

UNIVERSITÉ de REIMS CHAMPAGNE ARDENNE

INSTITUT DE FORMATION TECHNIQUE SUPERIEURE
ÉCOLE DOCTORALE SCIENCES EXACTES et BIOLOGIE

THÈSE

Pour l'obtention du grade de

Docteur de l'Université de Reims Champagne Ardenne

(Spécialité informatique)

soutenue publiquement le

26 mai 2005

par

Frédéric HESCHUNG

Vers une construction et une adaptation automatiques des maillages

Jury

Président du jury	Dominique DENEUX	Professeur à l'Université de Valenciennes
Directeur de thèse	Yvon GARDAN	Professeur à l'Université de Reims
Co-encadrant	Christian MINICH	Maître de conférence à l'Université de Metz
Rapporteurs	Dominique BECHMANN	Professeure à l'Université de Strasbourg I
	Behzad SHARIAT	Professeur à l'Université de Lyon I

À mes parents,

Remerciements

Maintenant que cette thèse touche à sa fin, c'est avec bonheur que je rédige ces quelques lignes.

Tout d'abord je voudrais remercier M. Gardan de m'avoir accepté au sein de son équipe, et d'avoir fait ce qui lui était possible pour que j'y reste jusqu'à présent. Je dois également beaucoup à Christian Minich pour sa disponibilité, pour l'attention qu'il a portée à mon travail durant ces années, pour ses nombreuses relectures et corrections, ainsi que sa bonne humeur très motivante.

Je tiens à remercier également Mme Dominique Bechmann et M. Behzad Shariat d'avoir bien voulu juger mon travail en étant rapporteur, ainsi que M. Dominique Deneux d'avoir accepté d'être examinateur.

Ces cinq années à l'IFTS sont marqués par le passage de Romuald, Jérôme, Murielle, Adeline, et Erwan, ainsi que par les personnes qui y sont encore présentes : Lolo, Naomi, Mael, Reim, Philippe, le Mawhot', Mohand, Stéphane, Nico, Fred, Steph, et tous les autres, pardon de ne pas pouvoir vous citer tous et merci pour tout.

Merci à mes proches pour leur soutien.

Merci à Fadila.

Sommaire

INTRODUCTION	5
CHAPITRE I : ÉTAT DE L'ART.....	10
INTRODUCTION	10
I.1 UTILISATIONS ET TYPES DE MAILLAGE	10
I.1.1 DÉFINITION	10
I.1.2 MAILLAGES STRUCTURES ET NON STRUCTURES	11
I.1.3 APPLICATIONS EXPLOITANT UN MAILLAGE	12
I.2 METHODES DE MAILLAGE.....	17
I.2.1 MÉTHODES DE CONSTRUCTION DE MAILLAGES	17
I.2.2 ADAPTATION DE MAILLAGES	38
I.3 TYPES ABSTRAITS POUR LA MODÉLISATION DE FORMES.....	47
I.3.1 COURBES ET SURFACES PARAMETRIQUES	48
I.3.2 FONCTIONS IMPLICITES	50
CONCLUSION	53
CHAPITRE II : CONCEPTS POUR L'INTÉGRATION DES APPLICATIONS EXPLOITANT UN MAILLAGE.....	56
II.1 PROBLÉMATIQUE	56
II.1.1 INTERACTIONS ENTRE AEM	56
II.1.2 ÉCHANGE D'INFORMATIONS GÉOMETRIQUES	60
II.1.3 OBJECTIFS ET PROPOSITION	62
II.2 DESCRIPTION GÉNÉRALE DE L'ARCHITECTURE	63
II.3 UN SERVICE PARTICULIER : LES OPERATEURS D'ASSEMBLAGE	67
II.4 MODULE D'ASSEMBLAGE	68
II.4.1 ALGORITHMES D'ASSEMBLAGE	68
II.4.2 CHOIX DES ALGORITHMES ET DES PARAMETRES	72
II.4.3 LE PROJET DIJA	77
II.5 TRANSFERTS DU MAILLAGE ET GESTION DES MODIFICATIONS	81
II.5.1 INFORMATIONS À TRANSFÉRER À L'AEM	81
II.5.2 GESTION DES MODIFICATIONS	82
II.5.3 GESTION DES COMMUNICATIONS ENTRE MODULES	83
CONCLUSION	85
CHAPITRE III : APPROFONDISSEMENT DES OPÉRATEURS D'ASSEMBLAGE	88
III.1 OBJECTIFS ET PROBLÉMATIQUE	88
III.2 OPERATEURS DE COLLAGE, DE DÉCOUPAGE ET DE RAFFINEMENT.....	89

III.3 OPERATIONS BOOLÉENNES.....	90
III.3.1 GÉNÉRALITES	90
III.3.2 TEST D'APPARTENANCE D'UN POINT P A UN SOLIDE TRIANGLE	93
III.3.3 CALCUL DES POLYGONES D'INTERSECTION	95
III.4 REMAILLAGE.....	99
III.4.1 CONTEXTE	99
III.4.2 ÉTAT DE L'ART	100
III.4.3 TARF (TRIANGULATION PAR AVANCEMENTS ET RECVLS DE FRONTS)	103
III.4.4 AUTRES APPROCHES DU REMAILLAGE DES ZONES VIDES	108
III.5 CONCLUSION	114
CHAPITRE IV : ARCHITECTURE D'UNE BIBLIOTHÈQUE D'ALGORITHMES DE MAILLAGE GÉNÉRIQUES OU SPÉCIALISÉS (BAMGS)	117
IV.1 PROBLÉMATIQUE ET OBJECTIFS	117
IV.2 INDÉPENDANCE VIS-A-VIS DES AEM.....	119
IV.3 INDÉPENDANCE VIS-A-VIS DES MODÈLES GÉOMETRIQUES	121
IV.4 EXTENSIBILITE	123
IV.4.1 BIBLIOTHEQUES D'ALGORITHMES EXISTANTS	124
IV.4.2 CONCEPTS POUR LA PROGRAMMATION GÉNÉRIQUE	126
IV.4.3 PROPOSITION D'UN PATRON DE CONCEPTION	128
IV.4.4 EXEMPLE	133
IV.5 EXEMPLE.....	139
IV.6 DISCUSSION	140
IV.6.1 PATRON DE CONCEPTION	140
IV.6.2 BAMGS	141
IV.6.3 MISE EN OEUVRE	142
IV.6.4 MODULE D'ASSEMBLAGE ET GÉNÉRICITE	143
CONCLUSION.....	144
CHAPITRE V : MISE EN ŒUVRE.....	146
V.I OBJECTIFS	146
V.II TECHNIQUES DE MAILLAGE.....	147
V.II ARCHITECTURE IMPLEMENTÉE	148
V.II SERVICES GÉOMETRIQUES	151
V.II RÉSULTATS	154
IV.6.1 MAILLAGE POUR LA VISUALISATION	154
IV.6.1 MAILLAGE POUR LE PROTOTYPAGE RAPIDE	155
IV.6.1 MAILLAGE POUR LA SIMULATION	157
CONCLUSION.....	161
BIBLIOGRAPHIE	166
ANNEXE A : RAPPELS DE NOTATIONS UML (UNIFIED MODELING LANGUAGE)	I
ANNEXE B : REGLES D'UTILISATION DES SYSTÈMES DE CAO EN VUE D'UN TRANSFERT DE LA GÉOMETRIE	V
ANNEXE C : CONVERGENCE DE LA MÉTHODE PAR AVANCEMENT ET PAR RECVL DE FRONTS.....	VIII
ANNEXE D : CRITÈRE POUR LA QUALITE EN FORME	XX
ANNEXE E : SERVICES GÉOMETRIQUES	XXIII

Introduction

Les fonctionnalités des logiciels utilisés durant la conception d'un produit sont de plus en plus étendues : certains systèmes de CAO récents intègrent des solveurs numériques permettant de réaliser des calculs par éléments finis. De même des systèmes spécialisés initialement dans la simulation offrent parfois des services de modélisation géométrique. Ainsi les logiciels couvrent plusieurs activités telles la simulation, la modélisation géométrique ou le maillage ... Ils ne peuvent toutefois prétendre à être les meilleurs dans chaque domaine, si bien que les utilisateurs sont contraints d'utiliser plusieurs logiciels différents et, donc, d'échanger de l'information entre ces logiciels. Dans les informations échangées apparaissent de manière récurrente les maillages, indispensables aux outils de simulation et de prototypage rapide. Construire un maillage de bonne qualité et prêt à l'emploi est une opération difficile et coûteuse car elle exige des interventions manuelles. Il faut par exemple spécifier les conditions de chargement, les densités de mailles et corriger les erreurs dans le maillage car les fissures, recouvrements et inversions de mailles sont encore monnaie courante. Par exemple, une étude menée en 2002 dans un service d'étude de pièces par Steve Fenves (National Institute of Standards and Technology, USA) a établi que sept ingénieurs sur quinze ne faisaient pas de l'analyse mais du maillage : soit ils redessinaient la pièce CAO pour l'idéaliser et la préparer au maillage, soit ils corrigeaient les mailles produites par les logiciels de maillage.

D'autres applications que celles relevant purement du secteur de l'ingénierie s'appuient sur des maillages. C'est par exemple le cas des bibliothèques de visualisation les plus communément utilisées, qui admettent en entrée un ensemble de polygones convexes approchant la frontière des objets à visualiser. Globalement le maillage est donc une nécessité pour des applications significatives du monde de la CAO.

Pour la production de ces maillages, l'attitude actuelle est de développer un modelleur géométrique et un mailleur spécifique pour chaque application. La présente étude prend le contre-pied de cette approche en proposant un mailleur que nous qualifions de générique car il répond aux exigences suivantes : 1) être capable de produire de manière aussi automatique que possible un maillage de bonne qualité pour les trois types d'applications que nous utilisons pour illustrer ce mémoire : visualisation, prototypage rapide et analyse par éléments finis ; 2) s'adapter à plusieurs modèles géométriques, en particulier BRep et CSG et 3) être extensible, de façon à pouvoir s'adapter à d'autres applications exigeant un maillage, aux évolutions des besoins de chaque application et aux améliorations des techniques de maillage.

Le premier chapitre est consacré à un état de l'art. Il définit ce qu'est un maillage et présente les différentes applications qui exploitent un maillage (AEM), les différents types de maillage utilisés, ainsi que les méthodes de construction et d'adaptation associées. L'adaptation est l'opération consistant à optimiser une ou plusieurs propriétés d'un maillage ou à modifier le type des mailles, comme la transformation de triangles en quadrangles. Comme les algorithmes sur les maillages s'appuient sur divers modèles géométriques, ces derniers font également l'objet d'une brève présentation. Tous les algorithmes présentés sont analysés de manière à souligner leurs similitudes et à dégager des parties génériques. La nature très interdisciplinaire du sujet conduit à une étude bibliographique relativement volumineuse.

Le deuxième chapitre présente d'abord les difficultés rencontrées par les utilisateurs des trois applications citées plus haut. On constate que ces difficultés proviennent en partie de la mauvaise qualité géométrique des modèles transférés. En effet, la qualité géométrique influe directement sur la qualité du maillage et donc sur la qualité du résultat de l'AEM, qu'il s'agisse d'un calcul par éléments finis, d'une visualisation, réaliste ou sommaire, ou de la fabrication d'une pièce en prototypage rapide. Or, aujourd'hui, la représentation par les frontières est utilisée majoritairement comme modèle de transfert et comme modèle géométrique pour la construction des maillages. Nous en déduisons que les BReps sont à l'origine de la majorité des problèmes rencontrés. Ces constats mènent à une réflexion sur l'organisation des interactions entre les systèmes impliqués (CAO, mailleur et AEM) et débouchent sur l'objectif que nous nous fixons : concevoir un mailleur générique capable de fournir des maillages satisfaisant différents critères de qualité tout en s'abstrayant du modèle géométrique sous-jacent. Notre contribution apparaît donc dans ce chapitre au travers de la présentation de ce que nous appelons l'architecture externe du mailleur générique, qui décrit les interactions entre les différentes applications. Cette architecture permet de fournir un

maillage optimisé ou au moins adapté à l'utilisation qui en sera faite, en évitant les problèmes actuels de qualité des maillages.

Les interactions du mailleur avec les autres systèmes définis dans le deuxième chapitre ont des conséquences sur la manière de construire et d'adapter les maillages. Un des points clés de cette architecture est de construire des maillages pré-optimisés à partir de l'historique de construction, en évitant donc la construction d'une représentation par les frontières. Nous proposons en effet de réaliser les opérations booléennes directement sur les maillages en tenant compte de la pertinence des différentes parties de l'arbre de construction relativement à l'exploitation qui sera faite de ce maillage. Ceci a des conséquences sur les propriétés du maillage dans certaines zones. Dans le troisième chapitre nous nous attardons sur l'obtention d'un maillage respectant des contraintes de forme et d'agrandissement des mailles, comme cela peut être nécessaire pour un calcul par éléments finis. Le maillage est pré-optimisé entre autres parce que les mailles produites aux endroits des jonctions, où se concentrent les contraintes, sont construites judicieusement.

Le quatrième chapitre porte sur l'architecture interne du mailleur générique. Nous y présentons une organisation des différents algorithmes et structures de données impliqués dans les méthodes de maillage. Cette organisation exploite les similitudes et les spécificités de chaque méthode. Elle repose sur un nouveau patron de conception, dédié à la modélisation des algorithmes génériques. Une description détaillée des services qu'offre le mailleur est également réalisée, pour mettre en évidence la satisfaction des besoins exprimés dans le chapitre précédent. L'aspect générique du mailleur apparaît à deux niveaux : il peut tout d'abord s'appuyer sur tout type de modèle géométrique, à condition que ce dernier fournisse les fonctions géométriques nécessaires à la méthode de maillage utilisée ; de plus, d'éventuelles innovations dans les modèles géométriques, les méthodes de maillage ou les applications utilisatrices sont prises en compte par le mailleur sans que l'architecture soit remise en cause.

Ce chapitre est suivi de la présentation d'une maquette qui illustre certaines des propositions faites dans ce mémoire et, d'une certaine manière, contribue à leur validation. On montre la faisabilité des architectures externe et interne, en se restreignant à des modèles géométriques et à des méthodes de maillages significatifs.

Le manuscrit se termine sur une conclusion qui résume le document, mentionne les objectifs atteints et ouvre des perspectives.

Chapitre I : État de l'art

Introduction

La problématique évoquée dans l'introduction concerne les problèmes de construction et d'adaptation de maillage ainsi que de la qualité des modèles géométriques. Ces problèmes qui sont abordés plus en détail au chapitre II, nécessitent pour apparaître clairement, une étude préalable des applications exploitant un maillage et des types de maillages qu'elles nécessitent. Cette première partie de chapitre sera suivie d'une description des principaux algorithmes de maillage : la construction et l'adaptation. Enfin une troisième partie présentera les modèles géométriques utilisés pour représenter les formes et sur lesquelles s'appuient les méthodes de construction et d'adaptation de maillages.

Nous avons profité de la présentation des différentes méthodes de maillage pour préciser la position de chaque famille d'algorithmes par rapport à ce que serait un mailleur générique. Ceci explique la longueur de ce chapitre qui ne se contente donc pas d'être une simple énumération de méthodes. Nous avons en effet cherché à dégager les parties d'algorithme susceptibles d'être factorisées. Cette analyse n'est cependant que préliminaire : elle propose des pistes sur la manière de généraliser les algorithmes mais le sujet n'est traité de manière approfondie qu'au chapitre IV.

I.1 Utilisations et types de maillage

I.1.1 Définition

Un maillage est *une partition d'un domaine de R^n* . Autrement dit, soit Ω un domaine fermé borné de R^n . M est un maillage si :

1. $\Omega = \bigcup_{K \in M} K$

2. tout élément K de M est d'intérieur non vide
3. l'intersection des intérieurs de deux éléments est vide
4. l'intersection de deux éléments de M est soit l'ensemble vide, soit un élément de dimension strictement inférieure.

Un maillage est donc un ensemble d'*éléments* ou *mailles* dont la dimension dépend du domaine à mailler et de l'espace dans lequel se trouve ce domaine. On parlera dans la suite de maillage :

- *linéique* lorsque le domaine est une courbe ou une droite ; il peut être plongé dans un espace de dimension un, deux ou trois. Les mailles sont des segments.
- *plan* lorsque le domaine et l'espace sont en dimension deux ; les mailles sont des polygones
- *surfactive* lorsque le domaine est en dimension deux et l'espace en dimension trois ; les mailles sont des "polygones" non forcément planaires (par exemple les quadrangles)
- *volumique* lorsque le domaine et l'espace sont en dimension trois : les mailles sont des polyèdres, très souvent des tétraèdres ou des hexaèdres...

Bien que certaines méthodes permettent de construire des maillages de dimension supérieure, elles ne seront pas abordées dans cette étude. Ne seront donc considérés que les maillages évoqués ci-dessus.

I.1.2 Maillages structurés et non structurés

Un maillage peut être vu également comme un graphe dont chaque nœud (ou sommet) est un point appartenant au domaine. Dans ce cas, chaque nœud possède un degré (i.e. le nombre de voisins) et si tous les nœuds n'appartenant pas à la frontière possèdent le même degré, alors le maillage est dit *structuré*. C'est une propriété intéressante pour la simulation de certains phénomènes physiques où les déformations sont importantes, comme l'écoulement de fluides pour lequel un maillage structuré est souhaitable. La construction de maillages structurés est plus difficile, notamment si l'objet contient des détails de petite taille. En conséquence, les méthodes de construction ne couvrent pas tous les types de formes géométriques et restent donc réservées à des formes particulièrement simples. Sinon, il faut avoir recours à des

méthodes de décomposition de domaine (voir plus loin I.2.1.b.2) ou à un découpage manuel de manière à ce que chaque partie puisse être maillée par une méthode de maillage automatique.

Typiquement, les maillages structurés sont formés de quadrangles ou d'hexaèdres, alors que les maillages non structurés regroupent tout type d'élément, souvent des triangles ou des tétraèdres qui sont les formes les plus répandues, mais aussi des quadrangles, des hexaèdres, ou encore un mélange de ceux-ci pour former alors un maillage hybride ([OWE98a]). Les maillages non structurés s'adaptent mieux aux formes complexes mais fournissent en général des résultats de simulation moins précis pour un même nombre de nœuds. Par contre les adaptations de maillage rendant la densité de nœuds hétérogène, comme une adaptation en fonction de la courbure, sont très difficiles à appliquer sinon impossible pour les maillages structurés.

I.1.3 Applications Exploitant un Maillage

Nous présentons ici les trois principales Applications Exploitant un Maillage (AEM) que sont la simulation (méthode des éléments finis et méthode des différences finies), la visualisation et le prototypage rapide. Ces domaines ne sont pas exposés en détails, nous ne présentons que ce qui est nécessaire à la bonne compréhension des besoins de ces applications en terme de maillage.

I.1.3.a Simulation

Les outils mathématiques les plus utilisés aujourd'hui pour simuler un phénomène physique sont la méthode des éléments finis et la méthode des différences finies. Toutes deux résolvent de manière approchée, des problèmes que l'on ne sait pas toujours résoudre de manière exacte. Elles nécessitent la discrétisation à la fois de l'espace dans lequel se trouve la solution cherchée et du domaine soumis aux contraintes, c'est-à-dire la géométrie. La méthode des éléments finis s'appuie sur un maillage triangulaire/tétraédrique non nécessairement structuré, alors que la méthode des différences finies s'appuie typiquement sur un maillage à base de quadrangles/hexaèdres. Le problème se réduit alors à la résolution d'un système d'équations linéaires représenté sous forme matricielle. Une solution approchée est calculée en chaque sommet du maillage, puis sur tout le domaine à l'aide de polynômes d'interpolation associés à chaque élément. Ceux-ci déterminent, avec d'autres facteurs (l'erreur d'approximation du maillage par exemple), l'erreur globale de la solution trouvée par rapport à la solution exacte.

Ces fonctions d'interpolation permettent également de réaliser le calcul sur une représentation simplifiée du solide. En effet, en augmentant le degré des fonctions d'interpolation, il est possible d'assimiler une plaque par exemple à un maillage bidimensionnel au lieu d'un maillage tridimensionnel. Cependant ces polynômes d'interpolation, pour être complètement définis, nécessitent des nœuds supplémentaires (figure I.1-1).



figure I.1-1. Exemple d'ajout de nœuds

La géométrie donnée en entrée est souvent simplifiée avant d'être maillée afin de réduire la durée des calculs. Ceci signifie que les détails de l'objet que l'expert juge négligeables pour le calcul sont tout simplement éliminés ou que, pour des raisons de symétrie, une partie de la pièce seulement est retenue. Remarquons que l'idéalisation de la géométrie de l'objet peut conduire à un objet non valide¹.

Ces méthodes requièrent un certain nombre de propriétés sur le maillage afin de pouvoir être utilisées de manière efficace. Dans la situation la plus simple, il est souhaitable que les éléments (triangles/tétraèdres, quadrangles/hexaèdres) soient les plus proches possibles de l'élément idéal (triangle équilatéral/tétraèdre régulier, carré/cube). C'est le cas dit isotrope, appliqué lorsque le gradient de contraintes est faible. Lorsque le gradient de contraintes est élevé, il est souhaitable que les éléments soient étirés de manière à anticiper par exemple une déformation, c'est le cas anisotrope. Concernant la taille des éléments, il est préférable que les tailles de deux éléments voisins restent proches, de manière à ce que le résultat soit cohérent. Sinon, la précision apportée par un élément de petite taille peut être perdue par un voisin de grande taille. Il est également souvent souhaitable que les éléments près de la frontière soient de bonne qualité.

Pour une meilleure précision des résultats, il est courant que la simulation consiste en plusieurs cycles de calculs-remailage, le maillage étant adapté à chaque itération : une analyse du résultat de l'étape précédente fournit une métrique exprimant l'écart idéal entre deux nœuds du maillage en chaque point du domaine et, dans le cas d'anisotropie, les

¹ Un objet est valide si 1) son adhérence est égale à son intérieur et 2) sa frontière est une variété de dimension $n-1$ avec n , la dimension de l'espace dans lequel se trouve l'objet

vecteurs exprimant les directions d'étirement. Cela permet de déterminer à quels endroits et dans quelle mesure le maillage doit être raffiné, ainsi que l'étirement éventuel des éléments.

La qualité d'un maillage se mesure entre autres par la qualité des éléments. La notion de qualité des mailles dépend de l'utilisation qui est en faite. Il existe donc différentes mesures de qualité. La forme d'un triangle par exemple, est souvent mesurée par le rapport entre le rayon du cercle circonscrit et la longueur de la plus petite arête. Le lecteur trouvera davantage de détails dans [GEO97] et [PEB01].

Pour finir, nous faisons quelques remarques sur le domaine de la simulation. Tout d'abord une simulation réalisée par une méthode de type éléments finis ou différences finies peut prendre beaucoup de temps, de quelques heures à plusieurs jours. Une phase délicate dans ce processus est la construction du maillage. En effet si le maillage est trop grossier, la durée des calculs sera courte, mais le résultat ne sera pas satisfaisant car trop approximatif. Par contre si le maillage est très fin, il y a deux inconvénients : premièrement le fait que le calcul sera peut être inutilement long car certaines zones ne nécessitent pas un maillage fin, deuxièmement, le processus lui-même risque de diverger et le calcul se terminera par une interruption extérieure (limite de temps dépassée par exemple). Il faut donc trouver un bon compromis de manière à ce que le maillage soit fin uniquement là où cela est nécessaire. Actuellement, malgré quelques efforts dans le domaine de la pré-optimisation, la spécification des densités de nœuds avant le premier calcul se fait encore manuellement en précisant par exemple le nombre de nœuds pour une zone donnée (une arête, l'intérieur d'une boîte, etc.) et est donc dirigée par un expert.

Il est important de noter que le processus de maillage doit pouvoir contrôler la taille des triangles, leur forme, leurs proportions et la variation de leur taille. La satisfaction de tous ces critères n'est pas exigée tous les cas.

I.1.3.b Visualisation

En général, les exigences sur les propriétés des maillages ne sont pas aussi fortes en synthèse d'image. Par contre, la taille des maillages à traiter, en terme de nombre d'éléments, est plus importante en général que pour les calculs de simulation. En effet, si les matériels d'affichage et de calcul de rendu sont de plus en plus performants, les scènes comportent de plus en plus d'objets, représentées par plusieurs millions de triangles. De plus la caméra ou les objets de la scène peuvent être amenés à se déplacer. Les deux facteurs rendent d'autant plus difficile les

calculs de rendu et l'affichage dans un intervalle de temps raisonnable. Pour réduire les temps de calculs tout en maintenant la qualité visuelle, plusieurs stratégies sont possibles :

- simplifier le maillage dans les zones planes,
- ne pas construire des éléments dont la taille à l'écran est trop petite (inférieure à celle d'un pixel),
- lorsque le point de vue est amené à changer, créer en pré-traitement, plusieurs maillages selon le niveau de détails souhaité et organiser ces maillages, par exemple de manière hiérarchique. Au moment de l'affichage, choisir le maillage le plus adapté à la position courante de la caméra ([HOP97]).

Comme pour la simulation, les triangles disproportionnés (plats, comportant des angles très aigus ou très obtus) ne sont pas souhaitables car ils sont sources d'erreurs pour certains calculs, comme le calcul des normales aux éléments. Les méthodes d'ombrage par interpolation comme les méthodes de Gouraud ou de Phong ou les méthodes de placage de textures sont sensibles à cela et la qualité de l'image peut s'en ressentir ([FOL95]).

Les éléments triangulaires sont les plus utilisés en visualisation car ils sont forcément plans et convexes, contrairement aux éléments quadrangulaires. De plus, les méthodes de maillage à base de triangles s'adaptent mieux à tout type de géométrie et les matériels d'affichage actuels ont été conçus pour une représentation des objets sous forme de triangles.

Les paramètres dont il serait idéalement utile de tenir compte lors de la construction ou de l'adaptation du maillage pour la visualisation sont le nombre de nœuds, l'erreur d'approximation et la qualité visuelle du résultat (c'est-à-dire l'objet dans la scène avec rendu, texture, radiativité, etc.). Ce dernier facteur n'est évidemment pas pris en compte en pratique de par sa difficulté.

Alors que la structure de données des maillages de simulation est relativement classique, celles des maillages pour la visualisation de scènes animées ou comportant de nombreux objets sont hiérarchiques et de plus, propres à chaque méthode.

I.1.3.c Prototypage rapide

Le prototypage rapide consiste à fabriquer un prototype du produit manufacturier en cours ou en fin de la phase de conception. La construction se fait par empilement de couches. Le

procédé le plus simple découpe chaque couche de l'objet dans des plaques de bois pour les assembler ensuite (stratoconception). D'autres réalisent une polymérisation par laser. La substance polymérisée peut être liquide (stéréo-lithographie) ou poudreuse (frittage de poudre : duraforme, métal, sable). Certains enfin déposent de la matière, comme les imprimantes 3D. Ces différents procédés s'appuient sur un maillage triangulaire pour réaliser le découpage de l'objet en couches et déterminer la forme de chaque couche. Le format de transfert typiquement utilisé entre les logiciels de CAO et ceux des dispositifs de prototypage rapide est le format STL pour STéréoLithographie qui consiste à stocker les sommets, les triangles et les normales aux triangles. Aucune information topologique du modèle CAO initial n'est maintenue. Le découpage en couches est donc réalisé directement sur les triangles afin de simplifier les algorithmes et réduire les difficultés inhérentes aux modèles de plus haut niveau. Les caractéristiques du maillage souhaitées pour cette A.E.M. sont une erreur de corde et un nombre de triangles minimaux. Le nombre de nœuds minimal réduit la durée du calcul de tranchage et facilite le transfert du fichier car sa taille est moindre. Les contraintes sont beaucoup moins fortes que pour un calcul par éléments finis, par exemple. Le dispositif de construction étant toujours limité dans sa précision, l'erreur de corde peut être majorée. Ici aussi, il n'est théoriquement pas souhaitable que les triangles soient disproportionnés, en particulier pour le calcul des normales aux triangles. En pratique, les normales sont calculées à partir du modèle BRep initial et conservées dans le fichier STL. Il n'est donc pas utile de les calculer à partir du modèle STL. Par ailleurs, on tolère d'autant plus les triangles de formes allongées qu'ils réduisent le nombre total de triangles. Il y a toutefois une situation dans laquelle la normale doit tout de même être calculée et dans laquelle, donc, la présence de triangles allongés est pénalisante : c'est le cas de la retro-conception (reprise de conception à partir d'une numérisation). Le maillage STL est alors construit à partir d'un nuage de points et les normales ne peuvent pas être calculées autrement qu'à partir des triangles.

La fabrication rapide d'un prototype nécessite des méthodes de construction et d'adaptation de maillage, à la fois un nombre de nœuds et une erreur d'approximation minimaux, ce qui constituent deux propriétés contradictoires.

Après avoir présenté brièvement les A.E.M. que nous avons retenues pour cette étude, nous allons maintenant voir les méthodes qui permettent d'obtenir des maillages utilisables, soit par construction, soit par adaptation.

I.2 Méthodes de maillage

Cette partie a pour but de présenter les principales techniques de maillage. Elles sont regroupées en deux principales catégories : les techniques de construction de maillage et les méthodes d'adaptation, c'est-à-dire les méthodes permettant d'optimiser une ou plusieurs propriétés d'un maillage ou de modifier le type des mailles, comme la transformation de triangles en quadrangles.

I.2.1 Méthodes de construction de maillages

Il existe différents critères sur lesquels une classification des méthodes de construction peut être faite : le type et la dimension des éléments produits, la dimension de l'espace, la famille d'algorithmes, le type de structure de données utilisé (CSG, Brep classique, Brep hiérarchique en fonction du niveau de détails).

La classification choisie procède d'abord par type d'éléments, puis par famille d'algorithmes. Nous verrons d'abord les méthodes produisant des éléments triangulaires ou tétraédriques, puis les méthodes associées aux autres maillages : quadrilatères, quadrangles, hexaèdres, pentaèdres, maillage mixte 2D ou 3D.

I.2.1.a Maillage par triangles/tétraèdres

Il existe trois grandes familles de méthodes produisant des triangles et des tétraèdres : les méthodes de type Delaunay, les méthodes par avancement de fronts et les méthodes par décomposition spatiale.

I.2.1.a.1 Méthodes utilisant le critère de Delaunay

Le critère de Delaunay est à l'origine de plusieurs méthodes de construction de maillages. Nous rappelons donc d'abord en quoi il consiste avant de présenter les méthodes qui l'exploitent.

Le critère de Delaunay, dit aussi *de la sphère vide*, est satisfait pour un couple de triangles (– resp. de tétraèdres en 3D) si aucun nœud d'un élément (triangle ou tétraèdre) n'est contenu dans l'intérieur du cercle (– resp. la sphère) circonscrit à l'autre élément (figure I.2-1).

Delaunay montra que si ce critère est satisfait pour tout couple d'éléments adjacents, alors il est vérifié pour toute la triangulation ([DEL34]).

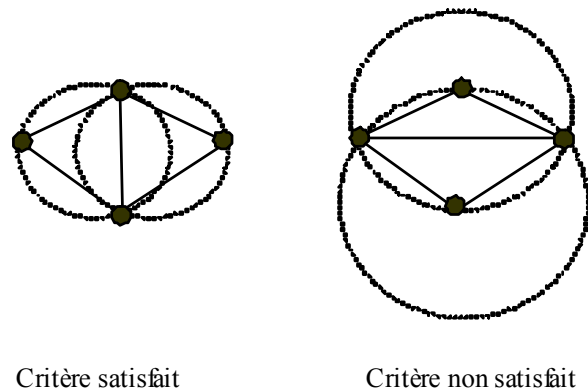


figure I.2-1. Critère de la sphère vide

Une propriété particulièrement intéressante d'une telle triangulation est qu'elle maximise le minimum des angles entre les arêtes. Elle permet ainsi d'avoir des triangles (- resp tétraèdres) les plus proches possibles du triangle équilatéral (-resp tétraèdre régulier).

De nombreuses méthodes utilisent ce critère, mais la technique de base prend en entrée un nuage de points et non une frontière (contour en 2D ou surface fermée en 3D) pour donner un ensemble de triangles ou de tétraèdres dont les sommets sont les points du nuage. D'autres méthodes sont ensuite apparues permettant d'intégrrer dans la triangulation des contraintes, c'est-à-dire des sommets et des arêtes en 2D et éventuellement des triangles en 3D. Le résultat obtenu n'est plus en général strictement une triangulation de Delaunay et est donc appelé *triangulation de Delaunay contrainte*. Ces deux approches sont illustrées dans les paragraphes suivants.

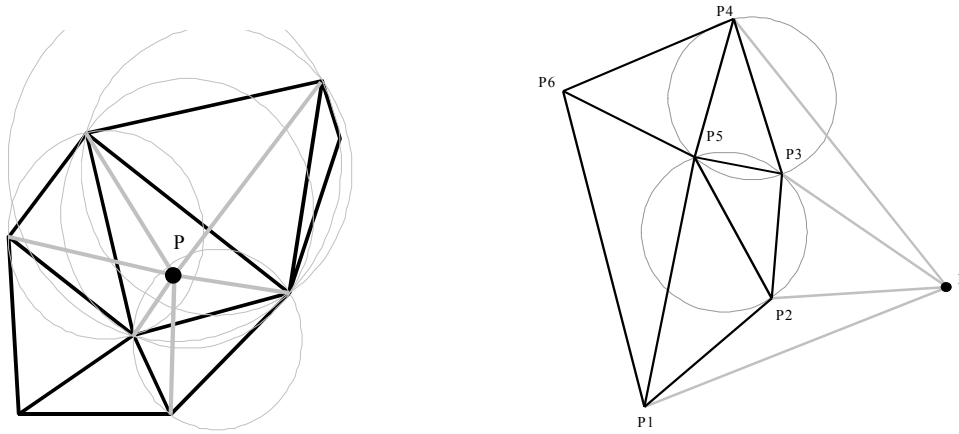
Noyau de Delaunay

Le noyau de Delaunay est une méthode permettant d'insérer un point dans une triangulation en modifiant localement la topologie de la triangulation existante tout en maintenant le critère de Delaunay. C'est pourquoi les méthodes utilisant le noyau de Delaunay sont qualifiées d'incrémentales. Appelons T_i la triangulation construite jusqu'alors et P le point que l'on souhaite y ajouter ; la triangulation obtenue, T_{i+1} , est définie par ([GEO97], [WAT81]):

$$T_{i+1} = T_i - C_P + B_P$$

où C_P est l'ensemble des triangles de T_i dont la sphère circonscrite contient P (figure I.2-2), et B_P est l'ensemble des triangles construits en reliant P aux extrémités de chacune des arêtes visibles depuis P . B_P est appelée la boule. Une arête est visible depuis un point s'il est possible

de relier ce point à chacune des extrémités de l'arête sans couper d'arête. Remarquons que l'ajout de la boule est réalisé après le retrait de la cavité C_P et qu'il est important de respecter cet ordre.



Cas 1 : P est dans la triangulation

Cas 2 : P est à l'extérieur de la triangulation

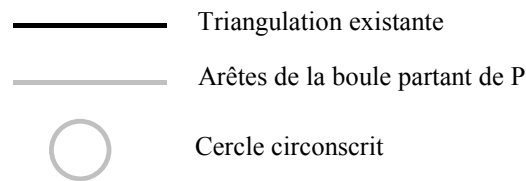


figure I.2-2 . Insertion de P (JGEO97)

Il y a deux possibilités pour le point P selon qu'il se situe à l'intérieur (au sens géométrique) de la triangulation ou non. On peut se ramener au cas où P est à l'intérieur en partant d'une triangulation initiale sans point interne d'un englobant du nuage de points. Dans ce cas, il s'agit de la méthode incrémentale réduite. Le noyau de Delaunay est défini pour les dimensions 2 et 3.

Triangulation de Delaunay contrainte

Le noyau de Delaunay permet d'obtenir la triangulation de Delaunay d'une enveloppe convexe du nuage de points. Or le domaine à mailler est en général limité par une frontière qui est soit un contour en 2D, soit un ensemble fermé de faces en 3D. Donc si les points de cette frontière apparaissent bien dans la triangulation, il n'en est pas de même des arêtes et des faces de cette frontière. Il est alors nécessaire d'insérer dans la triangulation existante, les arêtes et les faces de la frontière. Cela peut être réalisé en calculant l'intersection de l'arête (- resp. la face) à insérer avec les arêtes (- resp. les faces) du maillage. Une précaution peut être préalablement

prise sur les arêtes de la frontière de manière à ce que les cercles ayant ces arêtes pour diamètre, ne contiennent aucun autre sommet que les extrémités de l'arête ([GEO97]). Les arêtes ne vérifiant pas cette propriété sont découpées récursivement jusqu'à satisfaction. Dans tous les cas, rien ne garantit alors que le résultat soit encore une triangulation ou tétraédrisation de Delaunay au sens strict.

Cependant, le cas de la dimension trois pose davantage de problèmes [GEO97]. Tout d'abord, il n'est pas possible de créer une première triangulation sans point interne, comme en 2D. Ensuite, la précaution visant à découper les arêtes dont le plus petit cercle circonscrit ne contient aucune autre entité, ne peut pas s'appliquer en 3D sur les faces. En effet, une solution naïve serait de découper un triangle dont la plus petite sphère circonscrite (PPSC) contient une autre entité géométrique (sommet, arête ou face) par insertion d'un ou plusieurs nœuds. Mais cela risque de produire des triangles dont les PPSC dépassent la PPSC de la face avant le découpage. Enfin une autre difficulté est qu'il ne suffit pas de faire apparaître les arêtes d'une face dans le maillage pour que cette face y apparaisse également.

Construction des points

L'utilisateur d'une AEM souhaite en général obtenir le maillage d'une face plane ou gauche, ou encore d'un volume. Or jusqu'à présent, nous avons expliqué comment trianguler un nuage de points situés dans un domaine fermé, mais pas comment construire ces points.

Il y a deux manières de créer les points internes. Soit un nuage de points est construit avant la triangulation, soit les points sont construits au fur et à mesure que la triangulation évolue.

Dans le premier cas, une solution simple consiste à remplir l'espace de points en appliquant une grille dont les dimensions sont en accord avec les besoins en terme de répartition de sommets. Une autre solution est d'utiliser une décomposition spatiale par arbre quaternaire ou octal où la densité de nœuds est connue en chaque feuille de l'arbre. Mais ces deux manières de faire ne constituent pas un placement des points particulièrement pertinent, notamment pour les éléments finis car la répartition des points ne tient pas compte de la frontière du domaine. En effet, des points peuvent apparaître très près de la frontière et produire des triangles aplatis peu souhaitables.

[SHI93] propose de répartir les points en assimilant chaque point au centre d'une cellule. La taille des cellules est définie par l'écart souhaité entre les points pour toute position dans le

domaine à mailler. Si un vide permet d'accueillir une nouvelle cellule, elle est ajoutée. Ceci est réalisé jusqu'à recouvrir tout le domaine à mailler. Les cellules disposent d'un degré de liberté selon qu'elles se situent sur un sommet de la frontière, sur une arête ou à l'intérieur d'une face. Elles interagissent ensuite par des forces d'attraction/répulsion issues de la physique. Les déplacements de cellules qui en résultent font tendre le système vers un état d'équilibre de ces forces. Le nuage de points constitués des centres des cellules est ensuite triangulé par le noyau de Delaunay. L'état d'équilibre parfait correspond à une configuration des centres des cellules en hexagones. Un hexagone pouvant être triangulé par des triangles équilatéraux, ce type de méthode produit des résultats très intéressants autant pour les éléments finis que pour la visualisation. En effet, le placement des points peut également exploiter l'erreur de corde ou la taille des triangles à l'écran. Le temps de calcul nécessaire pour trouver l'état d'équilibre reste cependant important par rapport à d'autres méthodes.

Dans le second cas, la méthode cherche à construire un point à chaque pas du processus de triangulation. Selon la méthode utilisée, les points sont insérés sur les arêtes ([RIV96], [RIV01]) ou de manière à former un élément idéal en fonction de la configuration locale ([FRE94]) ou encore au centre des cercles circonscrits des éléments jugés mauvais ([RUP95], [SHW02]). En 2D dans le cas classique (isotrope non contrôlé), le triangle idéal peut être un triangle équilatéral, mais il peut avoir une autre forme par exemple lorsqu'il approche une surface gauche très perturbée dont la triangulation est réalisée dans l'espace paramétrique. Le nouveau point peut également être calculé en fonction de la configuration locale ([CHE89]).

Maillage contrôlé

Parfois l'AEM ou la représentation géométrique imposent de contrôler certaines propriétés du maillage comme la forme des éléments ou leur taille. Typiquement, c'est le cas lors de la triangulation d'une surface paramétrique perturbée. Le maillage est souvent construit dans l'espace paramétrique où les algorithmes géométriques sont plus robustes puisque l'espace est bidimensionnel. Mais les triangles obtenus n'ont plus la forme et la taille souhaitées à cause de la déformation des triangles due au passage de l'espace paramétrique vers l'espace réel. Il y a alors deux manières de procéder. La première est de modifier les triangles dans l'espace paramétrique de manière à ce que les éléments formés soient satisfaisants ([CHO98]). Ceci n'est pas toujours suffisant notamment lorsque la surface est très perturbée, c'est-à-dire lorsque les dérivées varient fortement. La seconde solution est de redéfinir la notion de distance dans l'espace paramétrique et de construire les éléments avec cette nouvelle

définition, de manière à ce qu'ils aient la taille, les proportions et les directions souhaitées. La notion de distance est alors redéfinie par une métrique ([GEO97]) qui est déterminée soit par un calcul par éléments finis précédent, soit à partir de la représentation de la surface (rayons de courbures principaux, rayon de courbure minimal), soit par une combinaison de plusieurs métriques. Le fait de changer la distance implique aussi d'adapter le critère de la sphère vide de Delaunay, qui devient le critère de l'ellipse vide (ou ellipsoïde en 3D). [CHE97] proposent également d'utiliser le critère de l'ellipse vide, l'ellipse n'est cependant pas définie par une métrique mais est l'approximation de la projection du cercle de l'espace réel dans l'espace paramétrique.

La synthèse qui suit résume les points importants des méthodes de Delaunay en rapport avec le sujet de ce mémoire :

Les méthodes de construction de maillage utilisant le critère de Delaunay produisent des maillages de bonne qualité. Elles reposent le plus souvent sur des bases mathématiques garantissant une valeur minimale pour les angles internes des éléments. Les problèmes de la dimension 3 restent des problèmes ouverts pour lesquels seules des méthodes empiriques existent ([GEO97]). Cependant, dans les deux cas (dimension 2 et 3), la démarche générale est la même :

- construction d'un maillage initial (sans point interne en 2D, et avec un ajout de quelques points en 3D). L'algorithme n'est pas le même dans les deux cas.
- création de nœuds : soit tous les nœuds au préalable, soit un à un au vol. Les méthodes sont en général similaires en 2D et en 3D (construction sur les arêtes, au centre des cercles circonscrits des éléments jugés mauvais, placement de points par systèmes à particules etc.)
- insertion de nœuds un à un avec la technique du noyau de Delaunay. Elle s'appuie sur le calcul de la boule et de la cavité, dont les implémentations changent avec la dimension.

Au regard de la conception d'un mailleur générique, nous constatons que certaines méthodes de Delaunay offrent un cadre permettant de regrouper les méthodes de construction d'un maillage à base de triangles (maillage plan ou surfacique) ou de tétraèdres. Par exemple en imposant le fait que la création de points se fasse à la volée, alors l'algorithme général

(algorithme I.2-1), ainsi que le noyau de Delaunay s'écrivent de la même façon dans les cas plan, surfacique et volumique. Seules les méthodes internes (calculs de la cavité, de la boule, du cercle circonscrit à un triangle) nécessitent un développement spécifique.

Algorithme Delaunay

Créer un maillage initial à partir de la frontière

Tant que le critère de satisfaction n'est pas vérifié **Faire**

Créer un point

Insérer ce point par le noyau de Delaunay

Ftant

Fin

algorithme I.2-1. Triangulation de Delaunay

I.2.1.a.2 Méthodes par avancement de fronts

Une autre catégorie de méthodes de maillages triangulaires procède par avancement de fronts. Le principe ([LOH88]) consiste à partir d'un front initial qui correspond à la frontière de la zone à mailler. Il s'agit donc d'un ensemble d'arêtes pour un maillage de surface et d'un maillage triangulaire fermé pour un maillage volumique. Un segment (- resp une face en 3D) y est choisi afin de former le prochain élément à ajouter au maillage. On calcule ensuite le point qui permettrait de construire un élément idéal à partir du segment (- resp de la face) choisi(e). Dans le cas le plus simple, cet élément idéal est un triangle équilatéral (- resp tétraèdre régulier), mais sa forme peut varier selon les besoins (maillage isotrope contrôlé ou anisotrope). Si des sommets du front appartiennent au voisinage de ce point, alors ils sont utilisés pour former des éléments candidats qui sont ensuite comparés entre eux au regard de certains critères. Le meilleur élément qui ne coupe pas le front est choisi et est ajouté au maillage. S'il n'y en a pas ou si aucun ne convient, l'élément idéal est alors ajouté au maillage. Si ce dernier ne convient pas, une méthode doit être appliquée pour forcer la création d'un élément "idéal", en réduisant sa taille par exemple en fonction du nombre de tentatives.

Pour les mêmes raisons que précédemment, le maillage peut être contrôlé, c'est-à-dire que les triangles souhaités sont contraints par la taille, la proportion de leurs dimensions et les directions d'étirement. Lorsqu'il s'agit de faire un maillage surfacique, il y a deux manières de

procéder : soit les triangles sont créés directement dans l'espace réel ([LAU96]), soit ils sont créés dans l'espace paramétrique ([CUI98], [TRI98]), en redéfinissant la notion de distance.

```

Algorithme AvancementFrontsTRITET // pour TRIangles/TETraèdres
    Construire les fronts
    Tant qu'il existe un front non vide Faire
        Choisir un élément EF du front (une arête en 2D, une face en 3D)
        Calculer le point P qui permettrait de créer un élément idéal à partir de EF
        Scruter le voisinage de P dans le but de réutiliser un sommet du front
        Si cela n'est pas possible Alors
            Si l'élément idéal ne coupe pas le front Alors
                L'ajouter à la triangulation
            Sinon
                Appliquer forcer la construction d'un nouvel élément
            Fsi
        Sinon
            Construire un élément avec le point déjà existant
        Fsi
        Mettre à jour les fronts
    Ftant
Fin
    
```

algorithme I.2-2. Avancement de fronts

Le reproche fait aux méthodes par avancement de fronts en général est le manque d'une base mathématique telle celle qui existe pour les méthodes utilisant le critère de Delaunay. De plus, particulièrement en 3D, il est difficile d'assurer la convergence et les calculs d'intersection peuvent devenir importants et sujets à des erreurs numériques. Cependant, les résultats de ces méthodes n'en sont pas moins intéressants, notamment en 3D, puisque d'après [OWE98a], environ 23 sur 64 codes de maillage triangulaire et tétraédrique pour éléments finis, pris dans des logiciels commerciaux, libres ou universitaires sont basés sur une méthode frontale.

Il existe par ailleurs des méthodes combinant les deux approches, Delaunay et avancement de fronts. Le principe est de créer les points internes comme dans une méthode par avancement

de fronts puis d'insérer ce point en faisant appel au critère de Delaunay. Ces méthodes allient alors les avantages des deux approches ([FRE94]).

Tout comme les méthodes de Delaunay, le squelette de l'algorithme reste le même en 2D et en 3D (algorithme I.2-2).

I.2.1.a.3 Méthodes par décomposition spatiale

Les premiers travaux basés sur une décomposition de l'espace sont issus du domaine de la visualisation d'objets représentés sous leur forme implicite. L'espace est découpé récursivement en cubes jusqu'à obtenir une taille limite donnée. Chaque cube fait l'objet d'un calcul d'intersection avec la surface. Les positions des sommets des cubes sont calculées relativement au solide (à l'intérieur ou à l'extérieur). 256 configurations d'intersection entre un cube et la surface du solide sont alors possibles, réductibles à 15 en tenant compte des symétries. A partir des points d'intersection des arêtes du cube avec la surface et de la configuration parmi les 256, un polygone est construit approchant localement la surface. Les polygones sont triangulés et assemblés formant ainsi un maillage fermé de la surface de l'objet. Pour réduire le coût du calcul d'appartenance d'un point au solide, une solution simple est de réaliser un suivi à partir d'un premier cube coupant la surface: c'est l'algorithme du suivi de cubes (ou *marching cube*, [LOR87]). Cette méthode peut également être utilisée pour construire un maillage tétraédrique d'un solide en découpant l'espace non pas avec des cubes, mais avec des tétraèdres, ce qui diminue le nombre de configurations possibles. La version surfacique de la méthode est adaptée éventuellement à la visualisation mais pas à un calcul par éléments finis car les triangles possèdent en général des angles trop aigus, c'est-à-dire que les éléments sont trop d'éléments aplatis. En ce qui concerne le maillage volumique, le découpage de l'espace en tétraèdres permet d'obtenir un maillage adéquat pour un calcul par éléments finis ([SCH96a], [SCH96b]) en fournissant un maillage très régulier vers l'intérieur du domaine. Par contre, les éléments de mauvaise qualité se retrouvent près de la frontière ce qui constitue un inconvénient pour les applications de simulation.

L'avantage majeur de cet algorithme est sa simplicité. Ce type de méthode est utilisé lorsque la représentation du solide est implicite, car, dans le cas d'une représentation par les frontières, il reste coûteux en temps. Un autre défaut important de cette approche est le non respect de la topologie du solide. En effet, si la taille des éléments de volume (cubes ou tétraèdres) est trop grande, le processus de découpage peut créer des trous dans le maillage. La figure I.2-3 montre en 2D une telle situation où le carré en gras est ignoré car ses quatre sommets sont

situés à l'extérieur du domaine. Le même phénomène existe en 3D. En remplaçant les cubes par des tétraèdres, le nombre de configurations ambiguës diminue, mais les tests d'intersection sont plus nombreux. Enfin le maillage final dépend de l'orientation du solide par rapport au découpage de l'espace et les éléments de plus mauvaise qualité se situent près de la frontière ce qui n'est pas souhaitable, surtout pour un calcul de simulation.

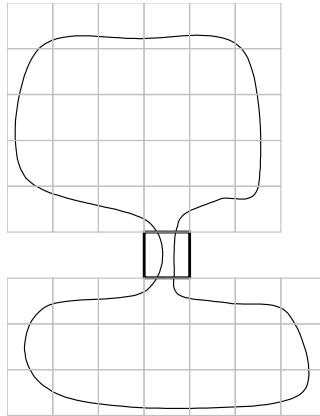


figure I.2-3 . Décomposition spatiale en 2D

Cette méthode de maillage (algorithme I.2-3) est sans doute la plus générique dans le sens où elle permet d'obtenir un maillage à partir de n'importe quel modèle géométrique. En effet le seul service qu'elle nécessite de la part du modéleur géométrique est le test d'appartenance d'un point au domaine (2D ou 3D) qui est réalisable autant pour une représentation implicite que pour une représentation par les frontières (voir I.3), même si, dans ce dernier cas, les performances et la qualité du résultat sont loin de ce qu'offrent les méthodes par avancement de fronts ou de Delaunay. En effet, le contrôle de la taille, de la forme et de l'orientation des éléments est ici difficile.

