



HAL
open science

Modélisation et simulation à base de règles pour l'animation physique

Fatma Ben Salah

► **To cite this version:**

Fatma Ben Salah. Modélisation et simulation à base de règles pour l'animation physique. Synthèse d'image et réalité virtuelle [cs.GR]. Université de Poitiers (France), 2018. Français. NNT: . tel-03192709v1

HAL Id: tel-03192709

<https://hal.science/tel-03192709v1>

Submitted on 18 Nov 2019 (v1), last revised 8 Apr 2021 (v2)

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.

THESE

Pour l'obtention du Grade de

DOCTEUR DE L'UNIVERSITE DE POITIERS

(Faculté des Sciences Fondamentales et Appliquées)
(Diplôme National - Arrêté du 25 mai 2016)

École Doctorale (n° 521) : Sciences et Ingénierie pour l'Information, Mathématiques

Secteur de Recherche : Informatique graphique

Présentée par :

Fatma BEN SALAH

Modélisation et simulation à base de règles pour l'animation physique

Directeur de thèse : Philippe MESEURE

Co-encadrants de thèse : Agnès ARNOULD et Hakim FERRIER BELHAOUARI

Soutenue le : 13/11/2018

Devant la Commission d'Examen

JURY

David CAZIER	Professeur, ICUBE, Strasbourg	Rapporteur
Guillaume DAMIAND	Directeur de recherche, HDR, LIRIS, Lyon	Rapporteur
Annie LUCIANI	Ingénieure de recherche, ICA, Grenoble	Examineur
Olivier TERRAZ	Professeur, XLIM, Limoges	Examineur
Philippe MESEURE	Professeur, XLIM, Poitiers	Directeur de thèse
Agnès ARNOULD	Maitre de conférences, XLIM, Poitiers	Co-encadrante

Remerciements

Il m'est agréable d'adresser mes sincères remerciements à tous ceux qui ont contribué de près ou de loin à l'aboutissement de cette thèse de doctorat. Tout d'abord, je voudrais remercier les personnes qui m'ont proposés le sujet de thèse et qui m'ont dirigé tout au long de ces années d'étude. J'exprime ma gratitude à mon directeur de thèse Monsieur Philippe MESEURE pour la confiance qu'il me témoigne, pour ses encouragements, pour sa contribution quotidienne à ce travail, pour ses précieux conseils, sa compréhension et son aide lors de la rédaction de ma thèse. Je présente aussi mes sincères remerciements à mes deux encadrants, Monsieur HAKIM FERRIER BELHAOUARI et Madame Agnès ARNOULD pour leurs conseils judicieux, leurs remarques pertinentes et le partage de leurs nombreuses connaissances dans ce qui concerne les modèles topologiques et la langage à base de règles Jerboa lors de nos discussions, qui m'ont aidé à l'accomplissement de mon travail.

Mes vifs remerciement vont également à tous les membres du jury de thèse qui ont accepté d'évaluer ce travail, pour l'honneur qu'ils m'accordent par leur présence pour ma soutenance de thèse.

Je remercie également tous les membres du laboratoire XLIM permanents et non permanents avec qui j'ai travaillé et échangé. Un merci particulier à mes collègues thésards et amis, mes amis de Poitiers (Ahlem, olfa, Marwa, Hajer, Lahcen, Hamza, Badran, Aymen).

Je tiens à remercier également tout les personnels du programme d'échange Erasmus Mundus - Al Idrisi entre l'Europe et le Maghreb, en particulier Monsieur Otis PIOCHE qui est le coordinateur à l'université de Poitiers, car cette thèse est partiellement financée par ce bourse.

Mes plus profonds remerciement vont à mes parents, ma belle-mère, mes sœurs et mes frères qui m'ont toujours soutenu, encouragé et aidé toutes au long de mes années d'études. Enfin, je ne peux terminer sans adresser un merci particulier à mon mari Maher pour son aide et son soutien moral qui m'a été essentiel durant ces années.

Table des matières

1	Introduction	1
1.1	Contexte et motivations	1
1.2	Contributions	2
1.3	Organisation du manuscrit	3
1.4	Publications	4
I	État de l’art	5
2	Simulation de corps déformables	7
2.1	Introduction	7
2.2	Modèles de simulation physique	8
2.2.1	Notion de base de la mécanique	8
2.2.2	Méthodes de modélisation	16
2.3	Animation Physique	33
2.3.1	Méthodes d’intégration	34
2.3.2	Boucle de simulation	36
2.4	Modifications Topologiques	37
2.4.1	Découpe	38
2.4.2	Fractures/déchirures	41
2.4.3	Adaptation de maillage	45
2.4.4	Mises à jour du modèle mécanique	50
2.5	Conclusion	51
3	Modèles topologiques	53
3.1	Introduction	53
3.2	Modélisation géométrique à base topologique	53
3.3	Modèles topologiques cellulaires	56
3.3.1	Modèles de représentation explicite de cellules	56
3.3.2	Modèles de représentation implicite de cellules	60
3.4	Modélisation à base de règles : Jerboa	68
3.4.1	Le langage Jerboa	69
3.4.2	Exemples de règles : création d’une surface en damier	81
3.4.3	Autre exemples de règles	86
3.5	Conclusion	90
4	Synthèse : Modélisation mécanique à base topologique	91
4.1	Introduction	91
4.2	Lien entre modèle topologique et modèle physique	92
4.3	Simulation à base de maillages tétraédriques	93

4.4	Simulation du modèle masses/ressorts pour maillages rectangulaires ou hexaédriques	95
4.5	Simulation avec maillages homogènes tétraédriques ou hexaédriques	97
4.6	Utilité des modèles topologiques en simulation mécanique	98
4.7	Conclusion	99
II	Contribution	101
5	Modélisation de corps physiques basés topologie	103
5.1	Hypothèses et modèle support	104
5.1.1	Hypothèses	104
5.1.2	Modèle support	105
5.2	Modélisation des interactions	107
5.2.1	Méthode générale	108
5.2.2	Modèles mécaniques discrets 2D	109
5.2.3	Modèles mécaniques discrets 3D	118
5.2.4	Modèles mécaniques continus	120
5.3	Adaptation de la modélisation pour les cartes orientées	127
5.4	conclusion	129
6	Simulation avec le langage à base de règles Jerboa	131
6.1	Introduction	131
6.2	Initialisation du modèle	132
6.2.1	Utilisation d'un maillage décrit dans un fichier	132
6.2.2	Création d'un maillage	140
6.3	Boucle de simulation	146
6.3.1	Règles sur la composante connexe vs règle sur un motif	147
6.3.2	Règles de calcul des forces	148
6.3.3	Résolution des équations du mouvement	164
6.4	Optimisation des calculs	165
6.4.1	Recensement des redondances	166
6.4.2	Analyse des redondances	168
6.4.3	Mise en œuvre des optimisations	170
6.5	Résultats	173
6.6	Conclusion	177
7	Transformation topologiques	179
7.1	Introduction	179
7.2	Principe	180
7.3	Modification topologiques	183
7.3.1	Séparation des éléments	184
7.3.2	Suppression d'éléments	186
7.3.3	Éclatement d'un sommet	187

7.3.4	Suppression d'arête	188
7.4	Critères	189
7.4.1	Critères atomiques	189
7.4.2	Critères cumulatifs	191
7.4.3	Critères dédiés à la séparation d'éléments	192
7.5	Exemples de résultats	194
7.6	Conclusion	198
8	Conclusion et perspectives	201
	Bibliographie	205

Table des figures

2.1	Exemples d'interactions.	9
2.2	Exemples de combinaison de liaisons.	9
2.3	Forces appliquées par un ressort.	10
2.4	Déplacement d'un point de sa position au repos à une position déformé (issue de [Nes08])	12
2.5	Exemples de déformation d'un objet.	12
2.6	Le tenseur des contraintes.	13
2.7	Loi de comportement élastique (issue de [Der12]).	14
2.8	Exemples de modélisation avec le modèle masse/interaction (issue de [CLF90]).	18
2.9	Le principe du <i>shape matching</i> (issue de [MHTG05])	19
2.10	Processus de la simulation numérique en utilisant une méthode avec maillage géométrique	19
2.11	Masse d'un sommet.	20
2.12	Représentation d'un tissu par un maillage dont les sommets sont les particules et les arêtes sont les ressorts (issu de [Zar03]).	20
2.13	Modèle de tissu de Provot [Pro95].	21
2.14	Ressort angulaire.	22
2.15	Ressorts de cisaillement en 3D.	22
2.16	Calcul de raideurs des ressorts pour des mailles rectangulaires [FZDJ14].	23
2.17	Exemples d'éléments linéaires utilisés par la méthode des éléments finis.	25
2.18	Exemple d'éléments finis avec des fonctions d'interpolation linéaire et quadratique.	26
2.19	Résultats de déformation et de découpe d'un foie avec la méthode masses/tenseurs (images issues de [CDA99]).	29
2.20	Décomposition de matrice de raideurs.	29
2.21	Les tenseurs de raideur qui permettent de calculer les forces appliquées sur chaque nœud.	30
2.22	Simulation de comportement de deux poutres (image issue de [NPF05]).	31
2.23	Principe de la décomposition QR.	33
2.24	Les étapes de la boucle de simulation.	36
2.25	Les étapes de la boucle de simulation avec transformations topologiques.	38
2.26	Découpe avec suppression d'élément (face en 2D).	38
2.27	Découpe d'une plaque par suppression de tétraèdre (issu de [CDA00]).	39
2.28	Découpe avec séparation des éléments.	39
2.29	Division d'un élément (issue de [WWD14]).	40
2.30	Découpe avec duplication des éléments	40

2.31	Exemples de résultats des objets avec la méthode de Norton (images issues de [NTB ⁺ 91])	42
2.32	Principe d'éclatement d'un sommet (cas 2D).	44
2.33	Basculement d'arête.	46
2.34	Modification d'éléments avec le raffinement $\sqrt{3}$ (cas 2D).	46
2.35	subdivision d'un tétraèdre (1-4 split).	47
2.36	Modification d'éléments avec le raffinement $\sqrt{3}$ (cas 3D).	47
2.37	Exemples de résultat des travaux de Iben <i>et al.</i> [IO09]	48
2.38	Exemples de résultat des travaux de Hutchinson <i>et al.</i> [HPH96]	48
2.39	Exemples de changement de résolution lors de fractures et déchirures.	49
2.40	Exemples de résultat des travaux de Dubunne <i>et al.</i> [DDCB01]	49
3.1	Exemple de représentation d'un objet par arbre CSG.	55
3.2	Une représentation par les bords d'un objet.	55
3.3	Modèle simplicial et modèle cellulaire.	56
3.4	Graphe d'incidence d'un objet 2D.	57
3.5	Graphe d'incidence ambigu.	57
3.6	Exemple d'un fichier OBJ.	58
3.7	Représentation d'un objet 2D avec la structure des arêtes ailées.	60
3.8	Représentation d'un objet 2D avec la structure des demi-arêtes.	61
3.9	Décompositions successives d'un objet géométrique 2D pour trouver la carte orientée correspondante.	62
3.10	Exemple de quasi-variétés.	62
3.11	Exemples de non-variétés.	63
3.12	Exemple d'objets orientable et non orientable.	63
3.13	Décompositions successives d'un objet géométrique 2D.	64
3.14	Conventions graphiques.	65
3.15	Exemples des orbites adjacentes au brin e	65
3.16	Représentation d'un objet 2D avec des plongements.	67
3.17	Modélisation d'une plante avec un L-system.	69
3.18	Exemple d'application d'une règle de transformation de graphe.	70
3.19	Ajout de plongement pour calculer la position du sommet à insérer.	71
3.20	Règle de couture de deux faces le long d'une arête.	71
3.21	Exemple d'application de la règle de couture de deux faces.	72
3.22	Règle de triangulation d'une face.	72
3.23	Application de la triangulation d'une face.	73
3.24	Instanciation du nœud n_0 de la règle de triangulation d'une face.	73
3.25	Instanciation du nœud n_2 de la règle de triangulation d'une face.	74
3.26	Instanciation du nœud n_1 de la règle de triangulation d'une face.	74
3.27	Ajout des liaisons explicites entre les instances.	75
3.28	Face rectangulaire triangulée.	75
3.29	Règle de triangulation et coloration d'une face.	76
3.30	Exemple d'application de la règle 3.29.	76

3.31 Règle de changement de position d'un sommet d'un quadrilatère et son application.	78
3.32 Règle de changement de la forme et du couleur d'un quadrilatère et son application.	79
3.33 Schéma d'utilisation de Jerboa.	79
3.34 Interface de l'éditeur de Modeleur de Jerboa.	80
3.35 Interface de modification d'une règle.	81
3.36 Création d'un sommet.	82
3.37 Règle de création d'une arête.	82
3.38 Principe de création d'une face.	83
3.39 Règle de création d'une face.	83
3.40 Résultat de l'application de trois règles successives pour créer une face.	84
3.41 Subdivisions successives d'une surface rectangulaire.	85
3.42 Principe de la subdivision avec Jerboa.	85
3.43 Subdivision d'un triangle avec Jerboa.	85
3.44 Règle de subdivision des faces d'un objet connexe.	86
3.45 Face subdivisée.	86
3.46 Règle de création d'un triangle.	87
3.47 Règle de suppression d'une face isolée	87
3.48 Règle de découpe d'une arête	87
3.49 Application successive de la règle d'isolement puis de la suppression d'une face.	88
3.50 Règle d'isolement et suppression d'une face	88
3.51 Règle de mise à jour des positions des sommets dans toute une composante connexe 3D.	89
3.52 Exemple d'application de règle de rotation de toute la composante connexe.	89
4.1 Résultats de transformation topologique de Meseure <i>et al.</i>	93
4.2 Exemples de résultats de l'approche de Fléchon <i>et al.</i>	96
4.3 Convention de stockage de brins dans des tableaux	97
5.1 Processus de modélisation.	104
5.2 Les divers plongements masse	106
5.3 Plongements liés aux particules.	107
5.4 Interactions à modéliser.	107
5.5 Exemple de modélisation.	109
5.6 Maille élémentaire (rectangulaire).	110
5.7 Sous-graphe et orbites supports pour les ressorts d'élongation.	110
5.8 Plongements associés à un ressort d'élongation.	111
5.9 Sous-graphe et orbites supports des ressorts de cisaillement surfacique isotrope.	112
5.10 Sous-graphe et orbites supports pour les ressorts de cisaillement surfacique anisotrope.	113

5.11	Plongements associés aux ressorts de cisaillement surfacique isotrope.	113
5.12	Plongements associés à un ressort de cisaillement surfacique anisotrope.	114
5.13	Orbite support d'un ressort angulaire.	115
5.14	Sous-graphe et orbites supports pour les ressorts angulaires.	115
5.15	Plongements associés à un ressort angulaire.	116
5.16	Orbites de plongement d'un ressort linéaire de Provot.	117
5.17	Sous-graphe et orbites supports utilisés pour les ressorts linéaires de Provot.	117
5.18	Plongements associés à un ressort linéaire de Provot.	118
5.19	Sous-graphe et orbites supports pour les ressorts de cisaillement volumique isotrope.	119
5.20	Sous-graphe et orbites supports pour les ressorts de cisaillement volumique anisotrope.	119
5.21	Sous-graphe et orbites de plongement pour la méthode des éléments finis linéaires sur un tétraèdre.	121
5.22	Numérotation des sommets d'un tétraèdre.	122
5.23	Sous-graphe et orbites de plongement des tenseurs.	124
5.24	Décomposition d'une matrice de raideur dans le cas des hexaèdres.	125
5.25	Orbites de stockage pour le co-rotationnel en 2D.	125
5.26	Identification de l'ordre des sommets en 2D.	126
5.27	Objet représenté avec une carte orientée.	127
5.28	Modélisation des ressorts linéaires de Provot	128
6.1	Exemple de fichier GMSH.	132
6.2	Règle générale du peuplement.	133
6.3	Calcul des longueurs des ressorts de cisaillement.	134
6.4	Règle de numérotation des sommets d'un élément 3D.	135
6.5	Règle de calcul de matrice de rigidité pour un tétraèdre.	137
6.6	Règle de calcul des tenseurs de rigidité pour un tétraèdre isolé.	139
6.7	Règle de marquage du premier sommet d'un élément 2D.	140
6.8	Règle d'initialisation de la matrice de comportement.	140
6.9	Règle de calcul de la matrice du raideur sur un triangle en 2D.	141
6.10	Règle de création d'une particule.	141
6.11	Règle de création d'une arête entre deux particules.	142
6.12	Règle de création d'une face.	143
6.13	Règle de subdivision d'une face rectangulaire.	144
6.14	Adaptation des valeurs des plongements après subdivision.	145
6.15	Principe de calcul de raideur des ressorts lors d'une subdivision.	145
6.16	Règle de calcul des forces des ressorts d'élongation.	151
6.17	Règle de calcul des forces des ressorts de cisaillement surfacique isotrope.	151
6.18	Règle de calcul des forces des ressorts de cisaillement surfacique anisotrope.	152

6.19 Règle de calcul des forces des ressorts de cisaillement volumique isotrope.	153
6.20 Filtrage explicite d'un parallélépipède.	154
6.21 Règle de calcul des forces des ressorts de cisaillement volumique anisotrope.	155
6.22 Règle de calcul des forces des ressorts linéaires de Provot.	156
6.23 Règle de calcul des forces des ressorts angulaires (les cadres pointillés bleus ne font pas partie de la règle, mais permettent, dans le texte, de désigner les sommets par une lettre et de reconnaître les brins qui les constituent).	157
6.24 Règle de calcul des forces en chaque sommet du tétraèdre.	161
6.25 Règle de calcul des forces appliquées par un sommet sur lui-même.	162
6.26 Règle de calcul des forces appliquées par un sommet de volume sur un autre.	163
6.27 Règle de calcul des forces appliquées sur les sommets par le corotationnel 2D.	164
6.28 Règles de mise à jour des accélérations, vitesses et positions de particules.	165
6.29 Mise en évidence de redondance pour les ressorts d'élongation.	166
6.30 Mise en évidence des redondances pour les ressorts de cisaillement surfacique anisotrope.	166
6.31 Mise en évidence de redondance pour les ressorts linéaires de Provot.	167
6.32 Mise en évidence des redondances pour les ressorts de cisaillement surfacique en 3D.	167
6.33 Ressort d'élongation cumulé	170
6.34 Règle de calcul des forces des ressorts d'élongation cumulés.	170
6.35 Règle de calcul des forces des ressorts de cisaillement surfacique cumulés en 3D.	171
6.36 Règles de calcul des forces des tenseur cumulés.	172
6.37 Simulations de différents types de tissus avec le modèle masses/ressorts.	173
6.38 Simulations de différents types d'objet 3D avec modèle masse/ressort.	174
6.39 Simulations d'une barre tétrédrique 3D avec différents modèle continu.	175
6.40 Simulation d'un tissu avec la méthode des éléments finis.	175
7.1 Algorithme de simulation d'une fracture/déchirement.	180
7.2 Principe de l'opération de décousure en 2D.	181
7.3 Principe de l'opération de décousure en 3D.	181
7.4 Influence de la décousure α_2 sur les plongements mécaniques.	182
7.5 Règle de séparation de deux faces adjacentes.	184
7.6 Règle de séparation de deux volumes adjacents.	185
7.7 Règle de suppression d'un volume isolé.	187
7.8 Règle du calcul de signes des volumes entourant un sommet (cas 3D).	187
7.9 Règle du calcul des tenseurs de contrainte en chaque volume.	190
7.10 Règle de calcul de l'équation du plan de fracture/déchirure.	190

7.11	Calcul de forces internes appliquées sur une arête.	193
7.12	Exemples de déchirures d'un tissu à maille triangulaire avec différents critères.	194
7.13	Exemples de déchirure d'un tissu maillé avec des quadrilatères. . . .	195
7.14	Exemples de fracture d'un objet 3D simulé en masses/ressorts avec différents critères.	196
7.15	Exemple de fracture d'un objet à mailles tétraédriques avec le critère de cumul de force pour séparer les volumes.	196
7.16	Simulation de modèle tétraédrique par l'approche du co-rotationnel et déchirure basée sur les forces élémentaires et l'éclatement de sommets.	197
7.17	Simulation d'un tissu avec le critère global sur l'arête.	197
7.18	Histogramme de simulation d'un tissu 2D avec des critères sur l'arête.	198

Introduction

Sommaire

1.1	Contexte et motivations	1
1.2	Contributions	2
1.3	Organisation du manuscrit	3
1.4	Publications	4

1.1 Contexte et motivations

La simulation physique des objets déformables a une grande importance pour diverses applications dans le domaine de l'informatique graphique telles que la réalité virtuelle, la simulation de phénomènes naturels, la simulation chirurgicale et les jeux vidéo. Pour cela, il existe dans la littérature de nombreuses méthodes basées sur les lois de la physique pour la simulation des déformations, par exemple, la méthode des masses/ressorts ou la résolution des équations de la mécanique continue par la méthode des éléments finis. Ces méthodes diffèrent par leur degré de précision et leur réalisme. Mais il est difficile de dire qu'une méthode est meilleure qu'une autre, car cette constatation dépend du contexte applicatif et du type de matériaux à simuler.

De plus, pendant les quinze dernières années, la simulation des transformations topologiques comme les fractures et les découpes est devenue un domaine très étudié en informatique graphique. Souvent, pour mieux gérer ces modifications topologiques, les chercheurs utilisent une structure topologique rudimentaire. Il faut effectuer plusieurs vérifications sur l'objet après sa modification pour garantir sa validité topologique. Cette étape se révèle assez lourde et fastidieuse, à cause du nombre de cas dégénérés à vérifier. Pour cela, plusieurs travaux ont montré l'utilité d'un modèle topologique pour la simulation physique. Les modèles topologiques décrivent les maillages sous forme de cellules (sommets, arêtes, faces, volumes). Ils sont définis mathématiquement et permettent de garantir la validité topologique de l'objet après une transformation.

Des modèles qui combinent un modèle topologique avec un modèle mécanique ont été proposés dans la littérature. Néanmoins, aucun modèle autonome, qui stocke toutes les informations nécessaires à la simulation dans la structure topologique, n'a vraiment été proposé. Les modèles existants utilisent assez systématiquement une

structure externes en parallèle de la structure topologique. Or ceci nuit à la robustesse du modèle. En effet, si l'objet subit des modifications, le modèle topologique assure sa propre cohérence alors que rien ne permet de s'assurer de la cohérence des structures additionnelles. De plus, les divers modèles proposés sont peu généraux, alors que les modèles topologiques les plus évolués le sont : ils se définissent de façon similaire en dimension 2 et 3 (et au delà), ils autorisent des maillages hétérogènes et offrent une panoplie de transformations diverses de ces maillages. Il nous semble intéressant d'étendre cette généralité aux modèles mécaniques supportés par les modèles topologiques.

1.2 Contributions

Compte tenu des avantages que présentent les modèles topologiques, nous proposons dans cette thèse, de créer, pour l'animation physique, un modèle général. Ce modèle combine à la fois un modèle mécanique, dont on peut choisir les interactions de contrôle des déformations, avec un modèle topologique puissant, en l'occurrence les cartes généralisées. Cette approche évite les structures externes.

Nos contributions concernent deux axes principaux :

Le premier axe s'articule autour de la création d'un modèle général qui permet la simulation de déformations de différents objets. En effet, nous voulons formaliser la simulation physique d'objets maillés en exploitant le modèle topologique et, pour cela, nous proposons une méthode qui permet de modéliser n'importe quelle interaction physique en la stockant dans le modèle topologique et en exploitant les relations entre cellules pour calculer les forces qu'elle engendre et les appliquer aux entités concernées. Notre modèle permet de simuler des objets en différentes dimensions (2D/3D), qui se reposent sur différents types de maillages (triangulaires, rectangulaires, tétraédriques, hexaédriques, hétérogènes...) en utilisant différents modèles d'interactions. Afin de pouvoir rapidement prototyper nos propositions, nous avons choisi d'utiliser un langage à base de règles de transformations de graphes, qui s'est révélé particulièrement adapté à la définition et à la simulation de notre modèle.

Un second axe se concentre sur la simulation de transformations topologiques et se base également sur le langage à base de règles. En utilisant la modélisation proposée pour chaque interaction, nous proposons des transformations topologiques comme la déchirure et la fracture. Ces transformations se basent exclusivement sur l'opération élémentaire de séparation d'éléments. Nous montrons que notre modélisation permet de minimiser la mise à jour des informations mécaniques après l'application d'une telle opération. En outre, les déchirures/fractures se déclenchent lorsqu'un critère donné est vérifié. Il existe plusieurs sortes de critères, que nous avons implantés dans notre plate-forme. Il s'appliquent sur des lieux topologiques divers et peuvent enclencher différentes transformations topologiques, sous réserve qu'un lien identifiant le lieu de la transformation à partir du lieu où le critère est testé soit établi. Nous avons d'ailleurs proposé de nouveaux critères permettant de piloter directement la séparation des éléments du maillage, ce qui rend la prise en

charge des déchirures/fractures plus directe. Finalement, nous avons formalisé le processus (test du critère, choix du lieu de la transformation), grâce au langage à base de règles, puisqu'il permet dans une même règle de tester le critère et de décider de l'application de la transformation.

1.3 Organisation du manuscrit

Ce manuscrit est divisé en deux parties : dans la première partie, qui regroupe les chapitres 2, 3 et 4, nous nous focalisons sur l'état de l'art, respectivement, des modèles mécaniques, des modèles topologiques et des travaux qui combinent les deux. La deuxième partie de ce manuscrit, qui comprend les chapitres 5, 6 et 7, est consacrée à nos contributions.

Dans le chapitre 2, nous présentons la simulation de corps déformables. Dans un premier temps, nous introduisons les différents modèles mécaniques existants dans la littérature pour simuler des objets déformables. Ce chapitre permet également d'introduire les pré-requis indispensables pour l'animation physique, les étapes de la boucle de simulation physique et finalement un bref état de l'art sur les modifications topologiques lors de la simulation physique.

Le chapitre 3 de cette partie est consacré à l'étude de différents modèles topologiques existants dans la littérature, nous nous intéressons plus particulièrement aux modèles des cartes combinatoires et plus précisément au modèle des cartes combinatoires généralisées (G-cartes) que nous utilisons dans notre travail. Nous introduisons également, dans ce chapitre, le langage à base de règles Jerboa qui est basé sur le modèle topologique des G-cartes et qui est l'une des raisons de l'utilisation de ce modèle.

Le chapitre 4, le dernier de la première partie, est consacré aux travaux de la littérature qui ont combiné un modèle topologique avec un modèle physique. Ce chapitre discute également de l'utilité d'un modèle topologique pour la simulation physique.

Dans le chapitre 5, nous présentons la modélisation faite pour les différents types d'interactions implantées dans notre plate-forme. Nous commençons, tout d'abord, par présenter les hypothèses de notre modèle et les principes de son implantation. Ensuite, nous proposons une approche générale permettant d'inclure dans le modèle tout type d'interactions. Cette approche est ensuite mise en place pour des interactions très variées, ce qui nous permettra d'illustrer son efficacité.

Le chapitre 6 est dédié à la simulation du modèle. Nous commençons par son initialisation, qui nécessite des règles Jerboa pour sa construction, afin de fixer la géométrie (sauf dans le cas où le modèle provient d'un fichier) et les informations mécaniques. Ensuite nous nous intéressons aux règles créées avec Jerboa et qui permettent de simuler les différents types de maillage avec les différentes interactions. Nous poursuivons ce chapitre en proposant quelques optimisations de calcul qui permettent de coupler des interactions concernant les mêmes particules en une seule et d'effectuer leur calcul en une seule règle. Quelques résultats de simulation

de différents types d'objets achèvent ce chapitre.

Nous nous focalisons, dans le chapitre 7, sur les transformations topologiques et plus précisément la prise en charge des fractures/déchirures par le langage à base de règles. Dans ce chapitre nous commençons par décrire les différentes modifications topologiques implantées dans notre modèle. Ces modifications se basent toutes sur la séparation d'éléments. Nous montrons ainsi l'utilité de la modélisation que nous avons proposée puisqu'elle réduit le nombre de mises à jour à faire après un tel changement topologique. Ensuite, nous nous focalisons sur la description des critères qui déclenchent ces transformations et nous proposons une approche générale pour la prise en charge des fractures/déchirures. Finalement nous donnons quelques résultats de simulations de ces modifications sur différents objets.

1.4 Publications

Nos travaux ont donné lieu aux publications suivantes :

- Un premier papier intitulé « Simulation de modèles mécaniques à base topologique à l'aide de règles de transformation de graphes » publié à JFIG (Journées Françaises d'Informatique Graphique) en novembre 2016 à Grenoble.
- Un deuxième papier conférence intitulé « *A general physical-topological framework using rule-based language for physical simulation* » et publié à VISIGRAPP/GRAPP (*International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*), en février 2017 à Porto.
- Un troisième papier conférence intitulé « *A Modular Approach Based on Graph Transformation to Simulate Tearing and Fractures on Various Mechanical Models* » publié à WSCG (*International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*) en Mai 2017. Ce papier a été sélectionné parmi les meilleurs articles et à été publié dans le journal de WSCG.

Première partie

État de l'art

Simulation de corps déformables

Sommaire

2.1	Introduction	7
2.2	Modèles de simulation physique	8
2.2.1	Notion de base de la mécanique	8
2.2.2	Méthodes de modélisation	16
2.3	Animation Physique	33
2.3.1	Méthodes d'intégration	34
2.3.2	Boucle de simulation	36
2.4	Modifications Topologiques	37
2.4.1	Découpe	38
2.4.2	Fractures/déchirures	41
2.4.3	Adaptation de maillage	45
2.4.4	Mises à jour du modèle mécanique	50
2.5	Conclusion	51

2.1 Introduction

L'objectif de cette thèse est d'utiliser un langage à base de règles de transformations de graphes pour simuler les déformations et les transformations topologiques de corps solides déformables. Dans cette perspective, le but de ce chapitre est de fournir une vue globale à la fois sur les modèles pour la simulation des solides déformables et sur les approches simulant les modifications topologiques lors de l'animation d'un objet. Dans ce chapitre, nous étudions tout d'abord les deux approches possibles pour décrire le comportement des solides déformables, à savoir la mécanique discrète ou la mécanique continue. Nous nous focalisons ensuite sur des modèles permettant de mettre en équation l'une ou l'autre des approches afin que le comportement soit calculable par ordinateur. Nous exposons les étapes de la simulation physique et détaillons quelques méthodes d'intégration numérique permettant de résoudre les équations. Finalement, nous présentons un bref état de l'art sur les transformations topologiques des objets déformables en 2D et 3D.

2.2 Modèles de simulation physique

Dans cette section nous nous intéressons aux modèles de corps solides déformables basés sur la mécanique et proposés dans la littérature. L'utilisation de lois physiques est une hypothèse forte de nos travaux, nous n'évoquons pas ici les modèles de déformation purement géométriques (par exemple [Bar84, SP86]).

Les recherches sur l'animation de corps physiques ont démarré dans les années 1980 [TPBF87, Luc85, Mil88] afin d'améliorer le réalisme des animations créées par ordinateur. Ces travaux sont basés sur des méthodes issues de l'ingénierie mécanique qui ont été adaptées au contexte de l'informatique graphique. Ces travaux se sont poursuivis selon deux voies : d'une part, la complexification des modèles afin d'obtenir des animations de plus en plus réalistes et, d'autre part, la gestion de la lourdeur de calcul afin d'exploiter les simulations en temps réel ou en temps interactif. L'ensemble de ces modèles se divise en deux groupes, selon la mécanique, discrète ou continue, sur laquelle ils se basent. Notons que, dans les deux cas, une discrétisation de l'objet est nécessaire, afin de rendre fini le nombre de degrés de liberté de l'objet étudié. Ainsi, nous commençons cette section par l'introduction des concepts de base de la mécanique, discrète puis continue. Ces concepts sont nécessaires pour comprendre le reste de ce manuscrit. Ensuite, nous présentons une série de modèles, discrets ou continus, qui exploitent ces diverses lois physiques, en étudiant leurs avantages et inconvénients.

2.2.1 Notion de base de la mécanique

Pour étudier la déformation d'un corps, il existe dans la littérature deux approches différentes pour décrire son comportement : l'approche Lagrangienne et l'approche Eulérienne.

- **L'approche Lagrangienne** : l'objet est dissocié en un ensemble de particules mobiles qui sont caractérisées par une position de départ et dont la trajectoire est déterminée au cours de temps ;
- **L'approche Eulérienne** : les grandeurs physiques (scalaires, vecteurs, tenseurs) et leurs variations sont décrites en chaque point de l'espace. Pour cela, un certain nombre de points fixes sont choisis dans l'espace de l'objet et les propriétés mécaniques et leurs variations sont calculées uniquement en ces points.

D'une manière globale, mais ce n'est pas une règle, l'approche eulérienne est plutôt utilisée pour les fluides. Concernant les solides déformables, l'approche lagrangienne est largement préférée. Pour cela dans la suite de ce manuscrit, nous ne nous intéressons qu'à la représentation des objets avec l'approche lagrangienne.

2.2.1.1 Mécanique discrète

En mécanique discrète, un objet de masse m est discrétisé, au préalable, en un ensemble de points massiques que l'on nomme *particules*. Chaque particule possède une masse m_i , une position \mathbf{x}_i et une vitesse \mathbf{v}_i . Une particule interagit avec toutes ou un sous-ensemble des autres particules par l'intermédiaire de liaisons qui appliquent des forces d'attraction ou répulsion sur ces particules. Ces liaisons peuvent être des ressorts qui obéissent, par exemple, à la *loi de Hooke* pour reproduire des comportements élastiques. Dans ce cas, la force est proportionnelle à l'écart de distance entre les particules par rapport à la distance au repos :

$$f = k \times \Delta l \quad (2.1)$$

avec k la raideur du ressort et Δl la déformation.

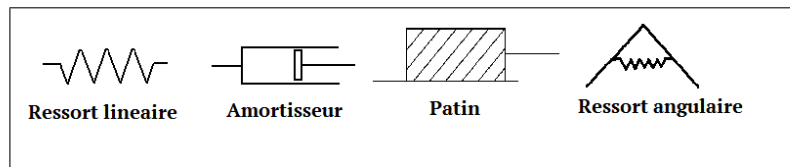


FIGURE 2.1 – Exemples d'interactions.

Ces interactions peuvent être de différents types (voir [LJF⁺91]). Quelques exemples sont présentés sur la figure 2.1 : ressorts linéaires ou non pour représenter un comportement élastique, amortisseurs pour la dissipation de l'énergie et pour permettre d'introduire un comportement visqueux dans l'objet, patins qui permettent d'introduire un comportement plastique et de la non-linéarité, etc. Certaines interactions peuvent comprendre plus que deux particules, par exemple, les ressorts angulaires permettant de s'opposer à la courbure/flexion. Il est possible de former des interactions plus complexes, construites par une combinaison en série ou en parallèle de différentes interactions élémentaires. Quelques exemples de ces combinaisons sont présentés sur la figure 2.2.

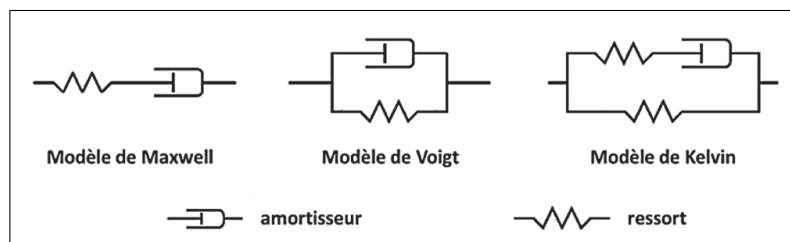


FIGURE 2.2 – Exemples de combinaison de liaisons.

Le modèle le plus utilisé pour simuler des objets déformables avec un comportement visco-élastique est celui de Voigt parce qu'il est le plus simple à modéliser et qu'il se calcule en tant que somme des forces engendrées par chaque interaction

élémentaire. Les autres assemblages fournissent des comportements plus complexes mais sont plus délicats à implanter.

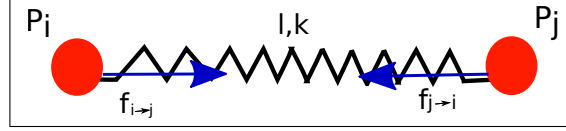


FIGURE 2.3 – Forces appliquées par un ressort.

Considérons par exemple que la liaison entre les deux particules est un ressort linéaire, comme montré sur la figure 2.3. La force appliquée par ce ressort linéaire sur l'une des particules se calcule à partir de la position des deux particules, comme décrit dans la formule suivante :

$$\mathbf{f}_{i \rightarrow j} = k_{ij} (\|\mathbf{x}_j - \mathbf{x}_i\| - l_{ij}) \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (2.2)$$

avec k_{ij} la raideur du ressort et l_{ij} sa longueur au repos. Par conséquent et en se basant sur la troisième loi de Newton, celle d'action/réaction, la force appliquée par la particule j sur i est déterminée par :

$$\mathbf{f}_{j \rightarrow i} = -\mathbf{f}_{i \rightarrow j} \quad (2.3)$$

Si le comportement désiré est un comportement visco-élastique, alors les vitesses des deux particules, c'est à dire les dérivées temporelles de leurs positions, notées $\dot{\mathbf{x}}_i$ et $\dot{\mathbf{x}}_j$, sont prises en compte pour calculer la force appliquée par la particule i sur j :

$$\mathbf{f}_{i \rightarrow j} = \left(k_{ij} (\|\mathbf{x}_j - \mathbf{x}_i\| - l_{ij}) - \lambda_{ij} (\dot{\mathbf{x}}_j - \dot{\mathbf{x}}_i) \cdot \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \right) \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (2.4)$$

où λ_{ij} est le coefficient d'amortissement. De même la force appliquée par la particule j sur i est l'opposée de $\mathbf{f}_{i \rightarrow j}$.

Pour animer l'objet, il faut déterminer la trajectoire de chaque particule au cours du temps, en appliquant le principe fondamental de la dynamique (PFD). Ce dernier indique que l'accélération d'une particule est proportionnelle à l'ensemble des forces appliquées sur elle comme montré par l'équation (2.5).

$$\mathbf{f}_i = m_i \ddot{\mathbf{x}}_i \quad (2.5)$$

avec m_i la masse de la particule i , $\ddot{\mathbf{x}}_i$ son accélération, autrement dit la dérivée seconde temporelle de sa position, et \mathbf{f}_i la somme des forces appliquées sur la particule à l'instant t . Cette somme comprend des forces intérieures \mathbf{f}_{int} issues de l'ensemble des interactions attachées à la particule (ces forces mesurent l'influence des autres particules sur la particule i), et les forces extérieures \mathbf{f}_{ext} (gravité, collision, force du vent, viscosité du milieu ambiant...).

$$\mathbf{f}_i = \mathbf{f}_{int} + \mathbf{f}_{ext} \quad (2.6)$$

Dans le cas particulier de forces extérieures constituées par la gravité \mathbf{g} et la viscosité γ_i d'un milieu ambiant, on obtient la formule suivante :

$$\mathbf{f}_i = \sum_j \mathbf{f}_{j \rightarrow i} + m_i \mathbf{g} - \gamma_i \dot{\mathbf{x}}_i \quad (2.7)$$

2.2.1.2 Mécanique des milieux continus

Nous introduisons dans cette partie les concepts de base de la mécanique des milieux continus. Au contraire de la mécanique discrète qui considère la matière comme un ensemble de points, la mécanique continue considère la matière à l'échelle macroscopique en tant que couverture continue de l'espace. Un objet est alors considéré comme un domaine borné par une surface qui délimite l'intérieur de l'extérieur de l'objet. Ce dernier se déforme et se déplace en respectant des contraintes externes comme par exemple la force de gravité. Le processus de la mécanique continue commence tout d'abord par déterminer le déplacement de chaque point de l'objet par rapport à sa position au repos. Ensuite, en utilisant ces déplacements, le tenseur de déformation peut être calculé, là aussi en chaque point de l'objet. Le tenseur de contrainte, correspondant aux forces dues à la déformation, est alors déterminé en appliquant une loi de constitution à partir du tenseur de déformation. Ensuite, en déterminant la densité d'énergie de déformation en chaque point, il est possible d'en déduire une densité de force. Nous détaillons, dans la suite, les diverses étapes de ce calcul.

Déplacement

L'application de forces à un objet non rigide provoque une déformation de ce dernier. Par conséquent, chaque point matériel constituant cet objet peut se déplacer depuis sa position au repos \mathbf{x}_0 vers une nouvelle position \mathbf{x} . Le déplacement $\mathbf{u}(\mathbf{x}_0)$ définit le vecteur déplacement entre la position au repos du point et sa position après la déformation comme montré dans la figure 2.4.

Tenseur de déformation

Le *tenseur de déformation* permet de décrire la déformation d'un objet suite à l'application d'une contrainte. Ce tenseur, noté $[\varepsilon]$, doit être invariant par translation ou rotation, car ces transformations ne représentent pas une déformation et sont, de ce fait, qualifiées de « rigides ». Le tenseur de déformation est représenté par une matrice 3×3 si l'objet est de dimension 3 :

$$[\varepsilon] = \begin{bmatrix} \varepsilon_x & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_y & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_z \end{bmatrix} \quad (2.8)$$

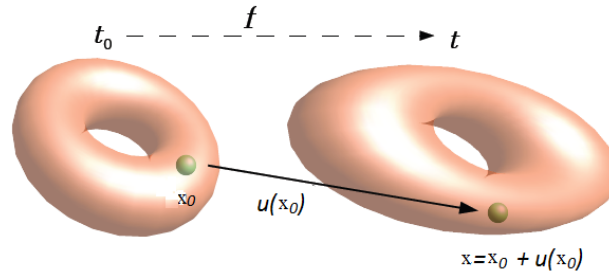


FIGURE 2.4 – Déplacement d'un point de sa position au repos à une position déformé (issue de [Nes08])

En inspectant la forme de ce tenseur, on peut remarquer qu'il caractérise deux types de déformation. La première, située sur la diagonale de la matrice, exprime l'élongation ou la compression le long des trois axes (\mathbf{x} , \mathbf{y} et \mathbf{z}). L'autre type de déformation, symbolisée par les trois éléments non diagonaux γ_{ij} (la matrice est symétrique), correspond au cisaillement. La figure 2.5 présente ces deux types de changement de formes à partir d'un objet au repos.

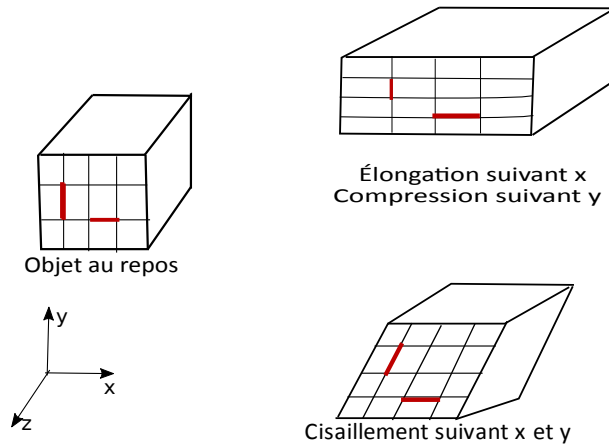


FIGURE 2.5 – Exemples de déformation d'un objet.

Il existe plusieurs solutions, plus ou moins complexes, pour calculer le tenseur de déformation. Parmi ces solutions, le tenseur de *Green Lagrange* permet de déterminer les valeurs de ce tenseur en fonction du gradient du champ de déplacements. Il s'écrit sous la forme suivante :

$$[\varepsilon] = \frac{1}{2} \left([\nabla \mathbf{u}] + [\nabla \mathbf{u}]^T + [\nabla \mathbf{u}]^T [\nabla \mathbf{u}] \right) \quad (2.9)$$

où ∇ est l'opérateur Nabla, $[\nabla \mathbf{u}]$ désigne la matrice représentant le gradient de \mathbf{u} et $[\nabla \mathbf{u}]^T$ sa transposée. Pour les petits déplacements, le tenseur de Green Lagrange peut être remplacé par sa forme linéaire qui est le tenseur de déformation de Cauchy. Il s'écrit sous la forme suivante :

$$[\varepsilon] = \frac{1}{2} \left([\nabla \mathbf{u}] + [\nabla \mathbf{u}]^T \right) \quad (2.10)$$

Le terme $\nabla \mathbf{u}^T \nabla \mathbf{u}$ est omis car non linéaire et négligeable (car quadratique) lorsque les déplacements sont petits. Le tenseur de Cauchy est généralement utilisé car il est plus rapide à évaluer et surtout, parce qu'étant linéaire, il peut être facilement interpolé. L'inconvénient de ce tenseur est qu'il n'est pas invariant en rotation et considère des déformations même pour de simples transformations rigides. Des forces sont alors artificiellement générées. Nous abordons plus précisément ce problème dans la section 2.2.2.3.

Tenseur des contraintes

Le *tenseur de contrainte* permet de décrire la répartition des forces internes en chaque point. En effet, une contrainte caractérise une force appliquée par unité de surface à un endroit donné de l'objet. Plus précisément, pour qu'une contrainte soit définie, il faut spécifier un élément de surface, et la contrainte correspond alors à la pression exercée sur cet élément. Elle est définie par un vecteur de dimension 3. Comme représenté sur la figure 2.6, les contraintes sont calculées pour trois éléments de surface qui sont orthogonaux respectivement aux axes \mathbf{x} , \mathbf{y} et \mathbf{z} . Chacune de ces contraintes se décompose en une composante perpendiculaire à la surface (contrainte normale, orientée selon l'un des axes) et deux autres composantes, tangentes à la surface, représentant des contraintes de cisaillement.

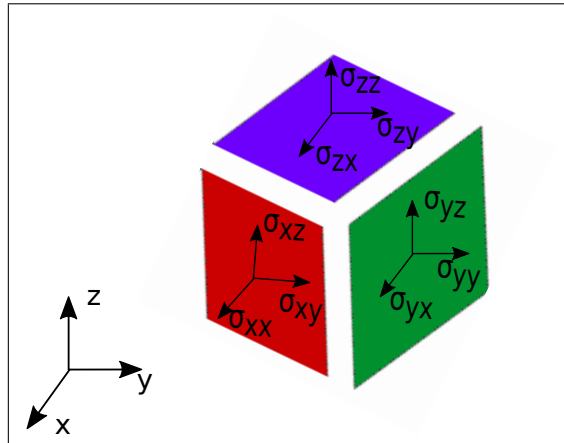


FIGURE 2.6 – Le tenseur des contraintes.

Ces trois contraintes sont regroupées dans une matrice 3×3 pour former le tenseur de contrainte. Ce dernier s'écrit ainsi, en chaque point du solide :

$$[\sigma] = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{bmatrix} \quad (2.11)$$

Loi de comportement

La loi de comportement définit la relation entre le tenseur de déformations et celui de contraintes :

$$[\sigma] = f([\varepsilon]) \quad (2.12)$$

Il existe dans la littérature différentes lois de comportement parce que, sous l'effet des forces extérieures exercées sur les matériaux, chaque type de matériaux se comporte d'une manière différente. Nous nous intéressons dans ce manuscrit aux lois de comportement élastique. Pour plus d'informations sur les autres comportements, le lecteur peut se référer, par exemple, à [YC93].

L'*élasticité* est un comportement qui correspond à plusieurs types de matériaux déformables. En physique, l'élasticité est la capacité d'un objet de se déformer suite à l'application d'une force extérieure, et à retrouver sa taille et sa forme d'origine lorsque cette force est supprimée.

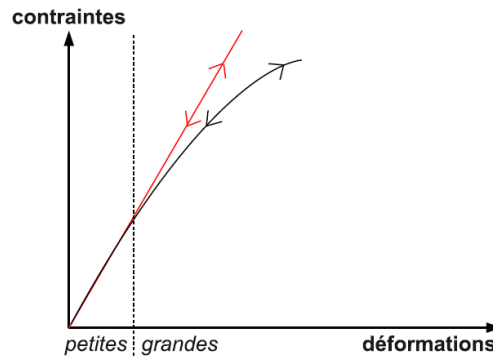


FIGURE 2.7 – Loi de comportement élastique (issue de [Der12]).

La partie en rouge de la figure 2.7 correspond à une *élasticité linéaire*, mais ne correspond qu'à de petits déplacements. En appliquant la loi de comportement élastique sur un matériau linéaire, la relation entre les déformations et les contraintes est linéaire. La relation entre tenseurs peut s'écrire de la façon suivante :

$$[\sigma] = [C] [\varepsilon] \quad (2.13)$$

Comme les tenseurs $[\varepsilon]$ et $[\sigma]$ sont d'ordre 2 (matrices 3×3), $[C]$ est un tenseur d'ordre 4, constitué de 81 coefficients.

Loi de Hooke

Dans le cas de l'élasticité linéaire, les matériaux peuvent être décrits par la loi de Hooke. Cette loi considère que le matériau utilisé est isotrope, c'est à dire qu'il possède les mêmes propriétés dans toutes les directions. Le tenseur de comportement C est calculé en se basant sur les paramètres rhéologiques¹ issus des mesures

1. La rhéologie est l'étude de la déformation de la matière sous l'effet d'une contrainte appliquée

effectuées sur des matériaux réels. En pratique, un matériau isotrope est désigné avec uniquement deux coefficients indépendants, λ et μ , appelées coefficients de Lamé. Avec ces coefficients, la loi de Hooke se définit ainsi :

$$[\sigma] = \lambda \text{tr}([\varepsilon]) [I] + 2\mu [\varepsilon] \quad (2.14)$$

où tr fournit la trace d'une matrice et $[I]$ est la matrice identité. Il existe une autre écriture de la loi de Hooke qui utilise le module de Young E et le coefficient de poisson ν pour définir les caractéristiques d'un matériau. Le module de Young E caractérise le degré d'élasticité d'un matériau et il est exprimé en Pascal. Le coefficient de poisson définit la compressibilité du matériau, s'exprime sans unité et est toujours compris entre 0 et 0.5. Ces paramètres sont très utilisés en mécanique parce qu'ils sont simples à interpréter et surtout directement mesurables. En utilisant ces deux coefficients, les coefficients de Lamé s'écrivent alors sous cette forme :

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)} \quad (2.15)$$

En conséquence, la loi de Hooke s'écrit :

$$[\sigma] = \frac{E}{1+\nu} \left(\frac{\nu}{1-2\nu} \text{tr}([\varepsilon]) [I] + [\varepsilon] \right) \quad (2.16)$$

Comme les tenseurs de déformations et de contraintes sont symétriques, il est possible de les représenter sous une forme vectorielle synthétique, énumérant leurs six valeurs significatives. On définit les vecteurs $\boldsymbol{\sigma}$ et $\boldsymbol{\varepsilon}$:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{xz} \\ \sigma_{yz} \end{bmatrix} \quad \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{xz} \\ \gamma_{yz} \end{bmatrix}$$

Le tenseur de comportement $[C]$ peut alors être défini sous la forme d'une matrice 6×6 :

$$[C] = \begin{bmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \quad (2.17)$$

Équation du mouvement

La loi de constitution ne permet pas, sous cette forme, d'exploiter la mécanique du point. Néanmoins, l'équation de **Navier** décrit le mouvement de chaque point de l'objet. Elle s'écrit sous la forme suivante :

$$\mathbf{f}_{ext} + \mathbf{f}_{int} = \rho \ddot{\mathbf{u}} \quad (2.18)$$

Ici, ρ est la masse volumique et $\ddot{\mathbf{u}}$ l'accélération du point d'application. \mathbf{f}_{ext} sont les forces extérieures appliquées à l'objet en ce point. Les forces internes en ce point sont calculées comme le vecteur divergence du tenseur de contraintes $\mathbf{f}_{int} = \nabla \cdot [\boldsymbol{\sigma}]$.

2.2.2 Méthodes de modélisation

Dans la section précédente, nous avons présenté les modèles mathématiques qui permettent d'établir une équation régissant le comportement d'un corps déformable. Or, avant de pouvoir exploiter puis résoudre la ou les équations mécaniques, une étape préalable de modélisation est nécessaire. En effet, le processus de modélisation n'est surtout pas à négliger, puisqu'il conditionne à la fois la faisabilité de la simulation et la conformité des résultats au phénomène observé dans la réalité. Souvent, il n'existe pas de solution analytique des équations. Aussi, dans cette section, nous présentons divers modèles, issus de stratégies de modélisation différentes en mécanique discrète ou continue.

Dans plusieurs de ces méthodes, la structure mécanique est issue d'un maillage géométrique. L'utilisation d'un maillage consiste à discrétiser le milieu spatial en un ensemble d'éléments géométriques et à s'appuyer sur cette structure pour exprimer les interactions entre éléments. Il existe cependant d'autres méthodes de résolution, dites sans maillage ou *meshless*. Comme leur nom l'indique, elles ne dépendent pas, pour la partie mécanique, d'une structure géométrique bien définie. Néanmoins, un maillage est généralement utilisé dans l'étape de visualisation.

Nous commençons cette partie par une discussion sur l'emploi des méthodes avec ou sans maillage. Nous nous intéressons ensuite aux modèles basés sur des maillages parce que notre approche est dédiée à ce type de modèles.

2.2.2.1 Discussion sur les maillages géométriques

Un corps déformable dont les équations sont régies par la mécanique des milieux continus possède potentiellement un nombre infini de degrés de liberté. Pour pouvoir calculer son comportement sur ordinateur, il faut nécessairement rendre fini ce nombre. Plusieurs stratégies ont été proposées.

Dans une première stratégie, que l'on peut appeler « modèles à jeu de déformations », les déformations autorisées sont définies *a priori*. La simulation consiste à déterminer les valeurs des paramètres de ces déformations au cours du temps. Par exemple, Pentland et Williams [PW89] proposent de calculer les modes de vibrations d'un corps déformable afin de contrôler ses transformations globales [Bar84] (courbure, torsion, etc.). Dans [CHP89], Chadwick *et al.* proposent d'utiliser une grille de type *Free Form Deformation* ou FFD [SP86] dont les nœuds obéissent aux équations physiques afin de modéliser la déformation du corps encapsulé dans la grille. Par ailleurs, les splines sont un outil très utile en modélisation pour représenter des

courbes, des surfaces ou des volumes, dont la forme est contrôlée par un nombre réduit de points, appelés *points de contrôle*. Il est possible de les utiliser en animation physique, par le biais de la mécanique Lagrangienne [RNG00, RNN00, TGAB08]. Toutes ces approches ont été proposées pour répondre au problème des simulation interactives ou temps-réel en réduisant de façon drastique le nombre de degrés de liberté du corps. Néanmoins, cette réduction peut globalement nuire au réalisme lorsqu'une déformation, non restituable par l'ensemble des degrés de liberté, est attendue par l'utilisateur. Par exemple, dans la simulation d'intestin grêle proposée dans [LFMC02, FLA⁺05], aucune déformation radiale n'est possible, toute la déformation se concentre sur la spline décrivant la forme longitudinale de l'intestin. Pour conclure sur cette stratégie, on peut dire qu'elle entretient un lien étroit avec la géométrie, puisqu'elle se base sur des transformations géométriques pour modéliser le comportement des corps.

Les autres stratégies considèrent une discrétisation de l'objet par des points, choisis *a priori*, qui pilotent les déformations de l'objet. Dans la seconde stratégie, celle des « modèles sans maillage », les points ne sont pas nécessairement choisis en lien avec la forme de l'objet, mais leur positionnement est plutôt motivé par le comportement et les déformations du corps. Il s'agit alors de définir les interactions entre les points, afin de définir les forces qui les animent. Dans un tel modèle, la structure de l'objet est purement mécanique. Certaines interactions peuvent se baser sur la géométrie, mais plutôt dans une optique de proximité : par exemple, une interaction est activée lorsque la distance entre les deux points qu'elle lie est plus faible qu'un seuil donné. Le problème de la visualisation se pose dans une étape ultérieure, la géométrie est donc globalement peu présente dans ce type de modèles.

La dernière catégorie comprend les « modèles maillés ». Elle est certainement la plus utilisée, en particulier en mécanique de l'ingénieur. Elle consiste à disposer les points afin de former un maillage géométrique de l'objet. Les points correspondent à des sommets ou à d'autres points caractéristiques (comme le milieu des arêtes) des éléments du maillage. Les interactions sont définies directement à l'aide du maillage, en exploitant les relations de voisinage topologique (interaction entre sommets liés par une arête, entre sommets d'un même élément, etc.).

La question de l'utilisation ou non d'un maillage est directement liée à l'importance du rôle que le modélisateur choisit d'attribuer à la géométrie. Car cette dernière n'est pas vraiment présente dans les équations mécaniques exposées en section précédente. Néanmoins, de nombreuses méthodes s'appuient sur la géométrie, et c'est à ces méthodes que s'adressent les contributions de notre travail.

Dans les sections suivantes, nous présentons brièvement les modèles sans maillage puis nous dressons un inventaire des modèles basés sur les maillages.

2.2.2.2 Modèles sans maillage ou *Meshless*

Les modèles sans maillage discrétisent l'objet en un ensemble de points obéissant à la mécanique du point. Ces particules sont dotées d'une masse et subissent des forces, extérieures ou intérieures. L'ensemble des influences entre les particules forme

un graphe d'interactions, qui structure l'objet. Aucun maillage n'est nécessaire, hormis peut-être pour la visualisation, mais dans ce cas, le maillage n'est pas utilisé pour définir le graphe d'interaction.

Le graphe d'interaction peut être complet, c'est-à-dire que chaque particule interagit avec toutes les autres. C'est le cas de la méthode *smoothed particle hydrodynamics* [Mon92]. Cette approche permet d'approximer les équations continues avec un ensemble de particules [MG04, FGBP11]. C'est également le cas des modèles inspirés de la dynamique globulaire [MP89] ainsi que les modèles basés sur les agglomérats [LJF⁺91, JSV97]. Si le graphe est complet, le nombre d'interactions est quadratique en nombre de particules. Pour réduire le temps de calcul, il est possible de délimiter l'action d'une particule uniquement à ces proches voisins. A chaque étape de la simulation, un sous-graphe représentant les couples de particules proches est extrait du graphe d'interaction. Ainsi, Desbrun *et al.* [DC95] ont proposé de limiter les distances d'interaction entre particules. De façon générale, le modèle SPH utilise également des rayons d'action bornés.

Cependant, le plus souvent, le graphe d'interaction est constitué, dès la modélisation, d'un sous-ensemble de l'ensemble des couples de particules. Le modèle le plus connu et le plus général est le modèle *masses/interactions* introduit dans les travaux de Luciani *et al.* [LJF⁺91] sur CORDIS ANIMA (voir figure 2.8(b)). Dans ce modèle, il est possible de définir l'ensemble des particules en interaction. Les interactions possibles sont multiples et définies de façon générique (la force d'une interaction est définie linéairement par morceaux en fonction de la distance entre les particules). Plus récemment, les approches périodynamiques [LBC⁺14] se basent également sur des graphes d'interactions.

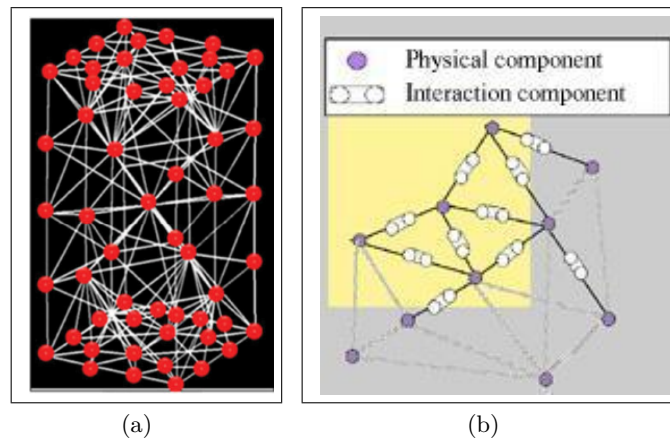
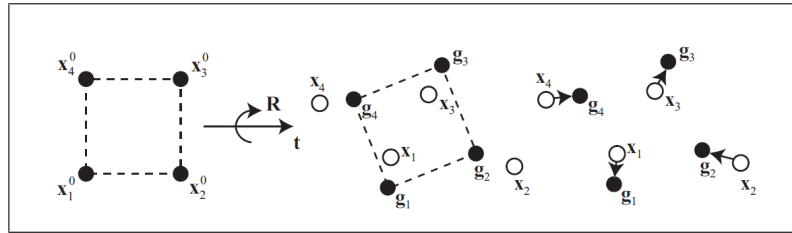


FIGURE 2.8 – Exemples de modélisation avec le modèle masse/interaction (issue de [CLF90]).

Il est possible de réduire considérablement le nombre d'interactions. Ainsi, en considérant une référence rigide de l'objet dont la position et l'orientation sont calculées par la méthode du *shape matching* [MHTG05], une particule n'est concernée que par une seule interaction, celle qui la lie à la référence rigide (voir figure 2.9).

FIGURE 2.9 – Le principe du *shape matching* (issue de [MHTG05])

Pour conclure, les méthodes sans maillage évitent d'introduire un maillage géométrique, dont les lois de la mécanique n'ont pas directement besoin, dans la modélisation mécanique du corps. Le principal problème de ces modèles est la visualisation, puisque le modèle mécanique n'est pas du tout construit sur des principes géométriques et surtout topologiques. Des principes d'habillage, par exemple par *mapping* [DKS⁺11] ou par surfaces implicites [DC95] ont été proposés. Pour plus d'informations concernant les méthodes sans maillage, le lecteur peut se référer à [FM03, GQ05].

2.2.2.3 Modèles maillés

L'utilisation d'un maillage permet de garantir un nombre fini de degrés de liberté. Le maillage décompose l'objet en un ensemble d'éléments (faces en 2D et volumes en 3D) juxtaposés et couvrant tout l'objet. L'objet se déforme alors en modifiant la position des sommets ou d'autres points caractéristiques des éléments (milieux des arêtes par exemple). Le maillage peut être bâti comme une grille alignée sur les axes, ce qui permet, au besoin, de définir, par différences finies, les opérateurs de dérivations par rapport aux coordonnées (x, y, z) que l'on trouve dans la mécanique des milieux continus. Néanmoins, dans la plus grande part des modèles, le maillage est quelconque et peut comprendre divers types d'éléments. Les plus classiques sont les triangles et quadrilatères en 2D, les tétraèdres et les hexaèdres en 3D. Une fois le maillage réalisé, il est possible de doter l'objet d'une mécanique discrète ou continue afin de simuler un comportement comme décrit sur la figure 2.10.

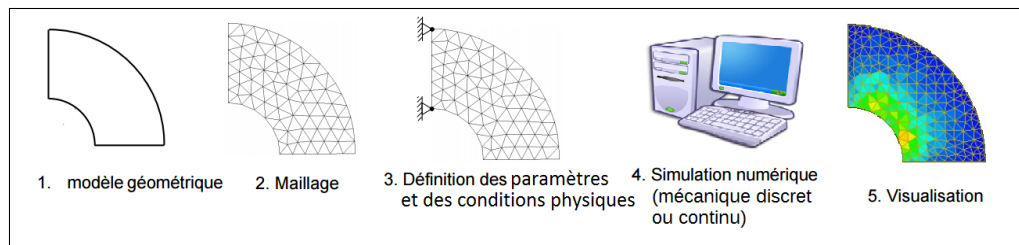


FIGURE 2.10 – Processus de la simulation numérique en utilisant une méthode avec maillage géométrique

Modèles discrets maillés

Dans les modèles maillés discrets, les sommets du maillage sont des particules dotées d'une masse. La masse d'une particule est définie généralement à partir des éléments qui l'entourent. Par exemple, sur la figure 2.11, un sommet (représenté en vert) est commun à quatre faces $F1$, $F2$, $F3$ et $F4$. Ces faces contribuent toutes les quatre à la masse de ce sommet.

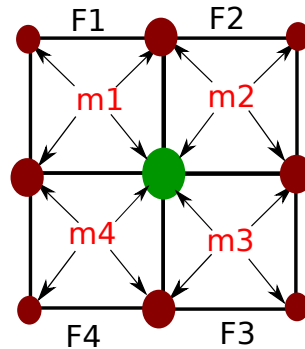


FIGURE 2.11 – Masse d'un sommet.

Les interactions se basent sur les relations topologiques entre sommets fournies par le maillage. Elles peuvent ainsi se positionner sur les arêtes ou dans les éléments. Elles peuvent être de n'importe quel type ou une combinaison de plusieurs interactions (voir section 2.2.1.1). Un exemple classique d'une telle construction est le modèle masse/ressort, souvent référencé sous l'abréviation MSS (*mass/spring system*), où les arêtes portent des liaisons constituées par des ressorts linéaires complétés, au besoin, par un amortisseur selon le modèle de Voigt (voir figure 2.12). Les liaisons ne se situent pas obligatoirement sur des arêtes. Par exemple, des ressorts de cisaillement peuvent être positionnés dans des faces ou des volumes.

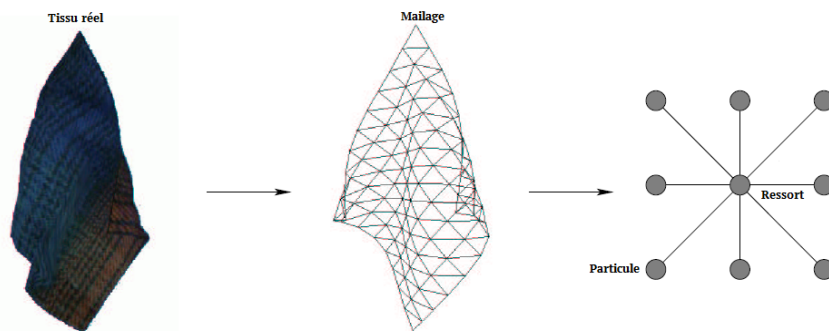


FIGURE 2.12 – Représentation d'un tissu par un maillage dont les sommets sont les particules et les arêtes sont les ressorts (issu de [Zar03]).

Dans la suite, nous montrons comment, dans la littérature, les modèles masses/ressorts ont été construits.

- **Modèle de dimension 2 :**

Un modèle masses/ressorts 2D a été proposé par Provot [Pro95] pour modéliser un tissu. C'est un modèle basé sur un maillage géométrique rectangulaire. Trois types de ressorts sont définis dans ce modèle, afin de modéliser différents comportements de tissu, comme présenté sur la figure 2.13. Le premier est constitué par les ressorts d'élongation qui lient chaque sommet à ses voisins auxquels il est relié par une arête (en noir). D'autres ressorts sont portés par les diagonales des faces pour maintenir leur forme et s'opposer au phénomène de cisaillement (en bleu). Pour gérer la courbure, Provot a introduit des ressorts linéaires qui connectent un sommet à son second voisin (en vert). La définition de chaque ressort entre les particules i et j est classique et se base sur une raideur k_{ij} et une longueur de repos l_{ij} . Les forces sont calculées par la formule (2.2) en page 10.

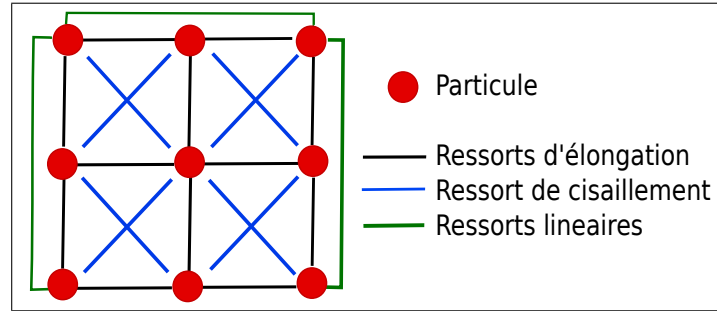


FIGURE 2.13 – Modèle de tissu de Provot [Pro95].

D'autres travaux se sont focalisés sur les maillages triangulaires [BMF03, GHDS03, BW98]. Tout d'abord, ces maillages ne nécessitent pas de ressorts de cisaillement. Pour la courbure, un ressort angulaire est utilisé pour contrôler l'angle entre deux faces triangulaires adjacentes. La force appliquée par ces ressorts est relative à l'angle entre les deux faces. Bridson *et al.* [BMF03] ont défini la formule suivante pour calculer les forces appliquées sur les sommets de deux triangles liés entre eux par un ressort angulaire positionné sur leur arête commune (entre les sommets 3 et 4, sur la figure 2.14) :

$$\mathbf{f}_i = k \frac{|\mathbf{e}^2|}{|\mathbf{n}_1| + |\mathbf{n}_2|} \sin \frac{\theta}{2} \mathbf{u}_i \quad (2.19)$$

avec i le sommet 1 ou 2, θ l'angle entre les normales des deux faces, k la raideur du ressort angulaire, $\mathbf{e} = \mathbf{x}_4 - \mathbf{x}_3$ l'arête commune. \mathbf{n}_i sont les normales aux faces (avec $\mathbf{n}_1 = (\mathbf{x}_1 - \mathbf{x}_3) \times (\mathbf{x}_1 - \mathbf{x}_4)$ et $\mathbf{n}_2 = (\mathbf{x}_2 - \mathbf{x}_4) \times (\mathbf{x}_2 - \mathbf{x}_3)$). \mathbf{u}_i désigne le déplacement de chaque sommet (suite à la variation d'angle). Il faut ensuite reporter l'opposée des forces subies par les sommets 1 et 2 sur les sommets 3 et 4.

- **Modèle de dimension 3 :**

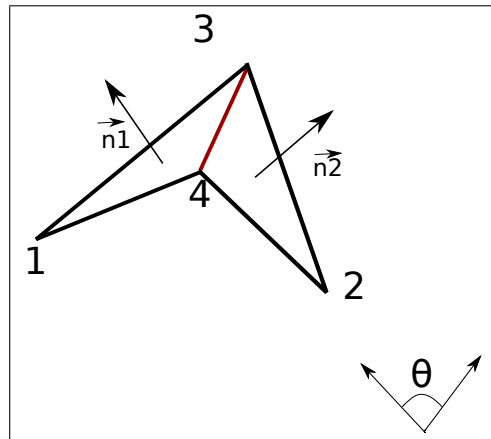


FIGURE 2.14 – Ressort angulaire.

En 3D, la démarche est similaire à la 2D. Des ressorts d'élongation peuvent être disposés sur les arêtes. Le cisaillement n'est géré que si les mailles sont hexaédriques, afin de maintenir leur forme. Les ressorts de cisaillement peuvent alors être placés dans le volume pour lier les sommets des diagonales de l'élément (en rouge dans la figure 2.15) ou (de façon non exclusive) le long des diagonales des faces, comme en 2D (en vert sur la figure 2.15). En 3D, les ressorts de courbure ne sont généralement pas utilisés, car les autres ressorts permettent de gérer ce phénomène.

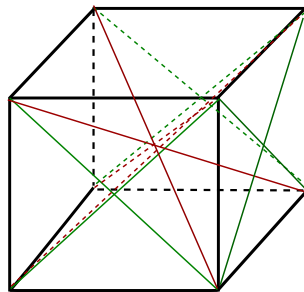


FIGURE 2.15 – Ressorts de cisaillement en 3D.

Les modèles masses/ressorts sont des modèles largement utilisés pour la simulation des objets déformables et plus précisément dans le domaine médical [TGG00], la simulation de tissu [BWK03, Pro95, HB99] et dans diverses autres approches. En effet, c'est un modèle simple, facile à implanter et qui permet d'obtenir des résultats de comportement visuellement satisfaisant, ce qui est suffisant pour des applications en informatique graphique. De plus, cette méthode est efficace en temps de calcul et elle permet de gérer proprement les modifications topologiques.

Bien qu'elle soit très utilisée, cette méthode n'est pas précise parce qu'il est très difficile de trouver les bons paramètres pour reproduire un comportement réel donné. Plusieurs travaux se sont focalisés sur la manière de trouver ces valeurs pour

les faire correspondre à un comportement observé ou désiré. Van Gelder [vG98] a défini une méthode pour trouver les raideurs des ressorts dans un maillage triangulaire en 2D et tétraédrique en 3D en se basant sur les paramètres rhéologiques du matériau qui constitue l'objet à simuler. Il se base sur les lois de la mécanique continue pour trouver les formules pour un matériau élastique isotrope. Pour cela, il calcule la raideur k_c d'un ressort avec une longueur au repos c et placé sur une arête commune à deux triangles $T1$ et $T2$ en utilisant la formule suivante :

$$k_c = \sum_{j=1}^n \left(\frac{E}{1+\nu} \right) \frac{\text{Aire}(T_i)}{|c^2|} + \left(\frac{E\nu}{1-\nu^2} \right) \frac{(a_i^2 + b_i^2 - c^2)}{8\text{Aire}(T_i)} \quad (2.20)$$

avec $i=1$ ou 2 pour désigner les triangles, E le module de Young, ν le coefficient de Poisson, c l'arête commune et a_i et b_i les autres arêtes du triangle i .

De la même manière, Fléchon *et al.* [FZDJ14] ont proposé des formules pour calculer les raideurs de ressorts pour des maillages rectangulaires en 2D et hexaédriques en 3D en utilisant les paramètres physiques des matériaux (le module de Young et le coefficient de Poisson). La formule proposée pour calculer les raideurs des ressorts d'élongation en 2D le long d'une arête est la suivante :

$$k_{elongation} = \frac{E(l_1^2(3\nu+2) - h^2)}{4l_1h(1+\nu)} + \frac{E(l_2^2(3\nu+2) - h^2)}{4l_2h(1+\nu)} \quad (2.21)$$

Comme on peut le remarquer dans cette formule, il existe deux termes correspondant à la contribution des deux faces adjacentes à l'arête. En pratique, chaque face apporte un ressort à l'arête (ressorts en rouge sur la figure 2.16), les deux ressorts se composent selon le schéma de Voigt, donc leurs raideurs s'additionnent pour former un ressort équivalent (ressort vert). Les raideurs se calculent à partir de la dimensions des faces, la longueur $h = h_1 = h_2$ de l'arête commune et l_1 et l_2 la longueur des autres arêtes.

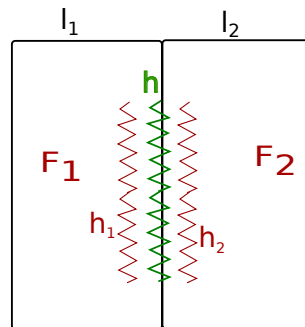


FIGURE 2.16 – Calcul de raideurs des ressorts pour des mailles rectangulaires [FZDJ14].

Fléchon *et al.* [FZDJ14] ont aussi proposé une formule qui permet de calculer la raideur des ressorts de cisaillement en 2D :

$$k_{\text{cisaillement}} = \frac{E(l^2 + h^2)}{4lh(1 + \nu)} \quad (2.22)$$

avec l et h respectivement la largeur et la longueur de la face.

De même en 3D, des formules pour les ressorts d'élongation et les ressorts de cisaillement à l'intérieur des volumes sont proposées.

Le comportement obtenu en utilisant le modèle masses/ressorts dépend du type de ressorts utilisé et du choix des valeurs de raideur et d'amortissement. Mais, ce modèle peut parfois exhiber de trop grandes déformations, qu'une augmentation des raideurs des ressorts peine à compenser. Pour cela, quelques travaux ont proposé d'ajouter des contraintes de déformation sur les ressorts pour avoir de meilleurs résultats. Par exemple Provot [Pro95] a défini des contraintes sur l'élongation des ressorts pour limiter l'élasticité du tissu lors de la simulation. Le fait d'utiliser des ressorts amène également des problèmes d'anisotropie : le ressort ne définit de force que dans une direction. Ainsi, l'objet peut plus ou moins bien réagir à des déformations, selon la direction dans laquelle s'exerce les contraintes. Certaines techniques visent à maîtriser les effets anisotropes [BC00].

Modèles continus

Les lois de la mécanique continue définissent la déformation et les contraintes de l'objet en tout point de l'objet à simuler. Ceci correspond à un nombre de degrés de liberté infini, qu'il convient de rendre fini pour permettre son calcul par ordinateur. Il existe dans la littérature deux grandes méthodes de discrétisation des équations de la mécanique de milieux continus. La méthode des différences finies et celle des éléments finis. Ces méthodes se basent toutes les deux sur une discrétisation spatiale, donc un maillage.

