son degré 3. Par contre, s_{20} , de degré 2, est quant à lui supprimé. En effet, s_{20} était issu de la subdivision de F_2 et

n'existait pas au regard de F_0 . s_{20} n'a donc plus de raison d'exister. Sa suppression entraîne la fusion de ses deux arêtes incidentes : (s'_3, s_{20}) et (s_{20}, s_4) en l'arête (s'_3, s_4) .

4.6 Découpe d'un objet 3D localement raffiné

Nous allons maintenant présenter, comme pour le cas 2D, comment adapter le processus de découpe pour qu'il gère un modèle composé de volumes n'ayant pas le même niveau de subdivision. Nous rappelons que la partie 2.3.3 et la section 3.4 explicitent le processus de découpe respectivement d'un point de vue topologique et physique.

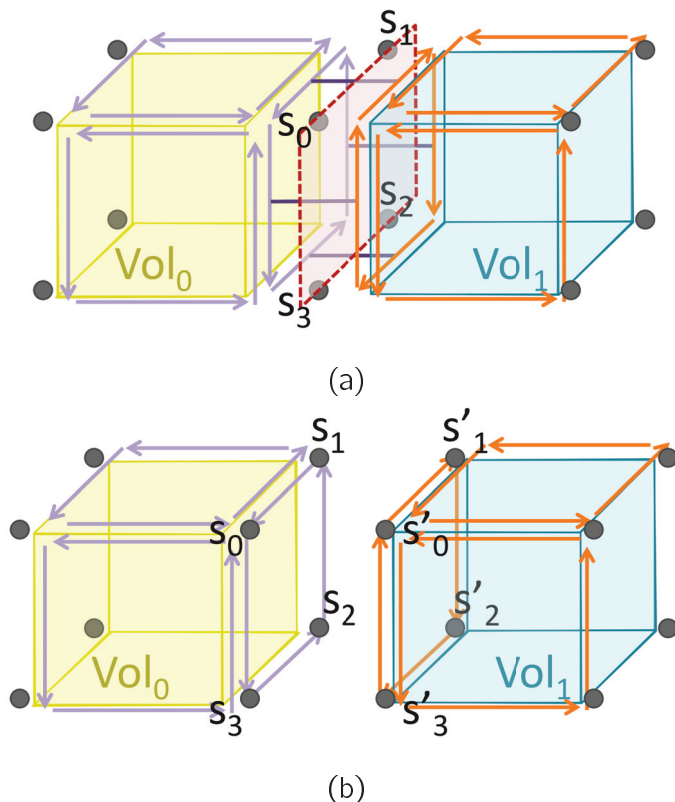
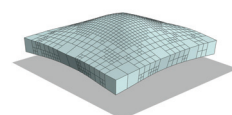


Figure 4.30 – Une 3-découpe avec la représentation d'une partie des brins. (a) Un modèle 3D composé de deux hexaèdres adjacents Vol_0 et Vol_1 , cousus ensemble par β_3 . Les pointillés rouges délimitent une zone représentant un plan de découpe. (b) Le modèle de (a) suite au processus de découpe effectué selon le plan de découpe. Les sommets composant la face commune à Vol_0 et Vol_1 sont scindés, tout comme ses arêtes. Vol_0 et Vol_1 sont ainsi 3-décousus selon la face définie par les sommets s_0, s_1, s_2 et s_3 . Les traits violets représentent les liaisons β_3 . Pour une question de lisibilité, les liaisons β_2 ainsi qu'une partie des brins incidents aux deux volumes ne sont pas représentés.

Nous rappelons que le processus de découpe prend en paramètre un brin (cf. la section 3.4). En 3D, si ce brin n'est pas 3-libre, alors il est 3-décousu ainsi que les trois autres brins composant la face incidente à ce brin, qui est alors scindée en deux. Les deux volumes qui étaient initialement incidents à cette face, ne sont plus adjacents. Enfin, en 3D, lors de la suppression de liaisons β_3 , des sommets ou des arêtes peuvent être scindés. La scission d'un sommet (resp. d'une arête) e en le sommet (resp. l'arête) e' , entraîne la copie de l'attribut stocké par e , au niveau de e' . e' est donc associé à un attribut renfermant les mêmes informations que celles relatives à l'attribut de e . Notons que ces informations peuvent alors



nécessiter une mise à jour des informations physiques. Cette mise à jour inclut celle de la masse des sommets qui ont été scindés, en procédant de la même manière que pour l'initialisation (cf. la sous-section 3.2.1), ainsi que celle de l'ensemble des ressorts ayant pour extrémité un sommet qui a été scindé comme expliqué dans la section 3.4.

La figure 4.30(a) montre un modèle 3D constitué de deux hexaèdres : Vol_0 et Vol_1 , cousus ensemble par β_3 . La figure 4.30(b) représente le modèle de (a) après un processus de découpe réalisé selon la face F définie par les sommets : s_0, s_1, s_2 et s_3 . Le sommet s_0 (resp. s_1, s_2, s_3) qui était initialement incident à Vol_0 et Vol_1 , a été scindé en deux sommets : s_0 et s'_0 (resp. s_1 et s'_1, s_2 et s'_2, s_3 et s'_3). s_0 (resp. s_1, s_2, s_3) est maintenant incident à Vol_0 et s'_0 (resp. s'_1, s'_2, s'_3) est incident à Vol_1 . La scission est également appliquée pour les quatre arêtes de F . De plus, les liaisons β_3 qui reliaient les deux cycles de brins composant $F : F_{Vol_0}$ et F_{Vol_1} , ont été supprimées.

Or, comme pour le cas 2D, le raffinement local du modèle a pu engendrer des jonctions T. Le processus de découpe doit donc gérer ces sommets particuliers. Plus précisément, le processus de découpe peut être amené à mettre à jour les extrémités de projection d'une jonction T, à ôter la propriété de jonction T à un sommet ou encore à supprimer un sommet de type jonction T. Nous explicitons ci-dessous ces trois cas de figure.

Nous précisons que la découpe est effectuée selon la face hiérarchique du volume ayant le niveau de subdivision le plus faible. C'est-à-dire que tous les brins incidents à la face hiérarchique du volume ayant le niveau de subdivision le plus faible, seront 3-décousus.

4.6.1 Mise à jour des extrémités de projection

Scission d'une extrémité de projection

Si l'une des extrémités de projection d'une jonction T est scindée au cours d'un processus de découpe, alors elles doivent être mises à jour. Nous rappelons que pour faciliter cette mise à jour, nous avons rajouté une information, stockée au niveau de la structure de données Particule, indiquant si la particule associée est une extrémité de projection. Elle correspond au nombre de jonctions T pour lesquelles cette dernière est une extrémité de projection.

Nous considérons une particule p comme étant une extrémité de projection. Si cette particule est scindée, alors le nombre de jonctions T pour lesquelles elle est une extrémité de projection, est remis à zéro. Puis, toutes les jonctions T qui avaient p comme extrémité de projection doivent mettre à jour leurs extrémités de projection. Pour cela, nous parcourons tous les sommets appartenant aux volumes incidents à p . Pour chaque sommet s parcouru, si s est une jonction T, alors nous appelons l'algorithme 14 sur un brin incident à s .

La nécessité de la mise à jour des extrémités de projection d'une jonction T lors de la scission de l'une de ses extrémités de projection, est illustrée par les figures 4.31 et 4.32. Dans la figure 4.31, la jonction T centrée s_8 (resp. s_{15}) a pour extrémités de projection : s_0, s_2, s_4 et s_6 (resp. s_2, s_{10}, s_{12} et s_4). La figure 4.32 correspond au modèle de la figure 4.31, suite à la découpe selon le plan correspondant à la zone délimitée par les pointillés en rouge. Suite au processus de découpe, nous constatons que la jonction T centrée s_8 (resp. s_{15}) doit mettre à jour ses extrémités de projection pour les sommets : s_0, s_2, s'_4 et s_6 (resp. s'_2, s_{10}, s_{12} et s_4). Il en va de même pour la jonction T frontalière : s_1 (resp. s_5, s_9, s_{13}), qui avait pour extrémité de projection : $(s_0$ et $s_2)$ (resp. $(s_6$ et $s_4), (s_2$ et $s_{10}), (s_4$ et $s_{12})$). Après le processus de découpe, toutes les extrémités de projection de ces jonctions T frontalières doivent être mises à jour. s_1 (resp. s_5, s_9, s_{13}) a maintenant pour extrémité de projection : $(s_0$ et $s_2)$ (resp. $(s_6$ et $s'_4), (s'_2$ et $s_{10}), (s_4$ et $s_{12})$).

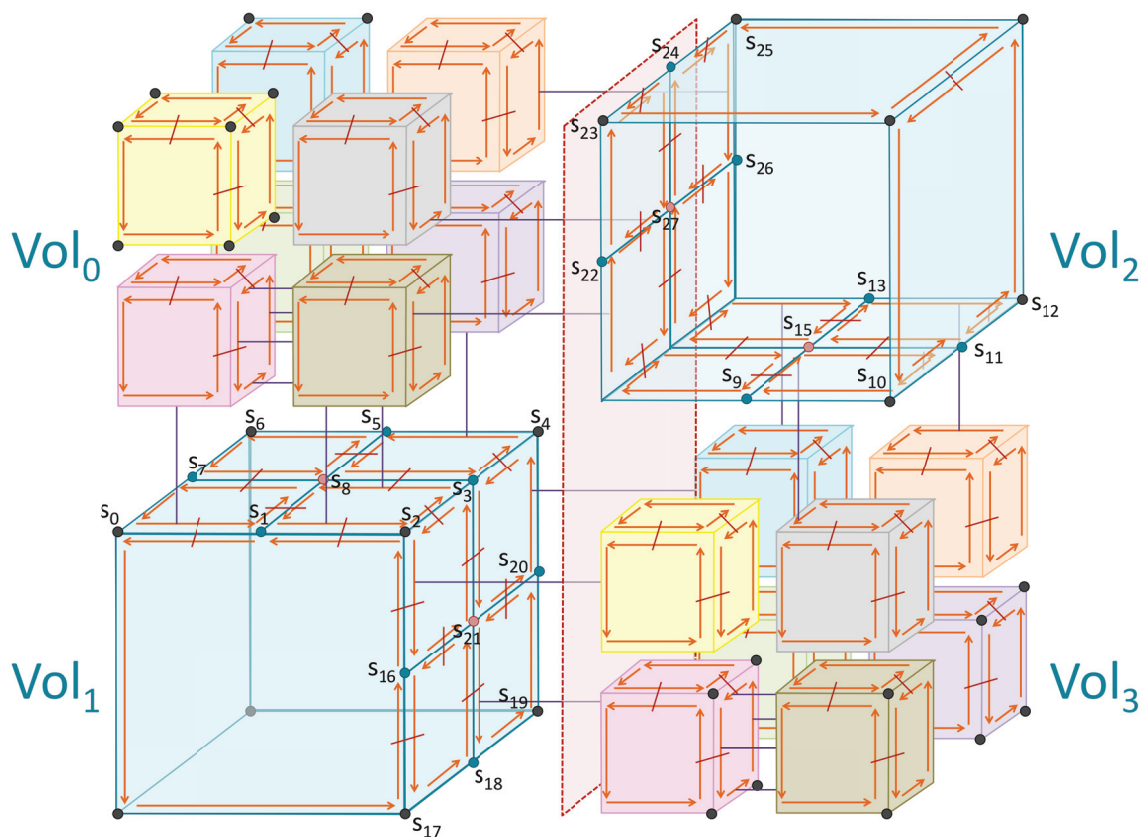
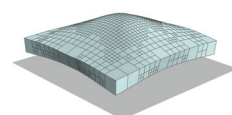


Figure 4.31 – La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D. Un modèle 3D initialement constitué de quatre hexaèdres : Vol_0 , Vol_1 , Vol_2 et Vol_3 , cousus ensemble par β_3 . Vol_0 (resp. Vol_3) a été subdivisé en huit sous-volumes : Vol_{0n} (resp. Vol_{3n}). Le trait rouge en pointillés délimite une zone correspondant à un plan de découpe. Les traits rouges (resp. violets) représentent les liaisons β_2 (resp. β_3). Pour une question de lisibilité, seulement une partie des liaisons β_2 et β_3 est représentée, tout comme une partie de l'ensemble des brins.



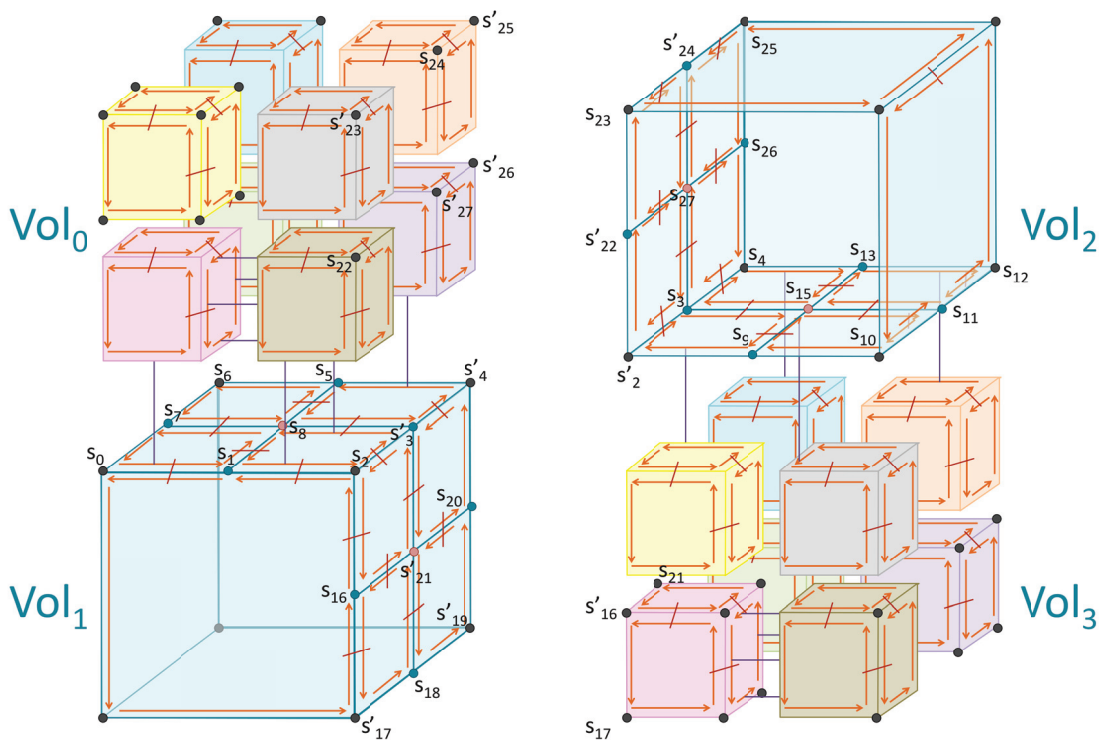


Figure 4.32 – La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D. Le modèle de la figure 4.31 après deux 3-découpures entre les volumes (Vol_0, Vol_2) et (Vol_1, Vol_3). Nous nommons F_{an} (resp. F_{bn}), les quatre faces incidentes à Vol_0 et Vol_2 (resp. Vol_1 et Vol_3). L'ensemble des sommets et arêtes composant les huit faces F_{an} et F_{bn} , ont été scindés. Nous précisons qu'un sommet s_i est scindé en les sommets s_i et s'_i . Enfin, les liaisons β_3 qui liaient deux à deux les volumes 3-découpés ont été supprimées.