Algorithme DécompositionSpatiale

Subdiviser l'espace en cellules (carrés, triangles, cubes, hexaèdres)

Déterminer une cellule C coupant la frontière

Tant qu'il existe une cellule coupant la frontière **Faire**

 Réaliser un suivi à partir de C

 Rechercher une cellule coupant la frontière

Ftant

Fin

Algorithme Suivi

Calculer l'intersection de la cellule avec la frontière

Triangler le polygone résultat

// et dans le cas d'un maillage volumique :

Remailler la cellule coupée par la frontière

Tant qu'il existe des voisins non visités **Faire**

 Appel récursif de Suivi sur les voisins non visités

Ftant

Fin

algorithme I.2-3. Décomposition spatiale

Ceci termine la partie sur la construction de maillages triangulaires/tétraédriques. Les méthodes de Delaunay et par avancement de fronts sont les plus utilisées lorsqu'elles peuvent s'appliquer. Elles sont en effet très souples au niveau du contrôle du maillage, et très générales quant à la classe d'objets auxquels elles s'appliquent. Les méthodes par décomposition spatiale sont en général utilisées en visualisation lorsque les contraintes sur le maillage sont moins fortes, ne nécessitant pas un contrôle important, ou lorsque la représentation des objets est implicite (voir plus loin I.3.2).

Nous avons vu dans les trois familles, maillage de Delaunay, par avancement de fronts et décomposition spatiale, qu'il existe dans la structure des algorithmes des parties communes permettant d'esquisser quelques algorithmes génériques. Cependant certains facteurs vont à l'encontre de l'approche générique, comme la difficulté d'adapter un algorithme de triangulation 2D à un algorithme de maillage tétraédrique. En effet, ceci nécessite le

développement de certains algorithmes spécifiques à la dimension des mailles. De plus chaque famille possède des caractéristiques propres : tandis que certaines méthodes de Delaunay et d'avancement de fronts permettent de contrôler l'étirement, la taille et les proportions des triangles, la méthode du suivi de cubes est adaptée à la représentation implicite des solides.

Le paragraphe suivant se penche sur la construction de maillages quadrangulaires et hexaédriques. La forme des mailles a une influence sur la manière de construire les maillages. Par conséquent, nous ne retrouvons pas exactement les mêmes familles d'algorithmes.

I.2.1.b Maillage par quadrangles/hexaèdres

Les maillages quadrangulaires ou hexaédriques sont essentiellement utilisés pour les calculs de simulation, en particulier pour les écoulements de fluides ou autres phénomènes créant de forts gradients de contraintes.

La construction de ces maillages reste une tâche délicate, notamment dans le cas de maillages structurés. Ceci est dû au fait que le maillage de la frontière contraint davantage la topologie du maillage à l'intérieur du domaine. Par exemple, un maillage quadrangulaire est contraint par l'arité du nombre de sommets sur le contour. En effet, si le nombre de points de la frontière est impair, il est impossible de construire le maillage sans insérer de nœud sur la frontière. Ceci a pour conséquence, en 3D, de rendre difficile le maillage d'une pyramide à base carrée, c'est le problème de Schneider (figure I.2-4).

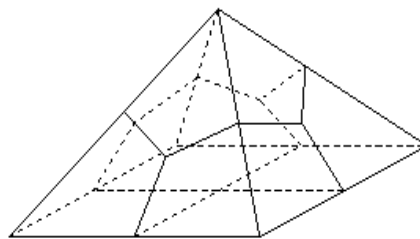


figure I.2-4. Le problème de Schneider ([SCH96a])

Certaines méthodes ne peuvent s'appliquer qu'à une classe restreinte d'objets, telle [STA98] qui ne s'applique qu'aux objets qui peuvent être obtenus par extrusion le long d'une courbe et dont la section ne change pas de topologie.

Nous allons voir maintenant les différentes méthodes de construction de maillages quadrangulaires et hexaédriques, qui sont regroupées en deux catégories : les méthodes par avancement de front, par décomposition en domaine.

I.2.1.b.1 Maillage par avancement de fronts

Le principe est de construire les éléments soit un à un à partir d'un maillage de la frontière, soit par niveaux. Dans le cas de la construction par niveaux pour un maillage structuré, les arêtes se faisant face doivent contenir le même nombre de nœuds, et en 3D, les faces en vis à vis doivent avoir un maillage surfacique de même topologie. Les arêtes ou les faces en vis à vis sont reliées par extrusion en formant ainsi des éléments quadrangulaires ou hexaédriques (selon la dimension). Dans [STA98], une face source et une face destination doivent être spécifiées (figure I.2-5). Après avoir maillé ces deux faces avec la même topologie de maillage, les nœuds internes sont créés par couches et leur position est obtenu par interpolation entre les nœuds de la face source et ceux de la face destination. Les faces source et destination ne sont pas obligatoirement planes et parallèles, la méthode tolère les torsions et les surfaces gauches peu perturbées. Cependant, les méthodes se basant sur ce principe nécessitent la plupart du temps un découpage préalable du domaine de manière à pouvoir appliquer l'algorithme.

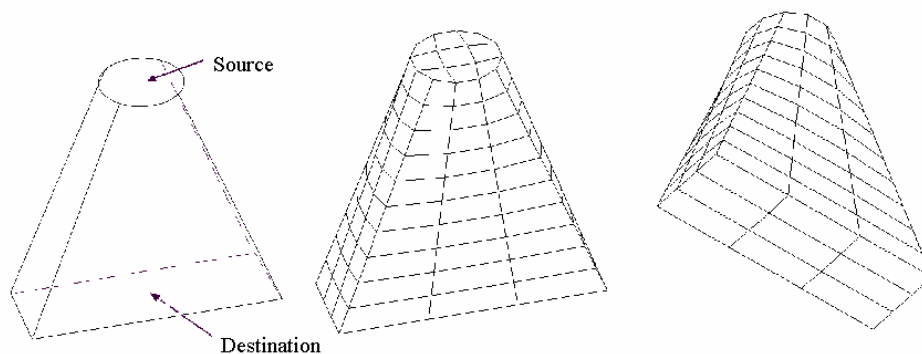


figure I.2-5. Maillage hexaédrique par niveaux ([STA98])

Une autre approche consiste à construire les éléments le long de la frontière, un à un ([WHI97]) ou par rangée ([BLA91]). Les méthodes de mise à jour des fronts (division et fusion de fronts) sont alors similaires à celles des méthodes par avancement de fronts pour les maillages triangulaires et tétraédriques. Un post traitement est appliqué pour améliorer la qualité des éléments en fusionnant les couples d'arêtes faisant un angle trop petit. Cet

l'algorithme dit de dallage (ou "paving" [BLA91]) a été repris en 3D pour donner l'algorithme de "plastering" ([BLA93]) où des hexaèdres sont placés d'abord le long de la frontière, puis le front ainsi formé avance vers l'intérieur du volume (figure I.2-6). Bien que la méthode tente d'adapter la construction des nouveaux éléments à l'espace restant, elle peut produire des cavités qu'il peut être difficile de mailler avec des hexaèdres.

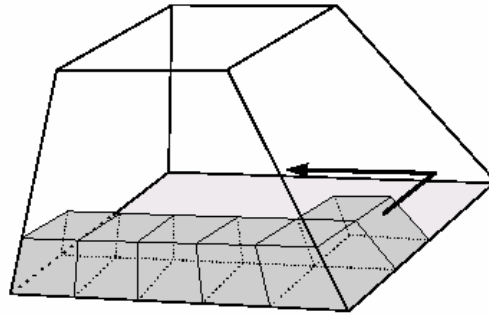


figure I.2-6. La méthode de dallage en 3D ([BLA93])

Parce que le maillage de la surface contraint fortement la topologie du maillage volumique, les auteurs de l'algorithme de "plastering" ont cherché à déterminer la topologie du maillage volumique avant sa géométrie ([TAU96]). Cet algorithme, appelé "whisker weaving" (tissage de brins) prend en entrée un maillage quadrangulaire de la surface de l'objet. Des "feuilles" sont calculées à partir d'un maillage surfacique quadrangulaire. Une feuille est obtenue à partir d'un suivi effectué sur le maillage surfacique dans une direction. L'ensemble de ces feuilles permet de déterminer la topologie du maillage volumique interne à partir de la topologie du maillage surfacique. Les hexaèdres sont construits un à un. Chaque intersection de trois feuilles est le centre d'un hexaèdre (figure I.2-7).

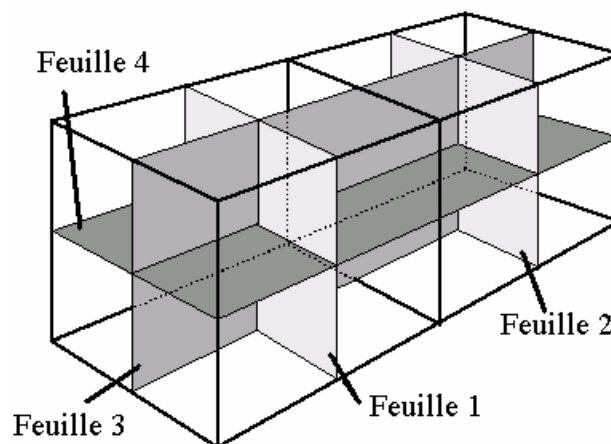


figure I.2-7. algorithme du tissage de brins ([TAU96])

Ces méthodes permettent d'obtenir des maillages dont les éléments de frontières sont de bonne qualité, telles que le requièrent certaines simulations, ce qui n'est souvent pas le cas des méthodes par décomposition spatiale du paragraphe suivant.

Dans cette partie, il n'apparaît pas de squelette d'algorithme de manière aussi évidente que précédemment, notamment pour généraliser la méthode de tissage de brins et les autres méthodes. En effet la différence se situe dans la manière d'aborder le problème puisque la topologie du maillage est déterminée avant sa géométrie. Quant aux algorithmes de dallage, les méthodes par ajout d'éléments un à un semblent pouvoir s'écrire sous la même forme que l'algorithme d'avancement de fronts à base de triangles (algorithme I.2-3). Les grandes étapes de la méthode d'avancement de fronts y apparaissent : construction d'un élément idéal, tests avec le voisinage de l'élément idéal, et mise à jour des fronts. Par contre, les méthodes de dallage créant les éléments par rangée nécessitent une modification mineure par rapport à la version triangulaire/tétraédrique de l'avancement de fronts (algorithme I.2-4).

Algorithme AvancementFront

Construire les fronts

Tant qu'il existe un front non vide **Faire**

 Choisir EF un élément du front (une arête en 2D, une face en 3D)

 Construire un élément idéal (ou plusieurs) à partir de EF

 Vérifier l'intégration de cet élément idéal dans le maillage existant en utilisant le voisinage

Si cela n'est pas possible **Alors**

 Appliquer une méthode de forçage de construction d'un nouvel élément

Fsi

 Mettre à jour les fronts

Ftant

algorithme I.2-4. Avancement de fronts généralisé

I.2.1.b.2 Maillage par décomposition en domaines

Il y a deux types de décomposition :

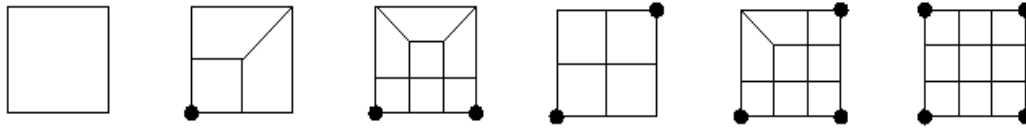
- ❑ soit l'espace est rempli de quadrangles en 2D ou d'hexaèdres en 3D, puis une méthode est appliquée pour que la frontière du maillage soit fidèle à la frontière du domaine à mailler,
- ❑ soit le domaine est décomposé de manière à ce que les sous-domaines obtenus puissent être plus facilement maillés.

Décomposition de l'espace

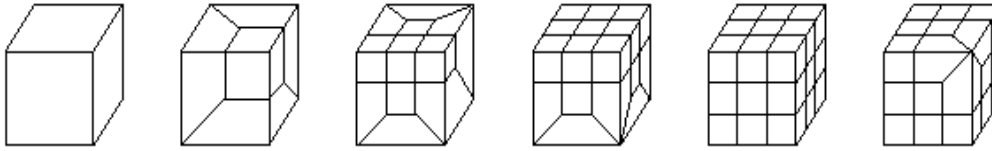
Dans le premier cas, une structure d'arbre quaternaire (en 2D) ou octal (en 3D) est souvent utilisée, de manière à découper récursivement un englobant (quadrilatère ou hexaèdre) jusqu'à atteindre une taille d'élément donnée. La correspondance avec la surface peut être réalisée de plusieurs manières : en calculant l'intersection des éléments coupés par la frontière, en projetant les sommets de la frontière du maillage sur la frontière du domaine ([SCH96a]), ou en contractant des arêtes de manière à réduire le volume de la partie extérieure du maillage ([WAL02]).

Mis à part [WAL02], ces approches permettent d'obtenir des maillages structurés lorsque la géométrie du domaine s'y prête. Dans le cas contraire, des efforts doivent être faits pour découper le domaine. Ce découpage peut parfois se faire automatiquement comme nous allons le voir ci-dessous ou à la main dans les pires cas.

Dans le cas où il n'est pas obligatoire que le maillage soit structuré, la taille des éléments voisins est souvent contrôlée en limitant à un la différence de niveau dans l'arbre entre deux éléments voisins dans l'espace. Si le niveau de deux éléments voisins n'est pas le même, alors des motifs d'éléments prédéfinis sont appliqués pour passer d'un niveau à un autre ([SCH96a]) (figure I.2-8).



Motifs d'éléments en 2D



Motifs d'éléments en 3D

figure I.2-8. Motifs d'éléments prédéfinis ([SCH96a])

L'intérêt de cette approche est qu'un maillage structuré peut être obtenu rapidement contrairement aux méthodes par avancement de fronts qui nécessitent de nombreux tests d'intersection. Par contre, les éléments les plus déformés se situent le long de la frontière alors que les éléments de bonne qualité sont situés plus à l'intérieur. Ceci est un inconvénient important de ce type d'approche. De plus, le plus souvent, la qualité du maillage obtenu dépend de l'orientation de la pièce et la détermination d'une orientation pertinente n'est pas triviale (figure I.2-9).

Un algorithme général se dégage très naturellement (algorithme I.2-5). La subdivision de l'espace peut s'appuyer sur une structure d'arbre ou non. De même, la seconde étape peut se réaliser de différentes manières selon que les éléments coupés sont supprimés puis le vide ainsi formé remaillé, ou qu'un suivi est réalisé comme dans l'algorithme I.2-3. Bien qu'il soit succinct, il demeure pertinent puisqu'il représente une manière de procéder encapsulé dans un sous-programme paramétré

Algorithme DécompositionSpatiale
 Subdiviser l'espace en cellules
 Traiter les éléments de frontières
Fin

algorithme I.2-5. Décomposition spatiale généralisée

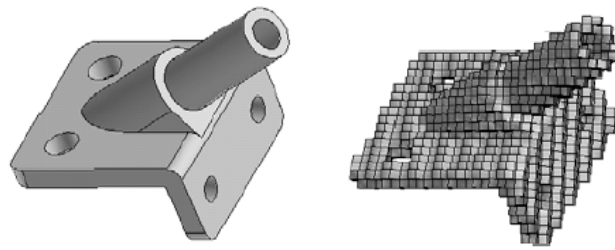


figure I.2-9. Exemples de décomposition de l'espace ([BLA00])

Décomposition en domaines

L'approche consiste à reconnaître les solides qui peuvent être maillés avec des hexaèdres. [BLA96] procède en reconnaissant des formes en "tonneau" à partir des faces puis les interfaces entre les tonneaux, pour les mailler indépendamment par des hexaèdres. La reconnaissance des solides qui peuvent être maillés par cette méthode appelée l'outil du tonnelier ("cooper tool"), se fait par une analyse des maillages des faces et détermine un axe d'extrusion s'il en existe. Dans ce cas, le processus de construction d'hexaèdres crée les éléments par extrusion des faces sources en suivant les faces latérales.

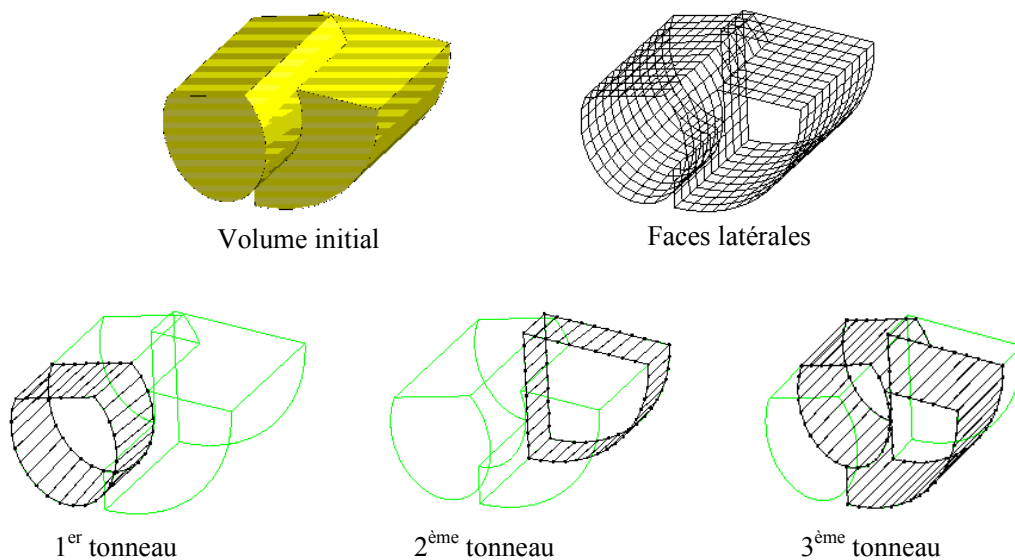


figure I.2-10. Décomposition d'un volume en tonneaux ([BLA96])

Cette méthode a été étendue dans [MIY00] pour reconnaître les tonneaux selon plusieurs axes (figure I.2-11). Dans cet article, un arbre est construit avant le maillage pour détecter et conserver les connexions entre tonneaux. Deux tonneaux sont connectés dans l'arbre si et

seulement s'ils le sont géométriquement. Les maillages volumiques de deux tonneaux connexes sont réalisés d'abord. Ensuite le raccord entre ces tonneaux est effectué en supprimant des hexaèdres dans le maillage volumique du parent dans l'arbre, créant un trou qui peut, après certaines précautions, être maillé à l'aide de la méthode de [BLA96] (l'outil du tonnelier).

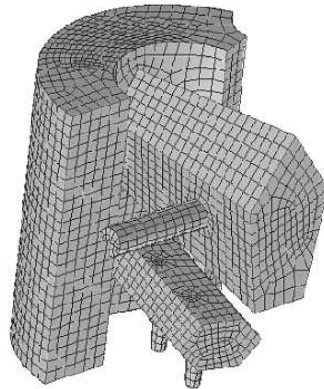


figure I.2-11. Maillage par extrusion selon différents axes ([MIY00])

Une autre méthode pour réaliser le découpage de l'intérieur du domaine utilise la notion d'axe médian en 2D et de surface médiane en 3D, appelés squelette ou objet médian (figure I.2-12). Le MAT ("Medial Axis Transform") consiste à construire un squelette ([TAM91], [PRI95], [SAM01]) qui est défini en 2D (- resp. 3D) comme étant le centre des cercles (- resp. sphères) de rayon maximum intérieurs au domaine. À tout point du squelette est également associée la plus petite distance à la surface de l'objet, ce qui, théoriquement au moins, rend réversible le MAT.

L'objet médian est utilisé pour découper l'intérieur du domaine en longeant les frontières de manière à rendre plus aisée la construction de maillage par quadrangles ou hexaèdres. L'axe médian n'atteignant pas toujours la frontière, il faut le prolonger de manière à former des sous-domaines fermés ([ARM95]). La construction du maillage se fait par sous-domaine à l'aide de l'une des méthodes précédentes.

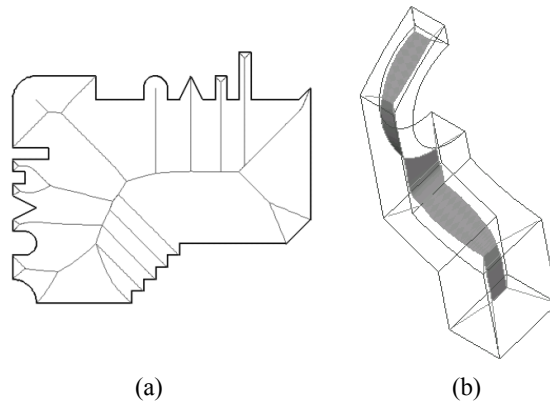


figure I.2-12. Objets médian en 2D (a) et en 3D (b) ([ARM95])

En général, le squelette est obtenu à partir d'un maillage de la frontière ([SHB95]), en exploitant par exemple un maillage triangulaire (ou un maillage tétraédrique en 3D) de Delaunay du domaine avec une répartition homogène des points ([SAM01]) ou son dual, le diagramme de Voronoï ([SHF98]), ces deux derniers fournissant une représentation approchée du squelette.

Par exemple dans [SAM01], une approximation de la surface médiane est calculée à partir d'un maillage tétraédrique de l'objet. Sur cette approximation est construit un maillage hybride à majorité de quadrangles. Enfin ce maillage hybride est extrudé vers la frontière du domaine pour chaque sous-domaine, perpendiculairement à la frontière, donnant un maillage hybride à majorité d'hexaèdres. Des opérations de fusions sont ensuite appliquées sur les paires d'éléments situés de part et d'autre de la surface médiane afin d'augmenter le nombre d'hexaèdres.

Les méthodes de maillage par décomposition du domaine sont insensibles à l'orientation de la pièce et les éléments le long de la frontière sont de bonne qualité. Par contre, le découpage en sous-domaines reste une tâche difficile qui est encore parfois réalisée à la main. La décomposition en tonneaux ne s'applique pas à tout type d'objet mais uniquement à ceux dont il est possible de reconnaître des parties pouvant être obtenues par extrusion. Par ailleurs, la construction de l'objet médian est délicate et seule une représentation approchée est souvent déterminée, même si elle s'applique à une classe d'objets plus importante.

La généralisation de ces méthodes n'est pas triviale car elles sont de natures diverses et les structures des algorithmes sous-jacents sont très différentes. En particulier, d'après Ted Blacker, auteur de la méthode de l'outil du tonnelier, le test vérifiant si le solide peut être maillé, la détection des faces sources et latérales et le maillage des faces sont des tâches

enchevêtrées. Par contre ce n'est pas le cas des méthodes basées sur l'objet médian qui peuvent se généraliser (algorithme I.2-6). Comme dans le paragraphe précédent, la partie commune à ces algorithmes est très restreinte, mais indique bien la démarche du traitement, bien que les différentes parties soient indépendantes et interchangeables.

Algorithme Décomposition Domaines MAT

Calcul de l'objet médian

Maillage de l'objet médian

Maillage séparé des sous-domaines

Fin

algorithme I.2-6. Décomposition de domaines

I.2.1.c Première synthèse

L'examen des méthodes de construction de maillage a permis de mettre en évidence une certaine part de généricité. A notre connaissance, il n'existe pas, à l'heure actuelle, d'algorithme suffisamment général pour modéliser toutes les méthodes produisant des éléments triangulaires, quadrangulaires, tétraédriques et hexaédriques. Mais il est possible d'identifier quelques catégories prépondérantes et de dégager un squelette commun aux méthodes d'une catégorie donnée (voir les algorithmes 1,2,3,4,5 et 6). Les étapes de ces squelettes nécessitent une implantation spécifique.

La partie suivante s'attache à nouveau à rechercher de la généricité, cette fois dans les méthodes d'*adaptation* de maillage. Ces dernières ont pour vocation de modifier un maillage existant pour qu'il satisfasse mieux un certain nombre de contraintes.

I.2.2 Adaptation de maillages

Il arrive qu'on ne sache pas construire un maillage avec les propriétés optimales ou qu'on souhaite réutiliser un maillage dans un autre contexte que celui prévu au départ. Dans ces cas, une *adaptation* est appliquée au maillage existant de manière à améliorer ses propriétés.

Il peut s'agir d'affiner le maillage là où une concentration de nœuds donnée est nécessaire, de le simplifier, par exemple pour l'adapter au point de vue, ou de modifier la position des nœuds pour améliorer la forme des éléments. Ce sont les trois types d'adaptation étudiés dans ce chapitre.

Nous n'élargissons pas davantage le spectre de notre étude car le but premier de ce mémoire est bien la conception d'un *mailleur* générique. Néanmoins, les enseignements tirés de l'étude des méthodes d'adaptation permettront d'envisager leur intégration dans le mailleur générique et des propositions en ce sens seront faites dans les chapitres suivants.

I.2.2.a Raffinement et transformation des éléments

Le raffinement de maillage n'est autre que l'ajout de nœuds dans le maillage de manière à rendre le traitement sur le maillage plus précis. L'ajout de nœuds est contrôlé par les besoins de l'A.E.M. L'ajout de nœuds peut aussi viser à transformer des triangles (-resp tétraèdres) en quadrangles (-resp. hexaèdres). On parle alors de transformation des éléments. Cette transformation peut également être réalisée par fusions des triangles/tétraèdres. Enfin un dernier type de transformation des éléments est l'augmentation du degré des éléments (au sens éléments finis et différences finies), c'est-à-dire l'ajout de sommets sur les arêtes des éléments (sans ajouter d'arêtes internes aux éléments).

I.2.2.a.1 Raffinement

Pour les maillages triangulaires/tétraédriques, les méthodes de raffinement les plus connues et les plus utilisées dans le domaine de la simulation exploitent le noyau de Delaunay (voir I.2.1.a.1). En effet la méthode incrémentale réduite de Delaunay, qui part d'une triangulation triviale englobant le domaine à mailler, n'est autre qu'une méthode de raffinement. La position des points créés dépend du traitement auquel va servir le maillage. Elle est, par exemple, dictée par la taille des triangles à l'écran si le maillage sert de base à un rendu réaliste ([SHI93]) ou par l'erreur de corde si l'application est le prototypage rapide ([SHI93]).

En ce qui concerne le raffinement de maillages quadrangulaires/hexaédriques et mixtes, le raffinement se fait par remplacement d'un élément par plusieurs autres en modifiant éventuellement les éléments avoisinants. Si les éléments utilisés sont uniquement des triangles et/ou des quadrangles, alors le raffinement d'un élément a pour conséquence le raffinement des éléments voisins (en utilisant les patrons décrits dans [SCH96a] et [SCH96b]). En 3D, la limitation de cette propagation se fait en introduisant d'autres éléments (pentagones/pentaèdres, hexagones, prismes, etc.). [YAM01] propose un certain nombre de motifs permettant de répondre au problème de Schneiders vu en I.2.1.b, mais le nombre d'éléments alors produits est très important, voire trop. Remarquons que si le maillage est

structuré, le raffinement d'un élément nécessite la propagation du raffinement sur tout le maillage ou presque.

I.2.2.a.2 Transformation des éléments

La transformation des éléments peut se faire de deux manières. Soit un raffinement en découpant chaque triangle en trois quadrangles et les quadrangles en quatre autres, soit en fusionnant les triangles par paires. La première solution aboutit à un maillage non structuré en général, composé entièrement de quadrangles. Les éléments sont de très mauvaise qualité en général et le maillage nécessite un lissage avant de pouvoir être utilisé. De plus le nombre d'éléments peut être plus important que voulu puisqu'il s'agit d'un raffinement. La seconde solution aboutit à un maillage à majorité de quadrangles. Les éléments sont souvent de mauvaise qualité également mais le nombre d'éléments est plus faible ([LAU97]). Certaines méthodes s'attachent à éviter ces inconvénients, comme [OWE98b] qui modifie localement le maillage triangulaire pour construire les quadrangles. Cette méthode possède l'avantage des méthodes par avancement de fronts sans leurs inconvénients. Elle évite notamment les calculs d'intersection en exploitant la topologie du maillage triangulaire existant. De son côté, [SHI98] dispose préalablement les points à l'aide d'un système à particules. Pour tenir compte du fait que le maillage final doit être composé de quadrangles étirés ou non (pour un maillage anisotrope), les cellules sont respectivement carrées ou rectangulaires ([VIS00]) et les forces d'interaction entre les cellules tiennent compte de la forme des cellules. Une fois un état d'équilibre atteint, une triangulation de Delaunay est effectuée sur le nuage de points composé des centres des cellules. Puis les triangles sont fusionnés 2 à 2 en cherchant à respecter la direction donnée localement par les cellules.

Le troisième type de transformation consiste à ajouter des nœuds sur les arêtes d'un triangle sans ajouter d'arêtes internes à l'élément. Cette méthode est appelée p-méthode. La méthode des éléments finis associe à chaque élément une fonction d'interpolation permettant de calculer les contraintes à l'intérieur de l'élément à partir des valeurs calculées aux nœuds. Prenons un triangle, possédant donc trois nœuds ; la fonction d'interpolation associée ne peut être que de degré 1. Pour une meilleure précision, les fonctions d'interpolation peuvent être de degré plus élevé. Des nœuds supplémentaires sont alors nécessaires. Ces nœuds sont ajoutés sur les arêtes (figure I.2-13). Mais cela ne se fait pas sans poser parfois des problèmes importants, notamment lorsque l'erreur d'approximation est importante par rapport à la

courbure ([GEO97]). Dans ce dernier cas, la forme globale de l'élément n'est plus proche du triangle et l'interpolation n'est plus significative (figure I.2-14).



figure I.2-13. Exemple d'application de la p-méthode (cas simple)

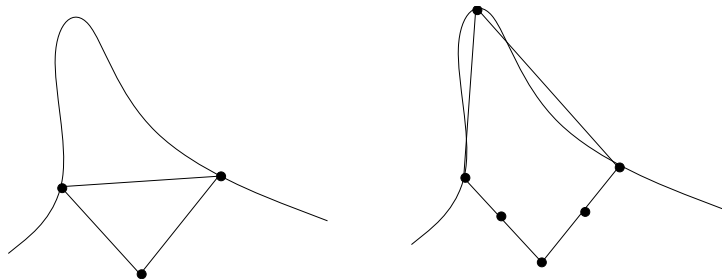


figure I.2-14. Exemple de difficulté de la p-méthode

En fait toutes ces méthodes de raffinement et de transformation des éléments semblent relativement différentes dans leur manière de procéder, même pour un même traitement (raffinement ou transformation d'éléments). Notons cependant qu'il s'agit le plus souvent d'une opération locale répétée successivement jusqu'à atteindre un état acceptable. L'opération locale ainsi que le test d'arrêt du processus dépendent alors de la méthode employée. Parmi ces méthodes, seule celle de K. Shimada ne respecte pas ce processus mais l'utilise. Le principe des systèmes à particules qu'il propose relève plus de l'optimisation de la position des sommets avant transformation des éléments que de la transformation elle-même et doit donc être dissociée des autres méthodes.

I.2.2.b Déplacements de noeuds

La construction et le raffinement d'un maillage nécessitent la plupart du temps un post traitement car les algorithmes produisent malgré tout des défauts que les méthodes d'optimisation réduisent considérablement en peu de temps. Ces défauts proviennent le plus souvent de la difficulté à construire de nouveaux points de manière à satisfaire au mieux des propriétés parfois contradictoires sur le maillage. Pour améliorer le maillage, un déplacement

des nœuds est alors appliqué de manière à rendre plus régulières la répartition des nœuds et, par suite, la forme des éléments.

La méthode la plus simple pour rendre la forme des éléments plus régulière, est de chercher à placer chaque sommet au barycentre de ses voisins. Chaque sommet est traité une fois, puis le processus recommence jusqu'à ce que le déplacement de chaque sommet soit inférieur à un seuil donné ou qu'un nombre d'itérations fixé à l'avance soit atteint. Le plus souvent, les points ne sont déplacés que d'une fraction de la distance entre la position actuelle et le barycentre. Cette méthode dite du Laplacien, possède des inconvénients, dont celui de produire des éléments de mauvaise qualité voire des éléments inversés dans les régions concaves. Différentes variantes existent dont les principales consistent à :

- pondérer la contribution de chaque voisin en fonction de la longueur des arêtes ou de l'aire des éléments ([BLA91])
- contraindre le mouvement des nœuds de manière à éviter les éléments inversés ([CAN97])
- adapter la méthode du Laplacien aux maillages anisotropes en prenant en compte la définition particulière de la notion de distance ([GEO97])

Ces méthodes ont l'avantage de fournir rapidement un résultat de qualité correcte. Cependant elles ne considèrent pas le problème dans son ensemble et ne fournissent pas un résultat optimal. Les méthodes d'optimisation, quant à elles, considèrent une fonction d'énergie. Plus le maillage est de bonne qualité, plus l'énergie associée est faible. Le but est de minimiser l'énergie globale du système. La qualité peut être calculée à l'aide de la qualité des éléments ([CAN98]), ou à l'aide des angles internes aux éléments partageant un même sommet ([ZHO00]). D'autres, comme K. Shimada et al. ([SHI00], [VIS00]), se basent sur la physique (voir I.2.1.a.1). Techniquement cela revient à résoudre un système d'équations non linéaire, c'est pourquoi les temps de calculs sont plus longs qu'avec les méthodes de type Laplacien. En contrepartie elles fournissent de meilleurs résultats.

En général, les méthodes d'optimisation ou celles basées sur la physique sont complémentaires des méthodes dérivées du Laplacien. Lorsqu'une amélioration de type Laplacien ne fournit pas les résultats escomptés, une méthode de minimisation d'énergie est appliquée. Certaines méthodes combinent les deux ([CAN98])

On ne peut proposer de squelette commun à ces deux familles d'algorithmes car l'une est itérative (Laplacien) et l'autre fait appel à la résolution d'un système d'équations (optimisation). Seules les fonctions de plus bas niveau telles la mesure de qualité des éléments semblent pouvoir être factorisées.

I.2.2.c Simplification de maillage

Les travaux de simplification de maillage concernent majoritairement le domaine de la visualisation, où les scènes peuvent comporter de nombreux objets plus ou moins complexes. Pour afficher rapidement une scène et permettre de s'y déplacer, le maillage est simplifié par réduction du nombre de triangles. Il existe différentes manières de simplifier un maillage. La plupart dégradent la précision d'approximation du maillage afin de réduire significativement le nombre de triangles. Voici en quelques lignes les principales méthodes, présentées dans [ERI96], [CIG98] et [LUE01] selon le même découpage :

- La première a pour principe de supprimer des entités géométriques, typiquement des sommets ([SCH92]) ou des triangles ([TAU98]) et de remailler les trous ainsi formés. Différentes techniques, utilisant en général une métrique, permettent de choisir quelles entités supprimer et dans quel ordre. La technique la plus simple se base sur l'erreur d'approximation : les entités géométriques dont la suppression engendre le moins d'erreur sont supprimées en premier. D'autres travaux ([COE96]) s'aident d'une réduction et d'un agrandissement du solide (offsets). Les suppressions de sommets sont considérées valides si elles maintiennent le maillage simplifié entre les deux objets décalés. Cela permet de s'assurer que la forme du maillage simplifié n'est pas trop éloignée de l'original.
- Un autre type de méthode appelé re-tiling (remaillage) consiste à positionner aléatoirement des points sur la surface et à les déplacer à l'aide de forces d'attraction/répulsion en les contraignant à rester sur la surface ([TUR92]). Ces points sont ensuite intégrés au maillage triangulaire puis les nœuds du maillage initial sont supprimés un à un et les trous sont re-triangulés. Ceci débouche sur un modèle simplifié.
- L'optimisation d'une fonction d'énergie, proposée par [HOP93], [HOP96] consiste à affecter une fonction d'énergie à trois caractéristiques du maillage : le nombre de nœuds, l'erreur d'approximation, et la régularité du maillage (déterminée à partir de la longueur des arêtes). La méthode cherche à minimiser l'énergie en appliquant des opérations choisies aléatoirement parmi un ensemble d'opérations valides. Ces opérations sont la

fusion de deux sommets, la division d'arête, et le retournement d'arête (identique à celui utilisé dans les méthodes de Delaunay). Elles visent à réduire le nombre de sommets, à réduire l'erreur d'approximation et à rendre la forme des éléments plus régulière.

- Une autre approche est d'agglomérer des sommets géométriquement proches pour les remplacer par un représentant. Les méthodes basées sur ce principe se distinguent par la manière de choisir les sommets à agglomérer. Il y a essentiellement deux manières de faire. La première utilise une grille tridimensionnelle, éventuellement adaptative, pour remplacer tous les sommets appartenant à une cellule de la grille par un seul sommet ([ROS93]). La seconde approche utilise une métrique pour déterminer les noeuds à fusionner ([GAR97]). Ces méthodes ne préservent pas toujours le genre de l'objet en particulier si la fusion de sommets peut se faire sur deux sommets n'appartenant pas à la même arête.
- Les ondelettes ([ECK95]) offrent un cadre mathématique permettant d'avoir, dans une seule structure, plusieurs représentations plus ou moins détaillées d'un même objet. L'algorithme procède en trois phases. Parmi les sommets du maillage initial, un petit nombre d'entre eux est extrait, de manière à former un ensemble réparti relativement uniformément sur la surface. Une triangulation de Delaunay contrainte est construite à partir de ces sommets uniquement, ce qui partitionne le maillage initial de l'objet en régions triangulaires. Chaque région triangulaire est associée à un ensemble de (petits) triangles du maillage initial. En deuxième étape, chaque région triangulaire fait l'objet d'une paramétrisation locale permettant de passer de la région triangulaire à la partie du maillage initial qui lui est associée. La paramétrisation est réalisée de manière à ce que l'ensemble des représentations paramétriques soit continu. La troisième phase consiste à découper les triangles du maillage grossier récursivement jusqu'à ce que la distance entre le maillage courant et le maillage initial (le maillage le plus fin) soit inférieure à une valeur donnée. Le continuum qui existe entre le maillage le plus grossier et le maillage le plus fin permet de passer d'une représentation grossière à une représentation détaillée de manière continue. Remarquons cependant que le maillage initial ne peut pas être atteint strictement : il ne peut qu'être approché à une tolérance près.

Ces méthodes s'intègrent dans un processus plus global visant à maintenir plusieurs représentations d'un même objet avec différentes précisions. En effet, la visualisation en temps réel de scènes complexes selon différents points de vue nécessite en général le calcul de

différents maillages pour un même objet. Puis au moment de l'affichage, une représentation adéquate est choisie ou calculée de manière à réduire le nombre de triangles à afficher. Il y a essentiellement trois types de processus pour gérer plusieurs représentations d'un objet ([LUE01]) : l'approche statique, l'approche dynamique, et celle dépendant du point de vue. L'approche statique consiste à maintenir pour chaque objet, plusieurs maillages. La construction des maillages se fait lors d'un pré-traitement. Puis, au moment de l'affichage, la représentation adéquate est choisie pour chaque objet. L'avantage est que le calcul des maillages est réalisé une fois pour toute et ne nécessite donc pas de calculs pendant l'affichage. De plus, la méthode est adaptée aux matériels d'affichage. Les inconvénients sont que le passage d'un niveau de détails à un autre pour un même objet peut être visible et que cette représentation occupe beaucoup de mémoire.

L'approche dynamique consiste à représenter les opérations de simplification permettant de passer d'un niveau de détail à un autre, plutôt que les informations géométriques. Il y a deux possibilités selon que la hiérarchie d'opérations est construite de manière ascendante ou descendante. La construction ascendante raffine un maillage grossier au départ en déterminant la géométrie des nouveaux sommets par des règles mathématiques (surfaces de subdivision). La construction descendante correspond aux méthodes de simplification décrites ci-dessus. Un maillage initial fin est simplifié, chaque opération de simplification étant stocké dans une hiérarchie. Par rapport à l'approche statique, celle-ci permet une granularité plus fine car la représentation est plus concise. Le passage d'un niveau de détail à un autre est donc moins flagrant. Pour améliorer encore la qualité visuelle et ne pas percevoir le raffinement à l'affichage, une interpolation entre les maillages peut être réalisée.

Enfin, l'approche dépendant du point de vue étend la simplification dynamique de manière à ce que le niveau de détails ne soit pas le même pour tout l'objet, mais dépende de la position de la caméra par rapport à l'objet. Ceci est tout à fait adapté par exemple au déplacement de la caméra au-dessus d'un terrain formé d'un seul objet ([HOP97]). La partie détaillée du terrain est celle qui est proche du point de vue alors que la partie proche de l'horizon est très grossière (figure I.2-15).

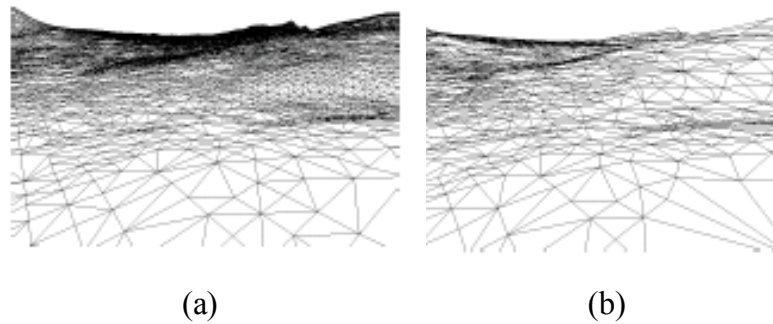


figure I.2-15. Maillage non adapté (a) et adapté (b) au point de vue

Parmi les cinq familles de méthode de simplification, trois obéissent au même schéma global : les méthodes supprimant des entités, celles remplaçant des entités, et les méthodes d'optimisation d'énergie. Elles ont en commun le fait d'itérer sur tout le maillage pour déterminer les lieux à traiter, de choisir une ou plusieurs opérations à appliquer et de réaliser un test de convergence en fonction de paramètres propres à chaque méthode (algorithme I.2-7). En fait ce schéma est exactement le même que pour les méthodes de déplacement de nœuds, à la différence près que les opérations locales, le choix des opérations à appliquer et le test de convergence sont différents. Cependant les autres méthodes, c'est-à-dire les méthodes de re-tiling (remaillage de tout le solide) et celles utilisant les ondelettes respectent chacune un schéma qui leur est propre et ne s'intègre pas dans le processus de l'algorithme I.2-7 .

Par ailleurs, certaines opérations locales apparaissent dans plusieurs méthodes :

- la suppression d'entités géométriques (sommets, triangles), puis remaillage des trous engendrés
- le remplacement d'entités géométriques par un seul sommet (remplacement d'un groupe de sommets, contraction d'arête, etc.). Il existe plusieurs stratégies pour positionner le nouveau sommet.

Par conséquent si toutes les méthodes ne s'intègrent pas dans un algorithme générique unique, certaines fonctions peuvent néanmoins être factorisées, au moins théoriquement.

<p>Algorithme Simplification_Itérative</p> <p style="padding-left: 40px;">Tant que non convergence Faire</p> <p style="padding-left: 80px;">Choisir les lieux à traiter</p> <p style="padding-left: 80px;">Choisir l'opération locale à</p> <p style="padding-left: 40px;">appliquer</p> <p style="padding-left: 80px;">Appliquer l'opération locale</p> <p style="padding-left: 80px;">Tester la convergence</p> <p style="padding-left: 40px;">FTant</p> <p>Fin</p>
--

algorithme I.2-7. Simplification itérative de maillages

I.2.2.d Seconde synthèse

En fait, en observant toutes les méthodes itératives, nous pouvons constater que la plupart d'entre elles peuvent s'écrire sous la forme d'un même algorithme générique qui est celui de l'algorithme I.2-7. Ces méthodes sont la construction de maillage par raffinement de type Delaunay, les méthodes par avancement de fronts (triangulaires, quadrangulaires, tétraédriques, hexaédriques) et les méthodes de simplification itératives. En effet, le Tableau I.2-1 montre comment l'algorithme I.2-7 généralise ces méthodes et à quoi correspond chaque étape pour les différentes familles de méthodes. Les méthodes qui n'obéissent pas à ce schéma sont celles qui ne pouvaient pas s'écrire sous la forme des algorithmes génériques décrits précédemment.

Algorithme général	Delaunay	Avancement de fronts	Simplification itérative	Déplacements de nœuds
Initialisation	Maillage frontalier	Initialisation des fronts	Rien	Rien
Tant que non convergence				
Choisir les lieux à traiter	Stratégie de création d'un point	Choix d'un élément d'un front	Choix de l'entité géométrique à traiter	Choix du nœuds à traiter
Choisir l'opération locale	Pas de choix	Choix de la manière de créer le nouvel élément. i.e. en s'aidant du voisinage ou non	Propre à chaque méthode : décimation de sommet, de triangles, compression d'arête, etc.	Simple déplacement, déplacement avec retournement d'arête, etc.
Appliquer l'opération	Noyau de Delaunay	Mise à jour des fronts	Appliquer l'opération choisie	Appliquer l'opération choisie
Tester la convergence	Dépend du processus de création de points	Les fronts sont vides	Critère de satisfaction propre	Critère de satisfaction propre

Tableau I.2-1. Correspondance entre l'algorithme générique et les familles d'algorithmes

I.3 Modèles géométriques en CAO

Il semble nécessaire de faire quelques rappels sur ce que nous entendons par modèle géométrique. Cette partie nous permettra également de nous faire une idée des services que peut proposer tel ou tel modèle qui pourraient être nécessaire à la construction ou à l'adaptation d'un maillage.

Le terme modèle géométrique est défini comme étant la manière de représenter la forme des objets. Nous exposons ici différentes façons de définir les courbes, les surfaces et les solides, parmi les méthodes les plus connues en CAO.

La figure suivante propose une classification des différentes représentations existantes. Nous voyons dans l'ordre les formes paramétriques qui sont très utilisées aujourd'hui en CAO et les fonctions implicites qui apparaissent davantage dans le domaine de l'image de synthèse et de

l'imagerie médicale. De manière générale, les formes paramétriques s'écrivent $x = g_1(t)$, $y = g_2(t)$ pour une courbe de paramètre t , et $x = f_1(u, v)$, $y = f_2(u, v)$, $z = f_3(u, v)$ pour une surface de paramètres u et v . Les courbes implicites sont caractérisées par une fonction de type $f(x, y) = 0$ en 2D et les surface implicites par $f(x, y, z) = 0$. Chacun de ces types comporte des sous-classes que nous allons présenter dans les paragraphes qui suivent.

Par ailleurs, ces modèles sont utilisés en combinaison avec des structures de données qui sont la représentation par les frontières (Brep : Boundary Representation) et l'historique de construction (CSG : Constructive Solid Geometry), très présents dans tous les domaines.

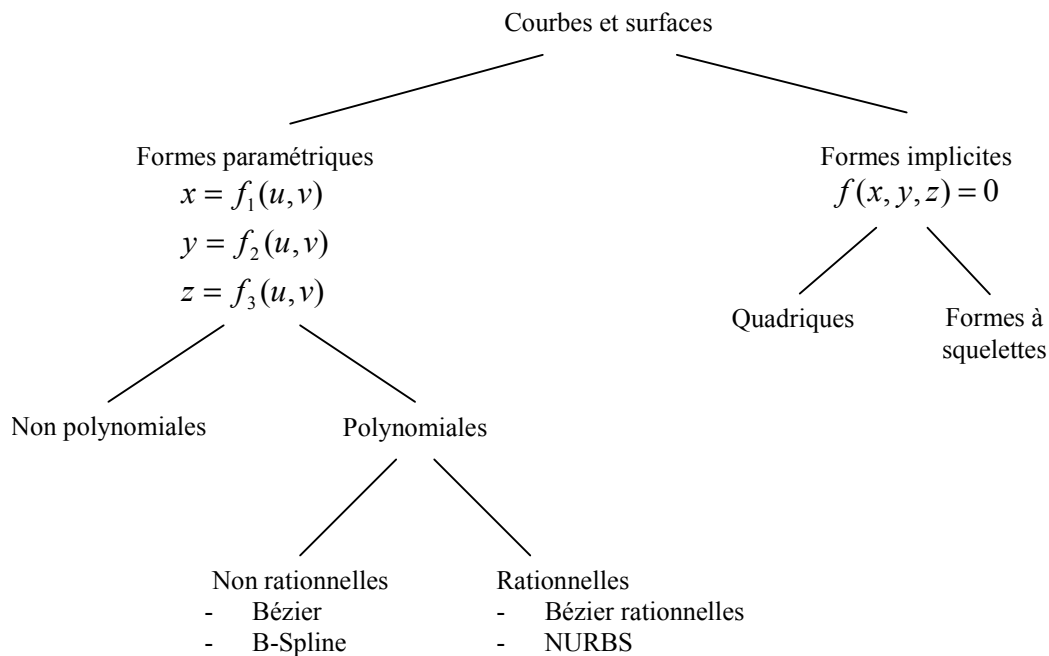


figure I.3-1. Les modèles géométriques de courbes et surfaces

De manière générale, la modélisation géométrique est soumise à des contraintes comme la validité des objets que l'on peut réaliser avec chaque modèle, ou la puissance, c'est-à-dire la capacité à représenter un objet réel le plus fidèlement possible. De plus, il est souhaitable de pouvoir effectuer certains traitements sur ces solides comme la visualisation, le calcul du volume, l'appartenance d'un point à un solide, le calcul de sections, etc.

I.3.1 Courbes et surfaces paramétriques

Les modèles de courbes et surfaces gauches les plus utilisés en CAO, reposent sur les formes paramétriques polynomiales. L'idée sous-jacente est de définir une courbe (resp. surface) par

déformation d'un segment (resp. un rectangle). Les courbes (resp. surfaces) sont définies à l'aide de polynômes et sont contrôlées par un ensemble de points de contrôle n'appartenant pas toujours à la courbe (resp. la surface). Les points de contrôle permettent, comme leur nom l'indique, d'avoir un contrôle sur la courbe (resp. la surface), de pouvoir la modifier en déplaçant un ou plusieurs de ces points. Ce qui différencie ces modèles entre eux sont les polynômes, qui donnent un caractère global (comme les Bézier) ou local (comme les B-spline, les NURBS) au contrôle.

Par ailleurs, il est utile de pouvoir appliquer le plus facilement possible certaines opérations à ces courbes et ces surfaces comme le raccordement, l'intersection de deux surfaces, le maillage ou des transformations géométriques. Le raccordement, par exemple, peut assurer une continuité géométrique (vecteurs tangents de même sens) ou paramétrique (continuité de la dérivée, i.e. du vecteur tangent, y compris sa norme). Certaines transformations peuvent être appliquées facilement sur toutes les formes paramétriques en ne transformant que les points de contrôle. Ceci n'est cependant vrai que pour les opérations de translation, rotation, et homothétie, pas pour la projection perspective.

Par contre, l'opération de calcul d'intersection des surfaces paramétriques pose d'importants problèmes. En effet, il n'est pas possible de représenter le résultat d'une intersection avec la même représentation. Ces surfaces ne sont autre que des champs de courbes appelés isoparamétriques, chacune étant déterminée en fixant un des deux paramètres. Malheureusement, la (ou les) courbes d'intersection n'appartiennent pas (sauf exceptions) à ces champs de courbes. Ceci fait que, sauf cas particuliers, l'intersection de deux surfaces est approchée ([BER95]).