- **Méthode des différences finies**

La méthode des différences finies est une méthode de résolution numérique qui propose une écriture discrète des opérateurs différentiels. Le cas le plus simple consiste à approximer l'objet par une grille régulière. Pour toute quantité physique ξ , il est possible d'évaluer sa dérivée partielle grâce à :

$$\frac{\partial \xi}{\partial x_i}(\mathbf{x}) \approx \frac{\xi(\mathbf{x} + \delta_i \mathbf{x}_i) - \xi(\mathbf{x} - \delta_i \mathbf{x}_i)}{2\delta_i} \quad (2.23)$$

Où $\xi(\mathbf{x} + \delta_i \mathbf{x}_i)$ et $\xi(\mathbf{x} - \delta_i \mathbf{x}_i)$ représentent la quantité physique sur les nœuds voisins dans la grille, x_i la coordonnée choisie pour la dérivation partielle et \mathbf{x}_i son axe. Par exemple, ξ peut représenter un déplacement, ce qui permet de calculer en chaque nœud de la grille, le tenseur de déformation selon les formules (2.9), page 12, ou (2.10), page 13. À partir de la loi de constitution, le tenseur de contraintes puis les forces internes peuvent être évalués afin d'appliquer l'équation de Navier (2.18).

Cette méthode a été introduite pour la première fois en informatique graphique par Terzopoulos *et al.* [TPBF87] pour permettre de simuler des corps déformables. Son problème principal est l'utilisation d'une grille régulière, car les points de la grille sont disposés avec un écart de δ_i le long de chaque axe pour pouvoir utiliser directement la formule (2.23). Debunne *et al.* [DDBC99, DDCB00] ont proposé d'utiliser des opérateurs différentiels discrets plus souples, afin d'approximer, en particulier, l'opérateur Laplacien, utilisé dans l'équation de Navier. Cette approche permet d'utiliser tout type de maillage et non des grilles régulières.

Néanmoins, la méthode des différences finies n'est généralement pas la méthode préférée pour discrétiser les équations de la mécanique continue. En effet, elle est relativement instable car trop approximative. De plus des problèmes apparaissent aux nœuds frontières. En se basant sur l'équation (2.23), pour évaluer une dérivée en un sommet \mathbf{x} de la grille, on a besoin de la valeur de la quantité à dériver aux sommets $\mathbf{x} + \delta_i \mathbf{x}_i$ et $\mathbf{x} - \delta_i \mathbf{x}_i$. Si \mathbf{x} est un nœud frontière, l'une de ces deux positions désigne un point en dehors de la grille. La dérivée n'est donc pas évaluable par cette méthode et il faut utiliser d'autres formules, plus approximatives. Pour ces raisons, en simulation de corps déformables, la méthode par différences finies est souvent délaissée au profit de la méthode par éléments finis.

- **Méthode des éléments finis linéaires**

Dans la méthode des éléments finis (nom abrégé en FEM, pour *Finite-Element Method*), l'objet à simuler est divisé en un ensemble d'éléments (faces en 2D et volumes en 3D) : des triangles, des quadrangles, des tétraèdres, des hexaèdres... comme montré sur la figure 2.17.

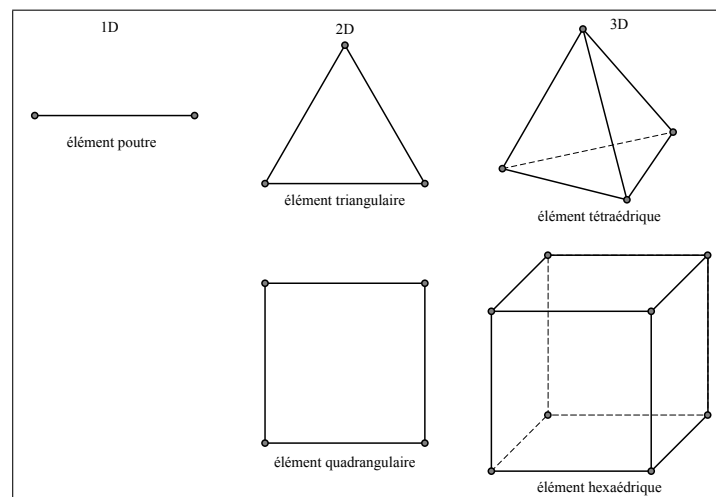


FIGURE 2.17 – Exemples d'éléments linéaires utilisés par la méthode des éléments finis.

Les propriétés physiques de tous les points d'un élément sont estimées par interpolation, en se basant sur les valeurs connues en certains points de l'élément, appelés

nœuds. Ces nœuds sont représentés en noir sur les figures 2.17 et 2.18. Dans le cas des éléments linéaires, ces nœuds sont les sommets des éléments (voir figure 2.17). Pour plus de précision, on peut augmenter le nombre de degrés de liberté et ajouter d'autres nœuds qui peuvent être par exemple, le centre de gravité, le milieu des arêtes, etc. Après la discrétisation de l'objet, la deuxième étape consiste à définir les fonctions d'interpolation, appelées également les fonctions de forme, pour chaque élément de maillage. Ces fonctions peuvent être linéaires ou quadratiques comme on peut le voir sur la figure 2.18.

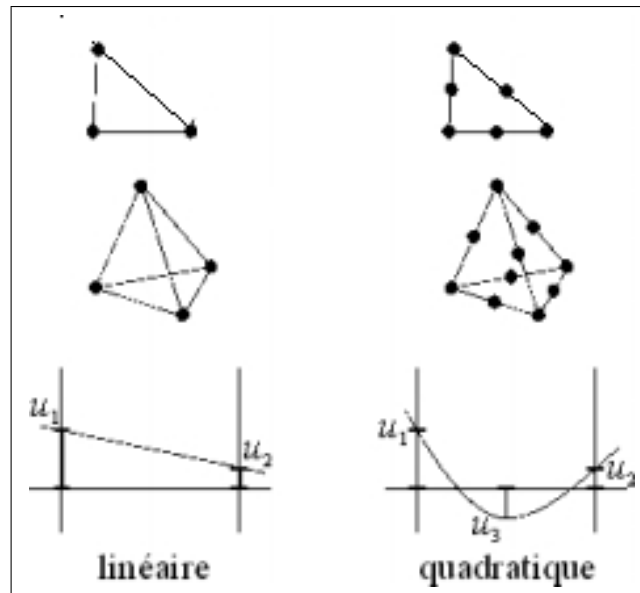


FIGURE 2.18 – Exemple d'éléments finis avec des fonctions d'interpolation linéaire et quadratique.

Une fois les fonctions d'interpolation choisies, on peut définir des équations de la mécanique continue pour chaque élément de l'objet. Le champ de tenseur de déformation sur un élément peut s'exprimer, grâce aux fonctions d'interpolation, sous la forme suivante :

$$\boldsymbol{\varepsilon}^{(e)}(x, y, z) = \left[B^{(e)}(x, y, z) \right] \mathbf{U}^{(e)} \quad (2.24)$$

Où $\left[B^{(e)}(x, y, z) \right]$ est la matrice de déformation de l'élément qui, en général, dépend de la position car elle se base sur les fonctions de forme de l'élément et plus précisément leurs dérivées par rapport aux coordonnées x , y et z . Néanmoins, si ces fonctions de forme sont linéaires, comme cela est le cas en 2D pour les triangles et en 3D pour les tétraèdres, les déformations se révèlent proportionnelles aux déplacements et la matrice $\left[B^{(e)} \right]$ est constante sur l'élément. En conséquence, compte tenu de la loi de Hooke, le tenseur de contrainte $\boldsymbol{\sigma}^{(e)}$ est constant sur l'élément et s'écrit sous la forme suivante :

$$\boldsymbol{\sigma}^{(e)} = [C] \boldsymbol{\varepsilon}^{(e)} = [C] [B^{(e)}] \mathbf{U}^{(e)} \quad (2.25)$$

Pour calculer les forces appliquées sur chaque nœud, il faut calculer tout d'abord l'énergie de déformation d'un élément, définie comme l'intégrale, sur le volume, des densités énergétiques de déformation $\boldsymbol{\varepsilon}^{(e)T} \boldsymbol{\sigma}^{(e)}$ calculées en chaque point de l'élément :

$$\xi^{(e)} = \int_{(e)} \boldsymbol{\varepsilon}^{(e)T} \boldsymbol{\sigma}^{(e)} dv \quad (2.26)$$

Puisque les tenseurs sont constants sur l'élément et les déplacements aux nœuds fixés, on obtient alors que :

$$\begin{aligned} \xi^{(e)} &= \int_{(e)} \mathbf{U}^{(e)T} [B^{(e)}]^T [C] [B^{(e)}] \mathbf{U}^{(e)} dv \\ &= \mathbf{U}^{(e)T} \int_{(e)} dv [B^{(e)}]^T [C] [B^{(e)}] \mathbf{U}^{(e)} \\ &= \mathbf{U}^{(e)T} v^{(e)} [B^{(e)}]^T [C] [B^{(e)}] \mathbf{U}^{(e)} \\ &= \mathbf{U}^{(e)T} [K^{(e)}] \mathbf{U}^{(e)} \end{aligned}$$

Avec $[K^{(e)}] = v^{(e)} [B^{(e)}]^T [C] [B^{(e)}]$ la matrice de rigidité de l'élément (e) , $v^{(e)}$ son volume (ou sa surface en 2D) et $\mathbf{U}^{(e)}$ l'ensemble des déplacements de ses sommets. Si on considère que des forces $\mathbf{F}^{(e)}$ sont appliquées uniquement aux nœuds de l'élément et doivent correspondre à la même énergie de déformation, on trouve que $\mathbf{U}^{(e)T} \mathbf{F}^{(e)} = -\mathbf{U}^{(e)T} [K^{(e)}] \mathbf{U}^{(e)}$. Cette égalité étant vérifiée pour toute valeur de déplacement, on obtient que les forces exercées sur les nœuds s'écrivent sous la forme suivante :

$$\mathbf{F}^{(e)} = - [K^{(e)}] \mathbf{U}^{(e)}. \quad (2.27)$$

Les matrices de rigidité de tous les éléments (e) peuvent être réunies dans une seule matrice globale, permettant de trouver la somme de toutes les forces \mathbf{F} subies par tous les nœuds du corps :

$$\mathbf{F} = -[K] \mathbf{U} \quad (2.28)$$

Où $[K]$ regroupe l'ensemble des $[K^{(e)}]$, et \mathbf{U} le déplacement de tous les nœuds. La matrice $[K]$ est symétrique et creuse parce que les nœuds de l'objet ne sont connectés qu'à certains autres et non à tous.

Dans l'approche statique, on cherche à déterminer les déformations, et donc les forces internes résultantes de l'application de forces extérieures \mathbf{R} . Le bilan des forces étant nul à l'équilibre, on résout l'équation :

$$\mathbf{R} - [K] \mathbf{U} = \mathbf{0} \text{ soit } [K] \mathbf{U} = \mathbf{R} \quad (2.29)$$

Ce qui revient à résoudre un système linéaire, lorsque la matrice $[K]$ est constante. On peut, en outre, imposer des contraintes de déplacement sur certains nœuds, les forces externes étant, dans ce cas, inconnues [GMTT89]. Néanmoins, la résolution de ce système est coûteuse en temps de calcul, aussi cette méthode n'est généralement pas utilisée directement en simulation temps réel. Diverses approches ont cherché à réduire sa complexité en considérant que seul le déplacement aux nœuds de surface ont un intérêt pour l'affichage [BNC96, JP99]. Il faut enfin noter qu'une animation basée sur cette méthode est qualifiée de pseudo-statique, puisqu'elle considère que la déformation est une suite d'états d'équilibre, ce qui est une hypothèse raisonnable seulement si les vitesses sont faibles.

Aussi dans le cas général, l'approche dynamique est préférable, et se base sur une formulation dite « explicite » des éléments finis. En effet, la formule décrite dans l'équation (2.27) ne vise qu'à calculer les forces exercées sur les nœuds d'un élément suite à la déformation de ce seul élément. Il est possible de calculer le cumul des forces issues des déformations de tous les éléments placés autour d'un nœud. On se base alors sur la deuxième loi de la dynamique exprimée en chaque nœud, ce qui s'écrit sous la forme suivante :

$$m_i \ddot{\mathbf{u}}_i + \lambda_i \dot{\mathbf{u}}_i - \mathbf{f}_i^{int} = \mathbf{f}_i^{ext} \quad (2.30)$$

Où m_i est la masse du nœud i , λ_i le coefficient de viscosité dans le milieu ambiant, \mathbf{f}_i^{int} est la somme des forces issues des déformations des éléments auxquels le nœud appartient, et \mathbf{f}_i^{ext} est la somme des forces extérieures. Cette équation suppose que la masse du corps, issue de chaque élément, se concentre sur les nœuds. Pour cela, Cotin *et al.* ont utilisé la technique dite du *mass-lumping*, qui revient à concentrer, sur la diagonale, l'ensemble des termes non diagonaux de la matrice de masse du corps. La matrice de masse est alors diagonale et on en déduit la masse de chaque nœud.

Nous avons présenté la méthode de base permettant de comprendre la méthode des éléments finis pour des éléments de base linéaires. Il est possible d'utiliser des éléments non simpliciaux, comme les quadrangles ou les hexaèdres. Dans ce cas, la stratégie de calcul reste la même, mais les fonctions de forme ne sont pas linéaires, et les tenseurs de déformation, et donc de contraintes, ne sont pas constants sur l'élément. Il est possible, dans ce cas, d'utiliser la technique des points de Gauss qui transforme l'intégrale de l'équation (2.26) en une somme finie de valeurs de la densité énergétique calculées en certains points bien définis, multipliée par la surface ou le volume de l'élément. Si la loi de Hooke et le tenseur de déformation linéarisé sont utilisés, la matrice de raideur reste constante par élément.

Pour conclure, nous constatons que la méthode des éléments finis est très répandue, en mécanique de l'ingénieur, mais également en informatique graphique, pour résoudre des équations basées sur les lois de la mécanique continue.

- **Masses/tenseurs**

Le modèle masses/tenseurs a été proposé par Cotin *et al.* [CDA00] pour réunir la

robustesse de la méthode des éléments finis et la simplicité des modèles masses/ressorts. Plus particulièrement, les auteurs ont défini cette méthode pour permettre de résoudre les problèmes de temps de calcul lors de la simulation de modifications topologiques. Cette méthode peut être considérée comme une autre façon d'exprimer les équations de la méthode des éléments finis linéaires explicites. Quelques résultats de l'approche de Cotin *et al.* sont présentés sur la figure 2.19 dans le cas de maillages tétraédriques.

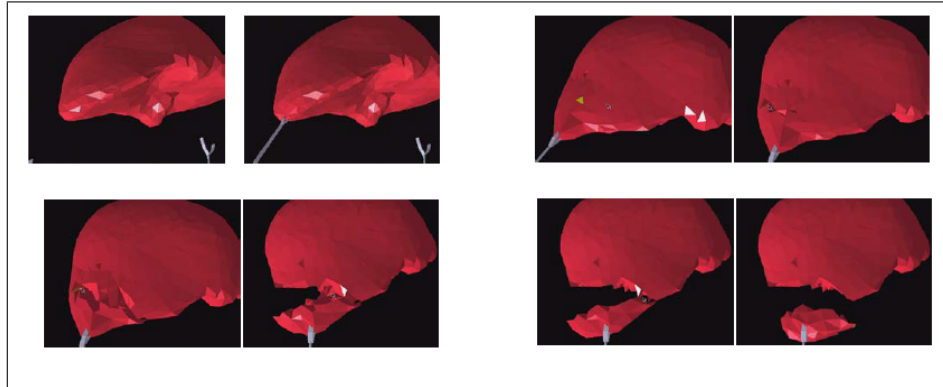


FIGURE 2.19 – Résultats de déformation et de découpe d'un foie avec la méthode masses/tenseurs (images issues de [CDA99]).

La méthode des masses/tenseurs revient à calculer, pour chaque élément, les effets de chaque sommet sur lui-même ainsi que son influence sur les autres sommets de l'élément. En réalité, ces influences entre déplacement des nœuds et les forces qu'ils subissent sont clairement explicitées dans la matrice de raideur $[K^{(e)}]$ de chaque élément. Si on considère, par exemple, des éléments tétraédriques, la matrice de rigidité est de taille 12×12 . Elle se décompose en 16 matrices de taille 3×3 , réparties entre les sommets et les arêtes du tétraèdre, comme montré par la figure 2.20.

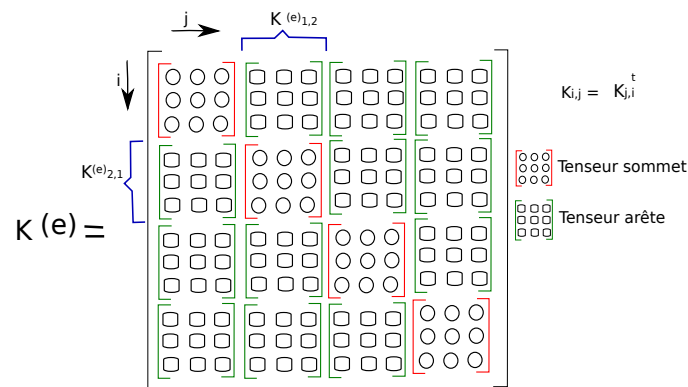


FIGURE 2.20 – Décomposition de matrice de raideurs.

Dans cette figure, les matrices (encadrées de rouge) $[K_{ii}^{(e)}]$ qui se trouvent sur la

diagonale montrent les forces subies par un sommet suite à son propre déplacement. Les autres matrices (encadrées de vert) $[K_{ij}^{(e)}]$ montrent comment le déplacement d'un sommet j entraîne l'application d'une force sur le sommet i . Comme la matrice de rigidité globale 12×12 est symétrique, l'influence du sommet i sur le sommet j est obtenue par $[K_{ij}^{(e)}] = [K_{ji}^{(e)}]^T$. Ces deux tenseurs sont associés à la même arête, joignant i et j . Aussi, au lieu de définir 16 tenseurs pour chaque tétraèdre, il suffit d'en définir seulement 10, comme montré sur la figure 2.21.

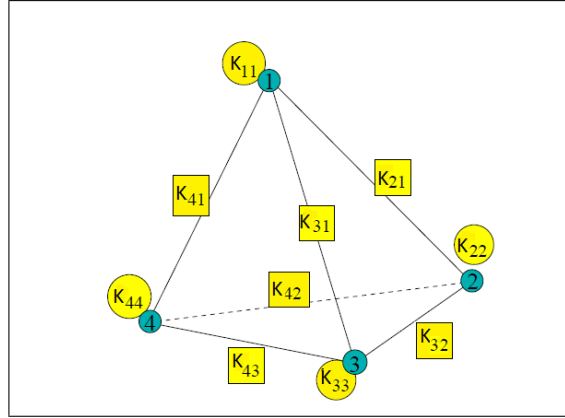


FIGURE 2.21 – Les tenseurs de raideur qui permettent de calculer les forces appliquées sur chaque nœud.

Cette approche peut se généraliser pour tout autre type d'élément linéaire (matrice de rigidité constante). On trouve alors des tenseurs liés aux sommets, aux arêtes, mais aussi aux diagonales de faces et de volumes, le cas échéant (par exemple dans les hexaèdres).

Dans un maillage, un sommet ou une arête appartient généralement à plus d'un élément. Pour chaque couple de nœuds (n_1, n_2) avec n_1 non nécessairement différent de n_2 , il faut prendre en compte les influences issues des déformations de tous les éléments possédant ces nœuds. Cette propriété est également vérifiée par certaines diagonales (par exemple les diagonales d'une face quadrangulaire partagée entre deux hexaèdres). Pour se rapprocher d'une simulation dont la complexité est analogue à celle d'un modèle masses/ressorts (où les interactions sont traitées par couple de particules), Il faut accumuler ces différentes influences. Par exemple, il est recommandé de cumuler les tenseurs des éléments incidents à un même sommet (*i.e.* : $[K_{ii}] = \sum_{(e)} [K_{ii}^{(e)}]$), ou une même arête ou diagonale (*i.e.* : $[K_{ij}] = \sum_{(e)} [K_{ij}^{(e)}]$). Ainsi le bilan des forces appliquées sur chaque sommet est calculé à partir de toutes les contributions :

$$\mathbf{f}_i = [K_{ii}] \mathbf{u}_i + \sum_{j \neq i} [K_{ij}] \mathbf{u}_j \quad (2.31)$$

Où $[K_{ii}]$ est la somme des contributions du déplacement du nœud i dans tous les éléments auxquels il appartient. $[K_{ij}]$ correspond à la somme de contributions

des éléments auxquels l'arête ou la diagonale ij appartient.

Cotin *et al.* [CDA99] ont défini une formule qui permet de calculer chaque tenseur $[K_{ij}^{(e)}]$ d'un tétraèdre (e) à part, sans utilisation de la matrice $[K^{(e)}]$ des éléments.

$$[K_{ij}^{(e)}] = \frac{1}{36v_{(e)}} (\lambda (\mathbf{n}_j \times \mathbf{n}_i) + \mu (\mathbf{n}_i \cdot \mathbf{n}_j) I_3 + \mu (\mathbf{n}_i \times \mathbf{n}_j)) \quad (2.32)$$

Où $v(e)$ est le volume, λ et μ sont les coefficients de Lamé, \mathbf{n}_i est le vecteur normal à la face opposée au sommet i du tétraèdre (e), de même pour \mathbf{n}_j . Tous ces tenseurs peuvent être calculés dans une phase préliminaire parce que les données de la formule (2.32) ne dépendent que de la géométrie du maillage au repos et des coefficients de Lamé.

Pour conclure, lors d'un changement topologique, il faut uniquement mettre à jour les tenseurs sur lesquels le ou les éléments modifiés ont une influence. Le problème de cette méthode est qu'elle nécessite que les matrices de rigidité soient constantes en chaque élément. Ceci n'est vérifié que dans le cas de l'élasticité linéaire (loi de Hooke) en petits déplacements (tenseur de déformation linéarisé). Picinbono *et al.* [PDA00,PDA01] ont étendu les travaux de Cotin pour pouvoir simuler avec la méthode masses/tenseurs des matériaux anisotropes avec une élasticité non linéaire en utilisant le tenseur de Green Lagrange.

- **Méthode du Co-rotationnel**

La méthode des éléments finis explicite et l'approche masses/tenseurs proposée par Cotin se basent sur le tenseur de déformation linéarisé, et, de ce fait, font l'hypothèse de petits déplacements. En particulier, le tenseur de déformation n'est pas invariant par rotation. Toute rotation subie par un élément est interprétée comme une déformation, qui génère des forces dans des directions ne permettant pas de réduire cette « déformation ». Ce phénomène aboutit à un gonflement de l'objet comme montré sur la figure 2.22, où la barre en rouge est simulée avec un tenseur de comportement non linéaire et celle en vert avec un tenseur de comportement linéaire.

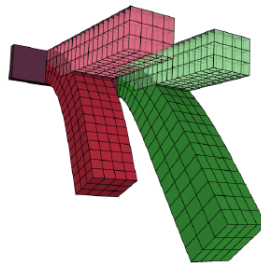


FIGURE 2.22 – Simulation de comportement de deux poutres (image issue de [NPF05]).

Pour résoudre ce problème, quelques approches proposent d'utiliser des tenseurs de comportement non-linéaires pour mieux calculer les déformations, par exemple, le tenseur de Green Lagrange, car il est invariant en rotation. Müller *et al.* [MG04] ont proposé une autre approche, dite du « co-rotationnel », qui peut être considérée comme une extension de la méthode des éléments finis linéaires. Le principe de cette méthode consiste à extraire un mouvement rigide (translation et rotation) de la transformation entre la position courante des nœuds et leur position de repos, afin de pouvoir définir la déformation de chaque élément proprement dite. Le calcul de la force nette appliquée à un élément par la formule (2.27), page 27, devient alors :

$$\mathbf{F}^{(e)} = [R^{(e)}] [K^{(e)}] \left([R^{(e)}]^T \mathbf{X} - \mathbf{X}_0 \right) \quad (2.33)$$

Où $[R^{(e)}]$ est une matrice 12×12 formée par la duplication, le long de la diagonale, de la matrice de rotation 3×3 calculée pour chaque élément, \mathbf{X} est un vecteur regroupant la position des sommets et \mathbf{X}_0 leur position au repos. Cette formule applique la rotation inverse sur la position des sommets dans la configuration déformée, afin d'exprimer la déformation dans le repère de l'élément au repos. La force correspondant à cette déformation est calculée en utilisant la matrice de rigidité. Il ne faut pas oublier de ré-exprimer les forces dans le repère global en appliquant la rotation de l'élément.

Il existe des méthodes assez peu coûteuses pour déterminer le mouvement rigide dans le cas d'éléments de type triangulaire ou tétraédrique. Aussi, cette méthode se restreint, la plupart du temps, à des maillages de simplexes. Pour extraire la rotation et la translation de la transformation subie par un tétraèdre, Nesme *et al.* [NPF05] proposent d'utiliser une décomposition QR. L'idée de cette méthode est de calculer comment un repère local d'un élément se transforme après déformation, en se basant sur la construction de Gram-Schmidt. Pour cela il faut choisir un sommet de l'élément comme origine et ordonner les arêtes (en respectant une orientation positive) issues de ce sommet. La première arête fournit le premier axe. Le deuxième axe doit appartenir au plan des deux premières arêtes, mais ne peut être construit immédiatement. Le troisième axe doit être orthogonal au plan contenant les deux premiers axes et, par conséquent, les deux premières arêtes. Aussi, on le construit à partir du produit vectoriel du premier axe et de la seconde arête. Une fois le troisième et le premier axes construits, il suffit de calculer leur produit vectoriel pour obtenir le second axe. La figure 2.23 montre un exemple dans lequel l'origine est le sommet B , la première arête choisie comme premier axe est BC , le deuxième axe lui est perpendiculaire et appartient au plan (BC, BA) et le troisième axe est orthogonal aux deux autres (il correspond simplement au produit vectoriel entre \mathbf{BC} et \mathbf{BA}).

Cette construction de Gram-Schmidt est appliquée aux trois vecteurs colonnes extraits de la matrice de transformation de l'élément. Elle indique alors comment le repère $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ est transformé en un autre repère orthonormé. Il s'agit bien d'une rotation, approximant la transformation de l'élément, en ignorant la partie défor-

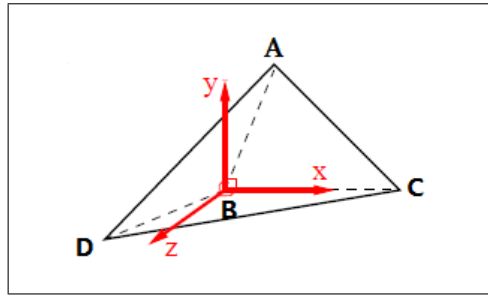


FIGURE 2.23 – Principe de la décomposition QR.

mation. Il faut noter que la rotation définie par cette méthode dépend du sommet d'origine et de l'ordre des arêtes choisi.

Tous les modèles présentés (masses/ressorts, méthode des différences finies, la méthode des éléments finis linéaires, masses/tenseurs et co-rotationnel) se différencient par leur degré de précision et réalisme, le temps de simulation et leur capacité à gérer des modifications topologiques. Mais il semble intéressant de regrouper plusieurs modèles physiques dans un même outil pour que l'utilisateur puisse comparer la déformation d'un même objet avec les mêmes propriétés physiques selon les approches choisies. De plus, il est aussi intéressant de simuler un même objet avec deux modèles physiques en même temps. Par exemple la simulation de l'intérieur d'un corps peut se faire avec une méthode des éléments finis explicite alors que sa surface est modélisée avec le modèle masses/ressorts afin de limiter les variations de volume global de l'objet. Ce type de simulation avec un modèle hybride est prévu, par exemple, dans le *framework* SOFA [Sof].

2.3 Animation Physique

Maintenant que les équations nécessaires ont été introduites dans la section précédente, il faut ici entreprendre leur résolution. Pour pouvoir simuler des corps déformables, il est nécessaire de connaître la position de chaque particule au cours du temps. Une équation de mouvement doit être écrite pour chaque particule. Pour tous les modèles mécaniques, il est possible de se ramener à l'équation (2.34). Cette équation réunit toutes les équations de toutes les particules dans un système à n équations :

$$[M]\ddot{\mathbf{x}}(t) + [D]\dot{\mathbf{x}}(t) + [K]\mathbf{x}(t) = \mathbf{f}_{ext}(t) \quad (2.34)$$

avec $[M]$, $[D]$ et $[K]$ les matrices respectivement de masse, d'amortissement et de raideur, \mathbf{x} la position ou le déplacement du point. C'est une équation du second ordre qui nécessite une double intégration pour trouver les positions des particules. On parle d'approche *dynamique*. Comme vu dans l'état de l'art des méthodes à

éléments finis, il est possible de se focaliser uniquement sur l'équation à l'équilibre. Il s'agit du cas *statique*, dont l'équation s'écrit ainsi :

$$[K] \mathbf{x}(t) = \mathbf{f}(t) \quad (2.35)$$

Ici, on cherche l'état d'équilibre du système en ignorant l'évolution au cours du temps, c'est à dire les états transitoires. Néanmoins, cette approche n'est adaptée qu'à des situations pseudo-statiques, où chaque étape du mouvement peut être considérée comme un équilibre.

Concernant l'approche dynamiques, l'évolution au cours de temps des degrés de liberté de l'objet est recherchée. Pour simplifier la résolution de l'équation (2.34), qui est une equation du second ordre, elle peut être réécrite sous la forme d'un système d'équations du premier ordre :

$$\begin{cases} \dot{\mathbf{v}} = [M]^{-1} (\mathbf{f}_{ext} - [D] \mathbf{v}(t) - [K] \mathbf{x}(t)) \\ \dot{\mathbf{x}} = \mathbf{v}(t) \end{cases} \quad (2.36)$$

Pour résoudre numériquement ce système, les algorithmes utilisés se basent sur une discrétisation temporelle, en d'autres termes, la solution n'est calculée qu'en certains instants. Le temps entre deux états calculés est appelé *pas de temps*, on le note Δt . Ce mode opératoire n'est pas un handicap car le système n'a généralement besoin d'être connu qu'à certains instants, correspondant souvent à une modalité d'interaction donnée (fréquence d'affichage ou de mise à jour d'un retour haptique, par exemple). Selon le théorème de Shannon, le pas de temps ne doit pas dépasser une certaine valeur, fonction de la fréquence naturelle d'oscillation du système :

$$\Delta t < \frac{1}{2f} \quad (2.37)$$

où f est la fréquence naturelle d'oscillation du système.

Si une trop grande valeur est choisie pour le pas de temps, toute résolution de l'équation du mouvement est incorrecte. Dans certains cas, cette non-conformité se traduit par une divergence de la résolution. Néanmoins, si le pas de temps choisi est trop petit, le temps de calcul de la résolution devient très important, et des états du système non réellement utiles sont calculés (typiquement, pour visualiser une animation sans saccades, quelques dizaines de millisecondes peuvent séparer deux états).

2.3.1 Méthodes d'intégration

Les méthodes de résolution par discrétisation du temps consistent, à partir d'un état connu du système à un instant t , à calculer son nouvel état à l'instant $t + \Delta t$, en s'appuyant sur les équations (2.36) page 34. Il existe dans la littérature différentes méthodes d'intégration numérique. On peut les distinguer en deux grands types : les méthodes explicites et les méthodes implicites.

2.3.1.1 Méthodes explicites

Les méthodes explicites sont les méthodes les plus simples à mettre en œuvre pour calculer l'évolution au cours du temps d'un système. Une méthode explicite des plus élémentaires est la méthode d'Euler explicite. Elle est basée uniquement sur le développement de Taylor d'ordre 1 qui s'écrit sur la forme suivante :

$$\mathbf{h}(t + \Delta t) \approx \mathbf{h}(t) + \dot{\mathbf{h}}(t)\Delta t \quad (2.38)$$

La méthode d'Euler explicite applique ce développement à la vitesse et à la position des degrés de liberté :

$$\begin{cases} \mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{v}(t)\Delta t \\ \mathbf{v}(t + \Delta t) \approx \mathbf{v}(t) + \dot{\mathbf{v}}(t)\Delta t \end{cases} \quad (2.39)$$

Compte tenu de sa simplicité, cette méthode est rapide en temps de calcul. Elle se comporte bien si le pas de temps choisi est adapté.

En se basant également sur le développement de Taylor d'ordre 1 et l'équation (2.39), les nouvelles positions peuvent dépendre des nouvelles vitesses, que l'on calcule juste avant de traiter les positions. Cette méthode s'appelle méthode d'Euler semi-implicite ou symplectique, et elle est plus stable.

$$\begin{cases} \mathbf{v}(t + \Delta t) \approx \mathbf{v}(t) + \dot{\mathbf{v}}(t)\Delta t \\ \mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{v}(t + \Delta t)\Delta t \end{cases} \quad (2.40)$$

Il existe diverses méthodes d'intégration explicite, comme par exemple la méthode de Störmer-Verlet (ordre 2) et Runge Kutta (en particulier ordre 2 et 4), qui peuvent être un peu plus compliquées à mettre en œuvre. Néanmoins, il a été montré, dans la thèse de Provot [Pro97] par exemple, que l'intégration d'Euler symplectique est plus stable que celle de Runge Kutta d'ordre 2. Ces diverses méthodes permettent d'augmenter la précision, mais elles sont plus coûteuses en temps de calcul et, souvent, également en mémoire.

2.3.1.2 Méthodes implicites

Les méthodes implicites permettent de résoudre les problèmes d'instabilité (convergence de la résolution) des approches explicites. Elles sont inconditionnellement stables (si le pas de temps est trop élevé, le comportement n'est pas correctement calculé, mais le système ne diverge pas). L'idée de ces méthodes est de calculer les positions et les vitesses à l'instant $t + \Delta t$, pas à l'aide des forces du temps précédent, comme le font les méthodes explicites, mais à l'aide des forces de l'instant considéré. Ceci permet de garantir que l'état en $t + \Delta t$ est compatible avec l'état en t . Parmi les méthodes implicites, la plus connue est celle d'Euler implicite qui s'écrit sous la forme :

$$\begin{cases} \mathbf{v}(t + \Delta t) \approx \mathbf{v}(t) + \dot{\mathbf{v}}(t + \Delta t)\Delta t \\ \mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{v}(t + \Delta t)\Delta t \end{cases} \quad (2.41)$$

La difficulté, ici, est que pour calculer $\mathbf{v}(t + \Delta t)$, il faut connaître $\dot{\mathbf{v}}(t + \Delta t)$ qui dépend des forces exercées sur le système et donc de $\mathbf{x}(t + \Delta t)$ et $\mathbf{v}(t + \Delta t)$. Or, la deuxième équation de ce système indique que $\mathbf{x}(t + \Delta t)$ n'est calculable que lorsque $\mathbf{v}(t + \Delta t)$ est connue. Pour résoudre cette équation, un système non linéaire doit nécessairement être résolu. Bien que ces approches soient plus difficiles à implanter que les méthodes explicites, elles sont utilisées dans plusieurs travaux [BW98, DSB99] en raison de leur stabilité.

Le choix de la méthode d'intégration a une grande importance dans la précision et la stabilité des simulations. Chacune de ces méthodes a des avantages et inconvénients, pour cela le choix dépend de l'application visée. Dans notre cas, un modèle simple est suffisant : nous nous intéressons aux principes des modèles et à la mise en œuvre algorithmique. Nous pouvons adapter les paramètres physiques d'un corps à simuler pour garantir une résolution correcte des équations, pour un pas de temps donné ou, au besoin, adapter le pas de temps au système étudié. Aussi, une intégration symplectique d'Euler est parfaitement adéquate.

2.3.2 Boucle de simulation

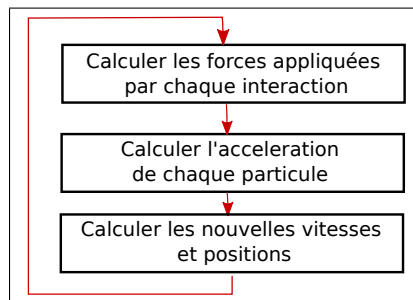


FIGURE 2.24 – Les étapes de la boucle de simulation.

Une fois que toutes les forces sont calculées et que la méthode d'intégration qui permet de calculer les nouvelles vitesses et positions est choisie, il faut appliquer la boucle de simulation décrite dans la figure 2.24 pour chaque pas de temps.

Comme décrit dans cette figure, la première étape consiste à calculer les forces appliquées par chaque interaction sur chaque particule. L'étape suivante consiste alors à calculer l'accélération de chaque particule, qui revient en fait à calculer (1) le bilan des forces appliquées sur chaque particule et (2) son accélération, en se basant sur la deuxième loi de Newton définie dans l'équation (2.5) en page 10. Ensuite, il faut calculer les nouvelles vitesses et positions de chaque particule en utilisant un schéma d'intégration bien défini. Ces mêmes étapes sont à suivre à chaque pas de temps de la simulation.

Cette boucle peut être utilisée pour simuler les déformations de n'importe quel objet avec un modèle discret ou continu. La seule différence réside dans la manière de calculer des forces, les autres étapes sont les mêmes pour tous les modèles physiques

dynamiques.

2.4 Modifications Topologiques

Les transformations topologiques sont très exploitées pour simuler les explosions ou les fractures d'un objet, par exemple dans les jeux vidéo ou en réalité virtuelle. On peut également citer les découpes et déchirures de tissus dans les simulateurs chirurgicaux. Nous nous focalisons dans cette section sur les transformations topologiques des modèles physiques, sujet largement étudié dans la littérature.

Tout d'abord, il nous faut différencier la découpe d'une part et la déchirure et la fracture d'autre part. La simulation d'une découpe en chirurgie virtuelle est généralement contrôlée par l'utilisateur, au moyen d'un outil de coupe (par exemple une lame à bistouri dans le domaine médical). En pratique, un simple contact entre le corps à inciser et l'outil est suffisant pour déclencher la découpe. Au contraire, la déchirure ou la fracture est une opération qui apparaît dans le corps simulé comme conséquence des contraintes qu'il subit. Dans nos travaux, nous considérons que les déchirures ou fractures sont les traductions d'un même phénomène, la différence s'expliquant essentiellement par les lois de comportement du corps simulé (corps rigide, corps déformable, corps ductile²...).

Nous ne considérons dans cette partie que les objets basés sur des maillages géométriques. Les découpes, déchirures ou fractures apparaissent comme des modifications de ce maillage, comme, par exemple, une séparation de faces ou volumes adjacents, la suppression d'une face ou d'un volume, etc. Par ailleurs, pour créer des simulations plus réalistes et plus précises, plusieurs approches ont proposé de raffiner la résolution d'un maillage avant de lui appliquer des modifications topologiques. Au contraire, des zones non manipulées voire invisibles peuvent bénéficier d'un passage à une résolution plus grossière du maillage, pour améliorer le temps de calcul. Ces changements de résolution sont également des modifications topologiques. Après des modifications du maillage d'un corps, y compris les changements de sa résolution, il est, la plupart du temps, nécessaire de mettre à jour son modèle mécanique.

Pour prendre en compte les modifications topologiques, la boucle de simulation présentée dans la section précédente doit être adaptée comme le montre la figure 2.25. Après le calcul des nouvelles positions, il faut évaluer un critère qui déclenche la modification. Ce critère peut être, par exemple, un contact entre l'objet et l'outil de coupe dans le cas des découpes, ou une intensité trop élevée des forces appliquées en un point du corps, dans le cas des déchirures ou fractures. Si le critère est vérifié, alors il faut appliquer la modification topologique correspondante et, ensuite, mettre à jour les propriétés mécaniques de l'objet après la modification du maillage. Nous exposons, dans cette section, un bref état de l'art de ces modifications topologiques, que nous regroupons par type.

2. Qui est capable de subir une déformation plastique (sans retour à sa forme d'origine) avant de se rompre dans des cas extrêmes.

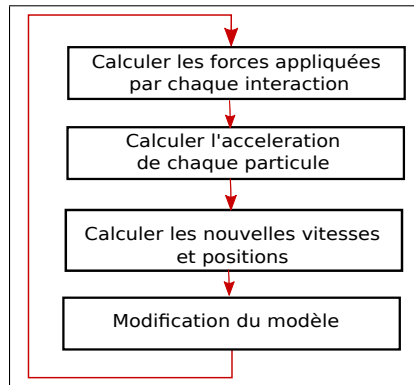


FIGURE 2.25 – Les étapes de la boucle de simulation avec transformations topologiques.

2.4.1 Découpe

Plusieurs méthodes pour la simulation des découpes ont été proposées dans la littérature. Nous avons décidé de classer ces méthodes de découpe suivant la modification topologique choisie. Ces modifications peuvent être regroupées en quatre catégories qui sont la suppression, la séparation, la division et la duplication d'éléments.

2.4.1.1 Suppression d'éléments

La méthode la plus simple pour simuler une découpe d'un objet consiste à supprimer un ou plusieurs éléments du maillage lorsque l'outil de coupe entre en collision avec le corps simulé comme montré sur la figure 2.26. L'élément ou les éléments à supprimer sont ceux qui se trouvent le long du chemin de coupe.

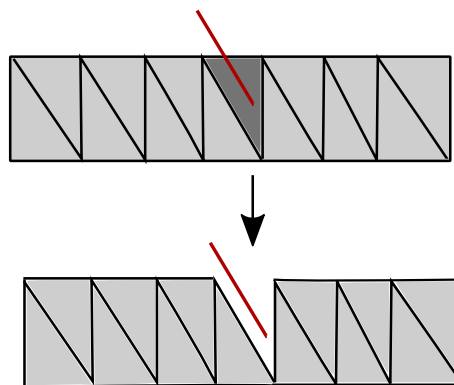


FIGURE 2.26 – Découpe avec suppression d'élément (face en 2D).

Grâce à sa simplicité et la simplification du maillage, cette méthode a été utilisée dans plusieurs approches [CDA00,FDA05] et elle a donné des résultats satisfaisants. Un exemple de résultat de l'utilisation de cette méthode dans les travaux de Cotin

et al. est donné sur la figure 2.27. Néanmoins, il faut signaler que la suppression des éléments engendre une perte de matière ce qui nuit au réalisme de la simulation. Pour pouvoir utiliser cette méthode tout en limitant la perte de masse, il faut que la taille des éléments soit petite ce qui conduit à simuler un nombre important d'éléments.

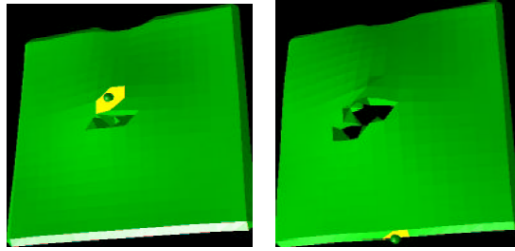


FIGURE 2.27 – Découpe d'une plaque par suppression de tétraèdre (issu de [CDA00]).

2.4.1.2 Séparation d'éléments

L'approche par séparation d'éléments consiste à séparer des éléments adjacents en « éclatant » leur bord commun. Le chemin de coupe est ainsi approximé par une suite d'arêtes en 2D ou de faces en 3D du maillage initial (voir figure 2.28). Cette méthode a été utilisée dans [Fri99] et [NvdS00]. Dans [NvdS00], un plan de coupe est défini par une surface triangulaire. Durant son déplacement, les faces du maillage les plus proches du plan de coupe sont éclatées. Pour réduire l'écart entre le chemin de coupe et les faces à éclater, les éléments du maillage sont déformés en rapprochant de la surface de coupe les nœuds les plus éloignés (méthode du *node snapping*).

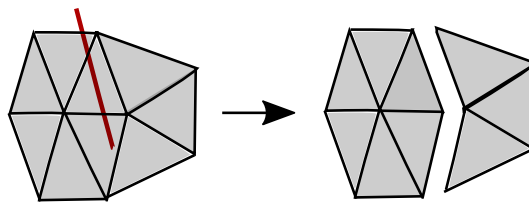


FIGURE 2.28 – Découpe avec séparation des éléments.

2.4.1.3 Division d'éléments

La division d'éléments consiste à couper en deux les éléments qui intersectent la surface de découpe. Plus précisément, l'intersection d'un élément avec la surface de découpe définit une arête en 2D ou une face en 3D qui sont insérées dans l'élément. Par l'éclatement de ces nouvelles cellules, l'élément est coupé en deux comme montré

sur la figure 2.29. Dans [BG00], Bielser *et al.* ont introduit cette méthode sur des tétraèdres et mis en évidence différentes configurations de découpe de l'élément.

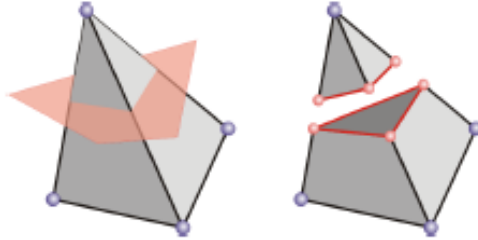


FIGURE 2.29 – Division d'un élément (issue de [WWD14]).

Ils ont montré que cette approche est graphiquement très réaliste parce que la découpe se fait exactement à l'endroit souhaité. Cependant cette méthode est assez coûteuse en temps de calcul et le nombre de tétraèdres augmente au fur et à mesure des coupes. Par ailleurs, cette méthode, purement géométrique, peut générer, selon le placement des cellules insérées, des éléments mal formés qui ne respectent pas les contraintes de proportion des éléments exigées par la simulation physique (les éléments *fins* et *longs* sont à éviter en simulation par éléments finis, par exemple).

2.4.1.4 Duplication d'éléments

Dans [MBF04,SDF07], la méthode des nœuds virtuels est introduite pour contourner les problèmes numériques qui résultent des éléments mal formés. L'idée de base est de créer un ou plusieurs duplicatas des éléments à couper. En chaque duplicata est défini un plan de coupe qui permet de séparer l'élément en deux parties : l'une contient de la matière, l'autre est vide. Ainsi, les duplicatas sont définis à la fois par des sommets originaux, qui étaient présents dans le maillage avant la découpe, et de nouveaux nœuds dits *virtuels*, qui permettent de définir topologiquement et géométriquement l'élément, mais sont situés dans la zone sans matière. Lors de la simulation, seules les parties contenant de la matière sont considérées, ce qui nécessite une adaptation mécanique. Un exemple de résultat de cette méthode est illustré dans la figure 2.30

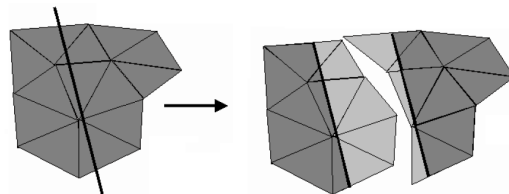


FIGURE 2.30 – Découpe avec duplication des éléments

L'un des inconvénients de cette méthode est l'utilisation d'un seul plan de coupe par élément. Aussi, des travaux postérieurs ont cherché à lever cette limitation [SDF07,WJST14]. Par ailleurs, cette méthode ne permet pas de garantir que le modèle obtenu après duplication est une quasi variété (cette notion est définie ultérieurement, dans la partie 3.3.2.2). Un élément peut, en effet, être lié à un élément adjacent et son duplicata par un même bord. Cela implique que le maillage soit décrit par un modèle compatible avec les duplications et limite fortement le recours aux modèles topologiques que nous décrivons dans le prochain chapitre.

2.4.1.5 Discussion

Pour plus d'informations sur les méthodes utilisées pour la découpe virtuelle des objets en 2D ou en 3D, il est possible de se référer à la publication de Wu *et al.* [WWD14] qui présente un état de l'art sur la simulation physique de découpe dans les objets déformables. Nous constatons que la découpe est délicate et que les différentes méthodes soit privilégient le maillage existant (suppression ou séparation d'éléments) soit modifient ce maillage, en introduisant de nouveaux éléments non nécessairement compatibles avec les exigences de la simulation physique. La méthode des nœuds virtuels apparaît comme un compromis entre ces deux familles d'approches, mais il contraint fortement le modèle décrivant le maillage de l'objet.

2.4.2 Fractures/déchirures

Les phénomènes de fracture et déchirure apparaissent dans un objet lorsqu'il subit des déformations trop importantes. Plusieurs travaux se sont focalisés sur l'étude de ces phénomènes, mais les méthodes sont toujours étroitement liées au modèle de comportement utilisé (discret ou continu). La déchirure a été principalement étudiée dans le cas des tissus (textiles ou biologiques). Au contraire, les fractures ont été largement étudiées, principalement pour deux types de matériaux, classique en ingénierie : les matériaux *ductiles* et les matériaux *fragiles*. En général, la principale différence entre ces deux types de matériaux est la capacité de déformation plastique que le matériau subit avant que la fracture se produise, en d'autres termes, sa rigidité. Les matériaux ductiles démontrent une grande quantité de déformation plastique avant la fracture alors que les matériaux fragiles se brisent dès les premières déformations. Dans ce manuscrit, nous considérons que la fracture et la déchirure sont des phénomènes qui peuvent être traités de la même manière, la différence de comportement étant plutôt due aux lois de comportement du matériau et non aux changements topologiques en tant que tels.

Pour déclencher une déchirure ou une fracture dans un objet, un critère est requis. Ce dernier permet de déterminer quand et où une déchirure/fracture apparaît, ainsi que la façon dont elle se propage dans l'objet. Plusieurs types de critères ont été définis dans la littérature. Ils peuvent être basés soit sur les forces ou les contraintes appliquées sur le corps [OH99] ou sur ses déformations [NTB⁺91]. Pour les modèles de type masses/ressorts, ces critères sont équivalents (car la force est

directement proportionnelle à la déformation du ressort, ou au carré, au cube, etc. de cette dernière). Lorsque la force ou la déformation dépassent un seuil, le critère est vérifié. Ce critère doit alors définir l'emplacement où une transformation topologique doit s'appliquer pour représenter la déchirure ou la fracture. Plusieurs types de transformations topologiques ont également été proposés dans la littérature, certaines étant similaires à celles utilisées pour la découpe, d'autres ayant été spécifiquement conçues pour les fractures.

2.4.2.1 Critères

Dans cette section, nous présentons les critères de déchirure/fracture que nous classifions, comme indiqué dans la discussion précédente, en approches basées sur les déformations (*strain*) et celles basées sur les forces/contraintes (*stress*).

- Critères basés sur les déformations

En 1988, Terzopoulos et Fleischer [TF88] ont présenté une technique pour simuler des déformations inélastiques et des fractures. Lorsque le déplacement d'un point de l'objet par rapport à sa référence non déformée excède un seuil de déformation, une fracture est enclenchée. La fracture se base complètement sur le schéma utilisé pour gérer la plasticité du modèle, en modifiant la forme de référence. L'objet étant représenté par une spline, la fracture est représentée visuellement via la mise à 0 de certains des coefficients gérant la continuité de la spline.

En 1991, Norton *et al.* [NTB⁺91] ont présenté une méthode pour animer des objets solides en 3D par un modèle masses/ressorts et qui se fracturent suite à une grande déformation. Lorsque la distance entre deux particules attachées dépasse un seuil (*i.e.* le ressort correspondant se déforme au delà d'une limite donnée), une fracture est appliquée. Des exemples de résultats sont donnés sur la figure 2.31.



FIGURE 2.31 – Exemples de résultats des objets avec la méthode de Norton (images issues de [NTB⁺91])

Ce même critère a été utilisé par plusieurs autres approches. Citons, par exemple celle de Hirota *et al.* [HTK00] pour simuler la fracture sur une surface d'argile, Boux

de Casson *et al.* [BL00] pour simuler la déchirure de tissus biologiques représentés par des mailles triangulaires en 2D (travail étendu plus tard aux tétraèdres en 3D). Un tel critère de déformation a également été utilisé dans [DKS⁺11] et [LBC⁺14].

- **Critères basés sur les forces ou les contraintes**

Dans les modèles utilisant des interactions entre couples de particules (modèle discrets en particulier), la force de l'interaction est directement liée à la déformation du lien d'interaction. De ce fait, les critères à base de forces peuvent être remplacés par des critères sur la déformation, ce qui peut être moins coûteux selon l'implantation de la simulation. Cependant, pour les modèles basés sur la mécanique continue, le tenseur de contrainte et celui de déformation sont liés par la loi de constitution, par exemple la loi de Hooke (équation (2.13) page 14). Compte tenu de cette loi, les critères sur les déformations et sur les contraintes fournissent des résultats différents.

La plupart des travaux qui se sont intéressés aux fractures des modèles continus se basent sur une résolution de type éléments finis. Les travaux fondateurs de O'Brien *et al.* en informatique graphique, montrent comment les tenseurs de contraintes calculés par une méthode des éléments finis peuvent être utilisés pour générer des fractures. Ils se sont intéressés aux matériaux fragiles [OH99] puis aux matériaux ductiles [OBH02]. Les maillages utilisés sont uniquement de type tétraédriques et l'interpolation de type linéaire. Les tenseurs de contraintes sont donc constants sur les éléments. La méthode revient à calculer un tenseur en chaque sommet à partir des tenseurs des tétraèdres auxquels les sommets appartiennent. Dans [OH99], les tenseurs des tétraèdres sont divisés en deux parties, pour séparer les effets de traction et de compression. Un *tenseur de séparation* est calculé, en chaque sommet, en combinant les tenseurs de traction des tétraèdres entourant le sommet, d'une part, et les tenseurs de compression, d'autre part. Récemment, Koshier *et al.* [KLB14] ont proposé un tenseur plus simple à calculer, en moyennant les tenseurs des tétraèdres incidents à chaque sommet et en les pondérant par la masse du volume. En chaque sommet, les valeurs propres de son tenseur sont calculées. Si au moins l'une d'elle dépasse un seuil pré-défini, la fracture est déclenchée. Un plan de fracture est défini comme orthogonal au vecteur propre associé à la valeur propre maximale et passant par le sommet. Notons que plusieurs autres approches ont utilisé les tenseurs aux sommets (tenseurs de séparation ou non) en 2D ou 3D, comme par exemple [PNdO14,BDW13,PO09,IO09,WRK⁺10]. Dans [OH99,KLB14], la méthode des éléments finis utilisée est dynamique, le corps est alors déformable (mais très faiblement). Néanmoins, il est possible de simuler le corps en tant que corps rigide et d'avoir recours à une méthode des éléments finis statique visant uniquement à calculer les tenseurs de contraintes correspondant à une déformation due à une collision pour décider des lieux de fracture [MMD⁺01,BHTF07,LHLW11].

2.4.2.2 Modifications topologiques

Une fois un critère vérifié, on en déduit l'emplacement d'une déchirure/fracture ainsi que certaines informations supplémentaires comme, par exemple, un plan de

fracture. Une transformation topologique doit alors être appliquée. Plusieurs types de transformations ont été proposés dans la littérature. Les différents types de modifications topologiques déjà cités dans la partie de découpe (section 2.4.1), peuvent également être utilisés pour les déchirures et fractures. Par exemple, dans [NTB⁺91], les auteurs ont choisi la suppression d'un ou plusieurs éléments du maillage. La séparation d'éléments est utilisée dans [SWB00] et [BL00]. Mais concernant les critères s'appliquant en un sommet du maillage, une opération plus complexe doit être définie afin de permettre d'éclater ce sommet en deux sommets, en s'appuyant sur le plan de la fracture. Dans [OH99, OBH02, KLB14] les tétraèdres intersectant le plan de fracture sont divisés (en utilisant, au besoin la méthode du nœud virtuel décrite en section 2.4.1.4). Dans [MMD⁺01], les tétraèdres incidents à un sommet sont regroupés en deux sous-ensembles, selon leur position par rapport au plan de fracture. Chaque face située entre deux tétraèdres de signes opposés est alors éclatée afin de séparer les deux tétraèdres. De cette manière, une « couronne » de faces est éclatée ce qui a pour conséquence d'éclater le sommet en deux. Cette méthode peut également être utilisée en 2D, on classe les faces en deux groupes que l'on sépare en éclatant les deux arêtes (en vert sur la figure 2.32) entre faces de signes opposés.

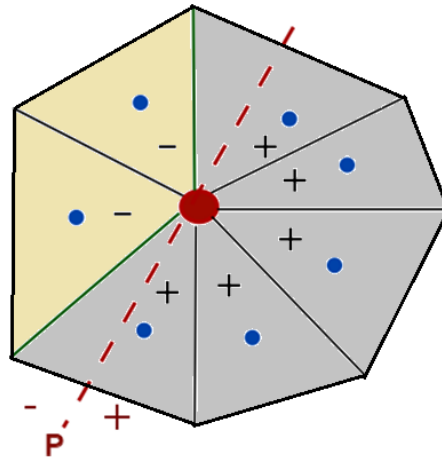


FIGURE 2.32 – Principe d'éclatement d'un sommet (cas 2D).

2.4.2.3 Lien entre les critères et les modifications topologiques

Nous venons d'exposer différents types de critères existants dans la littérature ainsi que quelques types de modifications topologiques qui se déclenchent suite à la vérification d'un critère. Compte tenu de ces différentes approches, il est légitime de se poser la question de la relation entre le critère et la modification topologique, en particulier lorsque l'emplacement du critère ne correspond pas au lieu de la modification. En effet, il apparaît que le lien entre les deux n'est pas immédiat et qu'il est même souvent arbitraire. Par exemple, dans [BL00] et [DKS⁺11], le critère utilisé est celui d'une trop grande elongation d'une arête, mais la modification topologique est l'éclatement d'un sommet. L'arête étant délimitée par deux

Critère	Modification Topologique	Références
Déformation	Éclatement de sommet	[SWB00, BL00, DKS ⁺ 11]
	Suppression d'élément	[NTB ⁺ 91]
	Division d'élément	[HTK00]
Force	Éclatement de sommet	[MMD ⁺ 01, OH99, OBH02, KLB14]
	Duplicata d'élément	[MBF04]
	Suppression d'élément	[CDA00]

TABLE 2.1 – Critères *vs* modifications topologiques de quelques travaux antérieurs.

sommets, le choix du sommet à éclater est parfaitement arbitraire, il peut tout au plus être conditionné par des considérations concernant la géométrie locale du maillage autour des sommets. Autre exemple, un critère de fracture calculé en un sommet résulte en l'éclatement de ce sommet, possible uniquement par éclatement de plusieurs arêtes/faces incidentes à ce sommet. La modification topologique dépasse largement la zone où le critère a été calculé. Enfin, il existe des combinaisons critères/modification topologique qui n'ont pas été étudiées. Le table 2.4.2.3, résume les critères et les modifications topologiques utilisés dans quelques travaux antérieurs. Comme décrit au dessus nous avons décomposé ces travaux en nous basant sur le type de critères utilisé, les déformations ou les forces internes, le lieu d'application (sommet ou élément) et la modification topologique qui est appliquée.

2.4.3 Adaptation de maillage

Dans la littérature, plusieurs approches ont considéré qu'il fallait modifier un maillage avant d'effectuer une déchirure ou une fracture, voire également une découpe. Par exemple, Nienhuys *et al.* [NvdS02] ont décidé de déformer le maillage initial pour correspondre au chemin de coupe. Les faces les plus proches de la surface de coupe sont identifiées et leur sommets sont projetés sur cette dernière (*node snapping*). Mais cette méthode, purement géométrique, peut conduire à des éléments mal proportionnés qui perturbent la simulation. Nous n'aborderons pas d'avantage cette méthode, car ce manuscrit se focalise sur les modifications de nature topologique et pas seulement géométrique. Aussi, dans cette partie, nous abordons des adaptations du maillage qui ne correspondent pas à des découpes, déchirures ou fractures. Ce sont, d'une part, les adaptations locales de maillage par réagencement des éléments et, d'autre part, les changements de résolution (raffinement/simplification de maillage).

2.4.3.1 Modification des éléments

Un réagencement des éléments d'un maillage vise à adapter le maillage avant de procéder à une modification topologique de type découpe/déchirure/fracture, afin que la simulation du phénomène soit plus précise et/ou pour éviter la forma-

tion d'éléments mal proportionnés. Par exemple, une opération classique en 2D, le basculement d'arête, consiste à modifier l'emplacement d'une arête entre deux triangles pour que le quadrilatère formé par les deux triangles soit découpé par l'autre diagonale.

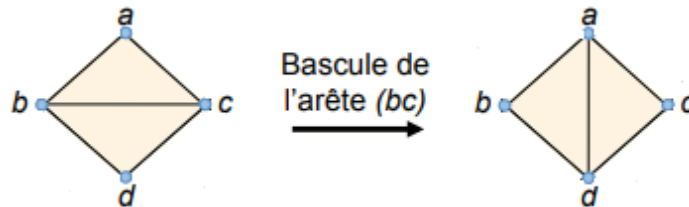


FIGURE 2.33 – Basculement d'arête.

On comprend sur la figure 2.33 que la nouvelle arête obtenue peut mieux approximer le chemin de coupe, que l'arête initiale. Cette idée intuitive est à la base des travaux de Paulus *et al.* [PUC⁺15]. Ils se basent sur une opération de raffinement plus élaborée, dite $\sqrt{3}$, qui nécessite de subdiviser, au préalable, les triangles en trois triangles, comme montré sur la figure 2.34(a).

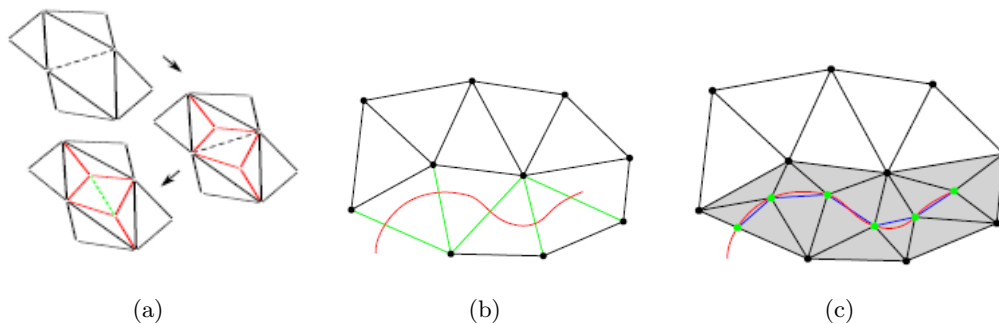


FIGURE 2.34 – Modification d'éléments avec le raffinement $\sqrt{3}$ (cas 2D).

En 2D, les arêtes traversées par la ligne de coupe sont d'abord sélectionnées, comme montré sur la figure 2.34(b). Ensuite, les triangles adjacents sont divisés, et les arêtes sélectionnées sont basculées afin de lier les nouveaux sommets (voir figure 2.34(c)), ce qui fournit des arêtes approxinant le chemin de coupe. Une version 3D a également été proposée, à base de division de tétraèdres et de la même manière qu'avec la version 2D.

Comme montré par la figure 2.36, si une des arêtes du tétraèdre est traversée par le plan de coupe, il faut divisé le tétraèdre en quatre sous-tétraèdres (Voir figure 2.35). Si le plan de coupe intersecte plus que trois arêtes, il faut alors insérer un sommet qui est le barycentre des intersections. Si moins de trois arêtes sont intersectées, les auteurs utilisent les points d'intersection entre les arêtes et l'exten-

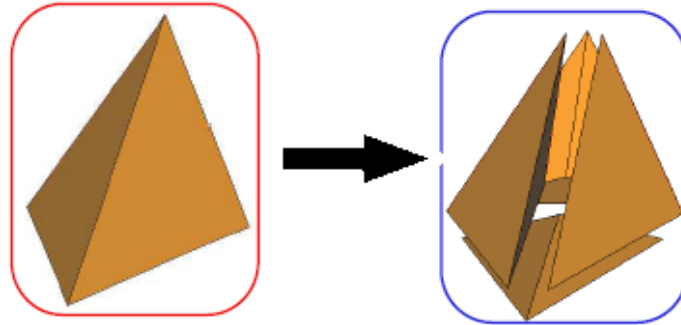
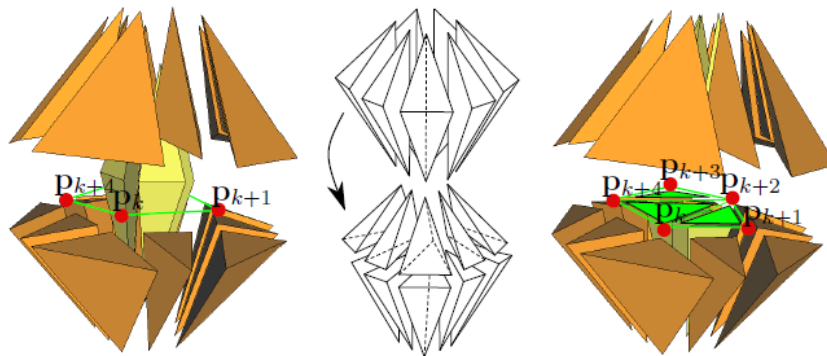


FIGURE 2.35 – subdivision d'un tétraèdre (1-4 split).

sion du plan de coupe. Après il faut ajouter les arêtes entre les sommets insérés. Finalement, la surface de séparation est triangulée comme montré sur la figure 2.36.

FIGURE 2.36 – Modification d'éléments avec le raffinement $\sqrt{3}$ (cas 3D).

Cette méthode permet d'avoir des éléments bien proportionnés, tout en limitant au maximum le nombre de nouveaux sommets insérés et donc les temps de simulation supplémentaires après la transformation.

Le coraffinement [Gui06] est également une méthode de modification d'un maillage permettant de le « fusionner » avec un autre maillage, typiquement, dans notre contexte, un motif pré-défini représentant une fracture. En effet, de nombreux travaux ont choisi d'utiliser un tel motif pour mieux contrôler le résultat final d'une fracture, comme par exemple [IO09, PO09, MCK13]. Ce sont alors les arêtes ou faces du motif qui sont éclatées et non celles du maillage initial. La figure 2.37 montre un exemple de résultat obtenu dans [IO09]. Ici, la fracture résulte de l'application d'un motif sur un verre plat.

2.4.3.2 Changement de résolution

Durant la simulation, il apparaît indispensable de définir un maillage de façon fine (donc avec beaucoup de petits éléments) là où l'objet est manipulé. Plus encore,

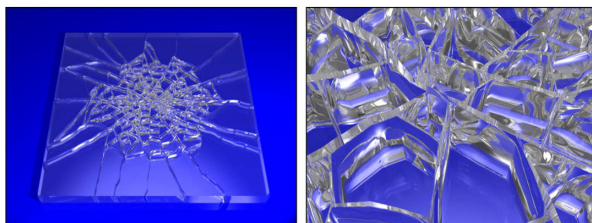


FIGURE 2.37 – Exemples de résultat des travaux de Iben *et al.* [IO09]

s'il subit une modification de type découpe, déchirure ou fracture, un changement de résolution peut permettre de simuler plus précisément le phénomène physique. Néanmoins, il est très coûteux de simuler des maillages à haute résolution (beaucoup d'éléments, beaucoup de degrés de liberté). Il semble donc utile de pouvoir modifier, à la volée, la résolution d'un maillage, c'est à dire le rendre plus fin, là où cela est utile (et on accepte le surplus de temps de calcul), et plus grossier là où il n'y a pas d'interaction et où l'objet n'est pas visible (d'où un gain en temps de calcul). On parle de *résolution adaptative*, qui est un procédé largement étudié dans plusieurs domaines de l'informatique graphique, en particulier, en animation physique.

L'un des travaux les plus connus pour la résolution adaptative est l'approche de Hutchinson *et al.* [HPH96]. Ils ont proposé une méthode adaptative pour un modèle mécanique masse/ressort pour simuler des tissus (maillage 2D) avec des plis (voir figure 2.38).

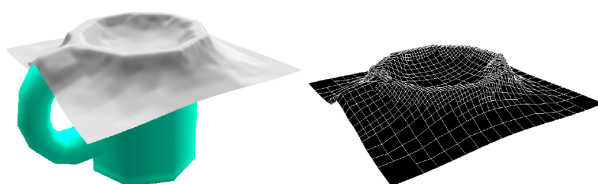
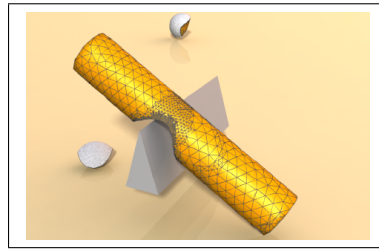
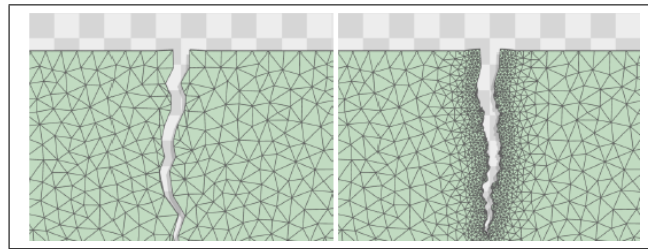


FIGURE 2.38 – Exemples de résultat des travaux de Hutchinson *et al.* [HPH96]

Dans cette méthode, le tissu est tout d'abord maillé grossièrement (niveau de résolution 0). Si, durant la simulation, l'angle entre deux ressorts, alignés et incidents à une même masse, s'éloigne trop des 180° , la zone est raffinée en ajoutant des masses et des ressorts. Ces derniers ont comme niveau de résolution 1. Cette opération peut être ré-itérée pour créer les niveaux de résolution supérieurs. De même, Villard *et al.* [VB05] ont également proposé une méthode adaptative pour la simulation d'un tissu 2D en se basant sur le modèle de Provot [Pro95]. Les auteurs montrent l'inconvénient d'utiliser un maillage grossier et proposent de raffiner localement des zones bien définies. Le critère qui est choisi pour raffiner est un critère géométrique dépendant de la courbure calculée en un sommet de maillage. Des approches utilisent des schémas de raffinement [WRK⁺10, KLB14, BDW13, PNdO14]... pour affiner la résolution d'un maillage et augmenter la précision des coupes, déchirures ou fractures (voir figure 2.39).



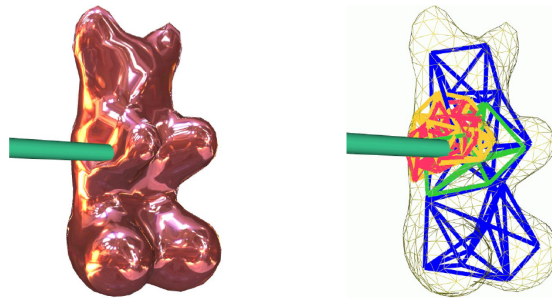
(a) Fracture (issue de [KLB14])



(b) Déchirure (issue de [BDW13])

FIGURE 2.39 – Exemples de changement de résolution lors de fractures et déchirures.

Dans les méthodes de résolution adaptative, la résolution du maillage dépend de la zone considérée et n'est pas constante sur tout le maillage. Une autre modélisation consiste à faire co-habiter plusieurs maillages d'un même objet à diverses résolutions, définis dès le départ de la simulation. Pour chaque traitement sur l'objet, l'application choisit la résolution la plus adaptée (par exemple, résolution grossière pour la simulation physique interactive, mais résolution fine pour la visualisation). Ainsi, Debunne *et al.* [DDCB01, DDCB00], proposent une animation en temps réel d'objets visco-élastiques qui offre des résultats avec une bonne précision dans les zones où il y a une grande force appliquée sur l'objet comme montré sur la figure 2.40. Le critère de raffinement utilisé est la valeur de courbure du champ de déformation.

FIGURE 2.40 – Exemples de résultat des travaux de Dubunne *et al.* [DDCB01]

L'objet est défini à différentes résolutions, mais ces dernières doivent être cohérentes. Aussi, les résolutions possèdent des nœuds dits *actifs* et d'autres dits *fantômes*. Dans une zone, seule la résolution utilisée à un instant donné contient des

nœuds actifs. Un nœud est dit fantôme, lorsque la résolution à laquelle il appartient n'est pas en cours d'utilisation pour la zone où il se trouve. Durant la simulation, il faut simuler uniquement les nœuds actifs, alors que les nœuds fantômes sont interpolés linéairement à partir des positions des nœuds des résolutions voisines. Lors d'un changement de résolution, les nœuds de la résolution choisie deviennent actifs et leur informations mécaniques (positions, vitesses) sont connues.

Signalons qu'il n'existe pas nécessairement un lien fort entre les diverses résolutions. Dans les méthodes de résolution adaptative, le maillage de résolution plus grossière est souvent inclus dans le maillage de résolution plus fine. En d'autres termes, l'opération de raffinement préserve le maillage existant ; les sommets sont gardés et participent à la définition du maillage fin, les arêtes, faces et volumes sont quant à elles subdivisées. On parle alors de modèle pyramidal [Kra08]. Néanmoins, il est possible de rendre les résolutions complètement indépendantes, comme l'ont proposé Debonne *et al.* [DDCB01, DDCB00].

2.4.4 Mises à jour du modèle mécanique

Après l'application de l'opération de modification topologique et comme les propriétés mécaniques de l'objet dépendent de la structure géométrique du maillage, une phase de mise à jour du modèle mécanique est nécessaire (masse, raideur des ressorts, matrices éléments finis...). plusieurs méthodes sont proposées dans la littérature. Par exemple concernant la mise à jour de masses, dans [HPH96], lorsqu'un sommet est ajouté, il récupère la même masse que les autres sommets du maillage. Ceci implique que la masse du corps croît au fur et à mesure des modifications topologiques. Dans d'autres travaux comme [FZDJ14, MDS10], la valeur de la masse d'un sommet est considérée comme la somme des contributions de chaque élément incident au sommet en question.

Pour les interactions, par exemple, lorsqu'un ressort est coupé en deux quand on insère un sommet au milieu de son arête support, les deux ressorts obtenus, deux fois plus courts que le ressort initial, doivent avoir une raideur double de la raideur de ce dernier pour préserver l'élasticité du corps (mise en série de deux ressorts) [HPH96]. Si de nouveaux ressorts apparaissent, leurs paramètres doivent pouvoir être calculés, à partir des caractéristiques mécaniques du corps modélisé. Nous devons alors revenir aux méthodes ayant permis d'identifier les paramètres du modèle maillé que nous avons présenté en section 2.2.2.3.

Il faut noter que les interactions en lien étroit avec les arêtes des maillages, par exemple les ressorts d'élongation ou les masses/tenseurs, sont généralement concernées par tout type de modification topologique (dès que l'arête support est modifiée par la transformation). Au contraire, certaines interactions, en lien avec les éléments, comme les ressorts de cisaillement ou les matrices éléments finis, ne sont concernés que par les modification de l'élément lui-même (par exemple, une subdivision). Dans tous les cas, quelles que que soit les interactions utilisées, toute modification impliquant des sommets nécessitent généralement un re-calcul de la masse du ou des sommets qui en proviennent.

2.5 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon des différents modèles physiques existants dans la littérature pour la simulation d'objets solides déformables. Ces modèles sont basés sur la mécanique discrète ou continue et diverses méthodes de calcul du comportement ont été exposées. Ensuite, nous avons présenté les différentes étapes de la boucle de simulation utilisée pour simuler les déformations des objets déformables en se basant sur n'importe quel modèle mécanique. Ce chapitre nous a également permis d'introduire de nombreux travaux portant sur les transformations topologiques des objets solides déformables pour la simulation des coupes, des déchirures et des fractures, ainsi que les adaptations locales de maillage et les changements de résolution.

Il faut se rendre compte ici qu'il existe un concept sous-jacent que nous n'avons pas encore réellement mis en évidence et qui est cependant au cœur de nos préoccupations. Il s'agit de la description topologique choisie pour expliciter les relations de voisinage (plus précisément d'adjacence et d'incidence) existant entre les diverses cellules (sommets, arêtes, faces, volumes) du maillage. Celle-ci est généralement peu décrite dans les publications, alors qu'elle est à la base du fonctionnement du modèle. Citons quelques exemples : un calcul élément fini sur un élément (face ou volume) doit pouvoir fournir des forces aux sommets de cet élément ; la masse d'un sommet du maillage est en lien direct avec les éléments situés autour de lui ; un ressort placé sur une arête doit pouvoir identifier les sommets sur lesquels sa force de déformation s'applique. Cette description topologique est encore plus nécessaire pour traiter les modifications topologiques de l'objet. Par exemple, pour séparer deux volumes (resp. deux faces), il faut connaître la face (resp. l'arête) qui les lie. Pour l'éclatement d'un sommet en 3D, tous les volumes autour de ce sommet doivent être identifiés et positionnés par rapport au plan de coupe ou de fracture, puis toutes les faces joignant des volumes à séparer doivent être identifiées pour être éclatées.