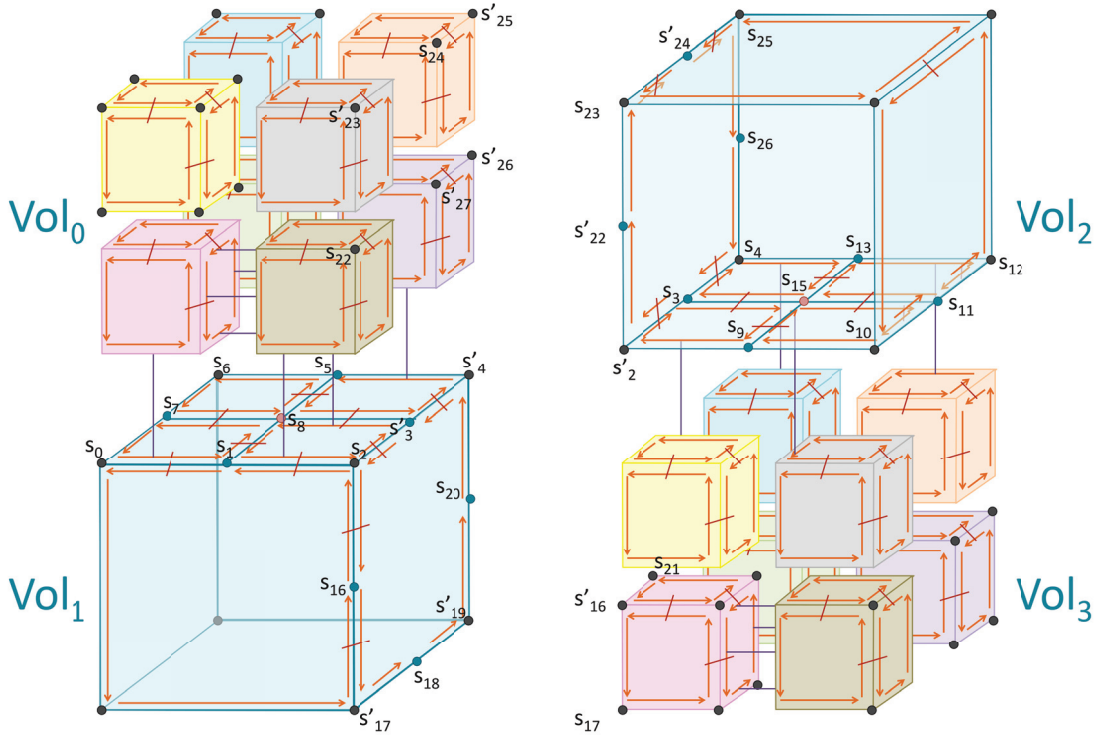


Figure 4.33 – La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D. Le modèle de la figure 4.32 suite à la suppression des arêtes internes à $F_{an|Vol_1}$ et $F_{bn|Vol_2}$. C'est-à-dire l'ensemble des brins incidents à $F_{an|Vol_1}$ (resp. $F_{bn|Vol_2}$), ayant pour extrémité la jonction T centrée s'_{21} (resp. s_{27}).

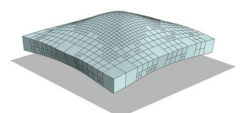
Scission d'une jonction T

La nécessité de la mise à jour des extrémités de projection d'une jonction T lors de sa scission, est illustrée par les figures 4.31 et 4.32. Dans la figure 4.31, la jonction T frontalière s_3 avait pour extrémités de projection : s_2 et s_4 . Après la découpe, ses extrémités de projection sont mises à jour par les sommets : s_2 et s'_4 (cf. la figure 4.32), en appelant l'algorithme 14 sur un brin incident à s_3 .

4.6.2 Enlever la propriété de jonction T à un sommet

Nous considérons qu'une jonction T, notée j , est scindée en deux parties que nous nommons : j_0 et j_1 . Lors de la mise à jour de la masse de j_0 (resp. j_1), nous parcourons un brin par volume incident à j_0 (resp. j_1). Si l'un d'entre eux n'est pas marqué, alors j_0 (resp. j_1) est toujours une jonction T. Sinon, si tous les brins sont marqués alors j_0 (resp. j_1) n'est plus une jonction T.

Dans la figure 4.32, représentant le modèle de la figure 4.31 suite au processus de découpe, les jonctions T frontalières et centrées, incidentes à quatre sous-volumes de Vol_0 : s'_{23} , s_{24} , s'_{25} , s'_{26} , s'_{27} et s_{22} , et incidentes à quatre sous-volumes de Vol_3 : s_{17} , s'_{18} , s_{19} , s'_{20} , s_{21} et s'_{16} , ont chacune d'entre elles, tous leurs brins incidents marqués comme hiérarchiques. Elles perdent leur propriété de jonction T et deviennent donc de simples sommets.



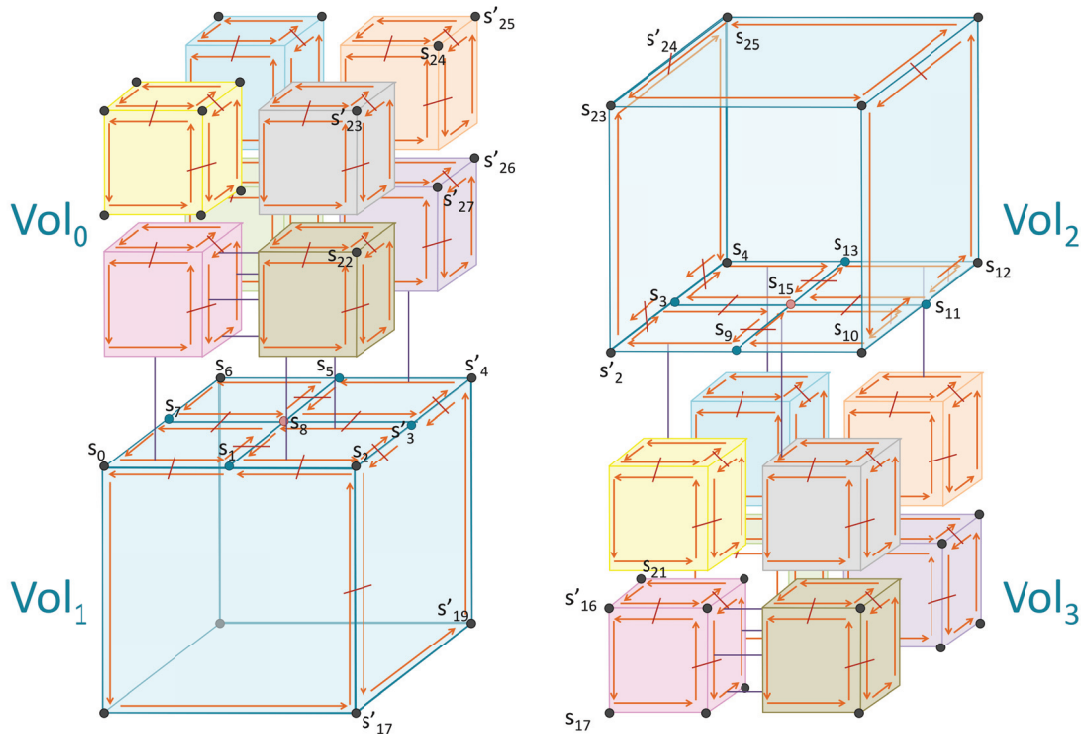


Figure 4.34 – La modification du processus de découpe pour gérer un modèle constitué d'éléments de différents niveaux de subdivision, pour le cas 3D. Le modèle de la figure 4.33, après la suppression des jonctions T frontalières de Vol_1 (resp. Vol_2), ayant un degré inférieur ou égal à 2. C'est le cas des sommets s_{16} , s_{18} et s_{20} (resp. s'_{22} , s'_{24} et s_{26}) pour le volume Vol_1 (resp. Vol_2).

4.6.3 Supprimer une jonction T

Nous considérons une arête a ayant pour extrémité une jonction T centrée ou bivalente. Si a vient à être scindée en a_1 et a_2 , alors l'une des deux arêtes résultantes sera supprimée quand son degré sera égal à 2. En effet, toutes les arêtes internes à une face hiérarchique d'un volume ayant le niveau de subdivision le plus faible, sont supprimés suite à une découpe selon cette face.

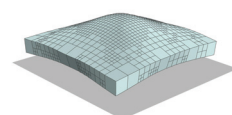
Dans l'exemple présenté dans les figures 4.31, 4.32 et 4.33, les arêtes scindées : (s'_3, s'_{21}) , (s_{16}, s'_{21}) , (s_{18}, s'_{21}) , (s_{20}, s'_{21}) incidentes à Vol_1 sont supprimées. Les arêtes scindées : (s'_{22}, s_{27}) , (s'_{24}, s_{27}) , (s_{26}, s_{27}) et (s_3, s_{27}) incidentes à Vol_2 sont également supprimées.

Lorsqu'une jonction T centrée ou bivalente est scindée, la partie résultante incidente au volume de niveau de subdivision le plus faible est forcément supprimée. Dans notre exemple, c'est le cas des jonctions T centrées s'_{21} et s_{27} .

Enfin, les jonctions T frontalières qui ont été scindées et qui ont un degré égal à 2, sont supprimées. Pour chaque jonction T frontalière j supprimée, nous décrétons ses extrémités de projection. Les deux arêtes incidentes à j , que l'on nommera a_1 et a_2 sont fusionnées en la même arête a . Nous devons ensuite mettre à jour les extrémités du ressort qui est associé à a , son niveau de subdivision correspondant à $\frac{l_{a_1}}{2}$ ou $\frac{l_{a_2}}{2}$, et sa raideur. Dans notre exemple, c'est le cas des sommets s_{16} , s_{18} , s_{20} , s'_{22} , s'_{24} et s_{26} (cf. la figure 4.34).

4.7 Conclusion

Nous avons présenté dans ce chapitre, les modifications apportées à notre modèle unifié afin de gérer le raffinement local des quadrilatères en 2D ou des hexaèdres en 3D constituant le maillage. Pour cela, nous avons tout d'abord défini de nouvelles notions que sont le niveau de subdivision, l'élément et le brin hiérarchique, ou encore les jonctions T avec les éléments non conformes. Une fois les nouvelles informations à ajouter au modèle présentées, nous avons introduit les algorithmes de base permettant le parcours de notre modèle localement raffiné. Pour le cas 2D, nous avons ensuite détaillé l'opération topologique de la subdivision d'un quadrilatère en quatre sous-quadrilatères, et pour le cas 3D, celle de la subdivision d'un hexaèdre en huit sous-hexaèdres. Puis, nous avons expliqué le processus de mise à jour des informations physiques nécessaires à la simulation physique, embarquées au niveau du modèle topologique. Enfin, nous avons détaillé les modifications à apporter au processus de découpe pour qu'il puisse gérer des modèles localement raffinés, avec principalement, la gestion des jonctions T.



Chapitre 5

«...Maman disait toujours, « la vie, c'est comme une boîte de chocolats : on ne sait jamais sur quoi on va tomber. » »
(Forrest Gump)

Résultats préliminaires

Sommaire

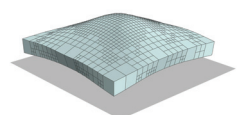
5.1 Découpe	173
5.2 Découpe et raffinement	176
5.3 Méthode adaptative	177
5.4 Conclusion	183

Dans ce manuscrit, nous avons tout d'abord décrit notre modèle unifié, son initialisation, les opérations de découpe, puis les différentes notions et concepts permettant de gérer son raffinement local avec les opérations de raffinement d'un quadrilatère et d'un hexaèdre inscrit dans un modèle localement raffiné, enfin l'adaptation de l'opération de découpe pour un modèle localement raffiné.

Nous présentons dans cette partie quelques résultats montrant la capacité de notre modèle unifié à gérer des changements topologiques, comme celui de la découpe ou du raffinement local. L'opération de raffinement local de quadrilatères et d'hexaèdres inscrits dans un maillage surfacique ou volumique, nous permet de proposer une méthode adaptative, où au cours de la simulation, chaque élément est testé et est raffiné si le critère donné est satisfait.

5.1 Découpe

Le processus de découpe commence par la sélection de l'ensemble des sommets que nous souhaitons découdre. En 2D, les arêtes ayant leurs deux extrémités sélectionnées sont alors scindées en deux parties. Les faces incidentes aux arêtes en question sont alors décousues deux à deux en supprimant les liaisons β_2 . En 3D, toutes les faces du modèle qui sont incidentes à deux volumes et dont les sommets les composant sont sélectionnés, sont scindées. Deux volumes incidents à une de ces faces sont alors décousus selon cette dernière. Les figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 montrent la capacité de notre modèle unifié à gérer le changement topologique de la découpe.



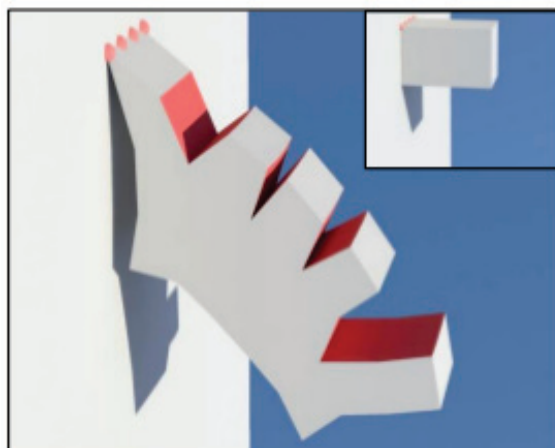


Figure 5.1 – Résultats illustrant le processus de découpe. En haut à droite : Un modèle 3D composé de $5 \times 3 \times 3$ cubes. Quatre de ses particules, situées en haut à gauche sont fixées. Ce modèle initial se déforme ensuite sous l'effet de la gravité et dix de ses faces sont 3-décousues.

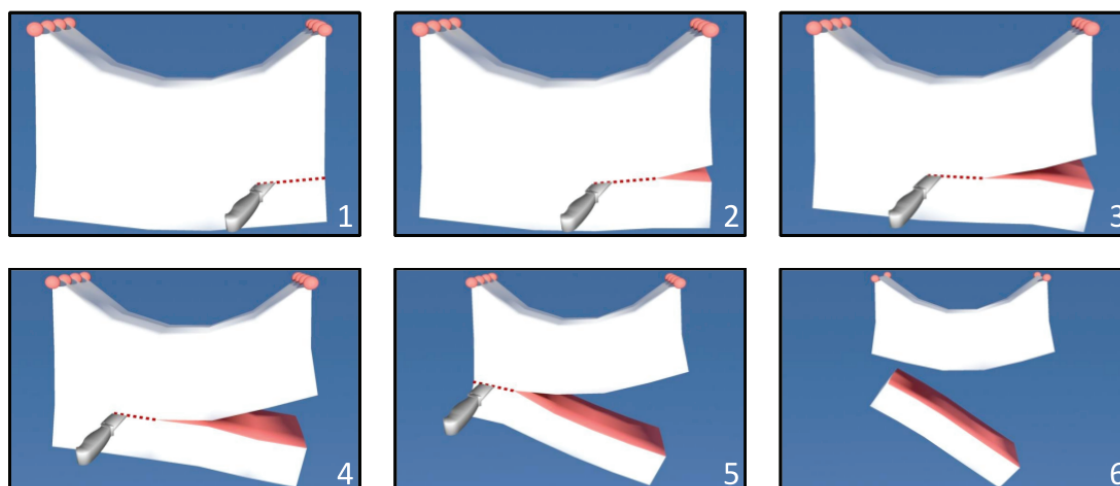


Figure 5.2 – Résultats illustrant le processus de découpe interactif. Un modèle 3D composé de $5 \times 3 \times 3$ cubes. Huit de ses particules, situées en haut à gauche et en haut à droite sont fixées. Ce modèle initial se déforme ensuite sous l'effet de la gravité, tout en étant progressivement découpé. Le schéma montre six états différents du modèle.