Les surfaces paramétriques, lorsqu'elles ne posent pas de problèmes à cause de leur intersection, sont très utiles pour la construction de maillage surfacique. Elles permettent la construction du maillage de ces surfaces en 2D dans l'espace paramétrique, sans avoir besoin de recourir à des algorithmes tridimensionnels souvent moins robustes. De plus les méthodes les plus avancées pour construire les maillages les plus contraints au niveau de la forme des éléments, c'est-à-dire les maillages anisotropiques, utilisent l'espace paramétrique pour construire le maillage en 2D. Cette approche semble difficile à transposer aux autres modèles de surfaces ou de volumes. Cependant, des précautions doivent être prises pour que les éventuelles propriétés requises sur le maillage soient vérifiées dans l'espace réel. En effet le passage de l'espace paramétrique à l'espace réel modifie la géométrie du maillage et peut avoir

pour conséquence de produire des triangles mal formés (étirés) voire plats dans l'espace réel bien que ceux-ci aient été bien formés dans l'espace paramétrique. Le passage d'un espace à l'autre ne conserve, en général, ni les distances, ni les angles (voir I.2.1.a.1).

Notons que le fait de construire le maillage dans l'espace paramétrique a pour conséquence de ne pas modifier le genre de l'objet, c'est-à-dire le nombre de trous débouchant, lorsque celui-ci est constitué d'un ensemble de surfaces. Et ceci reste vrai même si les mailles sont de grande taille (sous réserve que la taille d'une maille reste inférieure à la taille d'une surface)

Le maillage d'une surface paramétrique nécessite un certain nombre de services, que ce soit pour assurer la qualité en forme des triangles (maillage pour la simulation) ou pour contrôler la distance d'un élément à la surface (pour la visualisation ou le prototypage rapide). En effet, sous l'hypothèse que la surface soit une nappe régulière¹ de classe C^2 , il est possible d'établir des métriques permettant de contrôler l'erreur de corde (métrique du plan tangent, des rayons de courbures principaux, du rayon de courbure minimal). Ainsi, ces métriques sont exploitées par certaines méthodes de maillage pour déterminer la position des nœuds et la forme des éléments. De plus ces métriques peuvent être combinées avec d'autres métriques provenant du résultat d'un premier calcul par éléments finis, ou d'une pré-optimisation ([CUI98]), ou obtenues par interpolation des écarts entre les nœuds de la frontière ([FRA98]). D'autres méthodes de maillage ne font qu'exploiter la distance la plus courte entre deux sommets ou les dérivées afin d'obtenir par exemple les vecteurs tangents, les rayons de courbure ou l'erreur de corde et de contrôler le maillage. Les surfaces paramétriques les plus utilisées (Bézier, B-Splines, NURBS) sont polynomiales donc de classe C^∞ . Ces services sont donc réalisables.

I.3.2 Fonctions implicites

Les fonctions implicites, plus utilisées en réalité virtuelle et en animations qu'en CAO, offrent un moyen attrayant de modéliser des objets qui peuvent par ailleurs être solides ou non ([DES92]). Un objet est représenté par une fonction $F(P)$, décrivant pour chaque point P , sa position relativement à la surface. Plus précisément, soit iso une constante réelle, si $F(P)=iso$, le point est sur la surface, si $F(P)>iso$, P est à l'extérieur de l'objet (selon la convention adoptée) et $F(P)<iso$ quand P est à l'intérieur. C'est donc autant un modèle de solides que de surfaces.

¹ Régulière signifiant que la normale à la surface ne s'annule jamais

La fonction F peut être une fonction algébrique, ce qui rend les traitements sur le solide difficiles à maintenir, notamment à cause du degré parfois élevé des équations. Pour pallier cela sont apparues les formes à squelettes ([BLO90], [BLO00]). Le squelette est vu comme un ensemble de sources de potentiel auxquelles est associée une fonction exprimant la valeur de ce potentiel en fonction de la position. Ce potentiel exprime en général la distance d'un point à la source la plus proche ou une combinaison des sources les plus proches. Les sources et les fonctions potentielles peuvent être de natures différentes. Les squelettes les plus simples sont formés de sources ponctuelles nombreuses. C'est le cas des "blobs" (fonction exponentielle, [BLI82]), des "méta-balls" (fonction polynomiale à étendue limitée, [JIN00]) et des "soft-objects" (fonction polynomiale en la distance au carré, [WIT86]).

Mais la fonction F peut également être définie de manière procédurale ([BLO90]), ce qui permet d'exprimer des conditionnelles ou autres opérations difficiles à exprimer de manière analytique. Dans cet article ([BLO90]), les squelettes peuvent être plus complexes et sont composés de points, de splines et de contours composés de splines. A chaque élément est associée une procédure calculant la valeur du potentiel.

De manière générale, les opérations booléennes sont réalisées plus aisément qu'avec la représentation paramétrique des surfaces car il suffit d'appliquer une opération de mélange des potentiels. [PAS95] définit de manière analytique des fonctions de mélange permettant de réaliser l'union et l'intersection à l'aide de fonctions simples. Une attention particulière doit cependant être accordée au mélange des squelettes afin que deux squelettes ne devant pas fusionner restent bien distincts. L'exemple souvent cité est celui de la main où les doigts doivent être mélangés à la paume au niveau de l'articulation, mais pas au-delà. Ceci est réalisé en maintenant, pour chaque élément de squelette, l'ensemble des éléments de squelette avec lesquels il peut fusionner. Il est également possible d'effectuer d'autres opérations, comme une torsion autour d'un axe – en composant la fonction potentiel avec une fonction déformation ([ROS97]) – ou le calcul d'isosurfaces. Une isosurface est une surface obtenue par modification de la fonction rayon associée, en retranchant ou en ajoutant une constante, sans affecter la forme du squelette. Cela aboutit à une réduction ou une augmentation du volume répartie uniformément sur toute la surface de l'objet.

Cette représentation étant implicite, la visualisation des objets nécessite soit un lancer de rayons, soit une discrétisation (maillage). Pour mailler un tel objet, il y a essentiellement deux types de méthodes : soit une méthode par décomposition spatiale tel le marching cube

([LOR87]), soit une triangulation de points répartis préalablement sur la surface avec des méthodes basées par exemple sur un arbre octal ou sur des systèmes à particules ([WIT94]).

En outre, l'objet médian ("Medial Axis") est également une représentation implicite de l'objet et le MAT ("Medial Axis Transform") est un moyen de déduire une représentation implicite, le plus souvent approchée, à partir d'une représentation par la frontière. Cette représentation n'est cependant pas utilisée pour la modélisation mais pour des traitements divers : découpage du domaine, idéalisation de la géométrie, calcul d'offsets, ... La fonction F n'est pas disponible en général.

Les fonctions implicites permettent une modélisation géométrique intéressante dans le sens où la manipulation des squelettes est intuitive et permet de réaliser relativement aisément des combinaisons et des déformations de l'objet, qui peuvent même changer son genre. De plus ce modèle est tout à fait adapté à la représentation du vivant, difficile à représenter à l'aide de surfaces paramétriques. C'est pourquoi ces modèles sont utilisés entre autres pour les objets fortement déformables dans les domaines de la visualisation réaliste et de l'imagerie médicale.

Une difficulté de maillage liée à la représentation implicite des objets est qu'il est difficile de déterminer la densité du maillage de manière à ce que le genre de l'objet ne change pas. Ceci provient du fait que la surface de l'objet n'est pas connue explicitement, comme le nom l'indique. Ce problème n'apparaît pas avec les surfaces paramétriques où le genre peut être maintenu en réalisant le maillage dans l'espace paramétrique. Cependant des travaux ont été faits pour minimiser les risques de changement de topologie avec notamment l'utilisation d'un système à particules pour la répartition des points sur la surface avant le maillage ([WIT94]). De plus, un autre inconvénient est l'évaluation répétée de la fonction F nécessaire à la discrétisation ce qui implique un temps de calcul relativement important et nuit à un usage interactif.

Enfin pour mailler un solide, la méthode la plus simple (par décomposition spatiale) pour détecter la surface, ne nécessite que le test d'appartenance d'un point au solide, ce qui revient à une évaluation de la fonction potentiel F pour un point donné. En effet, le calcul du point d'intersection d'une arête avec la frontière, nécessaire au calcul d'intersection d'une cellule (cube, hexaèdre, tétraèdre) avec la frontière peut être réalisé par dichotomie (voir I.2.1.a.3). Cependant, quand il est nécessaire de contrôler le maillage par la courbure de la surface, comme c'est le cas dans [WIT94], il faut en général disposer des dérivées premières.

Ces modèles de courbes, de surfaces sont utilisés conjointement avec deux types de structure permettant de lier certaines de ces entités : la représentation par les frontières (BRep : Boundary Representation) pour les courbes et surfaces paramétriques, et/ou l'historique de construction (CSG : Constructive Solide Geometry). Ces structures peuvent également être utilisées simultanément.

I.3.3 Modèle BRep

Dans un modèle BRep (Boundary REPresentation, ou représentation par les frontières), la géométrie est décrite par la peau de l'objet, ainsi que le côté où se trouve la matière. La peau peut être représentée par un maillage (composé de facettes planes), ou par des surfaces analytiques ou paramétriques. Le modèle BRep ne contient que le résultat des opérations appliquées à la géométrie, c'est-à-dire l'ensemble des informations qui définissent la forme du solide. Deux types d'informations y sont conservées, souvent séparés dans le modèle informatique : la topologie (relations entre les entités géométriques qui sont les sommets, les arêtes, les contours et les faces) et la géométrie (forme ou position de chaque entité géométrique : les points, les équations de surface, éventuellement la normale aux faces planes, etc.). Ce modèle est très utilisé en CAO car il facilite un certain nombre de traitements : la cotation, le calcul de volume, la préparation pour une machine à commande numérique, mais aussi le maillage pour la visualisation, le prototypage rapide et la construction et l'adaptation de maillage pour la simulation notamment dans un cycle calcul par éléments finis / adaptation de maillage. Lorsque les surfaces sont gauches, on utilise parfois le terme de « BRep exacts » pour souligner le fait la géométrie est continue et ne constitue pas une approximation tel un maillage. Cependant à cause de l'ambiguïté de l'intersection de surfaces paramétriques et à cause des erreurs numériques induites par la représentation finie des réels, ces BRep ne peuvent être exacts. Nous garderons cependant ce terme concis pour les distinguer des BRep polyédriques.

I.3.4 Modèle CSG

Dans un modèle CSG (Constructive Solid Geometry, ou arbre de construction), le système conserve sous la forme d'un arbre, un historique de la construction de l'objet. Cet historique représente dans la plupart des cas :

- aux nœuds terminaux de l'arbre, des objets primitifs (cylindres, tores, parallélépipèdes, etc.) et éventuellement des demi-espaces. Les objets primitifs sont alors représentés par une abstraction (par exemple l'axe et le rayon pour un cylindre),
- aux nœuds non terminaux, les opérations servant à construire des objets (extrusions), à les combiner (opérations booléennes) ou à les transformer (déplacement, homothétie, etc.).

Le principal avantage de ce modèle est qu'il permet de revenir en arrière durant la conception, par exemple pour supprimer un congé ou déplacer un trou. De plus, les opérations booléennes sont triviales tant que l'on ne cherche pas à expliciter la frontière, c'est-à-dire à évaluer l'arbre.

C'est là son principal défaut car l'évaluation de sa frontière est nécessaire à certains traitements, en premier lieu la visualisation. Pour pouvoir visualiser l'objet en temps réel durant sa conception, les algorithmes de rendu utilisés exploitent un maillage surfacique triangulaire, la technique de lancés de rayons étant trop lente. L'obtention d'un tel maillage à partir seulement d'un arbre CSG est longue à obtenir car il faut soit appliquer une décomposition spatiale, soit construire un BRep intermédiaire. De plus, les sommets, les arêtes et les faces d'un BRep constituent des éléments de dialogue pertinents pour coter le solide. C'est pourquoi un BRep est conservé simultanément pour gérer à la fois la visualisation immédiate et le dialogue. La construction de maillage se fait alors directement à partir du modèle BRep dont la frontière est connue explicitement.

Dans les systèmes commerciaux, le modèle CSG est enrichi par des primitives étendues (extrusions, révolutions) et des caractéristiques de forme (ou features) que l'on peut assimiler à des sous-arbres paramétrés. Ces caractéristiques permettent de définir et d'appliquer au solide une rainure, une nervure, un trou, une poche, une gorge, un congé, un chanfrein, etc. Typiquement les logiciels de CAO utilisent un modèle hybride historique-BRep comportant des surfaces paramétriques et parfois des surfaces implicites. Ils proposent des caractéristiques de formes et la tendance actuelle est d'intégrer des informations sémantiques de plus en plus élevées, dans le but de modéliser la connaissance de l'expert pour prendre en compte des contraintes de fabricabilité en fonction d'un procédé, des règles métiers, etc.

Conclusion

Ce chapitre a présenté les différents thèmes abordés dans cette étude. Nous avons vu les A.E.M. ainsi que les propriétés qu'elles requièrent concernant les maillages. Les méthodes de calcul pour la simulation nécessitent des éléments respectant des critères de forme, direction, et proportion. Les variations de taille entre éléments adjacents doivent être progressives pour que la précision des calculs soit significative. Le domaine de la synthèse d'images exige également des éléments non plats. Cependant les contraintes de taille et d'aspect sont beaucoup moins fortes. Là aussi, il est préférable de minimiser le nombre de triangles, de manière à ce que la visualisation soit rapide et dans le cas d'images animées, que l'animation soit fluide. La fabrication rapide de prototypes est l'AEM la moins contraignante sur la forme des éléments puisque, dans une certaine mesure, les éléments peuvent être plus ou moins allongés.

La manière de spécifier les propriétés souhaitées peut différer d'une AEM à une autre bien que le traitement local au niveau des triangles puisse être le même. Ainsi, dans le cadre d'un mailleur générique, il faut déterminer comment les requêtes de construction et de raffinement de maillage vont se faire auprès du module de maillage et comment les paramètres de ces requêtes, connus des AEM, vont pouvoir être exploités au niveau du mailleur.

Nous avons vu ensuite les différentes techniques de maillage en fonction des éléments souhaités (triangles/tétraèdres, quadrangles/hexaèdres) et nous avons cherché à extraire ce qui est susceptible d'être factorisé parmi les différentes méthodes. Il existe un petit nombre de squelettes d'algorithmes généralisant le fonctionnement de la plupart de ces méthodes. Évidemment, chaque méthode a aussi une partie spécifique. Il s'agit soit de fonctions faiblement couplées à l'algorithme général de la méthode, soit de données dont la portée est globale et sur lesquelles ont lieu des modifications en différents endroits de l'algorithme. Le premier aspect va dans le sens de la conception d'un mailleur générique alors que le second s'y oppose car chaque méthode utilise des paramètres et des données globales qui lui sont spécifiques. La suite du mémoire détaille les solutions que nous proposons pour contourner ces difficultés.

La dernière partie du chapitre a brièvement présenté les principaux modèles géométriques utilisés en CAO. Nous avons vu les problèmes que posent chacun de ces modèles pour la construction de maillage, ainsi que leurs points forts. Les méthodes de maillage nécessitent

des services de la part du modèle géométrique qui a été utilisé lors de la conception de la forme du solide. Ces modèles étant variés et en évolution, la conception d'un mailleur générique passe également par l'abstraction du modèle géométrique sous-jacent. Il est donc nécessaire de s'intéresser à la manière avec laquelle le module de maillage et le modèle géométrique communiquent.

En résumé, la présente étude bibliographique pose trois problèmes principaux :

- la communication des A.E.M. avec le module de maillage,
- la communication du module de maillage avec le modèle géométrique,
- la nécessité, pour le module de maillage, d'une architecture interne capable de prendre en compte la diversité des contraintes imposées par les A.E.M.

Les réflexions que nous avons menées montrent que les solutions apportées aux deux premiers problèmes apportent de nouvelles contraintes sur la réalisation de l'architecture interne du mailleur générique. Nous avons donc choisi de présenter d'abord les solutions que nous proposons pour les deux premiers problèmes. Elles font l'objet du prochain chapitre.

Chapitre II : Concepts pour l'intégration des Applications Exploitant un Maillage

Après avoir vus les différentes familles d'algorithmes de maillage dans le premier chapitre, nous abordons ici leur cadre d'utilisation.

La première partie montre les défauts de la liaison actuelle entre les systèmes de conception et les AEM et en expose les raisons. Les parties suivantes proposent une vue générale de notre proposition, qui consiste en une architecture logicielle, et détaillent les composantes principales de cette architecture.

II.1 Problématique

Cette partie présente une réflexion sur la situation actuelle quant à la construction et l'adaptation du maillage. Les interactions entre système de conception et AEM sont d'abord exposées, puis suit une brève présentation des formats d'échange actuels et de leurs limites.

II.1.1 Interactions entre Applications Exploitant un Maillage

Que l'AEM soit un logiciel dédié au calcul, au prototypage rapide ou à la visualisation, des erreurs sont présentes dans les modèles géométriques, qu'il s'agisse de maillage ou d'un modèle plus riche. Ces erreurs proviennent de la manière avec laquelle les AEM et les systèmes de conception interagissent. Nous entendons par systèmes de conception, les systèmes de CAO ainsi que les logiciels de modélisation géométrique en général.

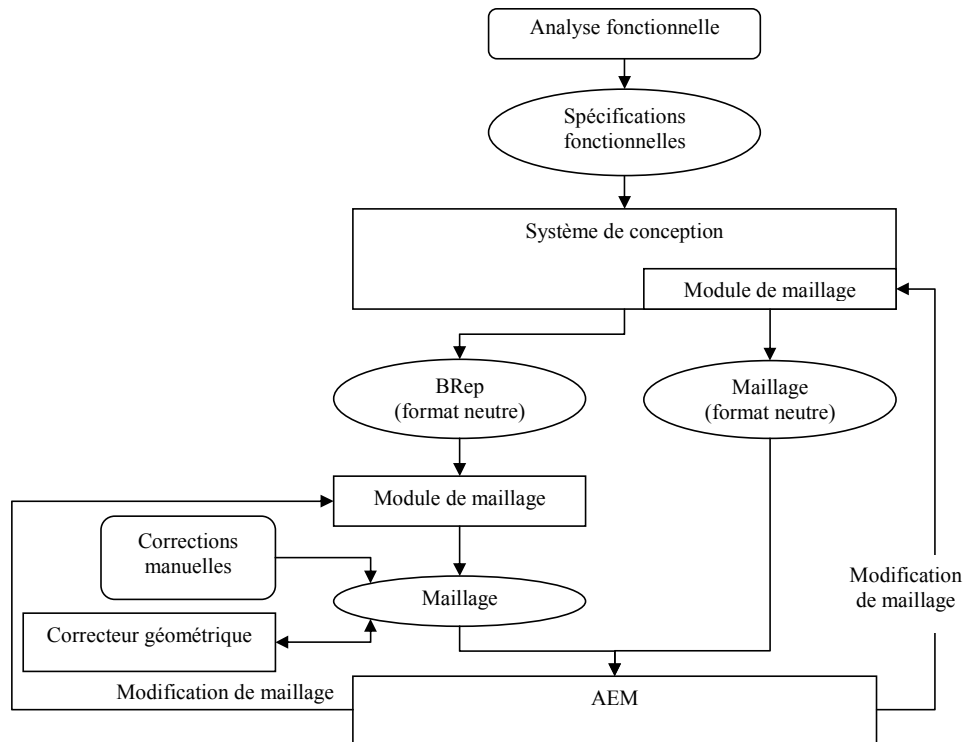
Le plus souvent, la conception d'un produit est précédée d'une analyse fonctionnelle durant laquelle sont données des informations sur les caractéristiques du produit (la résistance mécanique, les propriétés thermiques, l'esthétique, etc.). Après la modélisation géométrique,

un maillage est construit en vue de l'utilisation d'une AEM. Selon les AEM, il y a plusieurs scénarii.

En **prototypage**, le maillage est réalisé dans le système de conception, puis transféré à l'AEM par un fichier au format neutre conservant uniquement le maillage (figure II.1-1). Mais le fichier reçu comporte souvent des erreurs topologiques. Le maillage ne représente pas un ensemble de faces fermé cohérent. Ces erreurs sont des éléments inversés, des recouvrements de triangles ou encore des fissures rendant le maillage inexploitable. Elles sont corrigées automatiquement, lorsque cela est possible, ou manuellement par un opérateur.

Il y a deux scénarii pour réaliser une **simulation** :

- 1^{er} cas : les algorithmes de maillage sont intégrés au système de conception. Le modèle transféré au système de simulation est donc un maillage. Cependant, le manque d'intégration des systèmes fait que si l'AEM nécessite une modification du maillage, celui-ci est reconstruit complètement car il n'existe pas de protocole de communication standard pour formater les requêtes de modifications adressées au système de conception par une AEM.
- 2nd cas : les algorithmes de maillage sont extérieurs au système de conception (figure II.1-1). Le modèle transféré, servant de base à la construction du maillage, est alors un Brep dit exact, c'est-à-dire éventuellement composé de surfaces gauches. Il est stocké et transféré par un fichier au format neutre. Des erreurs apparaissent à la lecture du Brep exact (fissures, recouvrements de faces), ce qui pose des problèmes pour mailler correctement l'objet. Elles sont corrigées en partie automatiquement, et en partie manuellement. Le maillage est ensuite construit et si le module de maillage n'est pas intégré au système de simulation, il faut transférer le maillage en utilisant à nouveau un fichier au format neutre.



Légende :

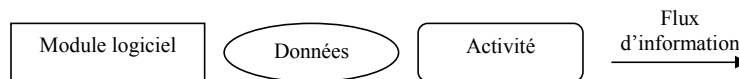


figure II.1-1. Interaction actuelle entre les logiciels

En ce qui concerne la **visualisation**, il y a deux cas selon que le maillage est utilisé pour une visualisation temps réel ou une visualisation plus réaliste nécessitant un temps de calculs plus important. Dans la première situation, le maillage est créé et utilisé immédiatement dans le système de conception et il n'y a pas de transfert. Il est inutile que le maillage soit très précis : il doit comporter relativement peu de triangles, pour que l'affichage ne ralentisse pas le dialogue homme-machine, tout en évitant malgré tout les ambiguïtés dans l'interprétation visuelle. Dans le second cas, il s'agit d'une visualisation plus fine, comme le nécessiterait une image de publicité plus ou moins réaliste du produit à fabriquer. Le maillage doit alors être plus fin aux endroits où cela est nécessaire (zones comportant des petits détails ou zones proches du point de vue). Les modules effectuant le maillage et les calculs d'image peuvent être extérieurs au système de conception et nécessitent alors un transfert de la géométrie (le BRep exact). Dans ce cas, c'est par un fichier au format neutre (Brep exact) que se fait le transfert et les inconvénients de cette procédure, déjà mentionnés plus haut, surgissent à nouveau.

Les trois scénarii qui précèdent soulignent la présence d'erreurs. Ces dernières proviennent soit d'erreurs inhérentes à la représentation BRep, soit d'utilisations malheureuses de cette même structure, soit de défauts des formats neutres. Nous consacrons la fin de cette partie à une étude plus détaillée de ces erreurs et de leurs causes.

Généralement, en CAO, la construction du maillage se base presque toujours sur le seul BRep. Cette façon de faire présente différents défauts. Premièrement, les modeleurs géométriques, et le BRep en particulier, comportent des défauts. Bien que la jointure des surfaces gauches soit bien définie mathématiquement, elle est, en pratique, à la source de nombreuses erreurs topologiques (recouvrement, fissure). Ces mêmes incohérences proviennent également des défauts créés à l'export des données vers un fichier au format neutre, des défauts créés à l'import des données, d'une divergence dans l'interprétation des normes d'échange, de la différence des représentations internes des données (BRep/CSG, B-spline/NURBS), d'une mauvaise utilisation du logiciel par l'opérateur (création de faces possédant des coins très fins, intersection de faces faisant un angle faible, manipulation de surfaces de degré inutilement élevé) ([BOR02], [BRU02]). Ces défauts sont corrigés parfois automatiquement mais c'est un problème difficile et ils exigent aussi des interventions manuelles ([PET01], [SHE01]). [GOS00] propose des règles d'utilisation de certains logiciels de CAO pour éviter les inconvénients liés au transfert. Elles suggèrent entre autres d'utiliser des restrictions pour éviter les carreaux triangulaires, les bords alignés et les angles vifs, d'éviter les arêtes trop petites et les situations où deux carreaux sont portés par même une surface (voir annexe B)

Ces défauts constituent une partie des raisons des interventions manuelles et des limitations imposées par le choix du BRep comme unique référence pour la construction du maillage. D'autres limitations viennent du fait que le BRep ne peut fournir que des informations sur la géométrie et la topologie. Or les facteurs qui peuvent avoir une influence sur les caractéristiques du maillage peuvent être d'une autre nature. En effet, l'analyse fonctionnelle ou les outils utilisés durant la conception peuvent être exploitées pour déterminer les spécifications du maillage (densités de nœuds, type d'éléments, maillage structuré ou non structuré, type d'algorithme de maillage à appliquer, etc.).

Considérons par exemple un maillage du système de remplissage d'un moule de fonderie dont on veut simuler le remplissage. Les endroits où le système de remplissage débouche dans les cavités du moule s'appellent les attaques. Ce sont des conduites de petite taille, rapportées aux

dimensions du système complet mais de grande importance pour la simulation de l'écoulement du métal en fusion car les contraintes y sont fortes. Si le maillage du système de remplissage est relativement grossier, afin de ne pas exagérer les temps de simulation, les attaques peuvent être amenées à disparaître ou voir leur géométrie très dégradée. Dans un cas comme dans l'autre, les résultats de la simulation ne sont plus significatifs. Si au contraire le maillage est fin, de façon à garantir la présence des attaques dans le maillage, la durée de la simulation devient disproportionnée. Pour éviter ces inconvénients, l'opérateur chargé de préparer la simulation doit fournir une quantité importante d'informations concernant le maillage. Dans de nombreux cas, ces informations ont déjà été fournies par les opérateurs qui l'ont précédé, pendant l'analyse fonctionnelle, la conception préliminaire ou détaillée. Il s'agit du modèle fonctionnel, des règles métiers, des caractéristiques de forme, des connexions etc. Par exemple, les fonctions associées à certaines parties de la pièce permettraient de préciser leur importance pour la simulation et donc la finesse du maillage dans ces zones. Ou encore le fait de savoir que telle partie de l'objet peut être obtenue par extrusion permet d'utiliser tel algorithme de maillage fournissant un maillage volumique à base d'hexaèdres. Il est donc nécessaire de chercher à exploiter des informations présentes dans le système de conception mais absentes du BRep.

Enfin un dernier inconvénient des BRep dits exacts est qu'ils ne le sont pas toujours, parce que le calcul d'intersection des surfaces est souvent difficile parce qu'il induit des équations de degré élevé et il est parfois tout simplement impossible de manière exact ([BER95]). Or les AEM réalisent leur traitement sur le maillage, pas sur la représentation « exacte ». Par conséquent, elles requièrent d'une part un maillage vérifiant un certain nombre de propriétés, et d'autre part, la possibilité de modifier le maillage.

II.1.2 Échange d'informations géométriques

Actuellement le transfert d'informations entre la CAO et d'autres AEM se fait par des normes d'échange tels que STL, DXF, IGES, VDA, SET ou STEP. Nous les présentons maintenant brièvement.

- STL (STereoLythography, [STL88]) est un format provenant du domaine du prototypage rapide. Ce format ne conserve explicitement que la géométrie du maillage, pas la topologie. Chaque triangle est défini par les coordonnées de ses sommets ainsi que le vecteur normal dirigé vers l'extérieur de l'objet. Il s'agit d'un modèle pauvre mais qui est assez répandu puisque la plupart des modeleurs géométriques proposent un enregistrement

dans ce format. Bien qu'il subsiste des problèmes de reconstruction de la topologie du maillage liés à la tolérance, il est de plus en plus utilisé car ces inconvénients sont moindres que ceux des formats neutres des BRep exacts.

- DXF (drawing interchange format) est une norme de fait. Ce format de fichier provient d'Autodesk. Il permet de transférer des plans comprenant des courbes de Bézier, des B-splines mais ne permet pas de conserver la topologie d'un BRep 3D.
- Iges (Initial Graphics Exchange Standard) est une norme de modélisation de la géométrie utilisé comme format d'échange. Aujourd'hui ce format permet de représenter les éléments d'un BRep exact, ainsi que l'historique de construction (CSG). Il y apparaît donc les entités de base : sommets, arêtes, faces, etc. ainsi que le support mathématique des courbes et surfaces (B-splines rationnelles, carreaux restreints). Cependant, les logiciels de construction de maillages n'exploitent pas les informations de l'historique. Ce format ne stocke pas d'information du produit en dehors de sa géométrie et de son paramétrage. C'est un format très utilisé à l'heure actuelle.
- VDA (Verband der Deutsche Automobilindustrie) permet de stocker un Brep exact. Elle est dédiée au transfert de surfaces gauches. La version VDA-IS (VDA- Iges Subset) est un sous-ensemble de Iges jugé trop ambigu où la possibilité de représentation est plus restreinte. La nature des informations stockées reste également dans ce cas purement géométrique.
- SET (Standard d'Échange et de Transfert) : conserve la définition de la géométrie et l'historique de construction d'un objet (BRep, CSG), des dessins techniques associés à un objet (plans, vues, annotations,...), des modélisations par éléments finis, des modélisations spécifiques aux domaines de la schématique fluide, de la schématique électrique / électronique et de la tuyauterie. Cette norme est vouée à être remplacée par STEP.
- STEP (Standard for the Exchange of Product) est une norme de spécification du produit comprenant évidemment la géométrie (BRep, CSG), mais aussi d'autres informations de plus haut niveau (entités pour le calcul par éléments finis, pour les machines à commande numérique, etc.). Il s'agit davantage d'un protocole de communication que d'un simple format de données. Step a été créé pour donner une réponse aux problèmes d'intégration des systèmes en fournissant des langages communs de spécifications : IDEF0 pour les processus de l'entreprise, IDEF1, EXPRESS_G et NIAM pour une vue conceptuelle des

données et des traitements. Cependant, sa mise en œuvre se révèle complexe car elle implique, pour chaque corps de métier, l'écriture de nombreuses bibliothèques de schémas et nécessite leur interfaçage avec les modèles internes à chaque logiciel.

- Par ailleurs, d'autres solutions sont en cours d'étude par l'OMG¹ pour échanger l'information en CFAO ([**OMG01**]). Elles sont basées sur le paradigme objet comme les composants DCOM ([**COM95**]) ou l'architecture CORBA ([**COR01**]).

La plupart de ces formats souffrent d'une inertie par rapport aux évolutions des systèmes. Même STEP qui est conçu pour répondre à ces problèmes d'interfaçage demande beaucoup de temps et de moyens avant d'être exploité. Par exemple, un des logiciels du marché, Catia, propose des outils de maillage et de spécifications du maillage (et par ailleurs un module de calcul par éléments finis) mais ne permet pas de sauvegarder ces informations dans un format STEP.

Nos travaux sont complémentaires de ces normes dans le sens où le transfert d'informations quelles qu'elles soient, nécessite un protocole de communication normalisé. Dans cette étude, nous cherchons à améliorer le mode d'interaction entre le système de conception et les AEM de manière à faciliter les évolutions tout en cherchant à assister l'opérateur dans une phase clé de la conception à savoir l'obtention de maillages.

II.1.3 Objectifs et proposition

Notre objectif est de réduire les interventions manuelles déjà évoquées : la correction des modèles géométriques et l'entrée des spécifications des maillages.

Pour réduire les interventions dues aux erreurs des modèles, nous proposons de réduire le nombre de fois que le modèle géométrique est interprété en **interrogeant** tant que cela est possible, **le modeleur géométrique ayant servi à sa conception**. Cela évite les erreurs dues aux différences d'interprétation et aux ambiguïtés des normes d'échange.

Pour réduire l'entrée des spécifications, nous proposons de **pré-optimiser les maillages** lorsque cela est possible. Les systèmes de conception prennent en compte de plus en plus de connaissances de manière à fournir les mêmes fonctionnalités que les logiciels dédiés à un métier. L'idée sous-jacente à la pré-optimisation est d'exploiter les connaissances qui sont codées aujourd'hui dans ces logiciels métiers et qui seront bientôt dans les logiciels

¹ *Object Management Group*

de conception à vocation plus générale. En simulation par exemple, classiquement, un premier maillage est construit et utilisé. Il est ensuite raffiné de façon à ce que la deuxième utilisation donne de meilleurs résultats. Les interventions humaines sont donc multipliées. Nous souhaitons éviter cette pratique en optimisant autant que possible les propriétés des maillages avant la première utilisation à l'aide d'informations issues de l'analyse fonctionnelle. Ces informations sont ensuite utilisées pour spécifier les propriétés du maillage.

Comme il n'existe pas qu'un système de conception, le mailleur doit pouvoir être utilisé par différents systèmes de conception et pour différentes AEM. Ceci implique que **ce module de maillage** n'est pas intégré à l'un des systèmes mais **doit rester indépendant**. Cela a plusieurs conséquences. Il faut en effet :

- s'abstraire du modèle géométrique
- être capable de produire tout type de maillage pour les trois AEM prises en compte dans cette étude
- pouvoir interfacer le module de maillage avec les autres systèmes, qu'il s'agisse de logiciels de conception ou d'AEM.

Nous voyons maintenant notre apport concernant l'interaction des systèmes pour une meilleure interopérabilité.

II.2 Description générale de l'architecture

Les spécifications du paragraphe précédent nous ont conduit à une architecture dont les composantes et le fonctionnement général sont présentés dans cette partie. Le formalisme choisi pour représenter les données et les traitements de l'architecture proposée est UML (Unified Modelling Language) dont un bref récapitulatif est proposé en annexe A.

La première composante de l'architecture proposée est le système de conception. Nous faisons l'hypothèse qu'il comporte entre autres trois modules : un modeleur géométrique qui peut être un Brep, un CSG, ou un autre modèle, un modèle de plus haut niveau sémantique (historique de construction, modèle fonctionnel) et un module d'assemblage, dont la principale fonction est de construire un maillage de la pièce en *assemblant* des maillages de parties de la pièce, d'où son nom. Les modèles de haut niveau sémantique sont de plus en plus intégrés aux systèmes de conceptions actuels, par contre, le module d'assemblage est une

nouveauté qui implique une refonte des logiciels de conception. Sa présence se justifie par les raisons suivantes, qui sont vues plus en détails dans la partie II.4 :

- Les modèles géométriques ne se maillent pas tous de la même façon. En effet, la procédure générale de maillage d'un BRep (arête par arête, face par face) n'est pas la même que celle d'un CSG (méthode globale de type décomposition spatiale). Par conséquent, Il faut donc avoir plusieurs algorithmes à disposition (pour les arêtes, pour les faces, pour les volumes).
- Les informations pouvant servir à la construction du maillage se situent au niveau du système de conception ou plus en amont encore (analyse fonctionnelle). Or le module d'assemblage dispose de ces informations puisqu'il est intégré au système de conception.
- Ne pas transférer plus d'informations sur le produit que nécessaire permet d'assurer la confidentialité des données dans le cas où l'AEM se situe chez un sous-traitant. De plus, le fait d'envoyer uniquement un maillage évite une partie des problèmes de transfert des données liés aux BRep (transfert des surfaces gauches).

Nous avons vu qu'actuellement, la construction des maillages se fait en divers endroits avec les inconvénients que cela comporte. Nous proposons de construire et adapter le maillage par un unique module de maillage.

Certaines AEM ont besoin d'un contrôle relativement important sur le maillage (contrôle du ratio d'aspect et de la taille des éléments, etc.), alors que d'autres nécessitent un maillage moins contraint mais le résultat doit être disponible rapidement. Il n'existe pas aujourd'hui d'algorithme de maillage permettant d'obtenir tout type de maillage en temps réel. Par conséquent, bien que cette voie soit intéressante, notre module de maillage prendra la forme d'une bibliothèque d'algorithmes de maillages plutôt que de se réduire à un unique algorithme.

Il s'agit d'un module extérieur à tout système, dont les fonctions sont invoquées depuis le module d'assemblage. Cette banque d'algorithmes aura en outre la faculté de fournir une liste d'algorithmes répondant à certaines exigences (être capable de mailler un domaine plan, produire des mailles triangulaires, être rapide etc.), ce qui permettra à son client de faire un choix parmi les bonnes possibilités seulement. Nous considérons donc ce module comme une bibliothèque indépendante d'algorithmes, d'où son nom : Bibliothèque d'Algorithmes de

Maillages Génériques ou Spécialisés (BAMGS). Le terme « Générique » sera justifié dans le chapitre 4, qui fait une analyse détaillée de la BAMGS. Trois raisons motivent le choix d'envisager un module de maillage indépendant :

- La concision : éviter la redondance d'algorithmes pour toutes les raisons connues en génie logiciel (facilité de développement du code, de maintenance, d'assurer la cohérence et les évolutions).
- La capacité à être utilisé avec différents systèmes de conception : il faut donc être indépendant du modèle géométrique, et par conséquent extérieur au système de conception.
- La capacité à produire des maillages utilisables par des AEM de nature diverse : il faut donc être indépendant de ces applications.

Les autres composantes de cette architecture sont les AEM. Un scénario générique de son fonctionnement comporte les phases suivantes :

- Le module d'assemblage reçoit une requête de maillage (*Cas TraiterRequete*, figure II.2-1) d'un opérateur qui souhaite appliquer un traitement nécessitant un maillage. Cette requête est accompagnée de spécifications sur le maillage en rapport avec l'utilisation qui va en être faite. Ces spécifications proviennent de l'analyse fonctionnelle, donc elles sont fournies par la base de connaissances du système de CAO dans la mesure du possible, ainsi que par l'opérateur.

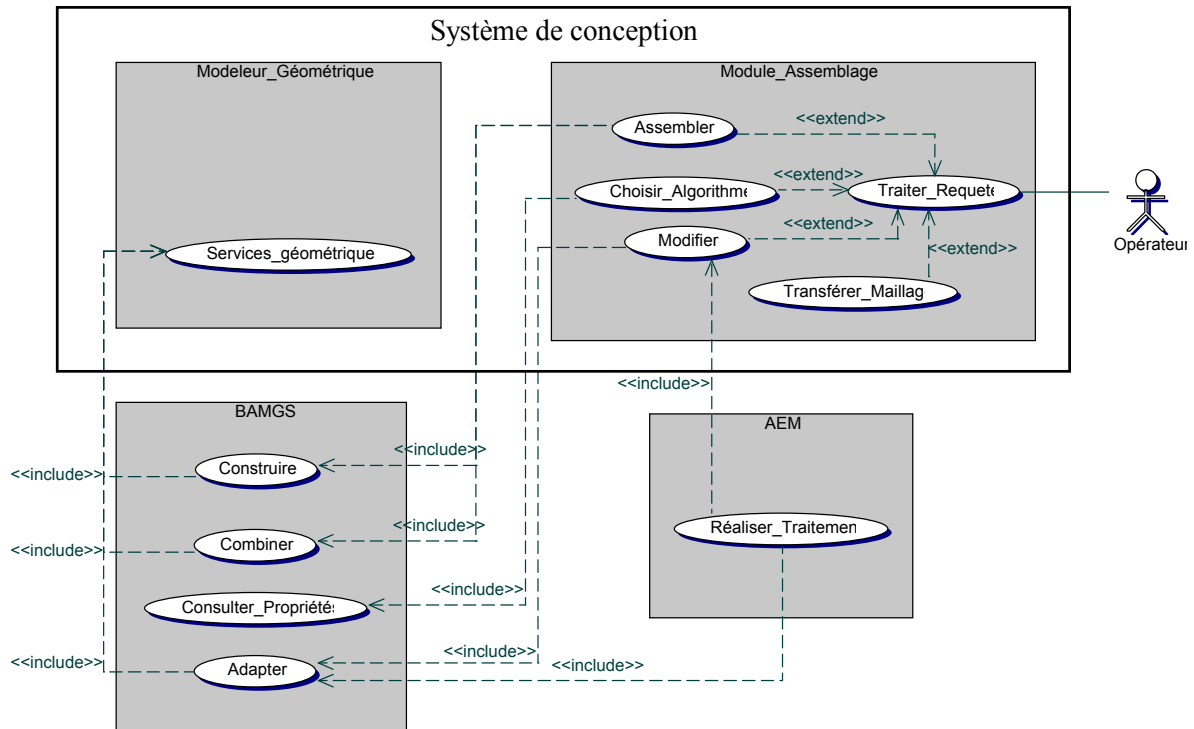


figure II.2-1. Schéma général

- Selon le type de modeleur et le langage associé à l'historique de construction, le module d'assemblage va procéder différemment pour mailler le solide et spécifier les paramètres du maillage (cas *Assembler*, figure II.2-1). Par exemple pour un Brep, l'entrée des données et la construction se font dans l'ordre arêtes, faces et solide. Le module d'assemblage va donc, entre autres choses, demander à la BAMGS des fonctions de maillage de face, choisir parmi les propositions qui lui sont faites, celle qui satisfait au mieux les spécifications attachées à la requête de maillage, et déclencher la fonction choisie.
- Les méthodes de maillage de la BAMGS, qu'il s'agisse de construction, d'adaptation ou d'autres opérations, ont besoin de consulter la géométrie. Pour cela, nous supposons que les entités géométriques peuvent être caractérisées par un identifiant et un modeleur. Lorsque le module d'assemblage invoque les méthodes de la bibliothèque d'algorithmes de maillage, il fournit les identifiants car il connaît le modeleur géométrique. La BAMGS appelle les fonctions du modeleur en désignant les entités par leur identifiant (Cas *Services_Geometriques*, figure II.2-1).
- Tout en assemblant le maillage, le module d'assemblage associe les entités du maillage (nœuds et mailles) aux entités du modèle géométrique et éventuellement aux

spécifications données par l'analyse fonctionnelle ou l'opérateur concernant l'AEM (par exemple les conditions aux limites pour un calcul par élément finis, ou la position des sources lumineuses et de la caméra pour un rendu réaliste). (Cas *Transférer_Maillage*, figure II.2-1). Le maillage peut alors être envoyé à l'AEM.

- L'AEM exécute le traitement qui lui incombe (Cas *RealiserTraitement*). Pour cela, il est possible qu'elle ait besoin de déplacer des nœuds, d'affiner le maillage ou d'en changer certaines spécifications. Selon les cas (voir II.5), l'AEM adresse alors une requête de modification du maillage soit au module d'assemblage ou directement à la BAMGS. (cas *Modifier*, figure II.2-1). C'est ce type de requête qui rend obligatoire l'association décrite à l'étape précédente entre, d'un côté, les entités du maillage et, de l'autre, les entités géométriques et les spécifications.

II.3 Un service particulier : les opérateurs d'assemblage

Parmi les services offerts par la BAMGS, il en est un dont nous devons donner une brève description pour faciliter la description à venir du module d'assemblage, c'est le service d'assemblage de maillages. On peut assembler des maillages à l'aide d'opérateurs qui sont le collage, le découpage et le raffinement et les opérateurs booléens. Nous décrivons ici brièvement leur rôle.

Le rôle de l'opérateur de collage et découpage est de permettre d'assembler des maillages qui possèdent une partie commune (des sommets, des arêtes et/ou des faces) ou de scinder un maillage en deux parties distinctes. Cela permet de réutiliser des maillages déjà existants, lorsqu'il s'agit par exemple de surfaces fonctionnelles. L'opérateur de collage sert également lors du maillage de deux faces portées par des surfaces paramétriques dont l'intersection est ambiguë. Le maillage a lieu en général dans l'espace paramétrique, or les images des deux sommets situés respectivement dans l'un et l'autre espace paramétrique ne peuvent pas avoir pour image exactement le même sommet dans l'espace réel. Les maillages sont donc construits dans l'espace paramétrique, puis collés dans l'espace réel.

Les opérateurs booléens ont pour but de permettre d'assembler des objets qui, durant la conception, ont été impliqués dans une opération booléenne. Dans ce cas, il est possible de construire le maillage de l'ensemble en réalisant l'opération booléenne sur les maillages préalablement construits des opérandes. Ceci permet entre autres de construire le maillage d'un arbre CSG sans l'évaluer préalablement par un BRep intermédiaire.

Lorsque les opérations booléennes sont réalisées de cette manière, il est possible que des intersections entre les maillages soient ignorées ou que des intersections aient lieu sur les maillages alors que les objets exacts n'en ont pas. Ceci peut en effet apparaître lorsque le maillage est relativement grossier par rapport à la complexité de la géométrie des opérands, et par rapport à leur proximité. Dans ce cas, les intersections ignorées ou indésirables peuvent être respectivement détectées ou ignorées par l'opérateur de raffinement. Il peut être déclenché soit par le module d'assemblage si la sémantique attachée à chaque opérande permet de déterminer le nombre d'intersections, soit manuellement.

Les aspects algorithmiques de ces opérateurs, en particulier les opérateurs booléens sont présentés dans le chapitre 3.

II.4 Module d'assemblage

Le module d'assemblage joue un rôle déterminant car c'est autour de lui que s'articule la communication entre les systèmes. Rappelons qu'il possède deux principaux rôles :

- ❑ celui d'assembler les maillages des différentes parties de l'objet. Cette tâche incombe à *l'algorithme d'assemblage*.
- ❑ celui de choisir la fonction de maillage adaptée au traitement que va subir le maillage. C'est l'éventuelle base de connaissances présente dans le système de conception qui va donner les informations permettant ce choix. À défaut, il s'agira de l'opérateur.

Il est également à sa charge de fournir à l'AEM toutes les informations dont elle pourrait avoir besoin pour émettre des demandes de modifications (raffinement, simplification, etc.) pour adapter le maillage à une nouvelle situation.

Cette partie présente les algorithmes d'assemblage pour les deux principaux modèles géométriques, le BRep et le CSG, ainsi que la manière de choisir l'algorithme de maillage et ses paramètres.

II.4.1 Algorithmes d'assemblage

II.4.1.a Cas d'un Brep

Dans le cas d'un BRep, la construction du maillage se fait arête par arête, puis face par face, etc. Pour chaque appel d'une fonction de construction, l'algorithme d'assemblage fournit à l'identifiant de l'entité géométrique à mailler et, éventuellement, le maillage des entités

Cette manière de construire ne permet de spécifier les propriétés du maillage que pour les entités que l'on peut désigner, à savoir des arêtes, éventuellement des contours, des faces et des solides. En particulier il n'est pas possible de désigner une partie du solide constituée de plusieurs faces à moins de définir un modèle supplémentaire (tel qu'un historique de construction). Il est de plus nécessaire de spécifier les propriétés pour les entités les plus petites (arêtes), ce qui est fastidieux. L'utilisation d'un historique de construction tel un arbre CSG permet d'améliorer cela.

II.4.1.b Cas d'un arbre CSG

Le maillage d'un CSG peut se faire de 3 manières :

- Par la construction d'un Brep intermédiaire, qui est ensuite maillé. L'arbre CSG est donc évalué, y compris les courbes d'intersection issues des opérations booléennes, impliquant les difficultés déjà évoquées (degré élevé des équations, impossibilité d'effectuer un calcul exact).
- Par une approche globale où tout le maillage est construit sans assemblage de maillages partiels (méthodes par décomposition spatiale, ou méthode s'appuyant uniquement sur un maillage de la frontière). Cette approche n'est pas toujours possible et le contrôle qu'elle offre dans le cas d'un maillage surfacique est souvent faible (notamment la forme des éléments, l'anisotropie, etc.). Elle est essentiellement utilisée pour le maillage de fonctions implicites et est peu utilisée en CFAO¹. Dans cette situation, le paramétrage de la fonction de maillage nécessite l'identifiant du solide et/ou l'identifiant de chaque objet primitif se trouvant aux feuilles de l'arbre CSG, ainsi que les paramètres de contrôle du maillage (comme la taille des mailles par exemple).
- Enfin une dernière possibilité est de mailler l'arbre nœud par nœud : les feuilles de l'arbre CSG sont maillées d'abord (cas *Construire1D* et cas *Construire2D*, figure II.4-2), puis les maillages obtenus sont combinés en appliquant les opérations booléennes des nœuds supérieurs, dans un ordre allant des feuilles vers la racine (Cas *Combiner*, figure II.4-2).

Cette approche offre deux avantages. Tout d'abord, la structure d'un arbre CSG est généralement telle que certains sous-arbres réalisent une fonction particulière (tel sous-arbre est une poignée, tel autre décrit le logement dans lequel une autre pièce coulissera ...). Une

¹ Conception et Fabrication Assistées par Ordinateur

partie des spécifications de maillage sont les mêmes pour tous les nœuds de tels sous-arbres (telle que le type des mailles, leur forme, etc.), ce qui simplifie considérablement le travail de spécification. Le deuxième avantage est qu'à l'inverse de la première méthode de maillage (sur les BRep), avec laquelle elle présente quelques similitudes, cette approche combine des maillages et non des objets exacts. Ceci délivre du problème du calcul des intersections de surfaces gauche et permettrait, mais il s'agit d'un apport strictement théorique, de combiner des objets ayant des représentations hétérogènes, même lorsqu'il est impossible de calculer les courbes d'intersection exactes.

L'approche soulève aussi des difficultés : là où se rejoignent deux sous-arbres, il faut harmoniser les spécifications des deux maillages et, si l'AEM l'exige, corriger la forme des mailles au voisinage de l'intersection (des mailles triangulaires ont pu être coupées et devenir très effilées). Par ailleurs, les méthodes de maillage qui offrent le plus de contrôle (anisotropie, proportion, taille des mailles) exigent un BRep. Pour pouvoir les utiliser avec une représentation CSG, il est nécessaire de conserver le BRep de chaque objet primitif.