La plupart des approches citées dans ce chapitre se basent sur une structure topologique rudimentaire. Certaines relations de voisinage n'y sont pas forcément décrites de façon explicite, ce qui peut obliger à retrouver ces voisinages, par le biais d'algorithmes plus ou moins coûteux. En outre, rien ne garantit la robustesse des modèles topologiques naïfs : après une modification complexe, le maillage peut se trouver dans un état incohérent ou exhiber des dégénérescences comme des faces ou arêtes pendantes. Nous affirmons ici la nécessité de recourir à un modèle topologique capable de fournir les relations de voisinages nécessaires à la simulation mécanique ainsi qu'une grande robustesse lors des opérations de modifications du maillage. Aussi, dans le chapitre suivant, nous présentons un état de l'art sur les modèles topologiques.

Modèles topologiques

Sommaire

3.1	Introduction	53
3.2	Modélisation géométrique à base topologique	53
3.3	Modèles topologiques cellulaires	56
3.3.1	Modèles de représentation explicite de cellules	56
3.3.2	Modèles de représentation implicite de cellules	60
3.4	Modélisation à base de règles : Jerboa	68
3.4.1	Le langage Jerboa	69
3.4.2	Exemples de règles : création d'une surface en damier	81
3.4.3	Autre exemples de règles	86
3.5	Conclusion	90

3.1 Introduction

Le chapitre précédent a montré que pour simuler des évolutions topologiques d'objets physiques maillés, il était préférable de se baser sur un modèle dédié spécifiquement à la représentation de la topologie de l'objet. La modélisation géométrique, notamment à base topologique, nous offre une panoplie de modèles topologiques que ce chapitre se propose de brièvement présenter. Dans l'ensemble de ces approches, notre choix s'est porté sur le modèle des cartes généralisées, que nous présentons de façon plus approfondie dans ce chapitre. Ce choix est en grande partie motivé par l'utilisation de l'outil Jerboa, qui propose de manipuler les cartes généralisées à l'aide d'un langage à base de règles de transformation de graphes. Durant nos travaux, nous nous sommes entièrement reposé sur cet outil, à la fois pour la modélisation et la simulation de nos modèles. Aussi, ce chapitre se termine par la présentation de l'outil Jerboa proprement dit et principalement de la syntaxe de son langage de règles.

3.2 Modélisation géométrique à base topologique

La modélisation à base topologique vise à proposer des structures pour représenter graphiquement des objets sur ordinateur. Ces structures permettent de représenter, manipuler et modifier des objets géométriques structurés en différentes

dimensions. La diversité de ces structures est due à l'existence de plusieurs classes d'objets à représenter dans différents domaines d'applications comme l'architecture, la géologie, la mécanique, le domaine médical, les jeux vidéo...

En modélisation géométrique à base topologique, on différencie la topologie de l'objet de sa géométrie. La structure topologique est la décomposition de l'objet en un ensemble de cellules topologiques (sommets, arêtes, faces...). Une cellule de dimension i est appelée ***i*-cellule**, où i est inférieur ou égal à la dimension de l'objet considéré. Par exemple pour un objet de dimension 3, on peut définir les volumes en tant que *3-cellules*, les faces comme *2-cellules*, les arêtes comme *1-cellules* et finalement les sommets comme *0-cellules*. Par ailleurs, dans le cas des objets fermés, pour tout $i \geq 1$, les bords d'une i -cellule sont des cellules de dimension $i - 1$. Ces derniers constituent sa frontière. Citons par exemple un volume de dimension 3 qui est délimité par des faces, en d'autres termes des 2-cellules. Ces cellules sont liées par des relations de voisinage. Deux types de relations topologiques peuvent se présenter :

- **les relations d'incidence** : ces relations relient deux cellules voisines avec des dimensions différentes (*i.e* relient une cellule aux cellules de son bord). Par exemple, les arêtes sont incidentes à la face à laquelle elles appartiennent.
- **les relations d'adjacence** : ces relations relient deux cellules voisines de même dimension n qui partagent un bord (par exemple deux faces voisines qui sont liées par une arête ou deux volumes reliées par un sommet)

Le modèle géométrique de l'objet vise alors à compléter le modèle topologique en lui ajoutant une forme, une position, etc. Dans la littérature, il existe deux grandes familles de modèles généralement utilisés pour représenter les objets : les modèles de construction géométrique de solides (CSG) et les modèles de représentation par les bords (B-Rep).

Les modèles de construction géométrique de solides (CSG)

Le modèle de construction géométrique de solides a été introduit par Laidlaw *et al.* [LTH86]. Il représente un objet par un arbre dont les feuilles sont les composantes élémentaires et les nœuds sont les opérations booléennes (la différence, l'union, l'intersection...) comme l'illustre la figure 3.1. Il aide à construire rapidement la forme d'un objet, mais son inconvénient est qu'il ne représente pas explicitement l'objet construit mais uniquement son processus de construction. De plus, la représentation d'un objet avec cette méthode n'est pas unique, plusieurs constructions pouvant aboutir au même objet.

Les CSG sont généralement utilisés pour modéliser un objet via son processus constructif. Cela peut être utile en modélisation, voire en conception et usinage par ordinateur, mais ce modèle se révèle inadapté pour la simulation de solides déformables.

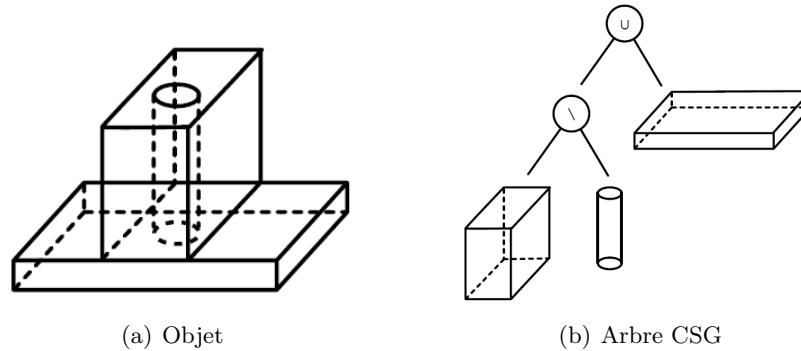


FIGURE 3.1 – Exemple de représentation d'un objet par arbre CSG.

Les modèles de représentation par les bords

Au contraire des modèles CSG, la représentation par les bords (*B-rep*) sert à représenter un objet par le bord qui le délimite, c'est à dire sa surface en 3D, ou sa courbe en 2D. Souvent, ces modèles sont décomposés en deux parties, une partie topologique définie par les faces, les arêtes, les sommets et leurs relations d'adjacence et d'incidence et une partie géométrique définie par les points, les courbes et les surfaces. Un exemple de représentation d'un objet par les bords est donné à la figure 3.2.

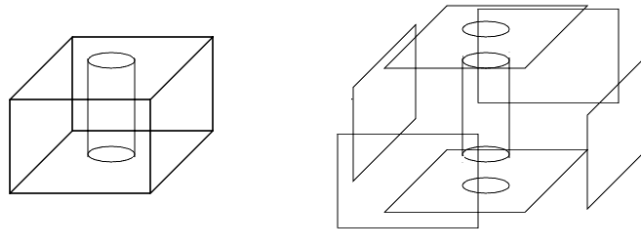


FIGURE 3.2 – Une représentation par les bords d'un objet.

Par rapport au modèle CSG qui utilise uniquement les composantes de l'objet et des opérations booléennes, les modèles B-rep sont plus flexibles et offrent plusieurs types d'opérations (chanfreinage, extrusion...). Les modèles de représentation par les bords les plus utilisés sont **les modèles cellulaires** (voir la figure 3.3). Ils autorisent la représentation d'un objet en utilisant des cellules régulières ou quelconques. Les modèles qui consistent à décomposer un objet en un ensemble de cellules régulières (plus particulièrement des triangles) dans des dimensions différentes (*i.e* sommets, arêtes, triangles, tétraèdres...) sont appelés **les modèles simpliciaux** (voir la figure 3.3(a)). Au contraire, il existe **les modèles cellulaires généraux** (voir la figure 3.3(b)) qui permettent la décomposition d'un objet en un ensemble de cellules quelconques de dimensions différentes (*i.e* sommet, arête, face, volume...).

Dans le reste de ce manuscrit nous nous intéressons uniquement aux modèles cellulaires, seuls modèles utilisés par notre approche. En effet, nous avons choisi

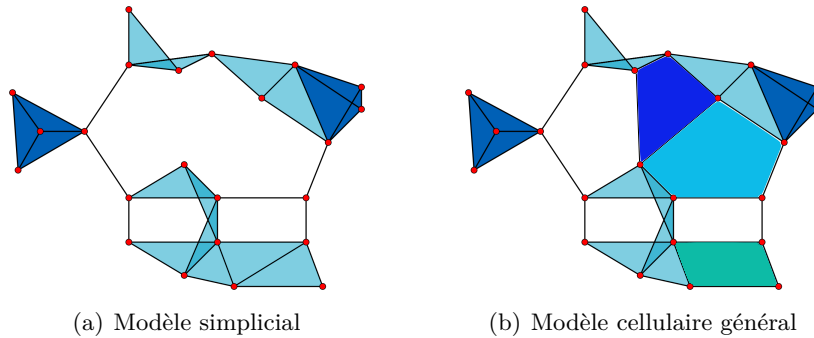


FIGURE 3.3 – Modèle simplicial et modèle cellulaire.

de travailler avec un modèle cellulaire parce que l'un des objectifs de cette thèse est de créer une plate-forme générale qui permet de simuler des objets déformables représentés par différents types de maillages, potentiellement hétérogènes, c'est à dire avec différents types de n -cellules en dimension n . En utilisant des modèles simpliciaux, il est impossible de représenter directement des maillages rectangulaires en 2D, hexaédriques en 3D et, en conséquence tout maillage mélangeant les éléments, hormis en re-subdivisant chaque élément en simplexes (triangles, tétraèdres, etc.). Nous introduisons dans la suite quelques modèles cellulaires : les graphes d'incidence, puis les cartes combinatoires orientées, puis finalement, les cartes généralisées que nous avons exploitées dans notre travail.

3.3 Modèles topologiques cellulaires

3.3.1 Modèles de représentation explicite de cellules

Graphes d'incidence

Le graphe d'incidence est un modèle cellulaire général qui permet de représenter des objets en tant que graphes orientés sans cycle. C'est un graphe dans lequel les nœuds correspondent aux cellules topologiques et les arcs du graphe correspondent aux relations d'incidences qui lient une cellule de dimension i à ses bords de dimension $(i - 1)$.

Un exemple de graphe d'incidence est donné par la figure 3.4. Les faces $F1$ et $F2$, qui sont des 2-cellules, ont comme filles les arêtes (1-cellules) (a, b, c, d, e et f) qui leur sont incidentes. De même les arêtes ont comme fils les sommets (0-cellules) qui leur sont incidents. Par exemple, la face $F1$ est incidente à l'arête b (b est l'une des arêtes de la face $F1$) qui est elle-même incidente au sommet A (A est une extrémité de b). Les relations d'adjacence se déduisent par les relations d'incidence à une même cellule : par exemple, les arêtes a et b sont incidentes à un même sommet A , donc elles sont adjacentes l'une de l'autre. Les relations d'incidence entre cellules de dimensions non consécutives se retrouvent par transitivité des relations du graphe d'incidence : la face $F1$ est incidente au sommet A , par l'intermédiaire des arêtes a

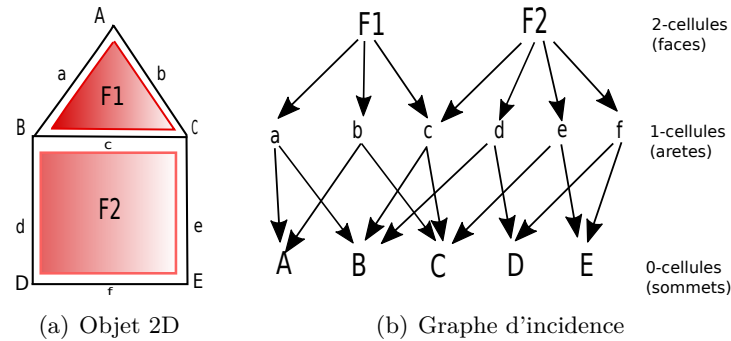


FIGURE 3.4 – Graphe d'incidence d'un objet 2D.

et b .

Cependant, l'exemple présenté par la figure 3.5 montre qu'un même graphe d'incidence peut représenter plusieurs objets ayant des topologies différentes parce qu'il ne présente pas d'une manière explicite l'ordre dans lequel les cellules sont placées les unes par rapport aux autres. Ainsi, sur la figure, les parcours de la face F_1 dans le sens trigonométrique de l'objet 1 (figure 3.5(a)) puis dans l'objet 2 (figure 3.5(b)), fournissent respectivement les deux suites d'arêtes suivantes : (a, c, e, d, f, b) et (a, c, f, d, e, b) . Comme on peut le remarquer ces deux suites sont différentes. On obtient également deux suites de sommets différents. Cette différence n'apparaît malheureusement pas dans le graphe de la figure 3.5(c). Le modèle n'est pas ordonné, il en résulte des ambiguïtés. Or, pour l'implantation de certains modèles mécaniques, nous avons besoin de connaître l'ordre des sommets, en particulier pour le modèle des éléments finis. Pour cette raison, dans ce document, nous avons choisi de nous placer dans le cadre des modèles ordonnés.

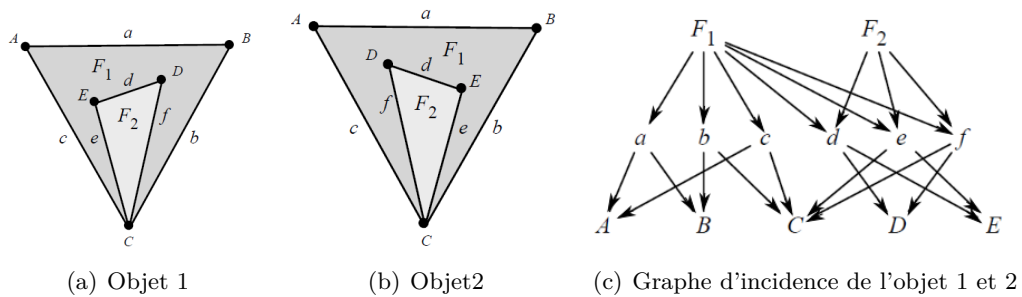


FIGURE 3.5 – Graphe d'incidence ambigu.

Graphes d'incidence indicés

Les graphes d'incidences indicés sont une implantation particulière des graphes d'incidence sous une forme ordonnée. Ce modèle est très répandu : on le trouve dans des formats de fichiers graphiques comme OBJ par exemple. Il est à la base de la plupart des fichiers de description de maillages d'éléments finis comme, par

exemple, le format msh utilisé par Gmsh [GR09]. Il est également utilisé par les *Vertex Arrays* proposés dans la bibliothèque graphique OpenGL. Il est, enfin, le modèle topologique de base proposé dans la bibliothèque SOFA [Sof], car c'est un modèle potentiellement très rapide. Ceci est dû à la structure en tableau qui évite l'éparpillement mémoire et une utilisation efficace des caches des processeurs.

Dans ce modèle, chaque cellule est désignée par un indice dans un tableau dédié aux cellules de sa dimension. Il y a donc autant de tableaux que de dimensions de cellules utiles à l'application. Pour chaque cellule, le tableau contient les relations d'incidence, voire d'adjacence, dont l'application a besoin, en désignant les cellules voisines par leur indice. Des informations géométriques peuvent être adjointes à chaque type de cellule. La plupart du temps, on associe au moins une position à chaque sommet.

À titre d'exemple, la figure 3.6 illustre une structure de base où sont d'abord énumérés les sommets (lignes démarrant par « v » pour *vertex*), essentiellement caractérisés par leur coordonnées de position. L'indice des sommets correspond à leur ordre d'apparition dans le fichier (en démarrant à 1). Ces sommets sont suivis par les faces triangulaires (lignes démarrant par « f »), définies seulement par les liens d'incidence avec leurs sommets (désignés par leur indice). Notons que l'ordre des sommets est important puisqu'il définit l'orientation de la face (le modèle est ainsi effectivement ordonné).

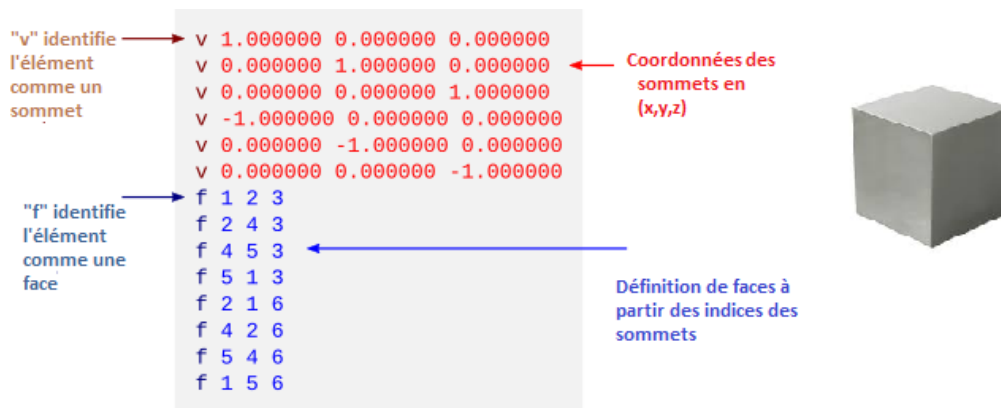


FIGURE 3.6 – Exemple d'un fichier OBJ.

Nous voyons que ce type de modèle est assez souple et peut s'adapter aux divers besoins de l'application, tant sur le plan des relations d'incidence/adjacence (on ajoute des indices supplémentaires dans les propriétés des cellules) que concernant les informations non topologiques à associer aux cellules (tout type d'information peut être ajouté aux cellules, à l'image des coordonnées des sommets). On peut le trouver sous une forme très épurée, comme dans le fichier OBJ (voir figure 3.6), ou sous une forme beaucoup plus riche.

Cette souplesse se paye néanmoins par un compromis : toute information de

voisinage non explicitement stockée dans la structure mais nécessaire à l'application doit être reconstituée. Pour cela, il faut, la plupart du temps, recourir à des algorithmes de recherche (sans tri) de complexité linéaire en nombre de cellules. Par exemple, dans le cas de la structure OBJ, pour trouver tous les triangles incidents à un sommet, il est nécessaire de parcourir l'intégralité de la liste des triangles pour trouver tous ceux qui référencent l'indice du sommet. Pour éviter ces recherches, il est possible d'enrichir la structure afin de pré-stocker l'information de voisinage.

Cependant, ce stockage pose deux problèmes. D'abord, il n'est pas de taille définie *a priori*, ce qui impose le recours à des structures à allocation dynamique (type tableaux de taille non fixe ou listes). Par exemple, le nombre de triangles autour de chaque sommet dépend du sommet considéré. Le second problème concerne le maintien de la cohérence des informations après une modification de la structure. Les graphes d'incidence indicés ne proposent, en effet, aucune contrainte de cohérence, *a priori*. Par exemple, sans contrôle de cohérence spécifique, un sommet pourrait potentiellement référencer une face comme incidente, alors qu'il n'est pas listé comme sommet de cette face. Plus la structure est complexe (i.e. avec de nombreux liens d'adjacence et d'incidence additionnels), plus il faut ajouter des contrôles de cohérence dédiés. Ces contrôles visent à assurer la réciprocity des relations d'adjacence et d'incidence, mais aussi les relations obtenues par composition. Par exemple, si un sommet est incident à une arête qui est elle-même incidente à une face, alors le sommet est incident à la face. Ou, comme autre exemple, si deux faces sont incidentes à une même arête, elles sont adjacentes. Ainsi, l'exhaustivité des contraintes de cohérence est difficile à garantir.

Par ailleurs, les graphes d'incidence indicés n'offrent aucune contrainte sur le type d'objet constructible (hormis le fait d'être un assemblage de cellules). Il est alors facile d'obtenir, par exemple après une modification topologique, des assemblages non obligatoirement désirés, comme, par exemple, deux faces ou deux volumes reliés par un seul sommet. Dans [FDA05], Forest *et al.* recensent un nombre important de cas topologiquement dégénérés mais codables dans le modèle, qui nécessitent des traitements supplémentaires pour être évités.

Finalement, on peut également critiquer le fait que les graphes d'adjacence indicés ne séparent pas nettement les informations d'ordre topologique des autres informations (tout est stocké dans un même tableau). Or, en cas de modification topologique, ce sont, d'abord et avant tout, les informations topologiques qu'il faut modifier et gérer de façon cohérente. Les autres informations peuvent être traitées dans un stade ultérieur où les nouvelles relations d'incidence et d'adjacence pourront être exploitées pour les mises à jour.

Il semble ainsi assez clair qu'un modèle plus rigoureux autant sur le plan des objets modélisables que de la cohérence des relations de voisinages est nécessaire sitôt que les modifications topologiques du modèle sont fréquentes. La suite de ce chapitre présente, à ce titre, des modèles beaucoup plus solides sur le plan topologique, évitant les écueils que nous venons d'exposer : les diverses familles de cartes combinatoires.

3.3.2 Modèles de représentation implicite de cellules

Les principaux inconvénients des graphes d'incidence exposés dans la section précédente proviennent essentiellement de la représentation qui est basée sur les cellules. En utilisant des éléments plus abstraits, il est possible de mieux définir et manipuler les relations topologiques. Les cellules sont alors représentées de façon implicite, à partir d'éléments abstraits. Des modèles de ce type ont d'abord été proposés en 2D avant d'être généralisés.

3.3.2.1 Modèles 2D

Les modèles topologiques 2D sont apparus dans les années 70. Parmi ces structures nous pouvons citer les modèles qui sont basés sur les arêtes. Ces derniers permettent et facilitent la recherche des informations d'adjacence entre les faces, les arêtes et les sommets. Parmi ces modèles, se trouve le modèle des arêtes ailées (*winged-edges*) [Bau75] et des demi-arêtes (*half-edges*) [Man88].

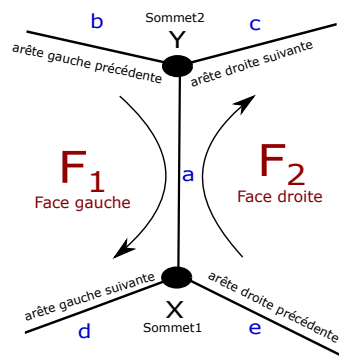


FIGURE 3.7 – Représentation d'un objet 2D avec la structure des arêtes ailées.

Le modèle des arêtes ailées est un modèle cellulaire orienté. Il se base sur un seul type de cellule, les arêtes, qui sont à ce titre orientées. Chaque arête regroupe huit informations détaillées sur la figure 3.7 :

- d'une part, les informations non topologiques (souvent géométriques) concernant les sommets de l'arête, X et Y (par exemple leur position), et concernant les deux faces incidentes à l'arête considérée, F_1 et F_2 (par exemple leur normale et/ou leur couleur).
- d'autre part, les informations topologiques, qui se résument en réalité à référencer l'arête précédente et l'arête suivante sur chacune des faces incidentes, F_1 et F_2 .

Comme on peut le remarquer sur la figure, les arêtes b , c , d et e sont les « ailes » de l'arête a . L'inconvénient de cette structure est que, pour une seule arête, il existe deux parcours (un pour chaque face) selon que l'on utilise ou pas le sens de l'orientation choisi pour l'arête.

La structure des demi-arêtes est apparue pour améliorer la structure des arêtes ailées. Chaque arête est décomposée en deux demi-arêtes orientées comme montré sur la figure 3.8. Alors qu'une arête ailée est liée à deux faces, une demi-arête n'est liée qu'à une seule face. Les deux demi-arêtes font référence l'une à l'autre, ce qui reconstitue l'arête et permet de représenter directement l'adjacence de deux faces. Chaque demi-arête recense également la demi-arête suivante (en noir) et la précédente (en bleu) sur la face. De la même manière que pour les arêtes ailées, il est possible d'attacher à la demi-arête les informations non topologiques concernant son sommet de départ (dans notre exemple le sommet X) ainsi que celles concernant sa face (ici F_1).

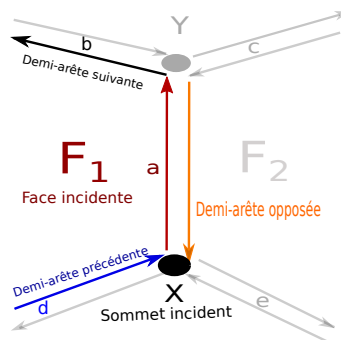


FIGURE 3.8 – Représentation d'un objet 2D avec la structure des demi-arêtes.

Les cellules topologiques de cette structure (faces, arêtes et sommets) sont représentées implicitement. En effet, une arête apparaît comme l'union de deux demi-arêtes. Une face apparaît comme un ensemble de demi-arêtes successives. Ces demi-arêtes référencent les mêmes informations de face. Un sommet regroupe l'ensemble des demi-arêtes qui partent de ce sommet. Pour le sommet X , cet ensemble contient la demi-arête a . Reconstruire les cellules nécessitent ainsi un parcours de la structure.

Les demi-arêtes permettent de représenter uniquement des objets 2D de type variété, avec ou sans bord selon que l'on oblige ou pas chaque demi-arête à s'associer à une autre demi-arête pour former une arête. Cette structure a pu être formalisée et généralisée aux dimensions supérieures à 2 par les cartes combinatoire orientées que nous présentons dans la section suivante.

3.3.2.2 Cartes combinatoires orientées

Il est possible d'étendre la formulation des demi-arêtes 2D aux dimensions supérieures grâce au formalisme des *cartes combinatoires orientées* (ou plus simplement « cartes orientées ») [BS85, DL14]. Les cartes orientées permettent de représenter les différentes subdivisions de l'espace et les relations d'incidence entre les cellules. Elles sont définies mathématiquement en dimension quelconque.

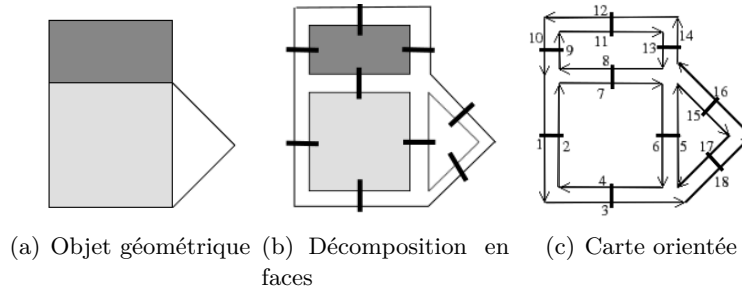


FIGURE 3.9 – Décompositions successives d'un objet géométrique 2D pour trouver la carte orientée correspondante.

Une carte orientée peut être définie par décompositions successives des différentes cellules d'un objet, comme on peut le voir sur la figure 3.9(a), pour un objet en dimension 2. Tout d'abord, cet objet est décomposé en faces liées le long d'une arête. Sur la figure 3.9(b), la relation d'adjacence entre deux faces est représentée par un trait noir que l'on peut appeler β_2 . De même, les faces sont subdivisées en arêtes. Ainsi, chaque arête est constituée de deux demi-arêtes qui sont appelées *brins* de la carte orientée présentée sur la figure 3.9(c). Chaque brin indique quel est son successeur pour la face dans laquelle il se trouve, par une liaison que l'on peut appeler β_1 . Chaque brin est ainsi lié à deux autres brins et la liaison se représente sous la forme d'une flèche.

Cet exemple représente un objet 2D, mais avec les cartes orientées, des objets de toute dimension peuvent être représentés. Les brins sont liés à autant de brins que l'exige la dimension, ce qui implique d'ajouter des liaisons β_n . Par exemple en 3D, il faut définir la liaison β_3 pour lier deux volumes.

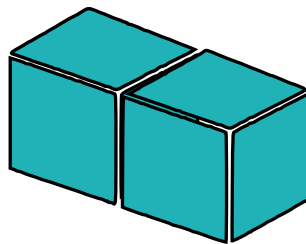


FIGURE 3.10 – Exemple de quasi-variétés.

Les cartes orientées permettent de représenter des objets de type quasi-variétés et orientables. Une quasi-variété est un ensemble de cellules de dimension n qui sont liées le long d'une cellule de dimension $(n - 1)$. Par exemple, sur la figure 3.10, les deux cubes (cellules de dimension 3) sont liés par une face (cellule de dimension 2).

Quelques exemples d'objets non quasi-variétés sont présentés sur la figure 3.11. Deux cubes (3-cellules) liés le long d'une arête (1-cellule) (voir la figure 3.11(a)) ou par un sommet (0-cellule) (voir la figure 3.11(b)) ne sont pas des quasi-variétés et ne sont donc pas représentables avec des cartes orientées.

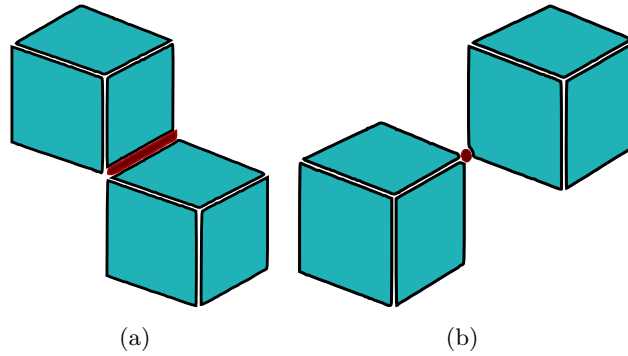
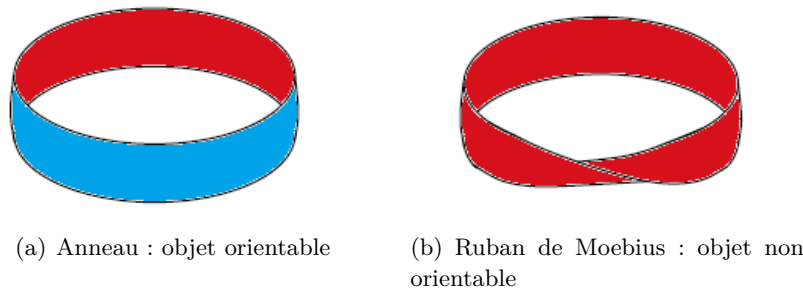


FIGURE 3.11 – Exemples de non-variétés.



(a) Anneau : objet orientable (b) Ruban de Moebius : objet non orientable

FIGURE 3.12 – Exemple d'objets orientable et non orientable.

Une surface est dite orientable, si elle admet deux faces : intérieure et extérieure (si la surface est fermée) ou recto et verso (sinon). Par exemple, l'anneau de la figure 3.12(a) est une surface orientable, car il a deux faces (respectivement rouge et bleue sur la figure). Au contraire, le ruban de Moebius de la figure 3.12(b) est non orientable, car il n'admet qu'une seule face. Cette notion d'orientation est extensible en toute dimension. Notons que certaines implantations du modèle des cartes orientées peuvent être restreintes aux objets fermés, car ce choix est plus efficace sur le plan algorithmique.

Comme pour les modèles 2D précédemment présentés, les cartes orientées ne représentent pas explicitement les cellules. Ces dernières apparaissent comme un ensemble de brins et leurs liaisons. Pour définir la géométrie et fournir aux cellules des données applicatives, il est donc, là aussi, possible d'ajouter des informations non topologiques, en les référençant par leurs brins. Ces informations sont appelées *plongements*. Cependant, contrairement aux premiers modèles topologiques proposés qui cantonnent les informations aux seules cellules, il est possible d'ajouter des plongements à des ensembles de brins ne correspondant pas obligatoirement à des cellules. Nous présentons et discutons des contraintes de ce mécanisme dans la section suivante.

3.3.2.3 Cartes combinatoires généralisées

Les *cartes combinatoires généralisées* ou plus simplement cartes généralisées (ou *G-cartes*) [Lie89, Lie91] permettent de représenter des quasi-variétés, avec ou sans bord, orientables ou non. Ce modèle se distingue ainsi des cartes orientées par sa capacité à représenter des objets non orientables, mais également par le fait que sa définition est homogène en toute dimension, car tous les liens qu'il utilise sont non orientés.

Initialement, les G-cartes sont définies algébriquement [Lie89]. Dans les travaux de Mathieu Poudret [Pou09] et Thomas Bellet [Bel12], les G-cartes ont été redéfinies en tant que graphes afin qu'elles puissent être manipulées par des transformations de graphes. Dans ce manuscrit, nous privilégions cette définition car l'outil que nous utilisons se base sur ce formalisme.

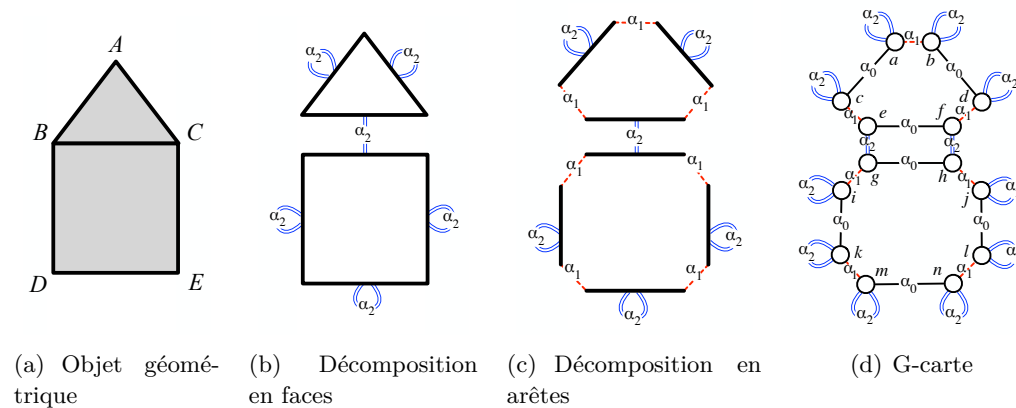


FIGURE 3.13 – Décompositions successives d'un objet géométrique 2D.

Comme pour les cartes orientées, la représentation d'un objet par une G-carte consiste à le décomposer successivement en cellules topologiques (volumes, faces, arêtes, sommets...) de dimension décroissante. Un exemple de décomposition d'un objet 2D est donné sur la figure 3.13. Dans cet exemple, l'objet initial (figure 3.13(a)) est décomposé en deux faces (figure 3.13(b)) qui sont liées le long d'une arête commune par des liens dits « de dimension 2 » nommés α_2 , car ils lient des cellules de dimension 2. Ensuite, ces deux faces sont décomposées en arêtes (figure 3.13(c)) qui sont liées par des liens dits « de dimension 1 » nommés α_1 . Finalement, chaque arête est décomposée en deux sommets (figure 3.13(d)), qui sont liés par des liens de dimension 0 nommés α_0 . Ces « sommets », plus précisément ces sommets issus d'une arête elle-même issue d'une face (qui est éventuellement issue d'un volume, etc.) sont appelés *brins*. Ils constituent les nœuds du graphe de la carte généralisée et les liaisons constituent les arcs du graphe.

Avant de continuer cette partie, tout au long de ce manuscrit, pour des raisons de simplification de lecture des schémas, nous utilisons une convention graphique introduite sur la figure 3.14. Les liaisons α_i sont représentées par des lignes noires pour α_0 , un trait rouge pointillé pour α_1 , et un double trait bleu pour α_2 .

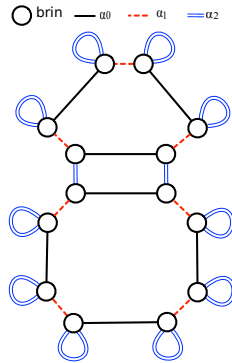


FIGURE 3.14 – Conventions graphiques.

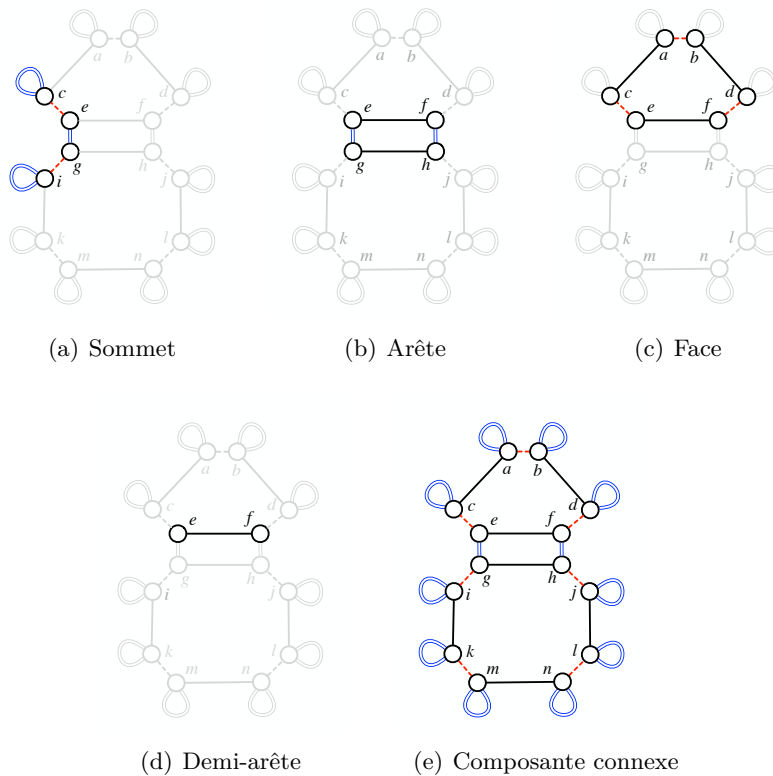


FIGURE 3.15 – Exemples des orbites adjacentes au brin e .

A l’instar des cartes orientées, les cellules topologiques des objets (volumes, faces, arêtes, sommets...) ne sont pas représentées explicitement dans les G-cartes. Elles sont définies à partir de sous-graphes appelés *orbites*. Par exemple, le sommet B (figure 3.13(a)) est défini sur la figure 3.15(a), par le sous-graphe comprenant le brin e et l’ensemble des brins atteignables par les liens de dimensions 1 et 2 (*i.e.* les brins c , e , g , et i) et l’ensemble des liaisons correspondantes. Cette orbite est notée $\langle \alpha_1, \alpha_2 \rangle(e)$. L’arête BC du même objet (figure 3.13(a)) est définie par le sous-graphe

comprenant le brin e et l'ensemble des brins atteignables par les liens de dimensions 0 et 2 (*i.e.* les brins $e, f, g,$ et h) et les liaisons correspondantes. Cette orbite se note $\langle \alpha_0, \alpha_2 \rangle(e)$ (voir figure 3.15(b)). La face ABC est définie par l'orbite $\langle \alpha_0, \alpha_1 \rangle(e)$ qui est le sous-graphe qui contient le brin e et l'ensemble des brins atteignables par les liens de dimensions 0 et 1 (*i.e.* les brins a, b, c, d, e et f sur la figure 3.15(c)) et les liaisons elles-mêmes.

Il faut préciser que les cellules ne sont que des cas particuliers d'orbites. Il existe, en effet, d'autres orbites. Par exemple, la demi-arête, que l'on appellera désormais en 2D *arête de face* par souci de précision et pour lever toute ambiguïté, contient les brins e et f sur la figure 3.15(d) ce qui correspond à l'orbite $\langle \alpha_0 \rangle(e)$. La composante connexe, c'est à dire, dans ce cas, tout l'objet (figure 3.15(e)), est désignée par l'orbite $\langle \alpha_0, \alpha_1, \alpha_2 \rangle(e)$. Lorsque l'orbite ne fait mention d'aucun α_i , alors il s'agit de l'orbite brin comme $\langle \rangle(e)$, qui ne comprend que le brin e lui-même. Nous définissons ainsi la notion d'orbite : pour un sous-ensemble orb de $\{\alpha_0, \dots, \alpha_n\}$ et b un brin d'une G-carte G de dimension n , l'**orbite** $\langle orb \rangle(b)$ est définie par le sous-graphe de G qui contient b , tous les brins de G atteignables par une suite de liaisons étiquetés par les éléments de orb , ainsi que ces liaisons.

Une carte généralisée de dimension n peut ainsi être définie comme un graphe¹ G dont les arcs sont étiquetés par α_i où i appartient à $[0..n]$ et qui satisfont les contraintes de cohérence topologique suivantes :

- non orientation : G est non orienté
- arcs adjacents : chaque brin est la source exactement de $n + 1$ liaisons respectivement étiquetées de α_0 à α_n
- cycles : pour tous indices i et j tels que $0 \leq i \leq i + 2 \leq j$, il existe un cycle² étiqueté par $\alpha_i \alpha_j \alpha_i \alpha_j$ dans G à partir de chacun de ses brins.

Cette définition et plus précisément la troisième contrainte permet d'assurer que l'objet est une quasi-variété. De plus, ces contraintes restent valides pour le bord de l'objet en utilisant les liaisons qui bouclent sur les brins du bord.

Comme pour les cartes orientées, des plongements (informations géométriques ou autres selon l'application visée) peuvent être ajoutés à la structure topologique. Pour contrôler la cohérence des plongements, on les associe obligatoirement à une orbite, que cet orbite représente une cellule ou pas. Ces plongements sont représentés par des étiquettes placées sur les brins de la G-carte comme défini dans [Bel12].

1. Soient n une dimension topologique et Π une famille d'informations π de type τ . Un graphe $G = (V, E, s, t, \alpha, \Pi)$ de dimension n et d'information Π est défini par un ensemble de brins V (les nœuds) et des liaisons E (les arcs), les fonctions source $s : E \rightarrow V$, cible $t : E \rightarrow V$ et d'étiquetage topologique $\alpha : E \rightarrow [0..n]$ tel que pour toute liaison $e \in E$, $s(e)$ est le brin source de e et $t(e)$ son brin cible et $\alpha(e)$ est sa dimension topologique, et des fonctions d'information $\pi : v \rightarrow \tau$ pour tout π de Π tel que pour tout brin de V , $\pi(v)$ est l'information π du brin v .

2. Un chemin est une suite de liaisons $e_1 \dots e_k$ tel que pour tout $1 < i < k - 1$ $t(e_i) = s(e_{i+1})$. Le chemin admet une source $s(e_1)$, une cible $s(e_k)$ et une étiquette topologique $\alpha(e_1)\alpha(e_1) \dots \alpha(e_k)$. Le chemin est un cycle si sa source et sa cible sont identiques $s(e_1) = t(e_k)$.

Pour qu'un plongement soit cohérent, une contraintes doit être satisfaite : il faut assurer que, pour tout plongement π porté par une orbite $\langle o \rangle$, tous les brins de cette orbite portent la même information. Un plongement se définit formellement sous la forme : $\pi : \langle orb \rangle \rightarrow \tau$, où π est le nom du plongement, $\langle orb \rangle$ est le type d'orbite auquel il est associé, et τ est le type de l'information. On note Π la famille des plongements π . Une carte généralisée est dite *plongée* lorsqu'on lui associe un ou plusieurs plongements π .

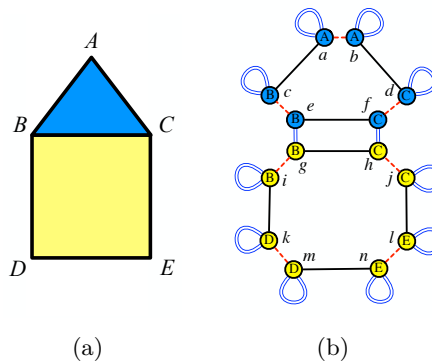


FIGURE 3.16 – Représentation d'un objet 2D avec des plongements.

Quelques exemples de plongement sont donnés sur la figure 3.16. Le plongement géométrique $\mathbf{point} : \langle \alpha_1, \alpha_2 \rangle \rightarrow \mathit{Point}$ définit la position de chaque sommet (A, B, C, D et E) dans un espace géométrique de dimension 2D/3D et est rattaché aux orbites $\langle \alpha_1, \alpha_2 \rangle$ correspondant aux sommets pour un objet 2D. Pour satisfaire la contrainte de cohérence de plongement, tous les brins du même sommet partagent la même position. Par exemple, sur la figure 3.16(b), les brins c, e, g et i partagent la même position B de type Point . De même, le plongement $\mathbf{color} : \langle \alpha_0, \alpha_1 \rangle \rightarrow \mathit{RGB}$ permet de définir la couleur des faces, sous la forme d'un triplet ($\mathit{rouge}, \mathit{vert}, \mathit{bleu}$) en l'associant aux orbites faces $\langle \alpha_0, \alpha_1 \rangle$. Tous les brins de la même face partagent la même couleur. Par exemple, sur la figure 3.16(b) les brins a, b, c, d et f appartiennent à la même face et partagent la même couleur, dans le cas présent le bleu.

Toute information que l'on souhaite attacher à une G-carte se traduit par un plongement, associé à un type d'orbites. Il convient alors de définir les orbites les plus adéquates pour porter l'information. Il peut s'agir des cellules ou d'orbites plus spécifiques comme, en 2D, les arêtes de face $\langle \alpha_0 \rangle$, les sommets de face $\langle \alpha_1 \rangle$, les extrémités d'arête $\langle \alpha_2 \rangle$, les composantes connexes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$, etc. En 3D, les définitions des orbites changent et de nouvelles orbites apparaissent : par exemple, les arêtes de faces correspondent aux orbites $\langle \alpha_0, \alpha_3 \rangle$ et les orbites $\langle \alpha_0 \rangle$ sont des arêtes de face de volume (*i.e.* les arêtes d'une seule face dans un seul volume). Néanmoins, certains éléments ne correspondent pas forcément à un type d'orbites. C'est le cas par exemple des diagonales de face. Dans ce cas, puisqu'on ne peut pas associer une information aux diagonales, il faut l'associer aux orbites qui englobent

ou qui permettent de retrouver les diagonales en question comme par exemple les orbites faces $\langle \alpha_0, \alpha_1 \rangle$.

Le choix d'un modèle topologique dépend du type d'objets à représenter dans l'application visée et de la nature des opérations à réaliser. On ne peut pas dire qu'il existe un modèle meilleur qu'un autre. Néanmoins, on peut dire qu'un modèle est plus adapté pour une classe d'objets ou bien pour une utilisation dans une application particulière. Nous avons choisi de travailler avec un modèleur cellulaire parce que, comme déjà mentionné dans la section précédente, nous voulons créer un cadre général pour simuler différents types d'objets avec différents types de maillages, y compris des maillages hétérogènes. La famille des cartes combinatoires est bien adaptée. Parmi les cartes combinatoires, nous avons choisi les G-cartes, tout d'abord, parce que c'est un modèle général qui permet de représenter une large classe d'objets et qu'il est défini mathématiquement avec des contraintes de cohérence. Mais ce choix est principalement motivé par l'ensemble des orbites utilisables, car elles sont plus atomiques avec les cartes généralisées qu'avec les cartes orientées. Par exemple, dans une 2-G-carte, on peut distinguer une arête de face (orbite de type $\langle \alpha_0 \rangle$), d'une extrémité d'arête (orbite de type $\langle \alpha_2 \rangle$). Au contraire, dans une carte orientée, l'orbite brin désigne à la fois les arêtes de faces et les extrémités d'arêtes. Il n'est donc pas possible de différencier les plongements associés aux arêtes des faces de ceux associés aux extrémités d'arêtes. De même, les opérations qui s'appliquent aux arêtes de faces sont indistinguables de celles qui s'appliquent aux extrémités d'arêtes. Surtout, il n'existe pas d'orbite correspondant aux extrémités des arêtes de faces. Donc pour lui associer une information, il convient d'utiliser deux plongements, le premier qui porte l'information de l'origine de l'arête de face, et l'autre qui porte celle de l'autre extrémité. Bien entendu, dans ce cas la cohérence des plongements ne peut plus être assurée formellement. Les G-cartes nous offrent ainsi la gamme d'orbites la plus vaste, d'où notre choix, mais le passage au cartes orientées pourra se faire, au prix de compromis et/ou de conventions. Finalement, une autre raison pour laquelle nous utilisons les G-cartes est l'utilisation d'un langage à base des règles qui est basé sur les G-cartes. L'homogénéité de ce modèle simplifie la définition des opérations avec les règles. Dans la partie suivante, nous présentons ce langage et la manière de l'utiliser pour créer des opérations de manipulation des objets.

3.4 Modélisation à base de règles : Jerboa

Les langages à base de règles ont été utilisés pour la modélisation géométrique des objets et l'animation. Parmi les langages les plus connus se trouvent les *L-systèmes* [PL90]. Ils ont été introduits pour modéliser le développement et la croissance des plantes. Un exemple de modélisation d'une plante est donné sur la figure 3.17. Les langages utilisés en informatique graphique ne sont pas limités à la modélisation de plante. Ils ont été utilisés dans plusieurs autres domaines, comme



FIGURE 3.17 – Modélisation d’une plante avec un L-system.

par exemple, la reconstruction de bâtiments [VAB10]. Ils ont également été utilisés dans un contexte topologique dans [TGM⁺09] pour modéliser la structure interne du bois, la croissance des feuilles et pour modéliser la structure interne des fruits [BTG15]... Quelle que soit l’application exploitant les L-systèmes, un ensemble limité d’opérations spécialisées est défini et codé de manière *ad hoc*. Ces dernières sont appliquées au moyen de certains types de règles et grammaires. Ainsi, chaque domaine d’application utilise un dialecte de L-système différent.

Au contraire, Jerboa définit les opérations topologiques et géométriques sous la forme de transformations de graphes qui modifient directement la structure topologique des objets et leurs plongements. Il est générique en dimension et en plongement et reste indépendant du domaine d’application. Comme un de nos objectifs est de proposer une plate-forme générale pour la simulation physique, l’utilisation d’un langage à base de règles paraît intéressant pour un tel travail exploratoire. En effet, ces langages permettent le prototypage rapide et facile des opérations. Outre sa généricité, le principal avantage de Jerboa est la vérification syntaxique des règles lors de leur création. Cette vérification permet de garantir la préservation de contraintes de cohérence topologique et de plongement des objets lors de leur transformation (*i.e.* lors de l’application des règles). Ainsi, des objets construits et modifiés uniquement par des règles sont garantis cohérents.

3.4.1 Le langage Jerboa

Une règle Jerboa se présente sous la forme suivante : $L \rightarrow R$. L est le membre gauche de la règle et correspond au motif à filtrer (c’est à dire le sous-graphe à reconnaître dans l’objet). R est le membre droit, il correspond au motif transformé (c’est-à-dire le sous-graphe qui remplace L dans l’objet). Les règles de transformation de graphe sont définies par des graphes qui décrivent l’objet avant et après une modification. Un exemple de règle est donné sur la figure 3.18. Cette règle permet d’insérer un sommet sur une arête du bord (car comme on peut le remarquer sur la règle, les liaisons α_2 sont libres).

L’application de la règle présentée dans la partie haute de la figure 3.18 se fait en associant les nœuds de partie gauche (L) de la règle sur les nœuds du graphe

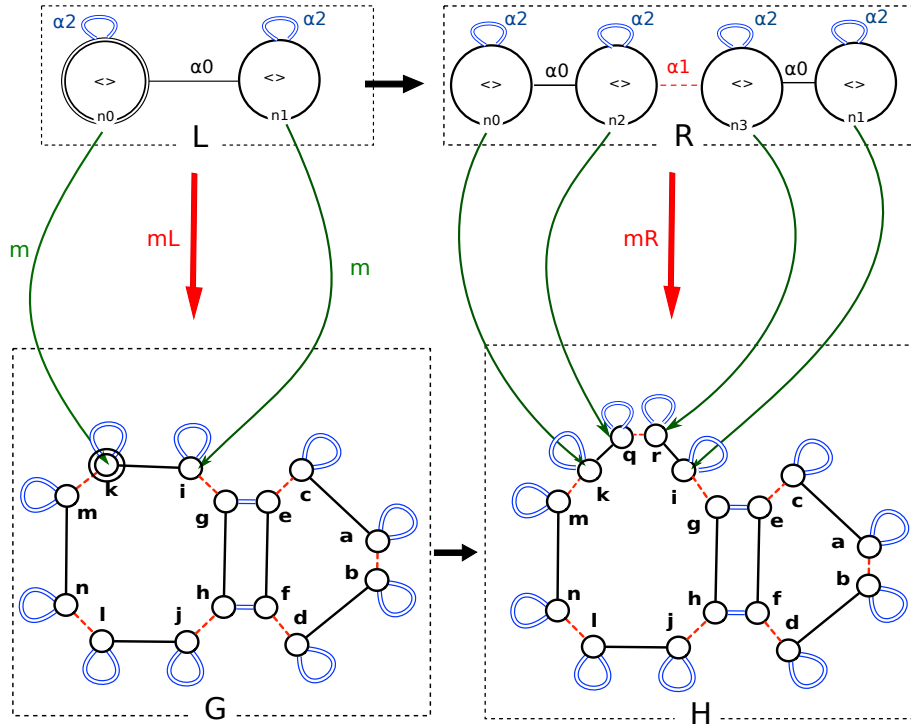


FIGURE 3.18 – Exemple d'application d'une règle de transformation de graphe.

G. Cette association se fait à l'aide d'un morphisme de filtrage³. Par exemple ici, dans le membre gauche, le nœud n_0 est associé au brin k du graphe G et le nœud n_1 au brin i . Dans le cas où il existe un morphisme de filtrage, la règle s'applique en remplaçant alors le motif filtré dans L par le motif décrit dans le membre droit qui sert ici à insérer un sommet sur une arête du bord, et pour cela il ajoute deux nouveaux nœuds (n_2 et n_3) qui sont associés respectivement aux brins q et r du graphe H (graphe transformé) par le morphisme de filtrage mR . S'il n'existe pas de morphisme de filtrage mL alors le motif L n'est pas reconnu dans le graphe G (le filtrage échoue) la règle ne s'applique pas et ainsi aucune transformation n'est appliquée.

Il faut noter que pour appliquer une règle, il convient à préciser son lieu d'application en construisant le morphisme de filtrage. Pour cela, dans le membre de gauche de la règle, on marque par un double cercle les nœuds à fournir. Ces nœuds s'appellent **les nœuds d'accroche** (*hook*) et sont les paramètres topologiques de la règle. Lors de l'application d'une règle Jerboa, chaque nœud d'accroche est associé à un brin de l'objet. Si chaque composante connexe du graphe gauche de la

3. Un morphisme $m : L \rightarrow G$ est une fonction qui transporte les brins et les liaisons en préservant la structure du graphe et son étiquetage. Pour toute liaison e de L , $s(m(e)) = m(s(e))$, $t(m(e)) = m(t(e))$ et $\alpha(e) = \alpha(m(e))$ et pour tout brin v de L et tout plongement π , $\pi(e) = \pi(m(e))$. m est un morphisme de filtrage s'il est injectif. C'est à dire, pour toutes liaisons e et e' de L , $m(e) = m(e')$ si et seulement si $e = e'$ et pour tout brin v et v' de L , $m(v) = m(v')$ si et seulement si $v = v'$.

règle possède un nœud d'accroche, alors le morphisme (quand il existe) est unique. Et la règle peut ainsi être appliquée sans ambiguïté. Sur la figure 3.18, le nœud d'accroche n_0 est associé au brin k , et le morphisme de filtrage est représenté par les flèches vertes. Notons que si l'on essaye d'appliquer cette même règle sur l'arête centrale, par exemple en associant le nœud n_0 au brin g , alors le filtrage échouera, car le brin g n'a pas de boucle α_2 contrairement au nœud n_0 .

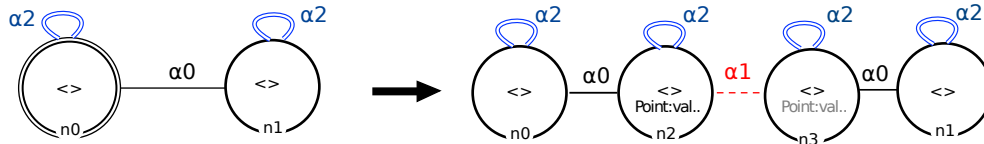


FIGURE 3.19 – Ajout de plongement pour calculer la position du sommet à insérer.

En complément de la transformation topologique, les règles permettent de transformer les plongements. Par exemple, dans la figure 3.19, la position du nouveau sommet est définie par le plongement point par une expression.

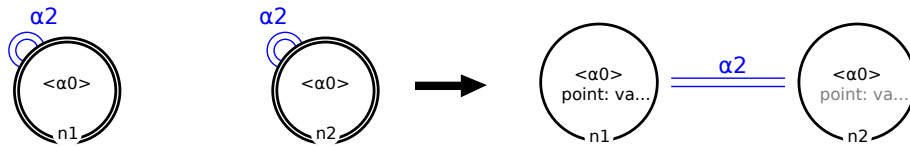


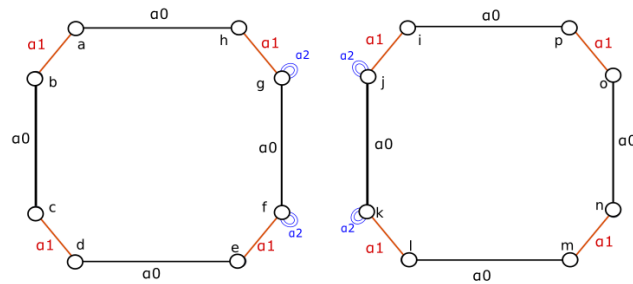
FIGURE 3.20 – Règle de couture de deux faces le long d'une arête.

Comme indiqué précédemment, il existe des règles qui contiennent plus qu'un nœud d'accroche dans le membre gauche comme par exemple dans la règle de couture présentée sur la figure 3.20 qui s'applique sur deux arêtes et permet de coudre deux faces le long d'une arête. Aussi, la règle contient deux nœuds d'accroche, car les deux arêtes sont indépendantes. D'autre part, chaque nœud est ici étiqueté par les orbites $\langle \alpha_0 \rangle$, chaque nœud filtre donc une orbite $\langle \alpha_0 \rangle$ complète. Ainsi le nœud n_1 filtre les brins g et f , et n_2 filtre les brins j et k .

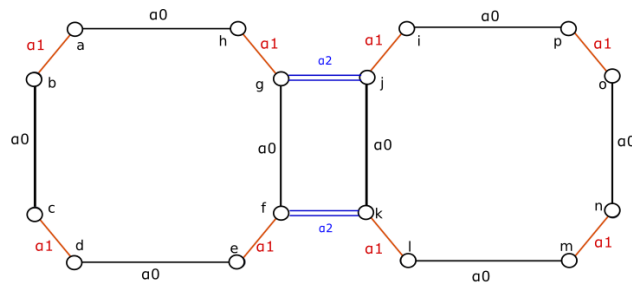
Nous détaillons dans la suite les principes utilisables dans Jerboa pour définir une règle, d'abord dédiée aux changements topologiques, et par la suite, dédiée aux changements de plongement. Nous donnons une brève description de l'outil Jerboa, de son éditeur et quelques exemples de règles.

3.4.1.1 Transformations topologiques

Pour définir des opérations pour différents types et formes d'orbites, le langage à base de règles est doté de *variables topologiques* [PACL08]. Pour cela, les nœuds sont étiquetés par des orbites, ce qui permet de désigner, en un coup, tous les brins de l'orbite en question, lors de l'instanciation de la règle.



(a) Avant l'application de la règle



(b) Après l'application de la règle

FIGURE 3.21 – Exemple d'application de la règle de couture de deux faces.

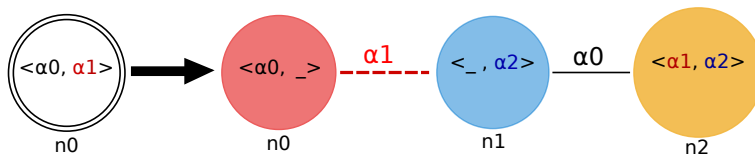


FIGURE 3.22 – Règle de triangulation d'une face.

Par exemple, la triangulation d'une face peut être définie par la règle présentée sur la figure 3.22. Dans le membre gauche, un seul nœud nommé n_0 définit toute la face à filtrer. Il est étiqueté avec l'orbite $\langle \alpha_0, \alpha_1 \rangle$, qui définit les faces en 2D.

L'application de cette règle sur le brin g de la G-carte présentée sur la figure 3.23(b) instancie la règle pour une face quadrangulaire et fournit le résultat de la figure 3.23(c). De même, son application sur le brin a instancie la règle pour une face triangulaire (voir le résultat sur la figure 3.23(d)). Ainsi la règle est instanciée avant d'être appliquée. Pour cela, sa variable topologique n_0 (le nœud d'accroche) est substituée par l'orbite d'application, $\langle \alpha_0, \alpha_1 \rangle(g)$ ou $\langle \alpha_0, \alpha_1 \rangle(a)$ dans notre exemple. Les autres nœuds de la règle sont instanciés par la même orbite, mais à renommage et/ou suppression des liaisons près.

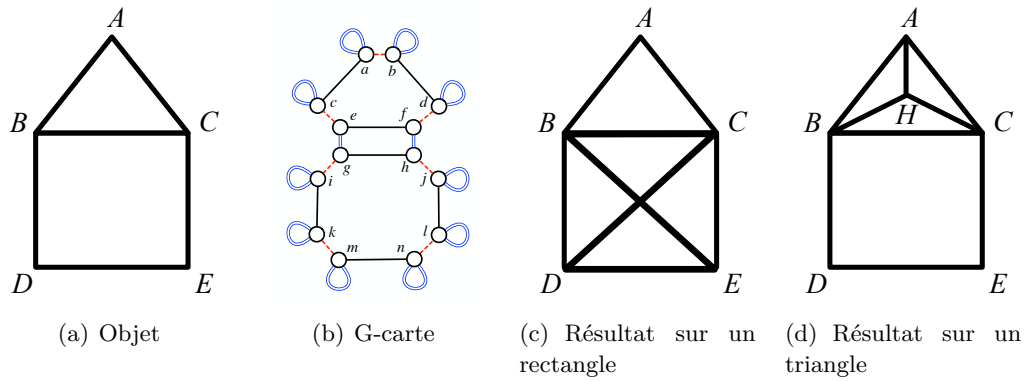


FIGURE 3.23 – Application de la triangulation d’une face.

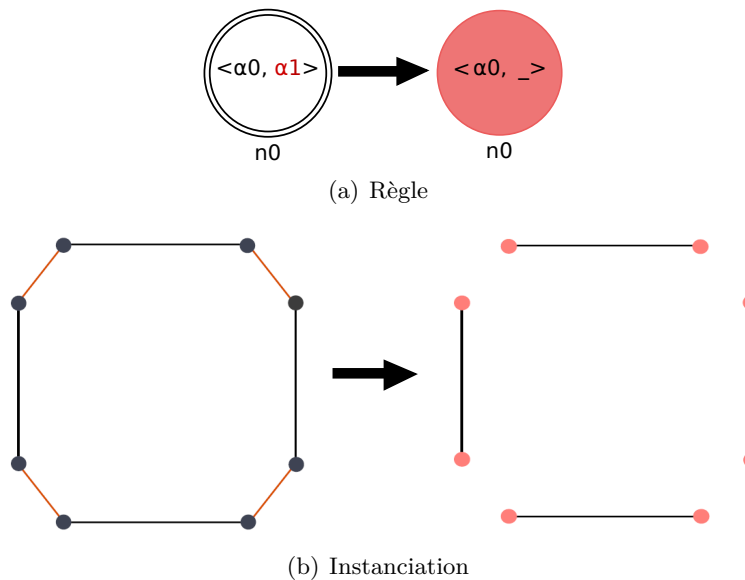


FIGURE 3.24 – Instanciation du nœud n_0 de la règle de triangulation d’une face.

Commençons tout d’abord par le nœud n_0 (en magenta) du membre droit de la règle. Il est étiqueté par $\langle \alpha_0, - \rangle$, ce qui signifie que les liaisons α_0 de la face filtrée sont préservées alors que les liaisons α_1 sont supprimées (ce qui est noté par le caractère « $_$ »). La sous-règle correspondante (figure 3.24(a)) a donc pour instanciation la règle de transformation de la figure 3.24(b). Son application a ainsi pour effet de découdre les arêtes de la face filtrée.

De la même façon, les liaisons peuvent être renommées. En se référant à la même règle de triangulation, le nœud n_2 (nœud jaune) est étiqueté par l’orbite $\langle \alpha_1, \alpha_2 \rangle$. Cela indique que les liaisons α_0 (resp. α_1) de l’orbite d’instanciation sont renommées en α_1 (resp. α_2). Ce qui permet de construire le sommet dual (constitué de brins jaunes) de la face filtrée au départ comme montré par la figure 3.25. Remarquons que lorsqu’un même nœud (comme n_0) apparaît à gauche et à droite de la règle,

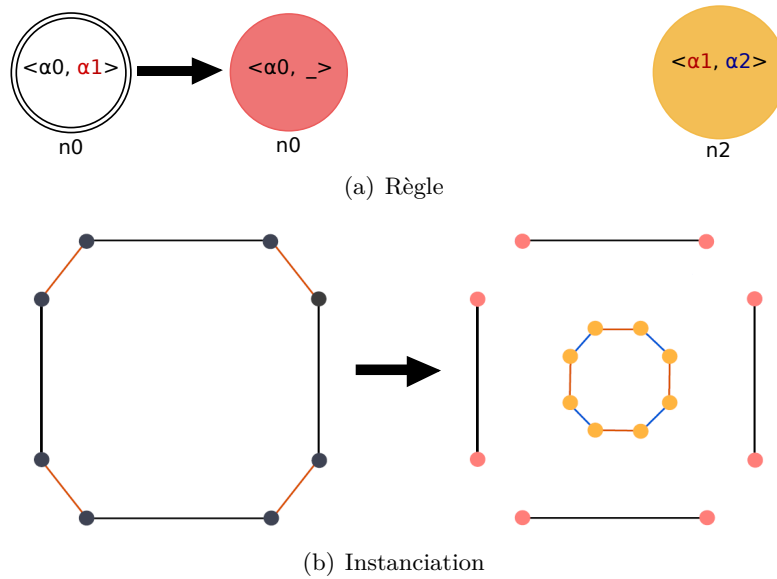


FIGURE 3.25 – Instanciation du nœud n_2 de la règle de triangulation d'une face.

les brins filtrés sont préservés lors de la transformation. Quand un nœud n'apparaît qu'à droite de la règle (comme n_2), alors la transformation ajoute de nouveaux nœuds. Et au contraire lorsque un nœud n'apparaît qu'à gauche de la règle, la transformation supprime des brins.

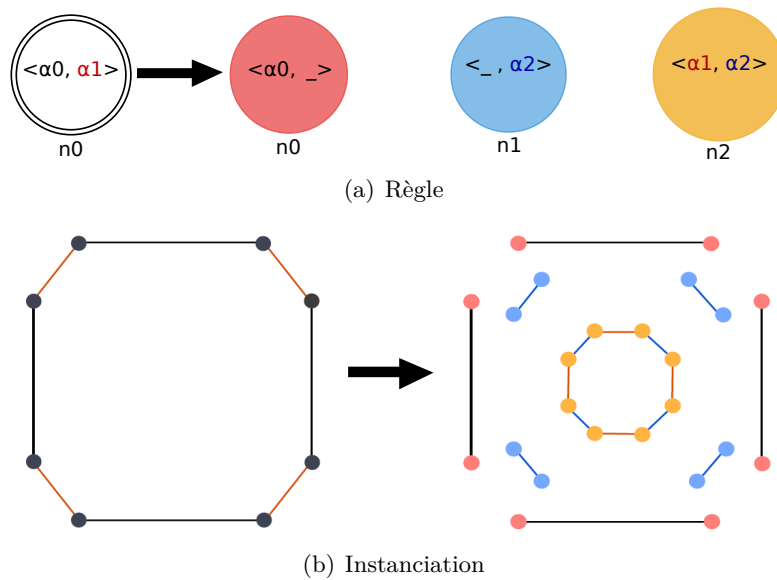


FIGURE 3.26 – Instanciation du nœud n_1 de la règle de triangulation d'une face.

Un nœud peut présenter à la fois des liaisons supprimés et d'autre renommées. C'est le cas pour le nœud n_1 (nœud bleu) qui est étiqueté par $\langle _ , \alpha_2 \rangle$, les liaisons α_0 de l'orbite filtrée sont donc supprimées et les liaisons α_1 sont renommées en α_2

comme montré par la figure 3.26.

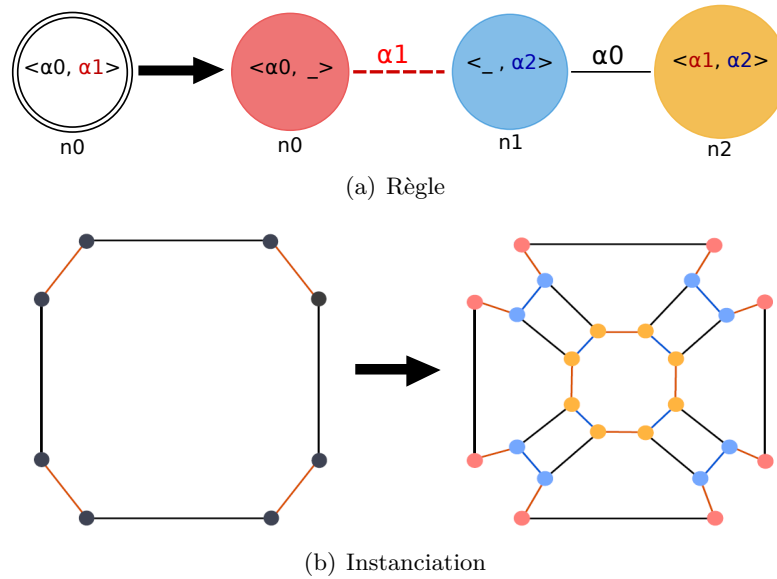


FIGURE 3.27 – Ajout des liaisons explicites entre les instances.

Enfin, les différents nœuds sont reliés par des liaisons dites *explicites*, lors de l'instanciation, ces liaisons sont dupliquées pour relier deux à deux chaque brins instance des nœuds. Par exemple et comme montré sur la figure 3.27, la liaison α_0 lie deux à deux les brins instances du nœud n_2 (en jaune) avec les brins instances du nœud n_1 (en bleu). De même, les brins instance de n_0 sont liés avec ceux de n_1 par les liaisons α_1 . Finalement, la règle instanciée peut être appliquée sur la G-carte 3.28(a), pour obtenir la G-carte transformée (figure 3.28(b)).

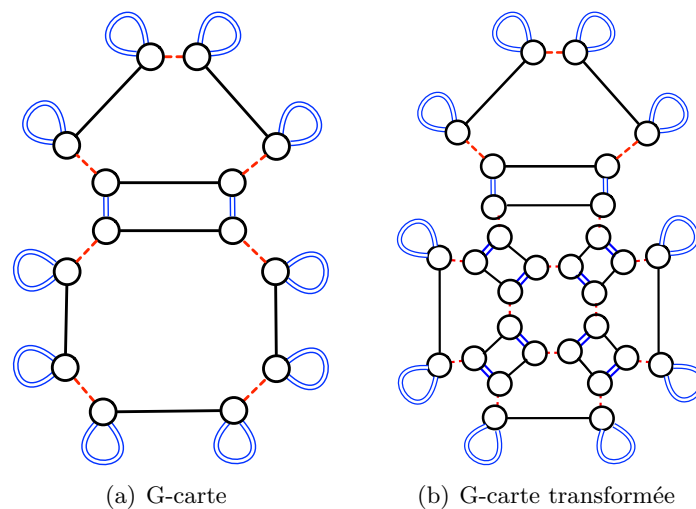


FIGURE 3.28 – Face rectangulaire triangulée.

L'éditeur de Jerboa vérifie automatiquement la préservation de la cohérence

topologique de l'objet lors de la création de la règle. En effet, dans [Pou09] des conditions syntaxiques sont définies pour garantir cette préservation. Par exemple, pour garantir la contrainte d'arcs incidents, tous les nœuds ajoutés au membre droit doivent posséder toutes les liaisons de α_0 jusqu'à α_n . En particulier, le nœud **n2** qui est ajouté à la règle 3.27 dispose bien de ses trois liaisons α_0 , α_1 et α_2 . La liaison α_0 est explicite entre les nœuds **n1** et **n2**. Les liaisons α_1 et α_2 sont implicite et portées par le nœud lui-même. Les autres contraintes de cohérence topologique (contrainte de cycle et celle de non orientation) sont également vérifiées.

3.4.1.2 Transformation des plongements

Comme expliqué précédemment, il faut ajouter des plongements à la structure topologique pour définir l'objet avec sa géométrie et ses propriétés physiques. Pour cela, le langage de règles permet de calculer les plongements pour les objets transformés [Bel12]. Rappelons que chaque plongement est porté par un type bien défini d'orbites. Nous poursuivons cette partie avec la même règle de triangulation précédente complétée par la transformation des plongements (voir figure 3.29). Le nœud **n2** du membre droit porte le plongement **point** qui permet de calculer les coordonnées du barycentre associé au nouveau sommet. Les trois nœuds du membre droit portent le plongement **color** qui colorie les nouvelles faces créées avec un mélange de la couleur de la face filtrée avec celle de la face voisine. Les nœuds **n0** et **n1** ne portent pas de plongement **point**, car les points des sommets filtrés restent inchangés. Ainsi, la règle complétée transforme l'objet comme montré par la figure 3.30 quand on l'applique sur le carré bleu.

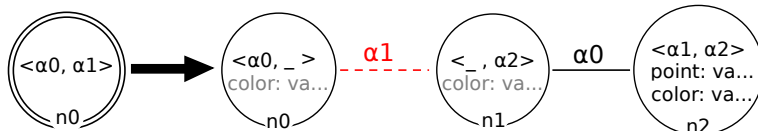


FIGURE 3.29 – Règle de triangulation et coloration d'une face.

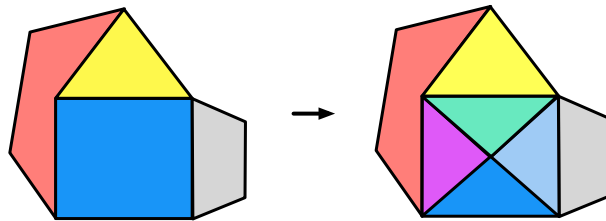


FIGURE 3.30 – Exemple d'application de la règle 3.29.

Cependant, la version courante de Jerboa utilisée dans cette thèse n'offre pas de langage pour spécifier la transformation des plongements. Cette dernière est donc écrite directement dans le langage cible, pour manipuler et calculer ces plongements, Jerboa offre différentes méthodes. Par exemple, pour calculer la couleur de chaque

nouvelle face triangulaire, nous faisons la moyenne entre la couleur de la face initiale et la face voisine comme on peut le voir dans l'application de la règle sur la figure 3.30. Tout d'abord, pour accéder au plongement d'un nœud filtré, ou plus exactement au plongement de ses instances (qui sont des brins de l'objet filtré), Jerboa utilise la méthode `ebd` qui est générique en fonction du plongement `T`. Cette méthode a comme prototype `<T> T ebd(String name)` (avec `T` le type du plongement et `name` le nom du plongement). Ainsi, par exemple, pour accéder à la couleur de la face initiale, il faut utiliser l'expression : `n0().<Color> ebd("color")` qui retourne le plongement `color` des brins filtrés par le nœuds `n0` qui est de type `Color`. De même pour `n0().<Point> ebd("point")`, permet d'accéder au plongement `point` du brin courant instance de `n0`.

Une autre méthode, très utilisée, existant dans la bibliothèque Jerboa est la méthode `JerboaNode alpha(int i)` qui permet d'accéder au brin voisin selon la liaison α_i . Par exemple, pour accéder à la face voisine du brin courant, nous utilisons l'expression `n0.alpha(2)`, et l'expression `n0.alpha(2).<Color> ebd("color")` permet alors d'accéder à la couleur de la face voisine.