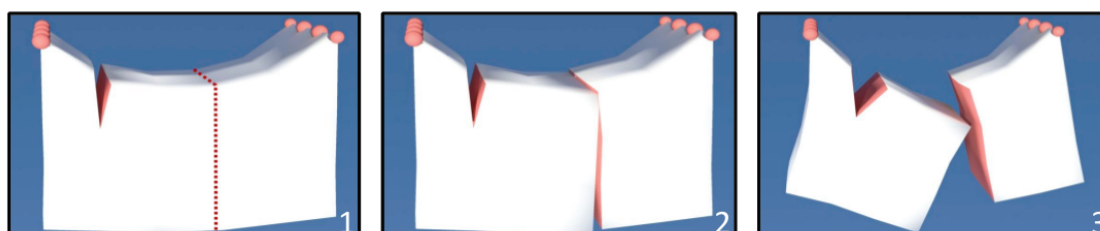


Figure 5.3 – Résultats illustrant le processus de découpe. Un modèle 3D composé de $5 \times 3 \times 3$ cubes. Huit de ses particules, situées en haut à gauche et en haut à droite sont fixées. Ce modèle initial se déforme ensuite sous l'effet de la gravité, tout en étant découpé. Trois de ses faces sont d'abord 3-décousues, suivies de neuf de ses autres faces.

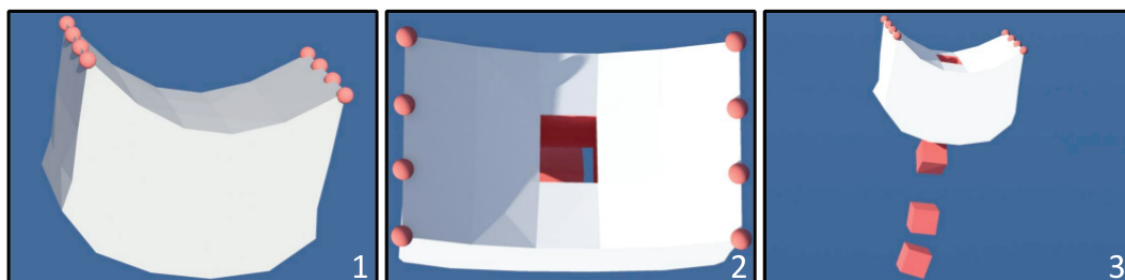


Figure 5.4 – Résultats illustrant le processus de découpe. Un modèle 3D composé de $5 \times 3 \times 3$ cubes. Huit de ses particules, situées en haut à gauche et en haut à droite sont fixées. Ce modèle initial se déforme ensuite sous l'effet de la gravité, tout en étant percé. Quatorze de ses faces ont été 3-décousues

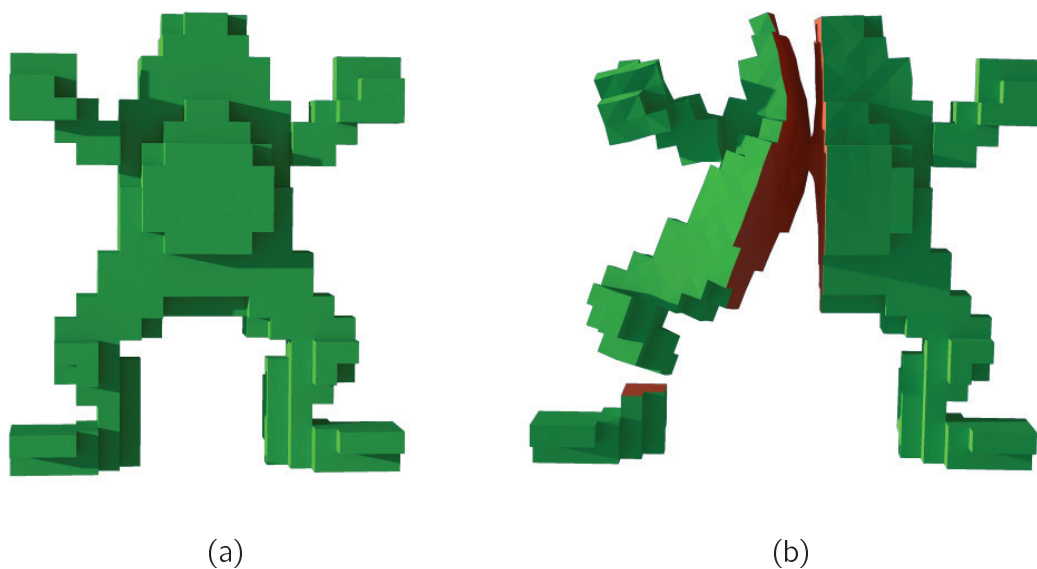
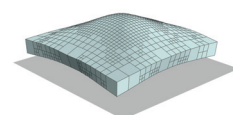


Figure 5.5 – Résultats illustrant le processus de découpe. (a) Un modèle 3D représentant une grenouille. La partie droite du modèle est fixée. (b) Le modèle de (a) déformé sous l'effet de la gravité. Notons qu'ici, la gravité s'applique de la droite vers la gauche. La grenouille subit une découpe sagittale partielle en son milieu et une découpe de son membre inférieur.



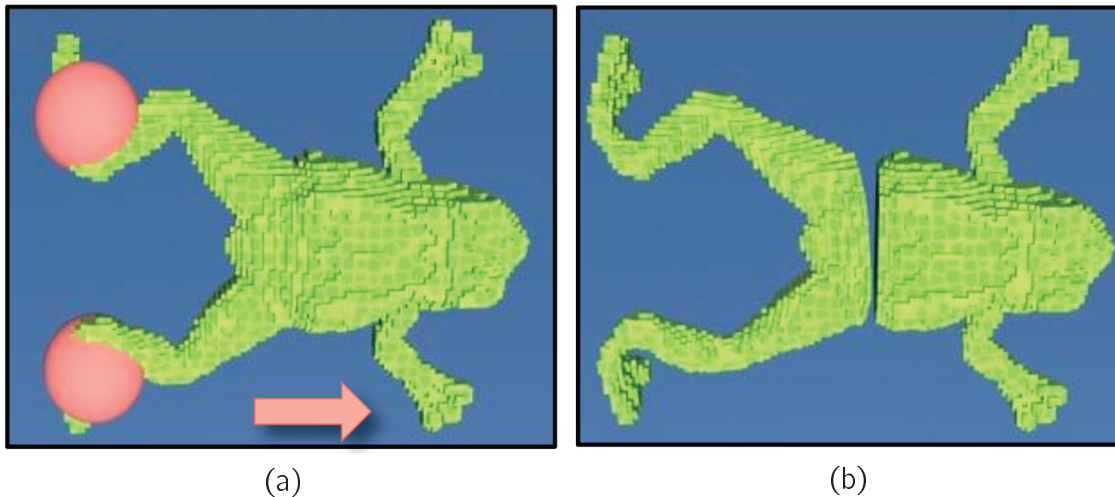


Figure 5.6 – Résultats illustrant le processus de découpe. (a) Un modèle 3D composé de 26125 cubes (32934 particules), représentant une grenouille où les pattes sont fixées. Cette contrainte est représentée par les deux sphères rouges. (b) Le modèle de (a) déformé sous l'effet de la gravité. Notons qu'ici, la gravité s'applique de la gauche vers la droite. La grenouille subit une découpe transversale en son milieu.

5.2 Découpe et raffinement

Les figures 5.7, 5.8, 5.9, 5.10, 5.11 et 5.12 montrent les capacités de notre modèle unifié à gérer les changements topologiques que sont la découpe et le raffinement local, pour des maillages surfaciques et volumiques. L'opération de découpe est réalisée le long des arêtes des maillages surfaciques et le long des faces des maillages volumiques. Le raffinement local consiste en la subdivision d'un quadrilatère en quatre sous-quadrilatères et d'un hexaèdre en huit sous-hexaèdres. Ces changements topologiques sont réalisés au cours de la simulation, de manière interactive par l'utilisateur. Enfin, notons que dans le cadre d'un maillage volumique localement raffiné, l'opération de découpe s'effectue selon les faces hiérarchiques.

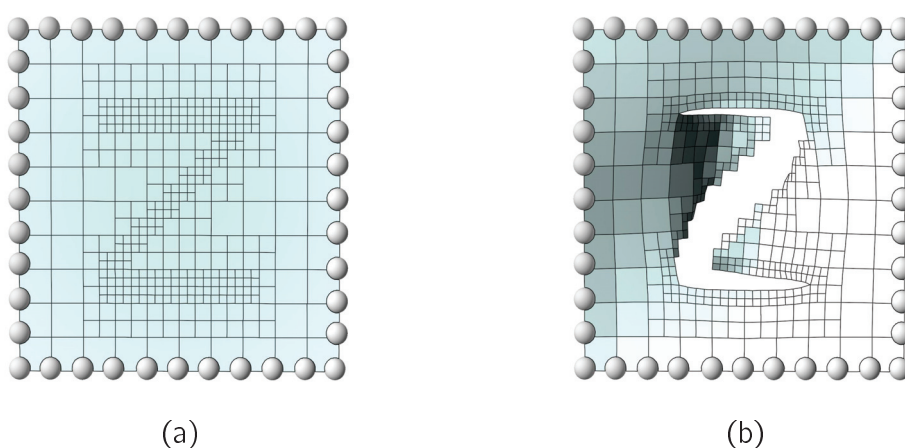


Figure 5.7 – Résultats illustrant le processus de raffinement et de découpe, pour un maillage surfacique. (a) Un maillage surfacique composé initialement de 10×10 carrés, est contraint par ses bords. Cette contrainte est représentée par les sphères grises. (b) Un état de déformation du maillage de (a), par une gravité allant de l'avant vers l'arrière, suite à un raffinement et une découpe réalisés par l'utilisateur.

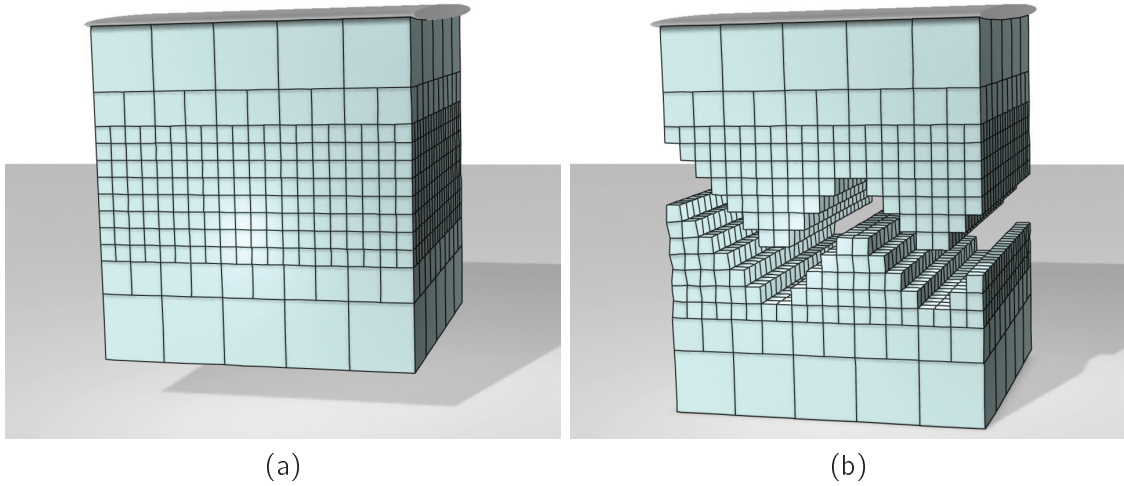


Figure 5.8 – Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique. (a) Un maillage volumique composé initialement de $5 \times 5 \times 5$ cubes, est contraint par le haut. Cette contrainte est représentée par le cylindre gris. (b) Un état de déformation du maillage de (a), par la gravité, suite à un raffinement et une découpe réalisés par l'utilisateur.

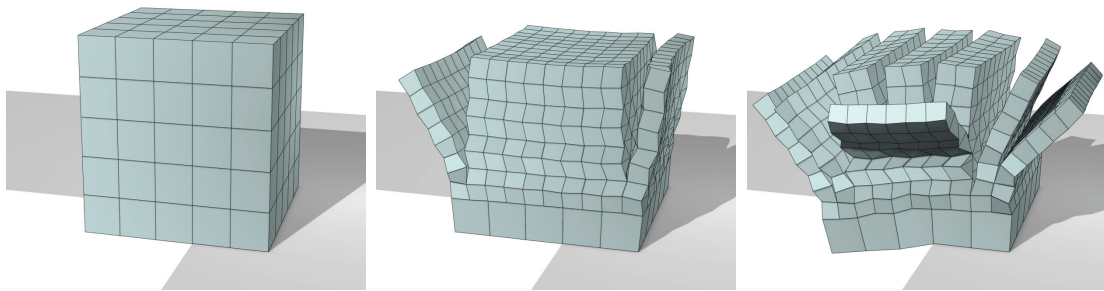


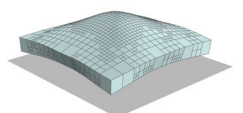
Figure 5.9 – Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique. Un maillage volumique composé initialement de $5 \times 5 \times 5$ cubes, est contraint partiellement par le bas et est soumis à la gravité. Nous observons deux états de déformation du maillage où l'utilisateur a raffiné, puis découpé le maillage tout en appliquant des forces et des contraintes au cours de la simulation.

5.3 Méthode adaptative

Nous rappelons qu'une méthode adaptative a pour objectif de permettre un compromis entre le temps de calcul et la précision de la simulation, en adaptant le maillage au moment et à l'endroit le nécessitant. Une méthode adaptative induit l'emploi d'un maillage présentant différents niveaux de subdivision au niveau de ses éléments. Nous avons fait le choix d'une méthode adaptative partant d'un maillage grossier dont les éléments sont raffinés au cours de la simulation, en fonction d'un critère géométrique basé sur la déformation des ressorts diagonaux des éléments constituant le modèle. Un élément est ainsi subdivisé, avec les opérations présentées dans ce manuscrit, s'il satisfait le critère géométrique suivant :

$$\frac{|l_d(t) - l_d(t_0)|}{l_d(t_0)} * 100.0 > seuil$$

où $l_d(t)$ et $l_d(t_0)$ sont respectivement la longueur au temps t et la longueur initiale d'un



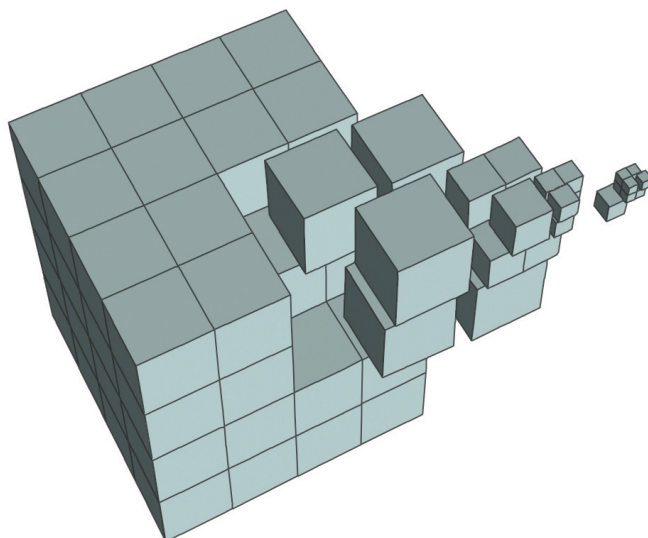


Figure 5.10 – Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique. Un maillage volumique composé initialement de $4 \times 4 \times 4$ cubes, est contraint partiellement. L'utilisateur a raffiné et découpé ce maillage. Les cubes découpés sont attirés par une même force, pour être détachés du maillage initial.