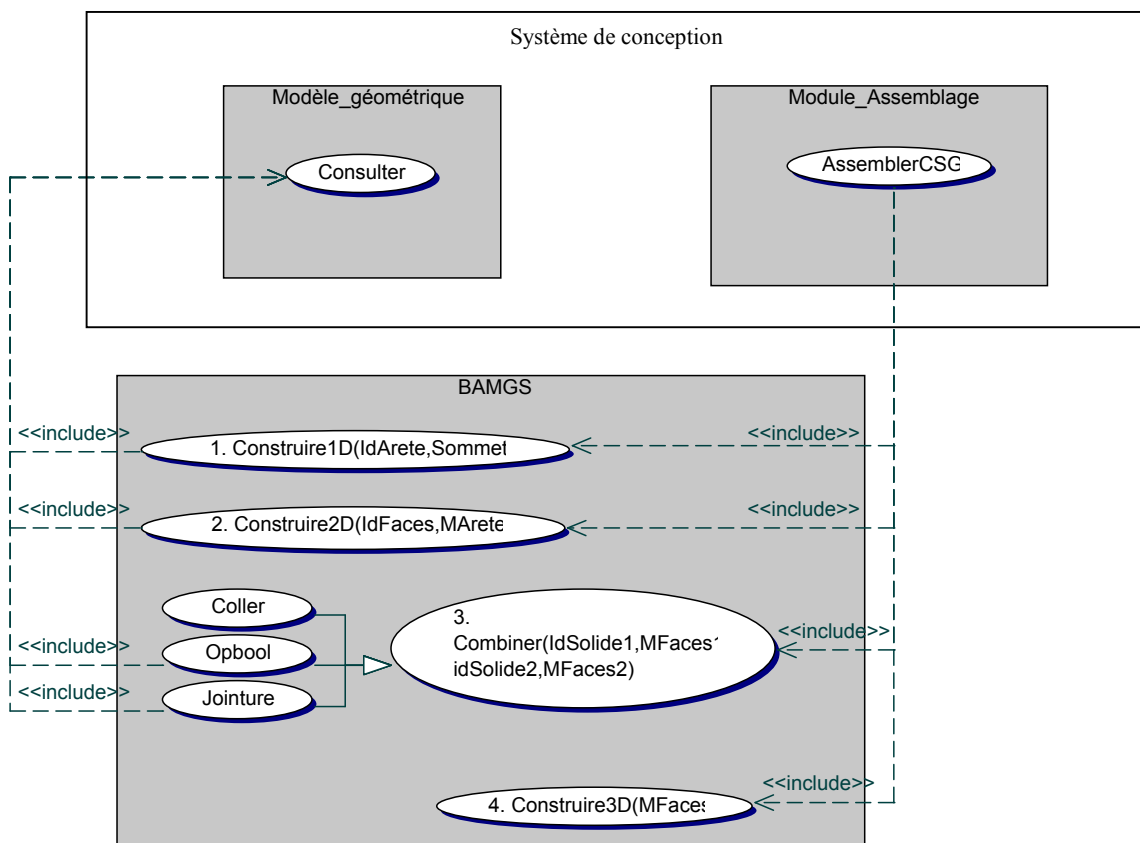


figure II.4-2. Cas d'utilisation du maillage d'un arbre CSG

La méthode que nous préconisons est la dernière car elle permet d'aboutir à des maillages satisfaisant des propriétés contraignantes comme l'anisotropie. Elle permet potentiellement de combiner des maillages portés par des modèles géométriques différents. Cependant, la deuxième méthode peut s'avérer très utile pour le maillage des surfaces implicites complexes utilisés surtout en synthèse d'images car dans ce cas, la construction du Brep d'une feuille de l'arbre peut être très complexe.

II.4.2 Choix des algorithmes et des paramètres

II.4.2.a Problématique

Rappelons que le module d'assemblage doit déclencher diverses fonctions de la BAMGS pour mailler des parties de l'objet puis assembler ces parties. Le choix, automatique ou assisté, des fonctions de maillage n'est pas chose aisée et fait l'objet de la suite de ce chapitre.

La BAMGS stocke des algorithmes et leurs propriétés : obtention d'un maillage en temps réel ou non, contrôle du maillage par un angle, par un écart nodal ou par une métrique, éléments de frontière de bonne qualité ou non, etc. Or les spécifications du maillage dont dispose le système de conception ne correspondent pas de manière immédiate à ces propriétés. La BAMGS ne peut donc interpréter les spécifications, en particulier pour choisir la méthode de maillage la plus adéquate dans un contexte donné.

Ce choix est le résultat d'un processus complexe, que nous n'envisageons pas de manière purement automatique et qui vise à concilier des contraintes, ou contextes, provenant de l'utilisation du maillage visée, de la BAMGS et du modèle géométrique.

Il y a trois contextes à prendre en compte pour pouvoir choisir une fonction de maillage :

- le contexte de l'AEM, à savoir :
 - le type d'AEM : éléments finis, différences finies, visualisation temps réel, visualisation différée et prototypage rapide
 - les spécifications de l'AEM : domaine de déformation plastique ou élastique pour la simulation, le type d'éléments (éléments de plaque, de coque, etc.), la position de la caméra et des sources lumineuses, la nécessité d'appliquer des textures, de simplifier la géométrie, etc.

- le contexte de la BAMGS, à savoir :
 - les paramètres des fonctions de la BAMGS (une métrique exprimant la forme et la taille des éléments, une distance entre nœuds, une métrique de simplification, une valeur d'angle, etc.)
 - les propriétés sur le déroulement des fonctions de la BAMGS (par exemple l'algorithme est interruptible¹) ou sur le maillage que ces fonctions produisent (l'algorithme est sensible à la frontière², les éléments sont triangulaires, etc.)
 - les services géométriques que cette fonction de maillage requiert (le vecteur normal en tout point de la surface, le cercle passant par trois points, etc.)
- le contexte du modèleur géométrique, c'est à dire :
 - les fonctions géométriques disponibles (points de contrôle, équations des surfaces, vecteur tangent en un point d'une surface, etc.)
 - l'historique de construction s'il en existe un, ce qui est le cas dans une grande majorité des systèmes de conception actuels, pour la facilité de conception qu'il apporte.

II.4.2.b Sélection des méthodes candidates dans la BAMGS

La première étape dans le choix de la (ou des) fonction(s) de maillage est d'établir une correspondance entre le contexte de l'AEM et celui de la BAMGS, c'est-à-dire qu'il faut traduire chaque spécification de l'AEM en une ou plusieurs spécifications de maillage. Par exemple, la spécification de l'AEM « simulation de l'écoulement d'un fluide par la méthode des différences finies » se traduit en « les éléments sont volumiques de type hexaédrique » et « la méthode de maillage doit être sensible à la frontière ». Les spécifications de maillage sont en fait les propriétés des fonctions de maillage conservées au niveau de la BAMGS. Cette correspondance des spécifications permet de faire une première sélection sur les fonctions de la BAMGS candidates. Cela n'est pas un simple problème de correspondance entre des données car la spécification pour l'AEM concerne des informations de différents niveaux sémantiques comme c'est le cas dans un modèle fonctionnel. Il s'agit de traduire progressivement des spécifications floues comme « ceci est un système de remplissage pour la fonderie » en « il faut faire une simulation de l'écoulement du métal en fusion » pour

¹ Mais fournit tout de même un maillage valide. C'est le cas d'un raffinement par le noyau de Delaunay.

² C'est-à-dire que les éléments touchant la frontière sont particulièrement de bonne qualité

valider une des fonctions principales, qui se traduit encore et encore en termes plus généraux jusqu'à pouvoir décider quel algorithme de maillage il faut utiliser, quels paramètres il faut lui donner, et quelles informations doivent être fournies à l'AEM pour exploiter le maillage.

Cette sélection, faite sur les propriétés des fonctions de maillage, est ensuite raffinée dans une deuxième étape avec le contexte géométrique. En effet, pour chaque fonction de maillage, la BAMGS conserve également l'ensemble des services géométriques qu'elle utilise. On peut ainsi rejeter les fonctions de maillage dont les services géométriques requis ne sont pas disponibles dans le modèleur géométrique.

Pour réduire les situations de blocage, la BAMGS comporte des fonctions réalisant certains services géométriques de manière approchée afin de proposer des solutions palliatives. Cet aspect est davantage détaillé dans le chapitre dédié à la BAMGS. Ceci étant, établir la relation entre un ensemble de services géométriques fournis et un service nécessaire ne peut s'envisager automatiquement. Ce doit être fait à la main, par un expert du domaine, au cours d'une phase de configuration du module d'assemblage.

Une troisième étape qui peut encore réduire le choix de l'utilisateur est l'utilisation des informations conservées par le système de conception. Le fait de savoir qu'une partie de l'objet a été conçue avec une caractéristique de forme « coque » peut, associé à d'autres informations (« nécessité d'un maillage surfacique »), permettre de décider qu'il faut appliquer des éléments de coque pour un calcul par éléments finis.

II.4.2.c Pré-optimisation

Nous avons vu que pour choisir une méthode de maillage, ce qui permettra d'automatiser la phase de maillage, il faut transformer les spécifications de l'AEM en spécifications de maillage. Or les spécifications de l'AEM et donc du maillage ne sont pas identiques sur tout l'objet. C'est le cas en simulation lorsqu'une partie de l'objet nécessite un maillage plus fin car elle remplit une fonction spécifique (un point d'appui par exemple). Il faut donc disposer d'informations permettant de faire varier les spécifications en fonction de la position sur ou dans la pièce. Or l'analyse fonctionnelle décompose les fonctions principales de l'objet en sous-fonctions jusqu'à ce qu'il soit possible d'attribuer à chacune d'elles une forme. Si on émet l'hypothèse peu restrictive que chaque forme est représentée par une partie de l'arbre CSG décrivant toute la pièce, donc par un sous-arbre, alors il est possible de lier certains sous-arbres à leur fonction ou à leur sémantique et, par là même, d'ajuster leurs spécifications à

cette sémantique. C'est ce que nous appelons la pré-optimisation : une optimisation des paramètres du maillage réalisée avant même qu'une première simulation ait été réalisée.

Reprenons l'exemple de la fonderie. Les attaques (partie étroite à l'entrée des cavités du moule) apparaissent dans l'analyse fonctionnelle car elles sont des parties importantes du système de remplissage des moules. En associant au terme « attaque », le fait que le maillage doit être tétraédrique, sensible à la frontière et que la taille des éléments est dirigée par le modèle fonctionnel, il devient possible de réduire les interventions humaines dans la préparation du maillage.

Par exemple, si le solide est l'agrégat de petits arbres CSG, nous pouvons nous servir de cette représentation pour entrer et conserver les spécifications de taille du maillage. Supposons qu'on veuille construire un maillage surfacique triangulaire qui va donner lieu ensuite à un maillage tétraédrique. Nous nous intéressons ici surtout à la spécification du maillage surfacique dont les mailles sont déterminantes pour la construction du maillage tétraédrique.

La densité de noeuds du maillage est supposée fournie durant la préparation de la simulation, par une distance entre les sommets (ou écart entre noeuds) pour chaque petit arbre CSG. Ces valeurs sont conservées au niveau de chaque feuille. Les opérateurs booléens vus précédemment nécessitent un écart entre noeuds sur tous les contours d'intersections. Cet écart peut être donné par l'analyse fonctionnelle lorsque cette intersection revêt une importance dans le calcul par éléments finis. Sinon elle est déterminée automatiquement par une règle de calcul utilisant les écarts entre noeuds présents localement.

Pour déterminer cette règle, remarquons que les contours d'intersection sont découpés en tronçons durant l'opération booléenne de manière à ce qu'un tronçon soit porté par deux surfaces exactement, puisqu'il est à l'intersection de deux solides (voir chapitre 3). Par conséquent, un tronçon est obtenu par les maillages de deux surfaces exactement donc par uniquement deux solides primitifs de l'arbre CSG. Pour chaque tronçon, l'écart des noeuds, lorsqu'il doit être calculé, est la moyenne des écarts de noeuds imposés sur chaque objet primitif mis en jeu localement. D'autres règles de calcul peuvent être appliquées : le minimum, le maximum, une pondération pour l'écart entre noeuds de l'un des opérandes, etc.

Les écarts entre noeuds sont donc conservés au niveau des feuilles, ainsi qu'au niveau des tronçons des contours d'intersection. Le module d'assemblage exploite ensuite ces

informations pour paramétrer l'appel des fonctions de maillage sur les objets primitifs ainsi que les opérateurs booléens.

Cette approche trouve toute son utilité dans la conception routinière où des fonctionnalités sont récurrentes. Cependant, l'architecture du projet DIJA, qui fait l'objet de la partie II.4.3, permet d'exploiter des informations communes à différents métiers et donc d'étendre la pré-optimisation à la conception innovante.

II.4.2.d Scénario générique de la phase de maillage

En résumé, voyons le scénario générique d'utilisation de la BAMGS. L'utilisateur envoie une requête au système de conception signifiant qu'il souhaite valider une des fonctionnalités du produit et, pour cela, un maillage est nécessaire. Le système de conception traduit cette requête en un ensemble de fonctions de maillage candidates avec les informations dont il dispose sur les contextes (AEM, maillage et géométrie). Il consulte pour cela les propriétés et les paramètres des fonctions de maillage de la BAMGS. L'opérateur termine de préciser les spécifications nécessaires au choix d'une méthode de maillage (ou plusieurs) soit pour tout le solide, soit pour chaque partie selon ses besoins. S'il existe des règles permettant de déterminer les valeurs des paramètres de cette (ou ces) méthode(s), elles sont appliquées, sinon les valeurs sont demandées à l'utilisateur.

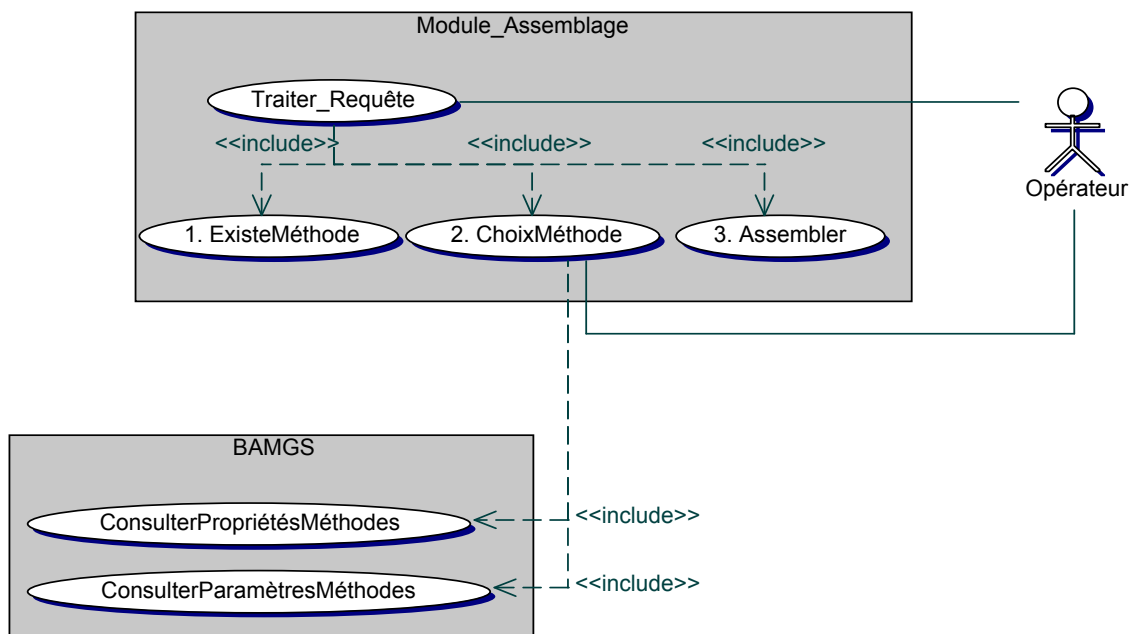


figure II.4-3. Choix d'une fonction de maillage

Pour que la phase de construction de maillage pré-optimisé soit automatique, il faut pouvoir entrer et manipuler les spécifications servant à l'AEM ainsi que les spécifications servant à la construction du maillage. L'architecture proposée dans le projet DIJA permet d'assister l'utilisateur dans la conception du produit, y compris dans l'entrée de ces différentes spécifications.

II.4.3 Le projet DIJA

Il est l'un des projets communs à toute l'équipe de recherche, et couvre différents domaines de recherche : la gestion des connaissances, le nommage des entités (géométriques ou non), le travail collaboratif, les interactions homme-machine, et bien sûr le maillage. Ce projet propose des nouveaux concepts pour assister les concepteurs dans leur tâche, dans le cadre du travail collaboratif. Ce dernier aspect n'est pas évoqué ici.

La philosophie générale du projet est présentée dans le premier paragraphe. Puis un second paragraphe montre comment DIJA pourrait tout à fait, supporter la pré-optimisation.

II.4.3.a Philosophie du projet

Le projet DIJA propose une architecture modulaire d'un système de CAO pour l'ingénierie collaborative. Elle est composée de plusieurs niveaux appelés également environnements. Chaque environnement correspond à un niveau d'abstraction de la conception, et comporte cinq modules (figure II.4-4) :

- un module historique : il conserve toutes les opérations relatives au modèle CAO (pas seulement la géométrie). Elles sont conservées sous la forme d'instructions liées entre elles au sein d'un langage. Chaque environnement possède son jeu d'instructions. Le langage associé est multi-niveaux et permet de désigner les entités du produit selon différentes vues (une par environnement).
- un module de base de connaissances qui fournit la méthode de conception relativement au niveau. Il s'agit d'un script ou d'un programme qui reflète le mode opératoire de conception relatif à un métier. Pour un niveau donné, il traduit les termes du langage de l'historique du niveau supérieur en termes de l'historique du niveau courant. Il s'aide des connaissances dont il dispose pour réaliser cette traduction. Le résultat de cette traduction constitue une spécification de l'objet qui est de plus en plus détaillée au fur et à mesure

que l'on descend dans les environnements jusqu'à obtenir la géométrie de l'objet dans l'environnement le plus bas.

- un module d'interactions homme-machine : il fournit les outils de conception adaptés au contexte en fonction du niveau d'abstraction (c'est-à-dire de l'environnement).
- un module de visualisation : il se charge de montrer à l'utilisateur les différentes informations sur le modèle en fonction du niveau (la géométrie, les outils, les bibliothèques disponibles, etc.) et du contexte (l'instruction en cours de paramétrage).
- un module superviseur : son rôle est de gérer les flux d'informations entre les modules et en particulier d'exécuter les scripts (c'est-à-dire les modes opératoires) donnés par le module de base de connaissances.

Il y a cinq environnements présentés du plus abstrait au plus concret. L'environnement application est le niveau d'abstraction le plus élevé et comporte l'ensemble des métiers dont a besoin l'utilisateur. C'est à ce niveau que l'utilisateur va choisir les métiers qui l'intéressent et les outils de conception associés. L'environnement métier est relatif à un métier donné. Par exemple, à la requête « évider une pièce », la base de connaissances répondra de manière différente selon que la conception ait un rapport avec le domaine de l'usinage ou celui de la plasturgie. Dans l'environnement commun, la base de connaissances propose l'opération « évider une pièce » parce qu'elle est commune à plusieurs métiers. L'environnement dialogue s'appuie sur des éléments de dialogue (les arêtes, les faces, et d'autres courbes dans le solide) pour donner une esquisse de la géométrie qui sera enfin complètement définie dans l'environnement géométrique. Le modeleur géométrique est en fait une partie du module de base de connaissances.

L'architecture constitue, dans son ensemble, plus qu'un système de conception : elle intègre les informations fonctionnelles qui se précisent au fur et à mesure de l'analyse et assiste l'utilisateur en s'aidant de bibliothèques de formes et d'outils prédéfinis (présents dans la base de connaissances).

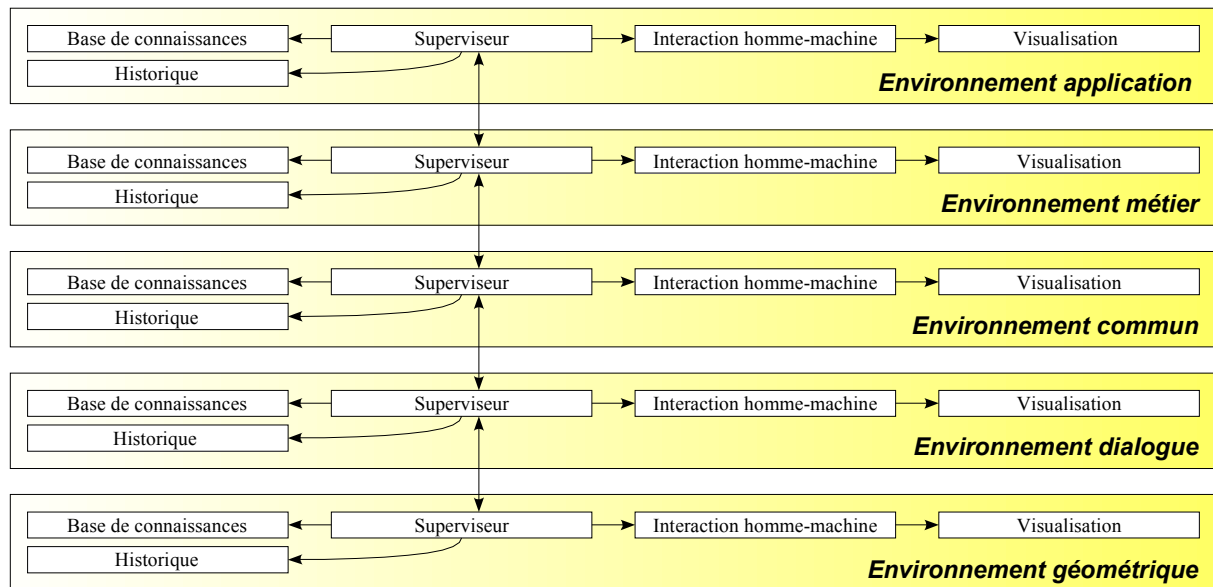


figure II.4-4. Architecture DIJA

II.4.3.b Spécifications des maillages avec DIJA

Notre étude s'est déroulée en parallèle de celles menées sur DIJA. À un certain stade, il est apparu que DIJA pourrait être d'un apport considérable dans la réalisation de notre architecture logicielle dédiée au maillage. Bien que DIJA ne soit pas encore complètement opérationnel, nous avons voulu approfondir cet aspect et les deux paragraphes à venir témoignent des réflexions que nous avons menées qui, bien que restant théoriques, confirment la potentialité de DIJA.

DIJA a pour rôle d'assister l'utilisateur dans sa tâche de spécification du produit, de sa géométrie et de tous les aspects afférents. En particulier pour valider certaines fonctionnalités, il faut faire appel à des AEM dont la préparation est plus ou moins complexe et plus ou moins longue.

La possibilité de traduire, à chaque niveau, une information (par exemple le terme « attaque » qui est du niveau métier) en informations de plus bas niveau (par exemple un « cône tronqué ») est tout à fait adaptée à la traduction des spécifications de l'AEM en spécifications de maillage. Dans l'exemple cité ci-dessus, le métier est celui de la fonderie dans le cadre de la conception d'un système de remplissage des moules. Au niveau métier, il faut associer au terme « fonderie », le terme « simulation de l'écoulement du métal dans le système de remplissage ». Cette contrainte est traduite au niveau commun en des termes de la physique et de la simulation à savoir dérivant qu'il s'agit d'un calcul par éléments finis, avec telle loi de comportement, pour tel phénomène physique. Ces contraintes sont utilisées pour la

préparation du maillage, et sont progressivement traduites en propriétés de maillage pour la BAMGS.

L'environnement dialogue permet de compléter les spécifications de maillage de manière très fine car dans cet environnement est conservé un Brep étendu, c'est-à-dire un Brep augmenté de courbes sur la surface du solide et à l'intérieur, courbes qui servent initialement d'éléments de dialogue pour la conception. Ceci permet à l'utilisateur de préciser la répartition des noeuds sur les éléments de dialogue si cette information n'a pas été donnée par les environnements supérieurs ou si le résultat ne satisfait pas l'utilisateur. En clair, cette approche multi-vues permet la spécification des données pour l'AEM *et* aide au choix de la fonction de maillage à invoquer.

Par exemple cette architecture permet de stocker la manière de traduire les spécifications générales, comme par exemple « simulation de l'écoulement d'un fluide » en spécifications directement exploitables : la méthode de maillage est celle de [BLA93], elle prend en paramètre un maillage quadrangulaire construit par la méthode de [VIS00], qui prend en paramètre une métrique calculée à partir des rayons de courbure principaux ([GEO97]). Le calcul de la métrique nécessite la connaissance de la surface qui doit être une nappe régulière de classe C^2 . Elle est caractérisée par un identifiant et son équation peut être obtenue par une fonction nommée *consulterEquation(Identifiant)* appartenant au modelleur géométrique.

Notons enfin un dernier avantage de l'architecture DIJA : elle offre un cadre idéal pour la réalisation de l'approche que nous proposons pour le maillage d'un objet, approche qui consiste à construire les maillages par parties pour les combiner ensuite. En effet, le langage de l'historique de construction permet de gérer et d'identifier les entités de la géométrie (des faces, des arêtes etc. ainsi que les éléments de dialogue) autant que les entités plus abstraites (« un cylindre » dans l'environnement commun, « une attaque » dans l'environnement métier lié à la fonderie). De manière générale, c'est la base de connaissances qui sait interpréter les termes du langage de l'historique du niveau courant (par les scripts et les programmes qu'elle fournit), et qui conserve également les fonctions du produit et les moyens de les vérifier. Elle stocke en particulier les spécifications de maillage pour chaque terme de l'historique de construction et également la manière de procéder pour mailler le solide.

II.4.3.c Utilisation de la BAMGS dans DIJA

Le scénario de construction du maillage est le suivant. Lorsque l'utilisateur souhaite valider une fonctionnalité du produit, un maillage adapté doit être créé. D'abord le module superviseur consulte la base de connaissances pour connaître la manière de procéder pour mailler le solide. Ce script correspond à l'algorithme d'assemblage dans lequel les différentes parties sont désignées par des termes de l'historique de construction. Il est exécuté par le superviseur. Pour un terme de l'historique de construction, l'architecture permet de retrouver les spécifications plus détaillées de maillage qui lui correspondent en interrogeant la base de connaissances à chaque niveau. Pour chaque terme de l'historique, c'est-à-dire chaque partie de l'objet, la méthode de maillage de la BAMGS qui lui est associée est invoquée. Les valeurs des paramètres sont soit déduites des spécifications pour l'AEM, soit données par l'utilisateur.

II.5 Transferts du maillage et gestion des modifications

Lorsque le maillage est construit, par une architecture DIJA ou par une autre, il faut le faire parvenir à l'AEM. Nous présentons ici les informations qui doivent transiter du système de conception vers l'AEM et inversement si l'AEM requiert une modification du maillage. Puis nous montrons quels moyens utiliser pour échanger ces informations.

II.5.1 Informations à transférer à l'AEM

Pour que l'AEM puisse réaliser son traitement, elle a besoin du maillage et, si possible, des spécifications sur le traitement qui lui incombe (par exemple les conditions aux limites pour un calcul par éléments finis), qui ont été soit fournies par la base de connaissances (ou déduites cf. DIJA), soit données par l'opérateur.

Il est possible qu'une AEM requière une modification du maillage. Par exemple suite à un premier calcul par éléments finis pour affiner la zone où se concentrent les contraintes ou lorsqu'une simplification du maillage est nécessaire à la visualisation. Ces modifications doivent être réalisées par la BAMGS. Or les fonctions de modifications de la BAMGS adressent des requêtes au modèle géométrique. Une solution maladroite serait donc de joindre le modèle géométrique au maillage envoyé à l'AEM, de manière à ce que cette dernière puisse à son tour fournir le modèle géométrique à la BAMGS quand elle utilise ses services. Nous proposons de joindre seulement les identifiants des entités géométriques impliquées dans le maillage; le nom du fichier natif conservant la totalité du modèle, ainsi que l'association entre les entités de maillage et les entités du modèle géométrique initial. Ceci

permet, d'une part de réduire l'information à transférer, et d'autre part d'assurer la confidentialité des données. Lorsqu'une modification est demandée, le système de conception charge à nouveau le modèle en mémoire, à partir du nom de fichier joint à la requête. Les identifiants servent à interroger le modèle géométrique, par exemple lorsque de nouveaux points doivent être créés ou lorsque des nœuds existants doivent être déplacés. Le fait d'envoyer des identifiants et le nom du fichier au lieu de tout le fichier CAO fait qu'on se réfère toujours au même logiciel et donc, que de multiples interprétations d'un même fichier par des logiciels différents est évité, et par là même, des erreurs.

II.5.2 Gestion des modifications

Une modification est toujours demandée par l'AEM. Comme pour la construction, il faut choisir une fonction de la BAMGS à invoquer pour réaliser cette modification. Les connaissances permettant de faire ce choix se situent au niveau du module d'assemblage et éventuellement au niveau de l'AEM. Il y a donc trois situations possibles selon que le choix est fait par l'AEM ou le système de conception (le module d'assemblage) et selon que la modification doit être effectuée en temps réel (par exemple pour un maillage de visualisation) ou non (par exemple un maillage pour une simulation).

Le premier cas est celui d'une AEM qui ne sait pas choisir la fonction adéquate dans la BAMGS et qui ne requiert pas une modification en temps réel. Alors, lors du transfert, le module d'assemblage fait parvenir à l'AEM, le nom du fichier natif contenant toutes les informations du produit ainsi que des propriétés génériques de la fonction de modification. Elles correspondent en fait aux principales familles d'algorithmes d'adaptation (raffiner, simplifier, transformer et lisser). L'AEM (avec éventuellement l'aide de l'opérateur) choisit le type de modification en sélectionnant la propriété adéquate, puis la retourne au module d'assemblage accompagnée des mailles concernées par la modification, les identifiants et le nom du fichier. Le module d'assemblage interagit avec la base de connaissances pour décider de l'algorithme de modification à appliquer et détermine également ses paramètres. Il invoque cette méthode de maillage auprès de la BAMGS qui lui retourne le maillage. Le module d'assemblage renvoie ce maillage à l'AEM.

Dans la deuxième configuration, l'AEM ne sait pas choisir l'algorithme mais elle a besoin de la modification rapidement. Elle demande alors auprès du module d'assemblage quelle fonction est adéquate pour la modification générique nécessaire. Le module d'assemblage choisit l'algorithme et envoie son identifiant à l'AEM, les paramètres, ainsi que les

identifiants nécessaires à la BAMGS (nom de fichier, identifiants des entités géométriques, etc.). L'AEM peut ensuite invoquer la fonction directement auprès de la BAMGS autant de fois que nécessaire avec les performances voulues.

Le troisième cas a lieu lorsque l'AEM dispose des connaissances expertes pour choisir l'algorithme de modification. Alors la situation est identique au deuxième cas à la différence près que le module d'assemblage ne décide pas complètement quelle fonction invoquer. En effet, il effectue une sélection de fonctions candidates à partir du type de modification requis et de la compatibilité entre les fonctions géométriques nécessaires et celles qui sont accessibles. L'AEM fait son choix parmi les fonctions sélectionnées et invoque la fonction directement auprès de la BAMGS.

Dans tous les cas, il est possible que les modifications de maillage de la BAMGS soient invoquées de manière asynchrone par rapport à la phase de conception. Ce qui signifie que le modèle n'est plus en mémoire et qu'il faut que le modelleur géométrique le recharge. L'AEM disposant du nom de fichier, il faut que le module d'assemblage le recharge lorsqu'il reçoit une requête de l'AEM. Remarquons également que, pour que les identifiants donnés par l'AEM au module d'assemblage après le chargement soient toujours valides, il faut que ceux-ci soient persistants.

Nous venons de voir comment sont gérées les modifications de maillage. Cependant, nous n'avons pas précisé par quel moyen ces modules communiquent.

II.5.3 Gestion des communications entre modules

Pour réaliser les requêtes entre les différents modules, il y a essentiellement trois possibilités.

- ❑ les appels de fonctions se font par utilisation d'une interface de programmation ou API¹.
- ❑ les appels se font à distance via un réseau
- ❑ les appels sont faits soit à distance, soit localement en faisant éventuellement migrer les codes des fonctions d'un site à un autre.

La première solution offre l'avantage d'être performante car tous les appels se font en local. Par contre cela ne permet pas de gérer les cas où l'AEM et le système de conception ne sont pas sur le même site.

¹ API : Application Programming Interface

En fait une seule application est créée et les modules doivent connaître les API des autres modules :

- la BAMGS appelle les fonctions de l'API du modelleur géométrique,
- le module d'assemblage appelle les fonctions de l'API de la BAMGS,
- l'AEM doit connaître l'API du module d'assemblage et éventuellement de la BAMGS.

Mais pour cela, il faut que les modules soient écrits dans des langages de programmation tel que l'accès aux API des autres modules soit possible et pas trop coûteuse. Par exemple l'interfaçage de C++ et Java est possible mais nécessite la conversion en profondeur de tous les objets, c'est-à-dire en tenant compte de l'imbrication des objets les uns dans les autres.

La deuxième solution permet de gérer des sites distants. Par contre, comme les appels se font systématiquement à distance lorsqu'il y a plusieurs sites, il est difficile d'assurer des performances temps réel.

Enfin la troisième solution semble la plus prometteuse puisqu'elle allie les avantages des deux solutions précédentes. Pour avoir des performances temps réel, le code de la fonction est téléchargé et exécuté en local. Pour émettre des requêtes sur un site distant, un appel de procédure distant est invoqué.

Dans tous les cas, pour remplacer tous les modules (modelleur géométrique, AEM, système de conception), par des logiciels existants, il est nécessaire d'accéder à leurs fonctionnalités via une API. De plus pour pouvoir faire des appels à distances, il faut un processus dédié à la communication dont le rôle est de scruter les requêtes qui arrivent et d'invoquer les méthodes de l'API correspondante.

Nous ne nous intéressons pas ici aux protocoles utilisés pour réaliser les appels à distance ou la migration dynamique du code. Notons cependant qu'il existe des outils pour réaliser les appels distants. Les plus connus sont Corba (Common Object Request Broker Architecture) de L'OMG ([**OMG01a**]), java RMI (Remote Method Invocation) de Sun, RPC (Remote Procedure Call) de Sun ([**SUN03**]), et DCOM (Distributed Component Object Model) de Microsoft. Mais seuls Corba et RPC sont indépendants de la plate-forme. Par ailleurs des travaux concernant la migration de code dans un environnement java sont proposés dans [**DEN03**] et font partie du projet DIJA.

Conclusion

Ce chapitre a recensé les défauts présents dans l'interfaçage des systèmes de conception et des AEM. Nous avons proposé une nouvelle architecture logicielle visant à réduire les interventions humaines dans les secteurs suivants : *la spécification du maillage* pour l'utilisation d'une AEM et *la correction des erreurs* présentes dans les modèles géométriques et les maillages.

L'architecture proposée tend à réduire les erreurs dans les modèles en évitant les interprétations multiples du modèle géométrique, qui sont la source de nombreuses erreurs. Evidemment des erreurs subsistent, en particulier celles provenant d'utilisations maladroites des modeleurs et contre lesquelles, seule une bonne connaissance des représentations internes de la géométrie est efficace.

La préparation des données pour l'utilisation d'une AEM peut être en partie automatisée sous la condition que les informations fonctionnelles soient accessibles lors de la phase de construction du maillage. Nous avons vu avec le projet DIJA, que cette automatisation ne concerne pas seulement la conception routinière, mais aussi la conception innovante. En effet, la décomposition fonctionnelle par niveaux procède par traduction des termes de l'historique de construction d'un niveau, en termes du niveau inférieur (moins abstraits). L'architecture permet de manipuler des termes métiers, mais aussi des termes communs à différents métiers, et des termes désignant les entités géométriques. Ainsi l'utilisateur est réellement assisté dans sa tâche, en accord avec son niveau d'expertise dans les métiers et les connaissances déjà présentes dans le système de conception.

Cependant, l'utilisation de cette architecture avec des systèmes de conception et des AEM existants nécessite des efforts de programmation importants. Le module d'assemblage par les accès à la base de connaissance, est dépendant du système de conception et par conséquent, il faut développer un module d'assemblage pour chaque système de conception. Cependant, le développement du module d'assemblage est un travail qui peut être capitalisé puisque, par exemple, la phase dédiée purement à l'assemblage est à peu près le même pour tous les BRep. Une interface virtuelle telle celles proposées dans [HAI98] et [TAU00], permettraient de factoriser cette partie du code. Les difficultés pour adapter le module d'assemblage d'un logiciel de conception à un autre viennent essentiellement de l'exploitation de l'information fonctionnelle, quand elle est disponible.

Un aspect nécessitant un développement important est la BAMGS. Une partie critique de cette architecture à laquelle nous nous sommes intéressés est l'intégration des algorithmes permettant de construire et modifier tous les types de maillages utilisés actuellement, au sein d'une architecture logicielle cohérente et extensible. La conception d'une telle bibliothèque nécessite une bonne connaissance des algorithmes de maillage qui ont été présentés dans le premier chapitre. Elle fait l'objet du chapitre 4.

Les opérations booléennes étant très présentes dans cette architecture, le chapitre suivant expose les problèmes algorithmiques qu'elles soulèvent

Chapitre III : Approfondissement des opérateurs d'assemblage

III.1 Objectifs et problématique

Nous avons expliqué dans le deuxième chapitre que le module d'assemblage est une composante du logiciel de CAO et qu'il a la responsabilité de mailler les solides décrits par le modèle géométrique. Pour cela, il maille des parties du solide à l'aide de fonctions de maillage, disponibles dans la BAMGS, et assemble ces parties à l'aide d'opérateurs dits d'assemblage, également fournis par la BAMGS. Ce troisième chapitre se concentre sur les opérateurs d'assemblage, à savoir le collage (et son symétrique, le découpage), le raffinement et les combinaisons booléennes. Ces trois opérateurs ont été présentés en 2.3 (chapitre 2, partie 3). Les deux premiers sont principalement utilisés par les modules d'assemblage destinés aux BRep, le troisième par les modules d'assemblage pour les modèles CSG.

Les quelques points algorithmiques présentant un intérêt dans les opérateurs de collage et de raffinement sont présentés dans la deuxième partie de ce chapitre.

La troisième partie porte sur les opérateurs booléens, dont la réalisation soulève principalement deux difficultés. La première est d'ordre numérique : les courbes d'intersection des deux opérands délimitent les parties à conserver sur chaque opérande. Dans notre cas, les opérands étant approchés (ce sont des maillages), leurs courbes d'intersection, en l'occurrence des polygones, sont d'autant plus entachées d'erreurs. De plus, les segments des polygones d'intersection servent à la construction de nouvelles mailles, construites dans la zone de raccordement des opérands. Or, pour un calcul par éléments finis, il est nécessaire que les nœuds soient aussi proches que possible des surfaces qu'ils

approchent, même si le maillage est grossier. Nous avons donc élaboré une stratégie pour améliorer la position des nœuds issus de l'intersection des maillages.

La seconde difficulté est davantage topologique : lors du calcul des intersections entre les deux maillages pendant une opération booléenne, les mailles qui sont coupées par l'autre maillage voient leurs propriétés (de forme, de taille, etc.) altérées. Par exemple, une maille triangulaire peut devenir un polygone de n cotés ($n > 3$) ou rester triangulaire mais être très aplatie. Or, les intersections et leur voisinage sont souvent des zones déterminantes pour les calculs par éléments finis car c'est souvent en ces lieux que se concentrent les contraintes. La qualité des mailles y est donc particulièrement importante. Nous avons essayé d'apporter une réponse à cette exigence dans le cas des maillages triangulaire en nous efforçant de développer une méthode qui produise des mailles satisfaisant à la fois deux critères : une variation de taille maîtrisée entre mailles adjacentes et des mailles de forme satisfaisante.

III.2 Opérateurs de collage, de découpage et de raffinement

Collage et découpage modifient les topologies de deux maillages de manière à, respectivement, les coller en fusionnant les entités communes (sommets, arêtes, faces) ou à les découper en dupliquant une partie des entités de dimension inférieure de manière à faire apparaître deux nouveaux maillage. Traditionnellement, la construction d'un maillage se fait à partir des maillages d'entités de dimensions inférieures (le maillage d'une face se base sur le maillage de ses arêtes, par exemple), ce qui laisserait penser que collage et découpage sont inutiles. Ces opérateurs servent cependant lorsqu'il faut mailler deux surfaces paramétriques voisines ou lorsque l'on souhaite découper un domaine en sous-domaines.

Ces opérateurs n'impliquent pas de difficultés majeures, l'algorithmique étant relativement simple. Pour le collage, la superposition des entités est vérifiée à partir de la distance qui sépare les sommets et de la similitude de leurs topologies. Si les sommets ne se superposent pas convenablement, l'opérateur fournit tout de même un maillage valide, même si le résultat n'est pas celui attendu. Comme les sommets ne sont presque jamais exactement superposés, l'utilisateur précise quel opérande est choisi pour la position des sommets du résultat final. Concernant le découpage, les entités formant la future séparation (les sommets, les arêtes ou les faces selon la dimension du maillage), doivent être spécifiées au préalable de manière manuelle.

L'opération de raffinement est, en elle-même, relativement simple car elle consiste simplement à raffiner deux zones, une sur chaque maillage, de manière à détecter les intersections ignorées lors d'un premier calcul d'intersection. Le degré de raffinement peut être contrôlé par la détection d'une intersection, par une erreur de corde, une taille de maille, le nombre de composantes connexes de l'intersection, en conjonction avec le nombre maximal d'itérations ou une limite de temps. La partie qui nous intéresse concerne le calcul des intersections qui est vue dans la partie suivante.

III.3 Opérations booléennes

III.3.1 Généralités

Les opérations booléennes sur les maillages tiennent une place importante dans l'organisation que nous proposons (système de conception et AEM). Comme l'a montré le chapitre 2, elles permettent de produire un maillage à partir d'un arbre CSG plutôt qu'à partir d'un BRep. Ainsi, on améliore la qualité géométrique des modèles transférés en évitant la construction d'un BRep intermédiaire et on permet la pré-optimisation des maillages en affectant des propriétés de maillage à des sous-arbres complets, plutôt qu'à des entités de bas niveau de type arêtes ou faces.

Nous avons développé une maquette pour mieux appréhender les problèmes algorithmiques inhérents à la combinaison de maillages et pour tester les solutions que nous proposons. Elle a été programmée en tout début de l'étude et ne vise ni à valider tous les concepts vus au chapitre précédent, ni à proposer une nouvelle méthode de combinaison de maillage.

Nous résumons dans ce qui suit les caractéristiques des principales méthodes d'opérations booléennes et résumons ce que nous avons mis en oeuvre dans le cadre de cette étude. Nous nous plaçons dans le cadre de la construction d'un maillage triangulaire pour un calcul par éléments finis. Par conséquent la taille et la forme des éléments sont importantes. Les maillages à combiner sont contraints par une taille de maille imposée sur les opérandes et par un facteur de qualité des triangles basé sur le rapport entre le rayon du cercle circonscrit et la longueur de sa plus petite arête. Pour le maillage, nous utilisons une structure de données qui conserve les surfaces et les courbes qui portent les entités du maillage. Dans la suite, cela permet entre autres de pouvoir créer de nouveaux points dans le maillage.

Les opérations booléennes sur les objets polyédriques ont déjà été largement étudiées ([GRA97], [LTH86], [MAN82], [HOF89], [HUB90], [PER95]), nous présentons ici les principales familles d'algorithmes. Elles suivent globalement le même schéma ([PER95]) :

1. Calcul des intersections entre les entités de la frontière de A et des entités de la frontière de B. Éclatement des entités.
2. Classification de chaque élément provenant de l'étape 1 par rapport à l'autre objet
3. Sélection, suivant l'opération booléenne, des éléments à retenir pour la composition de l'objet résultat
4. Fusionnement des différentes entités trouvées à l'étape 3 pour obtenir l'objet résultat.

Elles se distinguent par le type des entités (face ou arête) qui sont utilisées pour déterminer l'intersection des polyèdres. On distingue trois familles : les méthodes par découpage de faces, la méthode par faces, et la méthode par polygone de section.

La méthode par découpage de faces ([LTH86], [GRA97]) consiste à partager chaque face en sous-faces de manière à ce que chaque sous-face soit extérieure, intérieure, ou sur l'autre objet mais pas coupée par l'autre objet. Les sous-faces sont ensuite classées (intérieur, extérieur, sur) et seules celles qui doivent appartenir à l'objet résultat sont maintenues.

La méthode par faces ([HOF89]) consiste à calculer des intersections faces/faces de manière à ne conserver que les portions de faces intervenant dans la composition de l'objet résultat. Pour cela, le plan de chaque face de A fait l'objet d'une intersection avec les faces de B pour donner une section. Une opération booléenne en 2D est alors effectuée entre la section et la face de B, ce qui mène à une nouvelle section. Les sections obtenues sont ensuite accolées pour former l'objet résultat. Contrairement à la famille précédente, celle-ci n'engendre pas des faces supplémentaires.

La méthode du polygone de section ([MAN82], [MAT83]) consiste à classer les entités d'un solide (arêtes et faces) relativement à l'autre solide et inversement afin de déterminer les arêtes de l'un qui coupent les faces de l'autre et inversement. À partir de ces entités sont calculés des polygones d'intersection qui partagent chaque objet en plusieurs parties (lorsqu'il y a interpénétration des objets). Ces parties sont ensuite accolées selon l'opération (union, différence ou intersection).

La méthode que nous avons implémentée se classe dans la famille des algorithmes calculant le polygone de section, la construction de ces polygones se trouvant simplifiée par le fait que les faces sont convexes. La forme et la taille des triangles résultant des opérations booléennes peuvent être dégradées près des intersections. Comme notre approche exige une qualité en forme et en taille des triangles, un traitement complémentaire de remaillage est appliqué. La méthode comporte les étapes suivantes. Nous négligeons la présentation des cas particuliers bien que leur gestion ait nécessité un temps important de développement.

Premier classement des triangles

Les triangles d'un opérande qui n'appartiennent pas à l'englobant de l'autre opérande sont traités relativement à l'opération booléenne considérée. Ils sont ajoutés à l'objet résultat s'il s'agit d'une union, retirés s'il s'agit d'une intersection et pour une différence, ceux du premier opérande sont ajoutés, ceux du second sont retirés. Les triangles restants sont classés « indéterminés », et seront qualifiés après le calcul d'intersection.

Triangulation

La méthode que nous proposons peut être étendue aux maillages non triangulaires de surface. Si on souhaite réaliser cette extension, il faut, dans cette deuxième étape, trianguler chaque maille non triangulaire sans ajouter de points. Les mailles surfaciques non triangulaires sont le plus souvent des quadrangles qu'il suffit de scinder en deux triangles. Le but est de simplifier les calculs à venir, notamment le calcul d'intersection.

Calcul d'intersection

Dans un troisième temps, le calcul d'intersection, qui fait l'objet du paragraphe III.3.3 est effectué. À l'issue de cette tâche, les triangles coupés ne sont pas ajoutés au résultat, ce qui a pour effet d'isoler des composantes connexes de triangles de l'objet résultat provenant des deux opérandes (figure III.3-1). La topologie de la courbe d'intersection est construite à partir d'informations topologiques conservées durant le calcul d'intersection.

Second classement des triangles

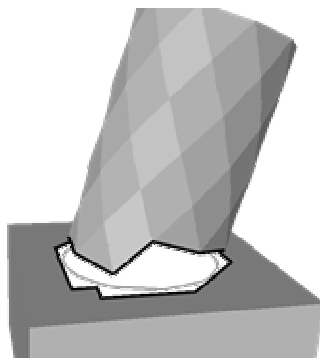
Parmi les triangles qui ne participent pas à l'intersection, il faut déterminer lesquels appartiennent à l'objet résultat et lesquels sont rejetés. Pour cela il faut distinguer ceux qui sont à l'intérieur de ceux qui sont à l'extérieur de l'autre maillage, quelle que soit l'opération

booléenne concernée (par exemple, pour la différence, on conserve les mailles du premier objet qui sont à l'extérieur du second et les mailles du second qui sont à l'intérieur du premier).

La technique utilisée consiste à classer un point de chaque composante connexe (intérieur ou extérieur) puis à propager l'information à tous les triangles de la composante connexe. La classification d'un sommet est effectuée soit par l'algorithme de la $\frac{1}{2}$ droite, soit par observation des informations géométriques de l'intersection (voir paragraphe III.3.2).

Remaillage

Les zones laissées vides (figure III.3-1) doivent être bouchées par de nouvelles mailles, qu'il faut construire entre les polygones d'intersection (en gris clair sur la figure) et les bords des groupes de mailles retenus à l'étape précédente (en gras sur la figure). Ces zones peuvent être relativement exiguës par rapport à la dimension des mailles ; de plus la différence de taille entre les mailles des polygones d'intersection (des segments) et celles des maillages auquel on les raccorde (des triangles) peut être importante. C'est pourquoi nous avons aussi cherché à contrôler le maillage reconstruit par la variation de taille des mailles. Cet aspect fait l'objet de la partie III.4



*figure III.3-1 Deux zones à remailler
(du polygone d'intersection vers 1) le cylindre et 2) la boîte)*

III.3.2 Test d'appartenance d'un point P à un solide triangulé

Il y a deux cas de figure pour une composante connexe : 1. elle a été obtenue en isolant un ensemble de triangles par intersection avec l'autre maillage, 2. il n'y a pas eu intersection des maillages (ils sont disjoints ou inclus l'un dans l'autre). Dans le premier cas, il existe un

sommet S de la composante qui appartient à une arête d'intersection A , qui coupe une face T de l'autre maillage. Par convention, l'arête est orientée de S vers T (figure III.3-2).

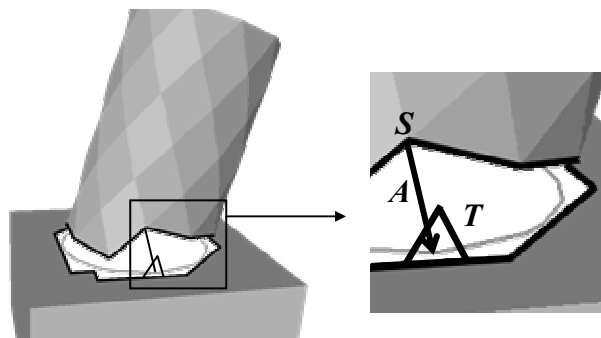


figure III.3-2. Classement d'une composante connexe de triangles

Si A entre dans le maillage, cela signifie que S est à l'extérieur, si A en sort, cela signifie que S est à l'intérieur. Habituellement ([LTH86], [HOF89] et [HUB90]), ce test repose sur le signe du produit scalaire entre le vecteur directeur de A et la normale à T (figure III.3-3). Nous préférons observer la position de S relativement au plan orienté¹ de la face (ou des faces) coupée(s). Les coordonnées de P sont injectées dans l'équation du plan afin de déterminer si P est du côté positif (A est entrante) ou du côté négatif du plan (A est sortante). Comme l'équation du plan est normalisée, ce test est effectué avec la tolérance utilisée pour la coïncidence de points.

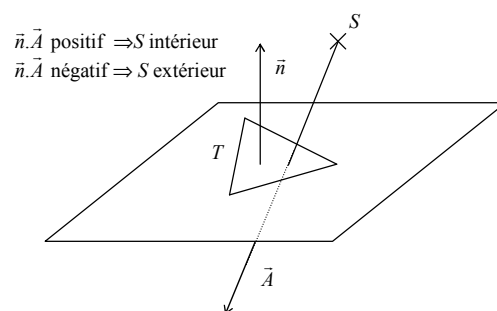


figure III.3-3. Position d'un point S par rapport à une face T

Les cas où A arrive sur un sommet et les cas de tangence de A avec des triangles sont plus difficiles à gérer et allongent le temps dédié au classement des points ([HOF89], [HUB90]). Dans ces cas, plutôt que de distinguer les différentes situations, on déplace légèrement son extrémité dans l'espoir d'arriver à une situation plus simple à traiter.

¹ Le plan orienté d'une face est tel que sa normale est dirigée vers l'extérieur.

Quand il n'y a pas d'intersection entre les maillages, on vérifie si un maillage est inclus dans l'autre en appliquant successivement la méthode de la demi-droite à un quelconque sommet du premier maillage, puis à un sommet du second. Elle consiste à lancer un rayon depuis le point à classer dans une direction arbitraire, puis à observer l'intersection du rayon avec la première face du maillage rencontrée (en l'absence d'intersection, le point est extérieur). Si le rayon entre dans l'objet, cela signifie que le point est à l'extérieur, si le rayon en sort, cela signifie que le point est à l'intérieur. Les mêmes raisonnements que précédemment sont appliqués pour mener ce test à bien.