La bibliothèque Jerboa possède d'autres méthodes comme la `collecte`, qui permet de construire la collection des brins d'une orbite donnée. Ces différentes méthodes sont utilisées pour construire les expressions de plongement, qui sont des portions de code écrit en Java ou C++. En complément des méthodes de Jerboa, les expressions de plongements utilisent les méthodes statiques fournies par les classes de plongement. Par exemple, dans la règle de triangulation 3.29, pour calculer la moyenne des couleurs des faces, nous utilisons la méthode `middle` définie dans la classe de plongement `color` qui calcule la moyenne de deux couleurs données. L'expression du plongement portée par le nœud `n2` de la règle s'écrit alors de la façon suivant : `value = Color.middle(n0().<Color>ebd("color"), n0().alpha(2).<Color>ebd("color"));`

Pour éviter de saisir plusieurs fois la même expression de plongement lors de l'écriture d'une règle, Jerboa propage automatiquement l'expression définie pour un nœud sur les nœuds de la même orbite support du plongement. Par exemple, dans la règle présentée sur la figure 3.29, comme `n0`, `n1` et `n2` appartiennent à la même orbite face $\langle \alpha_0, \alpha_1 \rangle$, le plongement couleur d'une face est défini pour le nœud `n2` (l'expression est noire) et propagé aux nœuds `n0` et `n1` (l'expression est alors grise).

Pour récapituler, les règles créées par Jerboa diffèrent grandement par leur objectifs. Il existe des règles qui apportent une modification de la structure (par exemple l'ajout d'un sommet et la couture de deux faces), on parle alors de *règle topologique*. Dans ces règles, on peut ajouter/supprimer des brins et/ou modifier/ajouter des liaisons. Mais il faut noter qu'il existe des règles qui permettent de modifier uniquement des plongements sans apporter de modification à la structure topologique comme par exemple l'ajout d'une couleur à une face et la modification de la position d'un sommet. La règle présentée sur la figure 3.31 permet de changer uniquement la forme des quadrilatères, car le membre gauche de la règle filtre explicitement un quadrilatère, en fournissant une nouvelle position à un des sommets

(sommet du coin supérieur gauche dans notre exemple). Un exemple d'application est donnée sur la même figure. Nous utilisons beaucoup ce type de règles dans notre travail, plusieurs autres exemples des règles qui ne modifient que les plongements sans modification topologique sont alors données dans les chapitres 5 et 6

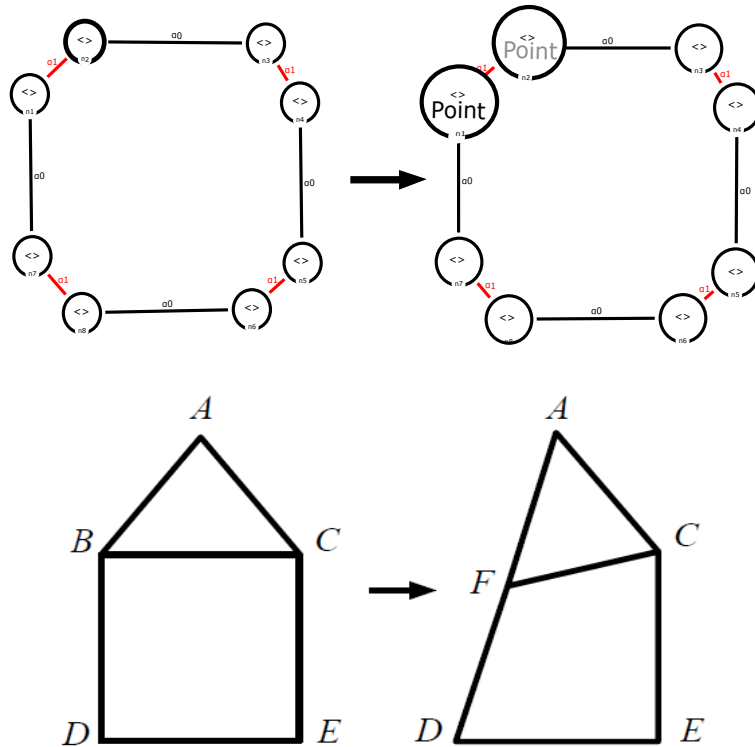


FIGURE 3.31 – Règle de changement de position d'un sommet d'un quadrilatère et son application.

Souvent, nous combinons à la fois des changements d'un plongement et de la topologie. Il est également possible de procéder à des changements simultanés de plongements différents. Par exemple, plutôt que de définir une règle comme celle présentée sur la figure 3.31 pour modifier la position d'un sommet et une autre pour modifier la couleur de la face quadrilatère, il est possible de combiner les deux, comme montré sur la figure 3.32, en définissant les deux plongements simultanément dans la même règle.

3.4.1.3 L'outil Jerboa

Jerboa est une bibliothèque qui permet de générer des modeleurs. Comme montré sur la figure 3.33, cet outil est composé : d'un *éditeur de modeleur* qui permet de générer le noyau de modeleur dédié correspondant, d'un noyau qui comprend le moteur d'application des règles et permet ainsi d'exécuter toutes les opérations spécifiées par les règles, et d'un visualiseur qui permet de réaliser facilement l'interface graphique des noyaux des modeleurs générés. Comme illustré sur la figure 3.33,

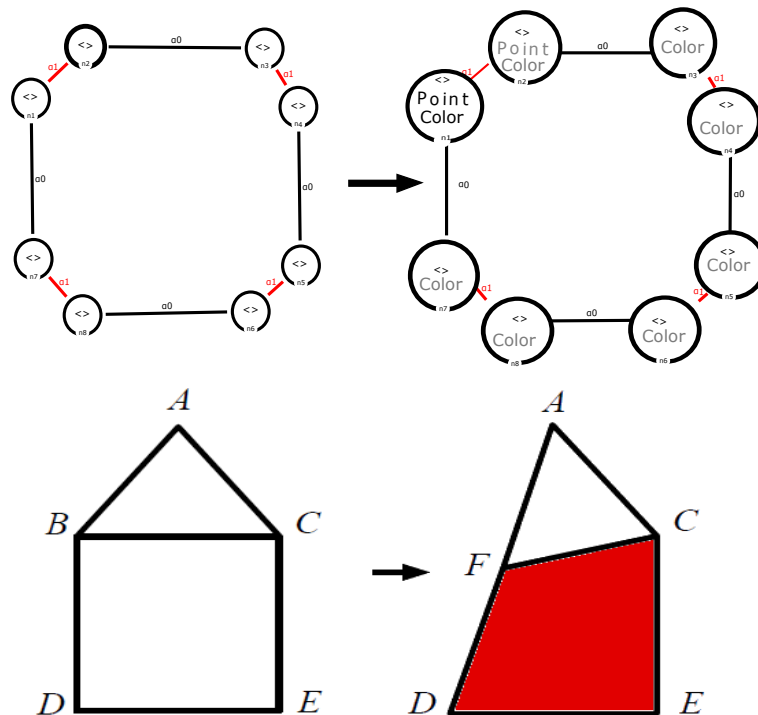


FIGURE 3.32 – Règle de changement de la forme et du couleur d'un quadrilatère et son application.

un utilisateur Jerboa est spécialiste de modélisation géométrique et développe, avec cet outil, un modeler dédié. Il utilise l'éditeur de Jerboa, crée les classes de plongements et les règles qu'il faut pour son modeler. C'est ce type d'utilisation que nous avons eu durant notre travail. L'utilisateur final de l'application est spécialiste du domaine cible, mais n'a pas besoin de connaître Jerboa, car il utilise directement le modeler dédié via son interface graphique.

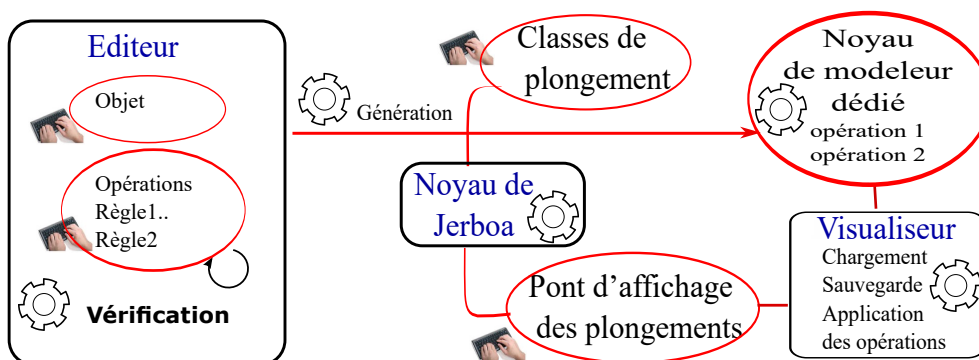


FIGURE 3.33 – Schéma d'utilisation de Jerboa.

Un premier prototype de Jerboa a été écrit par Thomas Bellet [BPA⁺10]. En-

suite en 2014, une version plus avancée, écrite en JAVA, a été introduite par Belhaouari *et al.* [BALB14]. Dans cette thèse, nous utilisons la version écrite en JAVA. De plus, il existe une autre version de Jerboa écrite en C++, qui est en cours de développement, introduite par Gauthier *et al.* [GBA15].

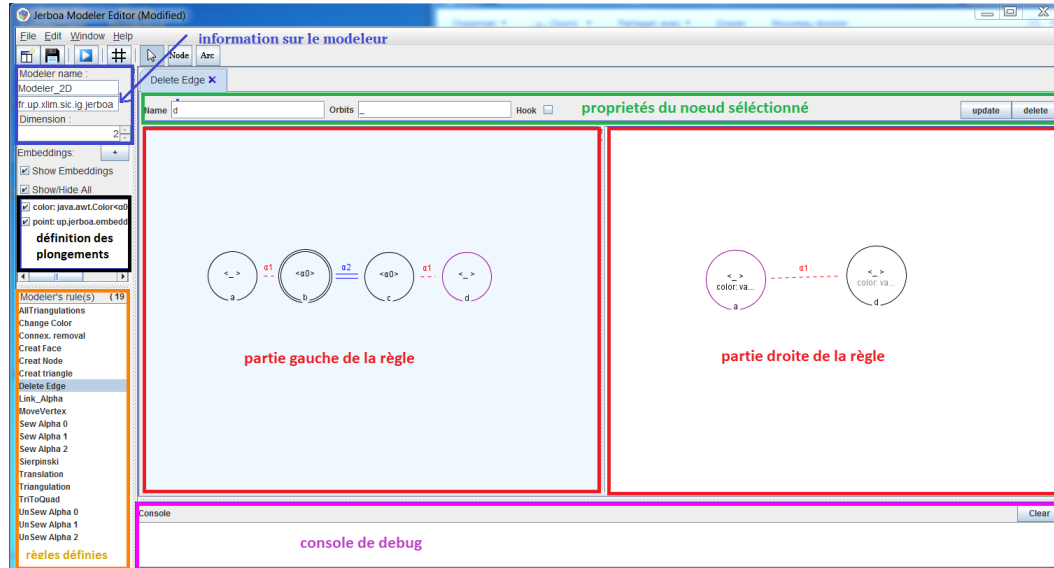


FIGURE 3.34 – Interface de l'éditeur de Modeleur de Jerboa.

Présentons en premier lieu l'éditeur de Jerboa. Pour créer un modeleur, il faut tout d'abord lui donner un nom et définir sa dimension topologique dans la partie liée aux informations sur le modeleur comme montré sur la figure 3.34 par le cadre bleu. L'étape suivante consiste à créer les plongements nécessaires et à les associer à des orbites de la structure topologique dans la zone appelée définition de plongements (voir cadre noir figure 3.34).

Après cette étape, l'utilisateur peut créer les opérations de son modeleur, sous la forme de règles (cadres rouges sur la figure 3.34). À la création d'une opération, l'interface de la figure 3.35 doit être renseignée. Outre le nom de l'opération et un commentaire, il est possible d'ajouter une pré-condition qui réduit les cas d'application de la règle. Par exemple, nous pouvons préciser que la règle de triangulation ne s'applique que sur les faces rouges ou que sur les faces à quatre cotés ou plus. Il est aussi possible d'ajouter à la règle des *extra-parameters* (pré-traitement), qui sont les paramètres de plongement de la règle, comme par exemple le vecteur de translation pour une règle de translation d'une composante connexe. Mais ce champ peut aussi permettre d'injecter du code qui est exécuté entre la phase de filtrage du membre gauche celle de transformation du membre droit. Nous l'utiliserons donc régulièrement pour effectuer des calculs communs aux différents plongements et ainsi optimiser l'exécution des transformations des plongements. Plusieurs exemples seront donnés ultérieurement dans le chapitre 6.

L'éditeur de Jerboa permet de vérifier automatiquement, lors de la création des

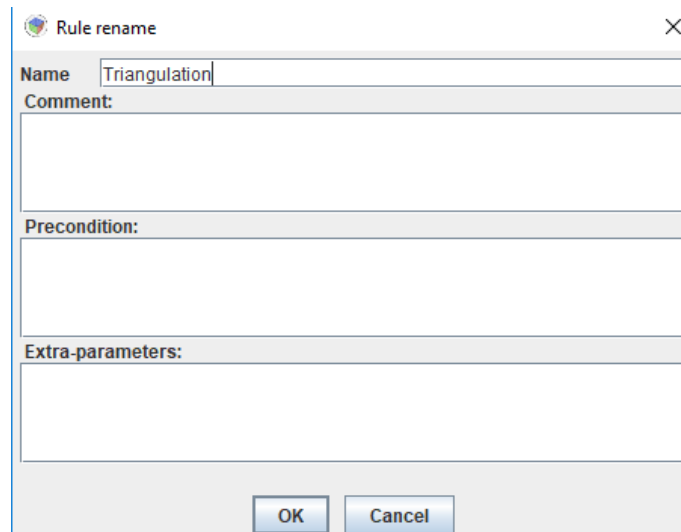


FIGURE 3.35 – Interface de modification d’une règle.

règles, la syntaxe graphique et les conditions de préservation de la cohérence topologique. Certaines vérifications sont aussi effectuées sur les expressions de plongements qui sont ensuite créés de manière cohérente par le moteur d’application des règles. S’il y a des erreurs, elles sont affichées dans la console (cadre en magenta figure 3.34). Finalement le moteur de règles de Jerboa assure que les règles préservent la cohérence des objets. Il s’agit d’une aide précieuse pour le développeur qui accélère grandement la mise au point des modeleurs dédiés.

3.4.2 Exemples de règles : création d’une surface en damier

Nous introduisons dans cette partie, un exemple d’utilisation du langage à base de règles pour la création d’une surface rectangulaire découpée en damier et quelques règles à appliquer pour faire des modifications topologiques et/ou modification des plongements sur cette surface.

Pour créer cette surface, nous avons conçu quatre règles qui sont appliquées successivement. Il s’agit de (1) la création d’un sommet, (2) la création d’une arête, (3) la création d’une face et finalement (4) la subdivision de face. Après l’application successive des trois premières règles puis plusieurs itérations d’application de la dernière règle, nous obtenons le damier. Dans la suite nous présentons ces règles une par une.

- Règle de création d’un sommet

Tout d’abord nous construisons une règle pour créer un sommet, comme le montre la figure 3.36. Comme c’est une règle de création, le membre gauche de la règle est vide. Elle crée un sommet libre en α_0 , α_1 , α_2 et α_3 , il n’est donc constitué que d’un seul brin. Néanmoins, il faut introduire sa position initiale. Pour

cela, nous avons créé un plongement `point` de type vecteur 3D initialisé à une valeur donnée. Si l'application utilise d'autres plongements, il faut penser à les indiquer explicitement dans le nœud, afin de les initialiser (au besoin, en fixant une valeur par défaut si le plongement en question n'est pas encore calculable à ce stade de la construction). Cette règle peut aisément être étendue pour ajouter d'autres informations de plongements comme des propriétés physiques (masse, ressort...).

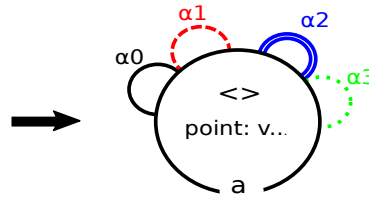


FIGURE 3.36 – Création d'un sommet.

- Règle de création d'une arête

À partir du sommet déjà créé, nous construisons une arête. Comme montré sur la figure 3.37, nous filtrons dans la partie gauche de la règle le sommet créé précédemment et nous ajoutons dans la partie droite de la règle un nouveau sommet. Pour créer une arête, qui a comme extrémités les deux sommets créés, nous ajoutons une liaison α_0 entre les deux. Après la construction de notre arête, nous définissons les valeurs des plongements associés au sommet nouvellement créé représenté par le nœud `b`.

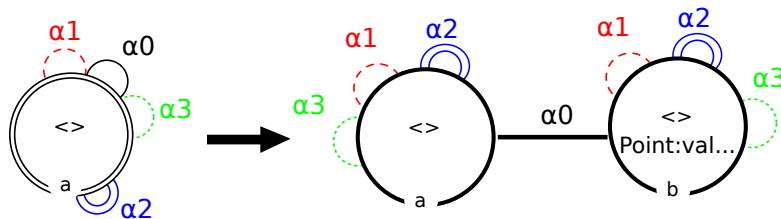


FIGURE 3.37 – Règle de création d'une arête.

Nous associons à ce sommet un nouveau plongement `point`. Là encore cette règle peut aisément être étendue à d'autres plongements.

- Règle de création d'une face

Nous proposons de créer une face rectangulaire grâce à l'extrusion de l'arête déjà créée par la règle précédente. Le principe est illustré sur la figure 3.38. À gauche, l'arête filtrée est l'arête précédente. À droite chaque brin filtré est dupliqué afin de construire successivement l'arête de départ (copie *a*), les arêtes de coté (copies *b* et *c*) et l'arête finale (copie *d*) qui est isomorphe à l'arête initiale.

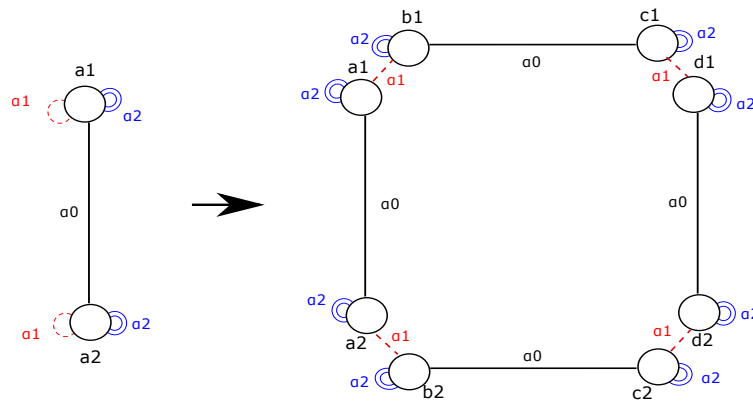


FIGURE 3.38 – Principe de création d'une face.

La règle de création est donnée sur la figure 3.39. Grâce à l'utilisation des variables topologiques, les brins de même lettre mais de numéros différents (figure 3.38) sont générés par un même nœud dans la règle (figure 3.39). Par exemple, les brins b_1 et b_2 sont créés par le nœud b de la règle. Comme les brins b_1 et b_2 ne sont pas liés entre eux (figure 3.38) le nœud b a sa liaison α_0 supprimée (figure 3.39).

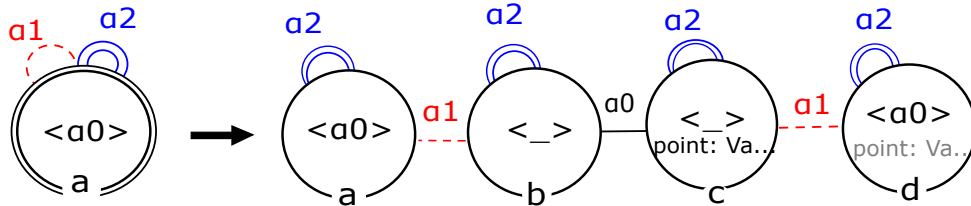


FIGURE 3.39 – Règle de création d'une face.

Dans la partie gauche de la règle, nous filtrons l'arête précédemment créée (le nœud a avec son orbite $\langle \alpha_0 \rangle$) et dans la partie droite, nous créons (de chaque côté)

une nouvelle arête (nœuds b et c , liés par α_0) qui lui est adjacente via une liaison explicite α_1 .

Le nœud c (et d) correspond aux nouveaux sommets (un de chaque côté), extrémités des nouvelles arêtes. Son plongement `point` est ici précisé. Il est défini simplement par translation des points d'origines. Le nœud d correspond à une duplication de l'arête initiale, où la liaison α_0 est préservée. Cette arête est liée aux sommets du nœud c par un lien α_1 . Puisque le plongement `point` des sommets est déjà précisé dans c , l'expression est propagée sur le nœud d .

Rappelons que s'il y a d'autres plongements, il faut ajouter les expressions des plongements adéquates pour chaque nœud afin de préserver la cohérence de l'objet construit. L'éditeur de Jerboa vérifie que tous les plongements sont effectivement définis.

La figure 3.40 présente le résultat après l'application successive de la règle de création d'un sommet, puis celle de création d'une arête et enfin la création d'une face.

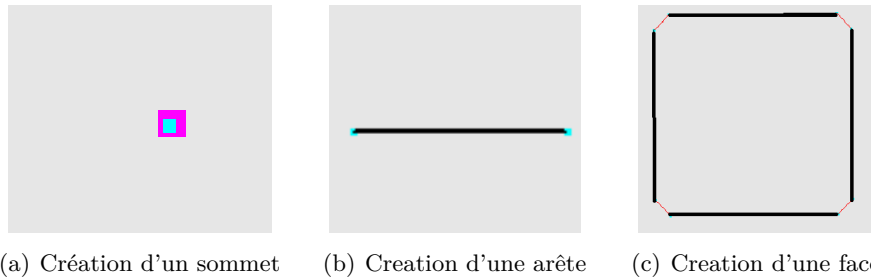


FIGURE 3.40 – Résultat de l'application de trois règles successives pour créer une face.

- Règle de subdivision d'une face

L'opération de subdivision consiste à diviser notre face rectangulaire en quatre sous-faces rectangulaires d'égales dimensions. Ceci revient à insérer un sommet au milieu de chaque arête et au centre de la face, puis à former les quatre nouvelles faces à partir de ces sommets. Il est possible de ré-appliquer cette opération sur les sous-faces de façon itérative, pour découper la surface initiale en damier, comme montré sur la figure 3.41.

La figure 3.42 illustre comment la subdivision peut être définie en exploitant les variables topologiques. Comme pour la règle de création de face, les brins de même lettre, mais de chiffres différents (par exemple tous les a_i) correspondent au même nœud dans la règle que nous allons définir. En pratique, pour chaque brin des sommets initiaux a_i , il faut insérer un nouveau sommet b_i au milieu de son arête incidente. Puis, il faut ajouter les arêtes qui relient les sommets nouvellement créés, à partir de c_i vers le centre de la face initiale d_i .

Remarquons que ce même schéma de subdivision peut être utilisé sur des faces non quadrilatère, comme le montre la figure 3.43

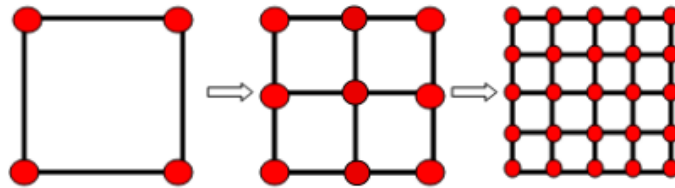


FIGURE 3.41 – Subdivisions successives d’une surface rectangulaire.

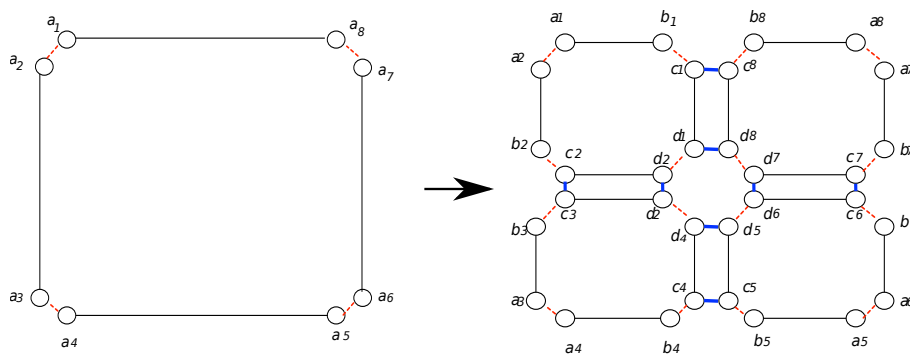


FIGURE 3.42 – Principe de la subdivision avec Jerboa.

La figure 3.44 illustre la règle de subdivision utilisée. Pour pouvoir appliquer le schéma de subdivision à toutes les faces (rectangulaires ou non) d’un objet, le membre de gauche filtre toute la composante connexe de l’objet 2D. Dans le membre de droit, nous supprimons tout d’abord les liaisons α_0 des faces initiales afin de découper les arêtes en deux. Ceci se traduit, dans le nœud **a** par une étiquette comprenant le symbole « $_$ » à la place de α_0 . Nous construisons ensuite le nœud **b** (*i.e.* tous les brins b_i) et on leur attribue comme valeur de plongement **point**, le milieu de l’arête initiale de la face. Ensuite il faut connecter ce nœud **b** au nœud **a** par des liaisons α_0 pour construire des arêtes (*i.e.* on construit des arêtes entre les paires de brins a_i et b_i). Puis nous ajoutons un autre nœud **c** qui est une copie

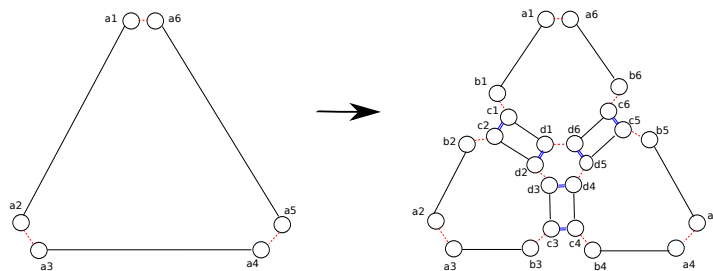


FIGURE 3.43 – Subdivision d’un triangle avec Jerboa.

du nœud b et qu'on relie à ce dernier par une liaison α_1 . Enfin, il faut construire le sommet central qui est le centre de la face initiale. Nous créons alors un nœud d qui a comme valeur de plongement `point`, les coordonnées du centre. Ce dernier nœud est lié au nœud c par une liaison α_0 . En exploitant le principe de renommage des liaisons dans l'étiquette du nœud c , les brins c_i sont liés entre eux par α_2 selon le schéma initial en α_0 des brins a_i . *Idem*, les brins d_i sont liés entre eux par α_2 et α_1 selon le schéma qui liait initialement les brins a_i entre eux par α_0 et α_1 .

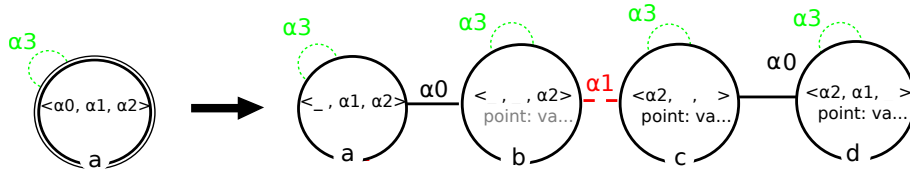


FIGURE 3.44 – Règle de subdivision des faces d'un objet connexe.

La figure 3.45 montre le résultat obtenu après une première, puis une seconde application de la règle. Il faut noter le fait que la règle s'applique à toute la composante connexe en un coup, c'est à dire sur toutes les faces en même temps. Nous ne sommes donc pas confrontés au problème du lien α_2 entre une arête de face qui a été subdivisée et une arête de face adjacente qui ne l'est pas encore (cas de la « jonction en T »).

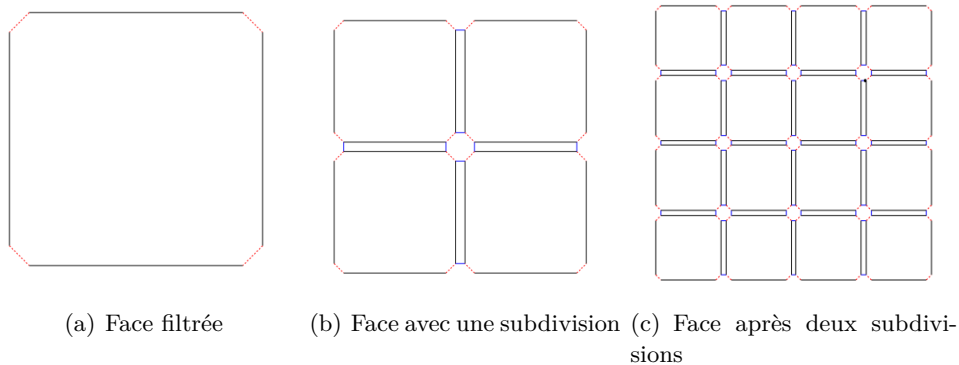


FIGURE 3.45 – Face subdivisée.

Pour pouvoir enchaîner les règles (trois règles de création de la surface, suivies d'un certain nombre de subdivisions), il est possible de les regrouper dans un script prenant la forme d'un programme Java. Notons que la prochaine version de Jerboa fournira un langage dédié pour définir plus facilement les scripts.

3.4.3 Autre exemples de règles

Dans cette partie, nous montrons d'autres exemples de règles dont nous nous sommes servis pour réaliser nos travaux. Nous avons montré, dans la section précédente, qu'il est possible de créer une face par extrusion. Néanmoins, il est également

possible de créer des faces en une seule règle comme montré sur la figure 3.46. Il s'agit d'une règle de création, le membre gauche est donc vide.

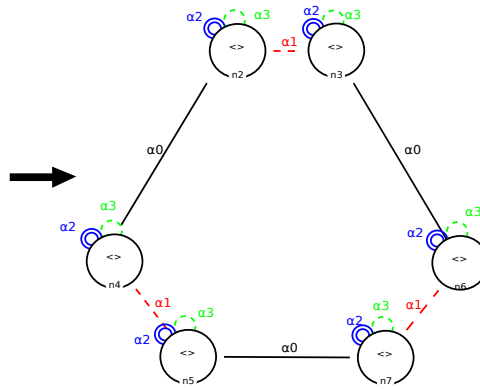


FIGURE 3.46 – Règle de création d'un triangle.

Au contraire, lorsque le membre droit de la règle est vide, alors il s'agit d'une règle de suppression. Par exemple, la règle de la figure 3.47 permet de supprimer une face isolée.

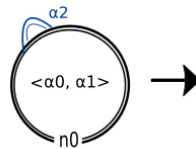


FIGURE 3.47 – Règle de suppression d'une face isolée

En effet, on reconnaît l'orbite de face $\langle \alpha_0, \alpha_1 \rangle$ en étiquette de nœud. En outre, la règle impose que la face soit libre en α_2 pour toutes ses arêtes, elle est donc obligatoirement isolée.

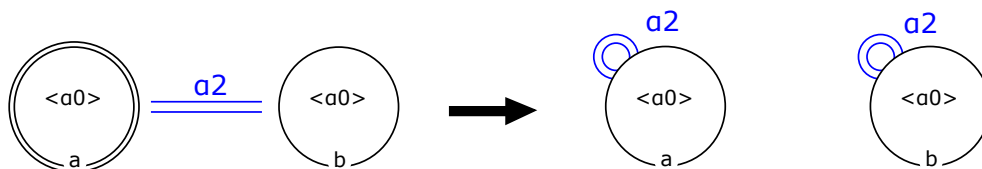


FIGURE 3.48 – Règle de découpe d'une arête

Pour isoler une face, il faut découper toutes ses arêtes. La règle présentée sur

la figure 3.48, permet la découverture d'une arête. En effet dans le membre gauche de la règle, nous avons filtré explicitement deux arêtes de face, représentées par les deux nœuds a et b liées par une liaison α_2 (*i.e* une arête commune à deux faces adjacentes). Alors que dans le membre droit, nous supprimons la liaison α_2 entre les deux arêtes de face pour obtenir deux arêtes libres en α_2 . Après l'application de cette règle sur toutes les arêtes d'une face, cette dernière devient isolée et par la suite, nous pouvons appliquer la règle de suppression d'une face isolée présentée sur la figure 3.47 comme le montre la figure 3.49.

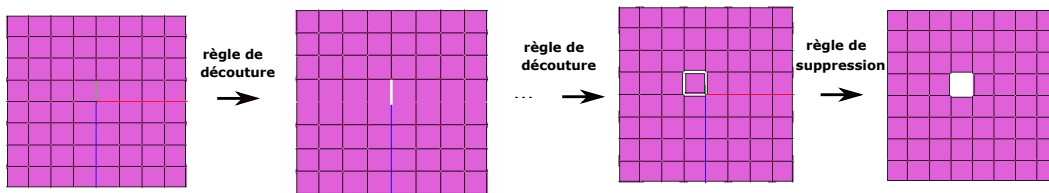


FIGURE 3.49 – Application successive de la règle d'isolement puis de la suppression d'une face.

Il faut noter que les deux règles 3.47 et 3.48 créées précédemment pour isoler une face et la supprimer, peuvent être réunies en une seule règle. Cette dernière est présentée sur la figure 3.50. Cette règle, comme on peut le remarquer, permet de filtrer dans le membre gauche une face, représentée par le nœud a dans la règle, avec toutes les arêtes des faces adjacentes (représentées par le nœud b dans la règle), alors que le membre droit permet de supprimer uniquement la face sélectionnée et de mettre à jour les liaisons α_2 des faces voisines.

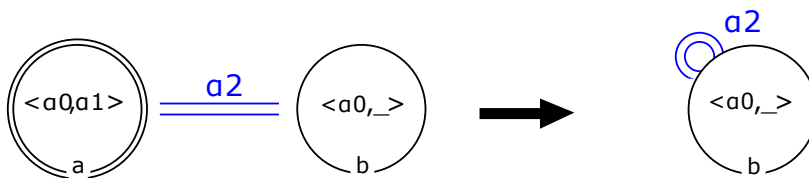


FIGURE 3.50 – Règle d'isolement et suppression d'une face

Avec Jerboa, il est également possible de mettre à jour un même plongement, pour toutes les orbites qui le portent. Par exemple, on peut demander de mettre à jour la position de tous les sommets d'une composante connexe, pour, par exemple, effectuer un déplacement ou une déformation d'un objet connexe. Ce type de règle prend toujours la même forme. Les deux motifs de la règle ne contiennent qu'un seul nœud qui est étiqueté par l'orbite composante connexe, à savoir l'orbite $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$ en 2D et $\langle \alpha_0, \alpha_1, \alpha_2, \alpha_3 \rangle$ en 3D. Le nœud du membre droit contient le plongement à mettre à jour. Ce type de règle permet de gagner en temps de calcul car tous les plongements sont recalculés en une seule application de règle. Par exemple, la règle présentée sur la figure 3.51, permet de mettre à jour les positions de tous les

sommets d'une composante connexe en 3D.

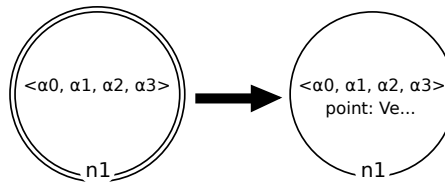


FIGURE 3.51 – Règle de mise à jour des positions des sommets dans toute une composante connexe 3D.

Un exemple d'application de cette règle, dans le cas particulier d'une rotation, sur la surface en damier précédemment créée est donné sur la figure 3.52

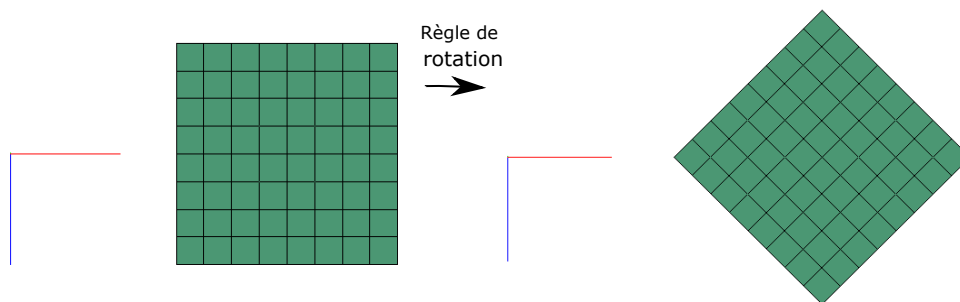


FIGURE 3.52 – Exemple d'application de règle de rotation de toute la composante connexe.

Pour obtenir le même résultat, il est possible d'écrire une règle qui filtre un brin et met à jour le plongement voulu. On applique alors cette règle, au travers d'un script, à tous les sommets de l'objet (ce script est équivalent à une boucle *for*). Cette approche est cependant moins efficace, en termes de temps de calcul, en général.

Cette section avait pour but de montrer les diverses formes que peuvent prendre les règles dans Jerboa. Il existe des règles de création ou de suppression des éléments et aussi des règles qui filtrent explicitement des cellules. Il existe également des règles plus génériques, dont les nœuds sont étiquetés par des variables topologiques et d'autres qui sont conçues pour n'effectuer que des mises à jour de plongement ou que des modifications topologiques. De même, il y a des règles qui filtrent toute une composante connexe et d'autres qui utilisent des scripts pour les appliquer sur toute la composante connexe. Pour cela, lors de la conception d'une règle, il faut réfléchir à son application, voire à sa généralisation (en type d'orbite, voire en dimension) afin de trouver la forme la plus pertinente qu'elle peut revêtir. Car souvent, plusieurs

possibilités co-existent, mais à des degrés divers de généricité et/ou d'efficacité.

3.5 Conclusion

Dans ce chapitre, nous avons présenté différents types de modèle topologique existants dans la littérature. Nous nous sommes focalisés sur les modèles cellulaires et en particulier sur ceux à représentation implicite des cellules qui offrent une solution robuste pour les opérations de modifications topologiques. En effet, ces dernières permettent en outre de garantir la propriété de quasi-variétés, indispensable à la simulation physique des objets maillés. Pour cela nous avons choisi de travailler avec le modèle topologique des G-cartes parce que c'est le modèle le plus atomique. Il nous fournit un nombre important d'orbites qui permettent de stocker les informations applicatives, dans notre cas liées à la mécanique, dont nous avons besoin. De plus ce modèle est exploité par le langage à base de règles Jerboa que nous avons utilisé. Ce langage permet de prototyper facilement un nombre très important d'opérations dans un temps réduit, car il permet de garantir la cohérence topologique de l'objet. Comme nous le détaillerons par la suite, il nous facilite la création d'un cadre général pour la simulation physique.

Synthèse : Modélisation mécanique à base topologique

Sommaire

4.1	Introduction	91
4.2	Lien entre modèle topologique et modèle physique	92
4.3	Simulation à base de maillages tétraédriques	93
4.4	Simulation du modèle masses/ressorts pour maillages rectangulaires ou hexaédriques	95
4.5	Simulation avec maillages homogènes tétraédriques ou hexaédriques	97
4.6	Utilité des modèles topologiques en simulation mécanique	98
4.7	Conclusion	99

4.1 Introduction

Nous avons présenté dans le premier chapitre de ce manuscrit, différents types de modèles mécaniques définis dans la littérature, qui se décomposent en deux grandes familles : les modèles basés sur la mécanique discrète et ceux basés sur la mécanique des milieux continus. Nous avons constaté que la simulation de déchirures, fractures ou découpes nécessitent l'emploi de modèles suffisamment souples, permettant de prendre en compte les modifications topologiques. Dans le deuxième chapitre, nous avons donné une panoplie de différents modèles topologiques, et nous nous sommes focalisés sur les cartes combinatoires (orientées ou généralisées) parce qu'elles permettent de représenter des quasi-variétés sous forme de maillages de type quelconque. Dans le présent chapitre, qui conclut de façon synthétique notre état de l'art, nous présentons des travaux qui proposent d'associer un modèle topologique avec un modèle mécanique pour la simulation physique de déformation et des transformations topologiques.

Nous commençons par une discussion visant à définir de façon plus précise le lien que l'on peut construire entre un modèle topologique et un modèle mécanique. Dans la suite, nous faisons l'hypothèse que l'objet est maillé et nous examinons divers modèles proposés dans la littérature. Nous classons ces modèles en fonction du

type de maillage et du modèle mécanique utilisés : modèles à base de maillages tétraédriques d’abord, puis maillages rectangulaires (2D) et hexaédriques (3D). Puis nous présentons une approche pour simuler des maillages homogènes quelconques (*i.e.* tétraédriques ou hexaédriques) pour des modèles mécaniques discrets et continus. Nous terminons ce chapitre par une synthèse des avantages qu’ont apportés les cartes combinatoires pour la simulation physique dans les divers travaux précédemment cités.

4.2 Lien entre modèle topologique et modèle physique

Le chapitre 2 a montré que les modèles physiques peuvent se baser ou non sur un maillage. Dans le cas de modèles physiques non maillés ou *meshless*, la condition de variété n’est utile que pour la visualisation de l’objet. Dans ce cas, une base topologique peut être un support au modèle géométrique servant à la visualisation de l’objet, si l’on souhaite pouvoir faire varier la topologie de cet objet. Dans le cas de modèles physiques maillés, il apparaît opportun qu’un modèle topologique sous-jacent au modèle mécanique permette de gérer les modifications topologiques que cet objet doit pouvoir subir. Dans le premier cas, on parle de modèle à *base mécanique* [LCM⁺14], dans le second cas, on parle de modèle à *base topologique*.

Concernant les approches à base mécanique, l’approche développée par les laboratoires ICA et XLIM [DKS⁺11] montre l’intérêt d’utiliser les G-cartes pour bâtir un habillage à topologie variable pour un modèle mécanique de type masses/interactions. Des phénomènes comme la fusion, les déchirures et les fractures peuvent être représentés par des transformations du modèle visuel, sans que le modèle masses/interactions sous-jacent, en charge du mouvement, ne nécessite de modifications. Dans cette approche, les auteurs utilisent le système CORDIS-ANIMA. Les sommets du modèle topologique sont ici directement associés aux particules (éléments *MAT*) du modèle mécanique. Des arêtes sont construites à partir de l’ensemble ou d’un sous-ensemble des interactions (éléments *LIA*). Cette approche a été généralisée par la suite [JAD⁺12], afin que chaque sommet soit associé à un ensemble d’éléments *MAT* pour calculer sa position au cours du temps. Une mise à jour topologiquement cohérente de ces sous-ensembles est alors proposée lorsqu’une modification topologique apparaît (qu’elle soit pilotée par le modèle mécanique ou déduite de son comportement) et qu’un sommet du modèle topologique se subdivise en plusieurs. Enfin, il est possible que le modèle mécanique soit explicitement à l’origine des modifications de l’habillage topologique : c’est par exemple le cas des éléments *MAT* éclatables [KLC14] qui permettent de mieux gérer les modifications du modèle topologique [RCDL18].

La suite de ce document se focalise sur les modèles à base topologique, donc exclusivement maillés. L’utilisation de modèles topologiques élaborés de type cartes combinatoires ou assimilés n’est pas très répandue dans le monde de la simulation. Citons, par exemple, la bibliothèque SOFA [Sof], qui se base essentiellement sur un modèle de graphe d’incidence indicé, avec les conséquences indésirables que nous

avons énumérées en section 3.3.1. Certaines simulations d’objets 2D avec modifications topologiques se sont sans aucun doute basées sur des modèles de type arêtes ailées ou demi-arêtes, sans toutefois nécessairement le signaler (par exemple, en simulation chirurgicale, pour la simulation de déchirure de tissus biologiques au début des années 2000). Seuls des travaux récents comme [SvWC14] mentionnent explicitement l’utilisation des demi-arêtes. Citons néanmoins des travaux précurseurs comme ceux de Nienhuys *et al.* [NvdS00], dans lesquels un modèle topologique simplicial est utilisé pour simuler la découpe dans un maillage, obligatoirement triangulaire en 2D ou tétraédrique en 3D, animé par la méthode des éléments finis. Nous présentons dans la suite de ce chapitre des approches à base topologique où le maillage est décrit par des cartes combinatoires.

4.3 Simulation à base de maillages tétraédriques

La première approche, dans laquelle est montré l’intérêt de simuler des objets déformables en basant un modèle physique sur un maillage décrit pas des cartes combinatoires est celle de Meseure *et al.* [MDS10]. Dans leur approche, les auteurs utilisent le modèle masses/ressorts comme modèle mécanique et le modèle des cartes généralisées comme modèle topologique. Dans le modèle proposé, seuls des objets représentés par des maillages tétraédriques peuvent être simulés, car seuls les ressorts d’élongation sont modélisés. Les auteurs proposent également dans leur approche, des méthodes de changements topologiques comme la séparation des volumes ou leur suppression pour effectuer des opérations de découpe et de déchirure au cours de la simulation. Ces transformations topologiques sont basées sur une opération fondamentale, la décousure α_3 , qui permet de séparer deux volumes adjacents. L’opération de décousure est également nécessaire à la suppression de volume puisque ce dernier doit préalablement être isolé (donc décousu de ses volumes adjacents) avant d’être supprimé. Quelques résultats de leurs transformations topologiques sont montrés sur la figure 4.1.

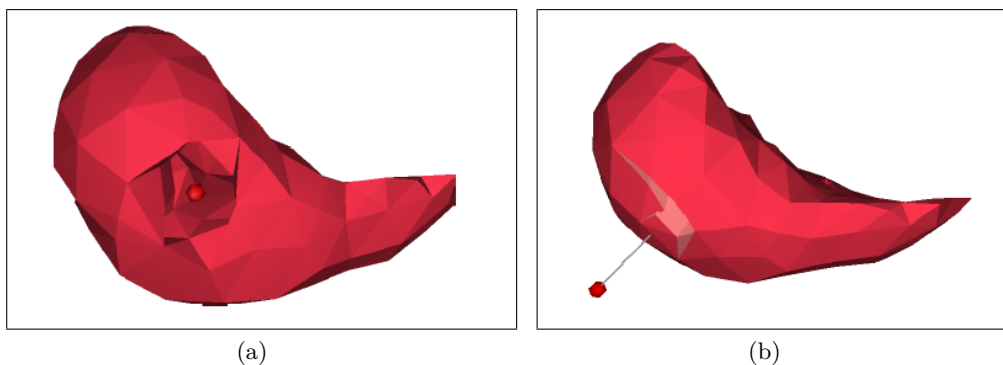


FIGURE 4.1 – Résultats de transformation topologique de Meseure *et al.*

Dans cette approche, il est discuté de l’intérêt de stocker dans la structure topo-

logique, en tant que plongement, toutes les propriétés ou informations mécaniques. Une particule est ainsi associée à un sommet c'est-à-dire à une 0-cellule. Dans les sommets sont plongées les propriétés des particules comme la masse, la viscosité dans le milieu ambiant, etc. ainsi que des informations d'état comme la position, la vitesse, le bilan des forces exercées. Dans ce modèle, les interactions sont retreintes aux ressorts d'élongation, positionnés sur les arêtes, ce qui empêche l'extension de l'approche à des maillages non simpliciaux, puisque ces derniers nécessitent des ressorts de cisaillement, ainsi qu'à son utilisation en 2D, puisque les ressorts de courbure ne sont pas pris en compte non plus. Le modèle se réduit ainsi aux seuls maillage tétraédriques.

Dans cette étude, il est mis en évidence que toute cellule qui, suite à une opération de découpe α_3 , éclate en plusieurs cellules doit mettre à jour ses informations mécaniques : la masse et la viscosité des particules aux sommets, la raideur des ressorts d'arêtes, etc. Par exemple, quand un sommet est éclaté, les auteurs choisissent de dupliquer sa position et sa vitesse dans les nouveaux sommets. Leur masse est obtenue en calculant la somme des masses fournies par chaque volume qui leur est incident. Précisons que, dans cette approche, les auteurs considèrent que la masse d'un tétraèdre est répartie équitablement entre ses quatre sommets :

$$m^s = \sum_{i \in \text{tetra}(s)} m_i / 4 \quad (4.1)$$

De la même manière que pour la masse, la raideur d'un ressort placé sur une arête est égale à la somme des raideurs apportées par chaque volumes incidents à l'arête (loi de Kirchhoff pour les ressorts disposés en parallèle). Après l'éclatement, chaque nouvelle arête doit recalculer sa propre raideur, à partir de ses volumes incidents.

Par ailleurs, pour des raisons de rapidité d'accès, les auteurs ont utilisé une structure simplifiée, à base d'indices, pour permettre un accès immédiats aux sommets, arêtes et volumes ainsi qu'aux faces externes (α_3 -libres) pour l'affichage. Quelle que soit la cellule, la structure simplifiée est constituée d'une liste fournissant, pour chaque cellule, un brin représentant la cellule et un lien vers le(s) plongement(s) porté(s) par celle-ci. Cette structure est nécessaire car le moteur Moka utilisé [Mok] ne propose pas d'outils efficaces pour parcourir spécifiquement les cellules (des parcours de type « Pour tous les sommets/arêtes/faces/volumes faire » ; les parcours proposés reviennent toujours à parcourir toute la carte, ce qui est coûteux). Ce manque a été pallié dans des moteurs de cartes plus récents comme CGoGN [CGo] par exemple. Cette solution a permis d'améliorer le temps d'exécution de leur méthode, mais elle nécessite la mise à jour de cette structure indicée redondante après tout changement topologique (aucune solution de mise à jour incrémentale n'est proposée, tous les tableaux sont complètement recalculés). Cette approche met en exergue le fait que toute structure redondante au modèle topologique se paie généralement par l'obligation de mises à jour et par l'absence d'une garantie de cohérence.

Plus récemment, Meseure *et al.* [MDST15] ont étendu leur approche pour simuler également des maillages tétraédriques, mais dont les lois de comportement

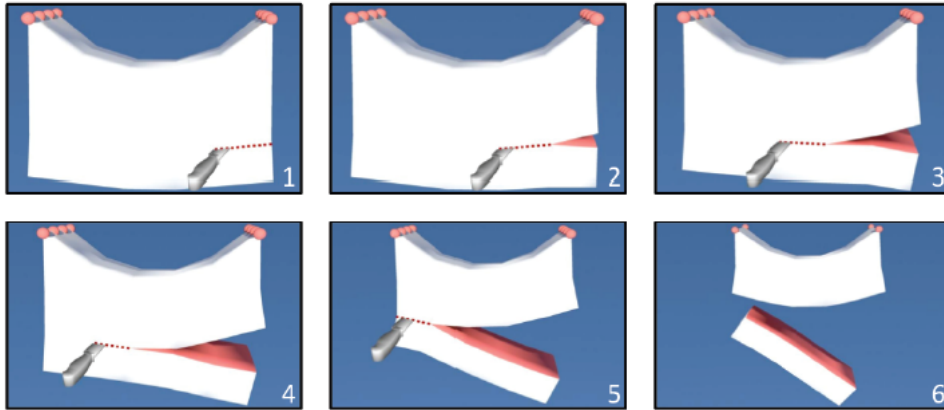
sont basées sur la mécanique des milieux continus, résolue par la méthode des éléments finis explicites (petits déplacements ou co-rotationnel). Des plongements sont à ajouter, comme par exemple le plongement dans les tétraèdres de leur matrice de raideur, ainsi que la matrice inverse des positions des sommets lors de l'utilisation de la méthode du co-rotationnel. Les auteurs ont également défini une transformation topologique plus complexe, à savoir une opération de subdivision offrant une résolution adaptative du maillage. Là aussi, des mises à jour des plongements sont nécessaires après la subdivision : découpage de ressorts en groupes de ressorts positionnés en série, calcul des matrices éléments finis des sous-tétraèdres, re-calcule des masses des sommets impliqués, etc. Le moteur de G-cartes n'ayant pas été modifié, les structures externes à base d'indices sont toujours utilisées dans cette version.

Dans [PUC⁺15], Paulus *et al.* ont également utilisé une base topologique de type cartes orientées avec un modèle physique basé sur la mécanique continue et résolue avec la méthode des éléments finis étendues (*extended FEM* ou *XFEM*) pour simuler les découpes dans des maillages tétraédriques. Cette méthode se base sur des fonctions d'interpolation linéaires ou quadratiques présentant des discontinuités aux zones de découpe. Ils ont également défini des transformations topologiques plus complexes comme la modification de maillage basée sur une subdivision $\sqrt{3}$ (méthode dite « flip/flap ») pour adapter la forme de l'élément à la découpe qui doit être appliquée au maillage. Des mises à jour de propriétés mécaniques (masses et matrices éléments finis) sont effectuées après la modification topologique. Les auteurs ont confirmé que leur travail est basé sur les cartes orientées mais aucune information concernant l'implantation et en particulier les plongements choisis n'est fournie.

4.4 Simulation du modèle masses/ressorts pour maillages rectangulaires ou hexaédriques

Les travaux de Fléchon *et al.* [FZDJ13] ont combiné la structure topologique des cartes orientées avec un modèle masses/ressorts, pour l'animation physique des objets déformables. Plus exactement, les auteurs utilisent des cartes orientées dotées d'un plongement de position en chaque sommet et appelées « complexes linéaires cellulaires » ou *Linear Cell Complex* (LCC). Dans ces travaux, les objets à animer sont représentés uniquement par des maillages rectangulaires en 2D et hexaédriques en 3D. Des changements topologiques sont proposés : l'opération de découpe, comme montré sur la figure 4.2 et les changements de résolution (raffinement/subdivision ou simplification de maillage).

Fléchon *et al.* [FZDJ13] reprennent les mêmes hypothèses de plongement des informations dans le modèle topologique que celles proposées dans [MDS10] : une particule est associée à chaque cellule sommet et des ressorts d'élongation aux arêtes du maillage. Notons néanmoins que dans ce modèle, les ressorts d'élongation font explicitement référence aux sommets qu'ils lient. Par ailleurs, l'utilisation de maillages non symplectiques implique l'ajout d'autres types de ressorts comme les ressorts de

FIGURE 4.2 – Exemples de résultats de l’approche de Fléchon *et al.*

cisaillement. Dans leur approche, les propriétés mécaniques de ces ressorts (longueur au repos et raideur) sont stockées dans des conteneurs indépendants de la structure du modèle, en indiquant explicitement les sommets sur lesquels ils s’appliquent. En effet, les ressorts de cisaillement sont positionnés sur les diagonales des faces quadrangulaires en 2D et des hexaèdres en 3D, qui ne sont pas des cellules dans les cartes combinatoires. Aussi, aucune solution de plongement des ressorts de cisaillement dans la structure n’est proposée. De plus, même si un modèle 2D est proposé, les ressorts contrôlant la courbure ne sont pas explicitement étudiés, mais on peut supposer qu’une prise en compte similaire à celle des ressorts de cisaillement est envisageable.

En utilisant des structures externes pour les ressorts de cisaillement ou en forçant les ressorts à gérer les sommets sur lesquels ils s’appliquent, la stratégie employée par Fléchon *et al.* est risquée. En effet, lorsque le modèle subit des changements topologiques, le modèle topologique assure sa propre cohérence, mais rien ne permet de s’assurer de la cohérence des liens topologiques gérés par les structures externes. On pourrait, par exemple, garder des interactions liées à une cellule qui a disparu.

Comme dans [MDS10], une mise à jour des propriétés mécaniques plongées dans les cellules concernées par une modification topologique est nécessaire. Ainsi, après l’éclatement d’un sommet, la masse de chaque nouveau sommet est à recalculer à partir de masses fournies par chaque élément incident au nouveau sommet. Lorsqu’une arête est scindée, les raideurs de ressort d’élongation des arêtes nouvellement créées sont recalculées en se basant sur les formules décrites dans la section 2.2.2.3, et en considérant la contribution de chaque face ou volume incident. Même si le fait de séparer deux éléments (faces ou volumes) ne semble pas concerner les éléments eux-mêmes, l’utilisation de structures externes compliquent les mises à jour. En réalité, pour chaque modification topologique, et pour tout type de ressort, une phase de post-traitement gérant la cohérence des extrémités des ressorts est systématiquement proposée. En effet, si un sommet éclate en deux sommets ou plus, alors qu’il est l’extrémité d’un ou plusieurs ressorts, il faut déterminer, pour chaque

ressort, à quelle nouveau sommet il doit être lié. Par exemple, si deux volumes sont séparés alors qu'ils partageaient une face commune, les ressorts touchés par cette séparation doivent rester dans leur volume d'origine : un ressort d'élongation reste sur une arête de volume, un ressort de cisaillement lie deux extrémités d'une même diagonale. Il ne faut pas que, suite à une mauvaise identification de ses extrémités après la transformation topologique, un ressort lie des sommets qui n'appartiennent pas au même élément. C'est pour cette raison que, dans les travaux qui ont suivi [GCZD15], les ressorts font désormais référence non à des sommets, mais à des brins, judicieusement choisis, les désignant. Les brins, en tant que tels, n'étant pas modifiés par les opérations topologiques utilisées, les ressorts désignent toujours les mêmes sommets.

4.5 Simulation avec maillages homogènes tétraédriques ou hexaédriques

Récemment, Golec *et al.* [GCZD15] ont étendu le modèle de Fléchon *et al.* pour permettre de simuler des mailles hexaédriques ou tétraédriques en utilisant non seulement le modèle mécanique masses/ressorts, mais également le modèle des masses/tenseurs. Dans cette approche, les auteurs ont utilisé la même modélisation que dans les travaux de Fléchon *et al.* mais veulent incorporer toutes les propriétés physiques dans le modèle topologique. De cette manière, les mises à jour des propriétés mécaniques après un changement topologique peuvent être gérées de façon cohérente. Une structure externe est néanmoins associée à chaque volume, à savoir un tableau de brins permettant de définir d'une part les cellules dans un ordre donné et, grâce à cet ordre, de stocker des informations liées à des groupes de cellules du volume (pour désigner, par exemple, des diagonales de volume par leur sommets).

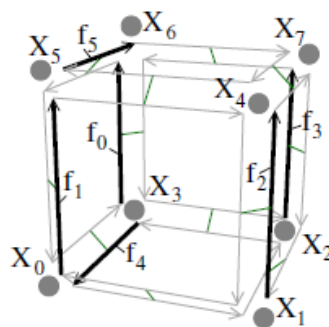


FIGURE 4.3 – Convention de stockage de brins dans des tableaux

Cette structure se base sur la convention décrite sur la figure 4.3. La structure de chaque hexaèdre est décrite dans un tableau bi-dimensionnel de brins $brins[j][i]$. Le premier indice j , pris dans l'ensemble $0, \dots, 5$ désigne une face de l'hexaèdre. Le second indice i permet de décrire les quatre brins constituant la face j . Ainsi, pour

chaque j , le brin $brin[j][0]$ désigne le premier brin de la face j , représentée en gras sur la figure 4.3. On retrouve le fait que plusieurs brins désignent un même sommet, par exemple, les brins $brins[0][3]$, $brins[4][1]$ et $brins[1][0]$. A partir du numérotage des faces, ce tableau permet de numéroter tout autre cellule, en particulier les sommets, mais également d'indiquer un ordre des diagonales du volume ou des faces. Mais il apparaît que les brins du tableau sont nécessairement liés par des liaisons bien précises, et cette structure se révèle complètement redondante avec la structure LCC de l'objet.

Dans cette structure, il est possible de plonger des informations comme les ressorts de cisaillement, mais aussi, plus globalement les influences de chaque sommet sur tous les autres, sous la forme de tenseurs (approche masses/tenseurs développée en section 2.2.2.3) correspondant aux arêtes, aux diagonales de faces et aux diagonales de volume.

4.6 Utilité des modèles topologiques en simulation mécanique

Dans ce chapitre, nous avons présenté des travaux qui utilisent des modèles topologiques pour la simulation physique de corps solides déformables. Comme on peut le remarquer, ces différentes approches ont utilisé des cartes combinatoires (orientées ou généralisées). Nous regroupons dans cette partie les arguments avancés par ces approches.

- La définition mathématique des modèles topologiques comme les cartes combinatoires assure leur robustesse. Après une modification, il est possible de s'assurer de la cohérence de la structure en vérifiant chaque point de leur définition ;
- De nombreuses opérations ont été définies, de façon robuste voire même formellement [LMA⁺02, CD99], pour ces modèles et sont autant de transformations topologiques que l'on peut exploiter dans un contexte d'animation physique ;
- Les objets représentables par les structures topologiques ne sont pas quelconques. Ainsi, les cartes combinatoires ne peuvent représenter que des quasi-variétés. Il est alors strictement impossible de simuler des objets présentant des faces (en 3D) ou arêtes (en 2D ou 3D) pendantes, des volumes liés seulement par une arête ou un sommet, des faces liées seulement par un sommet, etc. Aucune recherche et traitement des cas dégénérés comme proposé dans [FDA05] n'est nécessaire ;
- Certains modèles topologiques ne permettent de représenter que des assemblages de simplexes, ce qui peut être utile pour des maillages purement simpliciaux, mais cela restreint les possibilités de modifications du maillage. Au

contraire, des modèles comme les cartes combinatoires permettent de représenter de nombreux types de maillages, avec des cellules quelconques, de façon homogène (maillages constitués d'un seul type de cellules) ou pas, ce qui offre une grande souplesse dans les opérations de transformation du maillage.

- La définition des modèles topologiques est souvent générale à toute dimension, ce qui permet d'utiliser le même modèle quelle que soit la dimension topologique de l'objet à représenter. Néanmoins, dans les travaux précédents, les implantations 2D et 3D de modèles mécaniques basés topologie sont (au moins en partie) distinctes. La question de l'exploitation d'un modèle unique pour la 2D et la 3D (voire également pour la 1D) reste ouverte.

4.7 Conclusion

Dans ce chapitre, nous avons présenté les différents travaux de la littérature qui ont combiné un modèle topologique avec un modèle physique. Ces approches ont chacune eu recours à une structure annexe. Pourtant, suite à l'application d'une modifications topologique, la structure topologique assure sa propre cohérence alors que rien ne permet de s'assurer de la cohérence des structures additionnelles. Plus globalement, toute information persistante faisant référence à un élément de la structure topologique (par exemple un brin pour les cartes combinatoires) est potentiellement source d'incohérences si des mises à jour sont omises suite à une ou plusieurs modifications de la structure. En conséquence, l'utilisation de structures externes est critiquable et apparaît, de notre point de vue, comme une sous-exploitation des capacités de plongement des modèles topologiques.

Nous pensons que cet ajout d'information redondante est dû à une modélisation inadéquate des interactions. En effet, c'est une erreur de chercher des orbites topologiques qui supportent à la fois l'origine d'une interaction et les particules subissant ces forces. Certes, cette approche fonctionne pour les ressorts d'élongation, parce qu'ils sont naturellement associés aux arêtes. Mais pour les autres types d'interaction (cisaillement, courbure), de tels orbites n'existent pas nécessairement. Il s'agit donc de revoir la méthode de plongement des informations mécaniques afin d'exploiter totalement les solutions de plongement offertes par les modèles topologiques.

Par ailleurs, les divers travaux présentés se cantonnent à l'utilisation de mailles élémentaires identiques (maillages triangulaires, tétraédriques, hexaédriques, etc.). Ces restrictions ne sont pas justifiées : des modèles comme les cartes combinatoires permettent de manipuler des cellules de types différents et des approches de résolution mécanique comme celles des éléments finis prévoient, depuis leur conception, l'agglomération de différents types de cellules. Par exemple, des maillages volumiques constitués de tétraèdres et d'hexaèdres, ainsi que des volumes de transitions comme les pyramides et les prismes, sont tout à fait courants dans les moteurs éléments finis. Dans nos travaux, nous souhaitons une approche autorisant les maillages hétérogènes, où les données et les traitements sont à effectuer de façon la plus indé-

100 Chapitre 4. Synthèse : Modélisation mécanique à base topologique

pendante possible de la dimension. Pour cela, nous avons utilisé le langage à base de règles de transformation de graphes, Jerboa, décrit dans le chapitre précédent, pour :

- Formaliser la simulation physique en exploitant le modèle topologique des G-cartes qui nous a permis de stocker toutes les informations nécessaires pour la simulation dans le modèle sans utilisation d'une structure externe ;
- Simuler les déformations des objets déformables en 2D et 3D avec différents modèles mécaniques (masses/ressorts, méthode des éléments finis linéaires, approche par co-rotationnel, masses/tenseurs) ;
- Simuler avec différents types de maillages homogènes ou non ;
- Simuler différents types de transformations topologiques.

Dans la suite, nous présentons, dans un premier temps, la modélisation qui nous permet de plonger toutes les informations mécaniques, de façon adaptée, dans le modèle. Nous présentons, dans un deuxième temps, la simulation en nous basant sur le langage à base de règles. Puis, nous focalisons sur les différents types de modifications topologiques implantées dans notre plate-forme.

Deuxième partie

Contribution

Modélisation de corps physiques basés topologie

Sommaire

5.1	Hypothèses et modèle support	104
5.1.1	Hypothèses	104
5.1.2	Modèle support	105
5.2	Modélisation des interactions	107
5.2.1	Méthode générale	108
5.2.2	Modèles mécaniques discrets 2D	109
5.2.3	Modèles mécaniques discrets 3D	118
5.2.4	Modèles mécaniques continus	120
5.3	Adaptation de la modélisation pour les cartes orientées	127
5.4	conclusion	129

Dans le chapitre précédent, nous avons conclu qu'il est utile de combiner un modèle topologique avec un modèle mécanique pour la simulation physique de corps déformables et plus précisément lors de transformations topologiques. Nous avons également présenté les lacunes des travaux précédents (stockage de données dans des structures externes, simulation d'un seul type d'éléments, dimension fixe...). Le problème est lié au fait que les informations mécaniques à plonger dans le modèle ne peuvent l'être que dans des orbites, afin de contrôler ce plongement lors des opérations topologiques. L'objectif de ce chapitre est de trouver les plongements et les orbites les plus appropriés pour stocker ces informations mécaniques (les informations liées aux particules et les différents types d'interactions qui sont implantées dans notre plate-forme à l'aide du langage à base des règles de transformations de graphes Jerboa) en 2D et 3D. Nous commençons ce chapitre par la présentation des hypothèses et du modèle support utilisé dans notre travail. Nous détaillons ensuite la méthode générale utilisée pour trouver l'orbite adéquate pour chaque information et la modélisation faite avec les G-cartes pour chaque type d'interaction. Nous terminons ce chapitre avec une adaptation de la méthode générale proposée avec le modèle topologique des cartes orientées.

5.1 Hypothèses et modèle support

5.1.1 Hypothèses

Dans cette partie, nous introduisons les hypothèses utilisées dans le cadre de notre approche. Nous avons montré, dans la section 2.2.2.1 de l'état de l'art, que dans notre approche, nous nous intéressons exclusivement à la simulation des objets décrits par des maillages géométriques. La modélisation des interactions que nous utilisons dans notre plate-forme est faite avec le modèle topologique des G-cartes.

Notre processus de modélisation d'un objet commence par le choix d'un maillage géométrique (voir figure 5.1). Ce dernier peut être construit explicitement et directement avec le langage à base de règles comme nous le faisons en section 3.4.2 (page 81). Cependant, Jerboa permet également le chargement direct de fichiers (obj, gmsh, moka...) de maillages géométriques d'objets.

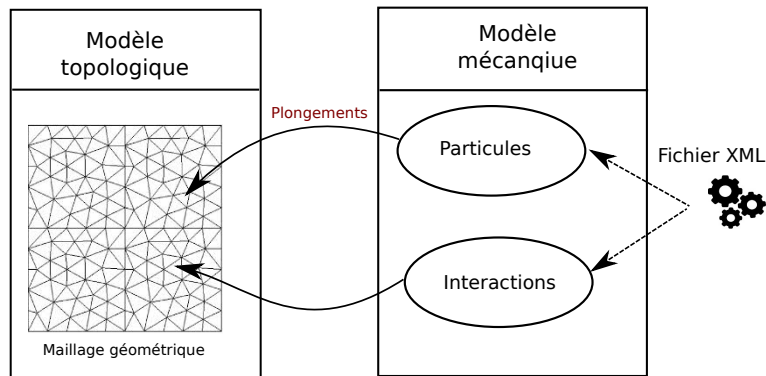


FIGURE 5.1 – Processus de modélisation.

Après le choix du maillage (construit ou chargé directement avec Jerboa), il faut le peupler avec les informations mécaniques qui sont incluses dans le modèle géométrique via le processus de plongement, comme montré sur la figure 5.1. Par exemple, après l'ajout de plongements à une surface damier créée au préalable, le modèle peut être considéré comme un tissu avec des propriétés mécaniques bien définies et prêt pour la simulation. Les propriétés mécaniques des différents types d'interactions implantées (ressorts et éléments finis), les masses et toutes les informations nécessaires à la simulation sont décrites dans un fichier XML.

Une autre hypothèse est très importante dans notre approche : nous supposons que les modèles utilisés sont des modèles particulaires. Plus précisément, une particule est placée sur chaque sommet du maillage et permet de l'animer, et la déformation de l'objet résulte du déplacement des sommets. Aucune hypothèse n'est faite concernant le maillage, il peut être hétérogène, c'est à dire constitué d'éléments de types divers (triangles et quadrilatères, tétraèdres et hexaèdres). Les lois utilisées peuvent être anisotropes.

5.1.2 Modèle support

Dans l'état de l'art sur les modèles mécaniques, nous avons présenté plusieurs modèles pour simuler les déformations de corps déformables. Dans notre approche, nous souhaitons utiliser un ensemble aussi vaste que possible de ces modèles et les combiner. Pour cela, il nous faut une base commune. L'objectif de cette section est de proposer un modèle support, alors que la modélisation des interactions proprement dite fait l'objet de la section suivante. Notre approche considère un modèle support dont le but est de contrôler les degrés de liberté du corps à simuler. Comme déjà dit, nous plaçons une particule en chaque sommet du maillage. Ceci implique la définition d'une masse, d'une vitesse et d'une position pour chacune des particules. Ces sont les informations à plonger dans le modèle. Un tel plongement a déjà été proposé par Meseure *et al.* [MDS10] et Fléchon *et al.* [FZDJ14]. Dans leurs travaux, ils ont associés toutes ces propriétés mécaniques aux orbites sommets ($\langle \alpha_1, \alpha_2 \rangle$ en 2D et $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$ en 3D). Mais cette solution est discutable, en particulier pour la masse, car elle provient des éléments, et c'est dans ces derniers qu'elle aurait pu être stockée. Il s'agit alors d'identifier l'orbite la plus adéquate et la plus naturelle compte tenu de l'origine et la signification physique de chaque plongement. Parfois, on peut trouver directement l'orbite qui convient à l'information physique. Par exemple pour la position, la vitesse et l'accélération de la particule, compte tenu de notre hypothèse initiale, ces données doivent être stockées dans des plongements portés par les sommets du maillage, qui correspondent aux orbites $\langle \alpha_1, \alpha_2 \rangle$ en 2D ou $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$ en 3D. Mais concernant la masse, plusieurs solutions sont possibles et il faut évaluer la pertinence de chacune.

Pour définir le plongement qui porte l'information de masse, nous considérons l'origine de cette propriété physique. Dans notre modèle, nous supposons que la masse est répartie sur la surface de l'objet 2D ou son volume en 3D. Il semble alors plus naturel d'associer la masse aux faces ou aux volumes plutôt qu'aux sommets. Cependant nous travaillons avec une structure discrète, dans laquelle les masses doivent être concentrées sur les particules, et non pas une structure continue. Le calcul de l'accélération de chaque particule nécessite de connaître sa masse. Or celle-ci provient des faces/volumes environnant le sommet support. Ainsi, chaque face/volume répartit uniformément sa masse sur ses sommets (coins de la face en 2D, coins du volume en 3D). Par exemple, la masse d'un sommet incident à deux faces est égale à la somme des masses associées aux deux coins incidents à ce sommet. Compte tenu de cette remarque, nous associons deux types de plongements pour la propriété physique masse. Le premier est le plongement de masse qui est lié à un sommet (coin) de face/de volume, donc le plongement associé aux sous-orbites $\langle \alpha_1 \rangle$ / $\langle \alpha_1, \alpha_2 \rangle$, et un deuxième plongement qui est lié aux sommets, donc associé aux orbites $\langle \alpha_1, \alpha_2 \rangle$ / $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$. La figure 5.2 montre deux faces quadrilatères adjacentes présentées avec Jerboa. Elle montre aussi les deux types de plongements en 2D, le plongement **Masse-coin** qui est porté par les orbites $\langle \alpha_1 \rangle$ soit sur les deux brins qui sont liés par une liaison α_1 . De même le plongement **Masse-sommet** qui est porté par les orbites $\langle \alpha_1, \alpha_2 \rangle$ soit sur tous les brins qui sont liés par les liaisons

Plongements mécaniques du modèle support			
Masse-sommet	Masse-coin	Vitesse	Position
orbites sommets	orbites sommets d'élément	orbites sommets	orbites sommets
2D : $\langle a1, a2 \rangle$	2D : $\langle a1 \rangle$	2D : $\langle a1, a2 \rangle$	2D : $\langle a1, a2 \rangle$
3D : $\langle a1, a2, a3 \rangle$	3D : $\langle a1, a2 \rangle$	3D : $\langle a1, a2, a3 \rangle$	3D : $\langle a1, a2, a3 \rangle$

FIGURE 5.3 – Plongements liés aux particules.

- Plongement **Masse-coin** : porté par les orbites sommets d'élément dans lesquels nous enregistrons la valeur de masse d'un coin d'une face ou d'un volume.
- Plongement **Masse-sommet** : porté par les orbites sommets et calculé à partir des masses-coins.

Dans la section suivante, nous introduisons la modélisation faite pour stocker les propriétés mécaniques des interactions dans les orbites les plus adéquates.

5.2 Modélisation des interactions

Pour simuler le comportement et la déformation des objets au cours du temps, il faut se baser sur des lois de comportements. Dans cette partie nous nous intéressons aux interactions décrites sur la figure 5.4.

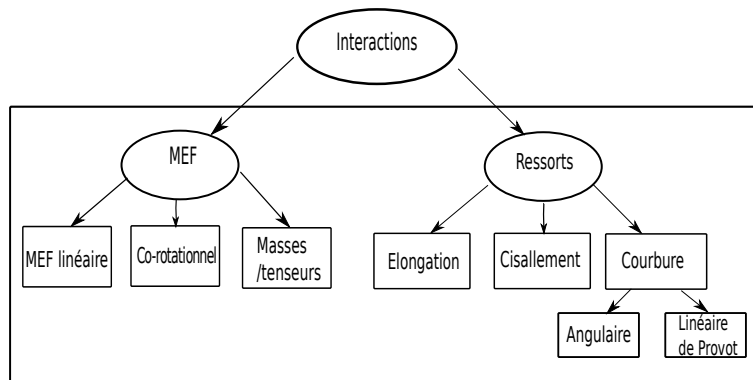


FIGURE 5.4 – Interactions à modéliser.

À partir de l'étude de quelques interactions, nous avons défini une méthode générale qui permet de trouver le lieu de stockage de chaque propriété mécanique ainsi que celui des forces appliquées par chaque interaction, selon son origine physique. Aussi, cette méthode facilite l'introduction de n'importe quelle interaction dans notre plate-forme. Nous expliquons alors, dans cette partie, les étapes de cette méthode et son application aux différentes interactions de la figure 5.4.

5.2.1 Méthode générale

Nous avons remarqué que les travaux antérieurs [MDS10, FZDJ13, GCZD15], introduits dans le chapitre 4, présentent des limites. En effet, dans ces approches, la solution pour les ressorts d'élongation est simple et immédiate, puisque leurs supports topologiques sont les arêtes du maillage. Mais les autres types d'interactions ne correspondent pas directement aux éléments topologiques des G-cartes (comme par exemple les diagonales de faces/volumes). Comme, ces lieux ne sont pas forcément des entités topologiques, le problème du recours à des structures annexes apparaît dans les travaux de Fléchon *et al.* et Golek *et al.*. Si, au contraire, nous nous focalisons sur l'origine des forces (un coin, une articulation entre faces, etc.) sans se soucier des points d'application, il est alors moins contraignant de trouver un lieu topologique pour associer le plongement des paramètres des interactions. Les lieux d'application des forces s'obtiennent alors grâce à un sous-graphe qui indique les voisinages à suivre à partir de lieu de stockage de l'interaction. Globalement, nous estimons que les limites des approches précédentes (le type d'éléments dans [MDS10] et l'utilisation de structures externes dans [FZDJ13, GCZD15]) peuvent être dues à une modélisation inadaptée des interactions car les auteurs de ces travaux cherchent des lieux topologiques qui correspondent à la fois à l'origine et aux lieux d'application des forces (par exemple un lien entre deux particules). Par conséquent, ils n'arrivent pas à trouver les lieux topologiques les plus adéquats et renoncent à certaines interactions et donc à certains éléments ou ont recours aux structures externes.