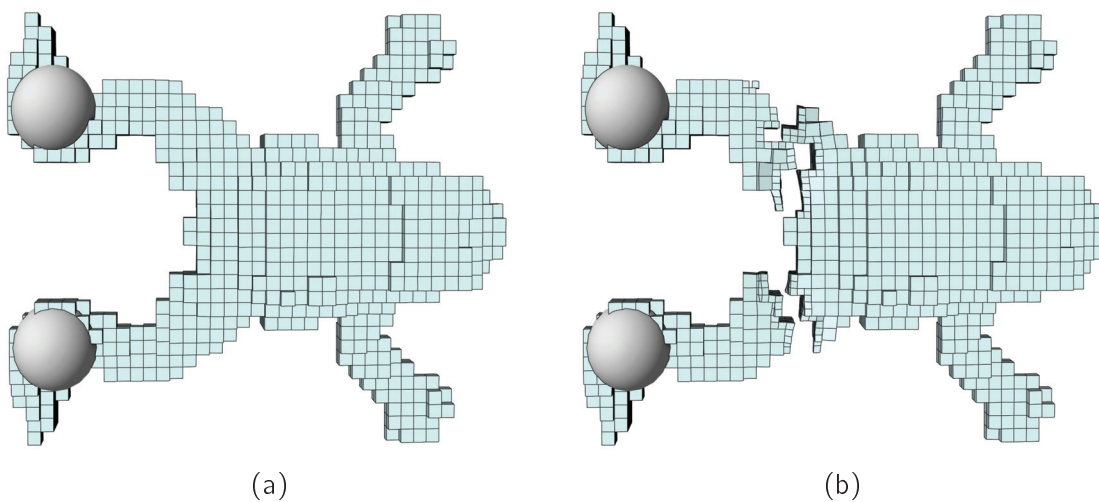


Figure 5.11 – Résultats illustrant le processus de raffinement et de découpe, pour un maillage volumique. (a) Un maillage volumique représentant grossièrement une grenouille, est contraint au niveau de ses pattes. Ces contraintes sont représentées par les sphères grises. (b) Un état de déformation du maillage de (a), par une gravité allant de la gauche vers la droite, suite à un raffinement et une découpe réalisés par l'utilisateur.

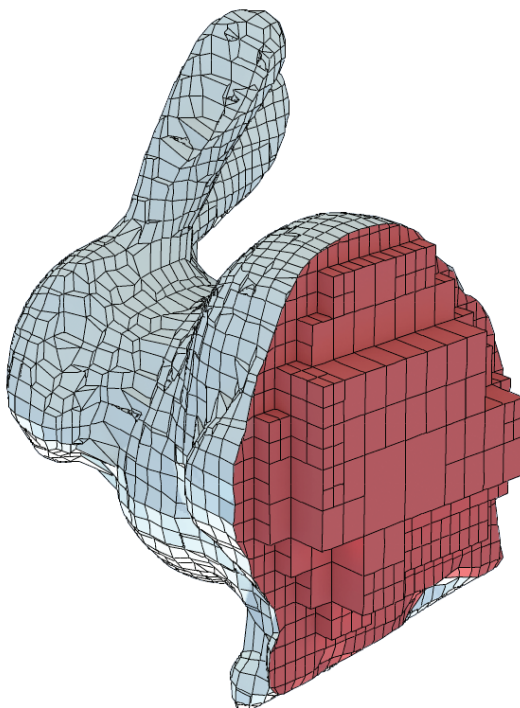
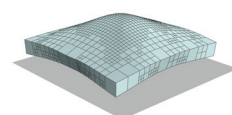


Figure 5.12 – Résultats illustrant le processus de découpe et de raffinement, pour un maillage volumique représentant un lapin. Il est issu d'une surface triangulaire représentant un lapin dont nous avons construit la boîte englobante. Cette boîte a été ensuite raffinée en huit sous-éléments. Ensuite, ces derniers ont été raffinés récursivement à chaque fois qu'ils intersectaient la surface, jusqu'à un niveau de subdivision maximal. Puis, les cubes se trouvant à cheval sur la surface ont été projetés pour donner un maillage lissé, tandis que ceux se trouvant à l'extérieur de la surface sont supprimés. Enfin, nous avons supprimé une partie des hexaèdres, pour laisser apparaître la qualité adaptative du maillage.



ressort diagonal d . Si le ressort diagonal d'un élément satisfait ce critère pour un seuil donné, alors cet élément est raffiné.

Notons que notre méthode permet évidemment la définition d'autres critères pouvant, par exemple, se baser sur les propriétés physiques des éléments en plus de la géométrie. Par ailleurs, nous avons implémenté une extension à ce critère, en raffinant les éléments adjacents à l'élément E satisfaisant le critère, s'il existe un écart de 2 entre leur niveau de subdivision et celui de E . En effet, les jonctions T induisent une approximation du comportement de l'objet simulé, et cette approximation est d'autant plus grande que le nombre de jonctions T projetées sur une même arête ou face est importante, c'est-à-dire lorsque l'écart de niveau de subdivision entre deux éléments adjacents est grand.

Les figures 5.13, 5.14, 5.15, 5.16, 5.17, 5.18 et le tableau 5.1 montrent les capacités de notre modèle unifié à gérer le raffinement adaptatif de maillages surfaciques et volumiques. Nous précisons que la figure 5.15 présente un résultat cumulant l'application d'une opération de découpe interactive en plus de la méthode adaptative de raffinement des éléments d'un maillage surfacique.

Les courbes données dans les figures 5.17 et 5.18 comparent la précision entre un modèle raffiné de manière adaptative au cours de la simulation et trois autres modèles présentant chacun un niveau de subdivision différent et fixé au cours de la simulation. Nous réalisons pour cela un test de flexion sur une poutre encastree 2D (*resp.* 3D). Chaque courbe correspond aux positions des particules situées sur la bordure haute de la poutre 2D (*resp.* 3D). Elles montrent l'état de déformation de cette poutre 2D (*resp.* 3D), pour différents niveaux de subdivision. Les courbes intitulées « Grossière », « Raffiné 3× » et « Raffiné 4× » correspondent à des niveaux de subdivision fixés qui sont respectivement 1, 8 et 16, contrairement à celle nommée « Adaptative 1–3 », ainsi que celles nommées « Adaptative 1–2 » et « Adaptative 2–3 » pour les expérimentations 3D. Le modèle « grossier » 2D (*resp.* 3D) est composé de deux carrés de dimension : $0.1\text{ m} \times 0.1\text{ m}$ (*resp.* deux cubes de dimension : $1\text{ m} \times 1\text{ m} \times 1\text{ m}$). La courbe « Adaptative 1–3 » (*resp.* « Adaptative 1–2 » et « Adaptative 2–3 ») correspond initialement au modèle « grossier », qui a été raffiné au cours de la simulation en des éléments ayant un niveau de subdivision allant de 2 à 8 (*resp.* 2 à 4 et 4 à 8). Le seuil du critère géométrique est 2% (*resp.* 1.2% en 3D) et $\rho = 1000\text{ Kg/m}^3$ (*resp.* $\rho = 100\text{ Kg/m}^3$, en 3D).

Les résultats montrent que la courbe associée à la méthode adaptative se trouve entre le modèle « grossier » et « raffiné ». Dans cet exemple, la précision de notre méthode adaptative est cohérente. Le tableau 5.1 montre la moyenne du temps de calcul d'une itération de la boucle de simulation, pour chacune des courbes représentées dans la figure 5.18. Nous rappelons que ces dernières correspondent à l'expérimentation 3D. Le tableau fait également correspondre à chaque courbe, le nombre d'éléments composant les modèles associés, avec leur niveau de subdivision. Nous pouvons voir que la version « Adaptatif 1–2 » est constituée de 2 éléments raffinés une fois et de 112 éléments raffinés deux fois. Sa moyenne de temps de calcul pour une itération, de 2.49ms , se situe bien entre celle du modèle « grossier » : 0.37ms et celle du modèle « Raffiné 3× » : 13.47ms . La version « Adaptatif 2–3 » est quant à elle, constituée de 18 éléments raffinés deux fois et 880 éléments raffinés trois fois. Sa moyenne de temps de calcul pour une itération, de 13.12ms , se situe juste en dessous de celle du modèle « Raffiné 3× » : 13.47ms . De plus, il est apparu que la moyenne du temps de calcul qu'a nécessité le processus de subdivision du modèle « Adaptatif 2–3 » a été de 28.66ms , ainsi que la distribution des forces des jonctions T au niveau de leur voisinage qui a atteint 5.86ms . Notons que le modèle « Adaptatif 2–3 » possède 120 jonctions T dans l'état représenté par sa courbe. Le faible gain en temps de calcul entre « Adaptatif 2–3 » et « Raffiné 3× », s'explique par le fait qu'ils ont un nombre d'éléments raffinés trois fois similaire (898 contre 1024). Une deuxième raison est le coût de la distribution des forces des

jonctions T sur leurs particules voisines, qui se fait à chaque itération. Enfin, nous précisons que tester les éléments à raffiner est également une étape coûteuse en temps et qui peut fortement impacter le temps global de la simulation, si elle est effectuée trop souvent.

Pour notre méthode adaptative, nous avons fait le choix, d'un critère de raffinement s'appuyant sur la géométrie des éléments, en testant le pourcentage de déformation des ressorts diagonaux, pour sa simplicité de mise en œuvre. Nous précisons que le choix du seuil est déterminant pour ce critère. En effet, une modification, même très faible de sa valeur peut engendrer le raffinement de tous les éléments du modèle, ou en engendrer aucun. Dans les résultats montrés par le tableau 5.1, nous avons fixé le niveau maximal de subdivision à deux fois pour le modèle « Adaptatif 1 – 2 », et trois fois pour le modèle « Adaptatif 2 – 3 ». Dans le tableau 5.1, nous constatons que ces deux modèles ne contiennent quasiment que des éléments ayant pour niveau de subdivision la valeur maximale donnée. Le fait que pratiquement tous les éléments d'un modèle adaptatif soient raffinés, ne met pas en valeur l'intérêt d'une méthode adaptative. Nous notons tout de même des premiers résultats encourageants en terme de précision et de temps. Mais, un critère de raffinement plus adapté ou encore une gestion des jonctions T moins coûteuse méritent d'être étudiés.

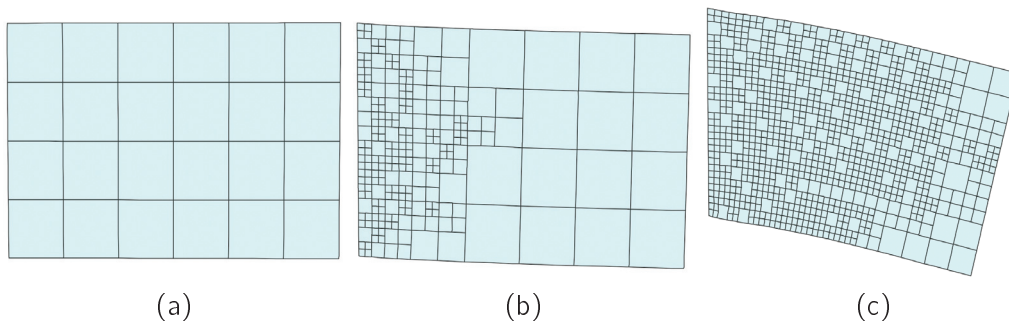


Figure 5.13 – Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique. (a) Un maillage surfacique composé de 6×3 carrés, contraint par son bord gauche et soumis à la gravité. (b) et (c) Deux états de déformation du maillage de (a). Le maillage s'est déformé tout en se raffinant selon un critère géométrique reposant sur la déformation des ressorts diagonaux des quadrilatères.

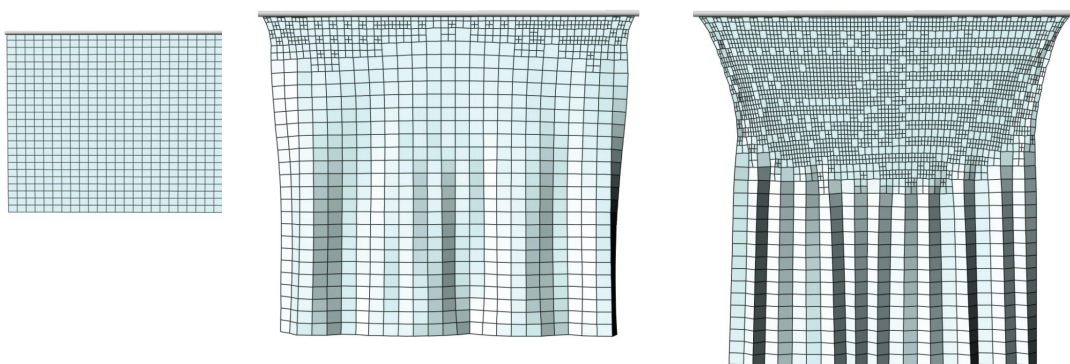
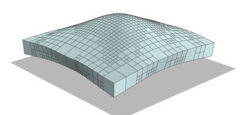


Figure 5.14 – Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique. Un maillage surfacique composé initialement de 25×25 carrés, contraint par le haut et soumis à la gravité. Cette contrainte est représentée par un cylindre gris. Le maillage surfacique est ainsi déformé tout en se raffinant selon un critère géométrique reposant sur la déformation des ressorts diagonaux des quadrilatères.