Notons que le calcul de l'équation du plan d'un triangle n'est fiable que si ce dernier ne comporte pas d'angle trop petit (ce qui est le cas également pour le calcul de la normale à une face). Nous ne rencontrons pas ce genre de problème puisque les maillages ont été construits en respectant la forme des triangles. Cependant, si l'AEM est une fabrication par prototypage rapide, où les triangles ne sont pas forcément bien formés, ce type de calcul se trouve entaché d'erreurs importantes pouvant conduire à des incohérences dans le modèle géométrique. Dans ce cas, le maillage doit être préalablement raffiné ([RUP95], [RIV01]) afin de supprimer les angles trop aigus.

III.3.3 Calcul des polygones d'intersection

Dans ce paragraphe est présentée la réalisation du calcul des courbes d'intersection entre deux maillages triangulaires de solides. Etant donné la nature des solides, ces courbes sont polygonales.

Le calcul des courbes d'intersection se fait sur les entités classées « non déterminées » par le premier classement : les arêtes du premier maillage et les triangles de l'autre maillage, puis inversement. Les entités ayant servi au calcul d'un *point d'intersection* sont conservées au niveau du point d'intersection car elles permettent de réordonner les points dans un ordre cohérent en utilisant seulement de la topologie et, donc, d'exprimer les intersections comme des successions d'arêtes. On procède comme suit : les intersections arête/arête, arête/sommet et sommet/sommet sont détectées et la redondance de points d'intersection est supprimée. Par contre toutes les informations topologiques de chaque intersection sont conservées. Lorsque l'ensemble des points d'intersection est déterminé, les *arêtes d'intersection* sont reconstituées, une arête étant définie par deux points d'intersection ayant mis en jeu les mêmes triangles. Ces arêtes sont ensuite mises bout à bout si 1) elles partagent une de leur extrémités à une tolérance près et 2) si les points confondus (à une tolérance près) ont été calculés avec la

même arête ([MAN82]). Nous appelons *ligne d'intersection* un ensemble d'arêtes connexes obtenues par intersection des deux maillages et situées sur exactement deux surfaces. Sur la figure III.3-1, la ligne d'intersection est unique et fermée. Sur la figure III.3-4, il y a trois lignes (elles sont en traits plus gras que les autres et seules deux d'entre elles sont partiellement visibles) mais elles sont ouvertes. L'ensemble des lignes sera fermé dans une étape ultérieure.

Les lignes d'intersection sont orientées relativement à chaque opérande à l'aide des arêtes de chaque opérande conservées au niveau des points intersections.

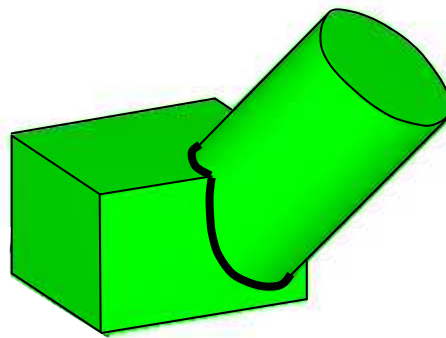


figure III.3-4. Lignes d'intersection non fermées

Les points d'intersection étant obtenus par un triangle et une arête, ils ne sont pas situés exactement sur la courbe d'intersection des deux surfaces. Si la précision peut être suffisante pour une visualisation, elle peut ne pas l'être pour un calcul par éléments finis, car il est souhaitable que les sommets soient sur les surfaces et les courbes qui portent le maillage, même si le maillage est grossier. De plus, il faut pouvoir ajouter et déplacer des points sur ces lignes d'intersection lorsqu'un raffinement ou un lissage (harmonisation des distances entre les sommets) doit être appliqué. L'idée générale justifiant l'étude des opérations booléennes sur les maillages repose sur le fait que :

- le calcul exact d'intersection des surfaces est parfois difficile, voire impossible et seules des intersections approchées peuvent être obtenues
- l'objectif est d'obtenir un maillage, qui, de toute manière, est une représentation approchée.

Par conséquent, nous cherchons à déterminer ces intersections non pas de manière exacte, mais de manière approchée avec un contrôle de l'erreur d'approximation des sommets, des arêtes et des faces.

L'erreur d'approximation d'un point d'intersection peut être évaluée en fonction des erreurs d'approximations des entités qui ont servi à son calcul. En effet, la plus petite distance d_{inter} d'un point d'intersection jusqu'à chacune des surfaces impliquées dans son calcul est majorée par le maximum des erreurs d'approximation e_{arete} de l'arête¹ et $e_{triangle}$ du triangle² mis en jeu, ajouté à l'erreur de calcul si elle peut être évaluée :

$$d_{inter} \leq \text{Max}(e_{arete}, e_{triangle}) + \varepsilon_{calcul}$$

(Cette remarque est vraie si l'arête et le triangle approchent partiellement la courbe d'intersection, c'est-à-dire si la courbe qu'approche l'arête, et la surface qu'approche le triangle se coupent. Si ce n'est pas le cas, c'est que, vraisemblablement, la taille des mailles n'est pas en adéquation avec la courbure).

De même les arêtes d'intersection correspondent à l'intersection de deux triangles, qu'il est aisé de retrouver à l'aide des informations topologiques conservées à leurs extrémités. Leur erreur d'approximation par rapport à chaque surface est majorée par le maximum des erreurs d'approximation des deux triangles. Par conséquent, pour que l'erreur d'approximation des points et des arêtes d'intersection soit acceptable, il faut que les triangles qui se coupent aient une erreur d'approximation inférieure à la tolérance géométrique. C'est pourquoi, avant le calcul d'intersection, les triangles susceptibles d'intervenir dans le calcul d'intersection sont raffinés jusqu'à obtenir une erreur d'approximation suffisamment faible.

Le schéma de raffinement est très simple : chaque triangle à raffiner est découpé en quatre triangles en scindant chaque arête en son milieu. Afin de réduire le raffinement de triangles non coupés par l'autre maillage, un filtre est fait sur les englobants des triangles. Ce schéma de raffinement ne se propage que si le triangle voisin est susceptible de faire partie de l'intersection et que son erreur d'approximation est trop importante, sinon il est simplement scindé en deux si cela n'induit pas d'angle trop petit.

¹ La plus grande distance entre l'arête et la courbe qu'elle approche

² La plus grande distance entre le triangle et la surface qu'il approche

Pour améliorer encore la précision, un déplacement des sommets sur les intersections est effectué. Il consiste à déplacer chaque sommet dans une certaine direction, d'une distance valant l'erreur de positionnement en ce sommet (en tenant compte de la position du point par rapport à la surface pour déterminer le sens de déplacement). La direction de déplacement est calculée par interpolation des normales à chaque surface aux sommets voisins. Cette méthode ne convient que si l'erreur de corde n'est pas trop importante, car les calculs de normales et de distance à la surface sont en général approchés (voir V.3.)

À ce stade, il est possible d'établir les bords des zones à remailler. Ils peuvent être composés de trois types d'arêtes. Celles qui proviennent d'une ligne d'intersection (les arêtes d'intersection), celles qui proviennent du bord du maillage des opérands et qui n'appartiennent qu'à un seul triangle (les *arêtes de bord*), et celles appartenant à deux surfaces d'un opérande (les *arêtes de frontière*). Les arêtes de frontière, lorsqu'il y en a, permettent de relier les arêtes d'intersection aux arêtes de bord (cas figure III.3-5). S'il n'y a pas d'arêtes de frontière, la zone à remailler est trouée (cas figure III.3-1).

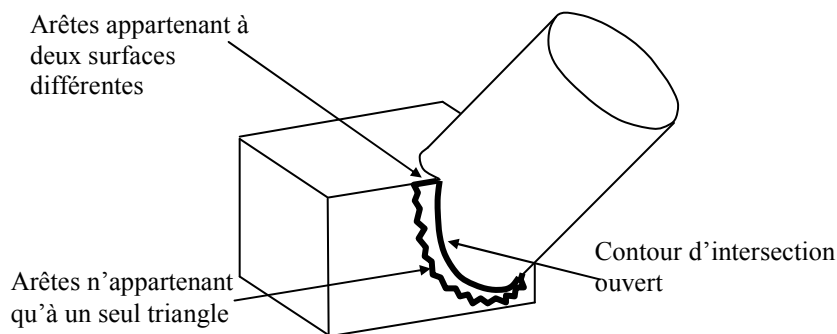


figure III.3-5. Construction du bord d'une zone à remailler

Le dernier traitement sur les lignes d'intersection est de les mailler pour rendre les écarts entre leurs nœuds compatibles avec les spécifications du maillage. En effet, à l'issue de la reconstitution des lignes d'intersection, le nombre de sommets et les écarts entre eux ne correspondent pas en général aux propriétés souhaitées, même si l'erreur d'approximation est maintenant contrôlée. Pour placer correctement les nœuds qui apparaîtront dans le maillage final, il faut une représentation adéquate des courbes d'intersection. Comme l'objectif est d'obtenir un modèle discret (un maillage) et qu'une représentation exacte n'est pas strictement nécessaire, et de toute façon pas toujours calculable, nous nous basons sur les lignes d'intersection pour construire les nouveaux points et pour déplacer les points lors d'une modification.

Les lignes d'intersection sont donc maillées en respectant les contraintes de taille imposées en créant les nouveaux points sur les arêtes, sans tenir compte de la position actuelle de leurs sommets.

Cette approche produit, lorsque la courbure de la surface est importante, beaucoup de triangles dont il faut calculer l'intersection. Pour ne pas aboutir à des résultats incohérents, le raffinement ne doit pas créer de triangles trop petits, c'est-à-dire des triangles dont une dimension est inférieure à la tolérance géométrique. Cependant l'avantage principal de cette méthode est que le calcul de l'intersection est possible dès lors que les solides des opérands peuvent être maillés et leur maillage raffiné avec un contrôle de la forme des triangles et de l'erreur d'approximation.

III.4 Remaillage

III.4.1 Contexte

Une approche naïve pour remailler les zones vides serait d'employer la même technique de maillage que celle utilisée pour la construction des maillages initiaux. Cependant, le maillage obtenu ne serait pas optimal car les mailles sur les maillages à raccorder peuvent avoir des tailles très différentes. De plus, les zones vides sont souvent exiguës et ne laissent pas toujours l'espace nécessaire au respect de certaines contraintes comme le respect conjugué de la taille et de la forme des éléments. Les éléments de plus mauvaise qualité au regard de la forme et de la taille se trouveraient dans les zones proches des contours d'intersection alors que nous avons dit que c'est précisément là que le maillage doit être particulièrement soigné. En conséquence, nous nous sommes efforcés de concevoir une méthode de remaillage dont le comportement serait le suivant :

- Quand la zone à remailler est bordée par un seul contour (c'est le cas de la zone à remailler de la figure III.3-1), on crée des triangles en partant de la portion correspondant au contour d'intersection ouvert et en se dirigeant vers le maillage existant (c'est à dire vers les arêtes n'appartenant qu'à un seul triangle). Quand la zone à remailler est bordée par deux contours (c'est le cas des zones à remailler de la figure III.3-5), on crée des triangles depuis le contour correspondant à la courbe d'intersection vers l'autre contour, qui correspond à un bord d'un maillage ouvert (sur la figure III.3-5, ceci signifie que l'on crée des triangles du contour d'intersection vers le cylindre, puis d'autres triangles de ce même contour d'intersection vers la couronne dentelée qui perce le carré).

- Les triangles construits doivent avoir une forme satisfaisante et une taille compatible avec les triangles adjacents (qualité en forme et maîtrise de la variation de taille).
- Comme les zones à remailler sont exiguës, il est souvent difficile de créer des triangles dont les tailles passent progressivement de la taille sur l'un des bords à celle du bord en vis-à-vis. Nous avons donc imaginé d'agrandir les zones à remailler en supprimant davantage de mailles dans la direction où l'on se dirige. Cette dernière contrainte nécessite de définir sur quelle distance il faut démailler de manière à ce que l'espace vide soit suffisamment grand pour que les contraintes sur les triangles puissent être respectées.

Le paragraphe qui suit présente un état de l'art sur les méthodes de construction et d'adaptation de maillage susceptibles de répondre à ces attentes.

III.4.2 État de l'art

Une première approche consiste à adapter la taille des mailles à l'espace disponible. [RUP95] définit pour cela la *taille caractéristique locale* (*local feature size* ou *lfs*) pour tout point P du domaine à mailler, comme le rayon du plus petit cercle centré en P et coupé par deux entités géométriques non incidentes entre elles (sommets ou segments) (figure III.3-6).

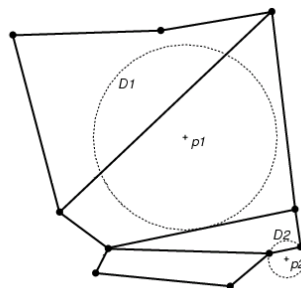


figure III.3-6. Taille caractéristique locale pour un point p_1 tangent à une arête et un point p_2 tangent à 2 points

Cela revient à établir une métrique construite à partir de l'espace disponible dans le but de maximiser la qualité en forme des triangles, quitte à ce qu'ils soient plus petits. La création de nouveaux points est faite au centre des cercles circonscrits des triangles comportant un angle trop petit ou ayant une taille supérieure à celle donnée par la métrique. L'insertion des points est faite à chaque création de point par le noyau de Delaunay. L'intérêt de cette méthode est qu'elle exploite le critère de Delaunay de manière à garantir la qualité en forme des triangles. Le maillage obtenu est une triangulation de Delaunay Contrainte (cf. chapitre 1). De plus, il est possible de majorer la taille des triangles par une autre métrique ou par un calcul par

éléments finis précédent. La variation de taille n'est pas prise en compte explicitement, et le maillage existant ne peut pas être modifié pour étendre la zone à remailler.

Les travaux de Ruppert ont été étendus par [OLL01] pour prendre en compte deux paramètres supplémentaires : un facteur d'échelle R et un facteur d'agrandissement G . Le premier paramètre consiste à réduire la taille préconisée par la taille caractéristique locale (lfs) en un point P , en la divisant par R . Théoriquement, on pourrait également l'utiliser pour augmenter la taille, mais la qualité en forme ne serait alors plus garantie. Le facteur d'agrandissement est utilisé pour s'assurer que les triangles ne grandissent pas plus que linéairement par rapport à la distance. Ainsi, en définitive, pour un point quelconque donné, la métrique renvoie la taille minimum entre :

- la taille caractéristique locale éventuellement réduite par le facteur d'échelle,
- et le minimum des écarts entre nœuds assignés aux sommets les plus proches, multipliés par $\frac{1}{G}$.

Ce qui s'écrit :

$$LS(p) = \min \left(\frac{lfs(p)}{R}, \min_{voisins, q_i} \left(LS(q_i) + \frac{1}{G} |q_i - p| \right) \right)$$

La stratégie de création de points est la même que dans la méthode précédente, à savoir au centre des cercles circonscrits aux triangles ne respectant pas les critères de forme et de taille.

Cette approche est évidemment plus intéressante car elle permet de contrôler la variation de taille explicitement et de garantir la qualité en forme des triangles. Cependant, pour cela, elle procède par raffinement, pouvant créer des triangles de très petite taille. De plus, elle n'offre pas le moyen de déterminer les triangles qu'il serait peut être judicieux de supprimer dans le maillage déjà existant. De même les sommets bordant le maillage existant, apparaissent obligatoirement dans le maillage final, même si leur position n'est pas optimale. Par contre, il est possible d'éviter la présence non nécessaire des arêtes de la frontière en autorisant l'application du noyau de Delaunay au voisinage de cette frontière.

La troisième approche ([OWE97]) consiste à contrôler la taille des triangles par une interpolation des écarts nodaux imposés sur la frontière ainsi que par deux paramètres. Le premier paramètre est une distance définissant une épaisseur le long de la frontière où la taille des éléments est constante afin d'y placer des éléments de très bonne qualité. Le deuxième

paramètre est une distance définissant une épaisseur le long de la première zone consistant en une zone de transition dans laquelle les triangles vont en grandissant ou en rapetissant. L'écart nodal à l'intérieur de la zone de transition est calculé par interpolation des écarts nodaux sur la frontière. Cette approche est intéressante dans le sens où une zone le long de la frontière est protégée de l'agrandissement. L'application de cette méthode peut être intéressante si la taille imposée sur le contour d'intersection est faible par rapport à celle du maillage existant par ce que la frontière est protégée d'une forte variation de taille. Par contre elle est d'autant plus importante dans la zone de transition. De plus, les arêtes de la frontière apparaissent obligatoirement dans le maillage final, ce qui n'est pas toujours souhaitable. Cette solution reste insuffisante parce que les frontières sont figées et parce que la variation de taille n'est pas un paramètre de contrôle, mais une conséquence des écarts nodaux sur la frontière.

La méthode qui semble la plus adaptée parmi les trois est celle d'Ollivier-Gooch ([OLL01]) car elle permet un contrôle explicite de l'agrandissement et de la forme des triangles. La difficulté de notre situation vient du fait que le maillage est contraint par une frontière dont une partie pourrait être supprimée. Par conséquent une telle méthode n'est pas applicable telle quelle. En effet bien que le noyau de Delaunay permette d'insérer un point dans une triangulation en fournissant des garanties quant à la forme des triangles produits, rien ne permet de dire que les sommets du maillage préexistant soient placés de manière pertinente. Or la méthode de Delaunay remanie la topologie du maillage mais pas sa géométrie. Autrement dit, aucun sommet n'est supprimé ou déplacé.

[SHI98] propose une solution consistant justement à adapter les sommets par suppression, création et déplacement. Les sommets du maillage sont assimilés à des particules pouvant interagir les unes avec les autres par des forces d'attraction/répulsion (voir chap. 1). A partir d'une métrique, qui fournit la taille souhaitée en tout point de la zone à mailler, une population de particules est construite : des particules sphériques sont créées aux sommets avec un rayon déterminé par la métrique. Ensuite, les arêtes sont remplies de particules par subdivisions récursives jusqu'à ce qu'il n'y ait aucun vide. Puis un principe similaire est appliqué pour remplir l'intérieur des faces. Un état d'équilibre des forces est recherché. Le nuage de points composés des centres des particules est ensuite maillé par une méthode de Delaunay contrainte. Il s'avère que cette méthode fournit en général des maillages de très bonne qualité. Cependant, la recherche de l'état d'équilibre reste coûteuse en temps et un remaillage du nuage de sommets après l'obtention de l'état d'équilibre est nécessaire. De

plus, la recherche de l'équilibre ne tient pas compte de la variation de taille, mais s'appuie sur une métrique de taille. Il n'y a pas de contrôle de la variation de taille à moins que la métrique ait été conçue en tenant compte de la variation mais [SHI98] ne donne aucune indication sur la manière d'y arriver.

Nous avons cherché à aller plus loin en cherchant à contrôler explicitement la variation de taille tout en produisant des triangles de forme acceptable. Nous avons exploré différentes voies que nous allons présenter maintenant.

III.4.3 TARF (Triangulation par Avancements et Reculs de Fronts)

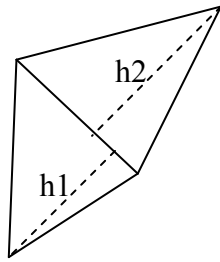
III.4.3.a Objectif

Nous rappelons que l'objectif est de contrôler par une variation de taille la construction d'un maillage triangulaire entre un contour d'intersection et le bord d'un maillage déjà existant, étant entendu que la qualité en forme doit rester assurée. Si nécessaire, on s'autorise de supprimer des mailles sur le maillage qu'on doit atteindre pour augmenter la distance sur laquelle on amortit la différence de taille entre la ligne de départ et la ligne d'arrivée.

Comme on ne sait pas combien de triangles il est nécessaire de supprimer pour mieux absorber la variation de taille, nous souhaitons réaliser la suppression de triangles et, si nécessaire, de sommets, au fur et à mesure des besoins. Toutefois, les arêtes appartenant à deux surfaces, qualifiées de *frontières* (voir III.3.3), ne peuvent être détruites car leur suppression impliquerait la perte de la limite entre deux faces et mènerait à une détérioration significative de la forme du solide maillé. On ne peut donc démailler au-delà de la frontière naturelle d'une face.

La première approche que nous avons tentée s'appuie sur une technique d'avancement de front (cf. Chapitre 1), avec cette nuance qu'il arrive à une partie du front de reculer, d'où le nom de la méthode : Triangulation par Avancement et Recul de Front (TARF).

Pour pouvoir mesurer la variation de taille, il en faut une définition précise. Nous posons que la variation de taille de deux triangles est le rapport de leur aire, ce qui équivaut, s'ils partagent une arête, au rapport de leurs hauteurs en leur arête commune (figure III.3-7).



$$\begin{aligned}
 \text{Variation de taille} &= \text{aire1} / \text{aire2} \\
 &= [(base \times h1)/2] / [(base \times h2)/2] \\
 &= h1 / h2
 \end{aligned}$$

figure III.3-7 : la variation de taille de deux triangles

III.4.3.b Principe

Le front initial est composé des arêtes qui limitent la zone à mailler. Il y a deux sortes d'arêtes : celles qui sont *fixes* et celles qui ne le sont pas qu'on appellera *flottantes*. Les arêtes flottantes sont susceptibles d'être supprimées si l'espace à mailler est trop exigu pour que la variation de taille soit respectée. Les arêtes flottantes sont celles du bord du maillage à « atteindre ». La disparition d'une arête flottante entraîne celle du triangle qui l'utilise. C'est ainsi que la zone à mailler est étendue. Initialement, les arêtes fixes comprennent les arêtes du contour d'intersection et les arêtes de frontière. Cela donne un front composée de deux parties : l'une composée d'arêtes fixes et qui avance et une autre composée d'arêtes flottantes et qui éventuellement recule. L'avancement et le recul de front visent à réduire l'écart entre les tailles sur les deux parties du front.

Comme nous l'avons dit, la technique visée respecte le schéma des méthodes par avancement de fronts. A chaque étape de l'algorithme (algorithme III.3-1), un segment $S_i S_{i+1}$ du front est choisi. Nous appliquons ensuite une heuristique basée sur la variation de taille maximale autorisée pour décider si, localement, l'espace entre les fronts est suffisant.

Cette heuristique s'appuie sur la distance qui sépare les fronts, la variation de taille maximale autorisée, l'écart entre nœuds présent localement sur le segment de départ (sa longueur) et l'écart entre nœuds sur le front vers lequel on se dirige. La distance entre les fronts est la distance entre le milieu I de $[S_i S_{i+1}]$ et son projeté J sur le reste du front perpendiculairement à $[S_i S_{i+1}]$. Appelons $[P_i P_{i+1}]$ le segment auquel appartient J , et m , la médiatrice de $[S_i S_{i+1}]$. On convient que l'écart entre nœuds sur le front d'arrivée est la longueur de $[P_i P_{i+1}]$.

On calcule alors deux grandeurs : 1. d'abord la hauteur du triangle idéal basé sur $S_i S_{i+1}$. Le triangle idéal est le triangle équilatéral dont la hauteur est agrandie ou réduite en fonction à la fois de l'espace entre les fronts, et du rapport des écarts entre nœuds sur le segment de départ

($[S_i S_{i+1}]$) et sur celui d'arrivée ($[P_i P_{i+1}]$). Puis 2. le nombre d'agrandissements successifs, le long de m , qu'il serait nécessaire d'appliquer pour atteindre l'écart entre nœuds de $P_i P_{i+1}$. Ces agrandissements sont représentés sur la figure III.3-8 par les points Q_1 , Q_2 et Q_3 . On cherche ainsi à savoir combien de triangles agrandis successivement il faudrait mettre bout à bout pour obtenir l'écart entre nœuds imposé par la variation de taille. L'objectif est de déterminer au plus tôt l'impossibilité de mailler en respectant cette variation de taille.

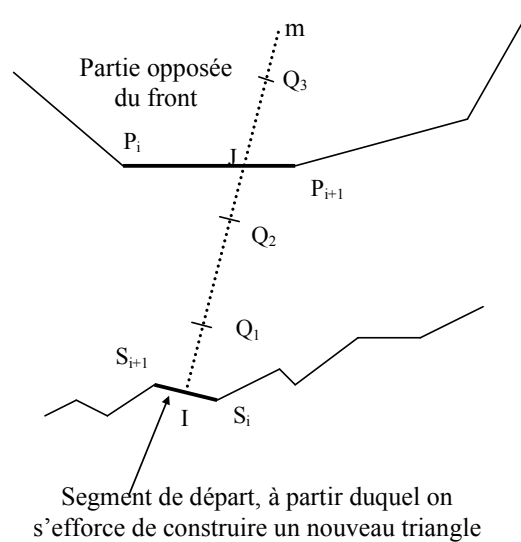


figure III.3-8. Evaluation de la distance entre les parties du front

Si le point Q_j à partir duquel l'écart entre nœuds est suffisamment proche de celui en $P_i P_{i+1}$ se situe au-delà de $[P_i P_{i+1}]$, alors on considère que l'espace disponible n'est pas suffisant. Dans ce cas, si l'arête gênante $[P_i P_{i+1}]$ est flottante, elle est supprimée, sinon un point est inséré en son milieu et ce, récursivement jusqu'à atteindre un écart entre nœuds inférieur à celui imposé par la variation de taille.

Soit $\Gamma = \{a_i\}_{1 \leq i \leq n}$ l'ensemble des arêtes des fronts. On nommera S_i et S_{i+1} les extrémités d'une arête a_i . La méthode itère sur la construction d'un nouveau triangle s'appuyant sur une arête fixe. Soit Ω , l'ensemble des arêtes fixes, Δ_m la plus grande variation de taille autorisée entre deux triangles adjacents et ε , la tolérance géométrique.

Algorithme

```

Tant que  $\Gamma \neq \{\}$  Faire
  Choix d'un segment  $S$  dans  $\Omega$ 
  Construction impossible = Test ( $S, \Gamma, \Delta_m$ )
  Si construction impossible Alors
    Supprimer la plus proche arête flottante sur la médiatrice de  $S$  jusqu'à ce que la
    construction soit possible ou qu'il n'y ait plus d'arêtes flottantes sur cette médiatrice
  Fsi
  Si construction impossible Alors
    Si agrandissement Alors
      Ajouter des nœuds sur l'arête d'arrivée
    Sinon
      Ajouter des nœuds sur l'arête de départ
  Fsi
  Mettre  $\Gamma$  à jour
Fsi
Appliquer la méthode frontale c'est-à-dire construire un triangle de base  $S$  :
  Construire un point  $Q$  de manière à former un triangle idéal avec  $S$ 
  Scruter le voisinage  $V_Q^\rho$  de rayon  $\rho > \varepsilon$ .
  Si il existe dans  $V_Q^\rho$  des nœuds permettant de construire avec  $S$  des triangles sans
  intersection avec les fronts Alors
    Prendre le meilleur
  Sinon
    Tester l'intersection du triangle idéal avec les fronts
    Si Non Intersection Alors
      Prendre le triangle idéal tel quel
    Sinon
      Chercher un autre triangle (selon les cas d'intersection, soit on
      ajoute un sommet à l'arête coupée, soit on diminue
      successivement le triangle idéal jusqu'à ce qu'il n'y ait plus
      d'intersection)
  Fsi
  Mettre à jour les fronts (Retirer  $S$  de  $\Gamma$  mais y ajouter les deux autres arêtes du
  triangle en tant qu'arêtes fixes)
Ftant
Fin

```

algorithme III.3-1 Triangulation par avancement et recul de fronts

III.4.3.c Convergence de la méthode

La convergence de la méthode est assurée par deux éléments principaux : la suppression de triangles est limitée aux triangles portés par la même surface et chaque étape de l'algorithme produit un nouveau triangle. Pour la preuve de la convergence, deux suites sont considérées : l'une mesure l'aire recouverte par la construction de triangles, et l'autre mesure l'aire découverte par la suppression de triangles. Les deux processus : construction et destruction de triangles, ne s'effectuent jamais sur les mêmes triangles. Ainsi aucune oscillation n'est possible. De plus, aucun triangle dégénéré (plat, inversé ou coupant un élément du maillage) n'est construit. Par conséquent, la suite calculant l'aire restant à mailler est convergente (voir annexe C).

III.4.3.d Critique

Une des difficultés, dans cette approche, est de construire à chaque pas, un nouveau triangle avec l'agrandissement nécessaire et suffisant. En effet, après avoir calculé quel serait le triangle idéal au regard de la configuration locale, le voisinage de ce qui serait le troisième sommet de ce triangle est observé et un sommet voisin est éventuellement choisi pour le remplacer. Le critère de choix d'un sommet voisin est difficile à établir car en général, il faut faire un compromis entre être restrictif et produire de très bons éléments mais malheureusement aussi de très mauvais (l'explication suit), ou être plus permissif mais alors la contrainte de forme et la variation de taille peuvent ne plus être respectées, ce qui produit aussi des éléments mauvais. Cet aspect est caractéristique des méthodes par avancements de fronts. Lorsque le voisinage est petit, des éléments de forme idéale sont construits sans prendre en compte leur voisinage, ce qui crée parfois des vides avec des angles très petits. Dans la plupart des cas, le rayon choisi vaut l'écart entre nœuds imposé localement ou à défaut, la longueur du segment sur lequel est basé la construction du nouveau triangle.

En conséquence, cette approche ne permet pas de donner des garanties ni sur la forme des triangles, ni sur le respect de la variation de taille. Malheureusement, utiliser le noyau de Delaunay pour insérer un nœud après l'avoir construit par cette méthode ne résout pas le problème car le noyau de Delaunay ne prend pas en compte l'agrandissement. Les triangles produits auraient une qualité en forme satisfaisante, mais rien ne garantirait le respect de la variation de taille.

Dans les deux cas, c'est-à-dire avec un avancement et recul de front utilisant ou n'utilisant pas le critère de Delaunay, l'amélioration de l'un des deux critères (variation de taille et qualité en forme) se fait parfois au détriment de l'autre.

Il peut arriver (voir la ligne désignée par une flèche, algorithme III.3-1) qu'il faille créer de nouveaux nœuds sur les arêtes de frontière. Ceci est gênant pour plusieurs raisons. D'abord il est difficile de déterminer la meilleure position des nœuds à créer sur ces arêtes. Ensuite cela implique de modifier le maillage au-delà des arêtes de frontière s'il en existe un : en effet, pour insérer un nœud sur une arête de frontière, nous employons la méthode de [RIV01], qui propage l'insertion de nœuds à travers le maillage existant afin de maintenir le critère de Delaunay. Là encore, cette insertion ne tient pas compte de la variation de taille. L'insertion de nœuds sur les arêtes de la frontière peut donc dégrader le maillage au-delà de la face en cours de traitement.

III.4.4 Autres approches du remaillage des zones vides

Les faiblesses de la TARF viennent principalement de l'heuristique choisie pour placer le sommet qui sert à construire un nouveau triangle. Nous nous efforçons d'améliorer ce point. Pour cela, nous cherchons à caractériser le lieu géométrique où doit se trouver un sommet pour former avec une arête du front un triangle satisfaisant les deux critères de qualité. Ce lieu est nommé dans la suite *zone caractéristique* d'une arête. La zone caractéristique d'une arête prend en compte l'agrandissement par rapport au triangle qui est déjà attaché à cette arête ainsi que la qualité en forme du triangle à construire, mesurée par le plus petit de ses angles. Les paragraphes qui suivent donnent les premiers éléments de cette deuxième approche du remaillage. Nous trouvons qu'elle mérite d'être approfondie mais n'avons pas eu le temps d'en finir l'analyse. Le premier paragraphe définit la zone caractéristique d'une arête, le deuxième donne des pistes pour réaliser un pré-traitement des arêtes limitant la zone à remailler en vue d'une construction de maillage. Le troisième porte sur la difficulté de déterminer une stratégie de création de point prenant en compte les deux critères de qualité.

III.4.4.a Zone caractéristique d'une arête

Dans les illustrations qui suivent, la variation de taille maximale est de 1,5 : le triangle voisin d'un triangle existant peut donc être plus grand ou plus petit de 50 %. La zone caractéristique d'une arête est définie par l'intersection de deux zones : une pour chaque critère de qualité.

Considérons le triangle (ABD) associée à l'arête AB (figure III.3-9), et soit C un point de la zone caractéristique de l'arête AB . Nous cherchons à caractériser C .

Soit (CH) la hauteur de (ABC) issue de C et (DJ) , la hauteur de (ABD) issue de D . La variation de taille maximale Δ_m , avec $\Delta_m > 1$ est décrite par le rapport des hauteurs de deux éléments voisins. Par conséquent, la zone caractéristique est limitée par deux contraintes $CI < \Delta_m \cdot DJ$ et $CH < (1/\Delta_m) \cdot DJ$, c'est-à-dire qu'elle s'inscrit dans une bande limitée par deux droites parallèles à l'arête.

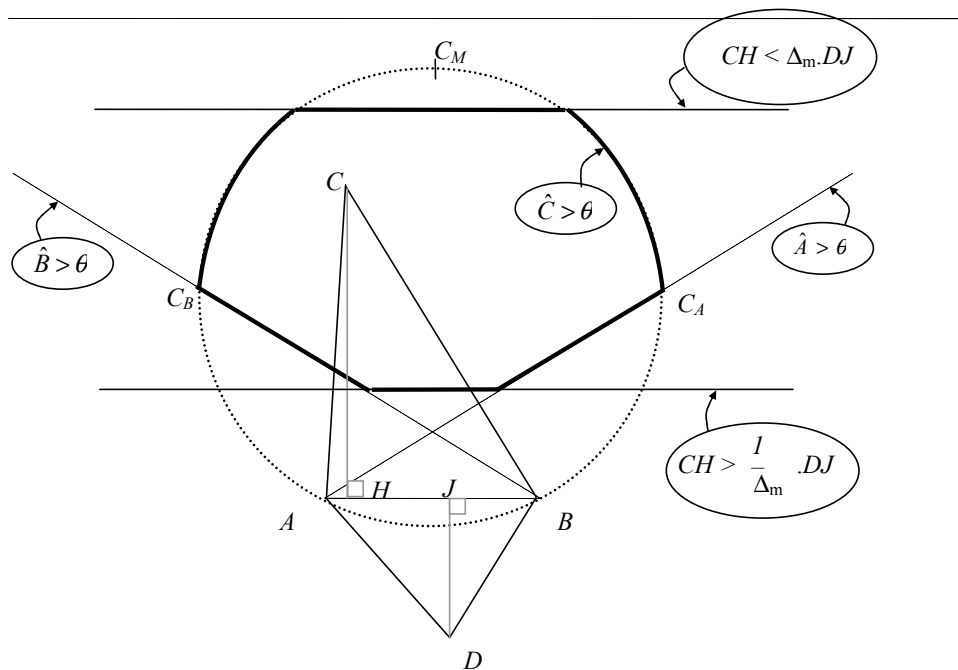


figure III.3-9. Zone caractéristique d'une arête (en gras)

La qualité en forme est décrite par la mesure du plus petit angle du triangle. Appelons θ , le seuil du plus petit angle en deçà duquel le triangle est considéré comme étant mal formé¹. Caractérisons la zone dans laquelle doit se trouver le point C pour que le triangle (ABC) soit bien formé. Premièrement, l'arête (AC) ne doit pas faire un angle inférieur à θ avec AB , donc le point C se situe « au-dessus » de la droite définie par la contrainte $\angle ABC > \theta$ (on note $\angle XBC$, la mesure de l'angle au point X dans le triangle ABC). Symétriquement, il se situe au-dessus de la droite définie par la contrainte $\angle BCA > \theta$. Enfin le fait que l'angle $\angle BCA$ ne peut pas être inférieur à θ contraint C à se trouver dans le cercle passant par les trois points suivants (preuve en annexe D) :

- C_A qui est la position de C lorsque C est sur la droite de la contrainte $\angle ABC = \theta$,
- C_B qui est la position de C lorsque C est sur la droite de la contrainte $\angle BCA = \theta$,
- C_M qui est la position de C lorsque le triangle ABC est isocèle en C .

On retient de ce qui précède que si C se trouve dans le contour défini par la combinaison des contraintes de taille et de forme (en gras sur la figure III.3-9), le triangle ABC possède une

¹ Les méthodes de Ruppert ([RUP95]), Shewchuk ([SHW02]), et [OLL01] permettent d'obtenir des triangulations ayant une valeur d'angle minimale de 30° , voire moins (jusqu'à 26°).

qualité en forme satisfaisante et une qualité en taille *relativement au triangle adjacent* également satisfaisante.

Notons que la relation d'appartenance d'un sommet à la zone caractéristique d'une arête est par définition symétrique pour deux triangles partageant une arête (figure III.3-10, (a)) : si deux triangles adjacents ont des tailles compatibles avec la variation maximale, alors le sommet non partagé du premier triangle est dans la zone caractéristique de l'arête partagée et il en est de même pour le sommet non partagé du deuxième triangle. Par contre, quand deux arêtes partagent un sommet, il se peut que la zone caractéristique de la première arête contienne le sommet non partagé de l'autre arête sans que la zone caractéristique de la deuxième arête contienne le sommet non partagé de la première (figure III.3-10, (b)). Une même absence de symétrie peut être constatée avec deux arêtes se faisant face (figure III.3-10, (c)).

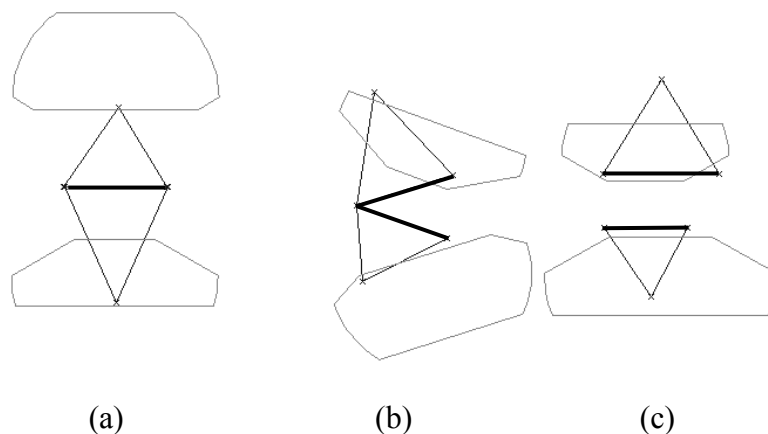


figure III.3-10. Symétrie de la relation pour deux triangles adjacents (a), non symétrie de la relation pour deux arêtes voisines (b), de deux arêtes quelconques (c)

Notre intention est d'exploiter les zones caractéristiques pour détecter au plus tôt l'impossibilité de construire de bons triangles à partir de certaines arêtes du front et aussi de combiner les zones caractéristiques de plusieurs arêtes pour construire des triangles dont les tailles ne sont pas compatibles qu'avec un seul des triangles adjacents : le fait de raisonner moins localement devrait augmenter les chances de produire un bon maillage.

III.4.4.b Problème de la création de points

Exploiter les zones caractéristiques pour construire un maillage satisfaisant les critères de forme et de variation de taille est une tâche ardue. Notre intention était de mieux cerner les endroits où il serait judicieux d'insérer un sommet, ce qui permettrait de créer un triangle avec

une arête fixe du front. Malheureusement, c'est un problème que nous n'avons abordé qu'à la fin du doctorat (ce que le plan de ce document ne suggère pas car il n'est pas chronologique) et que nous n'avons pas pu explorer, faute de temps, au-delà des quelques remarques qui suivent.

Soit la situation suivante : imaginons qu'on veuille construire un point à l'intérieur du polygone de la figure III.3-11 (a). L'idée venant naturellement est de vérifier si les zones caractéristiques des arêtes ont une partie commune. Sur l'exemple, c'est le cas puisque la zone grisée (figure III.3-11 (b)) appartient aux zones caractéristiques de *toutes* les arêtes. Par conséquent, les triangles construits avec un point de cette zone et les arêtes du polygone auraient tous une forme satisfaisante et une taille compatible avec les triangles qui leur font face de l'autre côté du polygone (figure III.3-11 (c)).

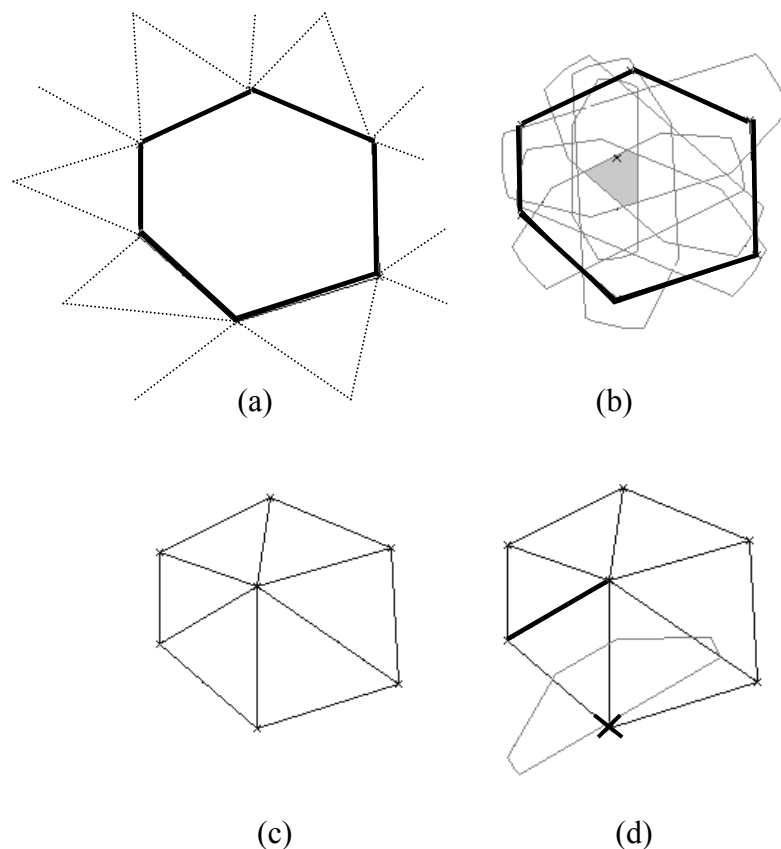


figure III.3-11. Utilisation de la zone caractéristique

En revanche, rien ne garantit qu'entre eux, ces triangles offrent les garanties désirées de taille et de forme. Par exemple, la figure III.3-11 (d) montre que le point mis en évidence

n'appartient pas à la zone caractéristique de l'arête en gras. Le choix fait pour le sommet commun aux triangles construits n'est donc pas judicieux. En conséquence, il faut balayer la zone grisée pour essayer « tous » les points qu'elle contient dans l'espoir d'y trouver celui qui permettra de construire un ensemble de triangles mutuellement compatibles en taille. Si aucun point ne permet de conclure positivement, cela signifie qu'il n'est pas possible de construire des triangles satisfaisant aux deux critères dans l'état actuel de la zone à remailler (s'il existait un tel point, il serait dans la zone). Ceci suggère de démailler afin d'agrandir la zone de travail et constitue donc une évolution possible de la TARF. Le cas de la figure III.3-11 (d) donne cependant une idée de la difficulté de la chose, même dans une situation pourtant très favorable : en général, les zones caractéristiques ne se recouvrent pas, leurs interférences sont nombreuses et dispersées et il est particulièrement difficile de savoir dans laquelle créer le prochain point.

Une autre possibilité d'exploiter les zones caractéristiques est de construire un premier maillage garantissant la forme des triangles puis d'appliquer un déplacement de points en vue de les placer à l'intérieur d'un maximum de zones caractéristiques pour améliorer les compatibilités de taille.

Chaque intersection de plusieurs zones caractéristiques est un endroit vers lequel on va s'efforcer de déplacer des points du maillage. Les possibilités de mouvement que cela offre constituent un espace de solution. Le fait de devoir parcourir cet espace de solutions pour trouver la meilleure signifie très probablement que nous devons avoir recours à des techniques d'optimisation. C'est l'impossibilité d'entreprendre une nouvelle recherche bibliographique dans ce domaine à ce stade du doctorat qui nous a amené à différer l'exploration des possibilités offertes par les zones caractéristiques en ce domaine. Tout au plus peut-on remarquer que les deux composantes principales d'une méthode d'optimisation sont le calcul du coût et la modification de la solution.

Dans notre cas, le coût est fonction du nombre de triangle dont la forme n'est pas satisfaisante, et du nombre de fois que deux triangles adjacents n'ont pas des tailles compatibles. Pour davantage de souplesse, on peut tolérer d'explorer un domaine plus vaste à la condition de pondérer les environs des zones caractéristiques de façon à pénaliser le non respect des contraintes. Cette pénalisation peut par exemple s'appuyer sur la pondération proposée sur la figure III.3-12 : 0 – donc aucune pénalité – quand le point est idéalement

placé, 1 quand seule la taille n'est pas correcte, 2 quand un seul angle est inférieur à la valeur seuil etc.

En ce qui concerne la modification de la solution, elle correspond prioritairement à un mouvement des sommets vers les intersections des zones. Le calcul du coût et le déplacement dans l'espace des solutions sont donc réalisables.

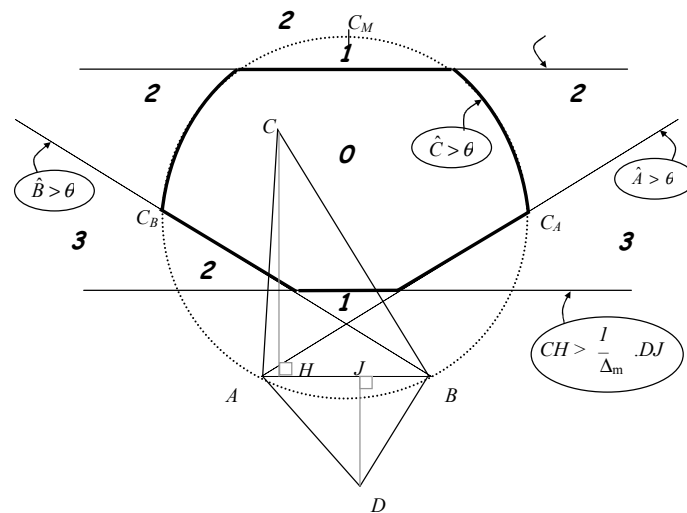


figure III.3-12. Pondération des parties de la zone caractéristique d'une arête

III.4.5 Bilan sur le remaillage des zones vides

L'objectif de la présente étude était de permettre la construction de maillages triangulaires en contrôlant non seulement la forme des triangles, mais aussi la variation de taille. Cette problématique découle d'une volonté de ne supprimer du maillage, au cours d'une opération booléenne, que ce qui est strictement nécessaire pour que le raccordement des maillages puisse se faire de manière progressive et avec satisfaction de ces deux contraintes.

Nous avons vu que l'obtention d'un maillage triangulaire contraint par une variation de taille est un problème difficile pour plusieurs raisons :

- parce qu'on cherche à démailler quand le respect de la variation de taille n'est pas satisfiable ;
- parce qu'il est difficile de déterminer dans quelles conditions cette variation de taille n'est pas satisfiable ;

- parce qu'on cherche à construire un maillage contrôlée par cette variation de taille. Cette dernière difficulté provient du fait que les critères de variation de taille et de qualité en forme peuvent être, dans certaines situations, antagonistes, c'est-à-dire qu'améliorer l'un peut détériorer l'autre.

Il serait nécessaire, afin d'avoir un certain nombre de garanties, d'exprimer le problème dans un contexte plus formel mais le temps imparti sur cet aspect relativement à la totalité de nos travaux ne nous permet pas d'approfondir davantage le sujet. Les méthodes vues ici ne constituent pas des propositions mais exposent les voies que nous avons empruntées et les problèmes que nous y avons rencontrés.

Dans l'immédiat, pour la phase de remaillage des zones vides, nous nous contentons d'une méthode existante par exemple celle de [SHI98], même si elle ne contrôle pas la variation de taille. Les résultats fournis sont en effet de bonne qualité concernant la forme des éléments grâce entre autres au fait que la population de particules, donc de points, est adaptative et que les points sont déplacés durant le processus pour améliorer le maillage. Le contrôle de la taille peut alors se faire en interpolant les tailles imposées sur les contours d'intersection ainsi que sur les arêtes qui limitent la face¹.

III.5 Conclusion

Ce chapitre a présenté les difficultés induites par les opérations booléennes appliquées à des maillages surfaciques. L'extension de cette approche pour des maillages volumiques n'est pas immédiate pour deux raisons. Tout d'abord il faut déterminer où se situe l'intersection des solides. Si cela est clairement défini pour des maillages surfaciques, ce n'est pas le cas des maillages volumiques. En effet, il faut décider, parmi les mailles intérieures appartenant à l'intersection des deux volumes, s'il faut conserver celles provenant d'un opérande ou de l'autre, puisque les mailles situées dans l'intersection des solides appartiennent aux deux. Une priorité donnée aux opérandes, témoignant par exemple de l'importance de chaque opérande vis-à-vis de l'ensemble, permettrait de conserver les mailles de l'opérande le plus prioritaire. La seconde raison est qu'il faut, là aussi, déterminer une stratégie pour le remaillage des vides après l'opération booléenne. Or si le maillage est hexaédrique, des précautions particulières doivent être prises lors de la phase de remaillage. Ce même problème se pose pour des maillages quadrangulaires structurés et a été traité récemment dans [WAN03].

¹ Même si le modèle géométrique est un CSG, certaines arêtes appartiennent à deux surfaces et limitent donc une face.

La phase de remaillage des zones vides au cours d'une opération booléenne avec un contrôle de la forme et de la variation de taille sur un intervalle aussi court que possible reste un problème ouvert que nous n'avons pas réussi à résoudre dans le temps qui lui était imparti dans cette thèse.

À défaut d'une méthode de type TARF, il est toujours possible d'utiliser d'autres méthodes, quitte à remailler des zones plus grandes. Ceci n'enlève rien à l'intérêt de construire un maillage à partir d'un arbre CSG, en combinant les maillages des feuilles puis des sous-arbres. Rappelons en effet que cela permet de réduire les problèmes de raccords entre les surfaces gauches, en fournissant une représentation « étanche », sans fissure ni recouvrement. De plus, la simplicité et l'absence d'équivoque de la représentation des maillages permettent un transfert de géométrie plus fiable, car moins soumis à des erreurs d'interprétation, et finalement pas moins précis, étant donné que des approximations sont systématiquement réalisées en fin de chaîne, que l'on parle de calcul par éléments finis, de visualisation ou de prototypage rapide.

Chapitre IV : Architecture d'une Bibliothèque d'Algorithmes de Maillage Génériques ou Spécialisés (BAMGS)

IV.1 Problématique et objectifs

L'objectif de la BAMGS est de mettre à disposition des systèmes de conception et des AEM, l'outillage logiciel nécessaire à la construction, à l'adaptation et à la combinaison de maillages. Cet outillage doit satisfaire trois contraintes : 1. la BAMGS doit être **indépendant des AEM**, 2. il doit être **indépendant du type de modèle géométrique**, et 3. il doit être **extensible** de manière à intégrer facilement de nouveaux algorithmes de maillage. Cette introduction donne un aperçu des propositions que nous faisons en ce sens. Elle est suivie de trois parties qui détaillent ces propositions et d'une critique qui met en évidence leurs limites mais aussi leurs perspectives.