Pour modéliser et calculer les forces de n'importe quel type d'interactions (discrètes ou continues) en 2D ou 3D pour différents types de maillage, nous avons proposé la méthode suivante :

- localiser le lieu topologique correspondant à la source des interactions et y stocker leurs paramètres. Si lors d'une transformation topologique, ce lieu est modifié, alors l'interaction est nécessairement traitée en même temps que la transformation ;
- à partir de chaque lieu, définir un motif (*i.e* un sous-graphe) à filtrer qui désigne les particules participant à l'interaction (utiles aux calculs ou subissant les forces) ;
- stocker chaque force calculée dans une sous-orbite incluse à la fois dans l'orbite du sommet d'application et dans le sous-graphe défini à l'étape précédente.

Prenons l'exemple d'un ressort placé sur une arête entre deux masses (*i.e* entre deux sommets), comme le montre la figure 5.5. La source de l'interaction dans cet exemple est l'arête (encadrée en vert dans la figure 5.5 et les suivantes). L'ensemble des caractéristiques physiques du ressort sont ainsi portées par cette orbite verte. Les particules et leur masse sont portées par les sommets (entourés en rouge sur la figure 5.5 et les suivantes). Le motif à filtrer doit inclure à la fois les brins portant

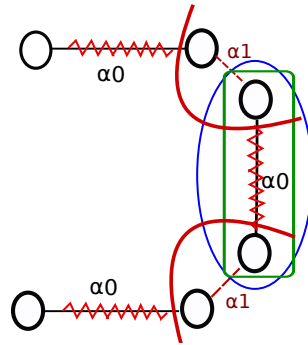


FIGURE 5.5 – Exemple de modélisation.

l'interaction et des brins portant chaque particule, pour accéder à toutes les informations nécessaires au calcul des forces (masses des particules et caractéristiques du ressort). Ce motif correspond, dans cet exemple simple, exactement à l'orbite de l'interaction (sous-graphe entouré en bleu sur la figure 5.5 et les suivantes). Les forces générées par ce ressort sont donc stockées dans les sous-orbites issues de l'intersection entre les ensembles de brins rouges et bleus, c'est à dire, dans notre exemple, les orbites brins. En cas de rupture du ressort (suppression de l'arête support), le motif bleu n'est plus filtré. Par conséquent, les forces correspondantes ne sont plus produites, sans qu'aucune mise à jour des données mécaniques ne soit nécessaire.

De façon générale, nous constatons que le point critique de notre méthode est de trouver le bon motif à filtrer, qui doit être choisi soigneusement, pour chaque type de ressort, afin d'être le plus général possible (pour être indépendant du type de mailles, par exemple). Comparativement, les sous-orbites où les forces sont stockées peuvent être trouvées immédiatement. En effet, la méthode choisit la plus grande orbite (voire, exceptionnellement, une sous-orbite) incluse dans l'intersection entre le sous-graphe bleu et l'orbite des sommets d'application des forces.

Nous commençons tout d'abord, par les interactions des modèles discrets et, dans une deuxième partie, nous nous intéressons aux interactions des modèles de la mécanique continue.

5.2.2 Modèles mécaniques discrets 2D

En utilisant la méthode décrite ci-dessus, nous allons commencer par la modélisation des ressorts utilisés en 2D et ensuite des ressorts utilisés en 3D. La structure mécanique en 2D que nous souhaitons modéliser s'inspire du modèle 2D de Provot [Pro95] présenté dans la section 2.2.2.3 (page 20) et basé sur trois types de ressorts (voir figure 5.6) : les ressorts d'élongation pour contrôler l'étirement et la compression, les ressorts de cisaillement pour maintenir la forme de la face (non nécessaire pour les faces triangulaires) et finalement les ressorts linéaires de Provot pour permettre de contrôler la flexion. Chaque ressort est supposé de type visco-élastique. Il est donc caractérisé par sa raideur, sa longueur au repos et son éventuel

amortissement.

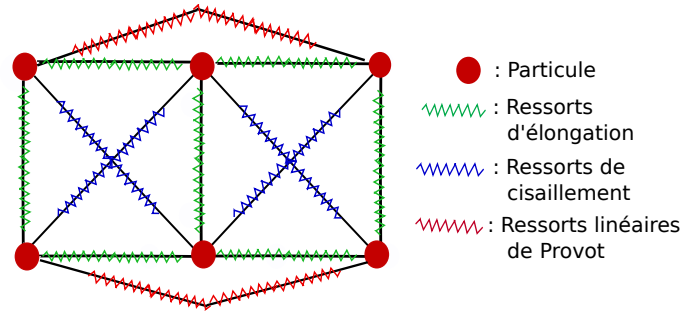


FIGURE 5.6 – Maille élémentaire (rectangulaire).

Nous avons également modélisé un autre type de ressorts, les ressorts angulaires, proposés dans [BW98] pour contrôler l'angle entre deux faces adjacentes. La méthodologie générale est appliquée pour chacun des ressorts cités au-dessus.

5.2.2.1 Ressorts d'élongation

Les ressorts d'élongation permettent de connecter deux particules placées sur les sommets d'un maillage géométrique et liées par une arête. En 2D, il semble naturel de stocker les propriétés mécaniques de ressort dans les orbites arêtes $\langle \alpha_0, \alpha_2 \rangle$ comme proposé dans la littérature. Néanmoins, nous proposons une solution qui exploite davantage la base topologique. En effet, du point de vue physique, pour une arête incidente à deux faces, chaque face latérale apporte une résistance à l'élongation (voir figure 2.16, page 23), d'où l'idée d'identifier la contribution de chacune des faces. Ceci peut se révéler utile lors d'un changement topologique comme la découpe (qui se traduit par une découure α_2 entre deux faces [MDS10]) : le ressort se trouve alors éclaté en deux ressorts élémentaires associés à chaque face. Par conséquent, on en déduit que les ressorts élémentaires correspondent en réalité aux arêtes de faces (orbite $\langle \alpha_0 \rangle$). Réciproquement, on considère que chaque face incidente apporte un ressort à l'arête commune.

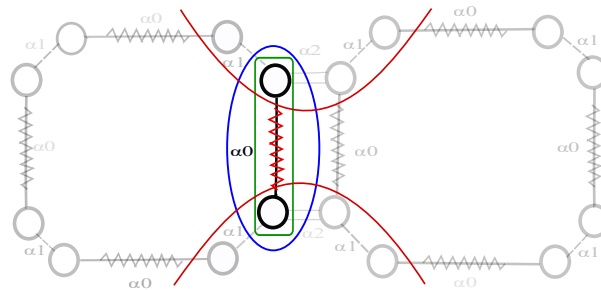


FIGURE 5.7 – Sous-graphe et orbites supports pour les ressorts d'élongation.

Sur la figure 5.7, les ressorts élémentaires sont portés par les arêtes de faces

(orbite $\langle\alpha_0\rangle$, encadrées en vert). Ils agissent sur les sommets de face placés à leurs extrémités (orbites $\langle\alpha_1\rangle$, entourés en rouge). Le motif à filtrer est donc simplement l'orbite $\langle\alpha_0\rangle$ des arêtes de faces (entouré en bleu). Les forces, une fois calculées, sont alors stockées dans les intersections entre les orbites arêtes de face $\langle\alpha_0\rangle$ (sous-graphe bleu) et les orbites sommets concernées (sous-graphes rouges). Dans ce cas, les intersections correspondent aux orbites brins $\langle\rangle$. Sur la figure 5.7, les faces sont représentées par un quadrilatère, mais ce principe peut être appliqué à n'importe quel type de faces indépendamment du nombre de sommets.

Nous avons créé un plongement nommé **R-elong-elem** porté par les orbites $\langle\alpha_0\rangle$ dans lesquelles sont stockées les propriétés du ressort comme montré sur la figure 5.8, extraite de Jerboa. Comme le plongement est associé aux orbites $\langle\alpha_0\rangle$, nous associons le plongement au nœud **n0** et il se propage automatiquement vers le nœud **n1**. Nous avons également créé le plongement dans lequel les forces sont stockées et nous l'avons appelé **F-elong-elem**. Il est porté par les orbites brins, donc chaque nœud **n0** et **n1** contient son propre plongement force.

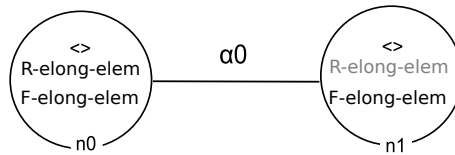


FIGURE 5.8 – Plongements associés à un ressort d'élongation.

La figure 5.7 illustre le fait que deux ressorts élémentaires contribuent à la même arête. Il en résulte un choix concernant l'optimisation du calcul des forces. Nous traitons cette question dans la section 6.4 du chapitre suivant.

5.2.2.2 Ressorts de cisaillement

Comme il est décrit dans le modèle de Provot, les ressorts de cisaillement correspondent, d'un point de vue géométrique, aux diagonales d'une face. Malheureusement, en cartes combinatoires, il n'existe pas un type d'orbite propre aux diagonales. Aussi, dans les travaux antérieurs, Meseure *et al.* [MDS10] n'abordent pas le problème car ils se contentent d'utiliser des mailles tétraédriques alors que Fléchon *et al.* [FZDJ13] et Golek *et al.* [GCZD15] ont utilisé des structures externes pour les stocker. Dans notre approche, le lieu de stockage des informations mécaniques d'une interaction est défini en se basant sur sa source physique. Pour trouver cette source, nous examinons deux raisons principales motivant l'ajout des ressorts de cisaillement :

- maintenir la forme des éléments ;
- préserver l'angle de chaque coin de la face.

Cette observation nous apporte deux solutions et donc deux manières différentes de modéliser les ressorts de cisaillement. Dans le premier cas, nous considérons un

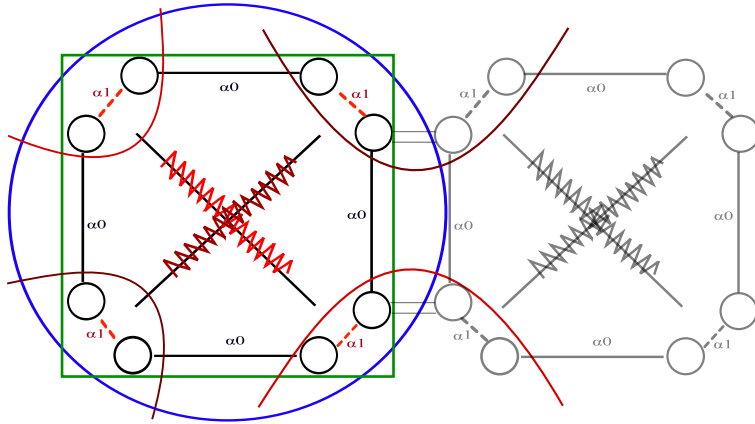


FIGURE 5.9 – Sous-graphe et orbites supports des ressorts de cisaillement surfacique isotrope.

comportement que nous qualifions d'*isotrope* de l'objet (les valeurs de raideur des deux ressorts diagonaux sont égaux). Une face produit des forces à chaque couple de sommets opposés liés par un ressort de cisaillement et les deux ressorts ont les mêmes paramètres. Alors que dans le deuxième cas, deux arêtes d'une face incidente à un sommet doivent garder un angle donné entre elles. Cet angle est contrôlé par des forces exercées entre les autres extrémités des deux arêtes. Il est ici possible d'associer des propriétés mécaniques différentes pour chaque couple d'arêtes. Par opposition à la précédente, cette approche est qualifiée d'*anisotrope*.

Dans le modèle isotrope, une maille rectangulaire doit nécessairement être utilisée. Dans ce cas, les deux ressorts de cisaillement du quadrilatère ont la même loi de comportement et la même longueur de repos et ils peuvent de cette manière partager le même plongement. Par conséquent, nous avons stocké une raideur, une valeur d'amortissement et une longueur de repos dans les orbites faces $\langle \alpha_0, \alpha_1 \rangle$. Chaque couple de sommets opposés peut être trouvé en parcourant simplement la face. Ainsi, la face est le sous-graphe nécessaire pour trouver les sommets d'application des forces. Comme précédemment, les intersections entre ce sous-graphe et les orbites sommets des particules subissant les forces sont utilisées pour trouver où stocker les forces calculées. Elles correspondent aux orbites $\langle \alpha_1 \rangle$, c'est-à-dire aux sommets de face, comme le montre la figure 5.9.

Le modèle anisotrope autorise l'utilisation d'éléments de forme quelconque et non pas uniquement rectangulaire (voir figure 5.10). Les propriétés mécaniques d'un ressort de cisaillement sont stockées dans les orbites $\langle \alpha_1 \rangle$ qui correspondent à l'articulation (indiquée en vert) entre les deux arêtes connectées par un sommet. Nous avons stocké dans ce plongement, la rigidité, l'amortissement et la longueur au repos. Il faut noter que, pour un quadrilatère, il existe deux orbites $\langle \alpha_1 \rangle$ différentes qui correspondent au même couple de sommets opposés. Pour éviter la redondance de calcul, nous associons un plongement pour un seul coin de face $\langle \alpha_1 \rangle$. En outre, les forces doivent être stockées dans l'intersection entre les deux arêtes connectées

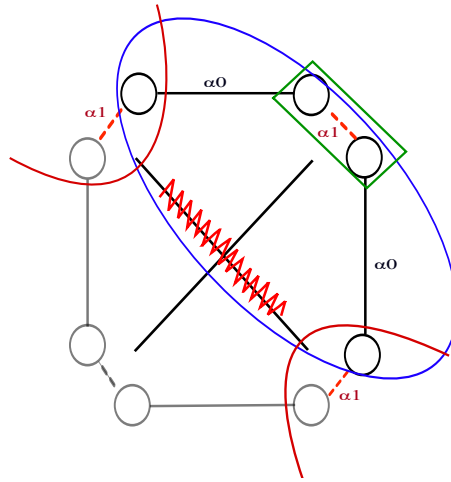


FIGURE 5.10 – Sous-graphe et orbites supports pour les ressorts de cisaillement surfacique anisotrope.

(sous-graphe bleu) et leurs extrémités non connectées (en rouge). L'intersection est constituée, comme on peut le voir sur la figure 5.10, des orbites brins $\langle \rangle$ des sommets sur lesquelles les forces s'appliquent. Cette solution peut être étendue à des faces plus complexes (avec plus de quatre sommets) et dans ces cas, il n'y pas de redondance. Elle est aussi compatible avec des faces triangulaires, mais inutile, parce qu'il existe déjà un ressort d'élongation qui lie les extrémités non connectées de deux arêtes adjacentes.

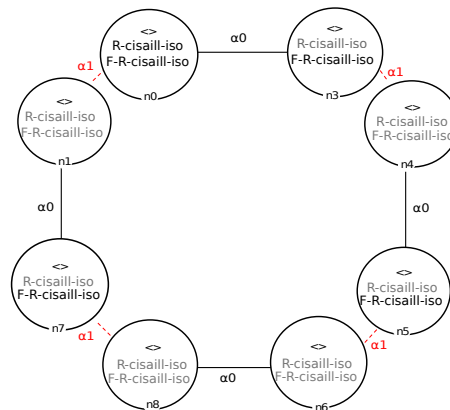


FIGURE 5.11 – Plongements associés aux ressorts de cisaillement surfacique isotrope.

Avec Jerboa, nous avons créé deux plongements pour chaque cas (isotrope et anisotrope). Pour le cas isotrope, le plongement appelé **R-cisaill-iso** correspond aux ressorts qui sont portés par les faces (orbites $\langle \alpha_0, \alpha_1 \rangle$). De même, comme montré sur la figure 5.11, nous avons créé avec l'outil Jerboa le plongement **F-R-cisaill-iso** associé aux orbites sommets de face $\langle \alpha_1 \rangle$ pour stocker les forces.

De la même manière, nous avons créé les plongements nécessaires pour les ressorts de cisaillement surfacique anisotrope comme montré sur la figure 5.12. Deux plongements sont introduits. Le plongement **R-cisaill-ani**, est porté par les orbites $\langle \alpha_1 \rangle$ et le plongement de stockage des forces appelé **F-R-cisaill-ani** est porté par les orbites brins $\langle \rangle$. Ici seuls les brins **n0** et **n3** portent ce plongement, car le motif vise à calculer les forces pour le seul ressort de cisaillement entre les sommets représentés par **n0** et **n3**. La particule associée au sommet représenté par **n1** (ou **n2**) ne subit pas de force de la part du ressort considéré.

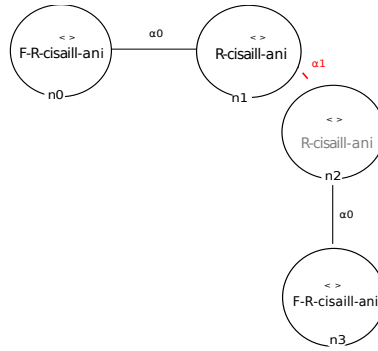


FIGURE 5.12 – Plongements associés à un ressort de cisaillement surfacique anisotrope.

5.2.2.3 Ressorts de courbure

Pour s’opposer aux flexions d’un tissu en 2D, Provot a introduit un ressort qui relie un sommet à son deuxième voisin [Pro95]. Ce type de ressorts peut également être ajouté à notre structure topologique. Néanmoins, il existe d’autres approches comme celles de [BW98, BMF03] qui ont proposé des ressorts angulaires qui contrôlent l’angle entre deux faces adjacentes pour gérer la courbure. Contrôler l’articulation entre deux faces par un ressort angulaire étant plus direct, nous proposons de présenter ce cas dans un premier temps, pour nous intéressons aux ressorts de Provot par la suite.

Ressorts angulaires

Les ressorts angulaires permettent de contrôler l’angle entre deux faces adjacentes. Plus précisément, la résistance à la flexion est principalement due à une limitation du débattement angulaire entre deux faces adjacentes. Si lors d’un changement topologique ultérieur, les deux faces se séparent, la résistance disparaît. Il paraît donc naturel d’associer ce ressort angulaire (raideur, amortissement) à l’arête commune des deux faces (orbite de type $\langle \alpha_0, \alpha_2 \rangle$), voir figure 5.13.

Comme le montre la figure 5.14, les forces s’appliquent sur plusieurs sommets qu’il faut retrouver en parcourant le sous-graphe bleu. Ce sont les sommets de l’arête ainsi que les sommets qui leur sont adjacents sur les deux faces (*i.e* les sommets *A*,

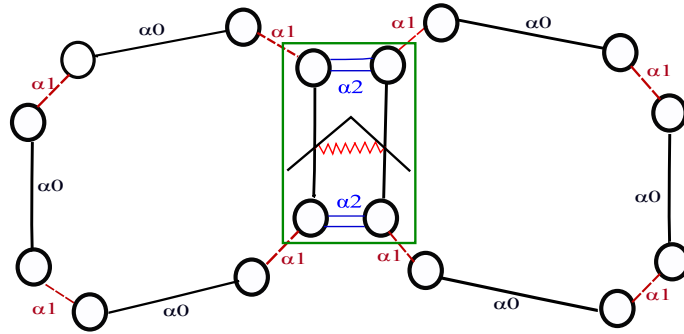


FIGURE 5.13 – Orbite support d'un ressort angulaire.

B, C, D, E et F). Une fois ces forces calculées, elles sont stockées dans les orbites brins $\langle \rangle$, car ces orbites sont l'intersection entre le sous-graphe de recherche (en bleu) et les orbites sommets (en rouge).

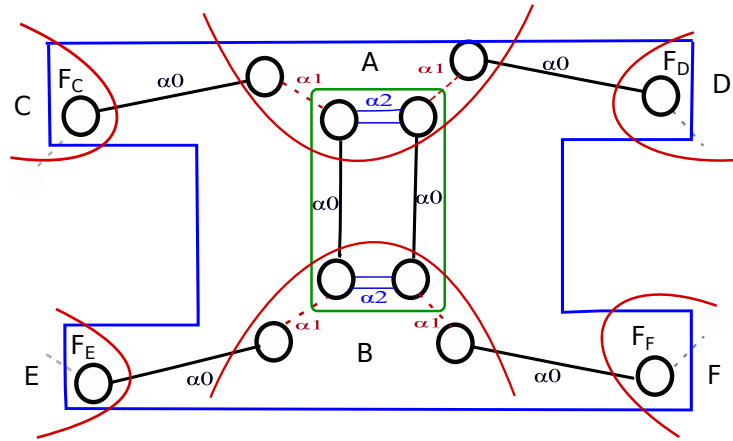


FIGURE 5.14 – Sous-graphe et orbites supports pour les ressorts angulaires.

Il faut noter que les forces de courbure subies par le sommet A (resp. le sommet B) pourraient être stockées sur l'orbite $\langle \alpha_2 \rangle$ représentant l'extrémité de l'arête AB , puisque c'est une orbite plus grande que l'orbite brin comprise dans l'intersection entre le motif et l'orbite sommet de A . Néanmoins, comme la force subie par A correspond à la somme des opposées des forces F_C et F_D (resp. F_E et F_F pour le sommet B), nous choisissons de stocker ces forces élémentaires $-F_C, -F_D, -F_E$ et $-F_F$ dans les sous-orbites brins $\langle \rangle$ respectives de l'arête AB . Notons que les sous-orbites $\langle \alpha_1 \rangle$, présentes également dans l'intersection entre le motif et les orbites des sommets A et B , ne répondent pas correctement à notre attente pour le stockage, car elles correspondent chacune à deux arêtes (dont elles forment l'articulation), ce qui ne permet pas de différencier les forces de courbure issues de chacune des arêtes. Nous les avons donc écartées. Au contraire, notre choix de stockage sur les orbites brins est spécifique à l'arête que nous considérons comme articulation, tout en faisant la distinction entre les forces issues de C, D, E et F . Ce choix a

également l'avantage de rendre homogène le choix du type d'orbite de stockage de toutes les forces générées par les ressorts angulaires.

Nous remarquons, par ailleurs, que le sous-graphe choisi est compatible avec d'autres types de faces. En effet, les liens entre les sommets C et E d'une part, et D et F , d'autre part, ne sont pas spécifiés dans le sous-graphe. En particulier, ces sommets peuvent être fusionnés. En d'autres termes, C et E peuvent devenir le même sommet et de même pour D et F , pour former des faces triangulaires. De même, ces sommets peuvent être liés par une arête et former des quadrilatères. S'ils sont liés par plus d'une arête, il s'agit d'autres types de faces. Ainsi, les maillages mixtes, avec différents types de faces, peuvent être utilisés avec le même sous-graphe et les mêmes orbites de stockage.

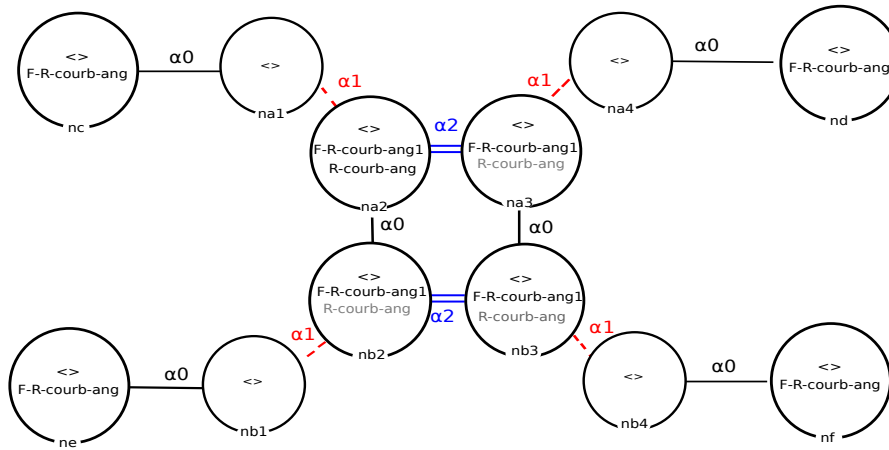


FIGURE 5.15 – Plongements associés à un ressort angulaire.

En utilisant Jerboa et comme montré sur la figure 5.15, nous avons créé trois plongements. Le premier, appelé **R-courb-ang**, stocke les propriétés du ressort et est porté par les orbites arêtes $\langle \alpha_0, \alpha_2 \rangle$ (représentées par les nœuds **na2**, **na3**, **nb2** et **nb3** sur la figure 5.15). Les deux autres stockent la valeur des forces appliquées et sont portés par les orbites brins $\langle \rangle$. L'un est appelé **F-R-courb-ang** et représente les forces subies par les sommets n'appartenant pas à l'arête de l'articulation (représentées par les nœuds **nc**, **nd**, **ne** et **nf** sur la figure 5.15), tandis que l'autre appelé **F-R-courb-ang1** stocke les forces spécifiquement pour les sommets de cette arête. Cette distinction est nécessaire afin d'éviter qu'un même plongement de force soit utilisé pour deux ressorts de courbure différents. Par exemple, dans le cas de la figure 5.14, la force $-F_C$ exercée sur A par l'arête AB , mais également une autre force exercée par l'arête BE sont stockées sur le même brin de l'orbite sommet de A et nécessitent donc deux plongements différents.

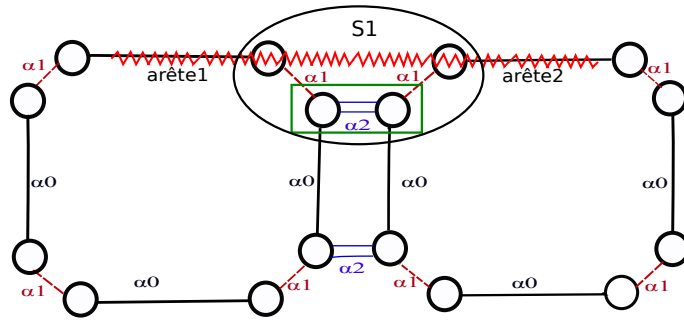


FIGURE 5.16 – Orbites de plongement d’un ressort linéaire de Provot.

Ressort linéaire de Provot

Le ressort de courbure linéaire de Provot décrit par la figure 5.16 peut être considéré comme un lieu qui contrôle l’angle entre les deux arêtes ($arête1$ et $arête2$). Ces deux arêtes sont adjacentes à une même arête qui lie les deux faces auxquelles appartiennent. Le sommet $S1$ relie ces deux arêtes. Il semble donc pratique de stocker ce type de ressorts dans les orbites $\langle \alpha_2 \rangle$, correspondant à la fois à l’arête commune et au sommet d’articulation, comme le montre la figure 5.16.

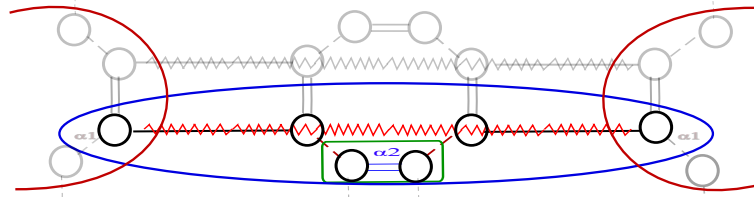


FIGURE 5.17 – Sous-graphe et orbites supports utilisés pour les ressorts linéaires de Provot.

Comme montré sur la figure 5.17, les faces n’ont pas besoin d’être représentées et le sous-graphe bleu relie deux arêtes connectées via un sommet. Par conséquent, nous pouvons appliquer cette modélisation à n’importe quel type de faces. Ce sous-graphe permet, à partir du lieu de stockage $\langle \alpha_2 \rangle$ de trouver les deux sommets entre lesquels un ressort de Provot est positionné. Les forces appliquées par ce ressort sont calculées et stockées dans les sous-orbites brins $\langle \rangle$ qui sont l’intersection du sous-graphe avec les orbites sommets (en rouge). À l’instar des ressorts angulaires, cette modélisation est compatible avec des triangles, même si les deux orbites $\langle \alpha_2 \rangle$ de l’arête de courbure portent des ressorts qui relient les mêmes deux sommets. Les maillages hétérogènes peuvent également bénéficier de ces ressorts.

En utilisant Jerboa et comme montré sur la figure 5.18, nous avons créé deux plongements : celui qui stocke les propriétés mécaniques du ressort, appelé **R-courb-Provot** et qui est porté par les orbites $\langle \alpha_2 \rangle$ et celui appelé **F-R-courb-Provot** qui stocke les forces et qui est porté par les orbites brins (représentées par les nœuds na et nb sur la figure 5.18).

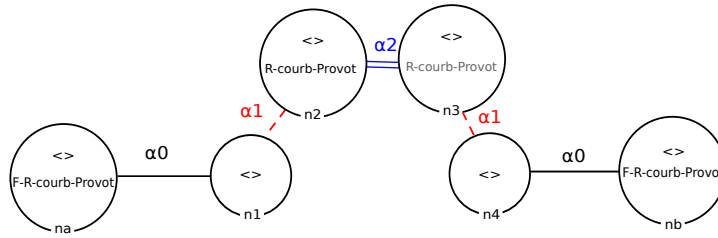


FIGURE 5.18 – Plongements associés à un ressort linéaire de Provot.

Les ressorts angulaires ou de Provot visent à simuler le même effet mais leurs implantations sont différentes. Ainsi, nous avons proposé dans notre modèle ces deux types de ressorts. La modélisation, que nous avons choisie pour les interactions en 2D, est conçue de façon générale. Grâce à cette généralité, les interactions peuvent être appliquées à tout type de maillages y compris les maillages hétérogènes. Néanmoins, l'utilisation de certains ressorts ne sont pas utiles à certains types de maillage même si la modélisation permet tout de même de les utiliser. Par exemple, l'utilisation des ressorts de cisaillement pour des mailles triangulaires, permet d'ajouter seulement de l'élongation à l'élément. Aussi, il n'est pas recommandé d'utiliser les ressorts de cisaillement pour les maillages triangulaires

5.2.3 Modèles mécaniques discrets 3D

La modélisation proposée pour les ressorts en 2D peut être étendue à la 3D. Les ressorts d'élongation correspondent aux arêtes des éléments du maillage comme en 2D. Les ressorts de cisaillement sont également utiles dans le cas 3D, plus précisément pour la simulation de maillages hexaédriques. Il est possible d'utiliser les ressorts de cisaillement sur les faces pour maintenir la forme d'un volume hexaédrique. Néanmoins, il existe aussi des ressorts de cisaillement en 3D qui sont portés par les quatre diagonales du volume comme montré sur la figure 5.19 et qui permettent de maintenir sa forme.

Pour stocker les forces appliquées par ce type de ressort, nous avons suivi la même méthode. En effet, comme ils sont à l'intérieur du volume et portés par ses diagonales, nous avons choisi de stocker leurs informations physiques dans les orbites volumes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$. L'existence de ces ressorts est ainsi conditionnée à l'existence du volume. L'intersection entre le sous-graphe associé au volume (en bleu) et les sommets d'application (en rouge) sont les sommets de volume (les orbites $\langle \alpha_1, \alpha_2 \rangle$) dans lesquels les forces peuvent être stockées. Comme dans le cas isotrope 2D, avec cette modélisation, les ressorts doivent partager leurs caractéristiques, donc leur longueur au repos et raideurs. Le volume doit donc être un parallélépipède rectangle.

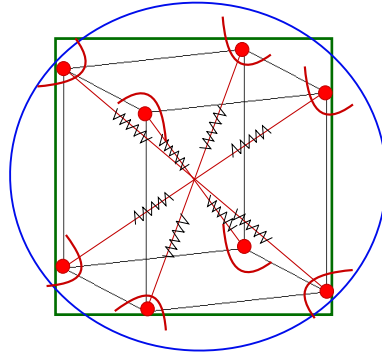


FIGURE 5.19 – Sous-graphe et orbites supports pour les ressorts de cisaillement volumique isotrope.

Il faut noter qu'il existe actuellement, un débat sur l'utilisation des ressorts de face ou de volume pour gérer le cisaillement. Dans notre approche, nous proposons les deux types de ressorts que l'on peut utiliser simultanément en considérant un ratio entre les raideurs comme étudié dans [Gol18].

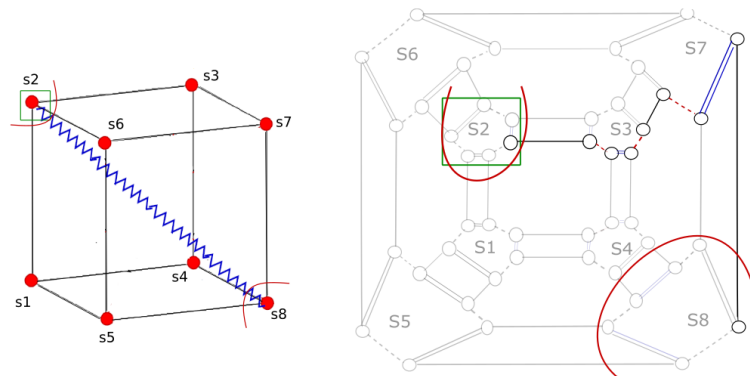


FIGURE 5.20 – Sous-graphe et orbites supports pour les ressorts de cisaillement volumique anisotrope.

Une approche anisotrope, permettant toute forme d'hexaèdre (*i.e.* non parallélépipède rectangle), peut également être envisagée. La figure 5.20 illustre le positionnement d'un ressort diagonal dans l'hexaèdre. On choisit de stocker le plongement sur le sommet de volume $\langle \alpha_1, \alpha_2 \rangle$ d'un des sommets de la diagonale. Un sommet n'appartenant qu'à une diagonale et le sommet opposé étant unique, il n'y a pas d'ambiguïté. Le graphe permettant de l'identifier consiste à construire un chemin de trois arêtes adjacentes qui lient les sommets opposés comme montré sur la figure 5.20. Les forces calculées sont alors stockées dans les orbites sommets de volume des particules correspondantes. Il faut néanmoins noter que l'on stocke le ressort sur l'un des sommets de la diagonale support, on constate donc deux possibilités pour le stockage d'un même ressort. Il faut remarquer qu'il aurait été possible d'adopter une solution similaire pour les ressorts des diagonales de face. Mais, dans le cas 2D,

nous avons une solution qui code l'interaction dans un coin de face, car cela correspond à la source de l'interaction (préserver l'angle). Malheureusement, en 3D, toute tentative de coder la diagonale de volume sur une orbite intermédiaire, symétrique du point de vue des sommets concernés, aboutit à des fortes redondances (plusieurs solutions de stockage pour une même diagonale) ou, au contraire, une ambiguïté (plusieurs diagonales pour une orbite de stockage des paramètres). En 3D, nous avons donc dû déroger à la règle de placement des interactions selon leur origine. Ce compromis ne pose pas de problème concernant la simulation de l'interaction selon la méthode que nous exposons dans le chapitre suivant.

Concernant les ressorts de courbure, à notre connaissance, ils sont rarement utilisés en 3D, ils permettent de piloter les angles dièdres entre faces de volumes adjacents. Or ces angles sont contrôlés indirectement par la forme des volumes donc par les ressorts d'élongation et de cisaillement. Nous ne proposons donc pas de modélisation des ressorts de courbure en 3D.

Pour conclure, pour passer d'un modèle 2D à un modèle 3D, nous avons uniquement ajouté les ressorts de cisaillement volumique. La table 6.1 présentée dans le chapitre suivant résume les différents types de ressorts implantés dans notre outil, les sous-graphes utilisés pour trouver les lieux de stockage de forces, les orbites dans lesquels les propriétés mécaniques sont plongées et les orbites de stockage des forces appliquées par chaque type de ressort.

5.2.4 Modèles mécaniques continus

Nous nous focalisons maintenant sur la modélisation des modèles de la mécanique continue. Nous allons suivre la même méthode décrite auparavant. En effet, il faut trouver tout d'abord les plongements mécaniques les plus adéquats et les lieux de stockage des forces qui en découlent. Tout d'abord, nous allons présenter la modélisation par éléments finis en petits déplacements, en élasticité linéaire. Ensuite, nous traitons la modélisation avec l'approche de co-rotationnel pour prendre en compte les rotations. Enfin, nous décrivons l'application de notre méthode sur la méthode masses/tenseurs.

La méthode des éléments finis linéaires

Comme vu dans la section 2.2.2.3 (page 19), la méthode des éléments finis se base sur un découpage en éléments (faces en 2D et volumes en 3D). Nous supposons, dans un premier temps, que les éléments sont des tétraèdres. Chaque tétraèdre est caractérisé par une matrice de raideur de taille (12×12) . Pour cela, nous avons stocké ces matrices dans les orbites volumes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$ (en vert sur la figure 5.21).

Les forces calculées sont stockées dans les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$ qui sont l'intersection entre les sous graphes du tétraèdre (en bleu) et ses sommets (en rouge).

Pour calculer les forces appliquées sur chaque sommet, on a besoin de calculer la matrice de rigidité $[K^{(e)}]$ et la stocker dans une phase préliminaire en chaque

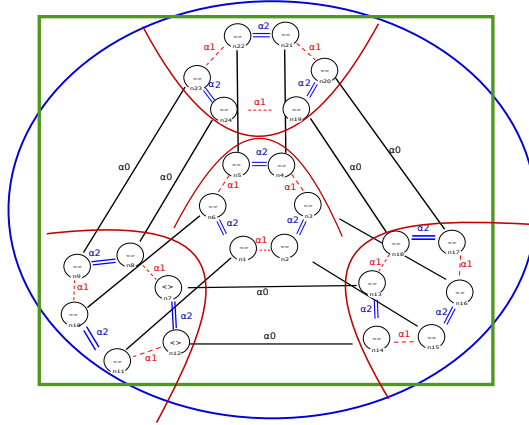


FIGURE 5.21 – Sous-graphe et orbites de plongement pour la méthode des éléments finis linéaires sur un tétraèdre.

élément de l'objet. Néanmoins cette matrice nécessite une numérotation au préalable des sommets de l'élément en garantissant que le volume du tétraèdre soit positif. Pour garantir la pérennité de cette numérotation, nous proposons de choisir un brin marqué par un plongement spécifique (comme montré sur la figure 5.22). Ce plongement est appelé **Premier-sommet** et il est de type booléen. Il est stocké sur les orbites brins $\langle \rangle$ et marque le premier sommet de l'élément. Il faut noter qu'avec les G-cartes un brin est associé à un et un seul volume. Le plongement **Premier-sommet** sur un brin désigne donc le premier sommet pour la numérotation de ce volume. À partir de ce brin et en suivant les relations d'adjacence des G-cartes dans un ordre fixé, le numéro de chacun des autres sommets est déterminé comme montré sur la figure 5.22. Par exemple, nous avons marqué le brin $s1$ (en bleu) avec ce plongement et donc le sommet auquel il appartient doit être le sommet numéro 1 de ce tétraèdre. Le second sommet est celui qui est lié au brin bleu avec une liaison α_0 . Il est désigné par $s2$ et est coloré en jaune. Le troisième sommet est atteint en parcourant à partir du brin bleu les liaisons α_1 puis α_0 (brin $s3$, également coloré en jaune). Le dernier sommet est celui qui est atteint en suivant les liaisons α_2 , α_1 et α_0 à partir du brin bleu (brin $s4$, également en jaune). Puisque le plongement est fixé une fois pour toute dans toute la phase d'initialisation, il nous garantit que les sommets sont systématiquement numérotés de la même façon dans un volume car le filtrage par le motif de la figure 5.22 est unique. Le brin porteur du plongement peut être choisi arbitrairement, mais nous devons garantir que le volume du tétraèdre obtenu soit positif. Rappelons ici qu'une G-carte n'est pas orientée. Si un volume négatif est trouvé, il convient d'utiliser un brin adjacent par α_1 ou α_0 au brin initialement testé.

En plus du plongement **Premier-sommet**, il faut définir d'autres plongements nécessaires pour calculer les forces appliquées par ce modèle. Ces plongements sont :

- Le plongement **M-raideur** qui permet de stocker la matrice de rigidité pour

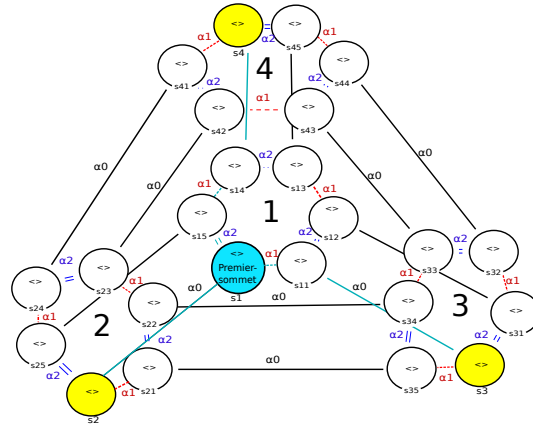


FIGURE 5.22 – Numérotation des sommets d’un tétraèdre.

chaque élément de l’objet. Il est porté par les orbites volumes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$. Il faut noter que le calcul de ce plongement se fait une seule fois dans une phase d’initialisation ;

- le plongement **F-sommet-MEF** qui permet de stocker les forces appliquées sur chaque sommet d’un tétraèdre. Il est porté par les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$.

Il faut noter que les éléments utilisés actuellement sont les triangles et les tétraèdres, mais une extension aux quadrilatères et aux hexaèdres ne pose pas de problème particulier avec notre méthode. Les plongements sont du même type mais avec un redimensionnement des matrices et vecteurs, seule la partie algorithmique est modifiée (par exemple, pour un hexaèdre, la matrice $[K^{(e)}]$ est de taille 24×24).

La méthode du Co-rotationnel

Comme le modèle d’élasticité linéaire en petits déplacements n’est pas compatible avec les rotations, nous avons également implémenté la méthode du co-rotationnel [MG04]. Cette approche, comme décrite dans l’état de l’art, est considérée comme une extension du modèle d’élasticité linéaire et permet de prendre en compte les rotations. Avec cette méthode, à partir du déplacement des sommets d’un tétraèdre nous extrayons un mouvement rigide (translation ou rotation) du tétraèdre, ce qui permet d’isoler la déformation.

Comme cette méthode se base sur la méthode des éléments finis, nous avons utilisé les mêmes plongements que précédemment. Il faut noter que seule la manière de calculer les forces appliquées sur les sommets de chaque tétraèdre est différente de celle de la méthode des éléments finis. C’est l’équation (2.33, page 32) qui est utilisée. Cependant, nous remarquons que cette formule dépend du terme $[R]$ qui désigne la matrice de rotation. Pour calculer cette matrice, nous avons besoin de la matrice inverse des positions initiales du tétraèdre, il faut alors prévoir un plongement appelé **MT-PO-Inv** qui la mémorise, car elle peut être calculée une fois pour toutes en début

de simulation. Il est porté par les orbites volumes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$. Cette matrice doit être calculée une seule fois dans la phase préliminaire avant la boucle de simulation. Les forces sont calculées et stockées dans les mêmes orbites que pour la méthode des éléments finis linéaires.

Pour conclure, nous avons utilisé les mêmes plongements créés pour la méthode des éléments finis linéaires, seul un plongement porté par les orbites volumes a été ajouté.

Masses/tenseurs

La méthode masses/tenseurs, décrite dans l'état de l'art, peut être considérée comme une optimisation de la méthode des éléments finis linéaires. Notre méthode générale est également adaptée à ce modèle mécanique. Comme proposé dans l'approche de Cotin *et al.* [CDA00], cette méthode est basée sur la matrice de rigidité calculée avec la méthode des éléments finis qui est une matrice 12×12 . Cette dernière doit être décomposée en seize sous-matrices de taille 3×3 . Ces matrices peuvent être subdivisées en deux types : celles qui décrivent l'influence des déplacements des sommets i sur les forces qu'ils subissent eux-mêmes ($[K_{ii}^{(e)}]$) et celles qui décrivent l'influence des déplacements d'un sommet j sur les forces subies par les autres sommets i de l'élément ($[K_{ij}^{(e)}]$).

Nous suivons la même méthode générale : il faut trouver tout d'abord les orbites les plus adéquates pour stocker ces sous-matrices. Une sous-matrice $[K_{ii}^{(e)}]$ caractérisant le comportement pour un sommet i pour un volume donné, ces sous-matrices sont stockées dans les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$ (voir figure 5.23(a)). Elles sont appelées *tenseurs de sommet*. Les autres sous-matrices $[K_{ij}^{(e)}]$ dépendent de deux sommets qui, dans un tétraèdre, sont nécessairement liés par une arête. Elles sont appelées *tenseurs d'arête*. Comme $[K_{ij}^{(e)}]$ diffère de $[K_{ji}^{(e)}]$ tout en correspondant à la même arête de volume $\langle \alpha_0, \alpha_2 \rangle$, les deux tenseurs doivent être stockés de façon asymétrique sur l'arête, en exploitant les sous-orbites. Nous avons décidé de stocker, par convention, les matrices $[K_{ij}^{(e)}]$ sur les orbites extrémités d'arête de volume $\langle \alpha_2 \rangle$ du sommet i , à savoir celui qui subit la force (respectivement $[K_{ji}^{(e)}]$ sur les orbites $\langle \alpha_2 \rangle$ du sommet j) comme montré sur la figure 5.23(b). Elles sont donc bien stockées sur des orbites différentes.

Il faut noter qu'il existe une autre solution qui consiste à stocker un seul tenseur $[K_{ij}^{(e)}]$ par arête ij , uniquement sur l'orbite $\langle \alpha_2 \rangle$ du sommet i . L'autre tenseur n'est pas stocké mais peut être retrouvé car il est son transposé. Cette dernière solution nécessite des règles spécifiques permettant d'exploiter une même matrice $[K_{ij}^{(e)}]$ pour la force en i et en j en transposant la matrice pour j ce qui n'apporte pas de gain en temps de calcul, mais permet de déduire le coût mémoire Aussi, nous avons choisi cette solution.

La numérotation des sommets qui a été introduite pour les éléments finis classiques, n'est plus nécessaire pour les masses/tenseurs, car les tenseurs sont directement plongés dans les orbites des entités auxquelles ils s'appliquent. La deuxième

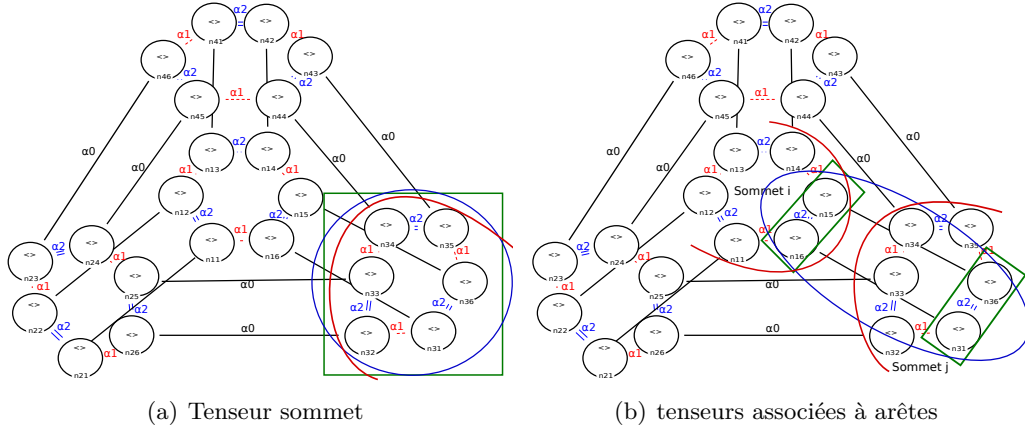


FIGURE 5.23 – Sous-graphe et orbites de plongement des tenseurs.

étape de notre méthode consiste à trouver le sous-graphe qui intersecte les sommets d'application. Nous utilisons deux sous-graphes différents : les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$ pour les tenseurs de sommet $[K_{ii}^{(e)}]$ et les arêtes de volume $\langle \alpha_0, \alpha_2 \rangle$ pour les tenseurs d'arête $[K_{ij}^{(e)}]$. Nous avons stocké les forces appliquées par les sommets sur eux-même dans les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$ et les forces appliquées par un sommet sur un autre sur les orbites extrémités d'arête de volume $\langle \alpha_2 \rangle$.

Pour récapituler, nous avons défini pour ce modèle, quatre plongements :

- Plongement **T-K-Sommet** pour stocker les sous-matrices $[K_{ii}^{(e)}]$ qui sont propres aux sommets de chaque tétraèdre, et qui sont stockés sur les sommets de volume (orbites $\langle \alpha_1, \alpha_2 \rangle$);
- Plongement **T-K-Arête** pour stocker les autres sous-matrices $[K_{ij}^{(e)}]$ dans les orbites extrémités d'arête de volume (orbites $\langle \alpha_2 \rangle$);
- Plongements **F-sommet-MT** et **F-arête-MT** pour stocker les forces générées et que l'on stocke sur les mêmes orbites que les tenseurs dont elles sont issues.

De même que pour la méthode des éléments finis linéaires, on peut exploiter le modèle masses/tenseurs avec des mailles hexaédriques. La matrice de raideur de taille 24×24 se décompose alors en 64 tenseurs qui se répartissent en 8 tenseurs de sommet, 24 tenseurs d'arête, 24 tenseurs de diagonale de face, et 8 tenseurs de diagonale de volume. Les tenseurs de sommets et les tenseurs d'arêtes se traitent comme dans le cas des tétraèdres (respectivement orbites $\langle \alpha_1, \alpha_2 \rangle$ et orbites $\langle \alpha_2 \rangle$ du côté du sommet subissant la force). Concernant les tenseurs de diagonales (de faces ou de volumes), nous nous inspirons des plongements proposés pour les ressorts en s'assurant néanmoins de supprimer la symétrie entre les sommets concernés. Aussi, les tenseurs de diagonale de face peuvent être stockés sur les orbites sommets de face de volume $\langle \alpha_1 \rangle$, nous choisissons ici le sommet correspondant au sommet

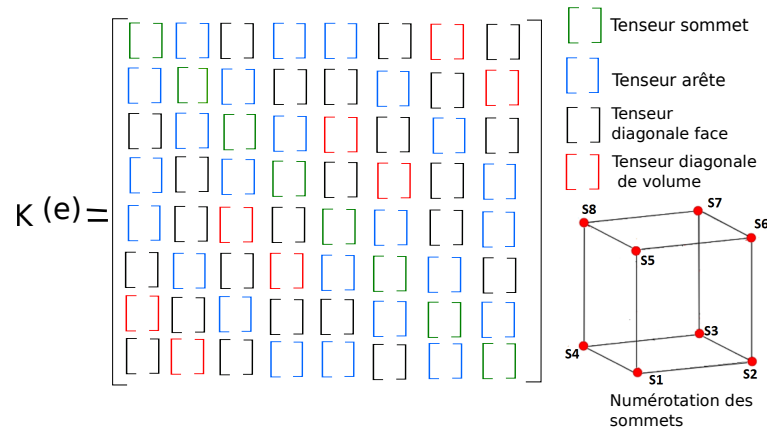


FIGURE 5.24 – Décomposition d’une matrice de raideur dans le cas des hexaèdres.

subissant l’interaction. Les tenseurs de diagonale de volume peuvent être stockés sur les orbites sommet de volume $\langle \alpha_1, \alpha_2 \rangle$ des particules subissant l’interaction. Dans ces deux derniers cas, il suffit de suivre un chemin d’arêtes pour trouver le sommet opposé le long de la diagonale (ce qui nous donne le sous-graphe de chaque interaction) et de calculer l’action exercée par ce sommet. Les forces calculées sont stockées dans les mêmes orbites que les tenseurs. Notons ici que, contrairement au cas des ressorts en 3D, aucune redondance n’est constatée car les tenseurs sont orientés sur les arêtes et les diagonales. Chaque tenseur ne possède donc qu’un seul choix d’orbites de stockage.

La méthode des éléments finis en 2D

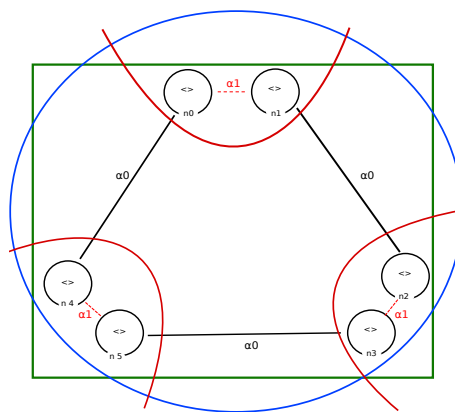


FIGURE 5.25 – Orbites de stockage pour le co-rotationnel en 2D.

L’approche du co-rotationnel offre une solution élégante à l’implantation des méthodes des éléments finis en 2D. Dans cette dimension, les calculs sont à effectuer en 2D, par exemple dans le plan (x, y) . Il faut donc faire en sorte d’amener chaque élément surfacique dans ce plan par une rotation. Après cette transformation, le

calcul des forces est similaire a celui du tétraèdre, mais avec un nombre de sommets valant 3.

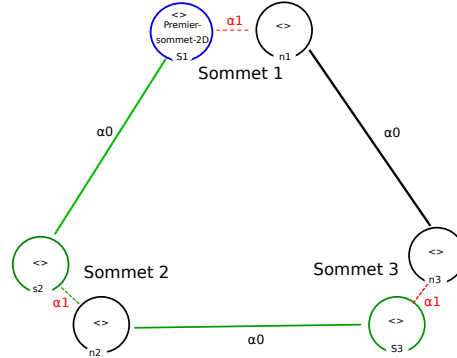


FIGURE 5.26 – Identification de l'ordre des sommets en 2D.

Comme indiqué par la figure 5.25, les matrices de raideur sont stockées dans les orbites faces $\langle \alpha_0, \alpha_1 \rangle$ (en vert), les sommets d'application sont les sommets du triangle (en rouge) et le sous-graphe (bleu) pour trouver ces sommets est le triangle lui même. Le lieu de stockage des forces est l'intersection entre les deux sous-graphes bleu et rouge, donc les orbites sommets de face $\langle \alpha_1 \rangle$. Pour cela, nous avons défini deux plongements nécessaires pour calculer les forces appliquées par ce modèle. Ces plongements sont :

- le plongement **M-rigid-2D** pour stocker la matrice de raideur pour chaque face de l'objet. Le calcul de ce plongement se fait une seule fois dans une phase d'initialisation ;
- le plongement **F-sommet-MEF2D** pour stocker les forces appliquées sur chaque sommet de face. Il est porté par les orbites sommets de faces $\langle \alpha_1 \rangle$.

Il faut aussi noter que, comme pour la méthode des éléments finis et le co-rotationnel en 3D, il faut ajouter un plongement sur un brin en chaque face qui permet d'identifier le premier sommet du triangle. Ce plongement est stocké dans les orbites brins $\langle \rangle$ sous la forme d'un booléen qui marque le premier sommet. Les sommets 2 et 3 sont obtenus par filtrage comme montré sur la figure 5.26. Par exemple, dans la figure 5.26, le sommet numéro 1 est celui auquel appartient le nœud (en bleu) marqué par le plongement **Premier-sommet-2D**. Le second sommet est celui qui est lié au nœud bleu avec une liaison α_0 . Le troisième sommet est atteint en parcourant à partir du nœud bleu les liaisons α_1 puis α_0 .

Les différents plongements nécessaires à l'implantation des différentes méthodes basées sur la mécanique continue sont résumés dans la table 6.2 présentée dans le chapitre suivant.

5.3 Adaptation de la modélisation pour les cartes orientées

Dans ce manuscrit, nous avons fait le choix de travailler avec le modèle topologique des G-cartes. Comme expliqué dans l'état de l'art, ce choix est principalement motivé par l'ensemble des orbites existants dans les G-cartes et l'atomicité de ces orbites. Néanmoins, compte tenu du fait que les cartes orientées (version duale) ont régulièrement été utilisées pour la simulation physique de corps maillés, il est opportun d'étudier comment la démarche que nous avons proposée peut être transposée à ces modèles. En réalité, les cartes orientées étant plus compactes en mémoire et proposant potentiellement moins d'orbites que les G-cartes, le principal challenge est de trouver des orbites de stockage qui permettent aux interactions d'identifier de façon non ambiguë les sommets d'application et où plusieurs forces de même nature mais de sources différentes ne soient pas à stocker dans le même plongement. Sans atteindre l'exhaustivité des différents cas d'interaction exposés dans ce chapitre, nous donnons néanmoins ici quelques pistes.

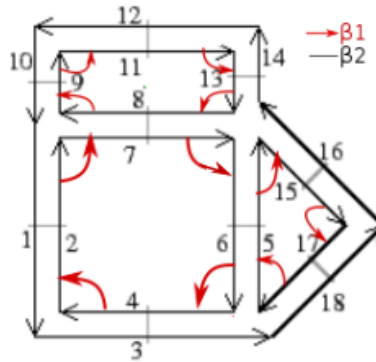


FIGURE 5.27 – Objet représenté avec une carte orientée.

Tout d'abord, pour certaines interactions, la transposition aux cartes orientées est assez immédiate. Typiquement, nous avons proposé que les ressorts d'élongation élémentaires soient stockés sur les arêtes de faces. Ces orbites ont un équivalent pour les cartes orientées duales, il s'agit du brin. De même, concernant les ressorts de cisaillement surfacique isotrope stockés, dans les G-cartes, dans les faces, les orbites $\langle \beta_1 \rangle$ peuvent être utilisées (voir figure 5.27). Quant aux ressorts de courbure angulaires, ils sont portés par les orbites arêtes (en 2D) ce qui correspond aux orbites $\langle \beta_2 \rangle$ des cartes orientées. Citons enfin un dernier exemple, celui des ressorts de cisaillement volumique isotrope qui sont portés par les volumes, donc les orbites $\langle \beta_1, \beta_2 \rangle$. Plusieurs interactions ou plongements de stockage de force se basent sur les sommets de volume. Ces plongements existent également dans les cartes orientées, mais nécessitent une composition de liaisons : $\langle \beta_1 \circ \beta_2 \rangle$.

Malheureusement, d'autres orbites, que nous avons utilisées dans les G-cartes, n'ont pas de correspondance directe en cartes orientées. Il faut alors trouver des adaptations pour gérer ces plongements. Prenons l'exemple du ressort de courbure

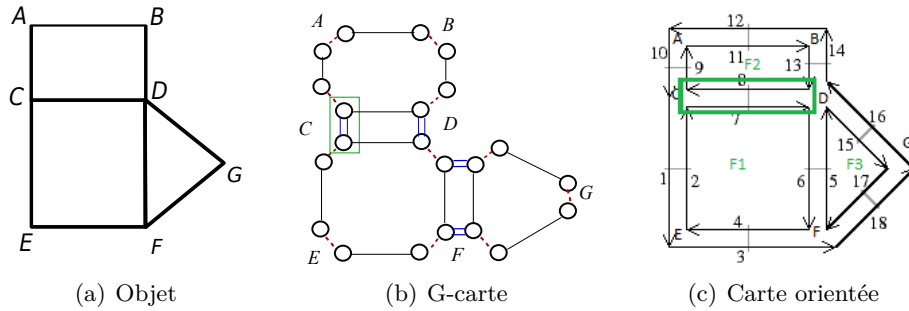


FIGURE 5.28 – Modélisation des ressorts linéaires de Provot

linéaire de Provot placé entre les sommets A et E de la figure 5.28(a) et qui contrôle la courbure le long de l'arête CD . Dans une G-carte, nous avons choisi d'enregistrer les caractéristiques du ressort dans l'extrémité C de l'arête CD (orbite $\langle \alpha_2 \rangle$), comme le rappelle la figure 5.28(b). Cette orbite n'a pas de correspondance directe en cartes orientées, car l'origine des arêtes est alors représentée par les brins eux-mêmes, comme le montre la figure 5.28(c).

Sur cette figure, le ressort de Provot entre les sommets A et E est dû à l'articulation entre les faces $F1$ et $F2$, il est donc prévu dans notre méthode de stocker ce ressort sur l'arête formée par les brins 7 et 8. C'est néanmoins cette même arête qui sert à stocker un ressort de Provot entre les sommets B et F . Pour lever cette ambiguïté, sachant que les cartes orientées ne proposent pas l'orbite extrémité d'arête, nous proposons la convention suivante. Le brin 7, correspondant au sommet C intermédiaire entre A et E , nous choisissons d'y stocker l'interaction qui les lie. Le brin 8 peut, de la même manière, stocker l'interaction entre B et F .

De même, nous proposons une convention analogue pour le stockage des tenseurs, puisque ces plongements sont également à stocker aux extrémités d'arête. Dans notre exemple, le tenseur exprimant l'action du sommet C sur D par la face $F1$ peut être stocké dans le brin 7. Malheureusement, le tenseur exprimant l'action dans l'autre sens, du sommet D sur le sommet C ne peut pas être stocké. En effet, le brin 8, l'autre brin de l'arête, est utilisé pour stocker l'influence de D sur C par la face $F2$ et non la face $F1$. Il existe alors deux possibilités : soit nous cumulons les tenseurs des faces incidentes à une même arête (par exemple, le brin 7 cumule la contribution de $F1$ et $F2$ aux actions de C sur D), mais nous n'avons plus l'information atomique, ce qui peut compliquer la prise en charge des modifications topologiques, soit nous nous rappelons que les tenseurs entre sommets sont transposés l'un de l'autre et qu'il suffit de n'en stocker qu'un seul (l'action de D sur C s'obtient avec le transposé du tenseur stocké dans le brin 7).

Finalement, la principale limitation apparaît lorsqu'il s'agit de stocker les forces élémentaires dans les orbites brins des G-cartes (cas des ressorts d'élongation élémentaires, cas des ressorts de Provot, etc.) Aucune orbite ne leur est équivalente dans les cartes orientées, puisqu'un brin des cartes orientées regroupe deux brins des G-cartes. Il convient alors d'utiliser deux plongements différents, l'un pour mémoriser les informations de l'origine du brin, et l'autre pour mémoriser les informations

de l'autre extrémité du brin. Nous renonçons alors à une propriété importante, que nous avons exploitée dans le cadre des G-cartes : avec les cartes orientées, les forces appliquées à un sommet ne sont plus forcément stockées dans des sous-orbités du sommet.

Pour résumer, nous constatons que nous pouvons utiliser un modèle de cartes orientées en se basant sur certaines conventions voire des compromis. Par contre, le fait de choisir les cartes orientées nous oblige à travailler uniquement avec des objets globalement orientés. Or, les lois de la mécanique n'exigent pas une telle propriété d'orientation globale. Seule une orientation locale est nécessaire : par exemple, s'assurer que deux faces adjacentes ont la même orientation lorsqu'on traite leur articulation. Avec les G-cartes, il est possible de garantir une orientation locale, sans recourir à une orientation globale. Ainsi, il est possible de simuler mécaniquement une bande de Möbius avec des G-cartes, mais cela n'est pas possible, en 2D, avec des cartes orientées. Convenons cependant que la majorité des objets à simuler sont orientables (la bande de Möbius et les autres objets construits sur le même principe étant les seules exceptions).

5.4 conclusion

Dans ce chapitre, nous avons décrit les hypothèses utilisées dans notre travail pour simuler des différents objets maillés. Nous avons également présenté la méthodologie générale qui nous a permis de modéliser tout type d'interaction que l'on veut inclure dans notre plate-forme. L'utilisation de cette méthode nous garantit que toutes les informations nécessaires à la simulation peuvent être stockées dans le modèle topologique sans recours à des structures externes. Nous avons présenté la modélisation de chaque type d'interaction par un plongement, qui est porté par un type d'orbite des G-cartes, un sous-graphe dont le but est d'identifier les sommets d'application et des orbités pour stocker les forces calculées. Finalement nous avons montré que notre méthodologie peut être appliquée, moyennant quelques compromis, à d'autres modèles topologiques comme les cartes orientées.

Une fois les différents plongements définis dans l'objet maillé, nous pouvons procéder à la simulation de cet objet en utilisant le langage à base de règles de Jerboa.

Simulation avec le langage à base de règles Jerboa

Sommaire

6.1	Introduction	131
6.2	Initialisation du modèle	132
6.2.1	Utilisation d'un maillage décrit dans un fichier	132
6.2.2	Création d'un maillage	140
6.3	Boucle de simulation	146
6.3.1	Règles sur la composante connexe vs règle sur un motif	147
6.3.2	Règles de calcul des forces	148
6.3.3	Résolution des équations du mouvement	164
6.4	Optimisation des calculs	165
6.4.1	Recensement des redondances	166
6.4.2	Analyse des redondances	168
6.4.3	Mise en œuvre des optimisations	170
6.5	Résultats	173
6.6	Conclusion	177

6.1 Introduction

Dans le chapitre précédent, nous avons présenté la méthode de modélisation générale utilisée pour trouver les orbites les plus adéquates afin de stocker toutes les propriétés mécaniques des différentes interactions dans une G-carte. Nous avons présenté la modélisation de chaque type d'interactions. Cette modélisation nous a permis de créer un modèle de prototypage général et flexible pour la simulation physique. Dans le présent chapitre, nous présentons les étapes de la boucle de simulation physique en utilisant le langage à base de règles Jerboa. Pour cela, nous présentons tout d'abord la phase d'initialisation des objets maillés. Ensuite, nous rappelons les étapes de la boucle de simulation, présentons son implantation avec le langage à base de règles et introduisons les règles créées pour calculer les forces appliquées par les différents types d'interactions. Finalement, nous concluons ce chapitre avec quelques solutions d'optimisation (en vue d'avoir des simulations plus performantes en temps de calcul) et quelques résultats de simulation.

6.2 Initialisation du modèle

Pour exploiter la modélisation proposée dans le chapitre précédent, nous devons, à partir d'un objet maillé, procéder au peuplement du maillage par les propriétés mécaniques de l'objet (raideur et viscosité de chaque type de ressort, calcul des matrices pour les méthodes des éléments finis...) et initialiser les forces appliquées par chaque interaction à zero. Nous avons utilisé différents types d'objets maillés. Ces derniers peuvent être créés avec le langage à base de règles ou bien chargés directement à partir de fichiers de maillage surfacique ou volumique. Nous commençons par cette dernière stratégie, car elle est immédiate. Nous présentons ensuite un exemple d'objet mécanique construit grâce au langage à base de règles Jerboa.

6.2.1 Utilisation d'un maillage décrit dans un fichier

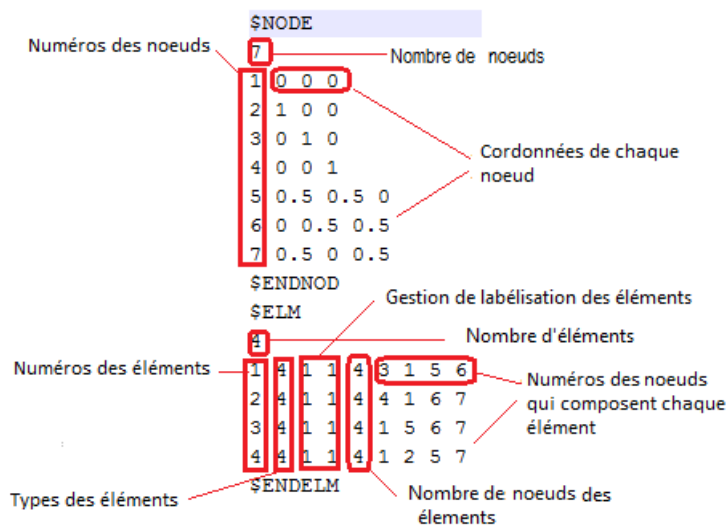


FIGURE 6.1 – Exemple de fichier GMSH.

L'outil Jerboa contient un chargeur d'objets surfaciques/volumiques via des formats comme OBJ, OFF, gmsh... Pour les objets 3D, nous avons choisi de travailler avec le format gmsh [GR09] car il est plus adapté à la simulation physique. Un exemple de fichier gmsh qui représente un objet composé de quatre tétraèdres liés les uns aux autres est donné sur la figure 6.1. Il faut noter que, même si l'objet est chargé à partir d'un fichier, celui-ci peut être ensuite modifié, par exemple en le subdivisant à l'aide d'une règle écrite avec Jerboa comme décrit dans la section suivante.

Après le chargement du fichier, l'étape suivante consiste à peupler le maillage géométrique avec les propriétés mécaniques de l'objet à simuler contenues dans un

fichier XML. Pour cela, nous créons des règles qui s'appliquent sur toute la composante connexe car elles visent à calculer des plongements sur tout l'objet. Ces règles ont la forme générale présentée sur la figure 6.2. Cette règle consiste à parcourir une composante connexe et à traiter toutes les occurrences des plongements cités dans le membre gauche (ici, initialiser les valeurs de surface et volume), quelle que soit leur orbite support.

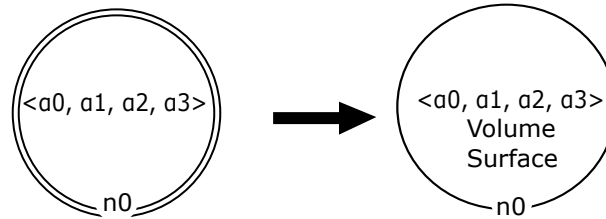


FIGURE 6.2 – Règle générale du peuplement.

La première étape de la phase d'initialisation, quel que soit le modèle mécanique, est l'initialisation du plongement *Masse-coin*. Pour calculer ce plongement, nous avons créé deux règles de la même forme que celle présentée sur la figure 6.2, une pour permettre de calculer tout d'abord le volume/la surface de l'élément et les stocker dans les plongements *Volume* et *Surface* portés respectivement par les orbites volumes $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$ et les orbites faces $\langle \alpha_0, \alpha_1 \rangle$. Ensuite, une seconde règle est créée pour calculer la masse en multipliant la masse volumique (resp. surfacique), stockée dans le fichier XML, par le volume (resp. la surface) de chaque élément et la répartir entre ses coins.

Dans une seconde étape, il faut initialiser les interactions. Cette étape dépend du type de modèle mécanique choisi. Nous décrivons, dans la suite, les plongements à initialiser pour chaque modèle.

Initialisation du modèle masses/ressorts

Pour initialiser les différents types des ressorts, nous avons utilisé une seule règle qui énumère tous les plongements des ressorts que l'on souhaite utiliser. Cette règle s'applique sur la composante connexe comme celle présentée sur la figure 6.2. Dans cette règle, nous avons initialisé les différents types de ressorts (*i.e.* les ressorts d'élongation, les ressorts de cisaillement surfacique isotrope et anisotrope, les ressorts de courbure angulaires et linéaires de Provot, et les ressorts de cisaillement volumique). Pour chaque ressort, il faut initialiser sa raideur, sa longueur au repos et l'amortissement. Pour cela, nous récupérons les valeurs de raideur et d'amortissement à partir du fichier XML, et nous calculons les longueurs au repos.

Lorsque le ressort est porté par une arête, comme dans le cas des ressorts d'élongation, Jerboa offre des méthodes pour effectuer facilement le calcul (voir la section 3.4.1.2, page 76). Il est ainsi facile de collecter les positions des deux sommets de l'arête courante, puis de calculer la distance qui les sépare.

Dans les autres cas, les ressorts ne sont pas portés par une arête. Il n'est donc

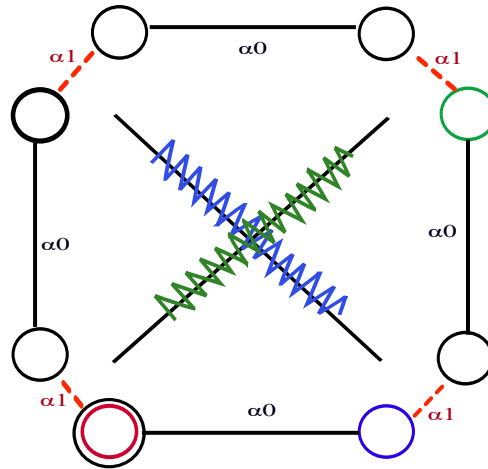


FIGURE 6.3 – Calcul des longueurs des ressorts de cisaillement.

pas possible de déduire directement la longueur du ressort à partir de son orbite support. Il convient donc d'extraire chaque sommet du ressort en parcourant les liaisons de la G-carte. Par exemple, les ressorts de cisaillement surfacique sont portés par les orbites faces en 2D (faces de volume en 3D) $\langle \alpha_0, \alpha_1 \rangle$, comme le rappelle la figure 6.3. À partir du brin représentant la face (en rouge sur la figure 6.3), il est possible d'accéder au sommet opposé (en vert sur la figure 6.3) à l'aide des liaisons successives α_0, α_1 et α_0 et, ainsi, calculer la longueur du ressort vert. Par hypothèse, la face est rectangulaire et le ressort bleu doit avoir la même longueur. Ce qui peut aisément être vérifié en accédant successivement aux brins bleu et noir par la même méthode des liaisons de voisinage.

Nous avons, également, initialisé dans cette même règle, tous les plongements associés aux forces appliquées par chaque type de ressort à zéro.

Initialisation des modèles des éléments finis linéaire 3D

Dans le cas de la simulation d'un objet représenté par un maillage tétraédrique chargé avec un modèle basé sur la mécanique continue (la méthode des éléments finis linéaires, co-rotationnel ou masses/tenseurs), il faut initialiser la matrice de rigidité pour chaque élément de l'objet. Dans une première étape, on s'occupe de la numérotation des sommets pour calculer cette matrice de rigidité. Pour cela, nous avons créé la règle présentée sur la figure 6.4. Il faut noter que pour le modèle masses/tenseurs, une fois les tenseurs positionnés nous n'avons plus besoin de la numérotation.

Dans cette règle, le membre gauche (en haut de la figure 6.4) filtre explicitement un tétraèdre (avec tous ses brins) et le membre droit (en bas de la figure 6.4), marque le premier sommet. En pré-traitement de la règle (voir la section 3.4.1.3, page 78 pour une présentation de la phase de pré-traitement (extra-parameter) des règles),

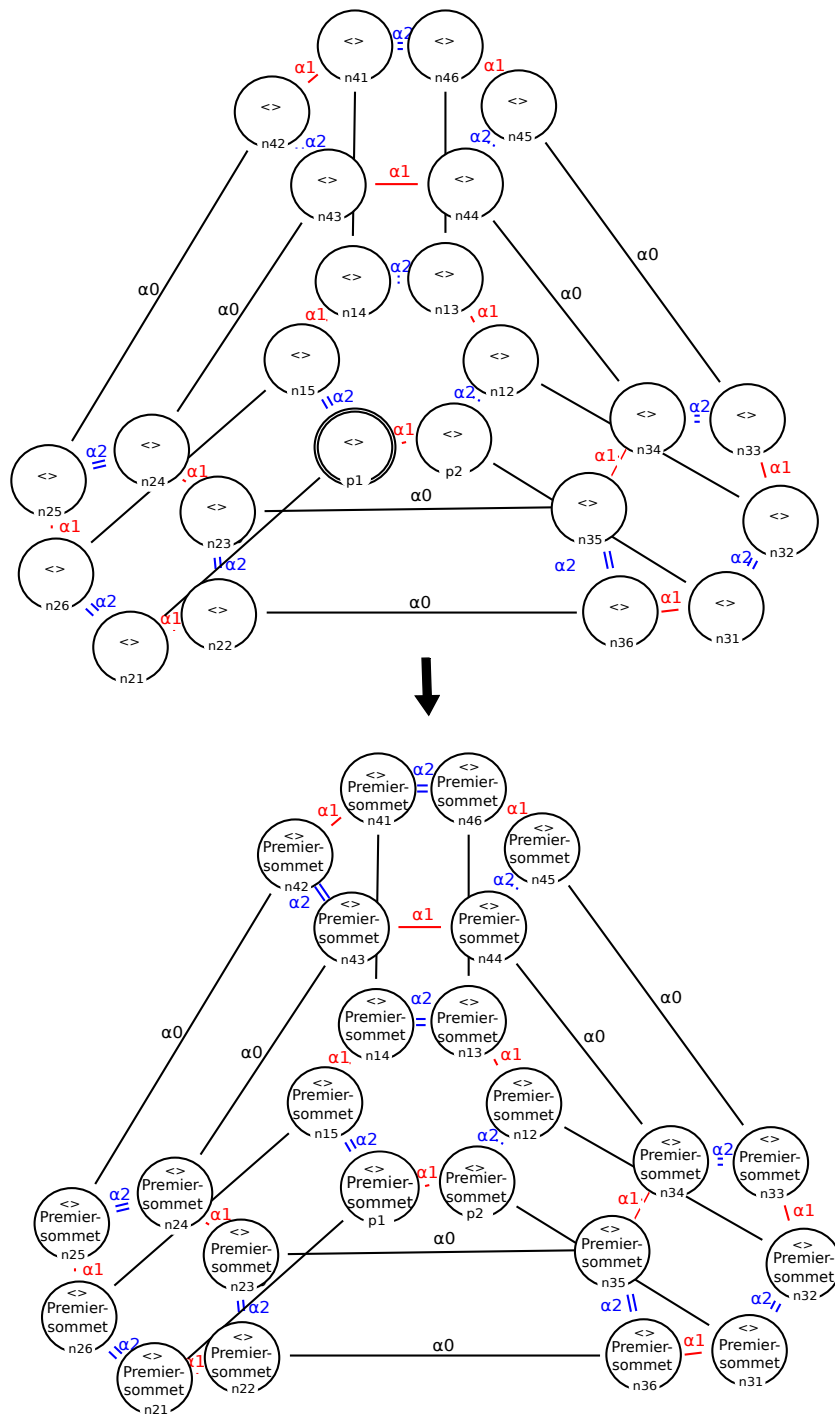


FIGURE 6.4 – Règle de numérotation des sommets d'un élément 3D.

la règle calcule le volume du tétraèdre. Si le volume est positif le nœud d'accroche **p1** est marqué comme premier sommet (le plongement **Premier-sommet** est mis à *true* pour **p1** et à *false* pour les autres brins). Dans le cas contraire (si le volume est négatif), c'est le nœud **p2** qui est marqué comme premier sommet, son volume sera alors bien positif. Ceci nous garantit d'avoir un et un seul nœud marqué avec la valeur *true* par tétraèdre. Ensuite, en suivant l'ordre de la numérotation de nœuds décrit sur la figure 5.22 (page 122), nous pouvons en déduire le numéro de chaque sommet. Pour appliquer cette règle sur tous les tétraèdres de l'objet, nous avons créé un script qui parcourt tous les tétraèdres et applique la règle sur chacun d'eux.