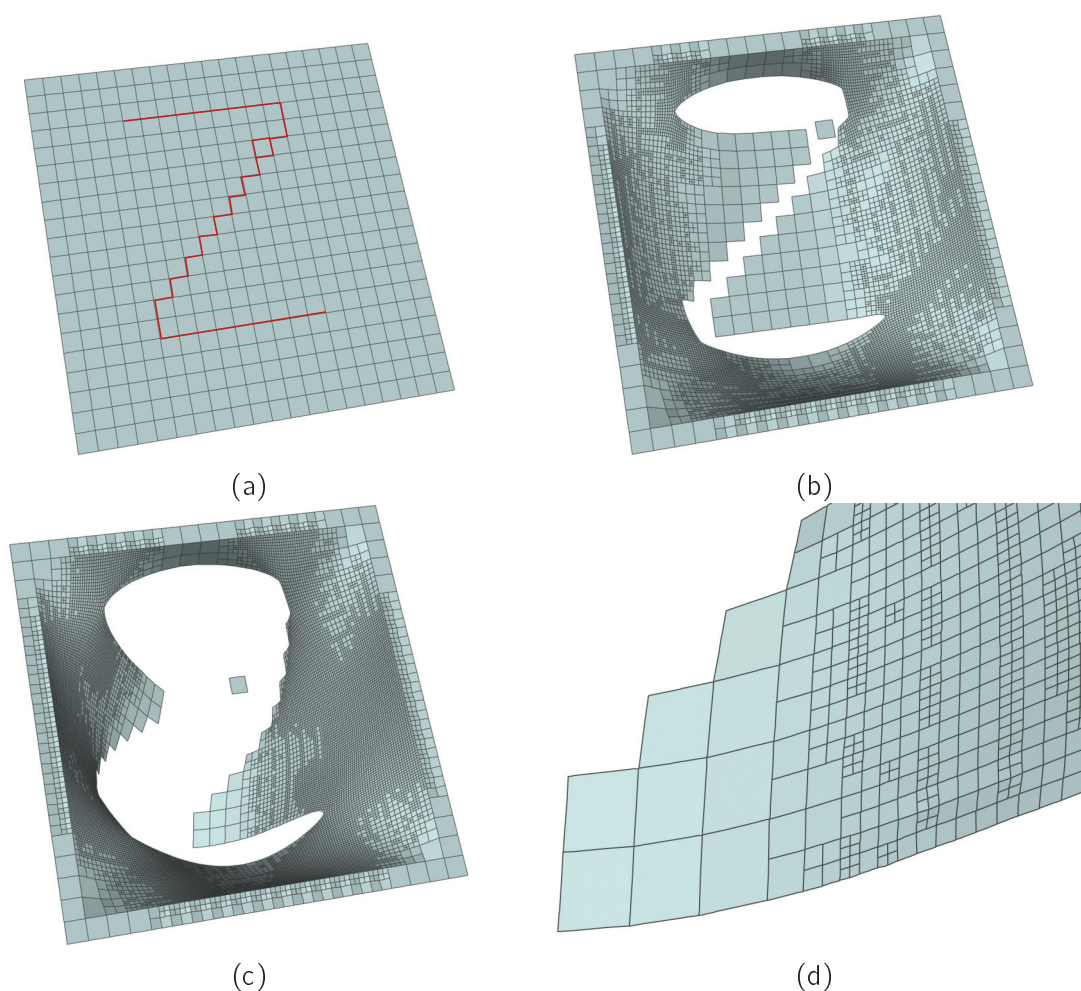


Figure 5.15 – Résultats illustrant le processus de subdivision adaptative, pour un maillage surfacique. (a) Un maillage surfacique composé initialement de 20×20 carrés, contraint le long de sa bordure, et soumis à une gravité allant de devant en arrière. L'utilisateur effectue une découpe dont le chemin est représenté en rouge. (b) et (c) correspondent à deux états de déformation où les quadrilatères sont raffinés au cours de la simulation selon un critère géométrique reposant sur la déformation des ressorts diagonaux des quadrilatères. (d) Un zoom du modèle de (c).

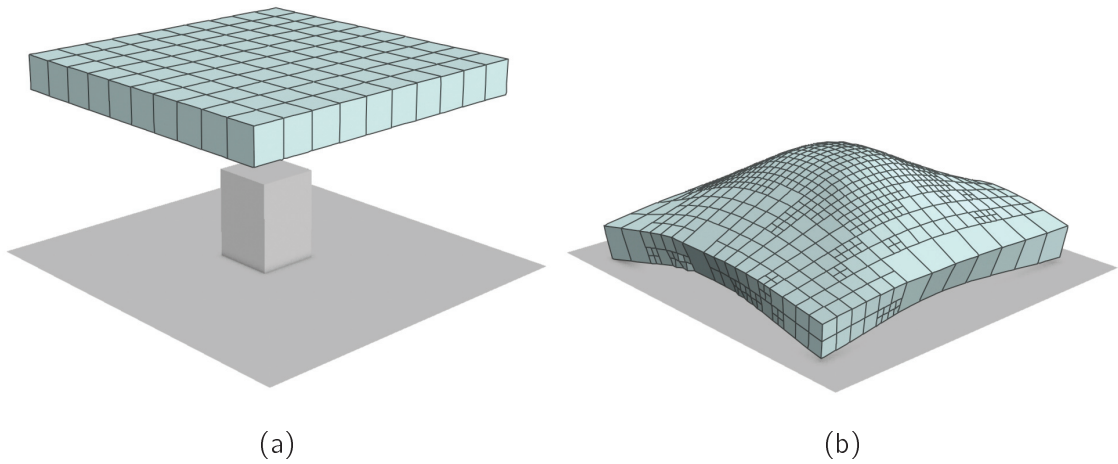


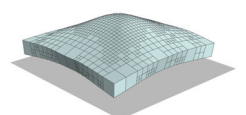
Figure 5.16 – Résultats illustrant le processus de subdivision adaptative, pour un maillage volumique. (a) Un maillage composé de 10×10 cubes. La scène est également composée d'un obstacle. (b) Le maillage de (a) après collision avec l'obstacle ayant entraîné, au cours de la simulation, le raffinement de ses hexaèdres, en fonction d'un critère géométrique reposant sur la déformation des ressorts diagonaux ($\rho = 400 \text{ Kg/m}^3$ $E = 30 \text{ KPa}$ et $\nu = 0.4$). Notons que les différentes étapes de déformation peuvent être visualisées, dans les bas de pages, tout au long de ce manuscrit.

	Temps (ms)	Nombre d'éléments
Grossier	0.37	2 elts nv. 1
Adaptatif 1-2	2.49	2 elts nv. 2 et 112 elts nv. 3
Raffiné $\times 3$	13.47	1024 elts nv. 4
Adaptatif 2-3	13.12	18 elts nv. 3 et 880 elts nv. 4
Raffiné $\times 4$	97.53	8192 elts nv. 5

Tableau 5.1 – Tableau des temps de notre méthode adaptative pour un maillage volumique. Un tableau donnant pour chaque courbe de la figure 5.18(b), la moyenne des temps de calculs pour un pas de temps de simulation, ainsi que le nombre d'éléments avec leur niveau de subdivision.

5.4 Conclusion

Les résultats présentés dans ce chapitre illustrent les opérations de découpe et de raffinement décrits dans ce manuscrit. Ils montrent la capacité de notre modèle unifié à gérer ces changements topologiques. Cependant, la définition du modèle unifié et de ses opérations, sont les prémisses d'un projet visant à montrer les bénéfices de l'emploi d'un modèle topologique complexe pour la simulation physique, tout en gérant des changements topologiques. Enfin, la validation physique de ce modèle et le choix du critère pour le raffinement, seront les étapes indispensables de ce projet, qui n'ont été que peu abordés dans ces travaux.



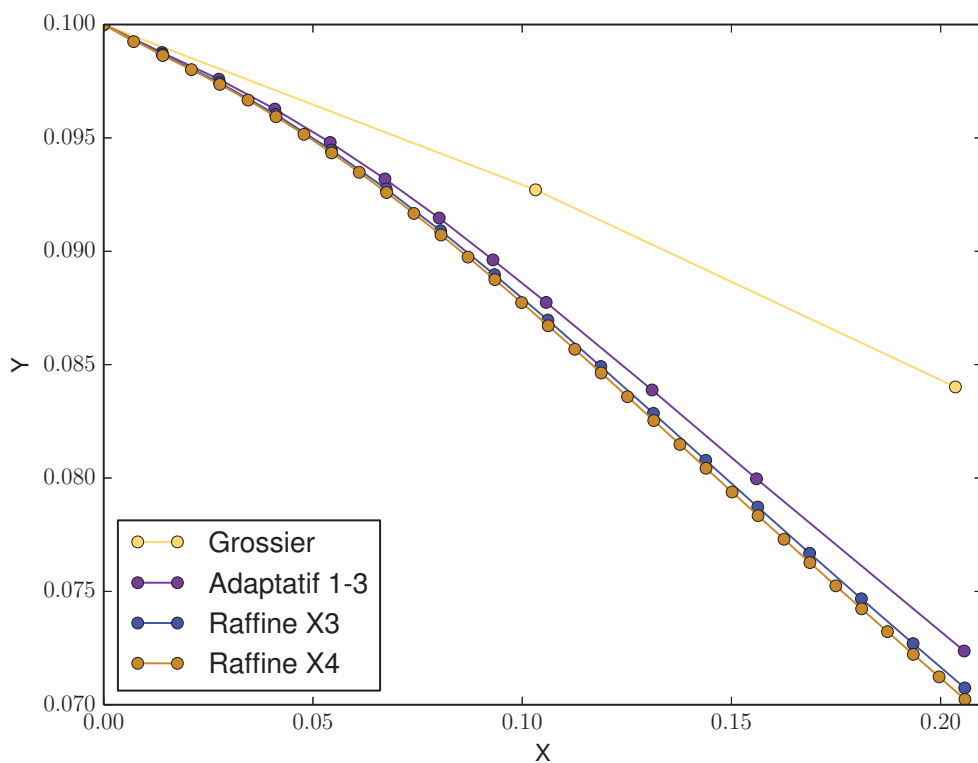
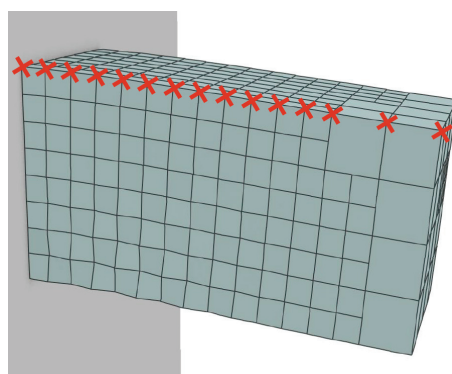
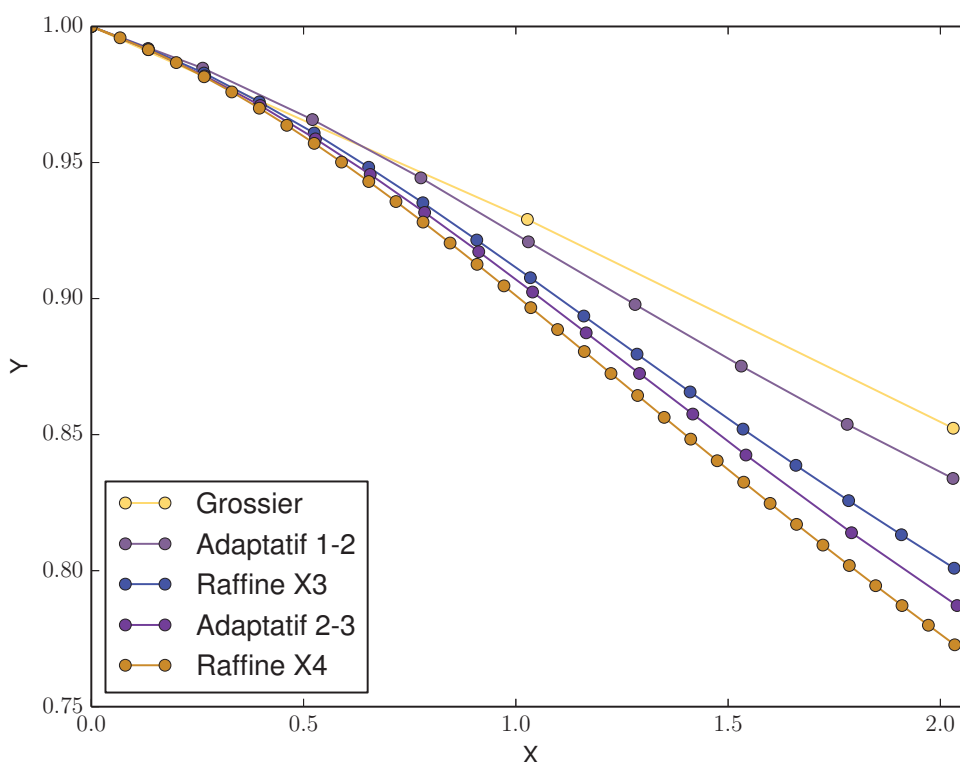


Figure 5.17 – Courbe de précision de notre méthode adaptative, pour un maillage surfacique. L'expérience réalisée correspond à un test de flexion réalisé sur une poutre 2D encastée. Chaque courbe correspond aux positions des particules situées sur la bordure haute de la poutre 2D. Les courbes présentent l'état de déformation de cette poutre 2D, pour différents niveaux de subdivision. Les courbes intitulées « Grossière », « Raffiné 3× » et « Raffiné 4× » correspondent à des niveaux de subdivision fixés qui sont respectivement 1, 8 et 16, contrairement à celle nommée « Adaptative 1 – 3 ». Le modèle « grossier » est un modèle 2D composé de deux carrés de dimension : $0.1 \text{ m} \times 0.1 \text{ m}$. La courbe « Adaptative 1 – 3 » correspond initialement au modèle « grossier », qui a été raffiné au cours de la simulation en éléments ayant un niveau de subdivision allant de 2 à 8. Le seuil du critère géométrique est 2% ($\rho = 1000 \text{ Kg/m}^3$).

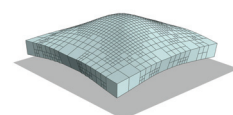


(a)



(b)

Figure 5.18 – Courbe de précision de notre méthode adaptative, pour un maillage volumique. L'expérience réalisée correspond à un test de flexion réalisé sur une poutre 3D encastree. (a) Une poutre 3D encastree soumise à la gravité. Le modèle représenté correspond à « Adaptive 2 – 3 ». (b) Chaque courbe correspond aux positions des particules situées sur la bordure haute de la poutre 3D, représentées par des croix rouges dans (a). Les courbes présentent l'état de déformation de cette poutre 3D, pour différents niveaux de subdivision. Les courbes intitulées « Grossière », « Raffiné 3x » et « Raffiné 4x » correspondent à des niveaux de subdivision fixés qui sont respectivement 1, 8 et 16, contrairement à celles nommées « Adaptive 1 – 2 » et « Adaptive 2 – 3 ». Le modèle « grossier » est un modèle 3D composé de deux cubes de dimension : $1\text{ m} \times 1\text{ m} \times 1\text{ m}$. La courbe « Adaptive 1 – 2 » (resp. « Adaptive 2 – 3 ») correspond initialement au modèle « grossier », qui a été raffiné au cours de la simulation en éléments ayant un niveau de subdivision allant de 2 à 4 (resp. 4 à 8). Le seuil du critère géométrique est 1.2% ($\rho = 100\text{ Kg/m}^3$).



Chapitre 6

*« π Golgoth avançait sur un pont d'air. A chaque nouveau pas, il aurait dû tomber. Mais il ne tombait pas. Il était à quinze mètres en amont de la falaise maintenant. Chaque fois que sa botte se leva et retomba, un bruit de vitre, de fer sur une vitre, se réverbéra jusqu'à nous. Il contracta. Il contracta face au ciel. Il contracta pour lui-même. »
(La Horde du Contrevent - Alain Damasio)*

Conclusion

Dans ce manuscrit, nous avons présenté nos travaux visant à l'élaboration d'un modèle unifié couplant le modèle topologique des complexes linéaires cellulaires, et un modèle physique de type système masses-ressorts. Notre modèle unifié permet la simulation physique d'objets déformables, tout en réalisant pendant cette simulation des changements topologiques.