Pour rester **indépendant des AEM**, le contrôle du maillage et l'invocation des fonctions de maillages ne se fait pas dans la BAMGS mais dans le système appelant, par exemple le module d'assemblage. L'appelant injecte des contraintes sur le maillage dans la BAMGS qui lui retourne un ensemble d'algorithmes les satisfaisant (voir chapitre 2). Pour cela, il faut que la BAMGS conserve l'association entre les algorithmes de maillage et les propriétés qu'ils satisfont.

Pour rester **indépendant du type de modèle géométrique** utilisé (BRep, CSG), toutes les fonctions qui en dépendent ne sont pas implémentées dans la BAMGS. L'accès à la géométrie se fait par un ensemble de fonctions que doit implémenter le modéleur géométrique. Le choix d'une fonction de maillage n'est alors plus contraint par le type de modèle géométrique, mais

par les fonctions géométriques que peut rendre ce modéleur. C'est la raison pour laquelle le module d'assemblage, dont le fonctionnement est très différent pour un BRep et un CSG, n'est pas conservé dans la BAMGS. Par contre, un algorithme de maillage d'un carreau paramétrique y trouverait sa place. Ce type d'algorithme requiert un certain nombre de services géométriques (par exemple la distance à une surface). Dans le souci de rester indépendant du modéleur géométrique, nous avons fait en sorte que les services requis soient en grande majorité réalisables dans tout type de modéleur.

Enfin pour que la BAMGS soit **extensible**, c'est-à-dire pour qu'il puisse intégrer de nouveaux algorithmes ou tolérer des évolutions d'algorithmes existants, nous cherchons à exploiter au maximum la redondance qui existe entre ces algorithmes de maillage de manière à réduire l'ajout de code aux parties réellement innovantes. Cela comprend la réutilisation de schémas généraux communs à plusieurs algorithmes de maillage comme cela a été mis en avant dans le premier chapitre.

- Le problème de l'extensibilité n'est pas trivial car les similitudes entre les algorithmes sont variées. Certaines concernent la dimension des mailles (2D/3D) (cf. le noyau de Delaunay, les raffinements de Delaunay ou les méthodes par avancement de fronts), d'autres se situent au niveau de la forme de base des mailles (méthodes de Delaunay pour les triangles/tétraèdres et méthodes de remplacement d'éléments par des motifs pour les quadrangles/hexaèdres), d'autres enfin ont des principes similaires quelles que soient la dimension et la forme de base des mailles (cf. systèmes à particules).
- Notons cependant que, globalement, la redondance de code peut être de deux grands types : des algorithmes spécifiques qui sont réutilisables, et des *algorithmes génériques* (ou *squelettes d'algorithmes*) décrivant le processus général d'une technique de maillage. Dans le premier chapitre nous avons vu que les algorithmes spécifiques sont souvent réutilisés par plusieurs algorithmes. Il s'agit par exemple de fonctions de mesure de la qualité des éléments, de constructions géométriques (droites, cercles, création de nouveaux points...), mais aussi des fonctions particulièrement innovantes qui ne suivent aucun algorithme générique. Dans ce cas la redondance est trivialement supprimée par une programmation objet ou au moins structurée. Le second type de redondance concerne les algorithmes génériques. Un algorithme générique est un algorithme « à trous », qu'il faut combler avant de pouvoir l'exécuter. Chaque trou a un sens relativement à l'algorithme générique (par exemple « choisir un lieu de traitement »), et constitue une grande étape

d'une technique de maillage, qui peut être réalisée de différentes manières selon le contexte (c'est-à-dire les contraintes imposées). Par exemple l'algorithme générique des méthodes itératives (cf. chapitre 1) peut être spécialisé en un raffinement de Delaunay, en une méthode de construction par avancement de fronts ou en une simplification de maillage, en fonction du choix fait pour chacune de ses étapes.

- Nous avons également comme objectifs à long terme de fournir une assistance dans la phase d'adaptation d'un algorithme générique à une fin particulière. On pourrait par exemple informer l'opérateur d'une incompatibilité entre les réalisations qu'il a choisies pour deux étapes de l'algorithme. Ceci n'est pas abordé dans le mémoire mais nous nous sommes efforcé de ne pas faire de choix qui empêcherait d'aller en ce sens à l'avenir.

IV.2 Indépendance vis-à-vis des AEM

La gestion des contraintes consiste à restituer un ensemble de *fonctions de maillage* à partir d'un ensemble de *propriétés* du maillage ou de la méthode. Les fonctions de maillage sont toutes les fonctions de la BAMGS, de bas niveau comme de haut niveau, participant aux algorithmes de construction, d'adaptation et de combinaison de maillages.

Les propriétés que nous considérons qualifient soit le maillage, soit la méthode. Concernant le maillage, les propriétés concernent le *type de mailles* (triangulaire, quadrangulaire, tétraédrique, etc.), la *forme des mailles* (au regard d'un certain critère de forme), leur *orientation* (anisotropie) et leur *taille*, l'*erreur d'approximation* (des mailles, des arêtes et des sommets), la *variation de taille* des mailles, la *régularité* du maillage (en observant la variation d'angle des normales aux mailles par exemple), la *structure* du maillage (maillage structuré ou non structuré), et la *taille du maillage* (en nombre de nœuds ou en nombre de mailles).

Les fonctions de maillage peuvent avoir comme propriétés le fait d'être *sensibles à la frontière* (les éléments de bonne qualité au regard d'un certain critère se situent le long de la frontière), *insensibles à l'orientation* (le maillage résultat ne dépend pas de l'orientation du solide) et *interruptibles* (il n'est pas nécessaire que l'algorithme termine complètement pour être en mesure de fournir un maillage valide). Les propriétés portent également sur les pré requis d'une fonction : le solide peut être obtenu par extrusion d'un contour le long d'une courbe (ce doit être le cas pour pouvoir appliquer certaines méthodes de construction de maillage hexaédriques), le maillage de la frontière du domaine à mailler doit posséder un

nombre pair de sommets (c'est le cas de toutes les fonctions de construction de maillage quadrangulaire qui n'ajoutent pas de sommets sur la frontière), etc. De même, les garanties que fournissent les fonctions sur le maillage résultat sont conservées dans les propriétés (angle minimum d'une triangulation, etc.).

Le lien entre les fonctions disponibles dans la BAMGS et leurs propriétés est conservé dans une base de données. Cette base a deux intérêts : 1. restituer un ensemble d'algorithmes de maillage pour un système appelant tel qu'un système de conception et 2. restituer un ensemble d'algorithmes de maillage pour un expert souhaitant intégrer un nouvel algorithme dans la BAMGS. Dans le second cas, cela lui permet de trouver rapidement les algorithmes déjà existants dans le but d'en réutiliser des parties. Parmi les fonctions de maillage, certaines effectuent un traitement de maillage de bout en bout et sont accessibles depuis un système extérieur, alors que d'autres sont plus restreintes et ne sont accessibles que pour des requêtes internes lors de l'ajout d'un nouvel algorithme (cas d'une mise à jour de la BAMGS).

Dans les deux cas, l'obtention d'un ensemble de fonctions à partir de propriétés est réalisée par une requête adressée à la base de données. Cette base lie des propriétés, des paramètres et des services géométriques à des fonctions de maillage. Le diagramme IV.2-1 propose un modèle conceptuel des données de cette base de données exprimé dans le formalisme MERISE (Modèle Conceptuel de Données).

On ne conserve d'une fonction, que ce qui permet de la retrouver parmi les autres, c'est-à-dire son nom, ses arguments, la classe dans laquelle elle se trouve et le chemin qui permet d'y accéder. Le corps des fonctions est conservé dans une seconde partie de l'architecture, qui est détaillée dans la partie IV.4. Une propriété d'une fonction de maillage peut être soit induite implicitement par le processus de maillage, soit contrôlée explicitement par un paramètre. Par exemple, dans la méthode de triangulation proposée par [SHW02], la propriété « contrôle de l'agrandissement » est induite implicitement par le processus de maillage qui agrandit les mailles en respectant leur qualité en forme. Par contre, dans la méthode de triangulation de [OLL01], l'agrandissement des triangles est contrôlé par un coefficient. En fait lorsque le système appelant émet une requête à la BAMGS pour choisir une méthode de maillage à partir de paramètres, il précise les propriétés qu'il souhaite contrôler. Il reste évidemment à sa charge de vérifier s'il dispose des informations nécessaires à ce contrôle. À une fonction de maillage sont également associés des paramètres (ou arguments), ceux-ci étant liés également à une ou plusieurs propriétés. Par exemple, la propriété « forme des triangles » est une

propriété qui peut être contrôlé de diverses façons : par la valeur du plus petit angle, par le rapport du rayon du cercle inscrit et de la longueur de la plus grande arête, etc. Donc à cette propriété sont associés plusieurs paramètres.

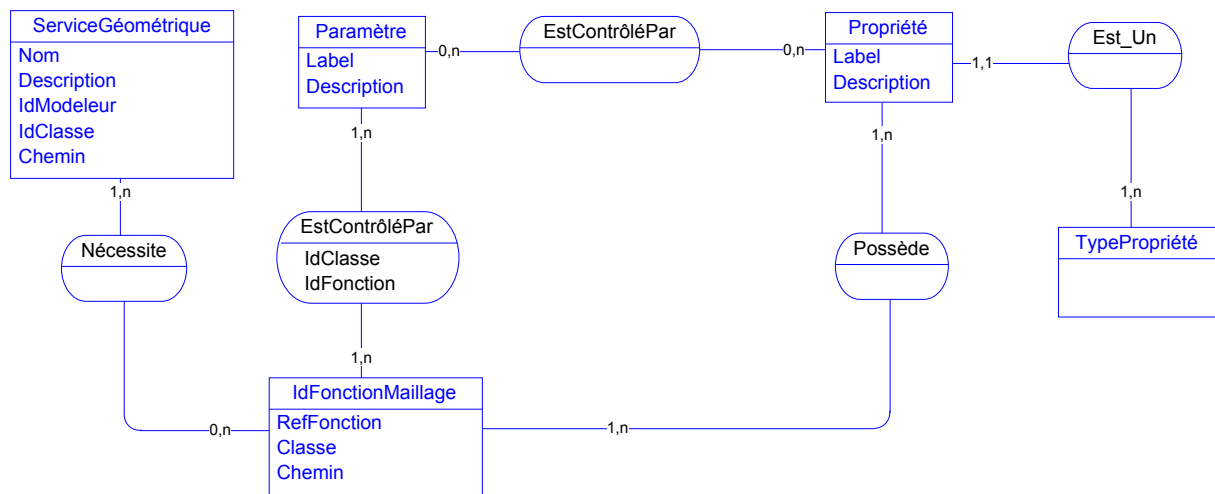


diagramme IV.2-1. Fonctions de maillage

Enfin certaines fonctions de maillage nécessitent parfois des services géométriques, c'est-à-dire des fonctions qui requièrent un accès au modeleur géométrique. Par conséquent, les services géométriques nécessaires sont associés aux fonctions de maillage. Ils contraignent également le choix d'une fonction de maillage car il est possible qu'un service géométrique nécessaire ne soit pas rendu par le modeleur géométrique. Par exemple, la plupart des méthodes de construction de maillages anisotropes réalisent la construction en 2D, éventuellement dans l'espace paramétrique s'il s'agit de surfaces gauches. Or l'obtention d'une surface paramétrique à partir d'une surface implicite n'est pas toujours possible, rendant impossible l'utilisation de ces méthodes.

Comme un algorithme générique possède un comportement différent selon le contexte, il n'est pas possible de lui attribuer des propriétés sur les maillages, ni des services géométriques. L'information utile à conserver pour ces algorithmes est le rôle de ses différentes étapes, conservées dans une propriété.

Le paragraphe qui suit présente en quoi consistent ces services géométriques et leur intérêt dans l'architecture.

IV.3 Indépendance vis-à-vis des modèles géométriques

Actuellement, les services géométriques nécessaires aux maillages sont spécifiques à un modèle géométrique (Brep, CSG). En effet, une stratégie couramment adoptée est de reconstruire le modèle géométrique pour pouvoir le mailler. Ceci rend les maillages

dépendants du modèle géométrique, ce qui s'oppose à l'intégration des systèmes (cf. chapitre II). Pour réaliser l'indépendance des algorithmes de maillage vis-à-vis du type de modèle géométrique, nous avons établi une liste de services géométriques de bas niveau (cf. annexe E) que la plupart des modeleurs peuvent satisfaire et nous imposons que tous les algorithmes de maillage consultent le modèle au travers de ces services. Par exemple une méthode de suivi de cubes (ou *marching cube*, [LOR87]) nécessite uniquement la fonction réalisant le test d'appartenance d'un point au solide, qui figure dans les services géométriques que nous avons recensés. L'intégration du suivi de cubes à la BAMGS est donc, concernant les services géométriques, immédiate et tous les modeleurs étant capables de vérifier l'appartenance d'un point à un solide, il pourra être utilisé quel que soit le modeleur géométrique sous-jacent. Par contre, certaines méthodes utilisant le critère de Delaunay nécessitent une notion de distance géodésique pour les surfaces gauches, ainsi que la construction d'un voisinage d'un point, l'appartenance d'un point à un voisinage, etc. Ces services peuvent être plus ou moins performants et plus ou moins difficiles à implémenter selon le modèle géométrique. Le même algorithme de maillage peut alors être invoqué avec des modeleurs différents mais leurs performances varient avec le modèle géométrique. Notons que cette variation des performances ne doit pas être imputée à l'organisation proposée : les algorithmes conservent leurs performances habituelles lorsqu'ils sont appliqués au type de modèle pour lequel ils ont été conçus. C'est lorsqu'ils sont utilisés sur d'autres types de modèles, ce que permet notre organisation, que les performances sont moindres. Notons également que le nombre de services géométriques nécessaires est faible, rapporté au nombre d'algorithmes de maillage.

Un inconvénient apparent de cette approche est que les modeleurs géométriques sont souvent vus par les outils de maillage comme des bases de données et les services géométriques disponibles sont réduits à des accès à cette base par des requêtes ([GEO97]). Pourtant, les modeleurs géométriques proposent souvent, par leur API¹, des services de bas niveaux. La différence principale réside dans le fait que les services que nous proposons comportent des fonctions de construction (de points en particulier), alors que, en utilisant des interfaces existantes ([HAI98], [TAU00]), cette construction est entièrement à la charge du système de maillage ce qui le rend davantage dépendant du modèle géométrique. Il n'existe pas à la connaissance de l'auteur, d'interface permettant d'accéder à ce type de fonction de manière unifiée pour différents *types* de modèles géométriques (CSG, BRep). Les travaux qui vont dans ce sens visent un accès unifié à des BRep ([HAI98], [TAU00]).

¹ Application Programming Interface

Par ailleurs, un des rôles essentiels d'un modèle géométrique est de permettre de vérifier un certain nombre de propriétés du solide. La simulation, le prototypage rapide ainsi que la visualisation sont des AEM dont la finalité est de vérifier certaines fonctionnalités du produit (propriétés mécaniques, thermiques, esthétiques, etc.). Par conséquent, nous pensons qu'il relève de la responsabilité du modéleur géométrique, de proposer les services géométriques nécessaires à la vérification des fonctions du produit, c'est-à-dire, souvent, à la construction et à l'adaptation de maillage. Cela revient à dire que pour utiliser tel service de maillage, il faudrait que tels services géométriques soient accessibles auprès du modéleur.

Ceci peut paraître comme une contrainte forte mais, l'accès au modèle géométrique par des fonctions de bas niveau, même si elles n'apparaissent pas encore dans les standards d'échange, reste la meilleure solution pour que le module de maillage soit indépendant du modèle géométrique.

Il est possible que certains services géométriques nécessaires ne soient pas disponibles. Les algorithmes de maillage qui en dépendent ne peuvent alors pas être proposés à l'application qui requiert leur service, ce qui peut engendrer des situations bloquantes où aucun algorithme n'est renvoyé à l'appelant. Lorsqu'une telle situation arrive, il faut que l'appelant modifie les propriétés de manière à pouvoir tout de même mailler l'objet. Pour réduire le nombre de situations bloquantes, certains des services géométriques sont réalisés, lorsque cela est possible, de manière approchée par la BAMGS. Par exemple, l'intersection entre une droite et la surface peut être complexe à résoudre si la surface possède un degré élevé ou si sa représentation est implicite, de type squelette. Cependant, si le test d'appartenance d'un point au domaine n'est pas trop coûteux en temps de calcul, il peut être intéressant de déterminer une approximation du point d'intersection par dichotomie.

Pour chaque fonction de la BAMGS, il est nécessaire de recouper les services géométriques nécessaires par les services disponibles. Pour le moment, cette correspondance est réalisée manuellement, dans une phase d'initialisation de la BAMGS. Les interfaces virtuelles ([HAI98], [TAU00]) tendent à réduire les interventions manuelles, mais ne les suppriment pas car tous les services géométriques ne sont pas fournis par ces interfaces.

IV.4 Extensibilité

Dans cette partie, nous nous intéressons à la manière d'organiser et de représenter les algorithmes de maillage. La représentation des algorithmes non génériques soulève des

problèmes d'organisation, mais pas de représentation. Par conséquent nous étudions plus particulièrement la représentation des algorithmes génériques dans l'architecture.

Cette partie comporte quatre paragraphes. Le premier présente deux projets, CGAL (Computational Geometry Algorithms Library, [VEL99], [BOI02]) et GrAL (Grid Algorithms Library, [BER00], [BER01]) portant sur des bibliothèques de fonctions géométriques comportant des algorithmes génériques. Leurs limites nous ont amené à étudier dans le deuxième paragraphe, les mécanismes de la programmation objet utilisés pour exprimer la généricité. Dans le troisième paragraphe, nous proposons notre propre solution pour représenter les algorithmes génériques au sein de la BAMGS. Le quatrième paragraphe illustre par un exemple l'utilisation de la solution proposée pour représenter un algorithme de maillage utilisant l'algorithme générique dit itératif du premier chapitre. La critique de l'approche apparaît dans la partie discussion du chapitre.

IV.4.1 Bibliothèques d'algorithmes existants

CGAL (Computational Geometry Algorithms Library, [VEL99], [BOI02]) et GrAL (Grid Algorithms Library, [BER00], [BER01]) sont deux bibliothèques de fonctions géométriques prenant en compte la généricité. CGAL fait partie d'un vaste projet européen et israélien dont le but est de réunir au sein d'un seul système, des algorithmes géométriques robustes, performants et faciles d'utilisation. De nombreuses publications sont apparues à l'issue de ce projet mais nous nous intéressons ici surtout aux concepts de programmation générique qu'il met en œuvre. Ces aspects sont décrits dans [VEL99] et [BOI02]. Ils sont fortement inspirés des concepts de programmation générique de la bibliothèque STL (Standard Template Library) [STE94] qui est d'ailleurs très utilisée dans CGAL. La bibliothèque STL (Standard Template Library) fournit des structures de données classiques : tableaux, listes linéaires, arbres, etc. Dans cette bibliothèque, la programmation générique consiste à dissocier les structures de données des algorithmes qui peuvent les exploiter de manière à pouvoir appliquer un traitement sans se soucier de la représentation. Un exemple typique d'algorithme générique consiste à parcourir un ensemble, à sélectionner les éléments à traiter au regard de certains critères, puis à appliquer un traitement sur chaque élément sélectionné. Cet algorithme doit être paramétré par la fonction qui teste si un élément est à traiter ou non, et par celle qui traite chaque élément. Ce même principe est utilisé pour les algorithmes géométriques dans CGAL, comme les fonctions de maillage. Une fonction de maillage est vue comme un parcours d'un ensemble, ce parcours effectuant un traitement sur chaque

élément de l'ensemble. La généralité s'exprime principalement grâce aux itérateurs (linéaires, cycliques, etc.) et aux *classes de caractéristiques* (*traits classes* en anglais). Les itérateurs permettent d'effectuer des parcours génériques quelle que soit la structure des données. Les *classes de caractéristiques* encapsulent les données et les traitements qui donnent un comportement particulier à l'algorithme générique (c'est-à-dire qui remplissent les « trous » de l'algorithme générique). L'utilisation d'un algorithme générique se fait en dérivant la classe de caractéristiques et en surchargeant les méthodes invoquées par l'algorithme générique.

Concernant le maillage, ces travaux ont été appliqués entre autres à des algorithmes de triangulations de Delaunay ([BOI02]). A notre connaissance, ces travaux n'ont pas été étendus à davantage de fonctions de maillage et les algorithmes génériques n'ont pas été publiés. Par ailleurs, la notion de *template*, ainsi que la surcharge d'opérateurs du langage C++ sont très présentes ce qui rend ces travaux relativement dépendant du langage.

Le second projet important dans ce domaine est GrAL (Grid Algorithms Library) ([BER01]) qui s'appuie sur les travaux de [BER00]. Ce dernier propose un cadre de développement (ou *framework*) pour la programmation générique dédiée au calcul scientifique en général, y compris le maillage. Le cadre est donc plus général que CGAL cependant, les concepts pour la programmation génériques sont similaires. L'objectif est toujours de dissocier les structures de données de la manière de les exploiter. Ainsi, les travaux portent essentiellement sur une manière générique de parcourir des ensembles d'entités géométriques (ensemble de sommets, ensemble d'arêtes, ensemble de faces, etc.) à l'aide d'itérateurs et de manipuler les informations conservées au niveau de chaque entité géométrique ou groupe d'entités géométriques. En particulier, leurs travaux ne portent pas sur les points communs qui peuvent exister entre différents algorithmes de maillage, et donc sur la manière d'exploiter cette généralité.

Ces deux projets constituent des avancées très pertinentes pour la programmation générique en algorithmique géométrique. Elles permettent l'écriture d'algorithmes génériques et donc une grande réutilisation des composants logiciels. Les inconvénients donnés par les auteurs concernent la difficulté de l'écriture des programmes, leur mise au point, ainsi que la compilation du code. En effet, cette méthodologie de programmation s'adresse à des développeurs expérimentés car elle offre une grande liberté et est quelque peu fastidieuse. De

plus, la mise au point est plus difficile parce que les messages des compilateurs ne sont pas toujours explicites.

Enfin, tous les langages objets n'intègrent pas les concepts mis en œuvre comme le paramétrage de classe, notamment Java qui est de plus en plus utilisé entre autres pour sa portabilité et la disponibilité de ses bibliothèques standard.

Par ailleurs la généricité est très forte et, donc, la sémantique attachée à chaque étape d'un algorithme générique est faible. Or les étapes des algorithmes génériques *dont nous avons besoin* ont véritablement un sens relativement au processus de maillage. Il s'agit de « déterminer un lieu à traiter », de « choisir l'opération à appliquer localement », etc. Cette sémantique est importante pour utiliser les algorithmes génériques. Elle donne une indication sur le rôle de chaque étape et permet ainsi de différencier les algorithmes génériques. Nous avons par exemple l'algorithme générique itératif, l'algorithme générique de décomposition spatiale et l'algorithme générique de décomposition en domaines.

Ces raisons nous ont amené à étudier les mécanismes de la programmation objet dédiés à la programmation générique, n'utilisant pas la notion de classe paramétrée telle qu'elle est définie dans CGAL et GrAL.

IV.4.2 Concepts pour la programmation générique

Nous souhaitons représenter un algorithme générique comportant des « trous » pour certaines de ses étapes. L'utilisation d'un algorithme générique se fait donc en deux phases : la *configuration* qui consiste à remplir les trous, puis son exécution. La configuration revient donc à donner un comportement particulier à un algorithme qui, par définition, est générique. Nous cherchons un patron de conception permettant de représenter cela.

Les patrons de conceptions sont des diagrammes de classes qui permettent de résoudre des problèmes connus, et qui possèdent des propriétés connues. Ils constituent des briques de base réutilisables pour la conception du diagramme des classes d'un système. Un certain nombre d'entre eux sont devenus des standards ([GAM95]). Nous en voyons ici deux, exprimant les concepts de base nécessaires à la programmation générique.

Le premier, appelé *méthode modèle* (*template method*), utilise directement le mécanisme d'héritage pour permettre la généricité. Une classe générale implémente un algorithme générique dont les étapes sont des méthodes (au sens programmation) dites *méthodes crochets*

déclarées abstraites dans cette classe générale. Ces méthodes ont éventuellement un corps vide, mais sont surchargées dans chacune des classes dérivées. Cependant, certaines étapes comme *methodeEtapeA* (diagramme IV.4-1) doivent pouvoir être communes à plusieurs configurations d'un algorithme générique. Pour ne pas dupliquer le code, les implémentations des étapes sont placées dans des classes extérieures à la hiérarchie (comme par exemple *UnOutil*) qui peuvent alors être utilisées par plusieurs configuration de l'algorithme générique.

Cette solution permet de représenter de manière dissociée l'algorithme générique (*algoGenerique*), les outils (*UnOutil*) qui servent à configurer l'algorithme générique, et les configurations (*UneImplementation*, *UneAutreImplementation*). Cependant, l'ajout d'une nouvelle configuration, même si elle ne fait que combiner des outils existants, nécessite l'écriture d'une nouvelle classe héritant d'*algoGenerique* et implémentant toutes les méthodes crochets.

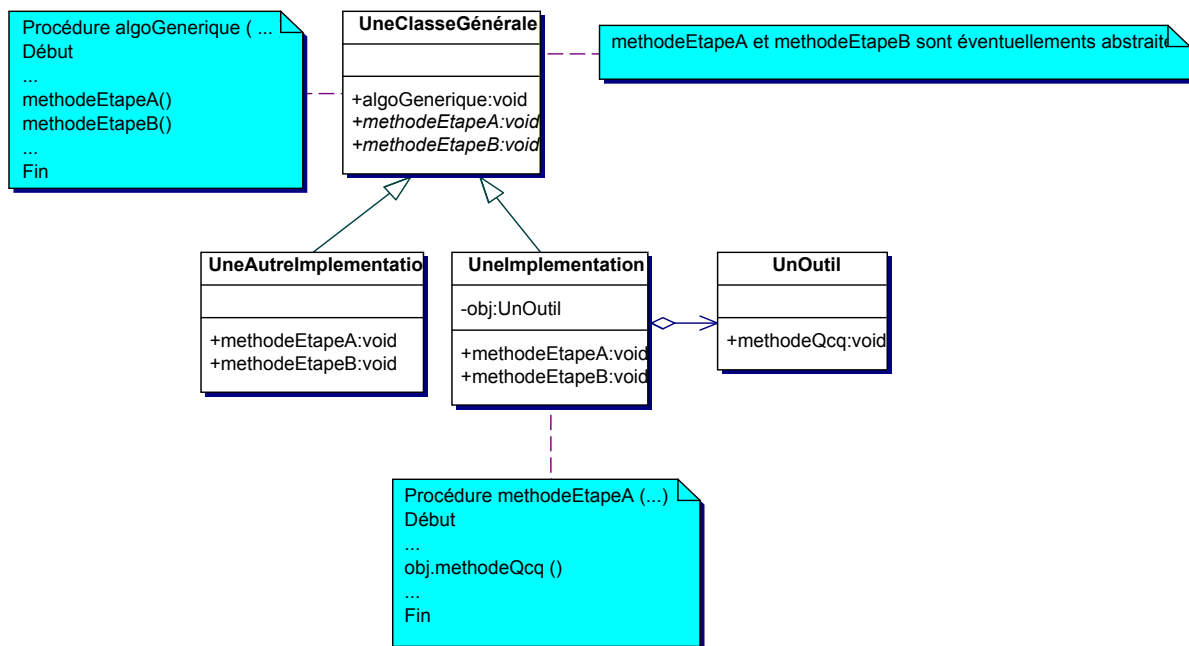


diagramme IV.4-1. La méthode modèle

Un deuxième mécanisme utilisé dans les patrons de conception consiste à dissocier une abstraction de son implémentation (diagramme IV.4-2). C'est le patron de conception nommé *adapteur d'objets* (*object adapter*). Toutes les étapes de l'algorithme générique sont des fonctions membres d'une classe abstraite (appelée aussi parfois interface). Une classe générale qui comporte un algorithme générique est associée à une ou plusieurs

implémentations virtuelles (*UneImplementationVirtuelle*). Toute classe implémentant une étape de l'algorithme générique doit alors dériver de l'implémentation virtuelle.

Alors que dans le mécanisme précédent, il fallait écrire une classe par configuration (*UneImplementation*), avec ce patron de conception, il devient possible de réutiliser des parties de configurations ayant déjà été définies durant d'autres configurations. Cela permet de mieux combiner l'existant.

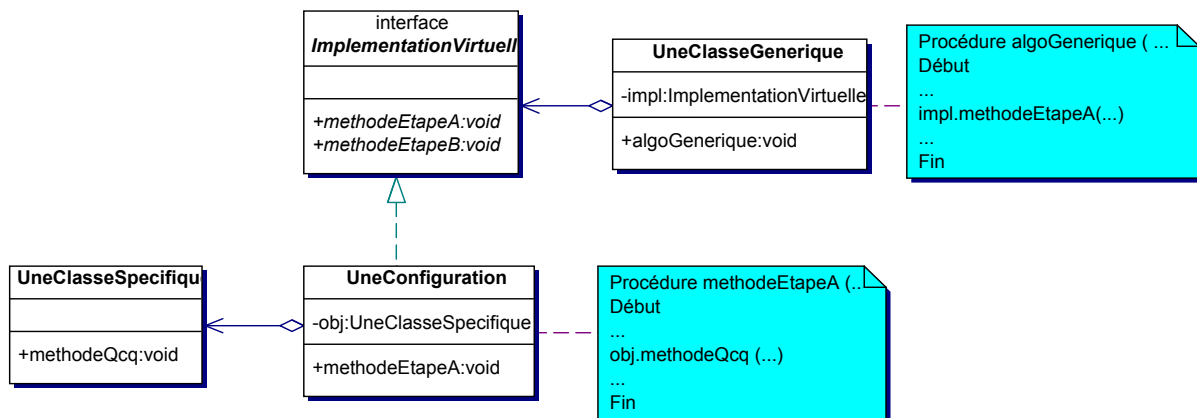


diagramme IV.4-2 L'adaptateur d'objets

Cependant, la configuration de l'algorithme générique nécessite à nouveau l'écriture de fonctions dont le nom et les paramètres sont imposés, comme les méthodes crochets du mécanisme précédent. Cet aspect est très contraignant pour deux raisons. Il faut d'abord compiler le code avant de pouvoir utiliser les configurations. Cela empêche par exemple de construire une configuration inédite à l'aide une interface graphique. Parmi les patrons de conception standard ([GAM95]), aucun ne permet d'éviter cette contrainte. De plus, les paramètres de la fonction réalisant une étape, ne sont pas toujours les mêmes. Par exemple, les paramètres de l'implémentation de l'étape « choix d'un lieu à traiter » de l'algorithme générique itératif sont différents selon que cette fonction est le noyau de Delaunay (qui prend en paramètre une triangulation, une maille et un sommet) et ou une opération de compression d'arête (qui prend en paramètre un maillage et une arête).

IV.4.3 Proposition d'un patron de conception

Les propriétés que nous cherchons à atteindre par un nouveau patron de conception sont :

- La non utilisation des classes paramétrées de type *template*

- La possibilité de ne pas avoir à recompiler le code du cadre de développement dans l'espoir de construire dynamiquement des configurations d'algorithmes génériques.

IV.4.3.a Algorithme générique

Le défaut des mécanismes décrits au paragraphe IV.4.1 est que le nom et les paramètres de la fonction qui implémente une étape d'un algorithme générique sont imposés. Pour éviter cela, il faut conserver au niveau de la classe qui comporte l'algorithme générique, les fonctions et les objets qui implémentent ces étapes. Ces étapes peuvent alors être réalisées par des classes qui n'ont absolument aucun lien entre elles. En particulier, il n'y a pas de contraintes sur le nom ou les paramètres des fonctions qui implémentent les étapes.

Par ailleurs, l'algorithme générique ne connaît pas le contexte dans lequel il sera exécuté, une fois qu'il aura été configuré. Par conséquent il ne connaît pas les types des données qui sont nécessaires à l'algorithme de maillage qu'on veut implémenter. Il ne peut donc manipuler les données que par une variable dont le type est générique. Il ne connaît pas non plus les paramètres d'appels des fonctions qui implémentent les étapes. Pourtant il doit faire les appels de ces fonctions. Les paramètres d'appels de ces fonctions sont des données prises soit dans les arguments fournis en entrée de l'algorithme générique, soit dans ses données locales. Dans tous les cas, il faut les trouver, éventuellement les mettre en forme pour pouvoir ensuite faire l'appel. Puisque l'algorithme générique ne détient pas les informations lui permettant de réaliser cette tâche d'initialisation des paramètres d'appel, il la délègue à une autre classe. Autrement dit, il faut configurer la phase d'initialisation de chaque appel à l'aide d'une fonction dédiée à cette tâche. Il faut alors également connaître l'objet sur lequel cette fonction s'applique (diagramme IV.4-3).

En résumé, pour une étape de l'algorithme générique, il faut conserver l'objet et la fonction qui l'implémente, ainsi que l'objet et la fonction qui extraient et formatent les données d'appel de cette étape. Ces objets et ces fonctions, nécessaires à l'exécution d'une étape, sont encapsulées dans la classe *ImplementationEtape*. Il y a donc autant d'objets *ImplementationEtape* dans un algorithme générique qu'il y a d'étapes. L'exécution d'une étape de l'algorithme générique consiste à 1) extraire et formater les données, et 2) faire l'appel de la fonction qui l'implémente et éventuellement retourner les résultats si nécessaire. Sous ces hypothèses, les tâches nécessaires à l'exécution de chaque étape se font de manière transparente. La variable *algo* de la classe *ImplementationEtape* (diagramme IV.4-3) est une

fonction qui peut être quelconque. Par conséquent, pour la déclencher, nous avons utilisé l'instruction « invoquer » que nous supposons applicable à tout **objet** de type fonction.

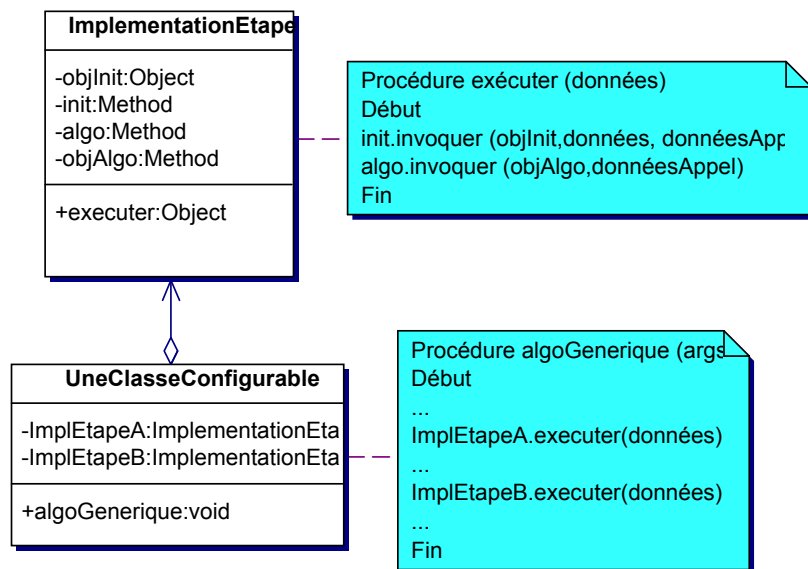


diagramme IV.4-3. Une classe configurable

IV.4.3.b Fonction d'initialisation des données

Les données servant à paramétrer chaque appel se trouvent donc dans la structure de données générique conservant toutes les données de l'algorithme générique (les arguments et les données locales). Pour effectuer l'appel à une fonction implémentant une étape, il faut retrouver ces données et les mettre en forme de manière à ce que les types correspondent à ceux attendus par la fonction. C'est la tâche *d'initialisation des données* d'un appel. Elle consiste en 2 sous-tâches : une projection et une transformation. La projection consiste à retrouver, dans les données génériques, les informations nécessaires à l'appel de la fonction. La projection n'est pas forcément simple puisque l'information requise peut être enfouie plus ou moins profondément dans les données génériques.

Lorsque ces données d'appel sont trouvées, il est possible qu'il faille les transformer de manière à ce que les types correspondent. Considérons par exemple deux appels de fonctions qui se suivent. La première renvoie un point qui est utilisée par la seconde. La première renvoie le point en coordonnées polaires mais la seconde demande en entrée un point en coordonnées cartésiennes. Une transformation est alors nécessaire entre les deux fonctions.

L'implémentation d'une méthode de configuration est spécifique à chaque configuration d'algorithme générique.

Techniquement, les méthodes d'initialisation peuvent se situer n'importe où dans l'architecture. Cependant, comme ces fonctions d'initialisation sont caractéristiques d'une configuration, nous convenons de les implémenter dans les classes qui configurent les algorithmes génériques (une classe de configuration existe pour chaque configuration d'un algorithme générique). Ces méthodes d'initialisation offrent deux avantages : d'abord l'enchaînement des implémentations des étapes est quelconque, ensuite ces implémentations peuvent être quelconques. Il devient alors possible de configurer les algorithmes dynamiquement, par morceaux. Il demeure cependant la contrainte de devoir écrire et compiler les méthodes d'initialisation des données des étapes.

IV.4.3.c Classes de configuration

Le diagramme IV.4-4 montre une configuration de l'algorithme générique donné dans le diagramme IV.4-3 pour une étape seulement. L'étape *EtapeA* est implémentée par la méthode *uneMethodeSpecifique* de la classe *UnOutil*. Une configuration similaire doit être effectuée pour chaque étape de l'algorithme générique.

Une classe de configuration (*MaConfiguration*, diagramme IV.4-4) comporte au moins trois fonctions. La première *configurer*, a pour rôle de remplir les « trous » de l'algorithme générique par les implémentations souhaitées. La configuration d'un algorithme générique (désigné par l'objet *generique*) consiste à construire et à donner un objet, de la classe *ImplementationEtape* à chacune de ses étapes. Il s'agit ici de *ImplEtapeA*. Cela nécessite de fournir :

- à l'objet *ImplEtapeA*, la méthode d'initialisation (*initMethodeEtapeA*) et l'objet associé (l'objet courant *this*) : phase 1.
- à l'objet *ImplEtapeA*, la méthode qui implémente l'étape (*uneMethodeSpecifique*) et l'objet associé (*atool*) : phase 2
- à l'algorithme générique (*generique*), l'objet *ImplEtapeA* configuré : phase 3

La deuxième fonction d'une classe de configuration, *executer*, convertit d'abord en arguments génériques les arguments qu'elle prend en entrée. Ces arguments sont en effet génériques puisque l'algorithme (générique) ne connaît pas ses contextes d'utilisation. Ensuite la méthode *executer* appelle l'algorithme générique supposé déjà configuré.

Les autres fonctions sont les méthodes d'initialisation des données utilisées pour la configuration de chaque étape (par exemple *InitMethodeEtapeA*).

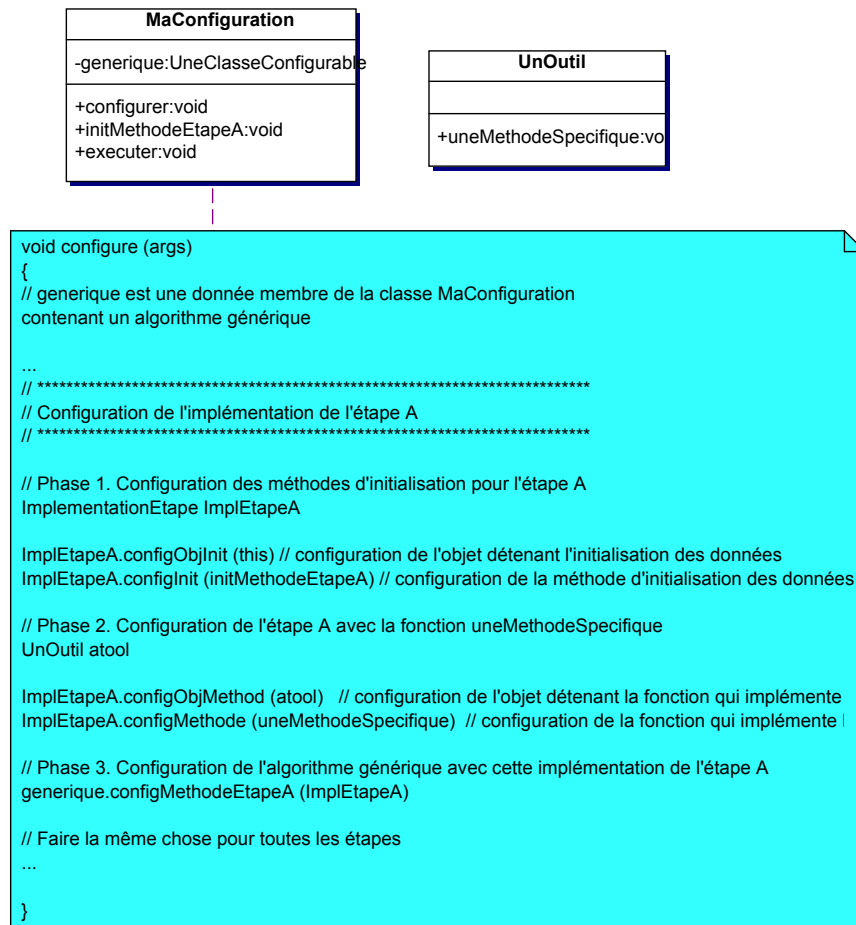


diagramme IV.4-4. Configuration d'un algorithme générique

IV.4.3.d Données de maillage

Un dernier aspect de l'architecture concerne la représentation des maillages et des données spécifiques à un ou plusieurs algorithmes. Les maillages sont conservés dans une structure de données proche d'un BRep auquel a été adjoint des informations permettant de prendre en compte des mailles volumiques. De plus l'héritage permet de manipuler indifféremment des mailles et des maillages de différentes sortes (triangulaires/quadrangulaires, 2D/3D). Les données spécifiques aux algorithmes de maillage consistent à associer des informations aux entités géométriques, notamment celles du maillage. Par exemple, une méthode par avancement de fronts en 2D nécessite de conserver les éléments du front et éventuellement des informations pour chaque élément du front comme les valeurs d'angle entre les segments ou la taille des mailles imposées localement.

Comme dans CGAL et GrAL, nous utilisons des structures de données adaptables à différents types pour le stockage de ces informations en se basant sur l'héritage et non les classes paramétrées, contrairement à ces bibliothèques. Ces structures sont parcourues à l'aide d'itérateurs généraux de manière à parcourir des structures éventuellement différentes avec les mêmes itérateurs, comme c'est le cas des *Collections* de Java (ou de la bibliothèque Standard Template Library, mais qui elle, utilise les *template* dont nous évitons l'usage). Les algorithmes de parcours sont réalisés uniquement à partir des itérateurs. Ces algorithmes sont donc génériques et peuvent être paramétrés à l'aide de fonctions de comparaisons des éléments, de sélection d'éléments et de traitements des éléments. De cette manière, les parcours sont dissociés des classes utilisées pour organiser les données. Par conséquent, l'organisation des données n'est pas plus détaillée ici car cet aspect a déjà fait l'objet d'autres travaux ([BER00], [STE94], [JCF])

Dans le paragraphe qui suit, nous nous intéressons à un exemple d'intégration d'un algorithme de maillage dans la BAMGS afin d'illustrer le patron de conception dans l'architecture.

IV.4.4 Exemple

Dans ce paragraphe sont présentés les différents modules de la BAMGS au travers de l'algorithme de Ruppert ([RUP95]). Cet algorithme a été choisi parce qu'il est simple, relativement connu, et qu'il permet d'illustrer les principes décrits dans ce chapitre. Brièvement, il consiste à construire une triangulation par raffinement jusqu'à satisfaire un critère de forme et de taille des éléments. Le principe du raffinement est de partir d'une triangulation initiale d'un englobant rectangulaire du domaine (plan) à mailler, puis de créer des points aux centres des cercles circonscrits aux triangles jugés mauvais. Chaque point est inséré par le noyau de Delaunay.

Après avoir présenté l'architecture générale des fonctions de maillage dans le premier paragraphe, nous voyons comment l'algorithme de Ruppert peut s'écrire sous la forme d'un algorithme générique et comment s'organisent les fonctions de maillage liées à cette méthode.

IV.4.4.a Architecture générale des fonctions de maillage

Les fonctions de maillages génériques et spécifiques se classent en quatre catégories, correspondant à quatre paquets de classes : les algorithmes génériques de maillage (paquet

AlgoGen), les configurations (paquet *Configurations*), les outils (paquet *Outils*) et les données de maillage (paquet *DonneesMaillage*).

Les données sont celles présentées dans le paragraphe précédent. Elles conservent les maillages ainsi que des informations diverses associées aux entités géométriques (entités du maillage ou autres).

Les configurations stockent toutes les configurations connues des algorithmes génériques de maillage, c'est-à-dire des classes ayant le même rôle que la classe *MaConfiguration* (diagramme IV.4-4) du paragraphe IV.4.3.c.

Enfin les outils sont des classes conservant les fonctions spécifiques (c'est-à-dire non génériques) comprenant des fonctions de bas niveau, mais aussi les algorithmes de maillage particulièrement originaux qui n'entrent dans aucun algorithme générique (comme la méthode du tissage de brins, [TAU96]). Dans la plupart des cas, les outils sont les fonctions servant à configurer les algorithmes génériques.

IV.4.4.b Algorithme de Ruppert ([RUP95])

Nous allons voir maintenant l'architecture de la BAMGS au travers de l'intégration dans la BAMGS, de l'algorithme de raffinement de Ruppert ([RUP95]), basé sur le critère de Delaunay.

L'algorithme IV.4-1 (a) est la formulation de la méthode de Ruppert tel qu'il l'a publiée. Pour l'intégrer dans la BAMGS, il est judicieux de la considérer comme un algorithme générique. Celui qui convient est celui des méthodes itératives, d'après le classement établi au chapitre 1. En effet, après une première phase d'initialisation qui consiste à construire une triangulation contrainte de Delaunay d'un englobant du domaine à mailler, des itérations améliorent l'état courant de la triangulation en appliquant des opérations locales. La méthode de Ruppert apparaît donc comme une *configuration* de l'algorithme itératif générique dont le principe est d'appliquer une succession d'opérations locales jusqu'à ce que l'ensemble satisfasse un certain critère.

```

Algorithme RaffinementDelaunay
Entree: graphe planeaire X;
           Angle minimum souhaité a
Sortie: triangulation de X, avec tous les angles > a.
Initialisation:
           Ajouter un englobant carré B à X
           Calculer les extrêmes de X : xmin, ymin, xmax, ymax
           Soit span(X) = max(xmax-xmin, ymax-ymin)
           Soit B le carré de côté 3 x span(X), centré en X
           Ajouter à X, les 4 segments de B
           Soit une liste de segments S = ensemble des arêtes de X
           Soit une liste de sommets V = ensembles des sommets de X
           Calculer la triangulation initiale de Delaunay DT (V)
répéter:
           Tant qu'un sommet s est envahi faire
               DecouperSegt(s)
           FinTantQ
           Soit t un triangle de mauvaise qualité (min angle < a)
           Soit p le centre du cercle circonscrit de t
           Si p envahi un des segments  $s_1, \dots, s_k$  alors
               pour i = 1 à k faire
                   DecouperSegt ( $s_i$ )
               FinSi
           Sinon
               DecouperTriangle(t) // ajouter p à V
           FinSi
Jusqu'à ce qu'aucun segment est envahi, et aucun angle < a
    
```

(a) Algorithme original du raffinement de Ruppert

```

Algorithme RaffinementdDelaunayRuppert
Entrée: Graphe planeaire X
           Angle minimum souhaité a
Sortie: triangulation de X, avec tous les angles > a.
// INITIALISATION
Calculer une enveloppe convexe triviale
Calculer une triangulation initiale de Delaunay
Stop = faux
DecouperSegtEnvahisToutSommet
Tant que ( NON Stop ) Faire
// CHOISIR UN LIEU
ChercherTriangleMauvaiseQualité (a,t, trouve)

// CHOISIR UNE OPÉRATION
Si (trouve) alors
    op=DecoupeRuppert
Sinon
    op=OperationNulle
FinSi

// APPLIQUER L'OPÉRATION
op.executer(t)

// TESTER LA CONVERGENCE
Stop = NON trouve
FinTantQ
Fin

Algorithme DecoupeRuppert
Entrée: triangle t, Triangulation de Delaunay courante
Sortie: Triangulation de Delaunay courante
    p = CentreCercleCirconscri(t) ;
    if ( NON SegmentEnvahi (p,  $s_1, \dots, s_k$ ) ) alors
        DecouperTriangle(t)
    FinSi
    DecouperSegtsEnvahisUnSommet(p)
Fin
    
```

(b) Réécriture du raffinement de Ruppert dans l'algorithme générique itératif

algorithme IV.4-1. Version originale (a) et modifiée (b) du raffinement de Ruppert

Il y a deux opérations locales dans l'algorithme de Ruppert : 1) une division récursive de segments dits « envahis » par un sommet, et 2) la division d'un triangle de mauvaise qualité par création d'un nouveau point inséré dans la triangulation par le noyau de Delaunay. Un segment est dit envahi par un sommet si le cercle ayant ce segment pour diamètre contient ce

sommet. Le processus est itéré jusqu'à ce qu'aucun segment et aucun triangle ne soit découpé. L'algorithme IV.4-1 (b) présente la vue de cet algorithme relativement à l'algorithme générique itératif. Les différentes phases apparaissent en commentaire. Le calcul d'une triangulation initiale sans point interne est un algorithme de maillage qui suit le principe de l'algorithme générique itératif. Par conséquent elle apparaît aussi comme une configuration. Par opposition, le test de la convergence est trivial puisqu'il ne consiste qu'en une expression booléenne simple. Par conséquent, l'implémentation de cette étape n'est autre qu'une *transformation* (au sens du paragraphe IV.4.3.b) et est donc implémentée par la méthode d'initialisation des données de l'étape chargée du test de convergence. Les autres étapes sont implémentées par des outils.

Les fonctions de maillage, c'est-à-dire l'ensemble des fonctions (génériques ou spécifiques) répertoriées par la base de données des propriétés, qui sont concernées par cet algorithme sont représentées par le diagramme IV.4-5 et le diagramme IV.4-6. Le diagramme IV.4-5 montre les paquets situés au haut de la hiérarchie des paquets de classes des fonctions de maillage. Le diagramme IV.4-6 est un détail du paquet *Outils*.