Maintenant que les premiers sommets sont marqués, nous procédons au calcul de la matrice de rigidité. Pour cela, nous avons créé la règle 6.5. Nous ne l'appliquons que sur un brin d'accroche **s1** marqué comme premier sommet. Puis pour calculer les valeurs de cette matrice, qui est propre à chaque élément, nous utilisons la formule suivante :

$$[K^{(e)}] = v[B^{(e)}]^T[C][B^{(e)}] \quad (6.1)$$

avec v le volume du tétraèdre au repos, $[C]$ la matrice de comportement de taille 6×6 qui dépend du coefficient de Poisson et du module de Young et qui est la même pour tous les tétraèdres qui constituent l'objet et $[B]$ la matrice de déformation de taille 12×6 qui est calculée en utilisant les positions au repos des sommets de chaque tétraèdre.

La matrice de comportement $[C]$ qui est la même pour tout l'objet, elle est calculée par une règle qui a le même format que la règle 6.2 et qui s'applique sur la composante connexe. Cette matrice a la forme définie par la formule (2.17), page 15, et est calculée en utilisant les coefficients de Lamé ou le module de Young et le coefficient de Poisson indiqués dans le fichier XML. Ensuite, en chaque élément, la matrice de déformation $[B]$ est calculée en pré-traitement de la règle 6.5 à partir du tenseur linéaire de Cauchy défini dans la section 2.2.1.2 (page 11). Finalement, la matrice $[K^{(e)}]$ est calculée et stockée dans le plongement **M-raideur**.

Initialisations du modèle co-rotationnel 3D

La phase d'initialisation pour la méthode du co-rotationnel, est similaire à la méthode des éléments finis linéaires décrite précédemment. Tout d'abord, le premier sommet de chaque élément est marqué, ensuite la matrice de raideur est calculée. Néanmoins, pour ce modèle, nous avons besoin de calculer la matrice de rotation qui dépend de la matrice inverse des positions initiales du tétraèdre. Nous avons créé une règle semblable à celle présentée sur la figure 6.5, qui est itérée sur tous les éléments de l'objet et calcule cette matrice pour chaque élément. Cette règle s'applique également à partir des brins marqués **Premier-sommet**. Pour calculer cette matrice, dans une première étape, nous récupérons le vecteur position de chaque sommet à partir de son brin représentant (voir la figure 5.22, page 122). Dans une seconde étape, nous calculons l'inverse de la matrice constituée de ces

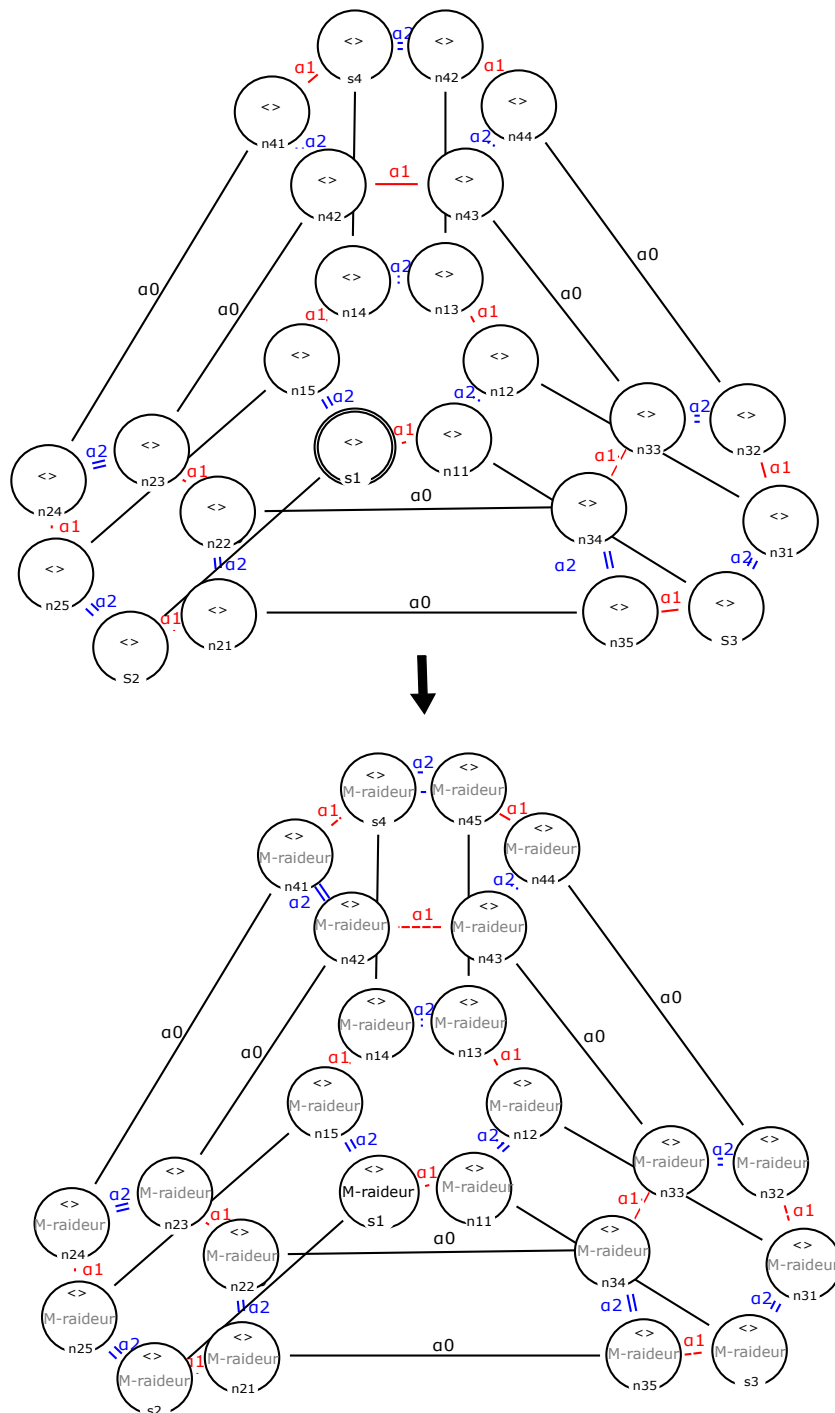


FIGURE 6.5 – Règle de calcul de matrice de rigidité pour un tétraèdre.

quatre vecteurs positions (exprimées en coordonnées homogènes) et la stockons dans le plongement `MT-PO-Inv`.

Initialisations du modèle masses/tenseurs

En ce qui concerne le modèle masses/tenseurs, il faut initialiser les divers tenseurs des sommets et d'arêtes dans le cas des tétraèdres. Pour cela, comme décrit dans l'état de l'art, nous allons utiliser les matrices de rigidité déjà calculées pour la méthode des éléments finis linéaires en chaque élément de l'objet et les décomposer chacune en seize sous-matrices de taille 3×3 .

Nous avons créé la règle présentée sur la figure 6.6. Son membre gauche (présenté en haut de la figure) filtre un tétraèdre. Le pré-traitement de la règle récupère la matrice de raideur générale associée à cet élément. Le membre droit (présenté en bas de la même figure) affecte tous les tenseurs aux sommets et arêtes correspondants. Pour cela nous avons extrait à partir de la matrice générale les quatre tenseurs de sommets et les six tenseurs d'arêtes (les six autres tenseurs d'arêtes, ne sont pas pris en compte car ils sont les transposés des six premiers). Finalement nous associons ces tenseurs aux plongements des sommets `T-K-Sommet` et à ceux des extrémités d'arêtes `T-K-Arete` qui leur correspondent. Cette règle s'applique sur un seul élément. Elle est donc appliquée sur tous les éléments de l'objet à l'aide d'un script. Le marquage du premier sommet et donc la numérotation qui en découle ainsi que la matrice de raideur ne sont plus utilisés durant la phase de simulation.

Initialisation du modèle co-rotationnel 2D

Cette méthode est similaire au co-rotationnel en 3D. On suppose néanmoins ici que l'objet est décrit dans un plan, par exemple (x, y) et que les positions peuvent s'exprimer en 2D. Pour calculer la matrice de raideurs, il faut tout d'abord numérotter, dans une phase initiale, les sommets du triangle. Pour cela, nous avons créé la règle présentée sur la figure 6.7 qui permet de marquer un premier sommet et de le stocker dans le plongement `Premier-sommet-2D` porté par les orbites brins.

Dans cette règle, nous filtrons dans son membre gauche un triangle et dans le membre droit, nous calculons le plongement `Premier-sommet-2D` (de type booléen) en fonction du signe de l'aire de la face. Si la valeur de l'aire est positive (orientation trigonométrique des sommets dans le plan), le nœud d'accroche `p1` est alors marqué comme premier sommet. Dans le cas contraire, c'est le nœud `p2` qui est marqué comme premier sommet. Il faut noter que ce concept d'aire positive n'a plus de signification une fois le triangle positionné dans l'espace 3D. Néanmoins, la numérotation choisie nous permet de fixer le sens de la normale de la face. Cette normale sera exploitée dans la phase de simulation, car elle permet de déterminer quelle rotation le triangle doit subir pour être ramené dans le plan (x, y) avec la même orientation que dans la phase d'initialisation. Il faut également créer un script pour appliquer cette règle sur tout l'objet. Maintenant, que les sommets sont numérotés, nous pouvons calculer la matrice de rigidité pour chaque triangle de l'objet. Comme

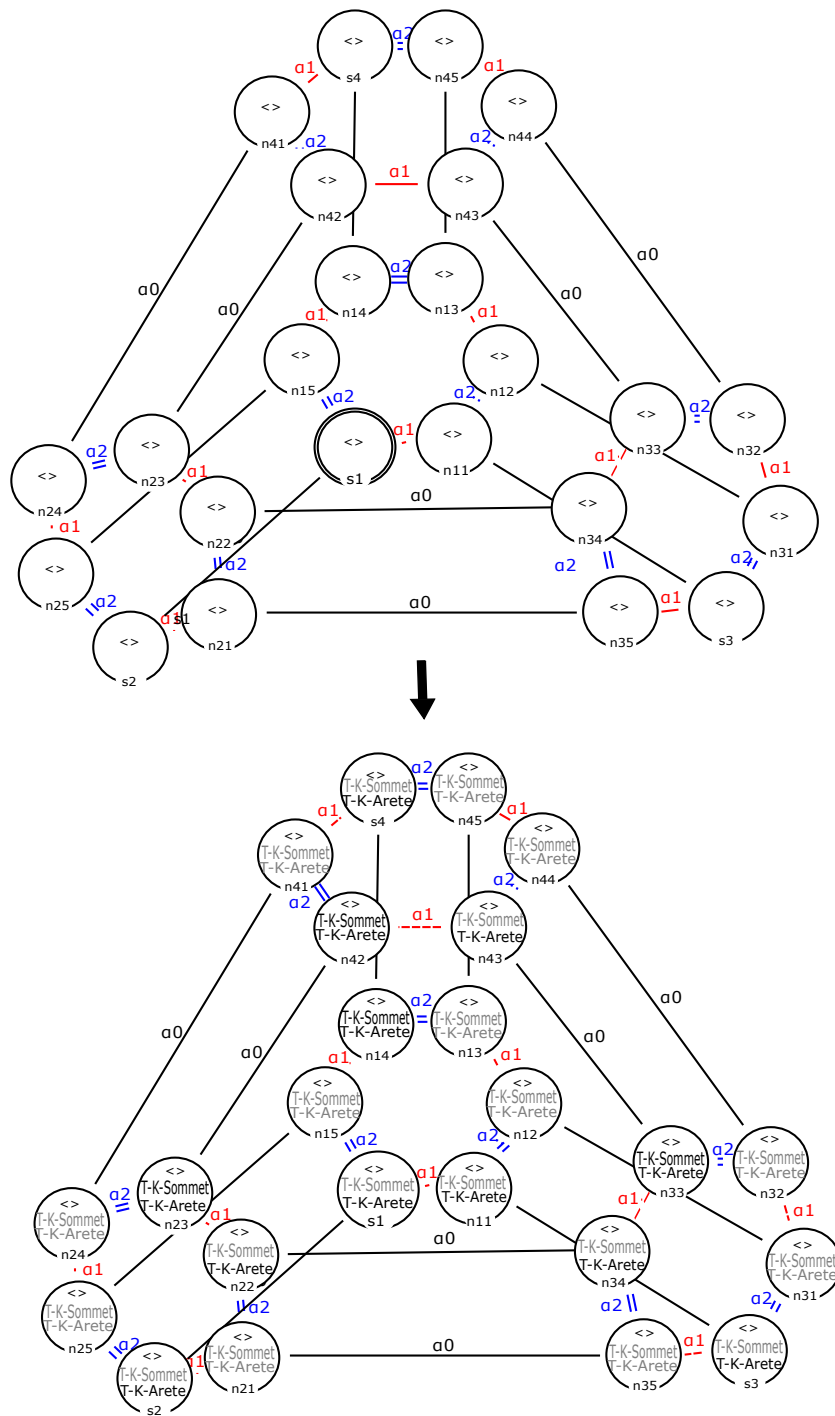


FIGURE 6.6 – Règle de calcul des tenseurs de rigidité pour un tétraèdre isolé.

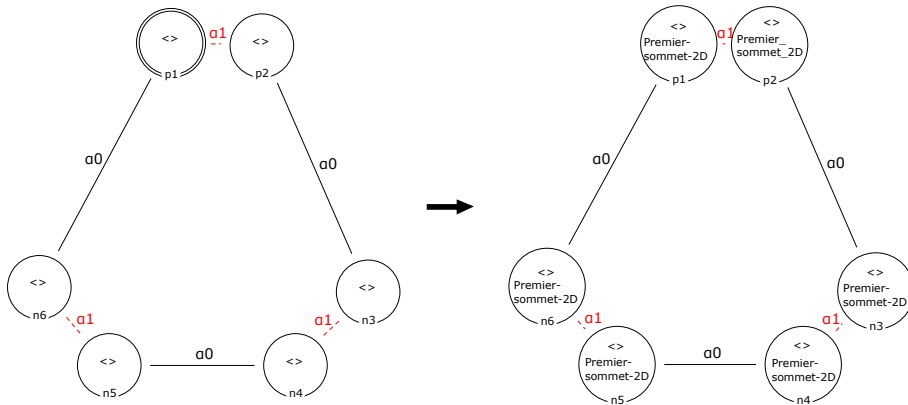


FIGURE 6.7 – Règle de marquage du premier sommet d'un élément 2D.

cette dernière dépend de la matrice de comportement (de taille 3×3) qui est la même pour tout l'objet, nous avons créé la règle 6.8 pour calculer cette matrice et la stocker dans le plongement `M-comportement-2D` porté par l'orbite composante connexe $\langle \alpha_0, \alpha_1, \alpha_2, \alpha_3 \rangle$ une seule fois.

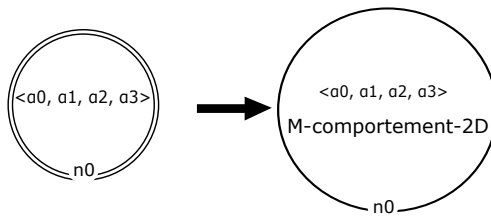


FIGURE 6.8 – Règle d'initialisation de la matrice de comportement.

Finalement, nous avons créé la règle de la figure 6.9 pour calculer cette matrice de rigidité et la stocker dans le plongement `M-rigid-2D` porté par les orbites faces. Cette règle filtre explicitement un triangle. Son pré-traitement récupère, tout d'abord, la matrice de comportement, puis calcule la matrice de déformation de l'élément (de taille 3×6). Ces deux matrices servent à calculer la matrice de raideur avec la même formule utilisée en 3D en substituant le volume par l'aire. Il faut noter que nous n'appliquons cette règle que sur les brins marqués comme premier sommet.

6.2.2 Création d'un maillage

En utilisant le langage à base de règles Jerboa, il est possible de créer directement le maillage géométrique de l'objet à simuler. Il peut ensuite être doté de propriétés mécaniques comme nous venons de le décrire ci-dessus. Il est cependant possible de traiter en même temps la création et l'affectation des propriétés mécaniques. C'est la démarche que nous décrivons dans cette partie. Nous proposons d'utiliser un principe de subdivisions successives qui adaptent les propriétés mécaniques à

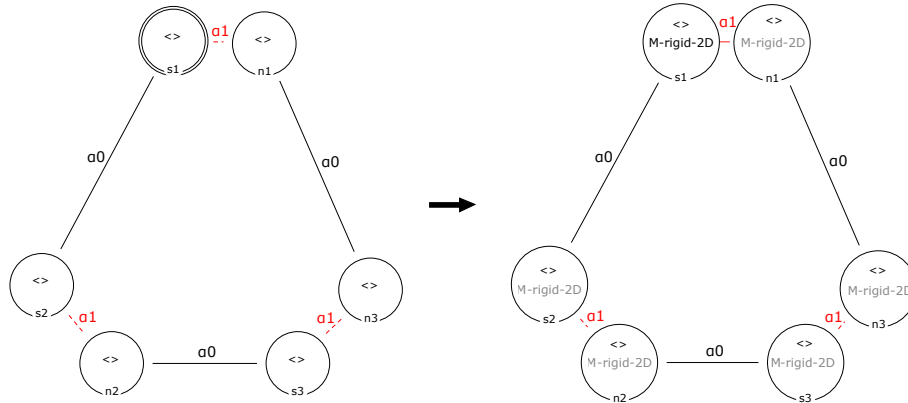


FIGURE 6.9 – Règle de calcul de la matrice du raideur sur un triangle en 2D.

chaque étape. Pour mieux comprendre ce processus, nous prenons l'exemple de la surface en damier décrite dans la section 3.4.2 (page 81) afin de construire un tissu à mailles rectangulaires que l'on peut simuler avec un modèle masses/ressorts. Nous avons modifié les règles de construction déjà fournies dans cette section en ajoutant les plongements mécaniques nécessaires à l'initialisation des ressorts et en mettant tous les plongements associés aux interactions des éléments finis à zéro.

Nous explicitons ici les modifications à appliquer à chaque règle. Nous commençons par la création d'une particule (voir figure 6.10). Elle associe les valeurs initiales (par défaut) des plongements qui sont affichés sur cette règle. Ces plongements comprennent en particulier les plongements :

- **Position** : ce plongement sert à initialiser la position de la particule ;
- **Fixe** : ce plongement permet de définir avec un booléen si la particule est fixe ou mobile. Cela permet de modéliser un tissu fixé par certains sommets. Il est ici initialisé à *true*.

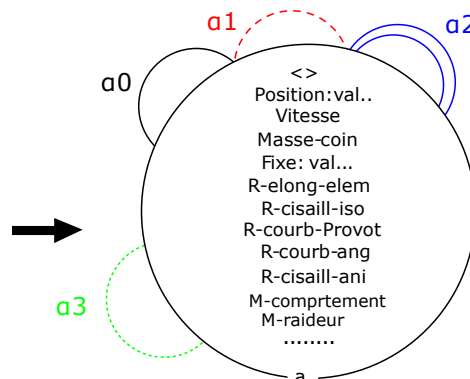


FIGURE 6.10 – Règle de création d'une particule.

Les autres plongements qui sont présentés sur la figure 6.10 (R-elong-elem, R-courb-ang, R-cisaill-iso, R-courb-Provot, Masse-coin, Vitesse ...) ainsi que tous les plongements dans lesquels des forces sont stockées, sont initialisés à zéro. En effet les orbites associées à ces plongements n'existent pas encore, à ce stade de la construction.

Une fois la première particule créée, nous créons une arête. De la même manière, la règle de la figure 3.37 (page 82) est modifiée pour initialiser les plongements mécaniques et nous donne la règle de la figure 6.11. Elle initialise tous les plongements des interactions et des forces à zéro, lorsqu'une nouvelle orbite de plongement est créée par le noeud *b*.

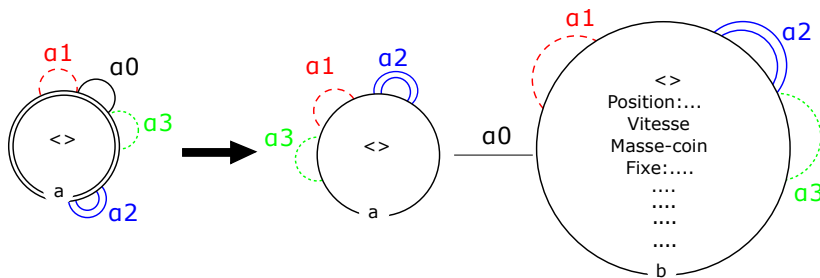


FIGURE 6.11 – Règle de création d’une arête entre deux particules.

Nous mettons également à jour la règle de création d’une face de la figure 3.38 (page 83) afin de créer le tissu complet, mais avec une unique face. Cette règle crée une face grâce à l’extrusion de l’arête créée précédemment (voir la section 3.4.2, page 81). Nous avons modifié cette règle en initialisant les plongements comme montré sur la figure 6.12.

Comme c’est une face qui est créée nous pouvons, à cette étape, lui associer des ressorts, une masse et calculer les masses de chacun de ses coins. Notons que les caractéristiques physiques de chaque ressort (longueur au repos, raideur et amortissement) sont regroupées dans un plongement unique. Les plongements à mettre à jour dans cette étape sont les suivants :

- plongement **Fixe** : il est positionné pour les nouveaux sommets à *false*, afin que seuls les deux coins initiaux du tissu soient fixes ;
- plongement **Masse-coin** : il prend la valeur de masse totale de l’objet (donnée en paramètre) divisée par quatre. Notre approche suppose en effet que la masse d’une face est répartie équitablement entre ses quatre coins. Comme ce plongement est porté par les orbites sommets de face de volume $\langle \alpha_1 \rangle$, nous le définissons sur la figure 6.12 pour les noeuds *b* et *c* et ils se propagent automatiquement aux autres noeuds, respectivement *a* et *d* ;
- plongement **R-elong-elem** : il comprend la longueur au repos de chaque arête de la face (la taille du tissu est enregistrée dans le fichier XML). La raideur du ressort est calculée à partir de la constante d’élongation. Il est également

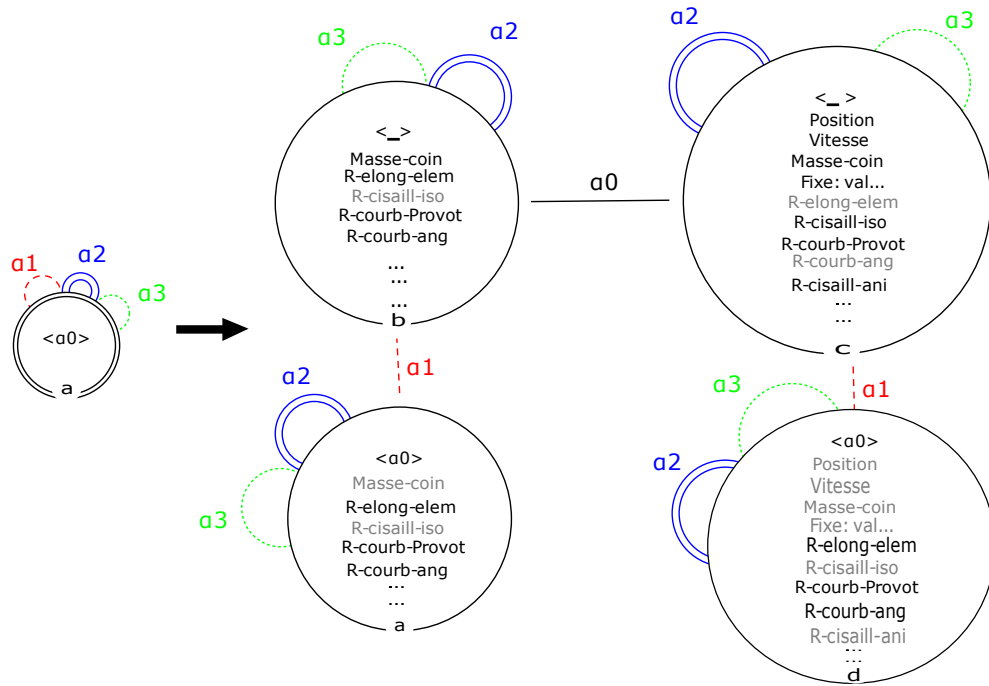


FIGURE 6.12 – Règle de création d’une face.

possible de calculer la raideur en se basant sur le module de Young et le coefficient de Poisson qui sont renseignés dans le fichier XML et en utilisant la formule 2.21 (page 23). Comme ce plongement est porté par les orbites arêtes de face de volume $\langle \alpha_0 \rangle$, il est initialisé pour les nœuds a, b et d et il se propage au nœud c depuis le nœud b car ces deux nœuds sont dans la même orbite de plongement ;

- plongement **R-cisaill-iso** : il comprend la longueur au repos, qui est associée à la longueur de la diagonale du rectangle. Concernant la raideur du ressort, elle est définie directement à partir du fichier XML ou bien elle est calculée en utilisant la formule 2.22 (page 24) en se basant sur le module de Young et le coefficient de Poisson. Comme ce plongement est porté par les orbites faces $\langle \alpha_0, \alpha_1 \rangle$, il est défini pour un seul nœud de la face, et se propage sur les autres ;
- plongement **R-cisaill-ani** : il utilise un traitement similaire à celui du ressort de cisaillement sur une face. Ce plongement est porté par les orbites sommets de face de volume, il est défini pour chaque orbite $\langle \alpha_1 \rangle$ de la face (donc quatre fois).

Les deux plongements associés aux ressorts de courbure (linéaire de Provot et angulaire), ne peuvent pas encore être traités, leur valeur est initialisée à zéro. Par ailleurs, il faut mettre à zéro tous les plongements de stockage des forces.

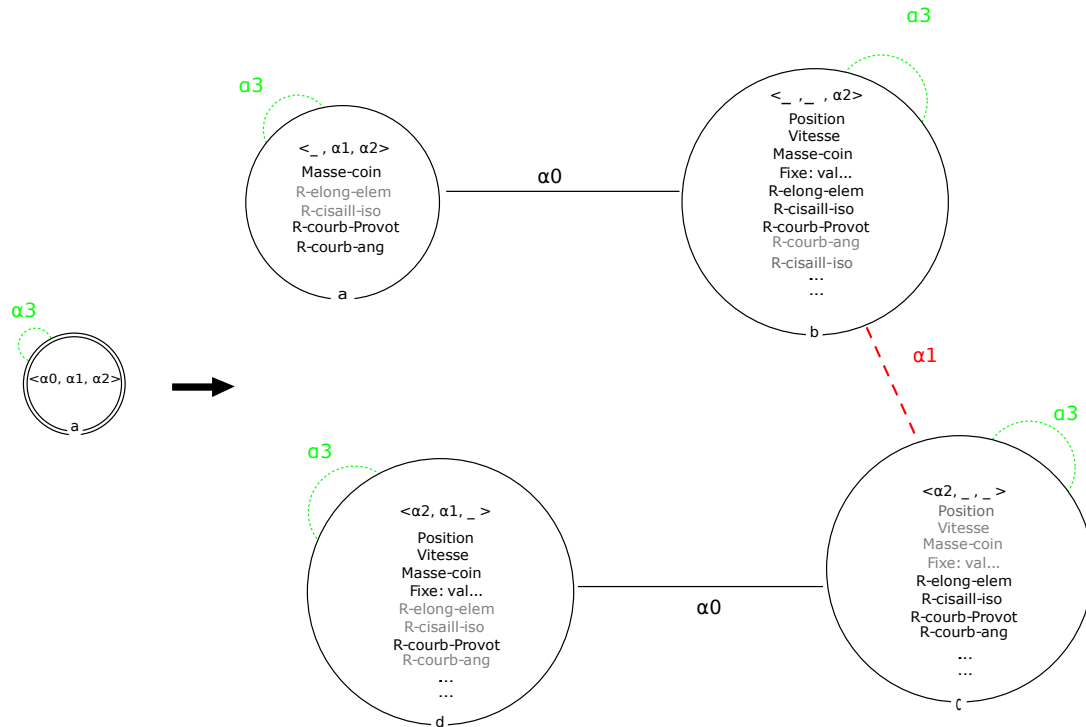


FIGURE 6.13 – Règle de subdivision d’une face rectangulaire.

Finalement, la face est subdivisée itérativement afin de créer les mailles du tissu. Pour cela, nous avons mis à jour les plongements correspondants dans la règle de subdivision montrée sur la figure 6.13. Dans cette règle, nous avons initialisé toutes les forces des ressorts associés aux sommets nouvellement créés à zéro. Il faut, également, adapter les plongements mécaniques et les plongements liés à la topologie comme illustré sur la figure 6.14

Les plongements à mettre à jour sont les suivants :

- plongement **Masse-coin** : il est calculé à partir des valeurs **Masse-coin** de la face initiale. Lorsqu’il y a une subdivision, chaque **Masse-coin** se divise en quatre et se répartit entre les sommets de la sous-face qui lui correspond. En d’autres termes, la **Masse-coin** de la face initiale devient la masse totale de la face nouvellement créée puis cette masse est distribuée à ses quatre coins (voir figure 6.14) ;
- plongement **R-elong-elem** (du ressort d’élongation) : il calcule les valeurs de longueur au repos et de raideur à partir des valeurs de la face courante. La longueur au repos de la face non subdivisée est divisée par deux pour obtenir la longueur au repos des ressorts des sous-faces. Concernant la raideur, comme

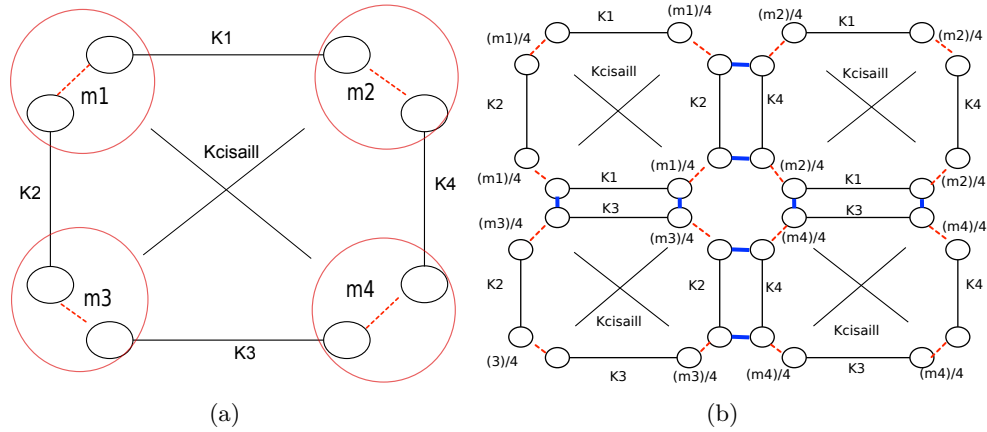


FIGURE 6.14 – Adaptation des valeurs des plongements après subdivision.

illustré sur la figure 6.15, nous considérons que la valeur de raideur reste la même pour tous les niveaux de subdivision. En effet, supposons que chaque ressort se subdivise d'abord en deux ressorts en parallèle, leur raideur est donc de $k/2$ si la raideur initiale du ressort est k . Ensuite chacun de ces ressorts est divisé en deux ressorts mis en série, donc de raideur k ;

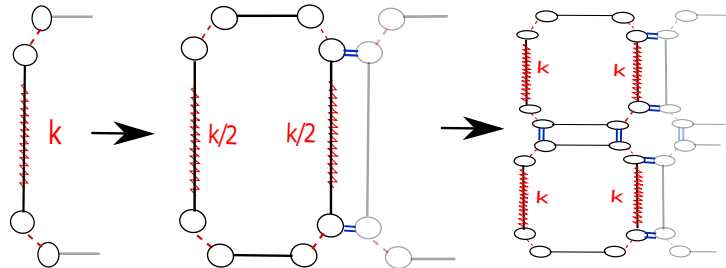


FIGURE 6.15 – Principe de calcul de raideur des ressorts lors d'une subdivision.

- plongement **R-cisaill-iso** (des ressorts de cisaillement isotrope) : comme nous l'avons fait pour les ressorts d'élongation, nous divisons la longueur au repos par deux et nous utilisons les mêmes raideurs de ressorts de cisaillement pour tous les niveaux de subdivision ;
- plongement **R-cisaill-ani** (des ressorts de cisaillement anisotrope) : le traitement est identique à celui du ressort de cisaillement isotrope.

Puisque des faces adjacentes sont apparues, nous introduisons des ressorts de courbure. Pour automatiser et utiliser la même règle de subdivision (pour la première subdivision de la face initiale et les niveaux de subdivision suivants), nous avons décidé de stocker les informations des ressorts de courbure lors de la création de la première face avant même la subdivision.

- plongement **R-courb-Provot** : il comprend la longueur au repos. Pour le calculer, il faut suivre les liaisons α_1 puis α_0 à partir des deux brins de l'orbite $\langle \alpha_2 \rangle$ qui portent le plongement. Ensuite, nous récupérons les positions des sommets associés aux brins obtenus suite aux parcours et qui désignent les particules reliées par le ressort. Pour la raideur, nous utilisons la même valeur de raideur pour tous les niveaux de subdivision ;
- plongement **R-courb-ang** : il comprend les valeurs initialement données en paramètre lors de création de la première face restent inchangées parce qu'elles dépendent d'un angle et non pas d'une longueur au repos (elles sont indépendantes de l'échelle).

Pour pouvoir définir un tissu avec plusieurs subdivisions, nous appliquons la règle de subdivision plusieurs fois sur tout l'objet.

Si l'objet créé en Jerboa se base sur la mécanique continue, nous initialisons, dans une première étape, tous les plongements nécessaires à zéro, et une fois l'objet complètement créé et les subdivisions appliquées, nous utilisons les règles déjà définies pour les objets construits à partir d'un fichier.

6.3 Boucle de simulation

Pour simuler un objet 2D/3D représenté par un maillage géométrique avec un modèle mécanique (discret ou continu), nous utilisons la boucle de simulation décrite dans la section 2.3.2, page 36. Rappelons que cette boucle commence, tout d'abord, par le calcul des forces appliquées par chaque interaction. Ensuite, il faut calculer l'accélération de chaque particule en utilisant le bilan des forces appliquées et le principe fondamental de la dynamique. Une fois l'accélération de chaque particule calculée, il faut évaluer sa nouvelle vitesse et position en utilisant une méthode d'intégration.

Avec notre modèle, la boucle de simulation décrite ci-dessus se traduit par la succession d'étapes suivante : pour calculer les forces internes du maillage, il faut parcourir le maillage et chercher les orbites qui stockent des propriétés mécaniques pour chaque type d'interaction. Une fois une interaction trouvée, il faut reconnaître son sous-graphe, tel qu'il a été présenté dans le chapitre précédent. S'il n'est pas reconnu, l'interaction doit être ignorée. S'il est reconnu, il faut identifier les sommets de stockage des forces. Les forces sont alors calculées et stockées dans les orbites qui conviennent. Une fois les forces de toutes les interactions calculées, nous dressons le bilan des forces en chaque particule et calculons son accélération. Finalement, nous calculons les nouvelles vitesses et positions des particules par intégration. Nous avons choisi d'utiliser la méthode d'intégration d'Euler symplectique car elle est simple et suffisante pour tester nos simulations.

Chacune de ces étapes se caractérise par une ou plusieurs règles Jerboa. On trouve ainsi une règle pour le bilan des forces, le calcul de l'accélération des particules, de leur vitesse et leur nouvelle position. Nous avons également choisi de

créer une règle pour chaque type d'interaction. Chacune de ces règles s'applique sur un motif précis correspondant au sous-graphe de l'interaction associée. Cependant, pour appliquer ces règles en chaque interaction portée par l'objet, il faut écrire un script qui parcourt cet objet à la recherche de ces interactions, en se basant sur le pseudo-code suivant :

Algorithme 1 pseudo-code de script d'application d'une règle sur tout l'objet.

```

règle ← règle-à-appliquer
orbite ← orbite-stockage-interaction
pour nœud : g-cartes faire
  si nœud non marqué alors
    règle.appliquer-règle(nœud)
    marquer-orbite(nœud,orbite)

```

Finalement, pour simuler la déformation d'un objet avec un modèle mécanique donné (pour une dimension bien définie et avec une sélection d'interactions), il faut créer un algorithme qui permet d'enchaîner plusieurs règles l'une après l'autre comme indiqué dans la section 3.4.2, page 81. En effet, dans ce script, il faut d'abord sélectionner les règles liées aux interactions puis appliquer les règles de bilan de forces, puis de calcul d'accélération et d'intégration.

Remarquons que Jerboa nous propose deux types de règles : celles basées sur un motif explicite que nous avons évoqué, plus haut, pour les interactions, et celles qui s'appuient sur l'intégralité d'une composante connexe, par exemple pour mettre à jour des plongements. Dans notre moteur de simulation, les deux types de règles ont été utilisés. C'est pourquoi nous discutons de la différence entre ces deux types et de leur utilité, compte tenu de notre contexte, dans la sous-section suivante. Puis, dans la suite, nous détaillons les étapes de la boucle de simulation, une par une, avec les règles créées en Jerboa.

6.3.1 Règles sur la composante connexe vs règle sur un motif

Avant de détailler les règles créées pour simuler les objets, nous présentons ici une petite comparaison entre les styles de règles de Jerboa. Il existe deux types de règles, celles qui s'appliquent sur la composante connexe et les règles qui s'appliquent sur un motif déterminé. Par exemple, si nous voulons calculer la valeur d'un plongement porté par l'orbite sommet et que le calcul est identique pour tous les sommets d'une composante connexe, nous pouvons utiliser une règle sur la composante connexe. La valeur est ainsi calculée pour tous les sommets en une seule application de la règle. Le parcours est alors géré par le moteur d'application de règles, en interne. Ce type de règles paraît utile si l'on veut calculer ou faire une mise à jour d'un plongement précis sur toute une composante connexe en un seul coup. Elle permet alors de gagner en temps de calcul car un seul parcours est effectué (lors du filtrage de la composante connexe), c'est pour cela que nous avons utilisé ce type de règle pour l'initialisation. Notons qu'il faut tout de même écrire un script qui parcourt

les composantes connexes si l'objet en contient plusieurs, les performances chutent alors. En outre, dans quelques cas et en utilisant ce type de règles, des redondances de calcul apparaissent. Par exemple, la force calculée pour un ressort d'élongation entre deux sommets se calcule pour chaque sommet (donc deux fois) alors que le principe d'action/réaction indique que les forces associées aux deux sommets sont opposées l'une de l'autre.

Au contraire, nous pouvons définir des règles qui s'appliquent sur un sous-graphe filtré explicitement en indiquant tous les nœuds qui le constituent ou bien qui est défini par une orbite. Ces règles sont utiles voire obligatoires lorsque l'on veut appliquer une règle sur un motif bien déterminé par exemple un triangle (et non une face en général). De même, pour faire des modifications topologiques locales dans l'objet, nous devons utiliser des motifs explicites, car seule une partie de l'objet est concernée par le changement. Typiquement, dans un certain nombre de cas, les deux types de règles sont possibles. Les calculs d'interactions en font partie. Toutefois, il vaut mieux choisir les plus efficaces en prenant en compte le temps de calcul et/ou la place mémoire.

6.3.2 Règles de calcul des forces

Compte tenu de la modélisation des interactions proposée dans le chapitre précédent, nous avons choisi, pour calculer les forces, de créer des règles qui s'appliquent sur les sous-graphes des interactions plutôt que sur la composante connexe. Ces sous-graphes entourés en bleu que nous avons définis dans la section 5.2.1 (page 108) du chapitre précédent sont, dans l'extrême majorité des cas, utilisés comme membre de gauche des règles (en réalité, il existe un cas, les ressorts de cisaillement volumique isotrope, où l'approche par composante connexe qui a été privilégiée, nous en détaillons les raisons dans la suite). L'utilisation du sous-graphe nous garantit de pouvoir déterminer les brins et orbites de plongement nécessaires au calcul des interactions et au stockage des forces.

Comme ces règles s'appliquent uniquement sur un motif, elles ne permettent de traiter qu'un type d'interactions en une seule zone de l'objet (désignée par le nœud d'accroche). Il faut alors créer un script qui parcourt l'objet afin de trouver toutes les interactions de ce type et demander l'application de la règle sur le nœud d'accroche (qui, par convention et à des fins d'efficacité, est toujours choisi dans l'orbite support de l'interaction). Une fois une interaction traitée, les brins qui composent son orbite sont marqués afin de chercher une nouvelle orbite d'interaction. Ce processus est ré-itéré jusqu'à avoir parcouru toutes les orbites supports.

Nous introduisons dans cette partie les règles que nous avons créées avec Jerboa pour calculer les forces de chaque interaction modélisée dans le chapitre précédent. Nous commençons tout d'abord avec les règles créées pour les interactions discrètes correspondant aux différents types de ressorts. Ensuite, nous nous focalisons sur les règles créées pour calculer les forces issue de la mécanique continue (éléments finis linéaires, masses/tenseurs et co-rotationnels). Nous rappelons aussi que tous les plongements des forces sont initialisés à zéro.

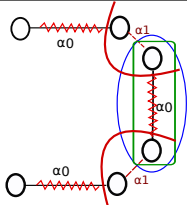
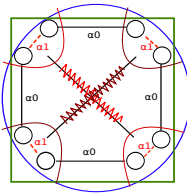
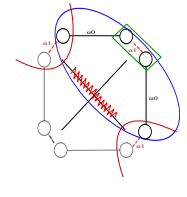
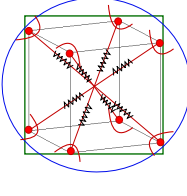
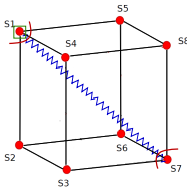
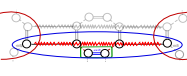
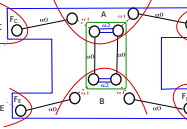
Ressorts	Mailles	Exemples d'application	Orbites des interactions	Orbites des forces	
Élongation	Élémentaire	Quelconque		Arêtes de face de volume : $\langle \alpha_0 \rangle$	Brins : $\langle \rangle$
Cisaillement	Surfacique isotropique	Rectangle		Faces de volume : $\langle \alpha_0, \alpha_1 \rangle$	Sommets de face de volume : $\langle \alpha_1 \rangle$
	Surfacique anisotropique	Quelconque		Sommets de face de volume : $\langle \alpha_1 \rangle$	Brins : $\langle \rangle$
	Volumique isotropique	Hexaèdre		Volume : $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$	Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$
	Volumique anisotropique	Hexaèdre		Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$	Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$
Courbure	Linéaire de Provot	Quelconque		Extrémités d'arête en 2D : $\langle \alpha_2 \rangle$	Brins : $\langle \rangle$
	Angulaire	Triangulaire et quadrilatère		Arêtes : $\langle \alpha_0, \alpha_2 \rangle$	Brins : $\langle \rangle$

TABLE 6.1 – Modélisation des interactions M/R

Cas des ressorts

Pour mieux comprendre cette partie, nous rappelons dans la table 6.1, les différents types d'interactions discrètes modélisées dans le chapitre précédent, avec un exemple d'application mettant en évidence les sous-graphe des interactions, les orbites de stockage des interactions et les orbites de stockage des forces. Comme expliqué ci-dessus, le sous-graphe nous sert à définir le motif de filtrage et il est indiqué dans le membre gauche de la règle. À partir du sous-graphe, nous trouvons les informations (position et vitesse des particules concernées) permettant de calculer l'interaction ainsi que les orbites où les forces calculées doivent être stockées.

Notons que le fait de se baser sur le sous-graphe de l'interaction dans le membre gauche est nécessaire mais non suffisant pour éviter les redondances de calcul. En effet, avec Jerboa, chaque calcul de plongement indiqué dans le membre de droite fait l'objet d'un traitement spécifique. Le principe d'action/réaction n'est pas exploité et le calcul des forces appliquées par un ressort sur ses extrémités est donc, ici aussi, effectué deux fois. Il nous faut donc réussir à « factoriser » le calcul de l'interaction et ensuite distribuer cette force et son opposée sur les sous-orbites des sommets concernés. Nous exploitons alors la phase de pré-traitement de la règle pour calculer la force exercée sur l'un des sommets. Le calcul du plongement récupère la valeur calculée et l'affecte ou affecte son opposée au sommet traité. Cette solution a été utilisée pour la plupart des interactions afin d'obtenir une exécution plus efficace en temps. Nous verrons d'ailleurs, dans la suite, qu'elle a également été utilisée, pour des raisons similaires, dans le cas des approches continues.

Concernant le calcul des forces proprement dit, lorsqu'elles sont issues des ressorts linéaires, nous utilisons la formule 2.2, (page 10) pour un comportement purement élastique. Si l'interaction comprend une composante d'amortissement, nous utilisons la formule 2.4 (10).

Nous allons maintenant expliquer la démarche décrite au-dessus pour les différents cas de ressorts.

- Ressorts d'élongation

Pour calculer les forces appliquées par les ressorts d'élongation qui relie deux sommets adjacents d'une face, nous avons créé la règle présentée sur la figure 6.16. Dans le membre de gauche, le motif à filtrer correspond à une arête de face, bâtie autour du lien α_0 , avec ses deux extrémités $\mathbf{n0}$ et $\mathbf{n1}$ qui correspondent, dans ce cas, aux sommets d'application des forces.

Dans la partie droite, nous indiquons que les forces de cette interaction s'appliquent aux sommets associés aux nœuds $\mathbf{n0}$ et $\mathbf{n1}$. Ces deux forces sont opposées. Aussi, pour éviter de calculer deux fois la même amplitude de force, la force exercée sur $\mathbf{n0}$ est calculée en phase de pré-traitement de la règle. Les positions et vitesses des deux sommets de $\mathbf{n0}$ et $\mathbf{n1}$ sont récupérées, la formule générale des ressorts est appliquée, avec la raideur, la longueur au repos et éventuellement l'amortissement de l'interaction, dont le plongement est stocké dans l'orbite α_0 . Le traitement final de la règle permet d'affecter la force calculée au plongement `F-elong-elem` de $\mathbf{n0}$

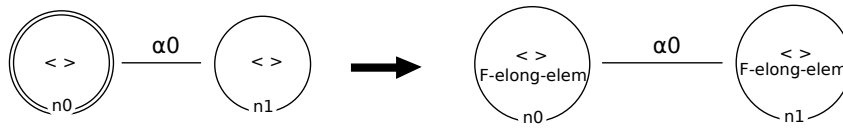


FIGURE 6.16 – Règle de calcul des forces des ressorts d'élongation.

et son opposée au plongement de $n1$. Cette règle s'applique sur une seule arête de face. Pour pouvoir l'appliquer sur tout l'objet en 2D ou 3D et pour tout type de maillage, nous avons créé un script. Pour mettre en œuvre ce parcours, nous parcourons toutes les orbites $\langle \alpha_0 \rangle$ qui portent les ressorts, et appliquons la règle sur l'un de ses brins. Nous marquons cette arête de face et nous passons au brin suivant non marqué, jusqu'à que tous les arêtes de faces de l'objet aient été marqués.

- Ressorts de cisaillement surfacique isotrope

Comme décrit dans le chapitre précédent, nous pouvons utiliser les ressorts de cisaillement pour tout type de maillage, mais dans le cas de maillages triangulaires/tétraédriques, ils sont inutiles. Nous avons différencié deux types de ressorts de cisaillement : ceux qui permettent de maintenir la forme d'une face rectangulaire via deux ressorts de même raideur et longueur au repos (cas dit « isotrope » dans la suite du document) et des ressorts de cisaillement qui permettent de maintenir l'angle entre deux arêtes d'une face quelconque (cas dit « anisotrope » dans la suite).

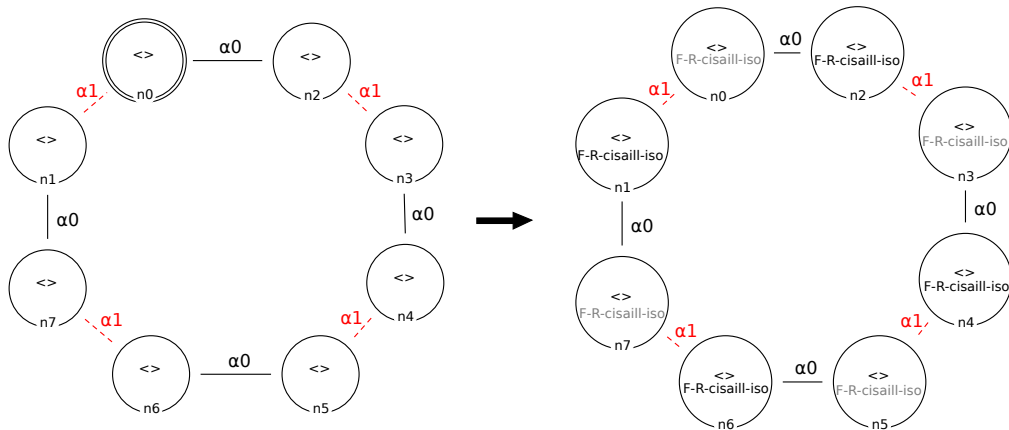


FIGURE 6.17 – Règle de calcul des forces des ressorts de cisaillement surfacique isotrope.

La règle 6.17 permet de calculer les forces dans le cas isotrope. Nous avons filtré dans la partie gauche une face rectangulaire explicitement, ce qui identifie

les sommets qui sont diagonalement opposés. Dans le membre droit de la règle, nous indiquons que les forces sont à calculer pour les quatre sommets de la face, symbolisés respectivement par les nœuds **n1**, **n2**, **n4** et **n6**. Comme les forces associées à **n1** et **n4** d'une part et à **n2** et **n6** d'autre part, sont opposées, nous calculons ces forces une seule fois dans la phase de pré-traitement de la règle. Les positions et vitesses portées par **n1** et **n4** nous permettent de calculer une première force, avec les paramètres de l'interaction (raideur, longueur au repos, amortissement) portés par la face. Une seconde force est alors calculée avec les positions et vitesses trouvées dans les nœuds **n2** et **n6**. Une fois les deux forces calculées, elles sont affectées respectivement aux plongements **F-R-cisaill-iso** portés par les orbites $\langle \alpha_1 \rangle$ de **n1** et **n2** et leur opposée à ceux de **n4** et **n6**. Un script général permet l'application de cette règle sur toutes les faces de l'objet.

- Ressorts de cisaillement surfacique anisotrope

Pour le cas de ressorts anisotropes, nous rappelons qu'ils sont stockés sur les orbites sommets de face de volume $\langle \alpha_1 \rangle$ et les forces sont stockées sur les orbites brins $\langle \rangle$ dans le plongement **F-R-cisaill-ani**.

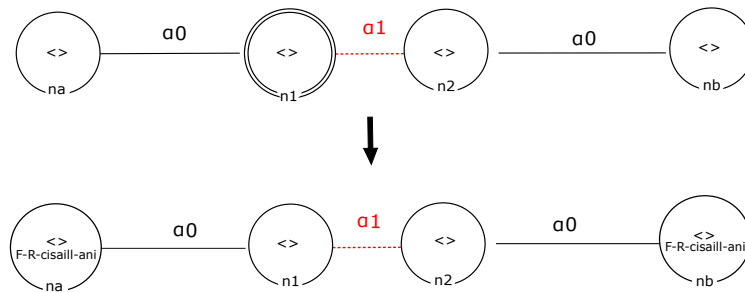


FIGURE 6.18 – Règle de calcul des forces des ressorts de cisaillement surfacique anisotrope.

Pour calculer les forces appliquées par les ressorts du cas anisotrope, nous avons créé la règle 6.18. Nous avons filtré dans le membre gauche (présenté en haut de la figure 6.18) deux arêtes liées par une liaison α_1 , ce type d'orbite stockant les propriétés du ressort que l'on veut traiter. Le membre droit (présenté en bas de la figure 6.18) indique que les forces calculées sont stockée (via le plongement **F-R-cisaill-ani** porté par les orbites brins) aux extrémités non liées des deux arêtes, représentées par les nœuds **na** et **nb**. Comme dans les deux cas précédents, les forces exercées sont opposées et la force pour **na** est calculée en phase de pré-traitement selon une approche similaire aux deux cas de ressorts étudiés précédemment.

Comme montré sur la figure 6.18, cette règle ne dépend pas d'un type donné de maille, elle peut au contraire s'appliquer sur n'importe quel type de faces. Dans le cas triangulaire, elle n'a cependant pas d'intérêt car elle correspond à l'ajout d'un

ressort sur une arête existante, comportant déjà un ressort d'élongation. Mais dans le cas de faces pentagonales ou avec plus encore de sommets, cette règle est tout à fait fonctionnelle. À noter qu'elle est à appliquer sur toutes les orbites $\langle \alpha_1 \rangle$ de l'objet, via un script.

- Ressorts de cisaillement volumique isotrope

Les ressorts de cisaillement volumique sont aussi modélisés dans notre modèle. Il sont appliqués uniquement dans le cas où la maille est supposée être un parallélépipède rectangle. Nous rappelons que les propriétés mécaniques de ces ressorts sont stockées dans les volumes (orbites $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$) et les forces dans les sommets de volume (orbites $\langle \alpha_1, \alpha_2 \rangle$).

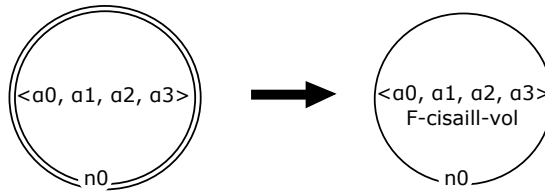


FIGURE 6.19 – Règle de calcul des forces des ressorts de cisaillement volumique isotrope.

La règle présentée par la figure 6.19 permet de calculer les forces de ces ressorts. Nous remarquons ici que le membre de gauche filtre la composante connexe et ne représente pas explicitement un hexaèdre tel que représenté sur la figure 6.20. En effet, la complexité de cette structure (48 brins) rend son identification par la règle coûteuse en temps de traitement alors que, dans notre cas, seuls huit identifiants de brin sont utilisés pour affecter les forces aux sommets de l'hexaèdre. Une telle règle ne vérifie donc pas que le volume est bien un hexaèdre. Nous faisons en sorte que, durant l'initialisation, chaque volume ne porte effectivement d'information de cisaillement volumique que dans le cas où il s'agit d'un hexaèdre. Le membre droit de la règle liste les plongements à mettre à jour sur toute la composante connexe, en l'occurrence les plongements `F-cisaill-vol` portés par tous les sommets de volume.

Avec cette règle, nous ne pouvons pas utiliser la phase de pré-traitement pour éviter la redondance de calcul, car cette phase concerne toute la composante connexe et non chaque volume. La règle parcourt en effet chaque orbite portant un plongement `F-cisaill-vol`, désigne un nœud de cette orbite, trouve le sommet opposé au sommet associé à ce nœud, en suivant successivement les liens $\alpha_0, \alpha_1, \alpha_0, \alpha_2, \alpha_1$ et α_0 . Si les informations de ressorts de cisaillement volumique sont trouvées dans le volume, la force est calculée à partir des positions et vitesses des deux sommets et la force obtenue est plongée dans le sommet considéré. Chaque interaction est donc calculée deux fois. À noter que cette règle demande à parcourir les composantes connexes de l'objet, si ce dernier en possède plusieurs.

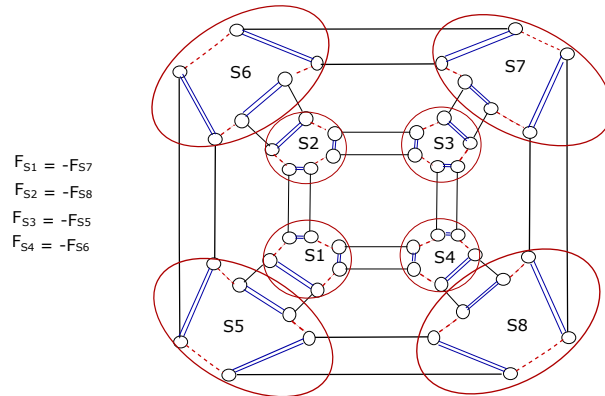


FIGURE 6.20 – Filtrage explicite d'un parallélépipède.

- Ressorts de cisaillement volumique anisotrope

Nous rappelons que nous avons choisi de stocker les propriétés mécaniques de ces ressorts dans les orbites sommets du volume $\langle \alpha_1, \alpha_2 \rangle$ d'un des sommets de la diagonale. Il sont appliqués uniquement dans le cas où la maille est supposée être un parallélépipède rectangle.

Pour calculer les forces appliquées par ces ressorts, nous avons créé la règle présentée sur la figure 6.21. Dans le membre gauche, nous avons filtré le sous-graphe qui filtre trois arêtes adjacentes représentées par les liaisons $\alpha_0, \alpha_1, \alpha_2, \alpha_1, \alpha_0, \alpha_2, \alpha_0$ à partir d'un nœud quelconque du sommet de volume qui stocke les propriétés du ressort. Le membre droit indique que les forces à calculer sont à stocker dans le plongement `F-cisail-vol-ani` dans les orbites sommets de volume des deux extrémités de la diagonale de volume représentée par les nœuds `na` et `nb`. Comme les forces exercées sur les deux extrémités sont opposées, nous avons calculé la force pour `na` en phase de pré-traitement, à partir des informations d'interaction stockée dans ce nœud. Cette force est alors stockée dans l'orbite sommet de volume dont fait partie `na`, et son opposée est plongé dans l'orbite de l'autre extrémité représentée par le nœud `nb`.

- Ressorts linéaires de Provot

Nous rappelons que nous avons choisi de stocker les propriétés mécaniques des ressorts linéaires de Provot dans les extrémités des arêtes en 2D (orbites $\langle \alpha_2 \rangle$) et les forces dans les brins (orbites $\langle \rangle$).

Pour calculer les forces de ces ressorts, nous avons créé la règle présentée sur la figure 6.22. Nous avons filtré dans le membre gauche (présenté en haut de la figure 6.22), deux arêtes (deux liens α_0) liées chacune par un coin de face (lien α_1) avec une troisième arête unique, liant les deux faces (lien α_2). Les deux extrémités libres de deux arêtes sont nommées `na` et `nb`. Dans le membre droit de la règle

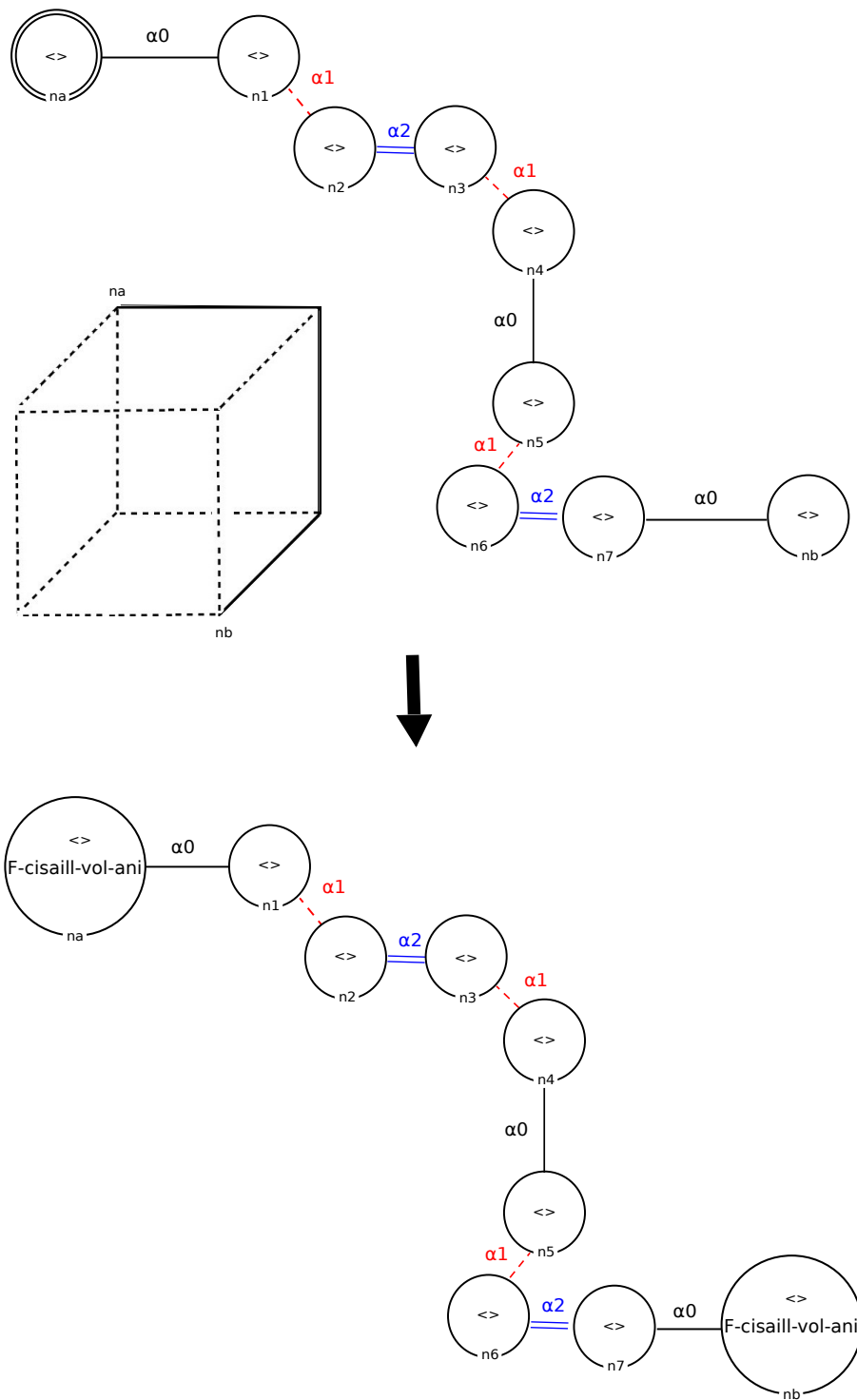


FIGURE 6.21 – Règle de calcul des forces des ressorts de cisaillement volumique anisotrope.

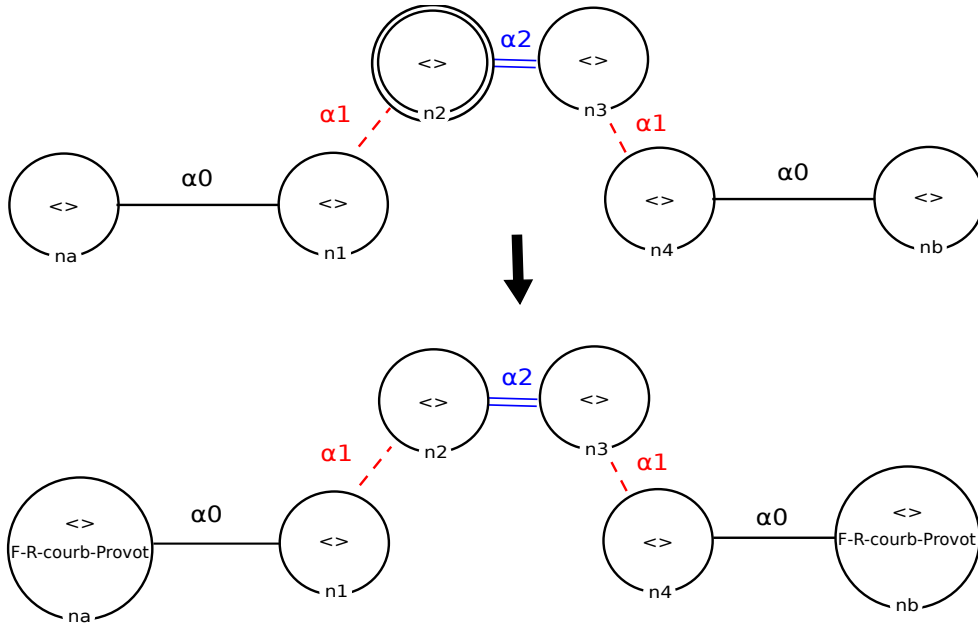


FIGURE 6.22 – Règle de calcul des forces des ressorts linéaires de Provot.

(présenté en bas de la figure 6.22), nous calculons dans la phase de pré-traitement la force liant les sommets représentés par na et nb . Cette force est ensuite stockée dans le plongement `F-R-courb-Provot` du nœud na et son opposée sur le nœud nb . Notons que cette règle suppose qu'il existe deux faces adjacentes (présence du lien α_2 dans la règle), mais ne fait aucune supposition sur les faces elles-mêmes. Il est donc possible d'utiliser cette règle pour tout type de face, y compris dans le cas de maillages hétérogènes.

- Ressorts angulaires (2D)

Dans notre modélisation, nous avons décidé de mettre les ressorts angulaires sur les arêtes liant deux faces adjacentes (orbites $\langle \alpha_0, \alpha_2 \rangle$) et les forces dans les orbites brins associées aux sommets concernés par cette interaction (qui sont, dans ce cas, au moins quatre). La règle à appliquer pour calculer ces forces est présentée sur la figure 6.23.

Dans cette règle, le membre de gauche (présenté en haut de la figure 6.23) représente l'articulation entre deux faces. On y trouve l'arête de cette articulation et les arêtes qui lui sont adjacentes sur les deux faces. Ceci permet de ne pas faire d'hypothèse sur le nombre de sommets des faces (triangles ou quadrilatères). L'arête qui porte le ressort angulaire est présentée par les quatre nœuds $na2$, $na3$, $nb2$ et $nb3$. Les quatre arêtes liées à cette arête ont leur autre extrémité représentée par les nœuds nc , nd , ne et nf . Ces sommets extrémités ainsi que les sommets de l'arête d'articulation subissent des forces liées à l'articulation. Ceci se retrouve dans le membre droit de la règle (présenté en bas de la figure 6.23). Chaque sommet extrémité des arêtes, représenté respectivement par nc , nd , ne et nf , récupère une force

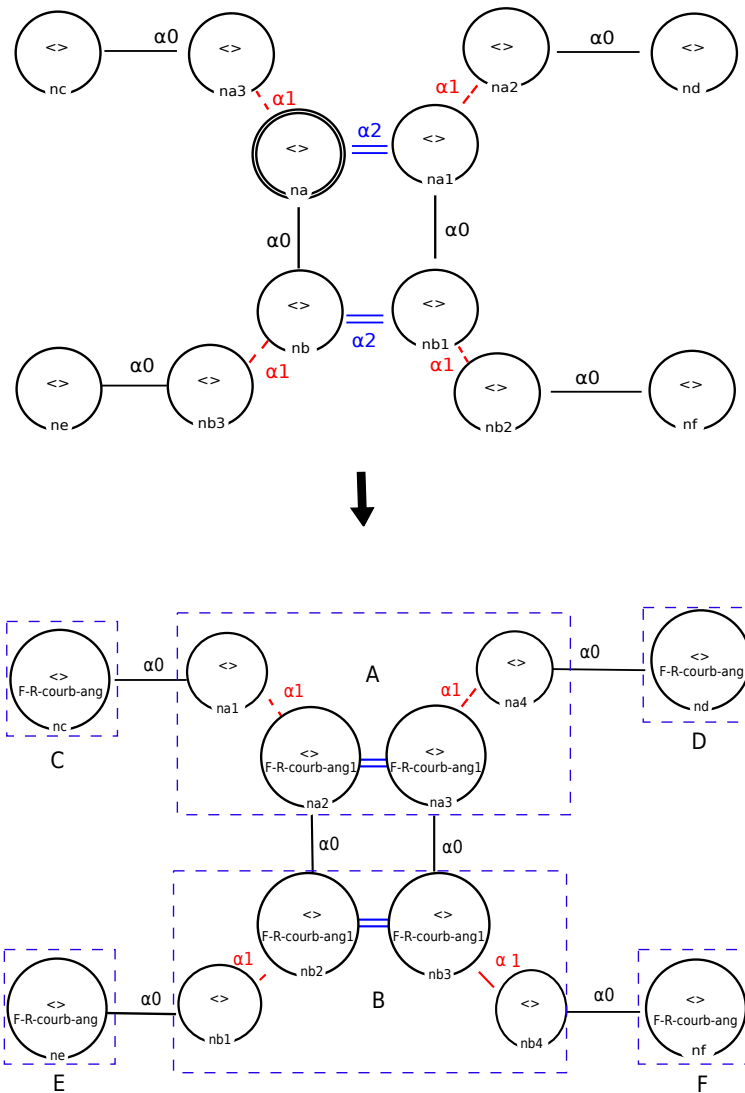


FIGURE 6.23 – Règle de calcul des forces des ressorts angulaires (les cadres pointillés bleus ne font pas partie de la règle, mais permettent, dans le texte, de désigner les sommets par une lettre et de reconnaître les brins qui les constituent).

de courbure dans le plongement `F-R-courb-ang` porté par ces mêmes nœuds. Pour respecter le principe d'action/réaction, des forces opposées sont stockées dans des plongements appelés `F-R-courb-ang1` portés par les nœuds de l'arête d'articulation `na2`, `na3`, `nb2` et `nb3`. Nous rappelons ici que la modélisation des forces nécessitent deux plongements différents car un seul plongement entraîne un conflit de stockage, dans un même brin, des forces issues de ressorts angulaires différents.

Afin de calculer les forces appliquées par ce type de ressort, nous avons utilisé la formule (6.2), qui est assez semblable à celle de Bridson [BMF03] mais simplifiée et étendue aux quadrilatères.

$$\mathbf{f}_M = \pm \frac{k_i \times \theta}{2 \times \|\mathbf{u} \times \mathbf{PM}\|^2} \times (\mathbf{u} \times \mathbf{PM}) \quad (6.2)$$

Avec k la raideur, \mathbf{u} le vecteur unitaire de l'arête commune (il est formé à partir des sommets A et B de l'arête) et θ l'angle entre les normales des deux faces. S'il s'agit d'un quadrilatère, la normale n'est pas obligatoirement constante sur la face, car cette dernière peut ne pas être plane. Dans ce cas, nous choisissons la normale du triangle défini par l'arête de l'articulation et le centre de la face. Le point P correspond à un point quelconque de l'arête (l'emplacement n'ayant pas d'influence sur le résultat, il est recommandé d'utiliser A ou B), et M correspond au sommet qui subit la force (dans notre cas les sommets représentés par les nœuds `nc`, `nd`, `ne` et `nf`). Comme détaillé dans [BMF03], quand un sommet subit une force, l'opposé de cette force doit être distribuée aux sommets de l'arête. Ici, nous adaptons la méthode aux quadrilatères en considérant que $\mathbf{f}_A = -\mathbf{f}_C - \mathbf{f}_D$. La force $-\mathbf{f}_C$ est stockée en `na2` et la force $-\mathbf{f}_D$ est stockée en `na3`. Nous procédons de façon similaire avec les forces $-\mathbf{f}_E$ et $-\mathbf{f}_F$ que nous stockons, respectivement, dans les brins `nb2` et `nb3` de l'orbite du sommet B .

Comme cette règle s'applique uniquement à deux faces adjacentes, nous avons écrit un script qui applique cette règle en chaque arête. À noter que si certaines faces ne sont ni triangulaires, ni quadrangulaires, il est déconseillé d'utiliser ce type de ressort (la règle utilisée ne permet pas de calculer une force en tous les sommets de ce type de face).

Les ressorts angulaires ou de Provot visent à simuler le même phénomène mais les forces sont calculées différemment. Aussi, les deux approches sont implantées dans notre modèle, car l'approche de ressort angulaire est plus efficace en terme de comportement lors de la simulation dans le cas d'un maillage rectangulaire et triangulaire, tandis que celle de Provot peut potentiellement être généralisée à d'autres types de faces.

En résumé, pour le calcul des forces des ressorts modélisés dans le chapitre précédent, nous avons créé une règle pour chaque type de ressort suivant : ressort d'élongation, ressort de cisaillement surfacique isotrope et anisotrope, ressort de cisaillement volumique isotrope et anisotrope, ressort linéaire de Provot et ressort angulaire. La plupart des règles peuvent être appliquées en 2D ou 3D pour différents

types de maillages. Cependant, les règles des interactions de courbure ne sont appliquées que pour les objets 2D, car les interactions correspondantes n'ont pas d'utilité en 3D. Par ailleurs, nous devons signaler que quelques règles parmi celles présentées ci-dessus correspondent à des interactions élémentaires. Plusieurs interactions élémentaires peuvent correspondre au même couple de sommets, ce qui aboutit à des redondances de calcul. Nous discutons de ce problème dans la section 6.4.

Cas des approches par éléments finis

Dans le chapitre précédent, nous avons présenté la modélisation de trois approches éléments finis, à savoir les éléments finis linéaires, les masses/tenseurs et la méthode du co-rotationnel. La table 6.2 synthétise ces modèles en indiquant pour chacun, la modélisation proposée pour le motif à filtrer, les orbites de stockage des propriétés mécaniques et celles de stockage des forces.

Nous commençons tout d'abord par présenter les règles utilisées pour calculer les forces avec les éléments finis linéaires. Toutes les règles qui vont être présentées dans cette section, ne s'appliquent qu'aux tétraèdres en 3D et aux triangles en 2D.

- Élément finis linéaires

Pour calculer les forces appliquées sur chaque sommet d'un tétraèdre, nous avons créé la règle présentée sur la figure 6.24. Nous avons filtré, dans le membre gauche (présenté en haut de la figure 6.24) un tétraèdre, en détaillant explicitement tous ses nœuds, et dans le membre de droite (présenté en bas de la figure 6.24), nous indiquons qu'il faut affecter une force pour chaque sommet dans le plongement `F-sommet-MEF`. Le processus commence par parcourir tous les brins de l'objet pour chercher ceux qui portent un plongement `Premier-sommet` dont la valeur est `true`. Ensuite, la phase de pré-traitement de la règle identifie, à partir du premier sommet, les trois autres sommets du tétraèdre et leur numéro. Les vecteurs déplacements de ces quatre sommets sont alors réunis dans un seul vecteur de douze coordonnées. Ensuite, nous le multiplions par la matrice de raideur de taille (12×12) déjà stockée en chaque élément comme indiqué dans la formule (2.27), page 27. La phase de pré-traitement se termine en mémorisant un vecteur de douze coordonnées \mathbf{f} qui regroupe les forces appliquées sur chaque sommet du tétraèdre $(f_1^x, f_1^y, f_1^z, f_2^x, f_2^y, \dots)^t$. Les routines d'affectation des plongements force consistent à aller chercher le vecteur force du sommet considéré dans le vecteur regroupant toutes les forces et le stocker dans le plongement `F-sommet-MEF` de chaque sommet du tétraèdre. Sur la figure 6.24, les forces sont stockées sur les orbites sommets de volume qui contiennent les nœuds `s1`, `s2`, `s3` et `s4`.