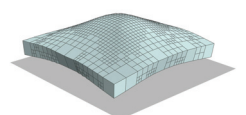
Les complexes cellulaires linéaires consistent en un modèle topologique nD , centré arête. Ils donnent accès à l'ensemble des relations d'incidence et d'adjacence entre les cellules de l'objet modélisé, facilitant la gestion des changements topologiques. Enfin, leur définition mathématique permet de garantir la validité topologique du modèle, au cours des changements topologiques.

Le système masses-ressorts est un modèle physique, présentant différents avantages comme la simplicité de ses équations, son implémentation intuitive, son interactivité et sa facilité à gérer les changements topologiques. Par contre, la méthode analytique proposée par Baudet *et al.* que nous employons pour le paramétrage des coefficients de raideur de notre système masses-ressorts, implique des modèles 2D composés de quadrilatères et des modèles 3D constitués d'hexaèdres.

La définition d'un modèle unifié évite la redondance d'information. De plus, le caractère unifié permet également une mise à jour facilitée et efficace des informations associées au modèle.

Ces travaux montrent comment utiliser un modèle topologique complexe pour la simulation physique d'objets déformables. Nous avons montré avec un modèle physique simple qu'est le système masses-ressorts, qu'il était possible d'allier un modèle topologique aussi complexe que sont les complexes cellulaires linéaires pour la simulation physique, et surtout l'intérêt de ce couplage au travers de l'implémentation de deux changements topologiques que sont la découpe du modèle en 2D, selon ses arêtes, et en 3D, selon ses faces, ainsi que la subdivision de quadrilatères ou d'hexaèdres. Nous avons détaillé dans ce manuscrit comment réaliser ces changements topologiques d'un point de vue topologique, puis d'un point de vue physique avec la mise à jour des informations embarquées au niveau du modèle topologique.

Enfin, l'opération de subdivision locale des quadrilatères ou des hexaèdres du modèle,



que nous avons implémentée, nous permet de proposer une méthode adaptative, consistant à débiter avec un modèle contenant peu d'éléments, et à subdiviser ces derniers, selon un critère géométrique reposant sur la déformation des ressorts diagonaux.

Pour conclure, ces travaux ont permis la réalisation d'un framework générique nommé « TopoSim », incluant la librairie CGAL. Cette dernière propose une implémentation du modèle topologique des complexes cellulaires linéaires, qui est à la base de notre modèle unifié. Ce framework permet la simulation d'objets déformables avec l'implémentation d'un schéma numérique d'Euler explicite symplectique et l'intégration d'un système masses-ressorts. Enfin, son interface graphique permet d'interagir avec la simulation, en appliquant des forces, des déplacements et des contraintes, ou encore en effectuant des changements topologiques comme la découpe et la subdivision d'éléments. Notons que cette dernière peut être réalisée de manière automatique dans le cadre d'une méthode adaptative.

Perspectives Le modèle topologique des cartes combinatoires duquel est issu les complexes cellulaires linéaires, est générique sur la topologie de ses cellules. Ils permettent ainsi de mixer en 2D, des polygones (triangles, quadrilatères, ...), et en 3D, des polyèdres (tétraèdres, hexaèdres, pyramides, ...). Cette propriété des cartes combinatoires est un atout considérable pour la problématique des méthodes adaptatives. En effet, elle permet de raffiner de manière quelconque les cellules, sans se soucier de la validité topologique et en ayant toutes les informations d'incidence et d'adjacence entre les cellules du modèle.

Par contre, il n'existe pas de méthode analytique unique pour la paramétrisation du modèle physique des systèmes masses-ressorts, pour des éléments quelconques. C'est pour cette raison, que nous sommes en train d'intégrer dans notre framework générique le modèle physique des masses-tenseurs proposant des fonctions de forme pour différentes formes d'éléments. De plus, les masses-tenseurs sont plus précis que les masses-ressorts, car ils sont directement issus de la mécanique des milieux continus. Enfin, ces derniers prennent en compte le volume des éléments 3D, qui est une limitation des systèmes masses-ressorts.

Notre méthode adaptative pourrait être considérablement bonifiée par l'ajout d'une opération de dé-raffinement des quadrilatères et des hexaèdres. De plus, une étude sur le choix du critère de raffinement sera une étape indispensable à notre méthode de raffinement.

Le framework « TopoSim », a permis d'étudier comment coupler un modèle topologique complexe avec la simulation physique. Il propose un premier modèle unifié et des opérations de parcours et de modification. Il est capable de gérer des changements topologiques comme la découpe ou la subdivision. De plus, notre framework propose une méthode adaptative. Pour conclure, il constitue donc une base à un nouveau sujet d'études, permettant de faire bénéficier à la simulation physique, de tous les avantages que peuvent offrir un modèle topologique complexe.

Chapitre 7

« Certes, vous ne pouvez voir qu'une de mes sections, ou Cercles; car votre œil ne possède pas la faculté de s'élever au-dessus de votre surface; mais il vous est au moins permis de constater que mes sections deviennent plus petites à mesure que je m'élève dans l'Espace. Voyez, je vais m'élever; et vous aurez l'impression que mon Cercle se rapetisse, pour se réduire à un Point et finalement disparaître. »
(Flatland - Edwin Abbott Abbott)

Annexes

Sommaire

7.1	Annexe A	189
7.2	Annexe B	191

7.1 Annexe A

Algorithme de subdivision d'un quadrilatère en quatre sous-quadrilatères

L'appel à l'algorithme 3 sera abrégé par « PBHFS » (Prochain Brin Hiérarchique Face Simple). La procédure « insertion-sommet-dans-arete » prend en paramètre un brin incident à l'arête dans laquelle sera insérée un nouveau sommet. La fonction « insertion-arete-dans-face » prend en paramètre les deux brins entre lesquels sera inséré une nouvelle arête. Elle retourne un brin incident à cette nouvelle arête.

Algorithme 17 : Scinder une arête en deux

Données : b : le brin hiérarchique d'une arête hiérarchique a à scinder.

Résultat : a est scindée en deux.

$b_1 \leftarrow \text{PBHFS}(b_0);$

Si $\beta_1(b) = b_1$ **Alors**

$\text{insertion-sommet-dans-arete}(b);$

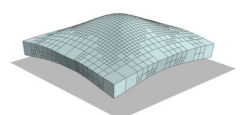
 // b et $\beta_1(b)$ correspondent aux deux brins résultants, incidents
 aux deux parties de l'arête venant d'être scindée.

$l_b \leftarrow 2l_F;$

$l_{\beta_1(b)} \leftarrow 2l_F;$

$\text{creation-particule}(\beta_1(b));$

$\text{creation-ressort}(\beta_1(b));$



Algorithme 18 : Scinder une face en deux

Données : b_1 et b_2 : les deux brins entre lesquels sera inséré une nouvelle arête scindant une face F .

Résultat : F est scindée en deux parties.

```

 $b_{nouvelle\_arete} \leftarrow$  insertion-arete-dans-face( $b_1, b_2$ );
marque-hierarchiquement( $b_{nouvelle\_arete}$ );
marque-hierarchiquement( $\beta_2(b_{nouvelle\_arete})$ );
marque-hierarchiquement( $\beta_{21}(b_{nouvelle\_arete})$ );
marque-hierarchiquement( $\beta_1(b_{nouvelle\_arete})$ );
 $l_{b_{nouvelle\_arete}} \leftarrow 2l_F$ ;
creation-ressort( $\beta_1(b_{nouvelle\_arete})$ );

```

Algorithme 19 : Subdiviser une face en quatre sous-faces

Données : b : un brin incident à la face F à subdiviser.

Résultat : F est subdivisée en quatre sous-faces : F_{00} , F_{01} , F_{10} et F_{11} .

Si b n'est pas un brin hiérarchique **Alors**

```

 $b \leftarrow$  PBHFS( $b$ );

```

// Subdivision des arêtes hiérarchiques de F

Pour i allant de 1 à 4 **faire**

```

  couper-arete-en-deux( $b$ );
   $b \leftarrow$  PBHFS( $b$ );

```

```

 $b_{milieu1} \leftarrow$  recuperer-brin-milieu( $b$ );

```

```

 $b_{milieu2} \leftarrow$  recuperer-brin-milieu(PBHFS(PBHFS( $b$ )));

```

// Scission de F en deux sous-faces F_0 et F_1

```

 $b_{nouvelle\_arete0} \leftarrow$  inserer-arete-dans-face( $b_{milieu1}, b_{milieu2}$ );

```

```

creation-ressort( $\beta_1(b_{nouvelle\_arete0})$ );

```

```

scinder-arete-en-deux( $b_{nouvelle\_arete0}$ );

```

```

marque-hierarchiquement( $b_{nouvelle\_arete0}$ );

```

```

marque-hierarchiquement( $\beta_{12}(b_{nouvelle\_arete0})$ );

```

```

marque-hierarchiquement( $\beta_{11}(b_{nouvelle\_arete0})$ );

```

```

marque-hierarchiquement( $\beta_{21}(b_{nouvelle\_arete0})$ );

```

// Scission de F_0 en deux sous-faces F_{00} et F_{01}

```

 $b_{milieu1} \leftarrow$  récupérer-brin-milieu( $b_{nouvelle\_arete0}$ );

```

```

 $b_{oppose} \leftarrow$  PBHFS(PBHFS( $b_{nouvelle\_arete0}$ ));

```

```

 $b_{milieu2} \leftarrow$  récupérer-brin-milieu( $b_{oppose}$ );

```

```

scinder-face-en-deux( $b_{milieu1}, b_{milieu2}$ );

```

// Scission de F_1 en deux sous-faces F_{10} et F_{11}

```

 $b_{nouvelle\_arete0} \leftarrow \beta_2(b_{nouvelle\_arete0})$ ;

```

```

 $b_{milieu1} \leftarrow$  récupérer-brin-milieu( $b_{nouvelle\_arete0}$ );

```

```

 $b_{oppose} \leftarrow$  PBHFS(PBHFS( $b_{nouvelle\_arete0}$ ));

```

```

 $b_{milieu2} \leftarrow$  récupérer-brin-milieu( $b_{oppose}$ );

```

```

scinder-face-en-deux( $b_{milieu1}, b_{milieu2}$ );

```

```

 $l_F \leftarrow 2l_F$ ;

```

7.2 Annexe B

Algorithme de subdivision d'un hexaèdre en huit sous-hexaèdres

L'appel à l'algorithme 5 sera abrégé par « PBHF » (Prochain Brin Hiérarchique Face).

Algorithme 20 : Tester si une face contient un brin hiérarchique à un volume

Données : b_0 : un brin hiérarchique d'une face F .

Résultat : Retourne un brin hiérarchique de la face F si elle en contient un, *NULL* sinon.

$b_1 \leftarrow b_0$;

Tant que b_1 n'est pas un brin hiérarchique et $b_1 \neq b_0$ **faire**

$b_1 \leftarrow \beta_1(b_1)$;

Si $b_1 = b_0$ **Alors**

Retourner *NULL*;

Retourner b_1 ;

Algorithme 21 : Marquer hiérarchiquement les brins incidents à une face hiérarchique simple

Données : b : un brin incident à une face hiérarchique simple F . b est le premier brin de F à devoir être marqué comme hiérarchique.

l : un niveau de subdivision.

regulier : un booléen indiquant si la longueur et la largeur de F sont équivalentes en terme de niveau de subdivision cumulé (par défaut *regulier* vaut Vrai).

Résultat : Les brins incidents à F sont correctement marqués hiérarchiquement.

Si *regulier* **Alors**

Pour i allant de 1 à 4 **faire**

 marquer-hierarchiquement(b);

$b \leftarrow$ prochain-brin-subdivision-face-simple(b, l);

Sinon

Pour i allant de 1 à 4 **faire**

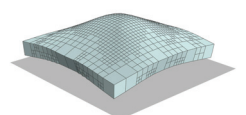
 marquer-hierarchiquement(b);

Si i est pair **Alors**

$b \leftarrow$ prochain-brin-subdivision-face-simple(b, l);

Sinon

$b \leftarrow$ prochain-brin-subdivision-face-simple($b, 2l$);



Algorithme 22 : Récupérer le prochain brin d'une face hiérarchique et simple en fonction d'un niveau de subdivision

Données : b : un brin hiérarchique et incident à une face hiérarchique simple F .
 l : un niveau de subdivision.

Résultat : Retourne le prochain brin de F après b , en fonction de selon l .

$cumule \leftarrow 0$;

Tant que $cumule < \frac{1}{l}$ **faire**

$cumule \leftarrow cumule + \frac{1}{l}$;
 $b \leftarrow \beta_1(b)$;

Retourner $\beta_1(b)$;

Algorithme 23 : Subdiviser une face en deux

Données : b : un brin incident à une face hiérarchique F . b correspond au brin hiérarchique associé à une des deux longueurs de F .

Résultat : F est scindée selon sa longueur, en deux sous-faces.

$b_{milieu1} \leftarrow \text{recupérer-brin-milieu}(b)$;

$b_{oppose} \leftarrow \text{PBHFS}(\text{PBHFS}(b))$;

$b_{milieu2} \leftarrow \text{recupérer-brin-milieu}(b_{oppose})$;

// F est scindée en deux sous-faces F_0 et F_1

$b_{nouvelle_arete0} \leftarrow \text{insertion-arete-dans-face}(b_{milieu1}, b_{milieu2})$;

marquer-hierarchiquement($b_{nouvelle_arete0}$);

marquer-hierarchiquement($\beta_1(b_{nouvelle_arete0})$);

marquer-hierarchiquement($\beta_2(b_{nouvelle_arete0})$);

marquer-hierarchiquement($\beta_{21}(b_{nouvelle_arete0})$);

// Traitement de la partie de F incidente au volume adjacent

marquer-hierarchiquement($\beta_3(b_{nouvelle_arete0})$);

marquer-hierarchiquement($\beta_{31}(b_{nouvelle_arete0})$);

marquer-hierarchiquement($\beta_{32}(b_{nouvelle_arete0})$);

marquer-hierarchiquement($\beta_{321}(b_{nouvelle_arete0})$);

$l_{nouvelle_arete0} \leftarrow 2l_{Vol}$;

creation-ressort($\beta_1(b_{nouvelle_arete0})$);

Algorithme 24 : Subdiviser un volume en deux (fonction intermédiaire)

Données : b : un brin.

num : un entier allant de 0 à 2, correspondant à l'étape de subdivision du volume.

l : un niveau de subdivision.

Résultat : La face incidente à b est correctement marquée hiérarchiquement.

Si $num = 0$ **Alors**

 marquer-hierarchiquement-face-simple(b, l);

Sinon Si $num = 1$ **Alors**

 marquer-hierarchiquement-face-simple(b, l, Faux);

Sinon Si $num = 2$ **Alors**

 marquer-hierarchiquement-face-simple($b, 2l$);

Algorithme 25 : Subdiviser un volume en deux

Données : b : un brin incident à un volume Vol .
 num : un entier allant de 0 à 2, correspondant à l'étape de subdivision du volume.
 b_{autre_volume} : un brin donnée-résultat stockant un brin incident au deuxième volume résultant (b correspondant au premier), si ce dernier n'est pas précisé alors il n'est pas utilisé.