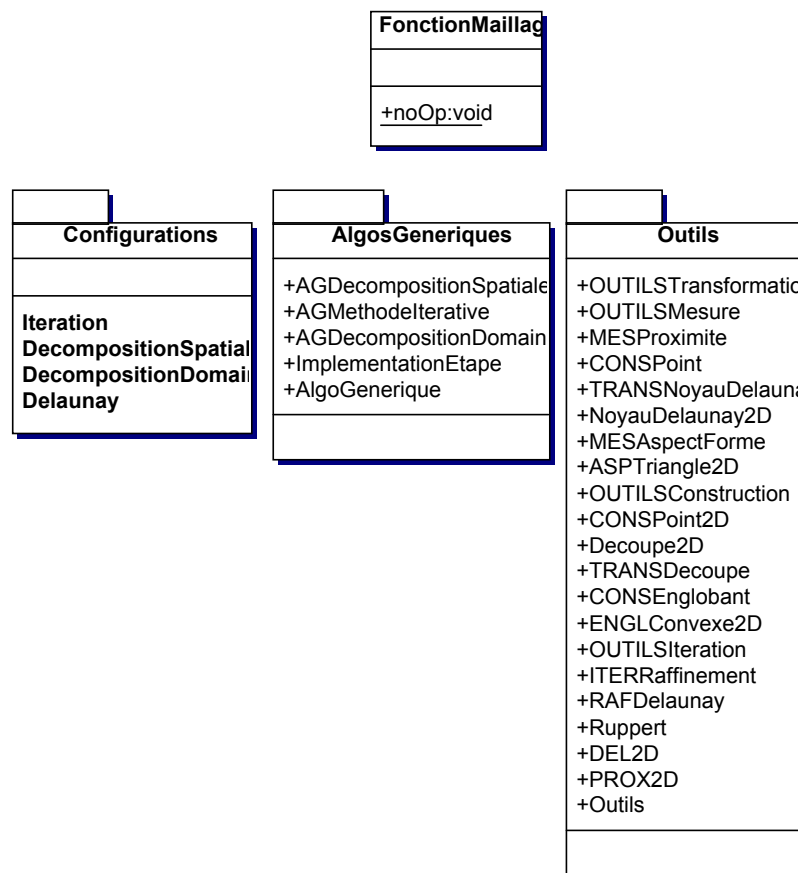


diagramme IV.4-5. Fonctions de maillage

Quatre types d'outils sont présents : OUTILSTransformation, OUTILSMesure, OUTILSConstruction et OUTILSIteration. OUTILSTransformation détient les méthodes concernant une transformation topologique et géométrique comme par exemple l'application d'un motif ou des fonctions nécessaires au noyau de Delaunay (construction de la cavité et de la boule notamment, [GEO97]). Les méthodes de mesure de qualité, les métriques, la définition de la notion de distance pour les espaces non euclidiens, etc. sont situées dans le paquet OUTILSMesure. Dans le paquet OUTILSConstruction se trouvent les méthodes de construction de lieux géométriques : points, enveloppes convexes, sphères circonscrites, triangle idéal selon certains critères, etc. OUTILSIteration contient les fonctions simples qui ne relèvent pas directement de l'algorithmique géométrique mais qui sont spécifiques à l'algorithme générique itératif. C'est le cas de la fonction choixOperation qui choisit la fonction à appliquer localement en fonction du contexte. Dans le cas de la méthode de Ruppert, elle est très simple (algorithme IV.4-1 (b)).

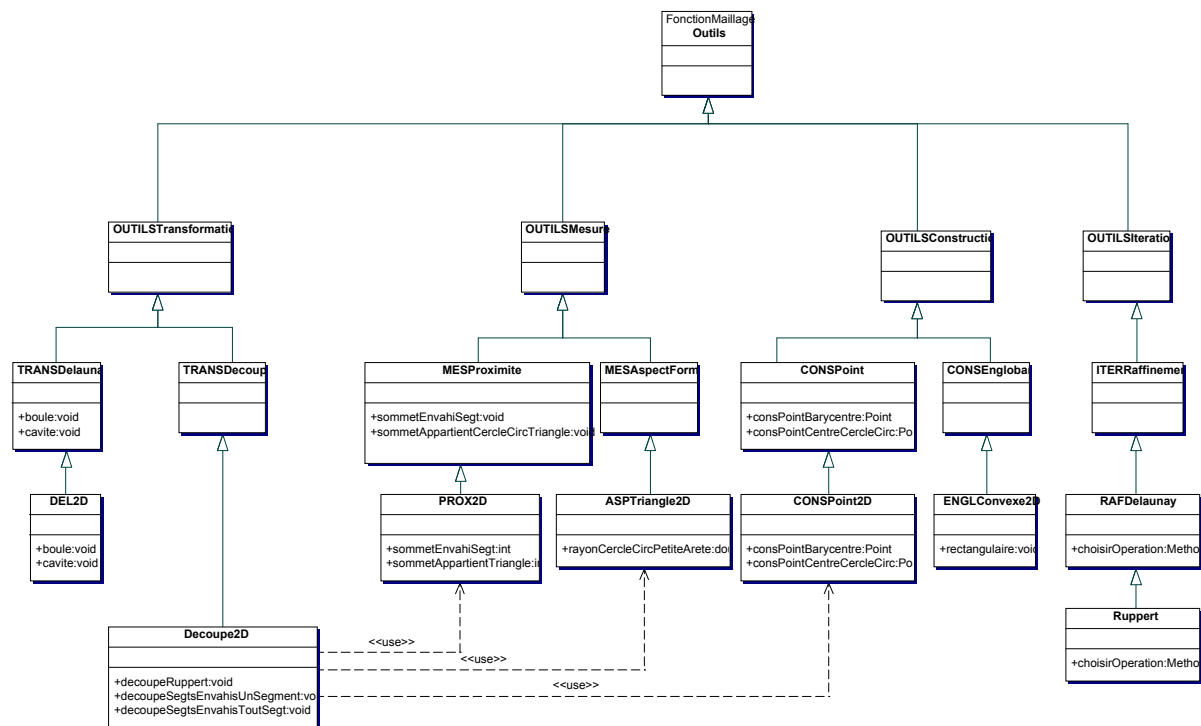


diagramme IV.4-6. Le paquet Outils

Observons deux étapes appelées depuis l'algorithme générique et utilisant des données communes : *chercherTriangleMauvaiseQualite()* et *op.executer()*. La première prend en entrée la valeur du seuil d'angle a , la triangulation et renvoie le triangle t ayant un angle inférieur à a ainsi qu'un booléen *trouve* indiquant si un tel triangle a été trouvé. La seconde exécute l'opération à effectuer localement, qui consiste ici en la méthode *DecoupeRuppert()*

qui accepte en entrée le triangle à découper ainsi que la triangulation et restitue les triangles ajoutés ou modifiés. Une référence de chaque donnée utilisée localement, comme par exemple le plus mauvais triangle t , ou les données passées en paramètre de l'algorithme générique, telles que la triangulation ou l'angle seuil a , sont placées dans une structure de données générique similaire à un tableau. La méthode d'initialisation des données de l'étape de choix du lieu en extrait les données nécessaires à l'appel de *chercherTriangleMauvaiseQualite()*, c'est-à-dire l'angle seuil, la triangulation, le plus mauvais triangle et le booléen. De même, la méthode d'initialisation des données de l'étape qui exécute l'opération correspondant au traitement local, extrait les données nécessaires à l'appel de *DecoupeRuppert()*, c'est-à-dire le plus mauvais triangle, la triangulation, ainsi que les triangles ajoutés ou modifiés. La structure générique ne contenant que des références vers les données, il n'est pas nécessaire de la mettre à jour après l'initialisation du plus mauvais triangle par *chercherTriangleMauvaiseQualite()*.

Nous constatons dans l'algorithme de Ruppert que les différentes parties d'un algorithme comme la mesure de la qualité d'un élément, la stratégie de création de points, le test de convergence ou le noyau de Delaunay peuvent être remplacées. Par exemple, il existe d'autres stratégies de création de points. Les points peuvent être créés au centre des triangles jugés mauvais ([WEA94]) ou encore sur les segments de Voronoï¹ ([REB93]) en cherchant à respecter une contrainte de taille. Les étapes de ces algorithmes peuvent réutiliser des implémentations déjà utilisées par ailleurs. Les parties innovantes sont alors la stratégie de création de points, éventuellement le test de convergence, et les méthodes d'initialisation des données permettant l'enchaînement des implémentations au sein de l'algorithme générique. Par ailleurs, certains algorithmes de maillage combinent l'approche Delaunay avec l'approche par avancement de fronts, notamment pour la construction de maillages tétraédriques [GEO96]. C'est un cas typique où la réutilisation de configurations est intéressante. La configuration est en effet celle d'une technique d'avancement de fronts pour un maillage tétraédrique à la différence près que la manière d'insérer le point dans le maillage, c'est-à-dire la manière de mettre à jour les fronts, consiste en plus en l'application du noyau de Delaunay en 3D.

¹ Un segment de Voronoï est le segment reliant les centres des cercles circonscrits à deux triangles respectivement.

IV.5 Exemple

Considérons que la construction et la modification d'un maillage pour une AEM de visualisation réaliste. L'analyse fonctionnelle fournit des spécifications relatives à cette AEM : le maillage doit être *dépendant du point de vue* et *minimiser le nombre de triangles* afin de réduire les temps de calculs liés à la lumière (radiosité) tout en maintenant une qualité visuelle satisfaisante.

L'opérateur invoque une requête de maillage auprès du système de conception en précisant le contexte : l'AEM est une application de rendu réaliste, le maillage est dépendant du point de vue, et le nombre de triangles faible. Le système de conception traduit les spécifications en propriétés de maillage : le maillage doit être triangulaire surfacique, le contrôle concerne la forme et la taille des mailles par une métrique. Il construit également cette métrique à partir de l'erreur d'approximation, de l'orientation des triangles par rapport à la caméra (seuls les triangles visibles par la caméra sont soumis à un raffinement induit par le 3^{ème} contrôle de la métrique à savoir :), la taille des triangles à l'écran.

Le système de conception injecte ensuite les propriétés dans la BAMGS. La BAMGS effectue en interne une sélection des fonctions capables de satisfaire ces propriétés et renvoie les références de ces fonctions au système de conception ainsi que l'ensemble de leurs propriétés (qui est un sur-ensemble de l'ensemble des propriétés injectées).

Le système de conception les classe en fonction de la possibilité de fournir les valeurs des paramètres de contrôle, des garanties qu'elles offrent, et du fait qu'elles ne cherchent pas à contrôler des propriétés inutilement contraignantes (par exemple la direction des éléments est inutile dans ce cas). Les mieux classées, (celles qui peuvent être paramétrées) sont soumises à l'opérateur s'il le souhaite et son choix est retenu. Le système de conception invoque alors une méthode (au sens programmation objet) d'initialisation sur la référence de la fonction de maillage retenue. Si cette fonction est une configuration d'algorithme générique, alors c'est à cet instant que la configuration a lieu (sinon s'il s'agit d'une fonction spécifique, rien de particulier n'a lieu). Ensuite le système de conception invoque la fonction de maillage via la référence en fournissant la métrique et récupère en retour le maillage. Il est envoyé à l'AEM qui réalise sont traitement.

Supposons que la position de la caméra change après ce premier traitement, induisant une modification de la métrique et donc de tout le maillage. Il faut donc déterminer une fonction de maillage capable de modifier le maillage à partir de la nouvelle métrique.

Il y a deux scénarii possibles selon qu'il impute soit au système de conception, soit à l'AEM, le choix de la méthode, en fonction des connaissances qui sont présentes dans chacun de ces systèmes.

Considérons dans un premier temps que ce choix soit effectué par le système de conception. Alors l'AEM doit effectuer une requête de modification de maillage auprès du système de conception en fournissant la nouvelle métrique, ainsi que le maillage. Le système de conception détermine une fonction d'adaptation de maillage de la même façon qu'il a déterminé une fonction pour la construction, en s'aidant des mêmes propriétés. Il initialise et invoque cette fonction, puis renvoie le maillage modifié à l'AEM.

Dans le second scénario, l'AEM dispose des connaissances permettant de faire le choix de la fonction d'adaptation de maillage. Alors elle doit être capable de traduire ses spécifications sur le maillage en propriétés compréhensibles par la BAMGS. Il peut alors effectuer le choix d'une méthode de la même manière que le système de conception. La différence réside dans la gestion du modèle géométrique qui été explicitée au chapitre 2 (conservation du modèle au format natif, transfert du nom du fichier du système de conception à l'AEM, nécessité de transférer l'association entre les identifiants du modèle géométrique et ceux du maillage, etc.)

IV.6 Discussion

IV.6.1 Patron de conception

Le patron de conception proposé permet une représentation des algorithmes génériques en évitant les classes paramétrées, (template) sur lesquelles sont basés GrAL et CGAL. Cela permet d'utiliser d'autres langages, en particulier Java qui est de plus en plus utilisé pour sa portabilité multi-systèmes et même multi-plateformes sans recompilation. La portabilité du code joue un rôle important dans l'interopérabilité des systèmes mais est très difficile à atteindre aujourd'hui et le fait d'élargir le panel des langages permettant la programmation générique est un atout. Le patron de conception proposé présente toutefois un petit inconvénient : il ne peut pas manipuler directement des types de base (entier, caractère, booléen, etc.). Ces derniers doivent être encapsulés dans des classes car toute variable manipulée doit être un objet.

Le patron de conception engendre aussi une légère perte de performances due en particulier à l'appel dynamique et aux méthodes d'initialisation des données, qui doivent extraire les paramètres avant un appel de fonction. Cette perte reste cependant négligeable parce que les algorithmes génériques susceptibles d'induire une perte de performances importante sont des fonctions qui sont appelées peu souvent. Les fonctions les plus utilisées sont soit des fonctions spécifiques, soit des fonctions génériques dont l'extraction des données est immédiate, comme c'est le cas pour le noyau de Delaunay. La transformation des données, quant à elle, n'engendre pas de surcoût puisqu'elle doit être faite de toute manière même dans une version non générique de l'algorithme.

IV.6.2 BAMGS

Les propriétés associées à chaque fonction de maillage permettent au système appelant de réaliser une sélection parmi les fonctions restituées par la BAMGS en réponse à une requête. Elles aident également à la maintenance et à l'évolution du système en fournissant à l'utilisateur chargé d'ajouter un algorithme, des informations pertinentes sur les fonctions existantes. Les algorithmes génériques possèdent des propriétés indéfinies, contrairement à leurs configurations. Cela évite d'induire l'utilisateur de la BAMGS en erreur car pour un même algorithme générique, les propriétés varient d'une configuration à une autre. Ainsi il apparaîtra que les pré-requis de la configuration 2D d'un algorithme de maillage ne sont pas les mêmes que ceux de la configuration 3D.

Par ailleurs, le patron de conception possède la faculté de réaliser dynamiquement le choix d'une fonction implémentant une étape permet de remplacer l'implémentation d'une étape par une autre. Cette tâche requière malgré tout, la compilation de la méthode d'initialisation des données associée. Une application intéressante de cette faculté est de permettre la construction de la configuration d'un algorithme générique par une interface graphique puisque seule une très petite partie du code doit être explicitement écrite et compilée. Pour atteindre cet objectif, nous envisageons de représenter les méthodes d'initialisation de données sous la forme de fonctions paramétrables par : 1. les fonctions de transformation et 2. par un système de désignation des variables dans la structure de données générique permettant de récupérer cette variable.

En allant plus loin, la sémantique des étapes dans les algorithmes génériques et la faculté d'associer dynamiquement des implémentations d'étapes permet d'envisager un contrôleur qui assurerait la cohérence des implémentations de chaque étape. Par exemple, en vérifiant

que les post-conditions de l'implémentation d'une étape sont cohérentes par rapport aux pré-conditions de l'implémentation de l'étape suivante. De ce fait, une autre piste à prendre en considération est de conditionner le choix de l'implémentation d'une étape par le choix de l'implémentation d'une autre étape. Par exemple, pour l'algorithme générique des méthodes itératives, le test de convergence dépend souvent de la stratégie de création d'une nouvelle maille. La prise en compte de ces aspects permettrait de faciliter l'ajout de nouveaux algorithmes dans la BAMGS. Ces progrès exigent entre autres que nous étudions ce que peuvent apporter les travaux en spécification formelle.

Même si un tel contrôleur n'est pas encore étudié, la base de données des contraintes et le patron de conception pour les algorithmes génériques facilitent d'ores et déjà l'intégration de nouveaux algorithmes de maillage.

IV.6.3 Mise en oeuvre

Remarquons que la mise en oeuvre complète d'un tel système nécessite des efforts de programmation importants. Au départ, il faut développer tous les outils de base. Ceci fait, il faut réécrire les algorithmes existants sous la forme d'une configuration d'un algorithme générique à l'aide de ces outils. Plus le développement de la BAMGS est avancé, plus les chances de pouvoir réutiliser des modules existants augmentent. Il n'en reste pas moins que la programmation générique s'adresse à des développeurs expérimentés. L'écriture du code est quelque peu alourdie par les vérifications nécessaires à tout appel dynamique de fonction. De plus, toutes les données manipulées sont génériques, ce qui a parfois pour conséquence de rendre les erreurs d'incompatibilités de types difficilement décelables. Enfin, la mise au point est un peu plus difficile également, puisque le code d'un algorithme générique, par définition, n'est pas explicite.

Par ailleurs la gestion des communications, abordée au deuxième chapitre, nécessite également des efforts de programmation. Il faut gérer en particulier les appels distants depuis le module d'assemblage du système de conception vers la BAMGS (invocation d'une fonction de maillage), ainsi que les appels de la BAMGS vers le modeleur (invocation des services géométriques).

- Ensuite, bien que les services géométriques nécessaires soient peu nombreux par rapport au nombre d'algorithmes de maillage, leur mise en correspondance avec les services

géométriques disponibles nécessite des efforts dans le cas de l'utilisation d'un modelleur géométrique actuel. En effet, les services nécessaires ne sont pas forcément fournis.

- Le développement pour la mise en correspondance des services géométriques peut paraître contraire à la généralité que nous cherchons à atteindre puisque, dans un certain sens, la BAMGS devient spécialisée pour un modelleur. Cependant, dans le chapitre 2, nous avons vu les difficultés rencontrées lorsqu'on s'efforce de réaliser l'interopérabilité entre les systèmes en échangeant des versions normalisées de modèles BRep. L'indépendance de la BAMGS par rapport aux modèles géométriques, et donc sa généralité, tient justement à l'accès aux services géométriques nécessaires. Par exemple, les normes issues de la CAO ne prennent pas en compte les surfaces implicites tels les squelettes ponctuels qui tirent leur origine du domaine de la synthèse d'image. Il est donc nécessaire de développer les interfaces nécessaires de manière à maintenir le module de maillage indépendant de la géométrie.
- De plus ces efforts offrent une assistance à la spécification de maillage. Tout d'abord les liaisons qui existent entre les propriétés et les fonctions de maillage, permettent 1) d'aider l'utilisateur à choisir un algorithme de maillage adéquat et 2) de choisir automatiquement l'algorithme le plus adapté si le système de conception le permet. Ensuite, la généralité présente dans la BAMGS permet de restreindre l'ajout de code aux parties réellement innovantes sans remettre en cause l'architecture, ni les algorithmes déjà présents. Elle permet par ailleurs de combiner des outils de maillage afin de construire et tester de nouveaux algorithmes.

IV.6.4 Module d'assemblage et généralité

Nous avons vu dans le deuxième chapitre que l'algorithme général d'assemblage est similaire pour différents BRep. Ce patron de conception pourrait donc s'appliquer à l'assemblage et les algorithmes d'assemblage seraient alors intégrés à la BAMGS. Il y aurait alors deux algorithmes génériques d'assemblages : un pour les BRep et un pour les CSG. L'accès à différents modelleurs d'un même type se ferait alors par le même algorithme générique configuré différemment pour un modelleur et pour un autre.

Cette solution n'a pas été retenue pour plusieurs raisons :

- ❑ Ces configurations s'avèrent inutilement lourdes devant les travaux sur les accès unifiés aux BRep ([HAI98], [TAU00]) qui évitent tout traitement supplémentaire pour utiliser avec un modéleur, un algorithme d'assemblage conçu pour un autre modéleur.
- ❑ Les algorithmes d'assemblages doivent, dans leur déroulement, exploiter les informations de haut niveau sémantique lorsqu'elles sont prises en compte dans le système de conception. Or comme il n'existe pas encore de protocole d'échange d'information de cette nature, l'algorithme qui les utilisent est dépendant du système de conception ou au moins de la base de connaissance.

Conclusion

Ce chapitre a présenté une architecture originale d'une bibliothèque d'algorithmes de maillage, capable de choisir un algorithme de maillage à partir de propriétés imposées par l'AEM ou plus généralement à partir des besoins de l'utilisateur en général. Elle est conçue en deux grandes parties : l'une gère les différentes contraintes qui dirigent le choix d'un algorithme, l'autre conserve judicieusement ces algorithmes.

Le patron de conception proposé pour la prise en compte de la généricité est également un aspect innovant parce qu'il offre à la BAMGS une grande extensibilité et constitue un premier pas vers la configuration automatique des algorithmes de maillage.

Les efforts de programmation nécessaires à la réalisation d'un tel système sont évidemment importants parce que :

- ❑ malgré la généricité, les algorithmes de maillage (en particulier les outils) sont nombreux
- ❑ l'interfaçage de la BAMGS avec les autres systèmes ne peut pas se limiter à l'utilisation des normes d'échanges actuelles si l'on veut à la fois éviter les importants problèmes d'échanges de données géométriques et assurer une indépendance par rapport à ces systèmes.

L'analyse de la BAMGS présentée ici peut encore être affinée car elle nécessite des connaissances très approfondies dans différents domaines : l'algorithmique géométrique, le génie logiciel, l'interopérabilité entre systèmes, les normes d'échanges, etc. Certains aspects, plus critiques, comme les algorithmes de maillage ou les concepts de programmation objets, ont été plus approfondis que d'autres. Il était cependant nécessaire d'aborder tous ces aspects

afin d'avoir une vue globale du problème. Cela a permis de prendre en considération les différents éléments ayant une influence sur la conception d'une bibliothèque d'algorithmes de maillage.

Chapitre V : Mise en oeuvre

V.1 Objectifs

Nous avons réalisé une maquette illustrant certains concepts proposés dans cette étude, notamment ceux présentés dans le deuxième chapitre. Elle comporte un modèle géométrique, un module d'assemblage, ainsi qu'une version réduite de la BAMGS. Pour réduire les développements, seuls des maillages de surface sont construits. L'objectif de la maquette est de montrer que notre architecture est réalisable et qu'elle permet d'obtenir des maillages de bonne qualité géométrique, qui peuvent être utilisés par les trois AEM déjà citées. En particulier, l'architecture :

- ne doit pas reconstruire le modèle géométrique exact, mais interroger le modelleur géométrique (au travers de ce que nous avons appelé les « services géométriques »)
- doit permettre de pré-optimiser les maillages en donnant la possibilité à l'utilisateur d'entrer les propriétés de maillage pour chaque partie du solide,
- doit construire des maillages de bonne qualité géométrique (pas de fissure, pas de recouvrement et pas de maille inversée),
- doit montrer qu'il est effectivement possible de construire des maillages ayant toutes ces qualités directement à partir d'un modèle géométrique de type CSG. Pour cela, elle doit utiliser la méthode proposée dans le chapitre 2, consistant à mailler l'arbre des feuilles à la racine.

Cette maquette ne vise pas à présenter les parties qui nécessiteraient un temps de développement déraisonnable pour être démonstratives, c'est-à-dire la programmation

générique ainsi que la gestion de la connaissance servant à déduire les propriétés de maillage des informations fonctionnelles. Cette dernière partie est représentée de manière très simplifiée : il y a une méthode de maillage différente pour chaque type d'AEM (visualisation, fabrication par prototypage rapide ou simulation).

Nous ne nous sommes pas non plus intéressés à la partie transfert des informations entre les modules (cf. II.1.5). Dans cette maquette, les transferts d'un module à un autre se font par appels de procédures au sein d'un même processus (au sens informatique).

Le modèleur utilisé pour conserver les entités de maillage et pour réaliser les opérations booléennes est ACIS, dont les fonctionnalités sont accessibles via son interface de programmation (API) pour le langage C++. La maquette est donc écrite en C++ également. Par ailleurs, elle s'intègre dans un projet de l'équipe comptabilisant un peu plus de 17 000 lignes de code dont 6600 proviennent de cette maquette réparties en 2000 pour le modèleur géométrique CSG (ses primitives et les surfaces), environ 3300 pour la BAMGS et 1300 pour le module d'assemblage et la création des solides.

Dans les sections qui suivent, nous voyons les techniques de maillage mises en œuvres, l'architecture générale (les modules et les scénarios de construction des maillages), les services géométriques requis (et implantés) de la part du modèleur et les résultats.

V.2 Techniques de maillage

Nous avons choisi de fournir un maillage constitué de polygones convexes à l'AEM de visualisation temps réel, ce qui correspond au format d'entrée de la bibliothèque d'affichage OpenGL utilisée dans cette maquette. Vraisemblablement, ces polygones sont remaillés en interne par la bibliothèque qui exploite au mieux le matériel afin d'accélérer une partie de ses traitements.

Pour le prototypage rapide, le maillage construit est triangulaire. Il est simplement dirigé par l'erreur de corde, la forme et la taille des triangles ne sont pas considérées. Il est en effet plus important, pour la durée et la fiabilité de l'algorithme de découpage, de minimiser le nombre d'entités géométriques (faces, arêtes, sommets) de manière à réduire le temps imparti au tranchage, à la reconstruction du modèle, ou encore au transfert via un réseau.

Concernant le calcul par éléments finis, il faut tenir compte de l'erreur de corde, de la forme des triangles et de la taille de maille souhaitée. La prise en compte de l'erreur de corde se fait

par une paramétrisation adéquate des surfaces. La qualité en forme est améliorée par le schéma de raffinement de Ruppert. Les paramètres de la fonction de maillage sont la valeur du plus petit angle, ainsi que la taille minimale et la taille maximale des arêtes. Une valeur d'angle qui donne des résultats satisfaisants est 25° .

La version originale de l'algorithme de Ruppert consiste à déterminer en premier un englobant du domaine à mailler (2D) et à lui appliquer une triangulation de Delaunay, sans points internes à la face à mailler. Ensuite les triangles ne respectant pas un critère de qualité basé à la fois sur la forme et la taille des triangles sont découpés par insertion d'un nouveau nœud au centre du cercle circonscrit au triangle considéré ou bien sur une arête de la frontière si le nouveau nœud est trop proche de celle-ci. La version implémentée ici construit donc une première triangulation de Delaunay contrainte, puis le schéma de raffinement est appliqué sur les triangles de mauvaise qualité jusqu'à ce que la qualité soit satisfaisante ou bien que la taille minimale d'arête ait été atteinte.

Une attention particulière a été prise pour les surfaces cycliques de manière à ce qu'il n'apparaisse pas de « couture » indésirable dans le maillage. Ainsi, la mesure de la longueur d'un segment nécessite de vérifier si ce segment est coupé par la limite de l'espace paramétrique ou non et de déplacer un des deux points d'une valeur d'une période. Un traitement similaire est utilisé pour le calcul de la distance d'un point à une droite ou à un segment, les calculs d'angle, la création d'un point sur un segment et la construction d'un cercle circonscrit à trois points.

V.3 Architecture implémentée

Le **modèle géométrique** que nous avons retenu est un arbre CSG afin d'implanter l'**algorithme d'assemblage** dédié aux arbres CSG qui fait l'objet du chapitre II. En plus des paramètres évidents des primitives, on conserve une définition des surfaces qui les limitent. Ces informations associées à chaque feuille permettent de réaliser tous les services géométriques dont la BAMGS a besoin (voir V.4).

Le fait que nous utilisions aussi ACIS, qui est un modèle BRep, ne doit pas induire en erreur : on ne fait pas l'hypothèse que le modèle géométrique du logiciel de CAO est un BRep et on ne reconstitue pas non plus un modèle BRep de la pièce à mailler. ACIS n'est là qu'en tant que structure de données pour stocker le maillage produit et pour réaliser les opérations booléennes sur les maillages.

Les interactions entre le modèle géométrique, le module d'assemblage et la BAMGS respectent le scénario présenté au chapitre II à la différence près que l'utilisateur, et non le système, détermine les propriétés du maillage de chaque partie du solide (erreur de corde ou taille des arêtes et angle minimum). Ce scénario est le suivant : l'utilisateur commande un maillage auprès du module d'assemblage en précisant l'AEM visée (visualisation, prototypage rapide ou simulation). Le **module d'assemblage** offre une fonction de maillage pour chaque AEM ; il y a en donc 3. Toutes fonctionnent sur le même principe : on combine les maillages des fils d'un nœud de l'arbre pour obtenir récursivement celui de la racine. Quelle que soit l'AEM, les *objets primitifs* sont maillés d'abord par un maillage à base de faces convexes. Dans cette maquette, cette fonction de maillage est spécifique à chaque objet primitif. La BAMGS construit le maillage en créant les entités géométriques dans ACIS (faces, contours, arêtes et sommets). Sur les arêtes naturelles du solide, c'est-à-dire là où deux surfaces se coupent, les sommets sont créés en tenant compte de l'erreur de corde (cas de la visualisation et du prototypage rapide) ou de la taille des arêtes (cas de la simulation). S'il s'agit d'un maillage pour le prototypage rapide, le maillage par faces convexes obtenu est triangulé. S'il s'agit de simulation, le maillage triangulaire obtenu est raffiné par le schéma de Ruppert de manière à contrôler la forme des triangles (cf. V.2).

Les maillages des primitives sont ensuite combinés par opérations booléennes, puis des post-traitements sont effectués pour que le maillage obtenu satisfasse les propriétés souhaitées. De manière générale, ce post-traitement consiste à :

- rendre les mailles conformes aux propriétés, c'est-à-dire les rendre convexes pour la visualisation et triangulaires pour le prototypage rapide et la simulation,
- lisser les sommets sur les intersections de manière à améliorer la répartition des nœuds et à réduire l'erreur d'approximation aux sommets,
- décimer les triangles plats,
- pour la simulation, raffiner le maillage si nécessaire par le schéma de Ruppert (cf. V.2)

Le lissage consiste en une heuristique qui a pour but d'homogénéiser la répartition des sommets sur l'intersection des opérands. Pour cela, un traitement assimilable au lissage de Phong est utilisé pour donner un aspect continu (ou *lisse*) à un ensemble de triangles

approchant une surface gauche, lors du calcul des couleurs d'une image. D'autres solutions que celle proposée sont possibles. Nous avons retenue celle-ci pour la simplicité de ses calculs et parce qu'elle cherche à se rapprocher des surfaces ce qui n'est pas immédiat avec des méthodes plus traditionnelles tels que des Béziérs ou des B-splines.

Initialement le lissage de Phong consiste à évaluer la normale en chaque sommet du maillage, puis les normales internes à chaque maille sont calculées par interpolation des normales aux sommets. La normale en chaque point est ensuite utilisée pour déterminer la couleur à l'écran (qui est proportionnelle à l'angle entre la normale au point et le vecteur de la lumière incidente).

Dans la situation présente, il ne s'agit pas de calculer une couleur mais une position de manière à ce que la ligne d'intersection semble plus lisse, c'est-à-dire plus régulière. Nous voulons en fait déplacer chaque sommet S au milieu de ses deux voisins sur la ligne d'intersection, S_{Prec} et S_{Suiv} , tout en essayant de se rapprocher de l'intersection exacte. Pour que la nouvelle position soit au « milieu » de ses voisins, le point de départ n'est pas S , mais le point de la ligne (polygonale) d'intersection à équidistance des voisins, nommé S_M . La normale en S_M est alors calculée par interpolation des normales à ses deux voisins (ici S et S_{Prec}). Le déplacement de S_M est déterminé par la plus petite distance (comparée en valeur absolue) de S_M à chacune des deux surfaces. La nouvelle position d'un point S est calculée de la manière suivante :

$$S_N = S_M + e_{SM} \cdot (\alpha \cdot \vec{N}_{Prec} + (1 - \alpha) \cdot \vec{N}_S)$$

avec :

- N_{Prec} et N_S , sont des normales, chacune interpolée linéairement à partir des normales à chaque surface respectivement aux points S_{Prec} et S^l
- S_M , le point sur la ligne polygonale d'intersection qui est équidistant à S_{Prec} et S_{Suiv} .
- e_{SM} la plus petite distance orientée (comparée en valeur absolue) de S_M à chacune des deux surfaces.
- α est tel que¹ $S_M = \alpha \cdot S_{Prec} + (1 - \alpha) \cdot S$

¹ N_{Prec} est à remplacer par N_{Suiv} si S_N est sur $[S S_{Suiv}]$

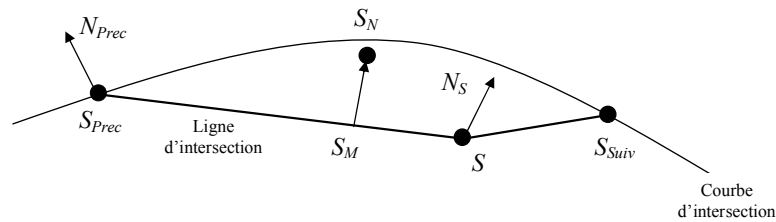


figure V.3.1 Méthode de lissage

Dans la BAMGS, la création de points et la création de certains objets géométriques tels que des cercles circonscrits à des triangles, ainsi que certaines mesures (distance entre deux points le long d'une courbe sur la surface) nécessitent l'interrogation du modèle géométrique, c'est-à-dire l'arbre CSG et notamment les surfaces qu'il conserve.

V.4 Services géométriques

L'interface géométrique est constituée des services que doit rendre le modèleur géométrique (l'arbre CSG) à la BAMGS pour qu'elle puisse construire les maillages. Cette interface comporte des fonctions de consultation et des fonctions de construction.

Ces fonctions sont pour la plupart des fonctions en 2D. La raison à cela est qu'il y a essentiellement deux approches pour construire un maillage surfacique : directement en 3D ou bien en 2D dans un espace paramétrique, suivi d'un plongement du maillage obtenu dans l'espace réel. C'est la deuxième solution qui a été retenue car les objets primitifs sont composés de surfaces pour lesquelles une représentation paramétrique est disponible, permettant ainsi de réaliser les opérations géométriques en 2D. De plus, certaines surfaces, comme les surfaces cylindriques par exemple, sont périodiques. Elles nécessitent alors des traitements en 2D spécifiques, comme le calcul de distance entre deux points nécessaires à certaines opérations géométriques dans l'espace paramétrique.

Fonctions de consultation :

```
- void uvAxyz (Identifiants IdSurf, const Points2D &, Points3D & )
```

Conversion des coordonnées paramétriques d'un point en ses coordonnées réelles.

```
- void xyzAuv (Identifiants IdSurf, const Points3D &, Points2D & )
```

Conversion des coordonnées réelles d'un point en ses coordonnées paramétriques.

```
- double distanceParam(Identifiants IdSurf, const Points2D&,
                        const Points2D& )
```

Renvoie la distance, dans l'espace paramétrique, entre deux points P_1 et P_2 (certaines surfaces sont cycliques, la distance retenue est donc la plus petite)

```
- double longueurGeodesique (Identifiants IdSurf, const Points2D& P1,
                             const Points2D& P2 )
```

Renvoie la longueur de la courbe qui est l'image du segment $[P_1P_2]$

```
- double distancePointSegment (Identifiants IdSurf, const Points2D& P,
                               const Points2D& O,
                               const Points2D& E )
```

Renvoie la distance, dans l'espace paramétrique d'un point P à un segment $[OE]$ (certaines surfaces sont cycliques, la distance retenue est donc la plus petite, voir figure V.4.1)

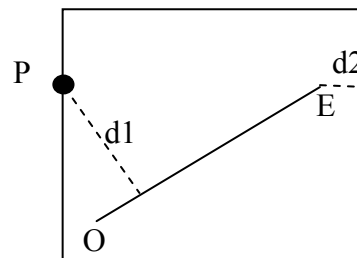


figure V.4.1 Distance point-segment dans l'espace paramétrique

```
- double distancePointDroite (Identifiants IdSurf, const Points2D& P,
                              const Points2D& O,
                              const Points2D& E )
```

Renvoie la distance d'un point P à une droite (OE) dans l'espace paramétrique. (Même remarque que ci-dessus concernant les surfaces cycliques)

```
- bool pointAppartientSegment (Identifiants IdSurf, const Points2D& P,  
  
                                const Points2D& O,  
  
                                const Points2D& E )
```

Teste si un point P appartient à un segment [OE] dans l'espace paramétrique.

```
- double angleSecteurParam (Identifiants IdSurf, const Points2D& O,  
  
                            const Points2D& A,  
  
                            const Points2D& B )
```

Calcule la valeur de l'angle en O d'un secteur AOB dans l'espace paramétrique (Même remarque que ci-dessus concernant les surfaces cycliques)

```
- double distanceOrientee (Identifiants IdSurf, const Points3D& )
```

Renvoie la distance entre un point et une surface (ou une approximation). Le résultat est positif si le point est du côté matière. (Même remarque que ci-dessus concernant les surfaces cycliques)

```
- double erreurArete (Identifiants IdSurf, const Points2D& O,  
  
                     const Points2D& E )
```

Renvoie l'erreur de corde d'une arête (ou une majoration).

```
- double erreurTriangle (Identifiants IdSurf, const Points2D& P1,  
  
                        const Points2D& P2  
  
                        const Points2D& P3 )
```

Renvoie l'erreur d'approximation d'un triangle (ou une majoration de celle-ci).

Fonctions de construction :

```
- Points2D creerPointSegment (Identifiants IdSurf, const Points2D& O,  
  
                              const Points2D& E,  
  
                              double alpha )
```

Crée un point dans l'espace paramétrique valant : $\alpha \cdot O + (1-\alpha) \cdot E$. (Même remarque que ci-dessus concernant les surfaces cycliques)

```
- void cercleCirconscritTriangle (Identifiants IdSurf, const Points2D& P1,  
  
                                const Points2D& P2,  
  
                                const Points2D& P3,  
  
                                Points2D& Centre,  
  
                                double& Rayon )
```

Construit le cercle (défini par un rayon et un centre) circonscrit à trois points P_1 , P_2 et P_3 dans l'espace paramétrique. (Même remarque que ci-dessus concernant les surfaces cycliques)

La différence essentielle de cette interface avec les interfaces existantes servant d'accès unifié à un BRep est que d'une part, elle ne concerne pas exclusivement un BRep, et que d'autre part, les fonctions concernent des mesures sur des entités géométriques (distances, angles, appartenance), et des constructions d'entités géométriques (points et cercles). Ce sont ces dernières qui permettent d'éviter la reconstruction du modèle géométrique au niveau de la BAMGS tout en permettant la création de nouveaux points sur les surfaces des solides. Si de nouvelles surfaces apparaissent, il faut et il suffit qu'elles implémentent ces services (et donc qu'elles disposent d'une représentation paramétrique pour le cas de cette maquette) pour qu'elles puissent être maillées.

V.5 Résultats

Cette partie présente quelques maillages surfaciques de solides relativement simples modélisés par un arbre CSG et maillés par la méthode proposée.

V.5.1 Maillage pour la visualisation

Les maillages produits sont formés de polygones convexes.

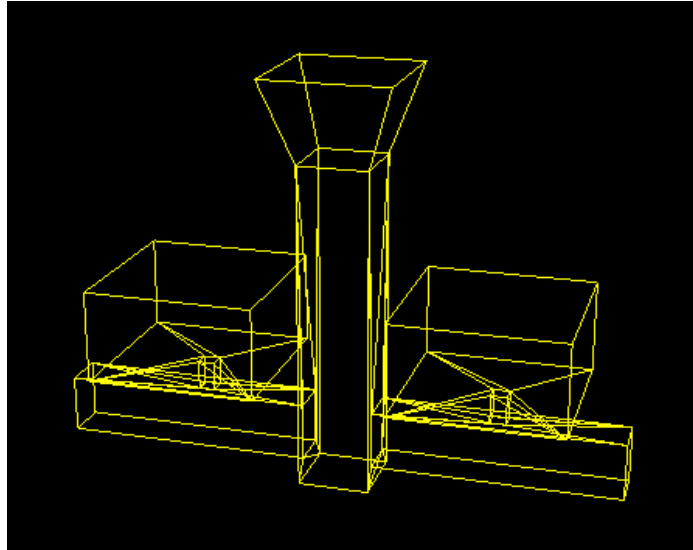


figure V.5.1 Maillage par polygones convexes d'un système de remplissage en fonderie.

Plus grande dimension de la pièce : 55

Erreur de corde : 0

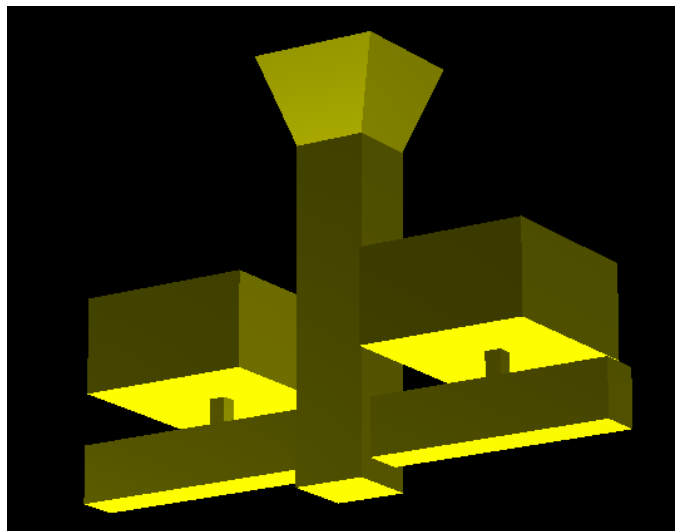


figure V.5.2 Visualisation en rendu

V.5.2 Maillage pour le prototypage rapide

Les maillages produits sont dirigés par l'erreur de corde. Il s'agit d'un renfort et d'un raccord.

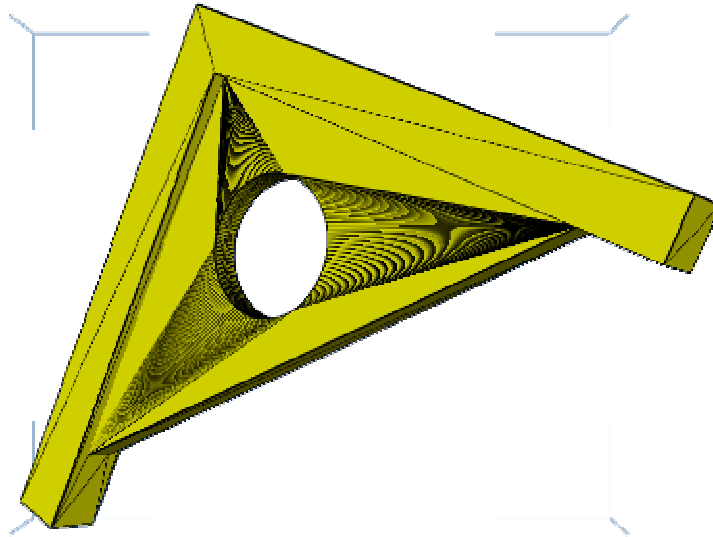


figure V.5.3. Maillage d'un renfort pour le prototypage rapide

Plus grande dimension de la pièce : 100

Erreur de corde : 0,00376477

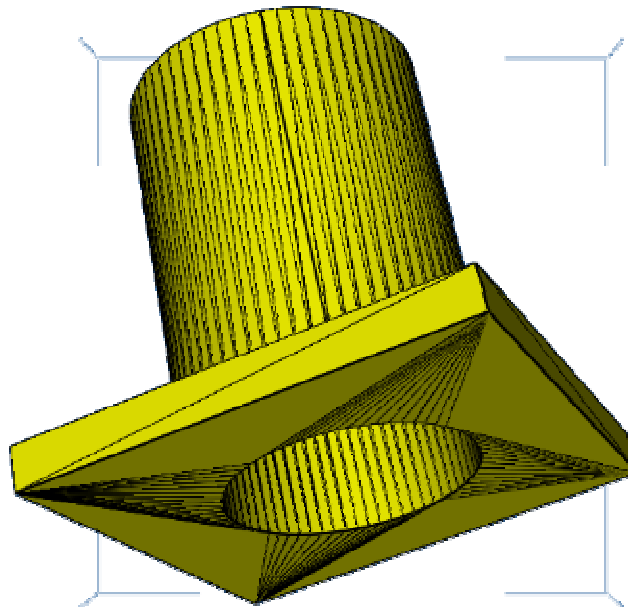


figure V.5.4 Maillage d'un raccord pour le prototypage rapide

Plus grande dimension de la pièce : 50

Erreur de corde : 0,00512008

V.5.3 Maillage pour la simulation

Les maillages produits sont dirigés par une contrainte de forme (limite du plus petit angle) et de taille (limites de la plus petite et de la plus grande arête). Pour chaque exemple sont données l'erreur de corde maximale et l'erreur d'approximation maximale des sommets. Ces grandeurs sont déterminées par la distance à la surface à laquelle ils appartiennent (ou le maximum des distances lorsqu'un sommet est porté par plusieurs surfaces).

V.5.3.a Union de deux cylindres

Sur cet exemple, nous voyons l'union de deux cylindres, en particulier les lieux où se coupent les maillages des deux surfaces cylindriques, induisant une erreur d'approximation des sommets ne dépassant pas 0,0422942.

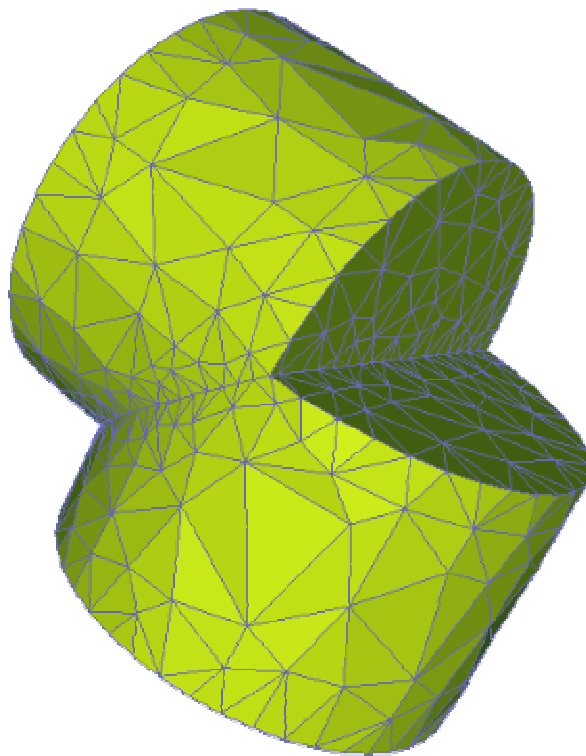


figure V.5.5. Maillage pour la simulation de l'union de deux cylindres

Plus grande dimension de la pièce : 20

Erreur de corde : 0,04831783

Erreur d'approximation des sommets : 0,0422942

V.5.3.b Disque troué

Les trous étant des lieux où se concentrent en général les contraintes, les cylindres utilisés pour les créer par une opération de différence ont été maillés plus finement que le disque plat.

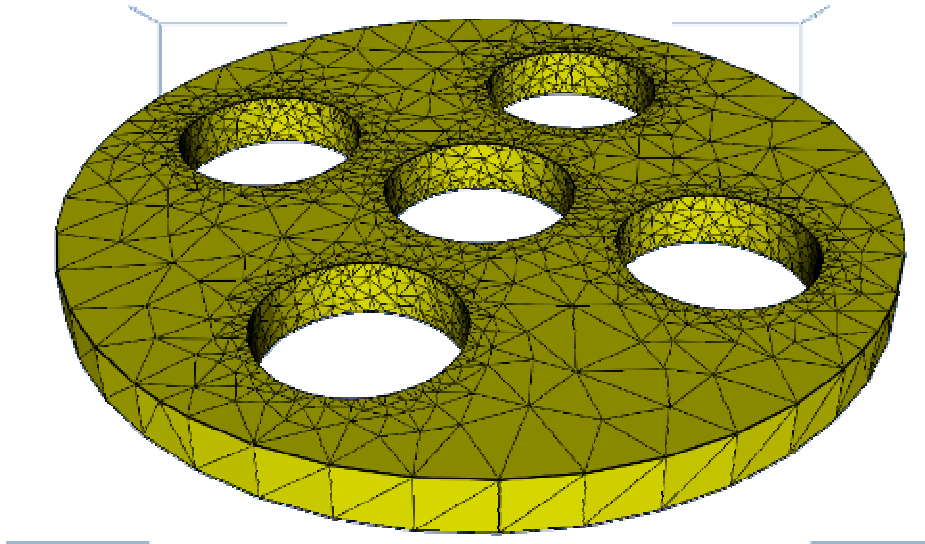


figure V.5.6. Maillage d'un guide pour la simulation

Plus grande dimension de la pièce : 150

Erreur de corde : 0,00512008

V.5.3.c Système de remplissage en fonderie

Dans le système de remplissage suivant, la pièce à fondre a été représentée par une boîte.

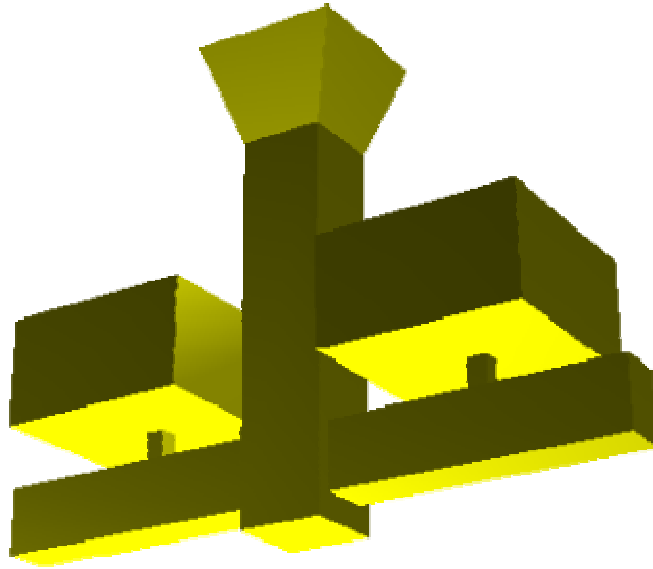


figure V.5.7 Vue générale du système de remplissage du moule

L'image suivante montre un plan rapproché de l'attaque qui constitue le raccord entre le canal et la pièce. L'attaque étant un lieu où se concentrent les contraintes lors d'une simulation de l'écoulement du métal en fusion, les propriétés de maillage qui lui ont été attribuées et conservées dans l'arbre CSG comprennent une taille d'arête minimale très petite devant celle de la pièce et celle du canal. On peut voir sur ce plan la progression de taille des arêtes.

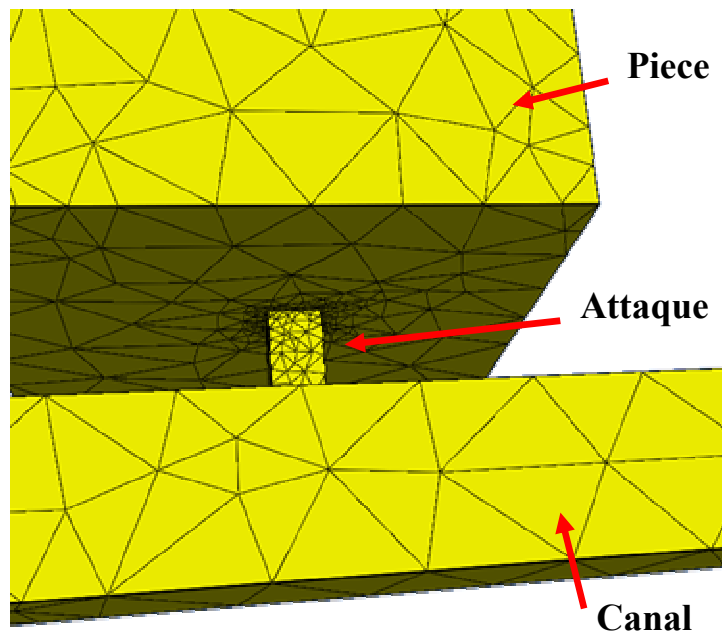


figure V.5.8 Vue rapprochée du maillage de l'attaque et de sa jonction à la pièce

Conclusion

Cette étude avait pour objectifs la conception d'un mailleur ayant deux principales caractéristiques. La première était qu'il produise des maillages de bonne qualité de manière aussi automatique que possible, pour les trois types d'applications : visualisation, simulation et prototypage rapide. La seconde caractéristique était qu'il soit extensible. Cela implique de s'adapter à différents modèles géométriques, en particulier BRep et CSG, de s'adapter à d'autres AEM, aux évolutions des besoins de chaque application, ainsi qu'aux améliorations des techniques de maillage existantes.