- Co-rotationnel 3D

L'approche co-rotationnel 3D diffère peu du modèle éléments finis linéaires du point de vue des règles, car seule la méthode de calcul des forces présente une différence. Pour cela, nous avons créé une règle similaire à celle présentée sur la

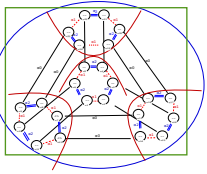
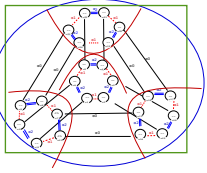
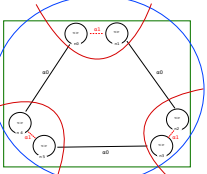
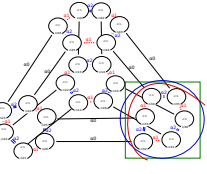
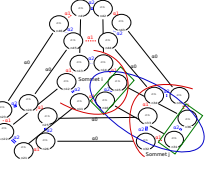
Interactions	Mailles	Motifs filtrés	Orbites des interactions	Orbites des forces
Linéaire	Tétraèdre		Volumes : $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$	Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$
Co-rotationnel	Tetraèdre		Volumes : $\langle \alpha_0, \alpha_1, \alpha_2 \rangle$	Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$
	Triangle		Faces : $\langle \alpha_0, \alpha_1 \rangle$	Sommets de face de volume : $\langle \alpha_1 \rangle$
Tenseurs élémentaires	Tétraèdre		Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$	Sommets de volume : $\langle \alpha_1, \alpha_2 \rangle$
			Extrémités d'arête d'un volume $\langle \alpha_2 \rangle$	Extrémités d'arête d'un volume $\langle \alpha_2 \rangle$

TABLE 6.2 – Modélisation des interactions continues

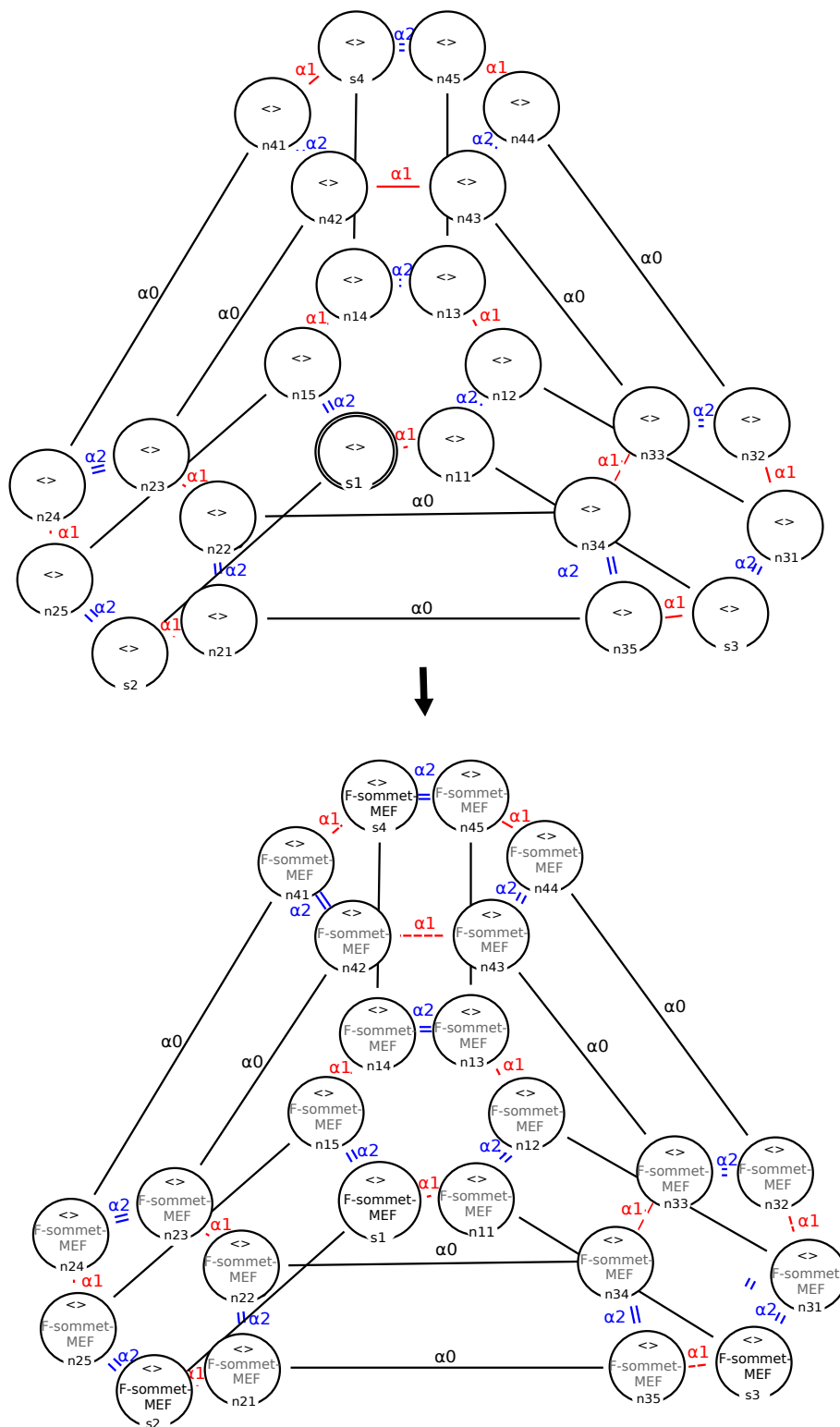


FIGURE 6.24 – Règle de calcul des forces en chaque sommet du tétraèdre.

figure 6.24. De même, dans les pré-traitements de cette règle, nous utilisons la même démarche que la méthode des éléments finis linéaires. Cependant, dans cette phase, une fois les quatre sommets (ordonnés) identifiés, nous déterminons le déplacement rigide subi par l'élément afin de l'ignorer et ne tenir compte que de la composante déformation par rapport à la forme initiale de l'élément.

Pour trouver la rotation, nous avons utilisé la décomposition QR [NPF05], décrite dans la section 2.2.2.3 (page 31). Il faut récupérer, tout d'abord, les positions (en coordonnées homogènes) des quatre sommets du tétraèdre pour construire une matrice de position de taille 4×4 . Ensuite, nous calculons une matrice de transformation en multipliant cette matrice de position par la matrice inverse des positions initiales, calculée en phase d'initialisation et stockée pour chaque tétraèdre. Puis, nous extrayons la matrice de rotation de l'élément $[R^{(e)}]$ en appliquant l'algorithme de Gram-Schmidt sur la matrice de transformation. La transposée de cette rotation est appliquée à la position des sommets pour déterminer leur déplacement par rapport à leur position au repos. Comme pour les éléments finis linéaires, les déplacements sont cumulés dans un vecteur 12 qui est multiplié à la matrice de raideur de l'élément pour obtenir le vecteur force global regroupant les quatre forces appliquées aux sommets de l'élément. La routine d'affectation des plongements forces `F-sommet-MEF` récupère les forces depuis le vecteur global des forces, mais leur applique la rotation de l'élément calculée avant de les plonger dans leur orbite. Cette approche « par sommet » permet d'éviter le calcul d'une matrice de rotation globale 12×12 telle que décrite dans la formule (2.33), page 32.

- Masses/tenseurs

Dans ce modèle physique, et dans le cas de volumes tétraédriques, certaines forces sont calculées en utilisant les tenseurs stockés sur les sommets et d'autres avec des tenseurs stockés dans les arêtes. Aussi, nous avons créé deux règles différentes. La première est présentée sur la figure 6.25.

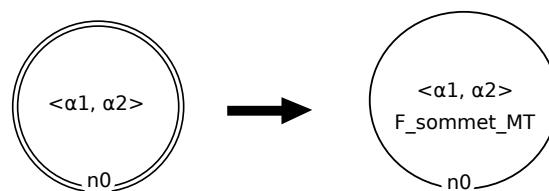


FIGURE 6.25 – Règle de calcul des forces appliquées par un sommet sur lui-même.

Nous avons filtré dans le membre gauche un sommet de volume (orbite $\langle \alpha_1, \alpha_2 \rangle$). Dans le membre de droite, nous calculons les forces en multipliant le tenseur associé à ce sommet par son vecteur déplacement et les stockons dans les plongements `F-sommet-MT` portés par les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$. Comme cette règle ne s'applique que sur un seul sommet de volume, il faut créer un script qui permet

de calculer ces forces sur tous les sommets de volume de l'objet. Cette règle peut être utilisée dans le cas d'autres volumes, par exemple les hexaèdres.

Ensuite, nous avons créé la règle présentée sur la figure 6.26, qui permet de calculer les forces appliquées par un sommet de volume sur un autre. Dans le membre gauche, nous avons filtré, explicitement, une arête de volume avec ses deux extrémités. Dans le membre droit, nous avons calculé les forces appliquées sur le sommet représenté par le nœud $n0$ en récupérant le tenseur associé déjà stocké sur sa sous-orbite $\langle \alpha_2 \rangle$. Nous le multiplions par le vecteur déplacement du sommet représenté par le nœud $n1$ et le stockons dans le plongement $F\text{-arête-MT}$. Pour la force appliquée sur le nœud $n1$, nous récupérons le tenseur stocké dans le nœud $n0$, puis calculons son transposé et on le multiplie par le vecteur déplacement du sommet représenté par le nœud $n0$. Finalement nous le stockons dans le plongement $F\text{-arête-MT}$ du nœud $n1$.

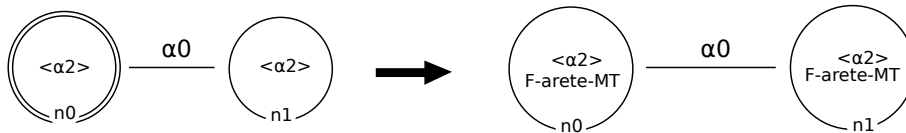


FIGURE 6.26 – Règle de calcul des forces appliquées par un sommet de volume sur un autre.

- Co-rotationnel 2D

Nous avons adapté la méthode du co-rotationnel pour simuler des objets en 2D. Ici, nous considérons des mailles uniquement triangulaires. Pour cela nous avons adapté la règle déjà proposée dans le cas 3D, nous la présentons sur la figure 6.27. Le membre gauche de cette règle correspond à un triangle décrit de façon explicite. Dans le membre droit, les forces calculées sont à stocker dans les plongements $F\text{-sommet-MEF2D}$ portés par les orbites sommets de face de volume. Pour le calcul des forces, nous exploitons une caractéristique importante de la décomposition de Gram-Schmidt à la base de la décomposition QR : les axes peuvent être traités dans l'ordre que l'on choisit, ce qui permet de sélectionner des directions de traitement privilégiées pour définir la matrice de rotation qui transforme l'objet de la configuration déformée à la configuration au repos. Aussi, le premier axe que l'on choisit de traiter est celui de la normale au triangle déformé. Nous obligeons cet axe à se transformer en axe z lors de la rotation. Ainsi une fois cette rotation effectuée, le triangle déformé se retrouve obligatoirement dans le plan (x, y) (modulo une translation selon z). Il suffit alors de désigner deux autres axes (l'un est une arête du triangle et l'autre est choisi orthogonal à cette arête) pour calculer la rotation, dans le plan (x, y) , permettant de minimiser la déformation du triangle par rapport à la configuration au repos. Une fois la rotation 3D totalement définie, nous appliquons cette rotation sur les déplacements des sommets. Nous multiplions la matrice de rigidité stockée dans la face par le vecteur qui regroupe les déplacements pour obtenir

le vecteur regroupant toutes les forces des sommets, dans le repère (x, y) . Il suffit d'appliquer la rotation inverse aux quatre forces constituant ce vecteur global, afin d'obtenir les forces pour chaque sommet, dans le repère du triangle déformé.

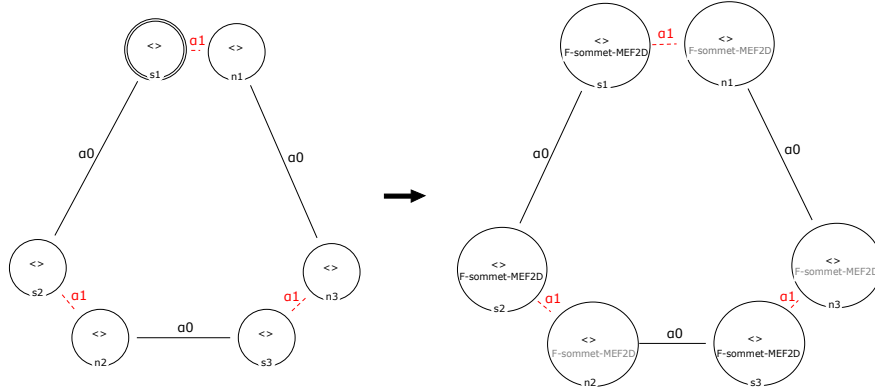


FIGURE 6.27 – Règle de calcul des forces appliquées sur les sommets par le corotationnel 2D.

Il faut noter que cette règle ne s'applique que sur le brin de chaque face qui est marqué comme premier sommet. Elle nécessite l'utilisation d'un script qui parcourt l'ensemble des premiers brins de l'objet.

Maintenant que toutes les règles de calcul des forces appliquées par les interactions sont créées, il s'agit désormais de mettre en œuvre les étapes de résolution des équations du mouvement.

6.3.3 Résolution des équations du mouvement

Nous avons utilisé la même boucle de simulation pour la simulation des différents objets en 2D et 3D, avec différents modèles mécaniques. Nous avons créé une règle qui permet de calculer l'accélération de chaque particule, une règle pour calculer les nouvelles vitesses et une dernière pour calculer les nouvelles positions. Comme ce sont des règles de mise à jour des valeurs de plongements, nous avons décidé de créer des règles qui s'appliquent sur toute la composante connexe et permettent de calculer les valeurs des plongements (portés par les orbites sommets) de tous les sommets de la composante connexe, en même temps. Les règles à appliquer sont présentées sur la figure 6.28.

Pour calculer les accélérations, nous collectons les forces stockées dans les sous-orbites de l'orbite sommet concernée ($\langle \rangle, \langle \alpha_1 \rangle, \langle \alpha_2 \rangle, \langle \alpha_1, \alpha_2 \rangle \dots$). Ensuite, nous ajoutons, si elles existent, des forces externes (la pesanteur, le vent, la viscosité ambiante...). Puis, nous divisons la force par la masse totale du sommet qui est un des plongements associés aux sommets et stockons le résultat dans le plongement *Acceleration* porté par l'orbite sommet. Les nouvelles vitesses et positions sont

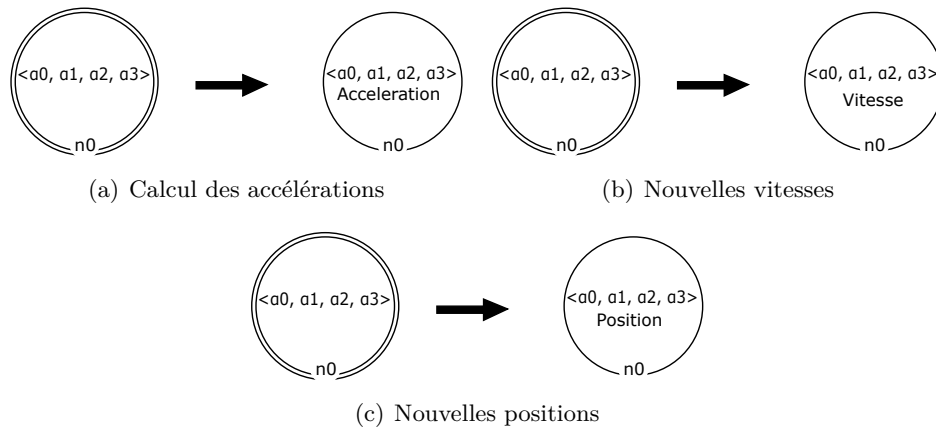


FIGURE 6.28 – Règles de mise à jour des accélérations, vitesses et positions de particules.

calculées en se basant sur les formules d’Euler symplectique (equation (2.40), page 35) et stockées respectivement dans les plongements *Vitesse* et *Position*. Ces trois règles s’appliquent à chaque pas du temps après les règles nécessaires au calcul des forces.

Pour chaque type de simulation, nous avons créé un script qui enchaîne les différentes règles nécessaires à la simulation du modèle mécanique choisi. Dans ces scripts, toutes les composantes connexes de l’objet sont parcourues, et l’ensemble des règles sont appelées pour chaque composante. Il faut noter aussi que nous pouvons simuler un objet en accumulant plusieurs interactions mécaniques. Par exemple, si nous voulons simuler un objet avec une méthode des éléments finis et la méthode des masses/ressorts uniquement sur la surface de l’objet, nous devons sélectionner dans ce script les règles qui calculent toutes les forces utilisées par le modèle.

6.4 Optimisation des calculs

Dans le chapitre précédent, nous avons proposé une modélisation des interactions et de la masse sous la forme de données élémentaires. Néanmoins, nous remarquons que la modélisation d’interactions élémentaires implique des redondances de calcul, en particulier lorsque plusieurs interactions lient les mêmes sommets.

Nous proposons dans cette section quelques optimisations pour avoir de meilleurs temps de simulation. Nous rappelons, tout d’abord, que nous avons déjà proposé une optimisation pour la masse dans la section 5.1.2 (page 105). En effet, nous avons décidé de stocker la masse dans le plongement *Masse-coin* porté par les orbites sommets de face (2D)/sommets de volume (3D). Comme nous avons besoin de toute la masse d’un sommet lors du calcul de l’accélération de la particule associée et que nous ne souhaitons pas recalculer la somme des masses-coins à chaque pas de temps, nous avons placé cette somme dans le plongement *Masse-sommet* porté par l’orbite sommet $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$.

6.4.1 Recensement des redondances

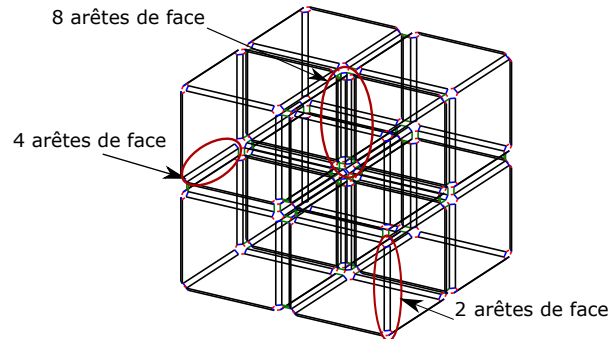


FIGURE 6.29 – Mise en évidence de redondance pour les ressorts d'élongation.

Nous dressons dans cette partie un inventaire des redondances d'interactions. Dans une première étape, nous allons reprendre, les différentes interactions modélisées dans le chapitre de modélisation une par une et étudier leurs redondances. Commençons tout d'abord par les ressorts.

En ce qui concerne les ressorts d'élongation élémentaires, nous avons choisi de les faire porter par les arêtes de face de volume (orbites $\langle \alpha_0 \rangle$). En 2D, pour deux faces adjacentes, il existe deux arêtes de faces (deux ressorts élémentaires) formant l'arête complète, alors qu'en 3D, on dénombre $2n$ arêtes de face de volume pour n volumes adjacents autour d'une arête. Par conséquent, il faut calculer les forces de ces $2n$ ressorts, à chaque pas de simulation, alors que ces ressorts concernent les deux mêmes sommets : la différence entre la distance qui les sépare à un instant donné et la distance au repos est identique pour tous ces ressorts. Or, les calculs de distances sont les parties les plus coûteuses du calcul. Par exemple, comme montré sur la figure 6.29, une arête partagée par quatre volumes possède huit arêtes de face, et une autre, sur le bord de l'objet et partagée par deux volumes possède quatre arêtes de faces.

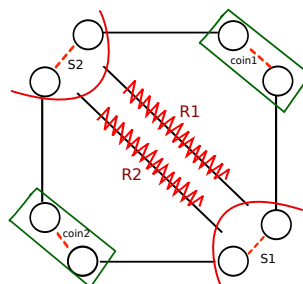


FIGURE 6.30 – Mise en évidence des redondances pour les ressorts de cisaillement surfacique anisotrope.

Le deuxième type d'interactions modélisé dans le chapitre précédent est celui des ressorts de cisaillement surfacique. Dans le cas de cisaillement surfacique isotrope,

il n'existe pas de redondances en 2D, alors que c'est le cas pour les ressorts de cisaillement surfacique anisotrope portés par les sommets de face ($\langle \alpha_1 \rangle$). Comme on peut le constater sur la figure 6.30, pour un quadrilatère, il existe deux coins de faces différents qui correspondent au même couple de sommets opposés S1 et S2. Par conséquent, le même couple de points est traité à deux reprises, car les deux ressorts R1 et R2 concernent les mêmes sommets.

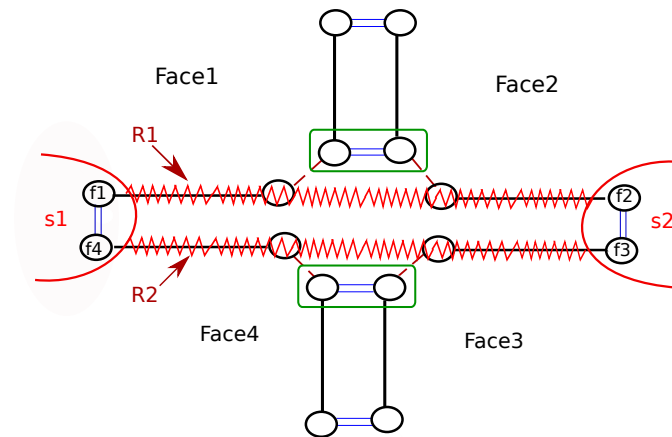


FIGURE 6.31 – Mise en évidence de redondance pour les ressorts linéaires de Provot.

En 2D, la redondance de calcul se rencontre également dans le cas des ressorts linéaires de Provot. En effet, comme montré sur la figure 6.31, ces ressorts lient deux mêmes sommets S1 et S2, mais l'origine de l'interaction ne concerne pas le même couple de faces. Par exemple, le ressort R1 contrôle l'articulation entre les faces 1 et 2 et R2 fait de même pour les faces 3 et 4.

En 3D, une redondance pour les ressorts de cisaillement surfacique isotropes ou anisotropes se rencontre également, dans le cas de deux volumes adjacents par une face, comme montré sur la figure 6.32 (face jaune). Les ressorts de cisaillement d'un même couple de sommets se traitent deux fois dans le cas isotrope et quatre fois dans le cas anisotrope (quatre orbites $\langle \alpha_1 \rangle$ désignent la même diagonale).

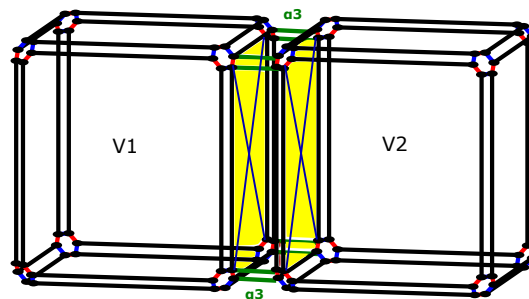


FIGURE 6.32 – Mise en évidence des redondances pour les ressorts de cisaillement surfacique en 3D.

La redondance de calcul se rencontre, également, dans le cas de ressorts de cisaillement volumique anisotrope. De la même manière que les cisaillement surfacique anisotrope, pour un hexaèdre, il existe deux coins de volumes différents qui correspondent au même couple de sommets opposés, comme montré sur la figure 5.20 (page 119). Par conséquent, le même couple est traité deux fois.

En ce qui concerne les interactions des méthodes des éléments finis, les redondances apparaissent uniquement pour le modèle de masses/tenseurs, lors du calcul des forces sur les extrémités d'arêtes et les sommets de volumes. En effet, chaque volume incident à un sommet apporte une force à ce sommet, mais toutes ces forces se basent sur le même déplacement de sommet. Il en résulte de nombreuses multiplications matrice-vecteur (n multiplications si n volumes incidents au sommet) sur un même vecteur. De même pour les tenseurs d'arête, où n multiplications d'un même vecteur déplacement sont effectuées si n volumes sont incidents à une même arête. Il faut noter que la méthode originale des masses/tenseurs [CDA00] préconise de précalculer la somme des tenseurs en chaque sommet et en chaque arête (orientée).

6.4.2 Analyse des redondances

Dans les paragraphes précédents, nous avons analysé la redondance pour les différents types d'interactions. Dans cette sous-section, nous discutons les différentes redondances rencontrées et les optimisations faites pour obtenir des simulations plus rapides. Signalons tout d'abord que certaines redondances peuvent être facilement évitées. C'est le cas lorsqu'il existe plusieurs lieux de stockage pour stocker une même interaction. Dans ce cas, il suffit de ne définir l'interaction que sur un seul lieu parmi les lieux possibles et de ne stocker aucune interaction sur les autres¹. Par exemple, on ne stocke un ressort de cisaillement surfacique anisotrope que dans un seul coin d'un quadrilatère et les ressorts de cisaillement volumique anisotrope que dans un seul sommet du volume. Si la face (resp. le volume) ne subit pas de modification, les orbites correspondant $\langle \alpha_1 \rangle$ (resp. $\langle \alpha_1, \alpha_2 \rangle$) ont le même cycle de vie.

Dans d'autres cas, plusieurs interaction, d'origines différentes, sont placées entre deux mêmes sommets. Typiquement, c'est le cas des ressorts d'élongation élémentaires concernant la même arête, en particulier en $3D$ où $2n$ ressorts élémentaires se cumulent pour n volumes incidents. Dans ce cas, plutôt que de considérer $2n$ ressorts nous considérons un ressort équivalent dont la raideur est la somme des raideurs des ressorts élémentaires (puisqu'ils sont placés en parallèles). Il faut alors créer un plongement dit « de cumul » pour stocker les propriétés du ressort équivalent. En considérant cette optimisation, le lieu de stockage des forces est également à redéfinir puisqu'il concerne un cumul de ressorts et un nouveau plongement est à créer. Comme les orbites de stockage changent, il faut créer des règles spécifiques pour gérer les interactions avec les plongements de cumul (des exemples de règles sont

1. Jerboa oblige toutes les orbites à supporter un plongement si un plongement a été défini pour le type d'orbite en question. Dans ce cas, il suffit d'indiquer que l'interaction n'est pas active, via, par exemple, un booléen.

données juste après cette discussion). Cependant, lorsqu'une modification entraîne un dédoublement d'un (ou plusieurs sommets), les ressorts ne se cumulent plus, car ils ne sont plus nécessairement en parallèle (ils ne concernent plus obligatoirement le même couple de sommets). Dans ce cas, il nous faut recalculer le ou, plus souvent, les plongement(s) de cumul de la zone concernée par la modification. Ce même cas de redondance se rencontre également avec le modèle masses/tenseurs. Nous choisissons de stocker, en chaque sommet, la somme des tenseurs stockés dans les sommets de volume qui forment cette orbite. De même, la somme des tenseurs élémentaires stockés dans les extrémités d'arêtes de volume (orbites $\langle \alpha_2 \rangle$) est stockée dans l'extrémité d'arête correspondante (orbites $\langle \alpha_2, \alpha_3 \rangle$). Ces tenseurs cumulés sont à recalculer après une modification topologique éclatant les sommets ou les arêtes supports.

Finalement, nous considérons séparément le cas des redondances faibles, typiquement si la redondance vaut deux. Ce cas apparaît très souvent. Nous pouvons citer, par exemple, les ressort d'élongation en 2D, les ressorts de courbure linéaires de Provot (comme montré sur la figure 6.31), les ressorts de cisaillement isotropes surfaciques en 3D. Une telle redondance apparaît également, pour les ressorts de cisaillement isotrope volumique car nous utilisons une règle qui induit un calcul séparé pour chaque extrémité du ressort (voir figure 6.19). Dans le cas d'une redondance de deux, la lourdeur induite par l'utilisation d'un plongement de cumul est à prendre en considération. En effet, après une modification topologique, deux ressorts qui agissaient sur le même couple de sommets peuvent ne plus agir sur le même couple (suite à une scission de sommet). Le calcul des ressorts élémentaires est alors préféré puisqu'il n'y a plus de redondance. Une première attitude face à cette redondance consiste à accepter qu'un même calcul d'interaction soit effectué deux fois. Si, néanmoins, nous estimons que cette redondance n'est pas acceptable, il est possible de mettre en œuvre une optimisation grâce à une règle qui prenne en compte les deux ressorts simultanément, calcule, à la volée, la somme des raideurs de deux ressorts (donc qui n'exploite aucun plongement de cumul). Notons que, dans ce cas, il faut tout de même créer des orbites de cumul des forces, qu'il faut penser à mettre à 0 après une modification topologique (car ces plongements peuvent ne plus être utilisés par la suite, mais entrent tout de même en compte lors du bilan des forces).

En résumé, nous avons différencié trois cas de redondance, celui que nous avons qualifié d'évitable, en ne stockant l'interaction qu'en un seul lieu. Dans le cas de redondance supérieur à deux, nous avons ajouté des plongements de cumul pour les interactions et les forces qu'elles provoquent. Finalement, si la redondance vaut deux, diverses approches sont possibles. Dans la suite, nous avons fait le choix d'utiliser tout de même un plongement de cumul dans ce cas.

6.4.3 Mise en œuvre des optimisations

Pour les interactions dont nous avons choisi de gérer explicitement le cumul, nous avons ajouté des plongements décrivant l'interaction équivalente, définie par la somme des raideurs et des éventuels coefficients d'amortissement, la longueur au repos étant celle partagée par toutes les interactions. Nous avons également ajouté des plongements de forces et, par conséquence, les règles à appliquer ont été adaptées aux nouveaux plongements. Dans une première étape, il faut trouver l'orbite adéquate pour le plongement de cumul et pour les forces.

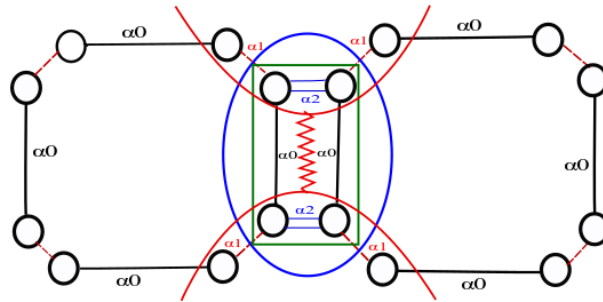


FIGURE 6.33 – Ressort d'élongation cumulé

Par exemple pour les ressorts d'élongation, nous avons proposé d'ajouter le plongement `R-elong-cumul` porté par l'orbite arête (2D/3D). La valeur de ce dernier est calculée, à partir du plongement `R-elong-lem`, en faisant la somme de toutes les raideurs des ressorts élémentaires stockés sur les arêtes de face ($\langle \alpha_0 \rangle$) constituant une même arête ($\langle \alpha_0, \alpha_2 \rangle$ en 2D ou $\langle \alpha_0, \alpha_2, \alpha_3 \rangle$ en 3D).

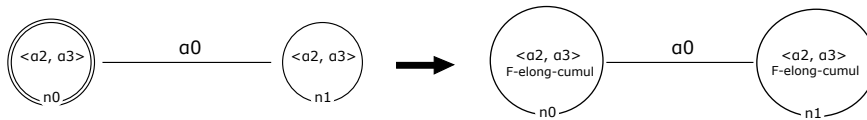
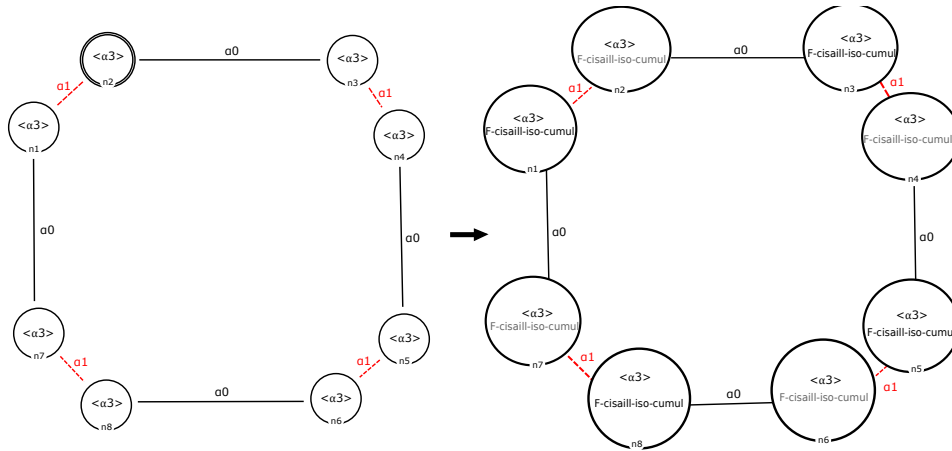


FIGURE 6.34 – Règle de calcul des forces des ressorts d'élongation cumulés.

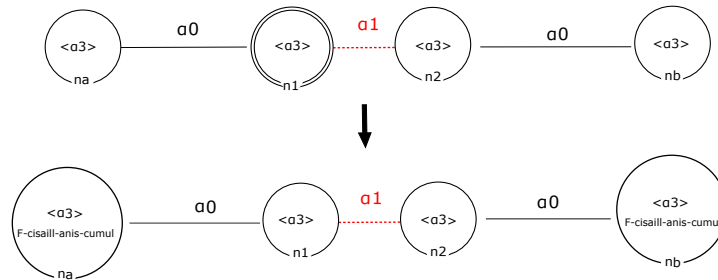
La figure 6.33 illustre cette modélisation. L'orbite de stockage des forces qui est l'intersection entre le sous-graphe bleu et le sous-graphe rouge est l'orbite extrémité d'arête ($\langle \alpha_2 \rangle$ en 2D et $\langle \alpha_2, \alpha_3 \rangle$ en 3D). Cette solution influe sur la règle à appliquer, que l'on présente sur la figure 6.34. Dans le membre de gauche, le filtre correspond à une arête, bâtie autour de la liaison α_0 mais incluant les orbites $\langle \alpha_2, \alpha_3 \rangle$, les deux extrémités n_0 et n_1 indiquent les sommets d'application. Dans le membre de droite, une force est calculée et stockée dans le plongement `F-elong-cumul` porté par les orbites $\langle \alpha_2, \alpha_3 \rangle$ du nœud n_0 et son opposé dans celui du nœud n_1 .

À noter que cette règle prévue pour la 3D peut également être utilisées en 2D, car la liaison α_3 n'est pas utilisée en 2D, mais existe dans le modèle (elle est définie comme lien identité pour tous les brins). De la même façon, pour les règles qui

suivent, nous les avons conçues pour la 3D mais elles s'appliquent également en 2D.



(a) Cas isotrope



(b) Cas anisotrope

FIGURE 6.35 – Règle de calcul des forces des ressorts de cisaillement surfacique cumulés en 3D.

Pour les ressorts de cisaillement isotrope surfacique en 3D, nous avons ajouté le plongement **R-cisaill-iso-cumul** porté par les orbites faces $\langle \alpha_0, \alpha_1, \alpha_3 \rangle$ dans lequel nous stockons le cumul des raideurs des ressorts élémentaires stockés sur les faces de volume. Les forces appliquées par ces ressorts doivent être stockées sur les orbites sommets de face $\langle \alpha_1, \alpha_3 \rangle$. Par analogie, un plongement **R-cisaill-anis-cumul** est ajouté pour les ressorts de cisaillement surfacique anisotrope, il est porté par les orbites sommets de face $\langle \alpha_1, \alpha_3 \rangle$ et les forces sont stockées sur les orbites extrémités d'arêtes de face $\langle \alpha_3 \rangle$. Les règles que nous avons créées pour prendre en compte ces optimisations ressemblent aux règles 6.17 (page 151) pour les ressorts isotropes et 6.18 (page 152) pour les ressorts anisotropes. Dans les nouvelles règles, nous avons ajouté les liaisons α_3 dans les nœuds (dans la variable topologique) pour prendre en compte les deux faces en parallèle et stocker les forces dans les plongements de cumul **F-cisaill-iso-cumul** et **F-cisaill-ani-cumul** comme montré sur les règles présentées sur la figure 6.35.

En ce qui concerne les ressorts de courbure linéaires de Provot, une solution de cumul a été envisagée, mais l'orbite de plongement de l'interaction de cumul s'est

révélée délicate à déterminer. En effet, une première approche consiste à cumuler les deux ressorts décrits sur la figure 6.31 dans le sommet central de l’articulation. Néanmoins, ce sommet est aussi l’articulation de deux ressorts, placés verticalement, sur la même figure. Si les paramètres des ressorts de Provot sont stockés au sommet, il faut supposer que ce sont les mêmes pour les ressorts « horizontaux » et « verticaux », donc que le modèle est isotrope. Nous pouvons, également, stocker le plongement de cumul sur l’une des orbites $\langle \alpha_2 \rangle$ qui stockent déjà un ressort de Provot élémentaire, mais dans ce cas, nous privilégions l’une des articulations au détriment de l’autre. En outre, lors d’une modification topologique, si l’articulation qui ne stocke par le ressort de cumul est supprimée, il faut penser à répercuter cette suppression sur le plongement de cumul stockée sur l’autre articulation. Cette approche contrevient à notre hypothèse de stockage des interactions sur des orbites impliquées dans l’origine des interactions. En réalité, l’orbite idéale pour stocker le cumul est l’orbite $\langle \alpha_2, \alpha_1 \circ \alpha_2 \circ \alpha_1 \rangle$, permettant de regrouper en une seule orbite les brins concernés par les orbites $\langle \alpha_2 \rangle$ des ressorts élémentaires. Malheureusement, les orbites basées sur des compositions de liens α ne sont pas implantées dans Jerboa (et n’existent pas non plus dans d’autres bibliothèques de manipulation de G -cartes, leur utilisation étant très rare, contrairement au cas des cartes orientées). Par conséquent, nous n’avons pas fait d’optimisation pour ces ressorts et nous avons accepté la redondance.

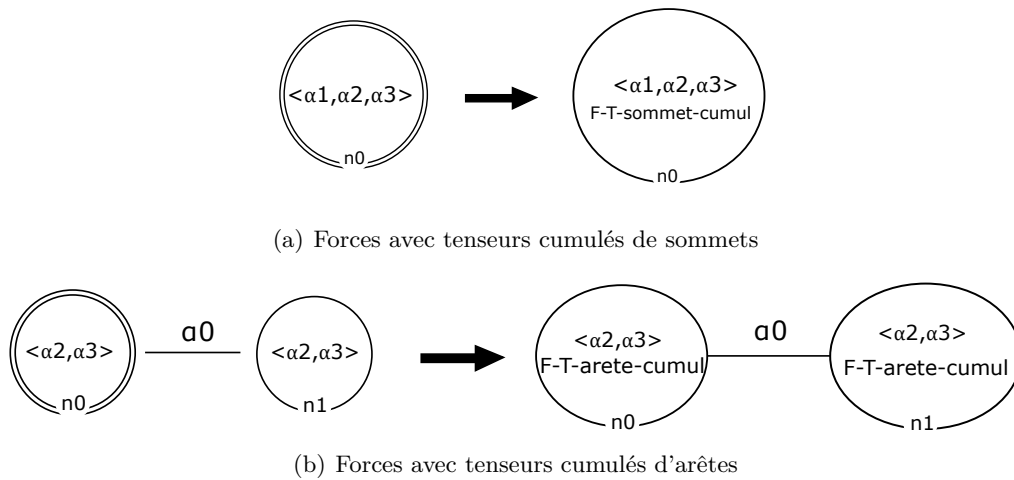


FIGURE 6.36 – Règles de calcul des forces des tenseur cumulés.

Pour le modèle masses/tenseurs, nous avons également ajouté des plongements de cumul. Comme chaque volume adjacent à un sommet fournit à ce dernier un tenseur $[K_{ii}^{(e)}]$, nous avons sommé ces tenseurs (qui sont stockés dans les sommets de volume $\langle \alpha_1, \alpha_2 \rangle$) et nous les avons stockés dans le plongement **T-sommet-cumul** porté par les orbites sommets $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$. De même, comme tous les volumes adjacents à une arête donnée fournissent des tenseurs aux extrémités de l’arête, nous avons ajouté un plongement **T-arete-cumul** qui correspond à la somme de tous les tenseurs $[K_{ij}^{(e)}]$ et porté par la sous-orbite $\langle \alpha_2, \alpha_3 \rangle$ du sommet i (autrement

dit l'extrémité i de l'arête). Pour les forces, nous avons ajouté deux plongements **F-T-sommet-cumul** et **F-T-arete-cumul** qui sont portés respectivement par les orbites sommets $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$ et extrémités d'arêtes $\langle \alpha_2, \alpha_3 \rangle$. Maintenant que les plongements sont définis, nous avons adapté les règles 6.25 et 6.26 créées dans le cas élémentaire. Nous avons ajouté les liaisons α_3 , comme montré sur la figure 6.36, pour prendre en compte toute l'arête/sommet et non pas uniquement l'arête de face et le sommet de volume. Le calcul se fait de la même manière évoquée précédemment dans le cas élémentaire, mais en utilisant les tenseurs cumulés.

Ces plongements de cumul, en soi, ne sont pas indispensables, car les plongements élémentaires sont suffisants dans notre approche pour simuler n'importe quel objet en 2D ou en 3D et ne visent qu'à optimiser les calculs. Néanmoins, l'utilisation des plongements de cumul entraîne certaines conséquences. En particulier, si la topologie change, il faut mettre à jour les plongements de cumul touchés par une modification topologique, ce qui peut augmenter le temps de traitement de l'opération. L'utilisation des plongements de cumul ne se justifie que pour des applications à temps critique. Quelques comparaisons de temps de calcul avec et sans optimisation sont données dans la section suivante.

6.5 Résultats

Nous présentons dans cette section quelques résultats d'animation, pour des maillages homogènes (un seul type de maille) ou mixtes en 2D et 3D avec différents modèles mécaniques. Des vidéos de démonstrations sont également accessibles à l'URL : <http://xlim-sic.labo.univ-poitiers.fr/jerboa/doc/japhy/>. Nous commençons par présenter quelques résultats de simulation en utilisant le modèle masses/ressorts (voir la figure 6.37). Tous ces objets sont simulés avec les différents types de ressorts présentés dans ce chapitre.

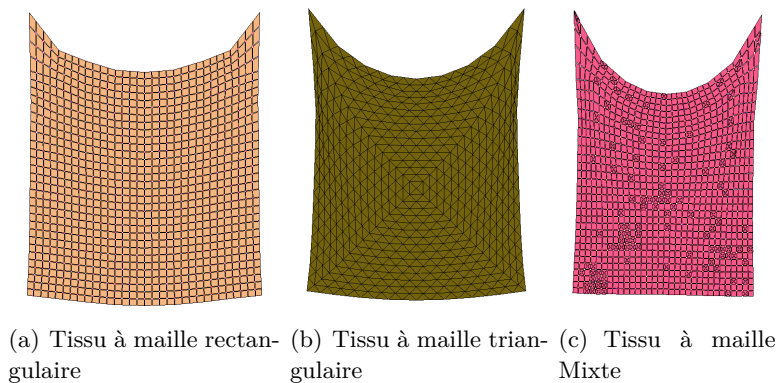


FIGURE 6.37 – Simulations de différents types de tissus avec le modèle masses/ressorts.

La figure 6.37(a) montre la simulation d'un tissu rectangulaire de taille $1m \times 1m$

qui contient 1024 éléments rectangulaires en utilisant des ressorts d'élongation élémentaires, des ressorts de cisaillement et des ressorts des courbure linéaire de Provot. Avec le même modèle mécanique, la figure 6.37(b) présente la simulation d'un tissu de taille $1m \times 1m$ dont les mailles sont triangulaires et en utilisant les ressorts d'élongation élémentaires avec des ressorts de courbure angulaires. Finalement la figure 6.37(c) présente un exemple de tissu à mailles mixtes (triangles et rectangles). Dans ces modèles, les arêtes portent des ressorts d'élongation, des ressorts de cisaillement surfacique isotrope et des ressorts linéaires de Provot.

Nous avons également expérimenté notre modèle en 3D et quelques résultats sont présentés sur la figure 6.38.

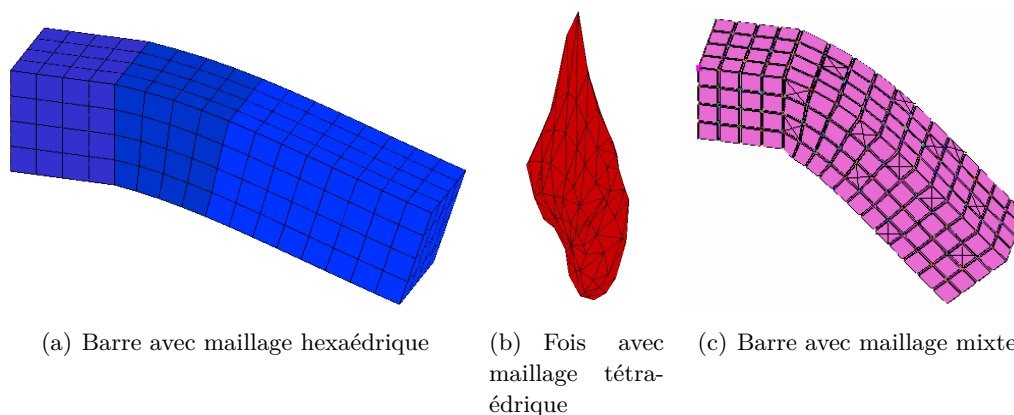


FIGURE 6.38 – Simulations de différents types d'objet 3D avec modèle masse/ressort.

La figure 6.38(a) montre le résultat de simulations d'une barre représentée par un maillage hexaédrique (de 64 éléments). La figure 6.38(b) présente la simulation d'un foie représenté avec un maillage tétraédrique constitué de 597 éléments. La figure 6.38(c) présente la simulation d'une barre représentée par un maillage mixte (des hexaèdres avec des faces rectangulaires et des autres triangulaires et des tétraèdres).

Notre plate-forme permet également de simuler des objets en 2D et 3D avec des modèles basés sur la mécanique continue. Quelques résultats sont présentés sur la figure 6.39. Dans cette figure, nous présentons la simulation du même objet, une barre avec le modèle des éléments finis en petits déplacements (figure 6.39(a)), avec le modèle du co-rotationnel (figure 6.39(b)) et finalement avec le modèle masses/tenseurs (figure 6.39(c)).

Finalement, la figure 6.40 montre la simulation d'un tissu à mailles triangulaires 2D en utilisant la méthode du co-rotationnel.

Il faut noter que la simulation d'objets plus complexes est possible avec notre modèle. En outre le degré d'exactitude des résultats est équivalent aux modèles originaux puisque les calculs sont les mêmes. Ces résultats sont conformes aux simulations obtenues sans modèle topologique et ont le même degré de précision.

Comme nous avons implanté différents types d'interactions, en incluant des optimisations pour certaines d'entre elles, nous présentons dans la table 6.5 quelques

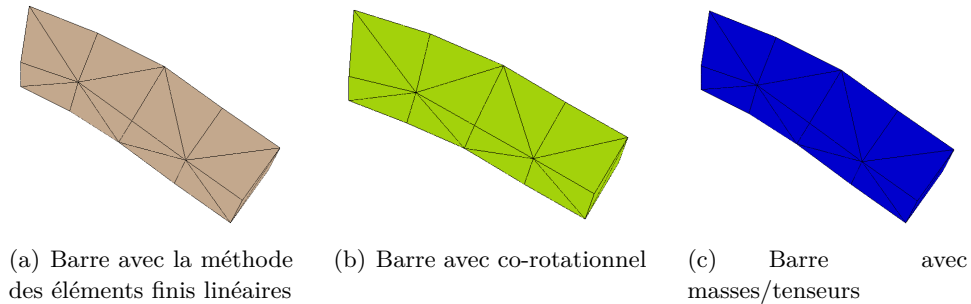


FIGURE 6.39 – Simulations d’une barre tétrédrique 3D avec différents modèle continus.

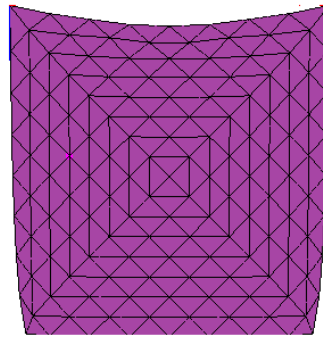


FIGURE 6.40 – Simulation d’un tissu avec la méthode des éléments finis.

mesures.

Pour comparer entre eux les ressorts d’élongation élémentaires et les ressorts d’élongation cumulés, nous avons utilisé, en 2D, un tissu à mailles rectangulaires de 256 mailles. L’utilisation des ressorts d’élongation élémentaires donne un temps de simulation 1,28 fois plus coûteux. Nous pouvons remarquer que le nombre des ressorts est le double dans le cas élémentaire mais le temps de simulation est supérieur uniquement de 28 %. Nous avons simulé le même objet avec des ressorts d’élongation et nous avons ajouté dans ce cas des ressorts de cisaillement surfacique isotrope et dans un autre cas des ressorts de cisaillement surfacique anisotrope. La simulation avec ces derniers donne un temps de simulation 1,25 fois plus coûteux que les ressorts de cisaillement isotropes. À la troisième ligne de ce tableau, nous avons comparé la simulation du même objet avec des ressorts de courbure de Provot et les ressorts de courbure angulaires. Ce tissu est simulé avec des ressorts d’élongation et de cisaillement isotropes. Nous avons trouvé que les ressorts angulaires sont plus coûteux avec un rapport de 1,5, alors qu’aucune optimisation n’a été utilisée pour limiter la redondance des ressorts de Provot.

En 3D, pour comparer l’utilisation des ressorts de cisaillement volumique et les

	Le plus rapide	Rapport temps plus long/ plus rapide
Élongation élémentaire vs élongation cumulé	Élongation cumulé	1,28
Cisaillement surfacique isotrope (en 2D) vs cisaillement surfacique anisotrope (en 2D)	Cisaillement de face isotrope	1,25
Courbure de Provot vs courbure angulaire	Courbure provot	1,5
Cisaillement volumique isotrope vs cisaillement élémentaire surfacique isotrope	Cisaillement élémentaire sur les faces du volume	1,2
Cisaillement cumulé sur les faces du volume vs cisaillement élémentaire sur les faces du volume	Cisaillement cumulé sur les faces du volume	1,4
Masses/tenseurs cumulés vs masses/tenseurs non cumulés	Masses/tenseurs cumulés	1,55

TABLE 6.3 – Rapports de temps de calcul entre modèles d’interaction similaires ou en version optimisée

ressorts de cisaillements isotropes sur les faces de volumes, nous avons utilisé un objet à mailles hexaédriques de 32 éléments, dotées de ressorts d’élongations. La simulation avec des ressorts de cisaillement volumique donne un temps de simulation de 1,2 fois plus coûteux qu’avec un cisaillement isotrope sur les faces des volumes. On peut remarquer que même sans utilisation de plongements de cumul les ressorts de cisaillement sur les faces sont plus rapides à simuler que ceux à l’intérieur des volumes. En effet, rechercher les composantes connexes pour appliquer une règle sur toute la composante s’avère moins efficace que le faire sur des orbites plus petites. Nous avons, également, utilisé ce même objet avec des ressorts d’élongation pour comparer l’utilisation des ressorts de cisaillement de face élémentaires et cumulés en 3D. Comme il y a moins de forces à calculer pour les ressorts de cisaillements cumulés sur les faces, ce modèle est plus rapide qu’avec les cisaillements élémentaires sur les faces. Enfin, pour mesurer l’impact des plongements de cumul avec la méthode des masses/tenseurs, nous avons utilisé une barre tétraédrique constituée de 20 éléments. Avec les plongements élémentaires, la simulation est 1,55 plus coûteuse

que l'utilisation de la méthode avec les plongements de cumul.

Pour résumer, il faut noter que les temps de simulation en utilisant des plongements élémentaires ne génère pas plus de 25 % de temps de calcul supplémentaire par rapport aux plongements cumulés, alors que le nombre d'interactions à calculer est souvent le double ou plus. Ceci signifie que le temps de traitement des règles en soi n'est pas négligeable.

Nous avons donné uniquement des rapports de temps de calcul parce que les temps de calcul absolus ne sont pas significatifs. Jerboa est un outil de prototypage, sans optimisation spécifique, que nous avons utilisé pour implanter les règles. Nous l'avons utilisé comme un outil pour montrer l'effectivité de notre approche. Nous devons noter que le temps du prototypage en utilisant Jerboa est incomparable avec le temps de codage habituel que les programmeurs dépensent pour programmer les différentes interactions que nous avons modélisées. Par exemple, seuls deux jours ont été nécessaires pour étendre le modèle masses/ressorts de la 2D à la 3D. La plupart du temps a été consacré à la construction des objets à simuler (par exemple la barre en 3D), avec le langage à base de règles, et non pas la conception des interactions. En effet, une seule règle a dû être ajoutée pour les ressorts de cisaillement volumique. Un temps équivalent a été consacré à la généralisation par rapport au type de mailles (rectangulaires vers triangulaires).

6.6 Conclusion

Dans ce chapitre, nous avons exposé tout d'abord la stratégie d'initialisation des objets à simuler qui peuvent être des objets chargés directement à partir d'un fichier qui décrit le maillage ou bien créés par le langage à base de règles. Ensuite, nous avons présenté sur les différentes règles créées avec Jerboa pour la simulation de ces objets. Nous nous sommes focalisés, tout d'abord, sur les règles utilisées pour calculer les forces appliquées par les différentes interactions modélisées dans le chapitre précédent. Puis, les règles nécessaires au calcul des accélérations, vitesses et nouvelles positions de chaque particule de l'objet. Dans la dernière partie, nous avons mis l'accent sur les redondances existant dans quelques règles et nous avons proposé diverses solutions pour éviter ou limiter cette redondance. Finalement, nous avons présenté quelques résultats de simulation utilisant Jerboa et son langage à base de règles. Nous avons conclu que l'utilisation des interactions élémentaires dans la plupart des cas ne génère que 25 % de temps supplémentaire. Néanmoins, il faut noter que l'utilisation de ces interactions élémentaires est très intéressante si l'objet subit des transformations simples de type décousure entre deux éléments. Dans ce cas, les plongements ne sont pas à mettre à jour, car leurs orbites ne dépendent pas des liaisons modifiées. De ce point de vue, les transformations apparaissent alors comme purement topologiques et ne traitent pas de mécanique. Dans le prochain chapitre, nous nous intéressons aux transformations topologiques et plus précisément aux phénomènes de déchirures et fracture qui peuvent apparaître dans un objet.

Transformation topologiques

Sommaire

7.1	Introduction	179
7.2	Principe	180
7.3	Modification topologiques	183
7.3.1	Séparation des éléments	184
7.3.2	Suppression d'éléments	186
7.3.3	Éclatement d'un sommet	187
7.3.4	Suppression d'arête	188
7.4	Critères	189
7.4.1	Critères atomiques	189
7.4.2	Critères cumulatifs	191
7.4.3	Critères dédiés à la séparation d'éléments	192
7.5	Exemples de résultats	194
7.6	Conclusion	198

7.1 Introduction

Dans les deux chapitres précédents, nous avons présenté tout d'abord la modélisation faite à l'aide des G-cartes. Nous avons montré que la méthode que nous avons utilisée permet de modéliser n'importe quelle interaction. Ensuite, à l'aide du langage à base de règles, nous avons créé toutes les règles nécessaires pour simuler des corps 2D et 3D avec différents modèles mécaniques. L'un des principaux intérêts de ce travail est d'assister la simulation de transformations topologiques dans les divers modèles mécaniques grâce à leur base topologique. En effet, le modèle topologique des G-cartes permet non seulement de décrire les relations d'adjacence et d'incidence entre les cellules, mais il facilite également les changements topologiques en garantissant la cohérence topologique après le changement. D'autre part, la modélisation élémentaire des interactions que nous avons proposée permet de limiter (voire éviter) toute mise à jour des informations mécaniques, dans certains cas de modifications.

Dans ce chapitre nous nous intéressons plus précisément à la déchirure ou à la fracture, avec le langage à base de règles. Comme montré dans l'état de l'art, toute modification topologique du corps mécanique (plus exactement de son maillage)

dépend tout d'abord d'un critère qui la déclenche et d'une opération topologique qui la met en œuvre. Nous commençons ce chapitre, par présenter le principe d'une transformation topologique et comment notre modèle doit s'adapter à la gestion des transformations. Ensuite, nous exposons, les différentes opérations de modifications topologiques implantées dans notre plate-forme, avec les règles créées en utilisant Jerboa. Puis, les différents types de critères utilisés sont exposés. Finalement, nous présentons les résultats de quelques combinaisons de critères et modifications topologiques.

7.2 Principe

Pour ajouter des transformations topologiques à notre plate-forme, il faut tout d'abord adapter la boucle de simulation décrite dans le chapitre précédent.

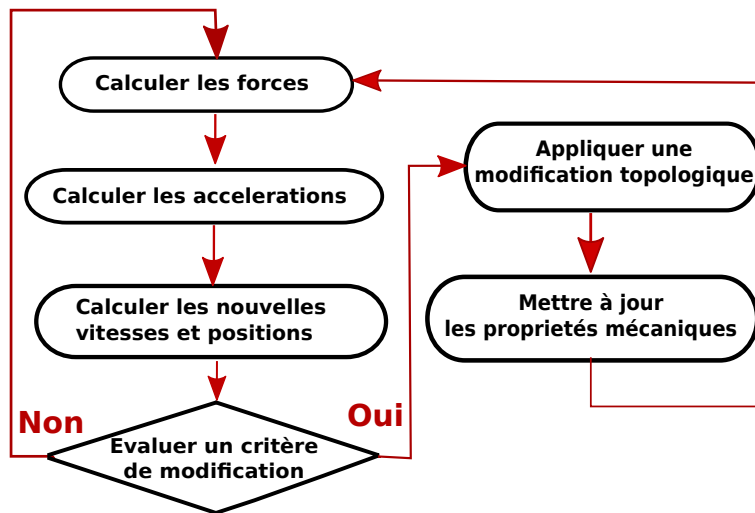


FIGURE 7.1 – Algorithme de simulation d'une fracture/déchirement.

En effet, comme montré sur la figure 7.1, après le calcul des nouvelles positions, il faut parcourir la structure, et évaluer le critère choisi partout où il est applicable. Si ce dernier est vérifié, il faut appliquer une modification topologique et, finalement, mettre à jour les propriétés mécaniques. D'une manière générale, une modification topologique d'un corps mécanique se traite en trois étapes :

- Vérification d'un critère à l'origine de la modification topologique ;
- Application d'une opération topologique qui modifie le maillage ;
- Mise à jour des plongements mécaniques, c'est à dire correction des propriétés physiques du modèle dépendant des cellules modifiées par l'opération topologique.

En utilisant l'approche à base de règles, les trois étapes citées ci-dessus se rassemblent dans une seule et même règle. Le membre gauche de cette règle présente

une pré-condition, pour que la règle ne s'applique que si cette condition (dans notre cas, le critère) est vérifiée, et le membre droit présente la modification topologique à appliquer et indique les plongements à mettre à jour.

Les modifications topologiques que nous considérons dans ce manuscrit (déchirure/fracture) peuvent être simulées facilement en utilisant des G-cartes et plus précisément avec l'opération topologique de *décousure* qui consiste, en 2D, à séparer deux faces adjacentes en éclatant en deux leur arête commune, et en 3D, à séparer deux volumes adjacents en éclatant en deux leur face commune. Cette opération permet de construire des modifications topologiques plus complexes en utilisant une succession de séparations d'éléments. Par exemple, il est possible d'isoler un élément puis éventuellement de le supprimer comme montré dans [MDS10]. Il est également possible de séparer plusieurs éléments, pour provoquer l'éclatement d'un sommet. Nous considérons que la séparation d'éléments est fondamentale car elle est à la base des autres modifications implantées dans notre plate-forme.

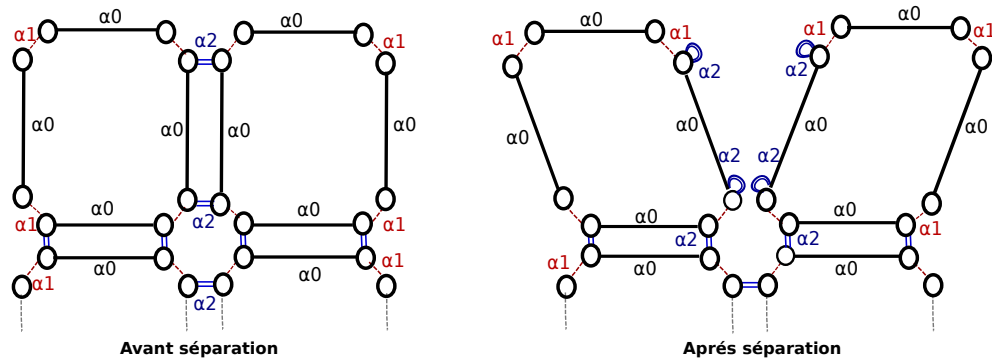


FIGURE 7.2 – Principe de l'opération de décousure en 2D.

L'opération de décousure en 2D consiste, en pratique, à séparer deux faces adjacentes en décousant les liaisons α_2 qui les lient, comme le montre la figure 7.2. D'une manière similaire, deux volumes adjacents peuvent être séparés en décousant les liaisons α_3 qui les lient, comme le montre la figure 7.3.

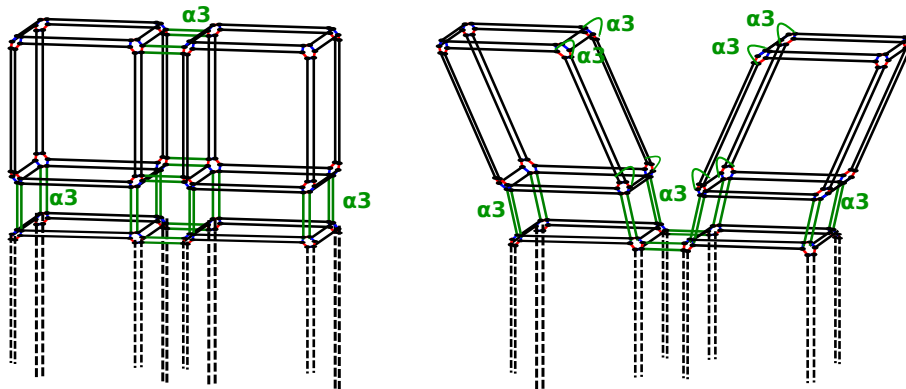


FIGURE 7.3 – Principe de l'opération de décousure en 3D.

Comme nous l'avons déjà dit, ces décousures de liens sont des transformations élémentaires faciles à mettre en œuvre sur le plan topologique dans des G-cartes. En outre, ces décousures présentent parfois des effets de bord, par exemple l'éclatement d'un ou plusieurs sommets, comme nous le constatons sur la figure 7.2. Le sommet central supérieur de la figure se scinde en deux après la décousure. L'opération élémentaire de décousure nécessite donc un traitement (une mise à jour) des plongements de toutes les entités topologiques concernées par la décousure.

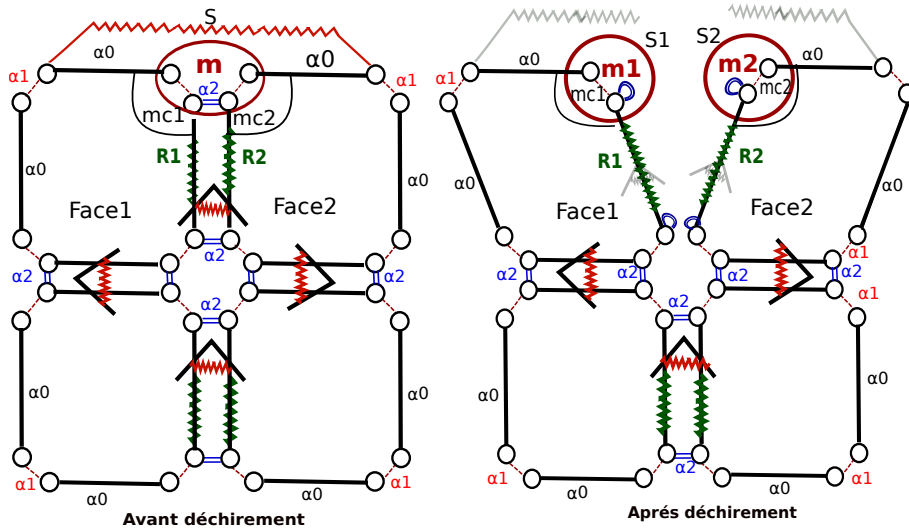


FIGURE 7.4 – Influence de la décousure α_2 sur les plongements mécaniques.

Étudions la prise en charge des plongements dans le cas d'une décousure α_2 entre deux arêtes (figure 7.4). Étudions tout d'abord les masses. La masse du sommet S entouré en rouge, qui est portée par l'orbite sommet, correspond à la contribution de deux masses $mc1$ et $mc2$ fournies par les deux faces adjacentes $Face1$ et $Face2$. Après éclatement du sommet, la masse du sommet $S1$ est apportée uniquement par $Face1$ (et vaut donc $mc1$), et de façon similaire, la masse du sommet $S2$ est uniquement apportée par la $Face2$ (elle vaut $mc2$). De façon générale, nous concluons que la masse des nouveaux sommets issus de la décousure doit être déterminée. De plus, toutes les interactions stockées sur des orbites qui contiennent les liaisons α_2 décousues doivent être traitées. Tout d'abord, les ressorts de courbure angulaire et/ou ceux de Provot placés entre les deux faces et portés respectivement par les orbites $\langle \alpha_2 \rangle$ et $\langle \alpha_0, \alpha_2 \rangle$ sont dupliqués lors de la décousure, mais ignorés par la suite. Plus précisément, le sous-graphe à filtrer lors du calcul de l'interaction ne correspond plus au membre gauche de la règle 6.22 (page 156) pour les ressorts linéaires de Provot et celui de la règle 6.23 (page 157) pour les ressorts angulaires, car ces règles filtrent deux faces cousues ensemble par α_2 . Les règles étant inapplicables, les forces de courbure ne sont plus calculées entre les faces séparées. Il faut néanmoins penser à mettre à zéro les forces correspondantes, car les plongements associés à ces forces continuent d'être pris en compte par la règle du calcul du bilan des forces

en chaque sommet. Par ailleurs, comme les ressorts d'élongation élémentaires sont portés par les orbites $\langle \alpha_0 \rangle$, ils ne sont pas influencés par la découpe des liaisons α_2 . Néanmoins, comme montré sur la figure 7.4, si les ressorts $R1$ et $R2$ ne sont pas modifiés après la modification topologique, ils ne se calculent toutefois plus avec les mêmes sommets. Grâce à notre modélisation, l'identification des nouveaux sommets de chaque ressort d'élongation élémentaire se fait, automatiquement, en suivant le sous-graphe qui est associé à cette interaction. Aucune mise à jour n'est donc nécessaire. Au contraire, les éventuels plongements de cumul, qui regroupent ces mêmes ressorts d'élongation et qui sont portés par l'arête à éclater, sont touchés par la découpe des liaisons α_2 . Il s'agit, dans le cas de la figure, de définir deux plongements de cumul, un pour chaque nouvelle arête, qui ne contiennent respectivement que $R1$ et $R2$. Notons enfin que, concernant les autres interactions 2D, les ressorts de cisaillement ou les diverses matrices utilisées par l'approche par corotationnel en éléments finis ne subissent aucune modification car ils ne dépendent pas des liaisons α_2 .

De façon similaire, en 3D, il n'existe pas de plongement d'interaction élémentaire tributaire des liaisons α_3 , ni pour les ressorts élémentaires ni pour les approches à base d'éléments finis. Par contre, les plongements de cumul des ressorts d'élongation, des ressorts de cisaillement surfacique, des masses/tenseurs (de sommet et d'arête) doivent être mis à jour après la séparation de deux volumes.

En résumé, que ce soit en 2D ou en 3D, l'utilisation de plongements élémentaires nous garantit l'absence de mise à jour des plongements masses ou interactions après une séparation d'éléments. Il faut cependant s'assurer de la mise à jour des plongements de cumul ainsi que la mise à zéro des forces des interactions qui disparaissent suite à l'application de l'opération. Ces considérations vont nous guider lors de la conception des règles à appliquer pour séparer des éléments

Notre approche de la simulation de fractures/déchirures se veut modulaire. Nous souhaitons que l'utilisateur puisse faire le choix d'une modification topologique et d'un critère adapté à cette modification. Aussi, dans la suite de ce chapitre, nous décrivons différentes modifications topologiques en nous basant sur le type de cellule topologique. Ensuite, nous décrivons les différents critères utilisés en se référant à chaque fois à la modification topologique la plus adéquate.

7.3 Modification topologiques

Comme décrit dans l'état de l'art, il existe différents types de modifications topologiques proposés dans la littérature. Nous allons nous focaliser dans notre travail sur celles qui dépendent de l'opération topologique de découpe en 2D /3D. En particulier, nous nous focalisons sur les modifications topologiques par type de cellule, ce qui nous permet de trouver le lien entre ces cellules et les emplacements où les critères de modification peuvent s'appliquer. Nous commençons par la modification fondamentale, à savoir la séparation d'éléments.

7.3.1 Séparation des éléments

La séparation des éléments est une modification topologique qui est le résultat direct de la découiture α_2 qui sépare deux faces et de la découiture α_3 qui sépare deux volumes. Nous présentons ici les règles créées en 2D et 3D pour appliquer cette opération. La règle présentée sur la figure 7.5 correspond à la séparation de faces.

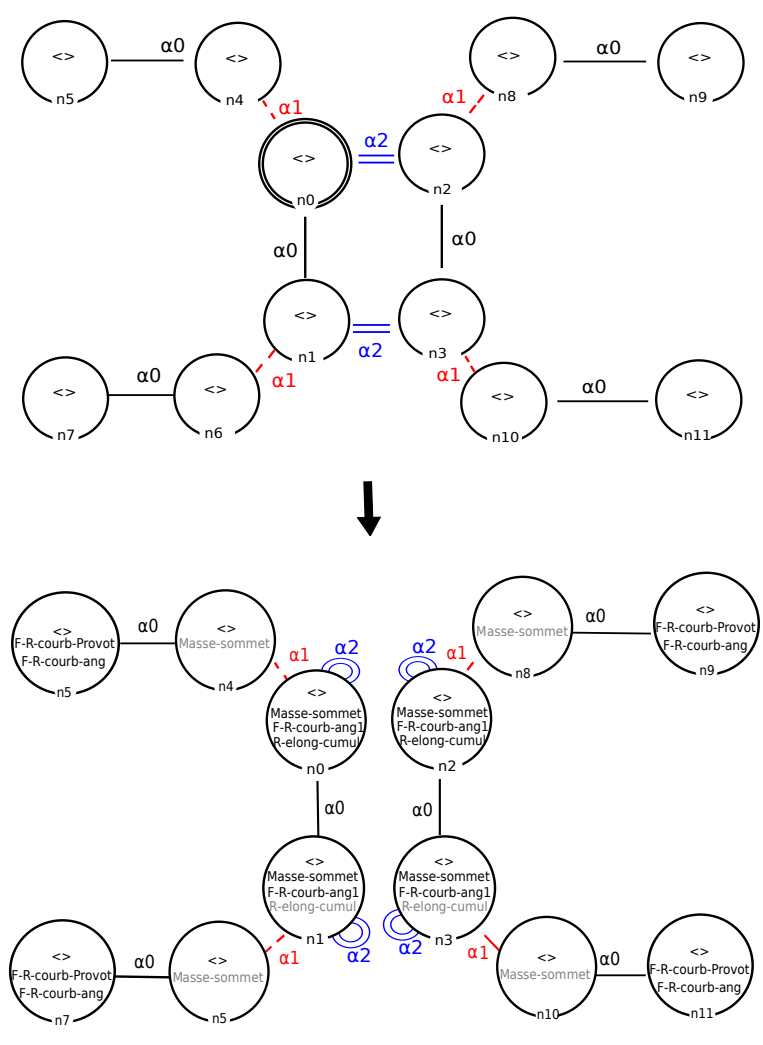


FIGURE 7.5 – Règle de séparation de deux faces adjacentes.

Dans le membre gauche (présenté dans la partie haute de la figure), nous avons

filtré explicitement deux faces adjacentes reliées par les liaisons α_2 . Le membre droit de la règle (présenté dans la partie basse de la figure) présente la modification topologique qui consiste à découper les liaisons α_2 pour séparer les faces. Comme les ressorts linéaires de Provot sont portés par les orbites $\langle \alpha_2 \rangle$ et les ressorts angulaires par les orbites $\langle \alpha_0, \alpha_2 \rangle$, nous les laissons dans les nouvelles arêtes ou extrémités d'arête (car Jerboa duplique automatiquement les plongements) mais, durant le reste de la simulation ils sont ignorés car les règles ne sont plus applicables. Dans la règle montrée sur la figure 7.5, nous mettons à jour simultanément les plongements de cumul (*Masse-sommet* et les diverses interactions de cumul, ici, pour la 2D, le cumul des ressorts d'élongation). Pour calculer les nouvelles valeurs de ces divers plongements, nous écrivons un algorithme qui parcourt les sous-orbites de l'orbite porteuse du cumul et récupérons les plongements d'interactions élémentaires correspondants qui y stockés.

Concernant le calcul de la masse de chaque sommet provenant d'un éclatement, il faut faire un inventaire des toutes les orbites sommets de faces $\langle \alpha_1 \rangle$ (ce sont effectivement des sous-orbites de l'orbite sommet) et faire la somme des valeurs des plongements *Masse-coin*. Dans notre règle, nous mettons alors à jour les plongements *Masse-sommet* des nœuds n_0 , n_1 , n_2 et n_3 . Notons que n_0 et n_2 , d'une part, et n_1 et n_3 , d'autre part, peuvent toujours faire partie de la même orbite sommet (ils restent alors liés par des liaisons non représentées sur la règle).

De même, l'arête qui se scinde en deux arêtes peut porter un ressort d'élongation de cumul. Chacune des deux nouvelles arêtes possède un plongement de cumul. Comme expliqué ci-dessus et illustré sur la figure 7.4, le plongement de cumul des nouvelles arêtes est initialisé avec les propriétés du seul ressort d'élongation élémentaire dont le plongement est stocké dans les orbites α_0 . Cette opération est effectuée pour les nœuds n_0 et n_2 de la règle 7.5, à l'aide du plongement *R-elong-cumul*.

Dans cette même règle, nous mettons à zéro les forces appliquées par les ressorts de courbure : plongements *F-R-courb-Provot* et *F-R-courb-ang* des nœuds n_5 , n_7 , n_9 , n_{11} et plongements *F-R-courb-ang1* des nœuds n_0 à n_3 . Dans le cas où l'un et/ou l'autre sommets sont scindés, les plongements (position, vitesse et accélération...) sont dupliqués automatiquement, et les nouveaux sommets sont pris en compte dans les prochains pas de simulation. Il faut noter que cette règle est générale est ne dépend pas du type de face.