Résultat : Vol est scindé en deux sous-volumes : Vol_0 et Vol_1 .

```

 $b_0 \leftarrow$  prochain-brin-hierarchique-face( $b$ );
 $\mathcal{C} \leftarrow$  un conteneur de brins;
 $b_1 \leftarrow \beta_2(b_0)$ ;
// Construire  $\mathcal{C}$  pour insérer la face  $F$  scindant  $Vol$ 
 $\mathcal{C} \leftarrow$  construire-cycle( $b_0$ );
insertion-face-dans-volume( $\mathcal{C}$ );
// Marquer hiérarchiquement  $F_{|Vol_0}$ 
 $b_\gamma \leftarrow \beta_{12}(b_0)$ ;
//  $b_\gamma$  est un brin de  $F$ 
subdivision-volume-en-deux-marquer( $b_\gamma$ ,  $num$ ,  $l_{Vol}$ );
// Marquer hiérarchiquement  $F_{|Vol_1}$ 
 $b_\gamma \leftarrow \beta_2(b_1)$ ;
subdivision-volume-en-deux-marquer( $b_\gamma$ ,  $num$ ,  $l_{Vol}$ );
Si  $num = 0$  Alors
    // Subdiviser  $F$  en quatre sous-faces
    subdivision-face( $b_\gamma$ );
    marquer-hierarchiquement-face( $\beta_3(b_\gamma)$ );
Sinon Si  $num = 1$  Alors
    // Subdiviser  $F$  en deux sous-faces
     $b_{longueur} \leftarrow$  prochain-brin-hierarchique-face-simple( $\beta_2(b_0)$ );
    subdiviser-face-en-deux( $b_{longueur}$ );
 $b_{autre\_volume} \leftarrow \beta_{232}(b_0)$ ;

```

Algorithme 26 : Subdiviser un volume en quatre

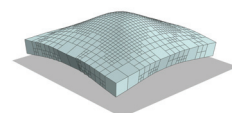
Données : b : un brin incident à un volume Vol .

Résultat : Le volume Vol est scindé en quatre sous-volumes.

```

 $b \leftarrow$  prochain-brin-hierarchique-face( $b$ );
subdivision-volume-en-deux( $b$ , 1,  $b_{autre\_volume}$ );
 $b \leftarrow \beta_{21}$ (prochain-brin-hierarchique-face( $b$ ));
subdivision-volume-en-deux( $b$ , 2);
 $b \leftarrow \beta_{21}$ (prochain-brin-hierarchique-face( $b_{autre\_volume}$ ));
subdivision-volume-en-deux( $b$ , 2);

```

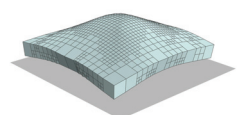


Algorithme 27 : Subdiviser un volume en huit sous-volumes

Données : b : un brin incident à un volume Vol .
Résultat : Vol est subdivisé en huit sous-volumes.
 $b \leftarrow$ obtenir-brin-hierarchique-element(b);
 $\mathcal{S} \leftarrow$ un conteneur de faces;
 $b_{\text{autre_volume}} \leftarrow$ un brin;
// Subdivision des faces hiérarchiques de Vol n'ayant pas été
préalablement subdivisées
Pour tous Les faces F incidentes à Vol **faire**
| $b_0 \leftarrow$ contient-brin-hierarchique(un brin de F);
| **Si** $b_0 \neq \text{NULL}$ **Alors**
| | **Si** $l_{b_0} = l_{Vol}$ ou est-simple(b_0) **Alors**
| | | subdivision-face(b_0);
| | **Sinon**
| | | stocker-face(\mathcal{S} , b_0);
// Les faces hiérarchiques de Vol ayant été préalablement subdivisées
sont marquées hiérarchiquement
Pour tous Les face $F' \in \mathcal{S}$ **faire**
| $b_1 \leftarrow$ un brin de F' ;
| marquer-hierarchiquement-face(b_1);
// Vol est scindé en deux sous-volumes Vol_0 et Vol_1
subdivision-volume-en-deux(b , 0, $b_{\text{autre_volume}}$);
// Vol_0 est scindé en quatre sous-volumes
subdivision-volume-en-quatre(b);
// Vol_1 est scindé en quatre sous-volumes
 $b_{\text{autre_volume}} \leftarrow$ PBHF(PBHF($b_{\text{autre_volume}}$));
subdivision-volume-en-quatre($b_{\text{autre_volume}}$);
Pour tous Les huit nouveaux sous-volumes Vol_n de Vol **faire**
| $l_{Vol_n} \leftarrow 2l_{Vol}$;

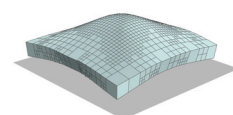
Bibliographie

- [BADSM08] M. Baba-Ali, G. Damiand, X. Skapin, and D. Marcheix. Insertion and expansion operations for n -dimensional generalized maps. In *Proc. of DGCI 2008*, volume 4992 of *LNCS*, pages 141–152, Lyon, France, April 2008. Springer Berlin/Heidelberg.
- [Bat82] Klaus-Jürgen Bathe. *Finite element procedures in engineering analysis*. Prentice-Hall civil engineering and engineering mechanics series. Englewood Cliffs, N.J. Prentice-Hall, 1982.
- [Bau72] Bruce G. Baumgart. Winged edge polyhedron representation. Technical report, Stanford, CA, USA, 1972.
- [Bau75] Bruce G. Baumgart. A polyhedron representation for computer vision. In *Proceedings of the May 19-22, 1975, National Computer Conference and Exposition*, AFIPS '75, pages 589–596, New York, NY, USA, 1975. ACM.
- [Bau06] Vincent Baudet. *Modélisation et simulation paramétrable d'objets déformables*. PhD thesis, Université Lyon 1, 2006.
- [BBJ⁺09a] Vincent Baudet, Michaël Beuve, Fabrice Jaillet, Behzad Shariat, and Florence Zara. Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation. February 2009.
- [BBJ⁺09b] Vincent Baudet, Michaël Beuve, Fabrice Jaillet, Behzad Shariat, and Florence Zara. Integrating Tensile Parameters in Mass-Spring System for Deformable Object Simulation. Technical Report RR-LIRIS-2009-034, Université Lyon 1, September 2009.
- [BCS⁺11] Richard Bezin, Benoît Crespin, Xavier Skapin, Olivier Terraz, and Philippe Me-seure. Topological operations for geomorphological evolution. In *VRIPHYS*, pages 139–148, 2011.
- [BD99] Jean-Pierre Braquelaire and Jean-Philippe Domenger. Representation of segmented images with discrete geometric maps. *Image and Vision Computing*, 17(10) :715 – 735, 1999.
- [BFA02] Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Trans. Graph.*, 21(3) :594–603, July 2002.
- [BHW94] David E. Breen, Donald H. House, and Michael J. Wozny. Predicting the drape of woven cloth using interacting particles. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '94, pages 365–372, New York, NY, USA, 1994. ACM.



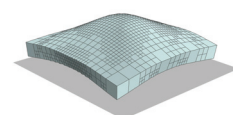
- [BKP⁺10] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy. *Polygon Mesh Processing*. Ak Peters Series. Taylor & Francis, 2010.
- [BL05] Y. Bhasin and A. Liu. Bounds for damping that guarantee stability in mass-spring systems. *Studies in health technology and informatics*, 119 :55, 2005.
- [BMD03] Luc Brun, Myriam Mokhtari, and Jean Philippe Domenger. Incremental modifications on segmented image defined by discrete maps. *Journal of Visual Communication and Image Representation*, 14 :251–290, 2003.
- [BN96] M. Bro-Nielsen. Medical image registration and surgery simulation, 1996. IMM-PHD-1996-25.
- [BNC96] Morten Bro-Nielsen and Stephane Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum*, 15(3) :57–66, 1996.
- [Bot05] Mario Botsch. *High quality surface generation and efficient multiresolution editing based on triangle meshes*. PhD thesis, 2005.
- [BP00] Marshall Bern and Paul Plassmann. Mesh generation. In *Handbook of Computational Geometry*. Elsevier Science, pages 291–332, 2000.
- [BPC⁺11] Richard Bézin, Alexandre Peyrat, Benoît Crespin, Olivier Terraz, Xavier Skapin, and Philippe Meseure. Interactive Hydraulic Erosion Using CUDA. *Machine Graphics and vision*, 20(2) :157–172, 2011. Extended version of ICCVG 2010 article.
- [BSSH03] Gérald Bianchi, Barbara Solenthaler, Gábor Székely, and Matthias Harders. Mesh topology for mass-spring models. In Springer-Verlag, editor, *MICCAI 2003*, pages 50–58, Berlin, 2003.
- [BSSH04] Gérald Bianchi, Barbara Solenthaler, Gábor Székely, and Matthias Harders. Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. In Springer-Verlag, editor, *MICCAI 2004*, pages 293–301, Berlin, 2004.
- [BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98*, pages 43–54, New York, NY, USA, 1998. ACM.
- [BZS⁺13] Romain Buttin, Florence Zara, Behzad Shariat, Tanneguy Redarce, and Gilles Grangé. Biomechanical simulation of the fetal descent without imposed theoretical trajectory. *Computer Methods and Programs in Biomedicine*, 111(2) :389–401, August 2013.
- [Bé13] Richard Bézin. *Simulation d'évolution topologique : cas de l'interaction fluide / solide*. PhD thesis, Université de Limoges, 2013.
- [CADC11] Hadrien Courtecuisse, Jérémie Allard, Christian Duriez, and Stéphane Cotin. Preconditioner-Based Contact Response and Application to Cataract Surgery. In G. Fichtinger, A. Martel, and T. Peters, editors, *MICCAI - 14th International Conference on Medical Image Computing and Computer Assisted Intervention - 2011*, volume 6891 of *Lecture Notes in Computer Science*, pages 315–322, Toronto, Canada, September 2011. Springer.
- [CD03] I. Chowdhury and S. Dasgupta. Computation of Rayleigh Damping Coefficients for Large Systems. *The Electronic Journal of Geotechnical Engineering*, 8, 2003.

- [CDA99] Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. Efficient linear elastic models of soft tissues for real-time surgery simulation. *Proceedings of the Medicine Meets Virtual Reality (MMVR 7)*, 62 :100–101, 1999.
- [CDC⁺96] S. Cotin, H. Delingette, J-M. Clément, V. Tasseti, J. Marescaux, and N. Ayache. Geometric and Physical Representations for a Simulator of Hepatic Surgery. In *Medicine Meets Virtual Reality IV*, Interactive Technology and the New Paradigm for Healthcare, pages 139–151. IOS Press, January 1996.
- [CDL⁺01] P. Cinquin, Emmanuel Dubois, S. Lavallée, Laurence Nigay, and J. Troccaz. Surgétique. de l'intérêt de la coopération entre médecin et machine au service du malade dans la réalisation de gestes médico-chirurgicaux. *Les Cahiers du Numérique-L'information médicale numérique*, 2(2) :197–207, 2001.
- [CEO⁺93] S.A Cover, N.F. Ezquerra, J.F. O'Brien, R. Rowe, T. Gadacz, and E. Palm. Interactively deformable models for surgery simulation. *Computer Graphics and Applications, IEEE*, 13(6) :68–75, Nov 1993.
- [CGA] CGAL. *CGAL User and Reference Manual*. CGAL Editorial Board, 4.2 edition. <http://www.cgal.org/>.
- [CGo] Combinatorial and geometric modeling with generic n-mpas. <http://cgogn.unistra.fr>.
- [CHCK02] Yoo-Joo Choi, Min Hong, Min-Hyung Choi, and Myoung-Hee Kim. Adaptive mass-spring simulation using surface wavelet. In *Proc. of VSMM 2002*, Sep 2002.
- [CHP89] J. E. Chadwick, D. R. Haumann, and R. E. Parent. Layered construction for deformable animated characters. *SIGGRAPH Comput. Graph.*, 23(3) :243–252, July 1989.
- [CNM07] J. Coirier and C. Nadot-Martin. *Mécanique des milieux continus - 3ème édition*. Dunod, 2007.
- [COR] Cordis-anima. <http://acroe.imag.fr/produits/logiciel/cordis/cordis.html>.
- [Cot97] S. Cotin. *Modèles anatomiques déformables en temps réel : Application à la simulation de chirurgie avec retour d'effort*. PhD thesis, Université de Nice Sophia-Antipolis, 1997.
- [Dam10] G. Damiand. *Contributions aux Cartes Combinatoires et Cartes Généralisées : Simplification, Modèles, Invariants Topologiques et Applications*. Habilitation à diriger des recherches, Université Lyon 1, Septembre 2010.
- [Dam12] G. Damiand. Combinatorial maps. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.0 edition, 2012.
- [DDCB01] Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani, and Alan H. Barr. Dynamic real-time deformations using space and time adaptive sampling. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 31–36, New York, NY, USA, 2001. ACM.
- [DDV14] Abdoulaye Abou Diakité, Guillaume Damiand, and Dirk Van Maercke. Topological Reconstruction of Complex 3D Buildings and Automatic Extraction of Levels of Detail. In Vincent Turre Gonzalo Besuievsky, editor, *Eurographics Workshop on Urban Data Modelling and Visualisation*, Proc. of Eurographics Workshop on Urban Data Modelling and Visualisation, pages 25–30. Eurographics Association, April 2014.



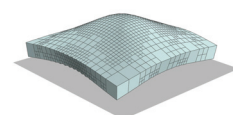
- [Del99] H. Delingette. Simulation d'interventions chirurgicales. In E. Dombre and W. Khalil, editors, *Deuxième Journées de la Recherche en Robotique (JNRR'99)*, pages 109–118, Montpellier, sep 1999.
- [Del08] Herve Delingette. Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualization and Computer Graphics*, 14(2) :329–341, March 2008.
- [DGW11] Christian Dick, Joachim Georgii, and Rudiger Westermann. A hexahedral multigrid approach for simulating cuts in deformable objects. *IEEE Transactions on Visualization and Computer Graphics*, 17(11) :1663–1675, 2011.
- [DKS⁺11] E. Darles, S. Kalantari, X. Skapin, B. Crespin, and A. Luciani. Hybrid Physical – Topological Modeling of Physical Shapes Transformations. In *Proc. of DMDCM 2011*, pages 154–157, Washington, DC, USA, 2011.
- [DKT95] Oliver Deussen, Leif Kobbelt, and Peter Tücke. Using simulated annealing to obtain good nodal approximations of deformable bodies. In *In Sixth Eurographics Workshop on Simulation and Animation*, pages 30–43. Springer, 1995.
- [DN08] Zhigang Deng and Ulrich Neumann. *Data-Driven 3D Facial Animation*. 1 edition, 2008.
- [DRVDP14] Boris Dalstein, Remi Ronfard, and Michiel Van De Panne. Vector Graphics Complexes. *ACM transactions on Graphics, Proceedings of ACM SIGGRAPH*, 33(4) :Article No. 133, July 2014.
- [Fau14] X. Faure. *Vers un modèle humain multi-échelles temps réel*. PhD thesis, Université de Lyon, 2014.
- [FB03] P. J. Frey and H. Borouchaki. Surface meshing using a geometric error estimate. *International Journal for Numerical Methods in Engineering*, 58(2) :227–245, 2003.
- [FDA05] Clément Forest, Hervé Delingette, and Nicholas Ayache. Removing Tetrahedra from manifold tetrahedralisation : application to real-time surgical simulation. *Medical Image Analysis*, 9(2) :113–122, 2005.
- [Fre03] Pascal Frey. Génération et adaptation de maillages de surfaces à partir de données anatomiques discrètes. Rapport de recherche RR-4764, INRIA, 2003.
- [GCMS00] Fabio Ganovelli, Paolo Cignoni, Claudio Montani, and Roberto Scopigno. A multiresolution model for soft objects supporting interactive cuts and lacerations. *Computer Graphics Forum*, 19(3) :271–281, 2000.
- [Gé06] O. Génevaux. *Simulation de liquides à l'aide des équations de Navier-Stokes, et visualisation, à destination de l'infographie*. PhD thesis, Université Louis Pasteur - Strasbourg I, Nov 2006.
- [HML00] G. Hirota, R. Maheshwari, and M.C. Lin. Fast volume-preserving free-form deformation using multi-level optimization. *Computer-Aided Design*, 32(8–9) :499 – 512, 2000.
- [HPH96] Dave Hutchinson, Martin Preston, and Terry Hewitt. Adaptive refinement for mass/spring simulations. In *Proc. of In 7th Eurographics Workshop on Animation and Simulation*, pages 31–45. Springer-Verlag, 1996.
- [JAD⁺12a] Thomas Jund, Ali Allaoui, Emmanuelle Darles, Xavier Skapin, Philippe Meseure, and Annie Luciani. Association Mouvement/Géométrie pour représentations volumiques. In *Actes de l'AFIG 2012*, pages 141–150, Calais, France, November 2012.

- [JAD⁺12b] Thomas Jund, Ali Allaoui, Emmanuelle Darles, Xavier Skapin, Philippe Meseure, and Annie Luciani. Mapping Volumetric Meshes to Point-based Motion Models. pages 11–20, Darmstadt, Germany, 2012. Eurographics Association.
- [JK09] Lenka Jeřábková and Torsten Kuhlen. Stable cutting of deformable objects in virtual environments using xfem. *IEEE Comput. Graph. Appl.*, 29(2) :61–71, March 2009.
- [JP99] Doug L. James and Dinesh K. Pai. Artdefo : Accurate real time deformable objects. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '99*, pages 65–72, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [KCS98] L. Kobbelt, S. Campagna, and H. P. Seidel. A general framework for mesh decimation. In *Graphics Interface*, pages 43–50, 1998.
- [KHS01] Kolja Kähler, Jörg Haber, and Hans-Peter Seidel. Geometry-based muscle modeling for facial animation. In *Proceedings of Graphics Interface 2001, GI '01*, pages 37–46, Toronto, Ont., Canada, Canada, 2001. Canadian Information Processing Society.
- [KKH⁺97] U. G. Kühnapfel, Ch. Kuhn, M. Hubner, H.-G. Krumm, H. Maass, and B. Neisius. The karlsruhe endoscopic surgery trainer as an example for virtual reality in medical education. *Minimally Invasive Therapy and Allied Technologies*, 6(2) :122–125, 1997.
- [Kra08] P. Kraemer. *Modèles topologiques pour la multirésolution*. 2008.
- [Kü00] U.G. Kühnapfel. Endoscopic surgery training using virtual reality and deformable tissue simulation, 2000.
- [LBOK13] Tiantian Liu, Adam W. Bargteil, James F. O'Brien, and Ladislav Kavan. Fast simulation of mass-spring systems. *ACM Transactions on Graphics*, 32(6) :209 :1–7, November 2013. Proceedings of ACM SIGGRAPH Asia 2013, Hong Kong.
- [Lie89] P. Lienhardt. Subdivisions of n-dimensional spaces and n-dimensional generalized maps. Juin 1989. ACM Symposium on Computational Geometry, Saarbrücken, Allemagne, pp. 228-236.
- [Lie91] P. Lienhardt. Topological models for boundary representation : a comparison with n-dimensional generalized maps. *Comput. Aided Des.*, 23(1) :59–82, 1991.
- [Lie94] P. Lienhardt. N-dimensional generalized combinatorial maps and cellular quasi-manifolds. 1994. International Journal on Computational Geometry and Applications, Vol. 4, n° 3, pp. 275-324.
- [LPC95] Jean Louchet, Xavier Provot, and David Crochemore. Evolutionary identification of cloth animation models, 1995.
- [LSH07] B.A. Lloyd, G. Székely, and M. Harders. Identification of spring parameters for deformable object simulation. *IEEE Trans. on Visualization and Computer Graphics*, 13(5) :1081–1094, Sept-Oct 2007.
- [LTW95] Yuencheng Lee, Demetri Terzopoulos, and Keith Waters. Realistic modeling for facial animation. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '95*, pages 55–62, New York, NY, USA, 1995. ACM.



- [Män87] Martti Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Inc., New York, NY, USA, 1987.
- [Mar06] Maud Marchal. *Modélisation des tissus mous dans leur environnement pour l'aide aux gestes médico-chirurgicaux*. PhD thesis, Université Joseph-Fourier-Grenoble I, 2006.
- [MDM⁺02] Matthias Müller, Julie Dorsey, Leonard McMillan, Robert Jagnow, and Barbara Cutler. Stable real-time deformations. In *Proceedings of the 2002 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '02, pages 49–54, New York, NY, USA, 2002. ACM.
- [MDS10a] P. Meseure, E. Darles, and X. Skapin. A Topology-Based Mass/Spring System. In *Proc. of CASA'2010 (short papers)*, St Malo (F), June 2010.
- [MDS10b] P. Meseure, E. Darles, and X. Skapin. Topology-based Physical Simulation. In *Proc. of VRIPHYS 2010*, pages 1–10, Copenhagen, Denmark, November 2010.
- [Mil88] Gavin S. P. Miller. The motion dynamics of snakes and worms. *SIGGRAPH Comput. Graph.*, 22(4) :169–173, June 1988.
- [MNB⁺06] Edoardo Mazza, Alessandro Nava, Margit Bauer, Raimund Winter, Michael Bajka, and Gerhard A. Holzapfel. Mechanical properties of the human uterine cervix : An in vivo study. *Medical Image Analysis*, 10(2) :125 – 136, 2006.
- [mok] Moka. <http://moka-modeller.sourceforge.net/>.
- [Mor07] Richard Moreau. *Le simulateur d'accouchement BirthSIM : un outil complet pour la formation sans risque en obstétrique*. PhD thesis, INSA de Lyon, 2007.
- [MSVCS04] Wouter Mollemans, Filip Schutyser, Johan Van Cleynenbreugel, and Paul Suetens. Fast soft tissue deformation with tetrahedral mass spring model for maxillofacial surgery planning systems. In Christian Barillot, David R. Haynor, and Pierre Hellier, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2004*, volume 3217 of *Lecture Notes in Computer Science*, pages 371–379. Springer Berlin Heidelberg, 2004.
- [NCC10] S. Natsupakpong and M. Cenk Çavusoglu. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models*, 72(6) :61–73, 2010.
- [Nes08] Matthieu Nesme. *Milieu mécanique déformable multirésolution pour la simulation interactive*. These, Université Joseph-Fourier - Grenoble I, June 2008. page web associée + vidéos : <http://www-evasion.imag.fr/Membres/Matthieu.Nesme/these/>.
- [NFP06] Matthieu Nesme, François Faure, and Yohan Payan. Hierarchical Multi-Resolution Finite Element Model for Soft Body Simulation. In *2nd Workshop on Computer Assisted Diagnosis and Surgery*, Santiago de Chile, Chile, 2006.
- [NMK⁺06] A. Nealen, M. Müller, R. Keiser, E. Boxerman, and M. Carlson. Physically based deformable models in computer graphics. *Computer Graphics Forum*, 25(4) :809–836, 2006.
- [NNUF04] R. Nogami, H. Noborio, F. Ujibe, and H. Fujii. Precise deformation of rheologic object under MSD models with many voxels and calibrating parameters. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 2, 2004.

- [NPF05] M. Nesme, Y. Payan, and F. Faure. Efficient, Physically Plausible Finite Elements. In *Eurographics'05 (short papers)*, Dublin (IRL), 2005.
- [PB81] S.M. Platt and N.I. Badler. Animating facial expressions. *ACM SIGGRAPH computer graphics*, 15(3) :245–252, 1981.
- [PBP96] Emmanuel Promayon, Pierre Baconnier, and Claude Puech. Physically based deformation constrained in displacements and volume. In *Proceedings of Eurographics'96*, Oxford, 1996. BlackWell Publishers.
- [Pic01] Guillaume Picinbono. *Modèles géométriques et physiques pour la simulation d'interventions chirurgicales*. These, Université Nice Sophia Antipolis, February 2001.
- [Pou92] J.P. Pouget. *Modélisation géométrique*. IREM Paris-Nord, 1992.
- [Pro95] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Proceedings of Graphics Interface 95*, pages 147,154, Toronto, 1995. Canadian Human-Computer Communications Society.
- [Pro97] E. Promayon. *Modélisation Et Simulation de La Respiration*. PhD thesis, Université Joseph Fourier, November 1997.
- [PT97] Les Piegl and Wayne Tiller. *The NURBS Book (2Nd Ed.)*. Springer-Verlag New York, Inc., New York, NY, USA, 1997.
- [RA77] H.B. Requicha A.A.G., Voeicker. Constructive solid geometry. Technical report, University of Rochester, 1977.
- [RA78] R.B. Requicha A.A.G., Tilove. Mathematical foundations of constructive solid geometry : General topology of closed regular sets. Technical report, University of Rochester, 1978.
- [RLGB⁺13] Chrystèle Rubod, Pauline Lecomte-Grosbras, Mathias Brieu, Géraldine Giraudet, Nacim Betrouni, and Michel Cosson. 3d simulation of pelvic system numerical simulation for a better understanding of the contribution of the uterine ligaments. *International Urogynecology Journal*, 24(12) :2093–2098, 2013.
- [SCB⁺] Patrick Schiavone, F. Chassat, T. Boudou, Emmanuel Promayon, F. Valdivia, and Yohan Payan. In vivo measurement of human brain elasticity using a light aspiration device. *Medical Image Analysis*, (4) :673–678.
- [Sch61] J. W. Schmidt. G. dahlquist, stability and error bounds in the numerical integration of ordinary differential equations. 85 s. stockholm 1959. k. tekniska högskolans handlingar. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 41(6) :267–268, 1961.
- [sdLR12] Cahier spécial de La Recherche. Les supercalculateurs relèvent le défi. Novembre 2012.
- [SHGS06] D. Steinemann, M. Harders, Markus Gross, and G. Szekely. Hybrid cutting of deformable solids. In *Virtual Reality Conference, 2006*, pages 35–42, March 2006.
- [SLF08] Andrew Selle, Michael Lentine, and Ronald Fedkiw. A mass spring model for hair simulation. *ACM Trans. Graph.*, 27(3) :64 :1–64 :11, August 2008.
- [SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. *SIGGRAPH Comput. Graph.*, 20(4) :151–160, August 1986.



- [SPP10] P. Schiavone, E. Promayon, and Y. Payan. LASTIC : A light aspiration device for in vivo soft Tissue characterization. In F. Bello and S. Cotin, editors, *Bio-medical Simulation : 5th International Symposium, ISBMS 2010*, volume 5958 of *Lecture Notes in Computer Science*, pages 1–10. Springer Verlag, January 2010.
- [STF⁺11] A Sala, G. Turini, M. Ferrari, F. Mosca, and V. Ferrari. Integration of bio-mechanical parameters in tetrahedral mass-spring models for virtual surgery simulation. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 4550–4554, Aug 2011.
- [SVO11] Gaizka San-Vicente-Otamendi. *Designing deformable modles of soft tissue for virtual surgery planning and simulation using the mass-spring system*. PhD thesis, Université de Navarre, 2011.
- [TBHF03] J. Teran, S. Blemker, V. Ng Thow Hing, and R. Fedkiw. Finite volume methods for the simulation of skeletal muscle. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '03*, pages 68–74, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.
- [TGG00] Matthias Teschner, Sabine Girod, and Bernd Girod. Direct computation of nonlinear soft-tissue deformation, 2000.
- [THMG04] M. Teschner, B. Heidelberger, M. Muller, and Markus Gross. A versatile and robust model for geometrically complex deformable solids. In *Computer Graphics International, 2004. Proceedings*, pages 312–319, June 2004.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *SIGGRAPH Comput. Graph.*, 21(4) :205–214, August 1987.
- [TT94] Xiaoyuan Tu and Demetri Terzopoulos. Artificial fishes : Physics, locomotion, perception, behavior. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '94*, pages 43–50, New York, NY, USA, 1994. ACM.
- [TW90] Demetri Terzopoulos and Keith Waters. Physically-based facial modelling, analysis, and animation. *The Journal of Visualization and Computer Animation*, 1(2) :73–80, 1990.
- [UCB13] Lionel Untereiner, David Cazier, and Dominique Bechmann. n-dimensional multiresolution representation of subdivision meshes with arbitrary topology. *Graph. Models*, 75(5) :231–246, September 2013.
- [Unt13] Lionel Untereiner. *Représentation des maillages multirésolutions : application aux volumes de subdivision*. PhD thesis, Université de Strasbourg, Nov 2013.
- [Van98] Allen Van Gelder. Approximate simulation of elastic membranes by triangulated spring meshes. *Journal of Graphics Tools*, 3(2) :21–42, 1998.
- [VB05] J. Villard and H. Borouchaki. Adaptive meshing for cloth animation. *Eng. with Comput.*, 20(4) :333–341, August 2005.
- [Vil11] Pierre-Frédéric Villard. *Simulation du Mouvement Pulmonaire pour un Traitement Oncologique - Application à la Radiothérapie et à l'Hadronthérapie*. Editions universitaires europeennes, May 2011. ISBN-13 : 978-613-1-56604-2 ISBN-10 :6131566046.
- [Voe77] Requicha A.A.G. Voeicker, H.B. Geometric modelling of mechanical parts and processes. Technical report, University of Rochester, 1977.

- [Wat87] Keith Waters. A muscle model for animation three-dimensional facial expression. *SIGGRAPH Comput. Graph.*, 21(4) :17–24, August 1987.
- [WGG99] Brian Wyvill, Andrew Guy, and Eric Galin. Extending the csg tree. warping, blending and boolean operations in an implicit surface modeling system. *Computer Graphics Forum*, 18(2) :149–158, 1999.
- [WMW86] Geoff Wyvill, Craig McPheeters, and Brian Wyvill. Data structure for soft objects. *The Visual Computer*, 2(4) :227–234, 1986.
- [WT91] Keith Waters and Demetri Terzopoulos. Modelling and animating faces using scanned data. *The Journal of Visualization and Computer Animation*, 2(4) :123–128, 1991.
- [WvO96] Brian Wyvill and Kees van Overveld. Polygonization of implicit surfaces with constructive solid geometry. *International Journal of Shape Modeling*, 2(4) :257–274, 1996.
- [WWD14] Jun Wu, Rüdiger Westermann, and Christian Dick. Physically-based simulation of cuts in deformable bodies : A survey. In *Eurographics 2014 State-of-the-Art Report*, pages 1–19, Strasbourg, France, 2014. Eurographics Association.
- [Zar03] Florence Zara. *Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2003.
- [ZGF07] D. Zerbato, S. Galvan, and P. Fiorini. Calibration of mass spring models for organ simulations. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 370–375. IEEE, 2007.
- [ZGHX05] Shaoting Zhang, Lixu Gu, Pengfei Huang, and Jianfeng Xu. Real-time simulation of deformable soft tissue based on mass-spring and medial representation. In Yanxi Liu, Tianzi Jiang, and Changshui Zhang, editors, *Computer Vision for Biomedical Image Applications*, volume 3765 of *Lecture Notes in Computer Science*, pages 419–426. Springer Berlin Heidelberg, 2005.