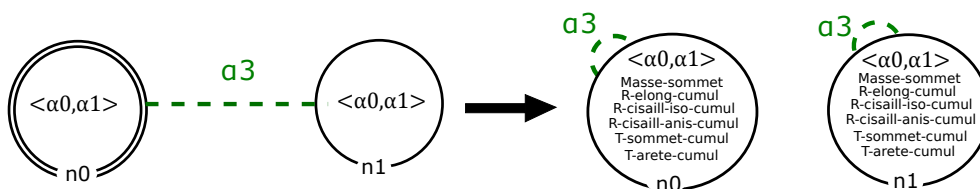


FIGURE 7.6 – Règle de séparation de deux volumes adjacents.

De la même manière, nous avons créé la règle de séparation de volumes, présentée sur la figure 7.6. Dans cette règle nous avons filtré dans le membre gauche les deux

faces liées par une liaison α_3 par lesquelles les volumes sont cousus. Ces faces sont définies par des orbites et peuvent donc être de type quelconque. Dans le membre droit de la règle, les liaisons α_3 sont décousues, ce qui sépare les deux volumes. Dans cette même règle nous avons mis à jour les valeurs des masses des sommets dans le cas où un ou plusieurs d'entre eux se scindent ainsi que les interactions de cumul (en l'occurrence, élongation, cisaillement surfacique et tenseurs sommets et arêtes pour le modèle masses/tenseurs). En effet, pour mettre à jour le plongement **Masse-sommet** des sommets susceptibles d'avoir été éclatés, il faut faire l'inventaire des sous-orbites sommets de volume constituant l'orbite de ces sommets et sommer les masses que l'on y collecte. De la même manière, nous calculons les nouvelles valeurs des ressorts élongation cumulés en faisant l'inventaire des sous-orbites $\langle \alpha_0 \rangle$ qui composent les orbites $\langle \alpha_0, \alpha_2, \alpha_3 \rangle$ des nouvelles arêtes et en faisant la somme des raideurs et des éventuels amortissements des ressorts élémentaires qui y sont plongés. Concernant les ressorts de cisaillement surfacique (isotrope/anisotrope) cumulés, comme la face entre deux volumes se scinde en deux, les valeurs des caractéristiques des ressorts de cisaillements de cumul associés aux nouvelles faces sont la copie exacte de celles des ressorts élémentaires associés à cette face. Enfin, les masses/tenseurs de sommet et d'arêtes sont également à recalculer. Les masses tenseurs des sommets éclatés sont recalculés en tant que somme des masses/tenseurs élémentaires des sommets de volume, stockés dans les sous-orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$ des nouveaux sommets $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$. Idem, les masses/tenseurs d'arêtes cumulés sont recalculés en effectuant la somme des masses/tenseurs des arêtes élémentaires, stockés dans les sous-orbites extrémités des arêtes de face $\langle \alpha_2 \rangle$ des nouvelles extrémités d'arête $\langle \alpha_2, \alpha_3 \rangle$.

Pour conclure, nous soulignons le fait que pour gérer une décousure entre éléments de dimension n , nous avons créé des règles en nous focalisant sur les cellules de dimension $n - 1$ qui les lient. Ainsi, nous considérons, par comparaison avec les autres modifications détaillées ci-dessous, que ces règles sont dédiées aux éléments de dimension $n - 1$.

7.3.2 Suppression d'éléments

La suppression d'éléments concerne les cellules de dimension n . Pour supprimer un élément qui peut être un volume en 3D ou une face en 2D, nous avons utilisé le même processus que proposé par [MDS10]. Ce processus consiste d'abord à isoler l'élément, c'est à dire en 2D, à découdre toutes les liaisons α_2 de toutes ses arêtes qui le lient à d'autres faces de l'objet. En 3D, cela revient à découdre toutes les liaisons α_3 qui lient le volume en question à ses voisins. Cette phase d'isolement peut être faite en itérant les règles de séparation des éléments (7.5) et (7.6) décrites ci-dessus sur les $(n - 1)$ -cellules du bord de l'élément. La deuxième étape consiste à supprimer l'élément isolé. Pour cela, la règle présentée sur la figure 7.7 est créée.

Dans le membre gauche de cette règle, nous filtrons un volume isolé (α_3 libre), la partie vide de droite représente la suppression complète du volume et de tous les plongements qui ont été supportées par son orbite ou n'importe quelle sous-orbite.

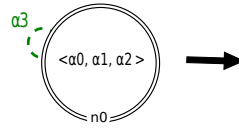


FIGURE 7.7 – Règle de suppression d'un volume isolé.

De même, en 2D nous avons créé une règle similaire où une face libre en α_2 est filtrée.

7.3.3 Éclatement d'un sommet

L'éclatement d'un sommet est une modification topologique qui traite les cellules de dimension 0 (les sommets). C'est une opération complexe qui s'effectue suite à la séparation de plusieurs éléments incidents à ce sommet. Comme proposé dans [OH99], pour éclater un sommet en 3D, une couronne de faces entourant ce sommet peut être décousue par α_3 . En 2D, seulement deux arêtes doivent être décousues comme montré sur la figure 2.32 (page 44).

Nous avons choisi de gérer cette opération globalement (i.e. tous les sommets « en même temps »), en deux étapes successives. La première étape suppose que le critère de fracture/déchirure fournit un plan de séparation (comme nous le verrons dans la section 7.4, page 189), qui est stocké dans les orbites sommets où le critère est vérifié. Il faut alors calculer la position de chaque élément incident à chacun de ces sommets par rapport à son plan, en injectant la position du centre de l'élément dans l'équation du plan et en récupérant le signe du résultat. Pour cette première étape, nous avons créé deux règles différentes, respectivement pour la 2D et la 3D, car nous nous basons sur des nouveaux plongements qui dépendent de la dimension. La figure 7.8 montre cette règle dans le cas 3D. La règle en 2D repose sur le même principe.

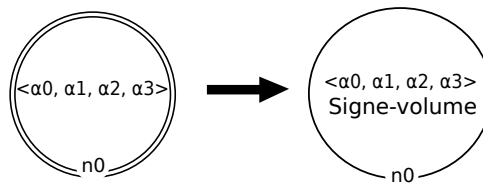


FIGURE 7.8 – Règle du calcul de signes des volumes entourant un sommet (cas 3D).

Cette règle s'applique sur toute la composante connexe. La position de chaque élément par rapport au plan de séparation fourni par chacun de ses sommets doit être calculée. Aussi, pour un élément, il y a autant de signes à calculer que de sommets de l'élément. C'est pourquoi les signes sont stockés dans les plongements **Signe-volume** portés par les orbites sommets de volume $\langle \alpha_1, \alpha_2 \rangle$, en 3D, et **Signe-face** portés par les orbites sommets de face $\langle \alpha_1 \rangle$ en 2D (les orbites choisies correspondent aux sommets des éléments). Le type de ces plongements est un

booléen. Dans le membre droit de la règle, le moteur de Jerboa parcourt toutes les orbites sommets d'éléments de la composante connexe et récupère l'équation du plan du sommet auxquels ils appartiennent. Si cette dernière est définie, le plongement associé au coin est calculé en récupérant les coordonnées du centre de l'élément dont il fait partie, et en les plaçant dans l'équation du plan. Nous associons alors la valeur *true* au plongement si le résultat est positif et *false* dans le cas contraire (ou si l'équation n'est pas définie). Dans une seconde étape, nous utilisons un script qui parcourt tous les sommets de la composante connexe. En 2D, nous examinons, en chaque sommet, si les brins des sous-orbites extrémités d'arêtes (orbites $\langle \alpha_2 \rangle$) possèdent des signes de plongement `Signe-face` différents. Si cela est le cas, nous appelons la règle 7.5 sur l'arête en question. De façon analogue, en 3D, nous inspectons, en chaque sommet, les orbites sommets de faces $\langle \alpha_1, \alpha_3 \rangle$ et examinons les signes des plongements `Signe-volume` portés par les brins des deux sous-orbites $\langle \alpha_1 \rangle$. Si les deux signes sont différents, nous décidons d'appliquer la règle 7.6 pour découdre par α_3 la face dont on considère le coin.

7.3.4 Suppression d'arête

Le suppression d'une arête est une modification topologique qui traite les cellules de dimension 1 (arête). Si un critère est appliqué sur une arête, il est souvent attendu que l'arête disparaisse [NTB⁺91]. Cependant, supprimer une arête d'un maillage est une modification topologique délicate. Elle peut, par exemple, être implantée en tant que fusion de ses sommets et entraîner un changement de type des éléments qui l'entourent. Par exemple, la contraction d'une arête d'un maillage à base de quadrilatères peut transformer les éléments adjacents en triangles. Par conséquent, comme solution, il est préférable d'utiliser des transformations plus générales qui prennent en considération l'arête concernée mais qui contrôlent également les éléments adjacents. Par exemple, la suppression des éléments qui entourent cette arête comme proposé par Norton *et al.* [NTB⁺91]. Cette modification est alors basée sur la suppression d'éléments décrite dans la section 7.3.2 (page 186) et qui est elle-même basée sur la séparation des éléments. Nous pouvons également utiliser l'éclatement d'un des sommets de l'arête comme proposé dans [DKS⁺11]. Il faut, dans tous les cas, noter que, quelle que soit la transformation choisie, cette dernière influence non seulement la cellule du critère (l'arête) mais aussi de nombreux éléments adjacents.

Pour résumer, toutes ces modifications topologiques se basent sur une seule opération, la séparation d'éléments adjacents. Avec cette opération, il n'y a aucun plongement fondamental à mettre à jour, uniquement les plongements d'optimisation si l'utilisateur a choisi de les utiliser pour accélérer la simulation. Nous étudions maintenant les critères permettant de déclencher l'application des modifications topologiques.

7.4 Critères

Les transformations topologiques, en particulier, la déchirure et la fracture nécessitent des critères pour les déclencher. Ces critères, comme déjà mentionné dans l'état de l'art, peuvent être basés soit sur la déformation, soit sur les forces internes appliquées à l'objet. Pour les différents types de ressorts (linéaires ou angulaires), les critères qui dépendent des déformations ou des forces sont équivalents car la plupart du temps, la relation entre la déformation et la force est monotone. Dans notre travail, nous avons opté pour les critères basés sur les forces internes appliquées par les interactions. En effet, ils sont plus adaptés à notre modélisation puisque nous stockons les forces de toutes les interactions dans les orbites/sous-orbites de l'orbite sommet. Nous avons choisi de les diviser en deux catégories selon que le critère se base sur la force d'une seule interaction ou sur le cumul de plusieurs interactions. Nous les avons appelés, respectivement, « critères atomiques » et « critères cumulatifs » .

7.4.1 Critères atomiques

Les critères atomiques, comme leur nom l'indique, sont des critères qui dépendent d'une seule interaction. Nous commençons par les ressorts. Le critère consiste à comparer les forces appliquées par un type de ressort choisi à un seuil donné. Pour cela, nous avons créé un script qui parcourt tous les sommets de l'objet et récupère les forces des ressorts appliquées sur chacun d'eux. Si l'intensité de la force de magnitude la plus élevée est plus grande qu'un seuil prédéfini, alors il faut enlever le ressort associé à cette force en appliquant la règle qui convient (suppression ou séparation d'élément). Pour enlever un ressort, la transformation topologique à faire dépend du type de ressort à enlever. En effet, les ressorts d'élongation sont portés par les arêtes et comme discuté dans la section précédente, des opérations élaborées permettent de supprimer l'arête. Si le même critère est vérifié sur un ressort de cisaillement, l'élément dans lequel est stocké ce ressort (volume en 3D ou face en 2D) peut être supprimé, comme décrit dans 7.3.2 (page 186). Finalement, si un ressort de courbure (linéaire ou angulaire) vérifie le critère de déchirure/fracture (la force ou le moment dépasse un seuil), les deux éléments liés par ce ressort doivent être séparés, comme décrit dans la section 7.3.1 (page 184).

Si on utilise un modèle basé sur la mécanique continue, le tenseur des contraintes de chaque élément (e) est calculé comme expliqué dans [MDM⁺02] :

$$\boldsymbol{\sigma}^{(e)} = [C] [B^{(e)}] \mathbf{U}^{(e)} \quad (7.1)$$

Avec $[C]$ la matrice de comportement, $[B^{(e)}]$ la matrice de déformation d'un élément (e) et $\mathbf{U}^{(e)}$ le vecteur déplacement de ses sommets. Pour cela, nous avons créé la règle présentée sur la figure 7.9 qui s'applique sur toute la composante connexe. Dans le membre droit de la règle, nous mettons à jour, sous forme vectorielle, le tenseur de contrainte $\boldsymbol{\sigma}^{(e)}$ de chaque élément dans son orbite volume (plongement

tenseur-contrainte-volume plongé sur les orbites volumes).

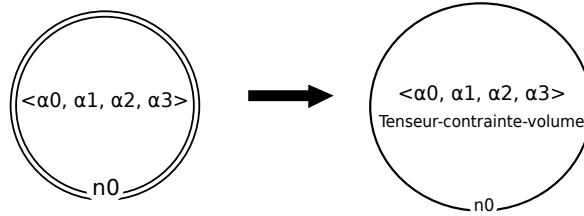


FIGURE 7.9 – Règle du calcul des tenseurs de contrainte en chaque volume.

Les valeurs propres des tenseurs de contraintes de chaque élément, une fois convertis en matrice, peuvent alors être calculées. Si au moins une de ces valeurs propres dépasse le seuil, nous décidons de supprimer l'élément concerné. Pour cela, nous avons écrit un script qui parcourt chaque élément et, à chacun d'eux, nous appliquons la règle 7.7 en ajoutant le critère sur les valeurs propres (et donc leur calcul) dans les pré-conditions.

Pour éviter la création de règles spécifiques aux interactions, il est possible d'utiliser une approche plus générale, qui peut être appliquée à la fois en 2D et en 3D. En effet, nous avons défini un critère qui consiste à vérifier si une force exercée sur un sommet, quelle que soit son origine, dépasse un seuil. À cet effet, nous récupérons toutes les forces élémentaires qui sont appliquées sur chaque sommet et nous comparons l'intensité de la force de plus grande magnitude au seuil. Si cette intensité est supérieure au seuil, nous calculons un plan perpendiculaire au vecteur support de la force. Nous faisons tout ce calcul dans le membre droit de la règle présentée sur la figure 7.10 et le stockage de ces équations se fait dans le plongement Equation-plan qui est porté par les orbites sommets $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$. Nous appliquons ensuite le script d'éclatement de sommet décrit dans la section 7.3.3 (page 187), qui se base sur l'équation du plan pour décider quels éléments incidents au sommet sont à séparer.

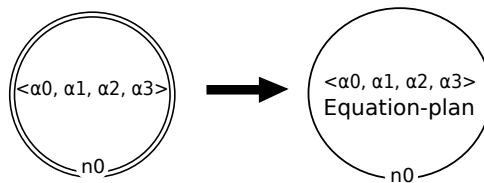


FIGURE 7.10 – Règle de calcul de l'équation du plan de fracture/déchirure.

Il faut noter qu'en raison du principe d'action/réaction, les deux extrémités d'un ressort sont susceptibles de dépasser le seuil au même pas de temps. Ceci est également vrai pour d'autres types d'interaction impliquant plusieurs sommets. Dans ce cas, notre méthode fait éclater tous les sommets concernés.

7.4.2 Critères cumulatifs

Les critères cumulatifs se basent sur la somme des forces appliquées sur chaque sommet de l'objet. Dans le cas de la mécanique continue, en nous basant sur l'approche de Koshier *et al.* [KLB14] en chaque sommet, nous calculons la moyenne des tenseurs des tétraèdres incidents en pondérant chacun par la masse de l'élément (e)

$$\boldsymbol{\sigma}^{(sommets)} = \sum_{(e)} m^{(e)} \boldsymbol{\sigma}^{(e)} / \sum m^{(e)} \quad (7.2)$$

Avec $m^{(e)}$ la masse des éléments (e) et $\boldsymbol{\sigma}^{(e)}$ leur tenseur de contrainte. Nous calculons ensuite les valeurs propres du tenseur de chaque sommet. Si au moins une de ces valeurs est supérieure au seuil donné, il faut calculer le vecteur propre correspondant (s'il y a plusieurs valeurs propres, nous choisissons la plus élevée). Finalement, nous calculons l'équation du plan de séparation qui est défini comme le plan perpendiculaire au vecteur propre. Tous ces calculs (évaluation du tenseur de sommet, calculs des valeurs propres et choix du vecteur propre de valeur propre la plus élevée) sont faits dans une seule règle qui permet de déterminer l'équation du plan dans le plongement `Equation-plan` porté par les orbites sommets. Cette règle est la même que celle présentée sur la figure 7.10, seule la manière de calculer le plan est différente. Nous appliquons finalement le script d'éclatement de sommet comme décrit en section 7.3.3.

Cette approche peut être adaptée à la 2D. Nous avons procédé alors de la même manière, par un script comprenant plusieurs étapes. La première étape consiste en une règle sur la composante connexe pour mettre à jour le tenseur de contrainte `tenseur-contrainte-face` de chaque face, en le calculant dans le plan (x, y) , selon une approche similaire à la règle 7.9. La seconde étape appelle une règle sur la composante connexe dont le but est de calculer le plan de séparation en chaque sommet. Ce calcul nécessite d'évaluer, dans le plan (x, y) , la moyenne des tenseurs des faces incidentes à chaque sommet, puis de trouver les valeurs propres de ce tenseur et le vecteur propre associés à la plus grande valeur propre, si cette dernière dépasse un seuil. Enfin, le script se termine par les étapes décrites dans le paragraphe 7.3.3, à savoir une règle sur la composante connexe pour déterminer le signe de chaque face par rapport au plan de séparation stocké dans chacun de ses sommets puis un parcours des sommets pour trouver les arêtes incidentes qui lient deux faces de signes opposés et appeler la règle 7.5, page 184, pour séparer ces faces.

Un critère cumulatif plus général peut également être proposé, quelles que soient les interactions utilisées. Ce critère consiste à faire la somme en chaque sommet de toutes les forces internes stockées dans chaque sous-orbite de l'orbite sommet. Pour cela, nous avons créé une règle qui s'applique sur toute la composante connexe qui a la même forme que la règle présentée sur la figure 6.2 (page 133) pour permettre de calculer la somme des forces internes et la stocker dans un plongement porté par les orbites sommets $\langle \alpha_1, \alpha_2, \alpha_3 \rangle$. Dans une seconde étape, nous avons créé une règle semblable à celle présentée sur la figure 7.10. Nous récupérons dans son membre droit, le cumul de forces appliquées sur chaque sommet, et si la norme de

ce cumul dépasse un seuil, nous appliquons un éclatement de sommet en utilisant un plan de fracture perpendiculaire à cette force. Néanmoins ce type de critère ne peut fonctionner que dans des cas particuliers, par exemple lorsque des forces internes doivent compenser une contrainte externe élevée, ou lorsqu'une déformation apparaît brusquement localement dans le maillage (après le déplacement d'un ou de plusieurs sommets). En effet, au repos, le bilan des forces internes est nul et par conséquent aucune déchirure/fracture ne peut survenir, même si certaines interactions atomiques produisent des forces très intenses.

7.4.3 Critères dédiés à la séparation d'éléments

Dans les critères cités précédemment, les modifications topologiques choisies s'appliquent, au départ, sur des sommets, des arêtes ou des éléments où un critère est vérifié mais se traduisent généralement par plusieurs opérations de séparation d'éléments. Ces opérations ne s'appliquent pas directement sur le seul lieu du critère, mais impliquent plusieurs éléments non pris en compte par le critère. Ce phénomène apparaît, par exemple, lorsque le critère s'applique sur une arête alors que la modification s'effectue sur l'un de ses sommets. De la même manière que Smith *et al.* [SWB00], qui ont proposé un critère basé sur des contraintes pour séparer deux volumes, nous proposons une solution basée sur la force appliquée sur une arête/face pour déclencher la séparation des éléments. En d'autres termes, nous voulons trouver des critères qui sont appliqués sur une cellule et qui déclenche une modification sur cette même cellule. En 2D, les faces sont séparées en divisant une arête, et donc cette dernière doit porter le critère de fracture. Plus précisément, comme le montre la figure 7.11, les forces appliquées sur l'arête AB par toutes les interactions supportées par les faces adjacentes $F1$ et $F2$ sont comparées à un seuil pour vérifier si l'arête s'éclate ou non. Pour cela, tout d'abord, il faut définir un plan de séparation, contenant l'arête, et dont la normale est à déterminer. Nous choisissons le plan divisant en deux parties égales l'angle dièdre entre les deux faces, défini par sa normale :

$$\mathbf{n} = \mathbf{AB} \times (\mathbf{n}_1 + \mathbf{n}_2) \quad (7.3)$$

Dans une seconde étape, nous devons calculer les forces appliquées par chaque face sur les deux sommets A et B . Il faut alors parcourir les sous-orbites des sommets A et B de chaque face (orbites $\langle \alpha_1 \rangle$) et collecter les forces qui y sont stockées. Ces forces sont notées $\mathbf{f}_A^{(i)}$ et $\mathbf{f}_B^{(i)}$, où i désigne les faces, et on note \mathbf{f}_i la somme des forces $\mathbf{f}_A^{(i)} + \mathbf{f}_B^{(i)}$. Le critère se base sur la quantité suivante :

$$|(\mathbf{f}_1 - \mathbf{f}_2) \cdot \mathbf{n}| \quad (7.4)$$

Le critère compare l'effet des forces appliquées par chaque face dans la direction orthogonale au plan de séparation. Si la valeur est plus grande qu'un seuil, cela signifie que les deux forces ne se compensent pas, et nous décidons alors de séparer les deux éléments adjacents. Il faut noter que cela se produit lorsque l'une des

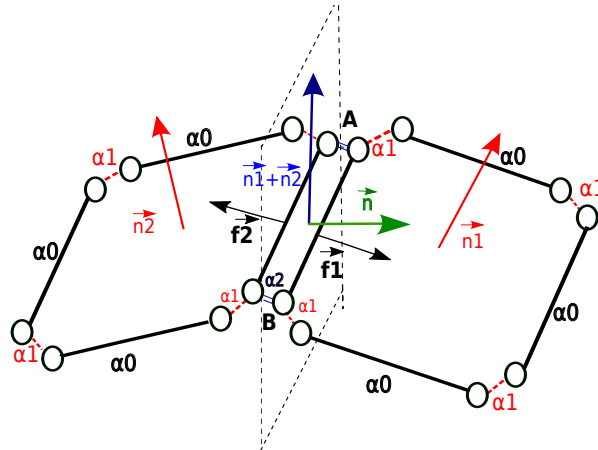


FIGURE 7.11 – Calcul de forces internes appliquées sur une arête.

deux forces \mathbf{f}_1 ou \mathbf{f}_2 a une grande intensité par rapport à l'autre ou que les deux forces s'exercent dans des sens opposés. Comme ce critère dépend des valeurs de forces élémentaires, nous avons choisi de l'appeler critère atomique d'arête. Pour faire ce calcul nous avons créé une règle de la même forme que celle présentée sur la figure 6.2 (page 133), et qui stocke la valeur calculée avec la formule (7.4) dans un plongement que nous avons appelée `critere-arete-elem`, qui est porté par les sous-orbités arêtes $\langle \alpha_0, \alpha_2 \rangle$. Nous enchaînons ensuite avec un script qui parcourt les arêtes et applique la règle 7.5 pour séparer les faces incidentes à condition que le plongement `critere-arete-elem` dépasse un seuil.

Nous avons, également, proposé un autre type de critère qui s'applique sur les arêtes en 2D et qui est un critère cumulatif. Nous calculons la somme des forces appliquées par les faces incidentes aux sommets A et B. Nous considérons alors le bilan des forces appliquées à l'arête $\mathbf{f}_A + \mathbf{f}_B$ (comme pour les sommets) et nous évaluons l'effet global de ce bilan de forces sur l'arête. Le critère consiste alors à comparer à un seuil ce bilan projeté sur le vecteur normal du plan séparateur par la formule :

$$|(\mathbf{f}_A + \mathbf{f}_B) \cdot \mathbf{n}| \quad (7.5)$$

La mise en œuvre consiste en une règle (sur la composante connexe) dont le but est de calculer le bilan des forces en chaque arête (plongement `force-cumul` sur les orbités $\langle \alpha_0, \alpha_2 \rangle$), puis de parcourir les arêtes en leur appliquant la règle 7.5 munie d'une pré-condition sur le bilan des forces.

Le critère proposé est applicable également en 3D pour une face et nous utilisons alors la séparation de deux volumes adjacents. Dans ce cas, la normale de la face (ou une normale moyenne si la face n'est pas plane) est utilisée pour projeter les forces appliquées sur les sommets de la face, par les deux volumes. La mise en œuvre est très similaire au cas 2D.

7.5 Exemples de résultats

Dans cette section, nous présentons quelques résultats de combinaisons entre les modifications topologiques et les critères topologiques que nous avons proposés, après leur implémentation avec le langage à base de règles. Quelques exemples vidéos de déchirures et fractures sont accessibles par l'URL suivante : <http://xlim-sic.labo.univ-poitiers.fr/jerboa/doc/japhy/>.

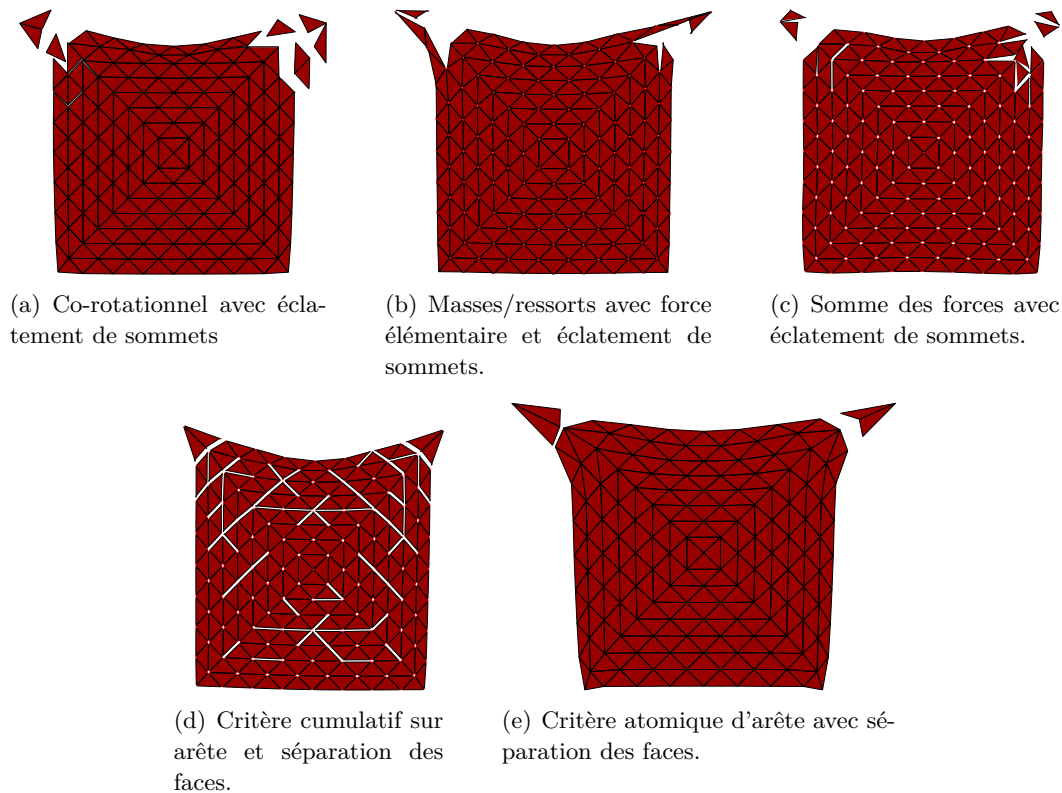


FIGURE 7.12 – Exemples de déchirures d'un tissu à maille triangulaire avec différents critères.

Tout d'abord, nous utilisons le même objet et le simulons avec différents critères pour observer la différence de comportement. Ainsi, l'objet que nous utilisons en 2D est un maillage triangulaire dont les deux extrémités supérieures sont toujours fixées.

Sur la figure 7.12(a), le tissu est modélisé avec le modèle de co-rotationnel. Nous avons choisi d'éclater les sommets si les forces appliquées par au moins l'une de ses faces incidentes dépassent le seuil donné. De même, la figure 7.12(c) présente le même tissu modélisé avec un modèle masses/ressorts. Nous avons décidé d'éclater ses extrémités (sommets) si la force générée par un des ressorts dépasse un seuil. Ensuite, la figure 7.12(d) présente ce même tissu où une arête est divisée lorsque la somme des forces appliquées sur cette dernière dépasse un seuil. Le dernier exemple montré sur la figure 7.12(e), présente le même tissu simulé avec

le modèle masses/ressorts et si le critère de forces atomique sur l'arête est vérifié, nous décidons de diviser cette dernière (en séparant les éléments).

Comme on peut le voir, les différents critères produisent des résultats complètement différents, sur le même objet, où la déchirure est concentrée sur le lieu de critère (l'arête par exemple) ou sur plusieurs éléments de l'objet (éclatement de sommets). Il faut noter que ces mêmes critères avec les mêmes règles peuvent être appliqués sur d'autres types de maillages comme le maillage mixte et le maillage rectangulaire (voir figure 7.13).

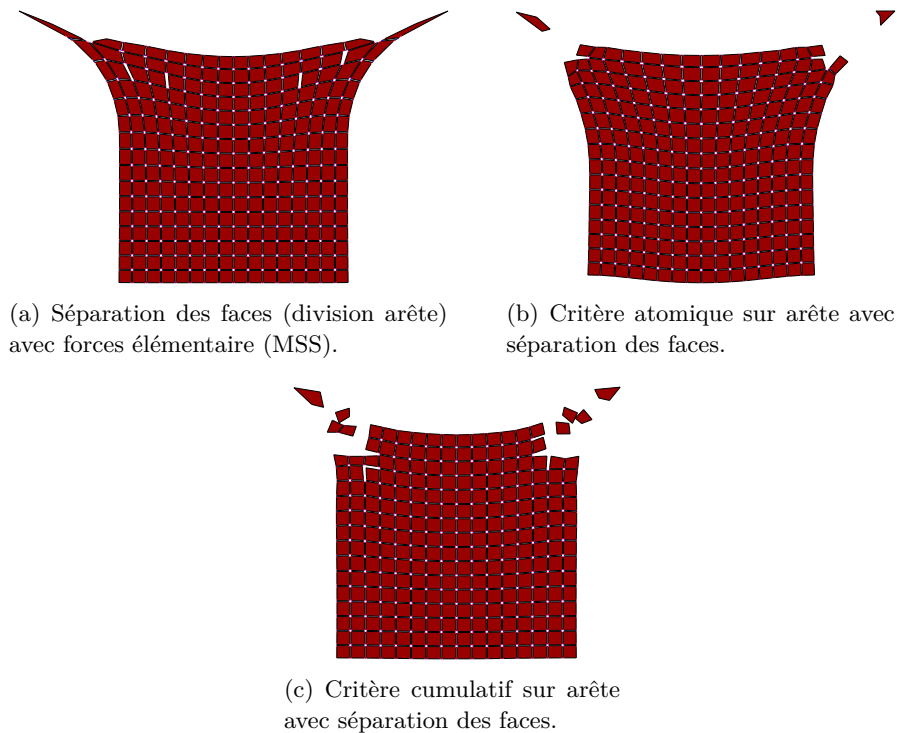


FIGURE 7.13 – Exemples de déchirure d'un tissu maillé avec des quadrilatères.

En 3D, nous avons, de la même manière, simulé le même objet, ici un tube rectangulaire composé de 32 hexaèdres.

Dans la figure 7.14(a), le critère utilisé est celui de forces atomiques appliquées sur un sommet et la modification est l'éclatement du sommet. Le même objet présenté sur la figure 7.14(b) est simulé avec un critère cumulatif de toutes les forces exercées sur un sommet. Sur la figure 7.14(c), le critère utilisé est basé sur les forces élémentaires appliquées par les ressorts de cisaillement volumique : si l'un d'eux fournit une force qui excède le seuil, le volume qui le contient est supprimé. Dans la figure 7.14(d), si les forces élémentaires appliquées par un ressort d'élongation dépassent le seuil, alors nous supprimons tous les volumes incidents à cette arête. Sur la dernière figure 7.14(e), le critère utilisé est basé sur les forces atomiques exercées sur les faces et permet de séparer les volumes incidents.

Il faut noter que ces mêmes critères peuvent être appliqués sur d'autres types de

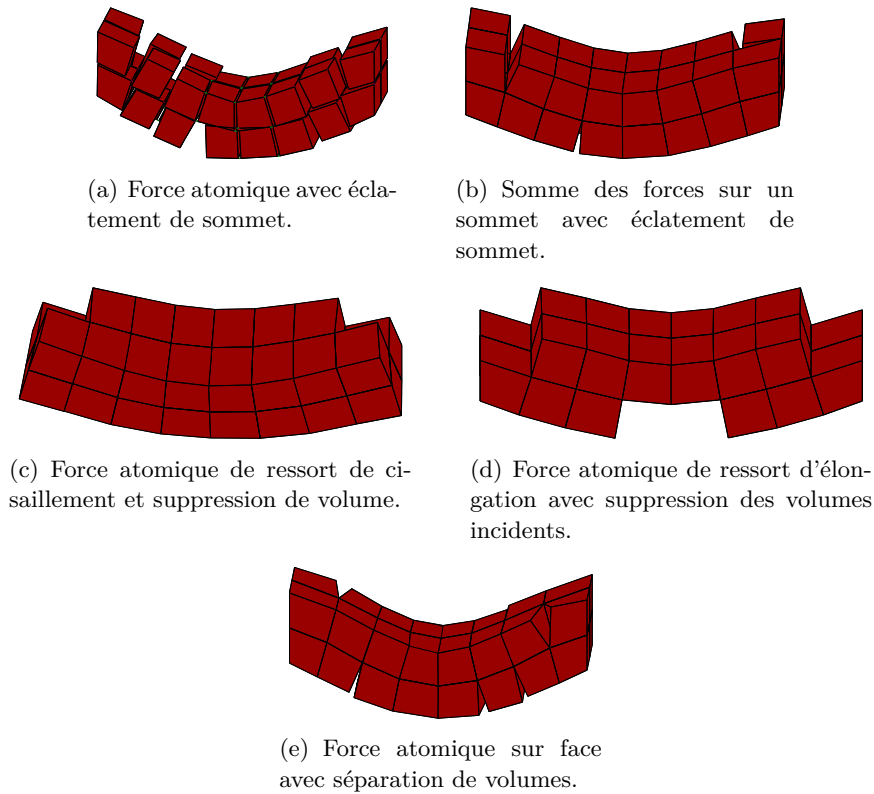


FIGURE 7.14 – Exemples de fracture d'un objet 3D simulé en masses/ressorts avec différents critères.

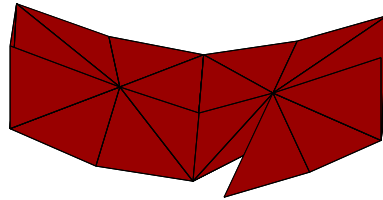


FIGURE 7.15 – Exemple de fracture d'un objet à mailles tétraédriques avec le critère de cumul de force pour séparer les volumes.

maillages. Par exemple, sur la figure 7.15, nous avons expérimenté, sur un maillage tétraédrique, le critère des forces appliquées sur les faces avec séparation des volumes. Un autre exemple, celui d'un foie constitué de 597 tétraèdres et simulé grâce au modèle du co-rotationnel est présenté sur la figure 7.16. Dans cet exemple le critère utilisé est celui des forces élémentaires avec éclatement de sommet comme modification topologique.

Comme déjà indiqué précédemment, Jerboa, même s'il permet la simulation de transformations topologiques sur des objets non triviaux (voir figure 7.16), est un outil dédié au prototypage et non à la simulation interactive. Pour montrer

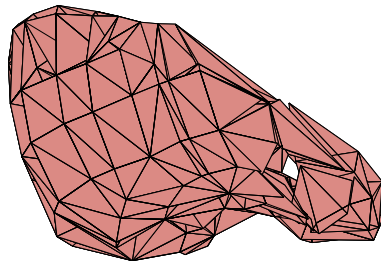


FIGURE 7.16 – Simulation de modèle tétraédrique par l’approche du co-rotationnel et déchirure basée sur les forces élémentaires et l’éclatement de sommets.

l’efficacité de notre modèle, quelques modèles ont été portés en C++ en se basant sur le moteur *Moka* [Mok] pour manipuler les G-cartes.



FIGURE 7.17 – Simulation d’un tissu avec le critère global sur l’arête.

Tout d’abord des modèles 2D spécifiques aux maillages triangulaires puis à base de quadrilatères ont été implantés. Néanmoins, en s’inspirant de la généralité de notre modèle fournie par le langage à base de règles Jerboa, il a été possible d’implanter, de façon efficace, un modèle générique permettant de mixer dans un même maillage tout type de mailles. La simulation d’un tel maillage implique un surcoût moyen de 1 % en temps calcul par rapport à la simulation de maillages purement triangulaires ou à base de seuls quadrilatères, ce qui peut-être considéré comme négligeable. L’implantation C++ actuelle, si elle n’utilise pas le moteur Jerboa, n’en respecte pas moins tous les principes décrits dans ce manuscrit (modélisation, plongements...). En particulier, les calculs d’interaction s’effectuent en parcourant les brins de la G-carte et en cherchant les plongements d’interaction. Les différents brins du motif associé à l’interaction sont alors identifiés par application des involutions α spécifiées dans le sous-graphe. Nous déterminons alors les sommets utiles au calcul de l’interaction. Ensuite, les forces sont calculées, cumulées en chaque sommet et stockées, individuellement, si la modification topologique le demande, en tant que plongement dans les sous-orbités des sommets, comme nous le faisons dans les règles.

Nous avons simulé un modèle de tissu 2D composé de 10×10 mailles rectangulaires. Des exemples de résultats de simulation de ce tissu, en utilisant le critère global sur l’arête pour séparer les faces (section 7.4.3) sont présentés sur la figure

	Simulation sans modification	Critère global d'arête	Critère atomique d'arête
Quadrilatères	0.37	0.41	0.55
Triangles	0.78	0.95	1.3
Mailles mixtes	0.61	0.72	0.97

TABLE 7.1 – Temps de simulation (ms) d'un tissu selon le type de maillage (implantation C++).

7.17. Pour cet exemple, la masse du tissu est de 100 g, la raideur des ressorts d'élongation a comme valeur $60 N/m$, celle des ressorts de cisaillement est égale à $43 N/m$, celle des ressorts de courbure est égale à $10 N/m$, l'amortissement ambiant vaut à $0.1 Ns/m$ et le seuil de déchirement vaut $0.5 N$.

Certains temps de simulation sont résumés dans la table 7.1 pour différents types de maillages, en utilisant la méthode d'intégration de Runge-Kutta 4 pour permettre une simulation plus stable.

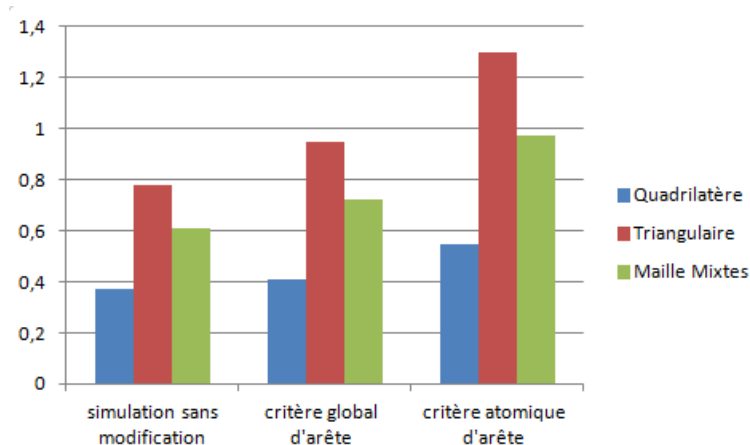


FIGURE 7.18 – Histogramme de simulation d'un tissu 2D avec des critères sur l'arête.

Comme on peut le constater sur l'histogramme de la figure 7.18, la simulation avec des mailles rectangulaires est plus rapide que les deux autres car le nombre de faces est moindre. Pour le maillage mixte, nous avons appliqué l'opération de triangulation à une face rectangulaire sur deux. Concernant les critères utilisés, on peut remarquer que le critère global est plus rapide que l'atomique (voir figure 7.18), car ce dernier nécessite de stocker toutes les forces atomiques dans le modèle au lieu de les cumuler et de ne stocker que le bilan des forces dans chaque sommet.

7.6 Conclusion

Nous avons montré dans ce chapitre la compatibilité de notre approche à simuler des modifications topologiques comme la déchirure et la fracture. Dans une première

partie, nous avons montré que ces modifications se basent sur une simple opération topologique de G-carte qui est la décousure α_2 en 2D et α_3 3D. Ensuite, nous avons présenté un ensemble d'opérations topologiques qui se basent sur cette opération fondamentale. Cependant, certaines de ces modifications comme l'éclatement de sommet et la suppression d'une arête impliquent beaucoup d'éléments adjacents et ne sont pas assez locales. Nous avons ensuite présenté les différents critères implantés dans notre plate-forme. Certains d'entre eux sont spécifiques à un modèle mécanique et d'autres sont généraux. Nous avons également proposé des critères permettant de contrôler les transformations élémentaires, ce qui permet une identification entre le lieu du critère et celui de la transformation. Avec cette plate-forme, l'utilisateur peut choisir le critère qu'il veut et la modification de maillage la plus appropriée. Finalement, quelques exemples de combinaisons de critères et transformations nous ont permis de montrer l'utilité de nos résultats. Comme Jerboa est uniquement un outil de prototypage rapide, un portage de notre modèle a été effectué en C++, mais pour des modèles spécifiques uniquement. Ce portage a permis de montrer l'efficacité de notre modélisation et son modèle de calcul d'interaction.

Conclusion et perspectives

Nous avons proposé dans ce manuscrit une plate-forme générale qui permet la simulation physique de corps déformables maillés en se basant sur un modèle topologique de type cartes généralisées (G-cartes) et un langage à base de règles de transformations de graphes, Jerboa. Nous avons choisi les G-cartes, car, en plus de leur définition mathématique, elles sont plus générales et atomiques que d'autres modèles topologiques et elles fournissent plus d'orbites pour stocker les informations. Ce modèle nous a permis de plonger directement tout type de données mécaniques, de paramètres d'interaction et de forces dans le modèle topologique, sans utilisation de structure externe. À cette fin, nous avons proposé un paradigme général à base de motif pour représenter tout type d'interaction. Jerboa, par son langage à base de règles, s'est révélé être un outil de prototypage particulièrement adapté pour reconnaître ces motifs, et nous a permis de construire un modèle général, permettant de choisir la dimension (2 ou 3), le type de mailles (rectangulaires, triangulaires, tétraédriques, hexaédriques, maillage hétérogène) et le type de lois de déformation, en d'autres termes, le type d'interaction (mécanique discrète ou continue, isotropie). Les interactions sont décrites à l'aide de graphes, de façon la plus générale possible, ce qui leur permet leur adaptation à la dimension et aux mailles du maillage.

Nous avons utilisé Jerboa car nous désirions prototyper rapidement nos propositions. Jerboa nous a permis de définir des règles non seulement pour simuler les corps physiques, mais également pour les construire. La version utilisée de Jerboa, programmée en Java, n'est pas dédiée au calcul rapide et ne génère pas de code optimisé pour implanter des règles. Cependant, quelques modèles spécifiques exploitant la généralité de (seulement) quelques uns des modèles d'interactions que nous avons proposés, ont été implantés en C++ et ont prouvé l'efficacité de notre méthode de gestion des interactions. Cette expérimentation a permis également de vérifier que le temps d'implantation d'un modèle avec les règles de Jerboa est plus court que celui de sa programmation manuelle. Compte tenu des résultats de simulation Java et C++, nous avons proposé quelques optimisations pour avoir un temps de calcul de simulation plus rapide, mais ces optimisations nécessitent des plongements qu'il faut mettre à jour après une transformation du maillage.

Dans ces travaux, nous avons également enrichi notre plate-forme avec la prise en compte des déchirures et des fractures. Nous avons proposé un modèle général, modulaire, où les déchirures et fractures se composent d'un critère et d'une transformation topologique à appliquer lorsque le critère est vérifié. Cette approche exploite complètement le langage à base de règles, puisque le critère apparaît comme une condition d'application d'une transformation topologique. Notre approche est

modulaire dans le sens où elle permet de choisir un critère et une modification topologique à appliquer. De nombreux types de critères ont été implémentés, certains sont spécifiques à un modèle mécanique donné et d'autres s'appliquent à tout modèle d'interactions. Pour produire des modifications de maillage lorsqu'un critère est vérifié, nous avons implanté plusieurs types de transformations topologiques, mais toutes se basent sur la séparation d'éléments, qui est l'opération fondamentale dans le cas des fractures/déchirures. Nous avons montré que cette opération ne modifiait aucune des interactions élémentaires que nous avons définies. Seules les optimisations (cumuls de masse, cumuls d'interaction) nécessitent des mises à jour. Le lien entre le lieu du critère et le lieu de la modification est explicité au travers du motif de filtrage de la règle. Cependant, ce lien pouvant ne pas être direct, voire se révéler arbitraire, nous avons proposé d'utiliser des critères et des modifications topologiques qui s'appliquent sur la même cellule topologique (face en 3D, arête en 2D).

Notre plate-forme permet de simuler des objets 2D ou 3D, mais nous ne proposons pas une approche gérant, de façon unifiée, la 2D et la 3D. Tout d'abord, Jerboa ne propose pas de modèle générique en dimension : il est nécessaire de préciser la dimension du modèle lors de sa création. Par ailleurs, on constate que certaines règles sont nécessairement spécifiques à la dimension : la séparation d'éléments consiste à éclater une arête en 2D et une face en 3D, les deux règles sont fondamentalement différentes. La suppression d'un élément (volume ou face) est un autre exemple d'une telle distinction. Notre approche s'est néanmoins attachée à rendre le système aussi souple que possible par rapport à la dimension. Ceci s'est traduit tout d'abord par le fait de choisir d'implanter tous les modèles en 3D (topologique). Ce choix nous a permis de mettre en commun certaines règles en les écrivant dans le cas 3D, de façon compatible avec le cas 2D (règles de calcul des ressorts d'élongation ou de cisaillement surfacique, par exemple). Néanmoins, les orbites ont des significations différentes selon la dimension (par exemple, la même orbite désigne un volume en 3D et une composante connexe en 2D) ce qui a un impact direct sur la façon de plonger les informations dans le modèle. Certains plongements peuvent être définis de façon indépendante de la dimension (ressorts d'élongation cumulés ou non), d'autres sont spécifiques à la dimension (masses-coins stockées dans les sommets de faces ou de volumes selon les cas). La question de la généralité en dimension reste donc ouverte, mais dépasse largement le simple cadre de la modélisation physique à base topologique, puisqu'elle repose, au moins en partie, sur une désignation, plus souple, aussi indépendante que possible de la dimension, des orbites.

Comme perspectives de notre travail, nous souhaitons achever l'implantation de certaines interactions non encore prises en compte dans notre plate-forme, comme par exemple la simulation de maillage hexaédriques avec les modèles de la mécanique continue (éléments finis linéaires, approche par co-rotationnel, masses/tenseurs), ou les ressorts de cisaillement volumique anisotropes. Nous avons discuté, dans ce document, de la partie théorique de leur implantation, mais nous n'avons pas pu fournir

d'illustrations dans nos exemples de simulation. Ensuite, une nouvelle version de notre simulateur est en cours de portage sur la nouvelle version de Jerboa. Cette version permet une prise en charge des plongements au travers d'un langage dédié, ce qui permet de porter le simulateur non seulement en Java, mais également en C++. Les premiers résultats de cette version, reprogrammée par un stagiaire de Master informatique en première année, sont prometteurs. Enfin, et à plus long terme, nous souhaitons également enrichir notre outil avec d'autres types de modifications topologiques, plus complexes, comme la résolution adaptative des éléments du maillage via un mécanisme de subdivision.

Bibliographie

- [BALB14] H. BELHAOUARI, A. ARNOULD, P. LE GALL & T. BELLET – « Jerboa : A graph transformation library for topology based geometric modeling », in *International Conference on Graph Transformations* (York, Royaume Uni), LNCS, no. 8571, 2014, p. 269–284. 80
- [Bar84] A. H. BARR – « Global and local deformations of solid primitives », *Computer Graphics (ACM SIGGRAPH)* **18** (1984), no. 3, p. 21–30. 8, 16
- [Bau75] B. BAUMGART – « A polyhedron representation for computer vision », *AFIPS, National Computer Conference and Exposition* (1975), p. 589–596. 60
- [BC00] D. BOURGUIGNON & M. CANI – « Controlling anisotropy in mass-spring systems. », in *Eurographics Workshop on Computer Animation and Simulation* (Interlaken, Suisse), 2000, p. 113–123. 24
- [BDW13] O. BUSARYEV, T. K. DEY & H. WANG – « Adaptive fracture simulation of multi-layered thin plates », *ACM Transactions on Graphics (ACM SIGGRAPH)* **32** (2013), no. 4, p. 52 :1–6. 43, 48, 49
- [Bel12] T. BELLET – « Transformations de graphes pour la modélisation géométrique à base topologique », Thèse, Université de Poitiers, 2012. 64, 66, 76
- [BG00] D. BIELSER & M. H. GROSS – « Interactive simulation of surgical cuts », in *Pacific Graphics Conference*, 2000, p. 116–125. 40
- [BHTF07] Z. BAO, J.-M. HONG, J. TERAN & R. FEDKIW – « Fracturing rigid materials », *IEEE Transactions on Visualization and Computer Graphics* **13** (2007), no. 2, p. 370–378. 43
- [BL00] F. BOUX DE CASSON & C. LAUGIER – « Simulating 2D tearing phenomena for interactive medical surgery simulators », in *Computer Animation Conference*, 2000, p. 9–14. 43, 44, 45
- [BMF03] R. BRIDSON, S. MARINO & R. FEDKIW – « Simulation of clothing with folds and wrinkles », in *ACM/Eurographics Symposium on computer animation* (San Diego, Californie, USA), 2003, p. 28–36. 21, 114, 158
- [BNC96] M. BRO-NIELSEN & S. COTIN – « Real-time volumetric deformable models for surgery simulation using finite elements and condensation », in *Eurographics Conference*, 1996, p. 57–66. 28

- [BPA⁺10] T. BELLET, M. PLOUDRET, A. ARNOULD, L. FUCHS & P. LE GALL – « Designing a topological modeler kernel : A rule-based approach », in *Shape Modeling International Conference* (Aix en Provence, France), 2010, p. 100–112. 79
- [BS85] R. P. BRYANT & D. SINGERMAN – « Foundations of the theory of maps on surfaces with boundaries », *The Quarterly Journal of Mathematics* **36** (1985), no. 141, p. 17–41. 61
- [BTG15] E. BOHL, O. TERRAZ & D. GHAZANFARPOUR – « Modeling fruits and their internal structure using parametric 3G-map L-systems », *The Visual Computer* **31** (2015), no. 6-8, p. 819–829. 69
- [BW98] D. BARAFF & A. WITKIN – « Large steps cloth simulation », in *ACM SIGGRAPH Conference*, 1998, p. 43–54. 21, 36, 110, 114
- [BWK03] D. BARAFF, A. P. WITKIN & M. KASS – « Untangling cloth », *ACM Transactions on Graphics (ACM SIGGRAPH)* **22** (2003), no. 3, p. 862–870. 22
- [CD99] D. CAZIER & J.-F. DUFOURD – « A formal specification of geometric refinements », *Visual Computer* **15** (1999), p. 279–301. 98
- [CDA99] S. COTIN, H. DELINGETTE & N. AYACHE – « Real-time elastic deformations of soft tissues for surgery simulation », *IEEE Transactions On Visualization and Computer Graphics* (1999), p. 62–73. vii, 29, 31
- [CDA00] —, « A hybrid elastic model allowing realtime cutting, deformation and force feedback for surgery training and simulation », *Visual Computer* (2000), p. 437–452. vii, 28, 38, 39, 45, 123, 168
- [CGo] CGoGN – <http://cgogn.unistra.fr>. 94
- [CHP89] J. E. CHADWICK, D. R. HAUMANN & R. E. PARENT – « Layered construction for deformable animated characters », *Computer Graphics (ACM SIGGRAPH)* **23** (1989), no. 3, p. 243–252. 16
- [CLF90] C. CADOZ, A. LUCIANI & J.-L. FLORENS – « Cordis-anima : système de modélisation et de simulation d’instruments et d’objets physiques pour la création musicale et l’image animée », in *Editions de la Maison des Sciences de l’Homme et ACROE, Colloque Modèles physiques* (Grenoble, France), 1990, p. 585–628. vii, 18
- [DC95] M. DESBRUN & M. CANI – « Animating soft substances with implicit surfaces », in *ACM SIGGRAPH Conference* (Los Angeles, Californie, USA), 1995, p. 287–290. 18, 19

- [DDBC99] G. DEBUNNE, M. DESBRUN, A. H. BARR & M.-P. CANI – « Interactive multiresolution animation of deformable models », in *Eurographics Workshop on Computer Animation and Simulation* (Milan, Italie), 1999, p. 133–144. 25
- [DDCB00] G. DEBUNNE, M. DESBRUN, M.-P. CANI & A. H. BARR – « Adaptive simulation of soft bodies in real-time », in *Computer Animation Conference* (Philadelphia, Pennsylvanie, USA), 2000, p. 15–20. 25, 49, 50
- [DDCB01] — , « Dynamic real-time deformations using space & time adaptive sampling », in *ACM SIGGRAPH Conference* (Los Angeles, Californie, USA), 2001, p. 31–36. viii, 49, 50
- [Der12] A. DERAM – « Environnement générique pour la validation de simulations médicales », Thèse, Université de Grenoble, 2012. vii, 14
- [DKS⁺11] E. DARLES, S. KALANTARI, X. SKAPIN, B. CRESPIAN & A. LUCIANI – « Hybrid physical topological modeling of physical shapes transformations », in *Workshop on Digital Media and Digital Content Management*, 2011. 19, 43, 44, 45, 92, 188
- [DL14] G. DAMIAND & P. LIENHARDT – *Combinatorial maps : Efficient data structures for computer graphics and image processing*, A K Peters/CRC Press, 2014. 61
- [DSB99] M. DESBRUN, P. SCHRÖDER & A. H. BARR – « Interactive animation of structured deformable objects », in *Graphics Interface Conference* (Kingston, Ontario, Canada), 1999, p. 1–8. 36
- [FDA05] C. FOREST, H. DELINGETTE & N. AYACHE – « Removing tetrahedra from manifold tetrahedralisation : application to real-time surgical simulation », *Medical Image Analysis* **9** (2005), no. 2, p. 113–122. 38, 59, 98
- [FGBP11] F. FAURE, B. GILLES, G. BOUSQUET & D. K. PAI – « Sparse meshless models of complex deformable solids », *ACM Transactions on Graphics (ACM SIGGRAPH)* **30** (2011), no. 4, p. 73 :1–10. 18
- [FLA⁺05] L. FRANCE, J. LENOIR, A. ANGELIDIS, P. MESEURE, M. CANI, F. FAURE & C. CHAILLOU – « A layered approach for surgery simulation of a virtual human intestine », *Medical Image Analysis* **9** (2005), no. 2, p. 123–132. 17
- [FM03] T.-P. FRIES & H.-G. MATTHIES – « Classification and overview of meshfree methods », Tech. report, TU brunswick, Allemagne, 2003. 19

- [Fri99] S. F. FRISKEN-GIBSON – « Using linked volumes to model object collisions, deformation, cutting, carving, and joining », *IEEE Transactions on Visualization and Computer Graphics* **5** (1999), no. 4, p. 333–348. 39
- [FZDJ13] E. FLÉCHON, F. ZARA, G. DAMIAND & F. JAILLET – « A generic topological framework for physical simulation », in *WSCG Conference*, 2013, p. 104–113. 95, 108, 111
- [FZDJ14] — , « A unified topological-physical model for adaptive refinement », in *VRIPHYS Conference* (Bremen, Allemagne), 2014, p. 39–48. vii, 23, 50, 105
- [GBA15] V. GAUTHIER, H. BELHAOUARI & A. ARNOULD – « Évaluation de la modélisation à base de transformation de graphes avec Jerboa », in *Journées du Groupe de Travail en Modélisation Géométrique* (Poitiers), 2015. 80
- [GCZD15] K. GOLEC, M. COQUET, F. ZARA & G. DAMIAND – « Improvement of a topological-physical model to manage different physical simulation », in *WSCG Conference*, 2015, p. 25–34. 97, 108, 111
- [GHDS03] E. GRINSPUN, A. N. HIRANI, M. DESBRUN & P. SCHRODER – « Discrete shells », in *ACM/Eurographics Symposium on Computer Animation* (San Diego, Californie, USA), 2003, p. 62–67. 21
- [GMTT89] J.-P. GOURRET, N. MAGNENAT-THALMANN & D. THALMANN – « Simulation of object and human skin deformations in a grasping task », *Computer Graphics (ACM SIGGRAPH)* **23** (1989), no. 3, p. 21–30. 28
- [Gol18] K. GOLEC – « Hybrid 3d mass spring system for soft tissue simulation », Thèse, Université Claude Bernard Lyon 1, 2018. 119
- [GQ05] X. GUO & H. QIN – « Real-time meshless deformation », *Journal of Visualization and Computer Animation* **16** (2005), no. 3-4, p. 189–200. 19
- [GR09] C. GEUZAIN & J. REMACLE – « Gmsh : a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities », *International Journal for Numerical Methods in Engineering* **79** (2009), no. 11, p. 1309–1331. 58, 132
- [Gui06] N. GUIARD – « Construction de modèles géologiques 3d par co-raffinement de surfaces », Thèse, Ecole des Mines de Paris, 2006. 47
- [HB99] D. H. HOUSE & D. E. BREEN – « Cloth modeling and animation », *Journal on Graphics, GPU & Game Tools* **4** (1999), no. 4, p. 51. 22

- [HPH96] D. HUTCHINSON, M. PRESTON & T. HEWITT – « Adaptive reffinement for mass/spring simulations », in *Eurographics Workshop on Computer Animation and Simulation* (Poitiers, France), 1996, p. 31–45. viii, 48, 50
- [HTK00] K. HIROTA, Y. TANOUE & T. KANEKO – « Simulation of three-dimensional cracks », *Visual Computer* (2000), p. 371–378. 42, 45
- [IO09] H. N. IBEN & J. F. O'BRIEN – « Generating surface crack patterns », *Graphical Models* (2009), p. 198–208. viii, 43, 47, 48
- [JAD⁺12] T. JUND, A. ALLAOUI, E. DARLES, X. SKAPIN, P. MESEURE & A. LUCIANI – « Mapping volumetric meshes to point-based motion models », in *VRIPhys Workshop*, 2012, p. 11–20. 92
- [JP99] D. L. JAMES & D. K. PAI – « Artdefo : Accurate real time deformable objects », in *ACM SIGGRAPH Conference* (Los Angeles, Californie, USA), 1999, p. 65–72. 28
- [JSV97] F. JAILLET, B. SHARIAT & D. VANDORPE – « Deformable volume object modeling with a particle-based system for medical applications », in *WSCG Conference* (Pilsen, République Tchèque), 1997, p. 192–201. 18
- [KLB14] D. KOSCHIER, S. LIPPONER & J. BENDER – « Adaptive tetrahedral meshes for brittle fracture simulation », in *ACM/Eurographics Symposium on Computer Animation* (Copenhague, Danemark), 2014, p. 57–66. 43, 44, 45, 48, 49, 191
- [KLC14] S. KALANTARI, A. LUCIANI & N. CASTAGNÉ – « A new way to model physics-based topology transformations : Splitting mat », in *Smart Graphics Conference*, 2014, p. 146–153. 92
- [Kra08] P. KRAEMER – « Modèles topologiques pour la multirésolution », Thèse, Université Louis Pasteur – Strasbourg I, 2008. 50
- [LBC⁺14] J. A. LEVINE, A. W. BARGTEIL, C. CORSI, J. TESSENDORF & R. GEIST – « A peridynamic perspective on spring-mass fracture », in *ACM/Eurographics Symposium on Computer Animation*, 2014, p. 47–55. 18, 43
- [LCM⁺14] A. LUCIANI, N. CASTAGNÉ, P. MESEURE, X. SKAPIN, S. KALANTARI & E. DARLES – « A new modeling pipeline for physics-based topological discontinuities : The dyanme process », in *Smart Graphics Conference*, 2014, p. 178–181. 92
- [LFMC02] J. LENOIR, L. FRANCE, P. MESEURE & C. CHAILLOU – « Simulation of a minimally invasive surgery of intestines », in *Virtual Reality International Conference*, 2002, p. 17–23. 17

- [LHLW11] N. LIU, X. HE, S. LI & G. WANG – « Meshless simulation of brittle fracture », *Computer Animation and Virtual Worlds* **22** (2011), no. 2–3, p. 115–124. 43
- [Lie89] P. LIENHARDT – « Subdivisions of n -dimensional spaces and n -dimensional generalized maps », in *Annual Symposium on Computational Geometry* (Saarbrücken, Allemagne), 1989, p. 228–236. 64
- [Lie91] — , « Topological models for boundary representation : a comparison with n -dimensional generalized maps », *Computer-Aided Design* **23** (1991), no. 1, p. 59–82. 64
- [LJF⁺91] A. LUCIANI, S. JIMENEZ, J.-L. FLORENS, C. CADOZ & O. RAOULT – « Computational physics : a modeler-simulator for animated physical objects », in *Annual Conference of the European Association for Computer Graphics*, 1991, p. 425–436. 9, 18
- [LMA⁺02] F. LEDOUX, J. MOTA, A. ARNOULD, C. DUBOIS, P. LE GALL & Y. BERTRAND – « Spécifications formelles du chanfreinage », *Technique et Science Informatiques* **21** (2002), no. 8, p. 1073–1098. 98
- [LTH86] D. H. LAIDLAW, W. B. TRUMBORE & J. F. HUGHES – « Constructive solid geometry for polyhedral objects », *Computer Graphics (ACM SIGGRAPH)* **20** (1986), no. 4, p. 161–170. 54
- [Luc85] A. LUCIANI – « Un outil informatique de création d’images animées. modèles d’objets, langage, contrôle gestuel en temps réel. le système ANIMA. », Thèse, Université de Grenoble, 1985. 8
- [Man88] M. MANTYLA – *An introduction to solid modeling*, Computer Science Pres, 1988. 60
- [MBF04] N. MOLINO, Z. BAO & R. FEDKIW – « A virtual node algorithm for changing mesh topology during simulation », *ACM Transactions on Graphics (ACM SIGGRAPH)* **23** (2004), no. 3, p. 385–392. 40, 45
- [MCK13] M. MULLER, N. CHENTANEZ & T. Y. KIM – « Real time dynamic fracture with volumetric approximate convex decompositions », *ACM Transactions on Graphics (ACM SIGGRAPH)* **32** (2013), no. 4, p. 115 :1–10. 47
- [MDM⁺02] M. MULLER, J. DORSEY, L. McMILLAN, R. JAGNOW & B. CUTLER – « Stable real time deformations », in *ACM/Eurographics Symposium on Computer Animation* (San Antonio, Texas, USA), 2002, p. 49–54. 189
- [MDS10] P. MESEURE, E. DARLES & X. SKAPIN – « Topology-based physical simulation », in *VRIPHYS Conference*, 2010, p. 1–10. 50, 93, 95, 96, 105, 108, 110, 111, 181, 186

- [MDST15] P. MESEURE, E. DARLES, X. SKAPIN & Y. TOUILEB. – « Adaptive resolution for topology modifications in physically-based animation », Tech. report, XLIM lab, 2015. 94
- [MG04] M. MULLER & M. GROSS – « Interactive virtual materials », in *Graphics Interface Conference*, 2004, p. 239–246. 18, 32, 122
- [MHTG05] M. MÜLLER, B. HEIDELBERGER, M. TESCHNER & M. H. GROSS – « Meshless deformations based on shape matching », *ACM Transactions on Graphics (ACM SIGGRAPH)* **24** (2005), no. 3, p. 471–478. vii, 18, 19
- [Mil88] G. S. P. MILLER – « The motion dynamics of snakes and worms », *Computer Graphics (ACM SIGGRAPH)* **22** (1988), no. 4, p. 169–173. 8
- [MMD⁺01] M. MULLER, L. McMILLAN, J. DORSEY, & R. JAGNOW – « Real time simulation of deformation and fracture of stiff materials », in *Eurographics Workshop on Animation and Simulation*, 2001, p. 113–124. 43, 44, 45
- [Mok] MOKA – <http://moka-modeller.sourceforge.net>. 94, 197
- [Mon92] J. MONAGHAN – « Smoothed particle hydrodynamics », *Annu. Rev. Astro. Astrophys* **30** (1992), p. 543–574. 18
- [MP89] G. S. P. MILLER & A. PEARCE – « Globular dynamics : A connected particle system for animating viscous fluids », *Computers & Graphics* **13** (1989), no. 3, p. 305–309. 18
- [Nes08] M. NESME – « Milieu mécanique déformable multirésolution pour la simulation interactive », Thèse, Université Joseph Fourier Grenoble 1, 2008. vii, 12
- [NPF05] M. NESME, Y. PAYAN & F. FAURE – « Efficient, physically plausible finite elements », in *Eurographics Conference*, 2005. vii, 31, 32, 162
- [NTB⁺91] A. NORTON, G. TURK, R. BACON, J. GERTH & P. SWEENEY – « Animation of fracture by physical modeling », *Visual Computer* (1991), p. 210–219. viii, 41, 42, 44, 45, 188
- [NvdS00] H.-W. NIENHUYS & F. VAN DER STAPPEN – « Combining finite element deformation with cutting for surgery simulations », in *Eurographics Conference*, 2000, p. 43–52. 39, 93
- [NvdS02] H. NIENHUYS & A. F. VAN DER STAPPEN – « A delaunay approach to interactive cutting in triangulated surfaces », in *Algorithmic Foundations of Robotics*, 2002, p. 113–130. 45

- [OBH02] J. F. O'BRIEN, A. W. BARGTEIL & J. K. HODGINS – « Graphical modeling and animation of ductile fracture », *ACM Transactions on Graphics (ACM SIGGRAPH)* **21** (2002), no. 3, p. 291–294. 43, 44, 45
- [OH99] J. F. O'BRIEN & J. K. HODGINS – « Graphical modeling and animation of brittle fracture », in *ACM SIGGRAPH Conference*, 1999, p. 137–146. 41, 43, 44, 45, 187
- [PACL08] M. POUDRET, A. ARNOULD, J.-P. COMET & P. LE GALL – « Graph transformation for topology modelling », in *International Conference on Graph Transformations* (Leicester, Royaume Uni), 2008, p. 147–161. 71
- [PDA00] G. PICINBONO, H. DELINGETTE & N. AYACHE – « Real-time large displacement elasticity for surgery simulation : Non-linear tensor-mass model », in *Conference on Medical Image Computing and Computer-Assisted Intervention*, 2000, p. 643–652. 31
- [PDA01] — , « Non-linear and anisotropic elastic soft tissue models for medical simulation », in *IEEE International Conference on Robotics and Automation* (Séoul, Corée), 2001, p. 1370–1375. 31
- [PL90] P. PRUSINKIEWICZ & A. LINDENMAYER – *The algorithmic beauty of plants*, The virtual laboratory, Springer, 1990. 68
- [PNdO14] T. PFAFF, R. NARAIN, J. DE JOYA & J. O'BRIEN – « Adaptive tearing and cracking of thin sheets », *ACM Transactions on Graphics (ACM SIGGRAPH)* **33** (2014), no. 4, p. 110 :1–9. 43, 48
- [PO09] E. G. PARKER & J. F. O'BRIEN – « Real-time deformation and fracture in a game environment », in *ACM/Eurographics Symposium on Computer Animation*, 2009, p. 165–175. 43, 47
- [Pou09] M. POUDRET – « Transformations de graphes pour les opérations topologiques en modélisation géométrique : Application à l'étude de la dynamique de l'appareil de golgi », Thèse, Université d'Evry val d'Essonne, 2009. 64, 76
- [Pro95] X. PROVOT – « Deformation constraints in a mass spring model to describe rigid cloth behaviour », in *Graphics Interface Conference*, 1995, p. 147–154. vii, 21, 22, 24, 48, 109, 114
- [Pro97] — , « Animation réaliste de vêtements », Thèse, Université Paris Descartes, 1997. 35
- [PUC⁺15] C. J. PAULUS, L. UNTEREINER, H. COURTECUISSÉ, S. COTIN & D. CAZIER – « Virtual cutting of deformable objects based on efficient topological operations », *Visual Computer* (2015), p. 831–841. 46, 95

- [PW89] A. PENTLAND & J. WILLIAMS – « Good vibrations : model dynamics for graphics and animation », *Computer Graphics (ACM SIGGRAPH)* **23** (1989), no. 3, p. 215–222. 16
- [RCDL18] J. RIFFET, N. CASTAGNÉ, E. DARLES & A. LUCIANI – « A topological-geometrical pipeline for 3d cracking-like phenomena », in *VISIGRAPP/GRAPP Conference* (Funchal, Portugal), 2018. 92
- [RNG00] Y. RÉMION, J. NOURRIT & D. GILLARD – « A dynamic animation engine for generic spline objects », *Journal of Visualization and Computer Animation* **11** (2000), no. 1, p. 17–26. 17
- [RNN00] Y. RÉMION, J. NOURRIT & O. NOCENT – « Dynamic animation of n-dimensional deformable objects », in *WSCG Conference* (Pilsen, République Tchèque), 2000. 17
- [SDF07] E. SIFAKIS, K. G. DER & R. FEDKIW – « Arbitrary cutting of deformable tetrahedralized objects », in *ACM/Eurographics Symposium on Computer Animation* (San Diego, Californie, USA), 2007, p. 73–80. 40, 41
- [Sof] SOFA – <https://www.sofa-framework.org>. 33, 58, 92
- [SP86] T. W. SEDERBERG & S. R. PARRY – « Free-form deformation of solid geometric models », *Computer Graphics (ACM SIGGRAPH)* **20** (1986), no. 4, p. 151–160. 8, 16
- [SvWC14] M. S. SOUZA, A. VON WANGENHEIM & E. COMUNELLO – « Fast simulation of cloth tearing », *SBC Journal on Interactive Systems* (2014), p. 44–48. 93
- [SWB00] J. SMITH, A. P. WITKIN & D. BARAFF – « Fast and controllable simulation of the shattering of brittle objects », in *Graphics Interface Conference*, 2000, p. 27–34. 44, 45, 192
- [TF88] D. TERZOPOULOS & K. W. FLEISCHER – « Modeling inelastic deformation : viscoelasticity, plasticity, fracture », *Computer Graphics (ACM SIGGRAPH)* **22** (1988), no. 4, p. 269–278. 42
- [TGAB08] A. THEETTEN, L. GRISONI, C. ANDRIOT & B. A. BARSKY – « Geometrically exact dynamic splines », *Computer-Aided Design* **40** (2008), no. 1, p. 35–48. 17
- [TGG00] M. TESCHNER, S. GIROD & B. GIROD – « Direct computation of non-linear soft-tissue deformation », in *Vision Modeling and Visualization Conference* (Sarrebruck, Allemagne), 2000, p. 383–390. 22

- [TGM⁺09] O. TERRAZ, G. GUIMBERTEAU, S. MÉRILLOU, D. PLEMENOS & D. GHAZANFARPOUR – « 3gmap l-systems : an application to the modelling of wood », *The Visual Computer* **25** (2009), no. 2, p. 165–180. 69
- [TPBF87] D. TERZOPOULOS, J. PLATT, A. BARR & K. FLEISCHER – « Elastically deformable models », *Computer Graphics (ACM SIGGRAPH)* **21** (1987), no. 4, p. 205–214. 8, 25
- [VAB10] C. A. VANEGAS, D. G. ALIAGA & B. BENES – « Building reconstruction using manhattan-world grammars », in *computer Vision and Pattern Recognition (CVPR)*, 2010, p. 358–365. 69
- [VB05] J. VILLARD & H. BOROUCAKI – « Adaptive meshing for cloth animation », *Eng. Comput. (Lond.)* **20** (2005), no. 4, p. 333–341. 48
- [vG98] A. VAN GELDER – « Approximate simulation of elastic membranes by triangulated spring meshes », *Journal on Graphics Tools* (1998), p. 21–42. 23
- [WJST14] Y. WANG, C. JIANG, C. SCHROEDER & J. TERAN – « An adaptive virtual node algorithm with robust mesh cutting », in *ACM/Eurographics Symposium on Computer Animation* (Copenhagen, Denmark), 2014, p. 77–85. 41
- [WRK⁺10] M. WICKE, D. RITCHIE, B. M. KLINGNER, S. BURKE, J. R. SHEWCHUK & J. F. O'BRIEN – « Dynamic local remeshing for elastoplastic simulation », *ACM Transactions on Graphics (ACM SIGGRAPH)* **29** (2010), no. 4, p. 49 :1–11. 43, 48
- [WWD14] J. WU, R. WESTERMANN & C. DICK – « Physically-based simulation of cuts in deformable bodies : A survey », in *Eurographics State of the Art Reports* (Strasbourg, France), 2014, p. 1–19. vii, 40, 41
- [YC93] F. YUAN-CHENG – *Biomechanics : Mechanical properties of living tissues*, Springer verlag, 1993. 14
- [Zar03] F. ZARA – « Algorithmes parallèles de simulation physique pour la synthèse d'images : application à l'animation de textiles », Thèse, Institut National Polytechnique de Grenoble, 2003. vii, 20

Résumé :

La simulation physique des objets déformables est au cœur de plusieurs applications dans l'informatique graphique. Dans ce contexte, nous nous intéressons à l'élaboration d'une plate-forme, qui combine le modèle topologique des Cartes Généralisées avec un ou plusieurs modèles mécaniques, pour l'animation physique d'objets maillés déformables, pouvant endurer des transformations topologiques comme des déchirures ou des fractures.

Pour offrir un cadre aussi général que possible, nous avons adopté une approche à base de règles de manipulation et de transformation de graphes, telle que proposée par le logiciel Jerboa. Cet environnement offre des possibilités de prototypage rapide de différents modèles mécaniques. Il nous a permis de définir précisément le stockage des informations mécaniques dans la description topologique du maillage et de simuler les déformations en utilisant une base topologique pour le calcul des interactions et l'affectation des forces. Toutes les informations mécaniques sont ainsi stockées dans le modèle topologique, sans recours à une structure auxiliaire.

La plate-forme réalisée est générale. Elle permet de simuler des objets 2D ou 3D, avec différents types de maillages, non nécessairement homogènes. Elle permet de simuler différents modèles mécaniques, continus ou discrets, avec des propriétés diverses d'homogénéité et d'isotropie. De plus, différentes solutions d'évolution topologique ont été proposées. Elles impliquent la sélection d'un critère déclenchant les modifications et un type de transformation proprement dit. Notre approche a également permis de réduire les mises à jour du modèle mécanique en cas de déchirure/fracture.

Mots clés : simulation physique, modèle topologique, langage à base de règles, transformations topologiques.

Abstract:

The physical simulation of deformable objects is at the core of several computer graphics applications. In this context, we are interested in the creation of a framework that combines a topological model, namely Generalized Maps, with one or several mechanical models, for the physical animation of deformable meshed objects that can undergo topological modifications such as tearing or fractures.

To obtain a general framework, we chose to rely on graph manipulation and transformation rules, proposed by the Jerboa software. This environment provided us with fast prototyping facilities for different mechanical models. It allowed us to precisely define how to store mechanical properties in the topological description of a mesh and simulate its deformation in a topologically-based manner for interaction computation and force distribution. All mechanical properties are stored in the topological model without any external structure.

This framework is general. It allows for the simulation of 2D or 3D objects, with different types of meshes, including non homogeneous ones. It also allowed for the simulation of several, continuous or discrete, mechanical models with various properties of homogeneity and isotropy. Furthermore, different methods to simulate topological modifications have been implemented in the framework. They include both the selection of a criterion to trigger topological modifications and a transformation type. Our approach also managed to reduce the number of updates of the mechanical model after tearing / fracture.

Keywords: Physical animation, topologically-based simulation, rule-based language, prototyping, topological transformation