La construction automatique d'un maillage implique la réduction du nombre d'interventions manuelles. Elles sont nécessaires pour corriger les modèles géométriques transférés ou les maillages produits à partir de modèles géométriques incohérents, mais aussi pour entrer les spécifications du maillage (choix de la méthode de maillage, répartition des nœuds, etc.). Ces inconvénients proviennent de trois causes. Premièrement, les BRep comportent des ambiguïtés (arêtes d'intersection de surfaces gauches) et des incohérences dues à une utilisation aux limites des possibilités des systèmes de conception. Pour éviter ces inconvénients, nous proposons de construire le maillage à partir de l'historique de construction sans construire de BRep intermédiaire. Deuxièmement, avant d'être maillé, le modèle géométrique est en général transféré via un fichier au format neutre. Ces formats d'échange comportent des ambiguïtés et impliquent une réinterprétation du modèle géométrique par un autre modeleur que celui qui a servi à la conception, créant alors des erreurs. Ceci est réduit par l'interrogation directe du modeleur géométrique utilisé pour la conception, au travers d'un ensemble de services géométriques qu'utilisent les fonctions de maillage et que doit implémenter le modeleur géométrique. Troisièmement, les formats d'échange standard accusent par définition, un retard par rapport aux systèmes de conception, concernant les informations de haut niveau sémantique qu'elles prennent

en compte. Ces informations peuvent servir à mieux construire le maillage afin de fournir des maillages pré-optimisés. Pour cela, l'architecture proposée permet au système de conception un contrôle sur le processus de maillage grâce au concept de *module d'assemblage*, qui a pour rôle d'invoquer et de paramétrer les fonctions de maillage. Cela lui permet également, en particulier s'il existe un historique de construction, d'attribuer des propriétés de maillage pour les parties de l'objet qui revêtent une importance particulière pour l'AEM visée. Dans ce cas, le module d'assemblage combine les parties maillées à l'aide d'opérateurs d'assemblages (collages/découpages et opérations booléennes d'union, d'intersection et de différence). Le fait de réaliser les opérations booléennes sur les maillages possède les trois avantages suivants : 1. produire un maillage « étanche » même si les surfaces sont paramétriques, 2. pouvoir combiner des objets sous les seules conditions qu'il est possible de les mailler et d'affiner leur maillage par un contrôle de l'erreur d'approximation, et 3. permettre le maillage d'un objet en prenant en compte les spécificités de chaque partie relativement à l'AEM. L'application des opérations booléennes a pour conséquence d'altérer les propriétés des maillages aux abords des intersections. Nous nous sommes intéressés à deux propriétés particulières : la qualité en forme et la variation de taille sur des maillages triangulaires. Nous avons abordé plusieurs voies et la technique d'optimisation esquissée dans cette étude semble la piste privilégiée. Elle consiste à améliorer le maillage en déplaçant chaque point de manière à le positionner au mieux relativement aux autres. La méthode n'est pas entièrement spécifiée mais nous pensons qu'elle mérite un approfondissement et fait donc partie des perspectives.

La seconde caractéristique du mailleur est qu'il est extensible. Pour s'adapter à différents systèmes de conception et à différentes AEM, il a été conçu sous la forme d'une bibliothèque indépendante d'algorithmes. Toutefois la sémantique étant connue du seul système de conception, une part de la responsabilité doit lui être laissée. Elle relève du module d'assemblage, qui applique le processus général de maillage en adéquation avec le modèle géométrique et qui exploite la sémantique. De plus, l'accès à la géométrie se fait directement par un ensemble de services géométriques, ce qui permet à la bibliothèque d'être relativement indépendante du modèleur géométrique et donc du système de conception. Cela nécessite cependant de réécrire les algorithmes de maillage en utilisant ces fonctions. Il faut également que les modèleurs géométriques les offrent sans quoi une phase fastidieuse de correspondance entre les services nécessaires et les services fournis doit être faite manuellement à la première utilisation de la bibliothèque.

Pour être indépendantes des AEM, les fonctions de la bibliothèque sont associées chacune à un ensemble de propriétés qu'elles confèrent au maillage. Elles permettent au système appelant (système de conception ou AEM) de choisir automatiquement la fonction de maillage adéquate dès lors qu'il sait faire la correspondance entre les contraintes de l'AEM et les propriétés de maillage. Ceci est justement un des objectifs de DIJA qui est un projet transversal à l'équipe de recherche. Le fait que les propriétés concernent des caractéristiques géométriques du maillage, permet à la bibliothèque de s'abstraire de l'AEM et donc d'être extensible. Cela contribue également à la construction automatique de maillages.

Pour prendre en compte les extensions de la bibliothèque et exploiter les redondances entre les méthodes de maillage, nous avons proposé un patron de conception permettant de représenter et exploiter les algorithmes génériques. Il s'agit d'algorithmes composés d'étapes dont l'implémentation peut être interchangée, facilitant ainsi les futures évolutions du système. Ces algorithmes génériques couplés aux propriétés associées à toutes les fonctions, permettent d'assister l'utilisateur dans le choix d'une fonction de maillage et lors de l'ajout d'une nouvelle méthode dans l'architecture. Ce patron de conception n'induit pas de pertes de performances significatives du fait que les algorithmes génériques sont en général invoqués peu souvent dans une phase de maillage puisqu'ils constituent les principes des grandes familles de maillage.

Chaque aspect étudié dans cette thèse ouvre de nouvelles perspectives. Les perspectives sur les opérateurs booléens sont de prendre en compte davantage de types de maillage, notamment les maillages structurés, qui induisent des difficultés importantes aux abords des intersections. En plus d'explorer complètement la méthode d'optimisation citée précédemment, il faudrait considérer des propriétés autres que la forme et la variation de taille, telle que l'anisotropie.

Une évolution de l'architecture de la BAMGS est de rendre plus automatique l'ajout d'une nouvelle méthode de maillage en contrôlant les pré et post conditions de chaque fonction. Cela permettrait de déterminer les enchaînements de fonctions possibles au sein d'un algorithme générique. Une autre voie à explorer et de chercher à rendre générique les méthodes d'initialisation des données, nécessaires à tout appel de fonction depuis un algorithme générique, dans le but de supprimer la compilation de code lors de l'ajout d'une nouvelle méthode. Cela permettrait de minimiser l'ajout de code lors de l'insertion d'une nouvelle méthode, le reste de la configuration pouvant se faire par une interface graphique.

Dans cette étude, le module de maillage est une bibliothèque d'algorithmes. Cependant, il serait intéressant, dans le cadre du travail collaboratif, de le considérer comme un serveur sur un réseau, répondant à des requêtes en délivrant des services de maillage. Selon les performances souhaitées et les matériels du client et du serveur, le code associé à ces services serait éventuellement transféré via le réseau vers le poste client. Cela correspond tout à fait à la philosophie du projet DIJA et les aspects liés au réseau ont déjà fait l'objet d'une première étude ([DEN03]).

Conclusion

Bibliographie

- [ARM95] Armstrong C.G., Robinson D.J., McKeag R.M., Li, T.S. Bridgett T.S., Donaghy R.J., McGleenan C.A., "Medials for Meshing and More", Proceedings 4th International Meshing Roundtable, p.277-288, 1995
- [BAL99] Balendran, B. "A Direct Smoothing Method For Surface Meshes", Proceedings, 8th International Meshing Roundtable, p.189-193, 1999
- [BEC02] Béchet E., Cuillière J-C, Trochu F., "Generation of a Finite Element Mesh from Stereolithography (STL) Files", Computer Aided Design, Vol.34, No.1, p 1-17, 2002
- [BER95] Berroug M., "Contribution à la Résolution du Problème d'Intersection de deux Carreaux de Surfaces", thèse de doctorat, Université de Metz, 1995.
- [BER00] Berti G., "Generic Software Components for Scientific Computing", thèse, TU Cottbus, 2000
- [BER01] Berti G., "A Generic Toolbox for the Grid Craftsman", 17th GAMM-Seminar, Leipzig, p.1-28, 2001
- [BEZ87] Bézier P., "Courbes et Surfaces", Collection Mathématiques et CAO, Tome 4, Édition Hermès, 1987
- [BIR84] Birrell A.D., Nelson B.J., "Implementing Remote Procedure Calls", ACM Transactions on Computer Systems, Vol.2, No.1, p.39-59, 1984

- [BIS98] Biswas, R., Strawn R.C., "Tetrahedral and hexahedral mesh adaptation for CFD problems", Applied Numerical Mathematics, Vol.26, p. 135-151, 1998
- [BLA91] Blacker T., Stephenson M.B., "Paving : A new Approach to Automated Quadrilateral Mesh Generation", International Journal for Numerical Methods In Engineering, vol.32,P. 811- 847, 1991
- [BLA93] Blacker T., Myers R.J., "Seams and Wedges in Plastering: A 3D Hexahedral Mesh Generation Algorithm", Engineering with Computers, VOL.2, p.83-93, 1993
- [BLA96] Blacker T., "The Cooper Tool", Proceedings 5th International Meshing Roundtable, pp.13-30, 1996
- [BLA00] Blacker T., "Meeting the Challenge for Automated Conformal Hexahedral Meshing", Proceedings 9th International Meshing Roundtable, p.11-19, 2000
- [BLI82] Blinn, J., "A Generalization of Algebraic Surface Drawing", ACM Transactions on Graphics, P;235-256, 1982
- [BLO90] Bloomenthal J., Wyvill B., "Interactives Techniques for Implicit Modeling", Symposium on Interactive 3D Computer Graphics, Vol.24, No2, p.109-116, 1990
- [BLO00] Bloomenthal J., "Implicit Surfaces", in Encyclopedia of Computer Science and Technology", 2000
- [BOI02] Boissonnat J.-D., Devillers O., Pion S., Teillaud M., Yvinec M., "Triangulations in CGAL", Computational Geometry, Vol.22, p.5-19, 2002
- [BOR02] Bory M., el Yafi F., "La Qualité des Modèles 3D. Problèmes et Remèdes pour des Applications en Aval de la Conception", Actes de conférence MICAD 2002, p. 161-168, 2002
- [BRU02] Brun J-M., "La qualité des Modèles Géométriques", Actes de conférence MICAD 2002, p. 149-160, 2002

- [CAN97] Canann S.A., Liu Y.C., Mobley A.V., "Automated 3D Surface Meshing to Address Today's Industrial needs", *Finite Element in Analysis and Design*, VOL.25, No.1-2, p.185-198, 1997
- [CAN98] Canann, S.A., Tristano J.R., Staten M.L. "An Approach to Combined Laplacian and Optimization-Based Smoothing for Triangular, Quadrilateral, and Quad-Dominant Meshes", *Proceedings 7th International Meshing Roundtable*, p.479-494, 1998
- [CHE89] Chew P., "Guaranteed-Quality Triangular Meshes", TR89-983, Department of Computer Science, Cornell University, Ithaca, NY, 1989
- [CHE97] Chen, H., Bishop J., "Delaunay Triangulation for Curved Surfaces", *Proceedings, 6th International Meshing Roundtable*, pp. 115-127, 1997.
- [CHO98] Cho, W., Maekawa T., Patrikalakis N.M., Peraire J. "Robust Tessellation of Trimmed Rational B-Spline Surface Patches", *Proceedings of Computer Graphics International CGI 98*, IEEE, p.543-555, 1998
- [CIG98] Cignoni P., Montani C., Scopigno R., "A comparison of Mesh Simplification Algorithm", *Computer & Graphics*, Vol.22, No.1, p.37-54, 1998
- [COE96] Cohen J., Varshney A., Manocha D., Turk G., Weber H., Agarwal P., Brooks F., Wright W., "Simplification Envelopes", *Computer Graphics*, Vol.30, p.119-128, 1996
- [COM95] "The Component Object Model Specification", <http://www.microsoft.com/oledev/olecom/title.htm>
- [COR01] "The Common Object Request Broker: Architecture and Specification, Revision 2.6", 2001, <http://www.omg.org/corba/corbiop.htm>
- [CUI93] Cuilliere J-C., "Pre-optimisation de maillages automatiques tridimensionnels pour les méthodes numériques application à l'ingénierie simultanée", Thèse, Institut National Polytechnique de Lorraine, 1993

- [CUI98] Cuillère J-C., "An adaptative method for the automatic triangulation of 3D parametric surfaces", Computer-Aided Design, Vol.32, No2., pp. 139-148, 1998
- [DEL34] Delaunay, B. "Sur la sphère vide", Bul.Acad. Sci.URSS class Sci. Nat., 793-800. 1934
- [DEN01] Denis L., "Étude de différentes Architectures pour un Système de CAO distribué", Rapport de recherche, équipe CMCAO, 2001
- [DEN03] Denis, L., "Vers une architecture adaptée à un système de CAO utilisable à travers Internet", thèse de doctorat, Institut de Formation Technique Supérieur, Université de Reims Champagne-Ardenne, Charleville-Mézières 2003
- [DES92] Desbrun M., "Modélisation et Animation de Matériau Hautement Déformables en Synthèse d'Images", Thèse, Institut National Polytechnique de Grenoble, 1992
- [ECK95] Eck M., DeRose T., DuChamp T., Hoppe H., Loundsbery M., Stuezle W., "Multiresolution Analysis of Arbitrary Meshes", Computer Graphics, ACM Press, Vol.29, p.173-182, 1995
- [ERI96] Erikson C., "Polygonal Simplification: An Overview", UNC Chapel Hill Computer Science Technical Report TR96-016, 1996
- [FOL95] Foley J., van Dam A., Feiner S.K., "Introduction à l'infographie", édition Vuibert, 1995
- [FRA98] François V., "Méthodes de maillage et de remaillage automatiques appliquées à la modification de modèle dans le contexte de l'ingénierie simultanée", Thèse, Ecole Doctorale Promena, Université Henri Poincaré Nancy I, 1998
- [FRE91] Frey W.H., Field D.A., "Mesh Relaxation: A new Technique for Improving Triangulations", International Journal For Numerical Methods in Engineering, Vol 31, p.1121-1133, 1991

- [FRE94] Frey P.J., Borouchaki H., George P-L., "Delaunay Tetraedralization using advancing-front Approach", in Proceedings 5th International Meshing Roundtable, pp.31-43, 1996
- [GAM95] Gamma , E., Helm, R., Johnson, R., Vlissides, J., "Design Patterns: Elements of Reusable Object-Oriented Software", Ed. Addison Wesley, Professional, 1995
- [GAR91] Gardan Y., "La CFAO, Introduction, Techniques et Mise en Œuvre", Edition Hermès, 1991
- [GAR97] Garland M., Heckbert P.S., "Surface Simplification Using Quadric Error Metrics", Computer Graphics, Vol. 31, 1997
- [GAR99] Garland M., "Multiresolution Modeling : Survey & Future opportunities", Eurographics 99, State of the Art Reports, p.111-131, 1999
- [GAR03] Gardan Y., "CAO : modélisation géométrique", Techniques de l'ingénieur, traité informatique, H3 750, Vol. HA, 2003
- [GEO96] Frey J., Borouchaki H., George PL "Tétraédrisation de Delaunay basé sur une approche frontale". Rapport de recherche INRIA 2882, 1996
- [GEO97] George P-L., Borouchaki H, "Triangulation de Delaunay et maillage, Applications aux éléments finis", éd. Hermès, 1997
- [GOS00] Association Gosset et association Galia, "Recommandations sur la qualité des échanges de données", Passeport pour les échanges de données techniques, <http://www.goset.asso.fr/>, <http://www.galia.com>
- [GRA97] Graysmith J.L., Shaw C.T., "Automated Procedures for Boolean Operations on Finite Element Meshes", Engineering Computations, Vol.14, No.7, p. 702-717, 1997
- [HAI98] Haines, R., Follen, G. "Computational Analysis PRogramming Interface", Proceedings of the 6th International Conference on Numerical Grid Generation in Computational Field Simulations, 1998.

- [HOF89] Hoffmann C. M., "Geometric and Solid Modeling: An Introduction", Morgan Kaufmann, San Mateo, CA, 1989.
- [HOP93] Hoppe H., DeRose T., Duchamp T., McDonald J., Stuetzle W., "Mesh Optimization", Proceedings, Computer&Graphics SIGGRAPH 93, p.19-25, 1993
- [HOP96] Hoppe H., "Progressive Meshes", ACM SIGGRAPH 1996, p. 99-108, 1996
- [HOP97] Hoppe H., "View-Dependant Refinement of Progressive Meshes", ACM SIGGRAPH 1997, p.189-198, 1997
- [HUB90] Hubbard P.M., "Constructive Solid Geometry for Triangulated Polyhedra", Tech Report CS-90-07, Brown University Computer Science, 1990
- [JCF] Java Development Framework :
<http://java.sun.com/j2se/1.3/docs/guide/collections/>
- [JEN03] Jena M.K., Shunmugaraj P., Das P.C., "A Non-Stationary Subdivision Scheme for Generalizing Trigonometric Spline Surfaces to Arbitrary Meshes", Computer-Aided Design, Vol.20, p.61-77, 2003
- [JIN00] Jin, X., Li X., Peng Y., "General Constrained Deformations Based on Generalized Metaballs" , Vol. 24, No. 2 , p. 219-231, 2000
- [LAU96] Lau, T.S., Lo H., "Finite Element Mesh Generation over Analytical Curved Surfaces", Computers&Industries, Vol.59, No2, pp. 301-309, 1996
- [LAU97] Lau T.S., Lo S.H., C.K. Lee, "Generation of Quadrilateral Mesh over Analytical Curved Surfaces", Finite Element in Analysis and Design, Vol.27, p. 251-272, 1997
- [LEE95] Lee M., Stepanov A.A., "The Standard Template Library", Rapport de recherché, Hewlett-Packard Laboratories, 1995
- [LIR02] Lira W.M., Coelho L.C.G., Martha L.F., "Multiple Intersection of Finite-Element Surface Meshes", Proceedings of the 11th International Meshing Roundtable, p.355-366, 2002

- [LOH88] Löhner R., Parikh P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method", International Journal for Numerical Methods in Fluids, Vol.8, p.1135-1149, 1988
- [LOO87] Loop C.T., "Smooth Subdivision Surfaces Based on Triangles", Thèse, Université de l'Utah, 1987
- [LOR87] Lorensen W.E., Cline H.E., "Marching cubes : a high resolution 3D surface construction algorithm", Computer & Graphics, Vol.21, No.2, pp.91-101, 1987
- [LTH86] Laidlaw D.H., Trumbore W.B., Hugues J.F., "Constructive Solid Geometry for Polyhedral Objects", proc. SIGGRAPH'86, ACM Computer Graphics, Vol. 20, No. 4, 1986
- [LUE01] Luebke D.P., "A Developer's Survey of Polygonal Simplification Algorithms", Computer Graphics & Applications IEEE, p. 24-35, 2001
- [MAN82] Mantyla M., "An Inversion Algorithm for Geometric Models", Computer Graphics, Vol. 16, No. 3, 1982
- [MAT83] Mantyla M., Tamminen M., "Localized Set Operations for Solid Modelling", Computer Graphics, Vol. 17, No. 3, 1983
- [MIC03] Microsoft Corp., <http://www.microsoft.com/com/tech/dcom.asp> , Distributed Component Object Model, 2003
- [MIY00] Miyoshi K., Blacker T. "Hexahedral Mesh Generation using Multi-Axis Cooper Algorithm", Proceedings of the 9th International Meshing Roundtable, p.89-97, 2000
- [OLL01] Ollivier-Gooch C., Boivin C., "Guaranteed-Quality Simplicial Mesh Generation Cell Size and Grading Control", Engineering with Computers, Vol.17, No3, p.269-286, 2001
- [OMG01] Object Management Group, "Revised submission CAD Services V1.0 Joint Proposal to the OMG in Response to OMG Manufacturing Domain Task Force CAD Services RFP", mfg/01-07-10, 2001, <http://www.omg.org/docs/mfg/>
- [OMG01a] Object Management Group, "The Common Object Request Broker : Architecture and Specification", 2001

- [OWE97] Owen, S. J., S. Saigal "Neighborhood-Based Element Sizing Control for Finite Element Surface Meshing", Proceedings, 6th International Meshing Roundtable, p.143-154, 1997.
- [OWE98a] Owen S.J. "A Survey of Unstructured Mesh Generation Technology", Proceedings of the 7th International Meshing Roundtable, p.239-267, 1998
- [OWE98b] Owen S.J., Staten M., Canann S., Saigal S., "Advancing Front Quadrilateral Meshing Using Triangle Transformations", Proceedings of the 7th International Meshing Roundtable, p.409-428, 1998
- [PAS95] Pasko A., Adzhiev V., Sourin A., Savchenko V., "Function Representation in Geometric Modeling: Concepts, Implementation and Applications" The Visual Computer, Vol.11, No 8, p.429-446, 1995
- [PEB01] Pebay, P.P., Baker T.J, "A Comparison Of Triangle Quality Measures", Proceedings of the 10th International Meshing Roundtable, p.327-340, 2001
- [PER95] Perrin E., "Une nouvelle approche pour les opérations booléennes : Formalisation et mise en oeuvre", thèse de doctorat, Université de Metz, 1995.
- [PET01] Petersson N.A. K.K. Chand, "Detecting Translation Errors in CAD Surfaces and Preparing Geometries for Mesh Generation", Proceedings of the 10th International Meshing Roundtable, p.363-371, 2001
- [PRI95] Price M., Stops C., Butlin G., "A Medial Object Toolkit for Meshing and other Applications", Proceedings of the 4th International Meshing Roundtable, p.219-229, 1995
- [REB93] Rebay S., "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm", Journal of Computational Physics, Vol. 106, p. 125-138, 1993
- [RIV96] Rivara M-C., "New Mathematical Tools and Techniques for the Refinement and/or Improvement of Unstructured Triangulations", Proceedings of 5th International Meshing Roundtable, p. 77-86, 1996.

- [RIV01] Rivara M-C., Hitschfeld N., Simpson B., "Terminal-edges Delaunay (small-angle based) algorithm for the quality triangulation problem", *Computer-Aided Design*, Vol.33, p. 263-277, 2001
- [ROS93] Rossignac J., Borrel P., "Multi-Resolution 3D Approximation for Rendering Complex Scenes", *Modeling in Computer Graphics*, p.445--465, 1993
- [ROS97] Rossin D., Cani-Gascuel M-P., Gascuel J-D., Opalach A., Desbrun M., "Plateforme d'expérimentation pour la modélisation par surfaces implicites", *Journées Modeleurs Géométriques*, 1997
- [RUP95] Ruppert J., "A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation", *Journal of Algorithms*, Vol.18, No3, p.548-585, 1995
- [SAM01] Sampl P., "Medial Axis Construction in Three Dimensions and its Application to Mesh Generation", *Engineering with Computers*, Vol.17, p.243-248, 2001
- [SCH96a] Schneiders R., Schindler R., Weiler F. "Octree-based Generation of Hexahedral Element Meshes", *Proceedings of the 5th International Meshing Roundtable*, p.205-216, 1996
- [SCH96b] Schneiders, R., "Refining Quadrilateral and Hexahedral Element Meshes", *Proceedings 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, p. 679-686, 1996
- [SCH92] Schroeder W.J., Zarge J.A, Lorensen W.E., "Decimation of Triangle Meshes" *Computer Graphics*, Vol.26, No.2, p.65-70, 1992
- [SHB95] Sherbrook E.C., Patrikalakis N.M., Brisson E. "Computation of the Medial Axis Transform of 3-D Polyhedra", *Proceeding 3rd Symposium on Solid Modelling and Applications*, p.187-197, 1995
- [SHE91] Shephard M.S., Georges K. "Three-dimensional Mesh Generation by Finite Octree Technique", *International Journal for Numerical Methods in Engineering*, vol.32, p.709-749, 1991
- [SHE01] Shen G., Sakkalis T., Patrikalakis N. M. "Boundary Representation Model Rectification", *Graphical Models*, No.63, p. 177-195, 2001

- [SHF98] Sheffer A., Etzion M., Rappoport A., Bercovier M., "Hexahedral Mesh Generation using the Embedded Voronoi Graph", ", Proceedings of the 7th International Meshing Roundtable, p.347-364, 1998
- [SHI93] Shimada K., "Physically-Based Mesh Generation: Automated Triangulation of Surfaces and Volumes via Bubble Packing", PhD Thesis, Massachusetts Institute of Technology, 1993
- [SHI98] Shimada K., Liao J-H., "Quadrilateral Meshing with Directionality Control through the Packing of Square Cells", Proceedings of the 7th International Meshing Roundtable, pp 61-75, 1998
- [SHI00] Shimada K., Yamada A., Takayuki I., "Anisotropic Triangulation of Parametric Surfaces via Close Packing of Ellipsoids", Internatinal Journal of Computational Geometry & Applications, Vol.10, No.4, 400-424, 2000
- [SHW02] Shewchuk J.R., "Delaunay Refinement Algorithms for Triangular Mesh Generation", Computational Geometry Theory and Applications, Vol.22, no1-3, p.21-74, 2002
- [STA98] Staten M.L., Canann S.A., Owen S.J. "BMSweep: Locating Interior Nodes During Sweeping", Proceedings of the 7th International Meshing Roundtable, pp 7-18, 1998
- [STE94] Stepanov A. A., Lee M., "The Standard Template Library", Technical Report HPL-9434, Hewlett-Packard Laboratories, 31 Mai, 1994
- [STL88] "Stereolithography Interface Specification ", 3D Systems Inc., 1988
- [SUN03] Sun Microsystems Inc., <http://java.sun.com/products/jdk/rmi/>. Java Remote Method Invocation for Java,
- [TAM91] Tam T.K.H., Armstrong D.G., "2D Finite Element Mesh Generation by Medial Axis Subdivision", Adv. Eng. Software, vol.13, No. 5/6, p.313-324, 1991
- [TAU96] Tautges T., Blacker T., Mitchell S.A., "The Whisker Weaving Algorithm: A Connectivity-Based Method for Constructing All-Hexahedral Finite Element

- Meshes", International Journal for Numerical Methods in engineering, Vol.39, pp. 3327-3349, 1996
- [TAU98] Taubin G., Gueziec A., Horn W., Lazarus F., "Progressive Forest Split Compression", Computer Graphics, Vol.32, p.123—132, 1998
- [TAU00] Tautges, T.J., "The common geometry module (CGM): A generic, extensible geometry interface", Proceedings of 9th International Meshing Roundtable, p. 337-359, 2000.
- [TRI98] Tristano J.R., Owen S.J., Canann S.A., "Advancing Front Surface Mesh Generation in Parametric Space Using Riemannian Surface Definition", Proceedings of the 7th International Meshing Roundtable, 1998
- [TUR92] Turk G., "Re-Tiling polygonal Surfaces", Computer Graphics SIGGRAPH 92, Vol.26, No.2, p.55-64, 1992
- [VEL99] Veltkamp R.C., "Generic Geometric Programming in the Computational Algorithms Library", Computer Graphics, Vol; 18, No.2, 1999
- [VIS00] Viswanath N., Shimada K., Itoh T., "Quadrilateral Meshing with Anisotropy and Directionality Control via Close Packing of Rectangular Cells", Proceedings 9th International Meshing Roundtable, p. 217-225, 2000
- [WAL02] Walton K.S., Benzley S.E., Shepherd J., "Sculpting: an Improved Inside Out Scheme for All Hexahedral Meshing", Proceedings of the 11th International Meshing Roundtable, p. 153.160, 2002
- [WAN03] Wang, L., "An algorithm for the intersection of quadrilateral surfaces by tracing of neighbours", Computer Methods in applied mechanics and engineering, p.2319-2338, 2003
- [WAT81] Watson D.F., "Computing the n-dimensional Delaunay Tesselation with applications to Voronoï polytope", Computer Journal 24(2),167 – 172, 1981.
- [WHI97] White D.R., Kiney P., "Redesign of the Paving Algorithm: Enhancements through Element by Element Meshing", Proceedings, 6th International Meshing Roundtable, p.323-335, 1997

- [WIT86] Wyvill G., McPheeters, et Wyvill B. "Data Structure for Soft Objects", the Visual Computer, Vol.2, No2, p.227-243, 1986
- [WIT94] Witkin A., Heckbert P., "Using Particles to Sample and Control Implicit Surfaces", SIGGRAPH'94, Vol.28, p.269-278, 1994
- [YAM01] Yamada S., Shimada K., "HEXHOOP: Modular Templates for Converting a Hex-Dominant Mesh to an All-Hex Mesh", Proceedings, 10th International Meshing Roundtable, p.235-246, 2001
- [ZHO00] Zhou T., Shimada K., "An Angle-Based Approach to Two-Dimensional Mesh Smoothing", Proceedings, 9th International Meshing Roundtable, p.323-335, 2000

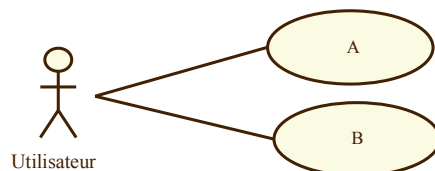
Annexe A : Rappels de notations UML (Unified Modeling Language)

Les concepts de l'approche objets sont relativement connus. Cependant les formalismes qui leur sont dédiés se ressemblant fortement, c'est pourquoi nous rappelons très brièvement ceux d'UML. En particuliers, nous voyons dans ce qui suit, les symboles des diagrammes de cas d'utilisation et des diagrammes de classes qui sont utilisés dans ce document.

A.1 Le diagramme de cas d'utilisation

Son objectif est de donner une vue externe du comportement du système. Le système est décomposé en termes d'acteurs et de cas d'utilisation. Il permet d'inventorier les fonctionnalités du système et participe à l'établissement des jeux de tests. On distingue trois types de relations soit entre acteurs et cas d'utilisation (la relation *communique*) et soit entre cas d'utilisation (relations *utilise* et *étend*).

La relation *communique* permet d'exprimer une fonction principale du système, à disposition directe d'un acteur.



Notation

La relation *utilise* exprime une relation de dépendance entre deux cas d'utilisation, cela signifie qu'une fonctionnalité a besoin d'une autre pour se réaliser. Elle permet d'exprimer une fonctionnalité comme une composition de plusieurs fonctionnalités :

- ❑ Le comportement de B est inclus dans le comportement de A
- ❑ La fonctionnalité B est nécessaire pour réaliser la fonctionnalité A



Notation

La relation *étend* permet d'exprimer une fonctionnalité comme une composition de sous-fonctions conditionnées :

- ❑ Le comportement de B peut être incorporé dans A
- ❑ Le cas d'utilisation spécialisé (c'est-à-dire B) peut être, mais pas nécessairement, déclenché par A.



Notation

La relation *généralise* permet d'exprimer l'héritage de comportement entre deux fonctions :

- ❑ Équivalent à l'héritage entre classes
- ❑ B hérite du comportement de A, peut le surcharger et le spécialiser
- ❑ B hérite des relations de communication décrites pour A

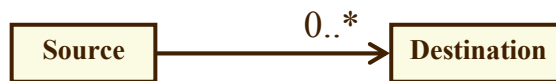


Notation

A.2 Les diagrammes de classes

Le diagramme des classes donne une vue statique des données. Il permet de représenter les classes et les relations entre elles. Il existe quatre types de relations : l'association, l'agrégation faible, l'agrégation forte (ou composition) et la généralisation. L'association et les agrégations sont annotées par des cardinalités permettant d'explicitier le lien. Les cardinalités ont exactement le même rôle qu'en base de données à la différence qu'elles se placent, pour une classe, du côté de la relation qui est opposé à la classe.

L'*association* permet de représenter une connexion physique entre deux classes. Elle peut être non directionnelle, uni- ou bidirectionnelle (flèche non fermée)



Un objet de *Source* est en association avec 0 ou N objets de *Destination*

Notation

Les agrégations (agrégation faibles et composition) sont des cas particuliers de l'association. Elles sont unidirectionnelles, transitives et antisymétrique.

L'*agrégation faible* exprime plus qu'une simple connexion physique entre classes. Elle représente une relation de type collection/membre ou composé/composant. Elles ne peuvent former qu'un graphe acyclique. Les composants sont indépendants des composés, notamment au niveau de leur durée de vie. La cardinalité à l'origine peut être supérieure à 1.



Notation

La relation de *composition* est plus forte que l'agrégation faible et force la cardinalité à l'origine à 0 ou 1. Un composé ne peut intervenir dans la composition que d'au plus un composant et la durée de vie du composant est limité par la durée de vie du composé. Les compositions forment un arbre.



Notation

La **généralisation** exprime un lien d'héritage entre deux classes. Elle est unidirectionnelle, et sans cardinalité. Une classe dérivée hérite de tous les attributs et de toutes les opérations de la classe de base. Une classe dérivée soit spécialise au moins une opération, soit ajoute au moins un attribut, soit les deux.

Notation



Annexe B : Règles d'utilisation des systèmes de CAO en vue d'un transfert de la géométrie

- La pointe d'un carreau triangulaire est considérée comme un côté de longueur nulle et peut poser de nombreux problèmes du fait que le vecteur normal est ambigu. Le moyen de contournement consiste à construire un carreau rectangulaire limité par une restriction (figure B.1).

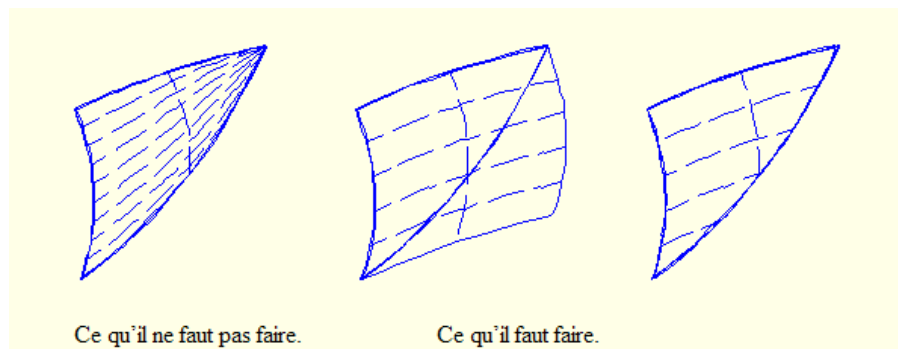
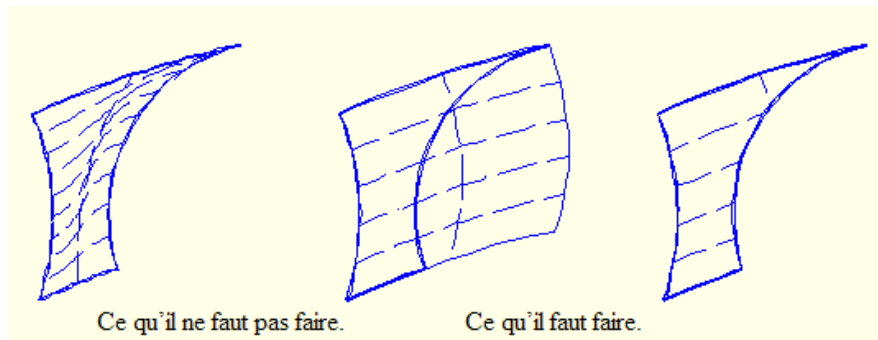
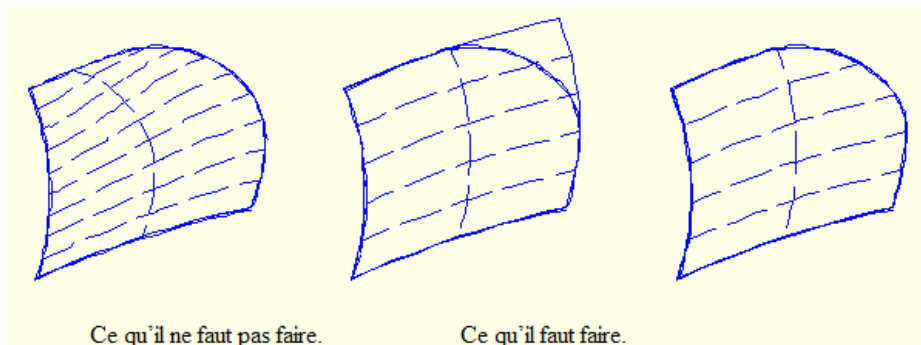


figure B.1 Traitement des surfaces triangulaires (Erreur ! Source du renvoi introuvable.)

- Les bords de surfaces ne doivent être ni alignés, ni à angle vif (leur angle doit être compris entre 2 et 178 °). Si cette recommandation sur les angles de carreaux n'est pas suivie, le vecteur normal ne peut pas être contrôlé et il est possible d'aboutir à des normales aberrantes (jusqu'à 180 degrés d'erreur). Le moyen de contournement consiste à construire un carreau rectangulaire limité par une restriction (figure B.2 et figure B.3).

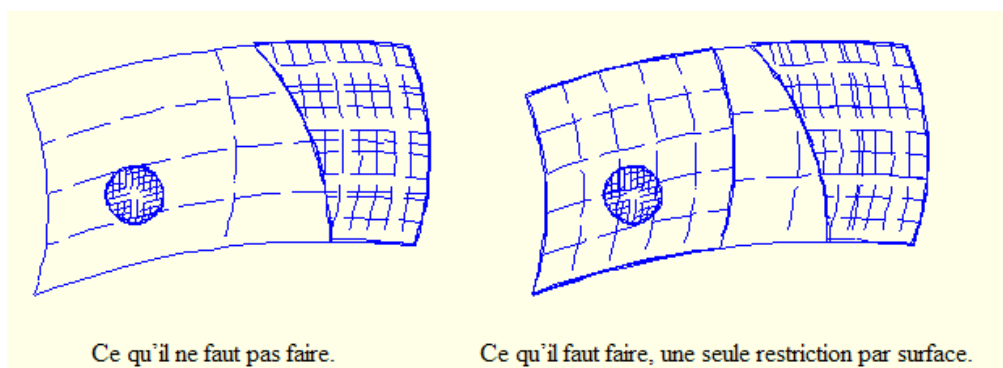


figureB.2 Surface ayant un angle vif



figureB.3 Surface avec un bord aligné

- Chaque surface ne doit comporter qu'une seule restriction. Lors des conversions, chaque restriction est traitée séparément et une surface d'appui doit être créée pour chaque restriction. Dans le cas où, sur le système source, plusieurs restrictions cohabitent sur une même surface d'appui, il faut dupliquer les surfaces d'appui sur le système cible (figure B.4). Cette situation est en effet incompatible avec certains systèmes de CAO.



figureB.4 Deux restrictions ont la même surface d'appui

- Lors de la construction de certaines surfaces, en particulier lorsqu'une arête est très petite par rapport aux dimensions des autres arêtes, il peut survenir un vrillage de la surface dans la zone de la petite arête avec inversion de la normale. (figure B.5)

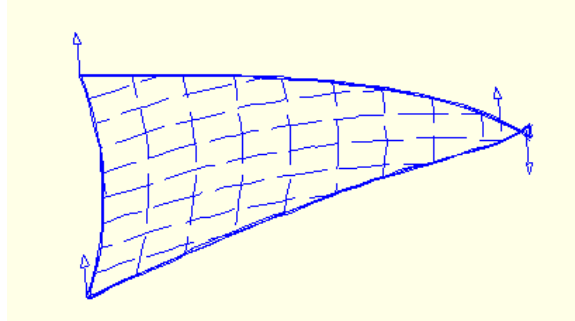


figure B.5 Surface vrillée

Annexe C : Convergence de la méthode par avancement et par recul de fronts

On dispose d'une zone à mailler délimitée par un ou plusieurs fronts sur lesquels se trouvent des écarts nodaux potentiellement hétérogènes. Ces fronts peuvent éventuellement constituer la frontière d'un maillage existant à l'extérieur de la zone à mailler. On souhaite mailler cette zone en respectant une variation de taille maximale donnée Δ_m . L'algorithme de construction de maillage est rappelé et les calculs nécessaires à sa mise en œuvre sont présentés ensuite.

C.1 Rappels de la méthode

Soit $\Gamma = \{a_i\}_{1 \leq i \leq n}$ l'ensemble des arêtes des fronts. Pour une arête a_i , on nommera S_i et S_{i+1} les extrémités de cette arête. La méthode choisit les arêtes de départ pour la construction d'un nouveau triangle parmi les arêtes construites par l'algorithme ainsi que les arêtes obtenues par calcul d'intersection de maillage (i.e. les arêtes des contours d'intersection). Soit Ω , l'ensemble de ces arêtes. Soit K , les arêtes flottantes du maillage préexistant, c'est-à-dire celles que l'on peut supprimer. Soit ε , la tolérance géométrique.

Hypothèses

H1. On suppose que les fronts sont cohérents, c'est-à-dire que pour tout couple de sommets (S_i, S_{i+1}) , $i \neq j$ issus d'un des fronts de Γ , $\|\overline{S_i S_j}\| > \varepsilon$.

H2. On suppose qu'il n'y a pas de recouvrement dans les fronts. C'est-à-dire que pour tout couple d'arêtes des fronts ne partageant pas de sommet, la plus petite distance entre les deux est supérieure à ε .

H3. L'intérieur des arêtes de Γ n'ont pas d'intersection.

Algorithme

```

Tant que  $\Gamma \neq \{ \}$  Faire
    Choix d'un segment S dans  $\Omega$ 
    Construction impossible = Test (S,  $\Gamma$ ,  $\Delta_m$ )
    Si construction impossible Alors
        Supprimer la plus proche arête flottante sur la médiatrice de S jusqu'à ce que la
        construction soit possible ou qu'il n'y ait plus d'arêtes flottantes sur cette médiatrice
    Fsi
    Si construction impossible Alors
        Si la première arête rencontrée par la médiatrice est plus longue que S Alors
            Ajouter des nœuds sur l'arête d'arrivée
        Sinon
            Ajouter des nœuds sur l'arête de départ et choisir une des arêtes
            nouvellement créées
        Fsi
    Mettre  $\Gamma$  à jour
Fsi
Appliquer la méthode frontale c'est-à-dire construire un triangle de base S :
    Construire un point Q de manière à former un triangle idéal avec S
    Scruter le voisinage  $V_Q^\rho$  de rayon  $\rho > \varepsilon$ 
    S'il existe dans  $V_Q^\rho$  des noeuds permettant de construire avec S des triangles sans
    intersection avec les fronts Alors
        Prendre le meilleur
    Sinon
        Tester l'intersection du triangle idéal avec les fronts
        Si Non Intersection Alors
            Prendre le triangle idéal tel quel
        Sinon
            Chercher un autre triangle (selon les cas d'intersection, soit on
            ajoute un sommet à l'arête coupée, soit on diminue
            successivement le triangle idéal jusqu'à ce qu'il n'y ait plus
            d'intersection)
        Fsi
    Fsi
    Mettre à jour les fronts (Retirer S de  $\Gamma$  mais y ajouter les deux autres arêtes du
    triangle en tant qu'arêtes fixes)
Ftant
    
```

Algorithme 1. Triangulation par Avancement Recul Front

Ces différentes étapes font appels à un certains nombre de calculs que nous présentons maintenant. Pour clarifier le discours, la méthode est exposée dans le cas d'un agrandissement, c'est-à-dire quand l'arête en vis-à-vis de $[S_i, S_{i+1}]$ est plus longue que $[S_i, S_{i+1}]$. Le cas d'une réduction est symétrique.

On cherche à construire un triangle en plaçant le nouveau point sur la médiatrice du segment $[S_i, S_{i+1}]$ (Figure B.1). Pour cela, on cherche à savoir combien de triangles agrandis

successivement il serait nécessaire de placer sur cette médiatrice pour atteindre la distance entre nœuds sur le front en vis-à-vis.

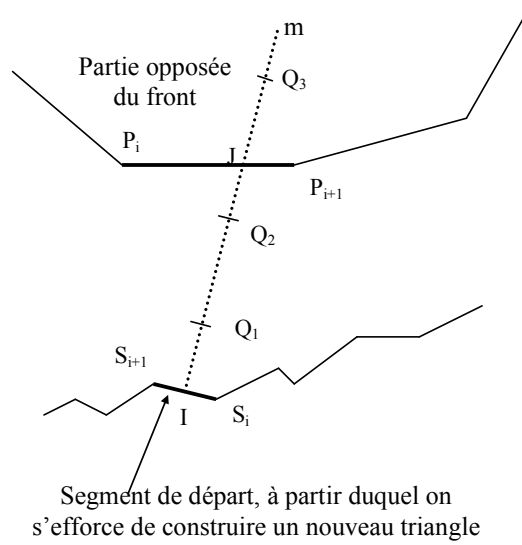


Figure B.1 Evaluation de l'espace disponible

La hauteur u d'un triangle agrandi d'un facteur Δ_{\max} lorsqu'il est accolé à un triangle équilatéral vaut :

$$u = \Delta_{\max} \frac{\sqrt{3}}{2} \left\| \overline{S_i S_{i+1}} \right\|$$

La suite u_n donne la hauteur d'un triangle après n agrandissement successifs :

$$u_n = \Delta_{\max}^n \cdot \frac{\sqrt{3}}{2} \left\| \overline{S_i S_{i+1}} \right\|$$

avec $[S_i S_{i+1}]$ le segment dont la médiatrice donne la direction d'agrandissement, et Δ_{\max} la variation de taille maximale.

L'écart entre nœuds d'un triangle après n agrandissement est :

$$ND(Q_n) = \frac{2\Delta_{\max}}{\sqrt{3}} u_{n-1} = \Delta_{\max}^n \left\| \overline{S_i S_{i+1}} \right\|$$

On cherche un entier p , tel que l'écart entre nœuds après p agrandissement soit supérieur à l'écart entre nœuds sur le front en vis-à-vis, c'est-à-dire tel que :

$$\begin{aligned}
 ND(Q_p) &\geq \|\overline{P_i P_{i+1}}\| \\
 \Delta_{\max}^p \|\overline{S_i S_{i+1}}\| &\geq \|\overline{P_i P_{i+1}}\| \\
 p &\geq \log_{\Delta_{\max}} \left(\frac{\|\overline{P_i P_{i+1}}\|}{\|\overline{S_i S_{i+1}}\|} \right)
 \end{aligned}$$

On choisit donc $p = \left\lceil \log_{\Delta_{\max}} \left(\frac{\|\overline{P_i P_{i+1}}\|}{\|\overline{S_i S_{i+1}}\|} \right) \right\rceil$ (l'entier supérieur le plus proche)

Donc au bout de p agrandissements, l'écart entre nœuds atteint l'écart entre nœuds visé (celui du front en vis-à-vis). Les p agrandissements appliqués le long de la médiatrice donnent un point Q_p . Si Q_p est placé au-delà du front, alors on estime que la distance n'est pas suffisante pour respecter la variation de taille et on cherche à modifier localement le maillage (en supprimant des arêtes flottantes ou en insérant des nœuds). Sinon on va chercher à construire un point en respectant la variation de taille maximale. Pour cela, il n'est pas forcément nécessaire d'appliquer l'agrandissement maximal ; c'est par exemple le cas si l'écart entre nœuds au départ est proche de l'écart entre nœuds sur le front en vis-à-vis. Il faut donc déterminer *la variation de taille suffisante* Δ . Pour cela, on calcule d'abord r , le nombre de pas pour atteindre non pas la distance entre nœuds, mais le front en vis-à-vis. Si les écarts entre nœuds sont identiques et que la distance qui les sépare est faible (mais ne nécessite pas de modification du maillage), alors r vaut 1. Sinon r vaut p additionné du nombre de fois qu'on doit ajouter à $\|IQ_p\|$, la hauteur du plus grand triangle (c'est-à-dire u_p) pour atteindre le segment $[IJ]$, soit le rapport de u_p sur la distance entre Q_p et le segment $[IJ]$.

$$r = \max \left(1, \left\lceil p + \frac{\|\overline{IJ}\| - \|IQ_p\|}{u_p} \right\rceil \right)$$

Puis, la variation de taille suffisante Δ est calculée en utilisant r dans l'équation :

$$ND(Q_r) = \|\overline{P_i P_{i+1}}\|$$

Ce qui donne

$$\Delta^r \frac{\|S_i S_{i+1}\|}{\|P_i P_{i+1}\|} = \frac{\|P_i P_{i+1}\|}{\|S_i S_{i+1}\|}$$

$$\Delta = \sqrt[r]{\frac{\|P_i P_{i+1}\|}{\|S_i S_{i+1}\|}}$$

C.2 Démonstration

On montre que 1. l'algorithme termine et 2. que la zone à remailler est vide.

C.2.1 Preuve de la terminaison

Supposons que l'aire à remailler soit finie et qu'à chaque itération soit construit un nouveau triangle de taille non négligeable.

Alors le nombre de triangles nécessaires pour recouvrir l'aire à mailler est fini. Ce nombre est majoré par le rapport entre l'aire totale de la face (y compris la partie déjà maillée) et l'aire d'un triangle de taille négligeable : $\frac{A_T}{\frac{\sqrt{3}}{4} \cdot \epsilon^2}$ avec A_T , l'aire totale de la face et ϵ , la tolérance géométrique.

Or si un triangle non négligeable est construit à chaque itération, alors le nombre d'itérations est fini également. Donc l'algorithme termine.

C.2.2 Preuve que l'aire à remailler est vide à la fin de l'exécution

L'algorithme termine ssi les fronts sont vides. Et les fronts sont vides ssi l'aire à remailler est vide (ou infinie ce qui ne peut pas être le cas ici). Or l'algorithme termine donc l'aire restant à mailler est vide.

Il reste à montrer que l'aire à mailler est finie et qu'un triangle de taille non négligeable est construit à chaque itération de l'algorithme.

C.2.3 Preuve que l'aire à mailler est finie

Le maillage de la zone est dû à une opération booléenne qui a été réalisée entre deux objets. Les opérandes sont soit des volumes fermés, soit des demi-espaces infinis auquel cas on suppose qu'un calcul préalable a permis de déterminer de quelle partie finie du demi-espace est constitué le solide (en déterminant les intersections de ce demi-espace avec les autres composants du solide, par exemple en appliquant une dichotomie, puis un suivi). Donc la surface sur laquelle est appliqué le remaillage est finie.

C.2.4 Preuve qu'un triangle non négligeable est construit à chaque étape

La seconde partie de l'algorithme consiste à construire un triangle avec la méthode d'avancement de front ; elle crée un nouveau triangle à chaque itération (même si la variation de taille n'est plus respectée dans certains cas que nous voyons dans la suite, et même si des triangles du maillage préexistant ont été supprimés). Il faut donc montrer que le triangle construit n'est pas de taille négligeable. On considère qu'un triangle est de taille négligeable si un de ses côtés est de dimension inférieure ou égale à une précision ε donnée, ou si le triangle est de taille négligeable, c'est-à-dire qu'une de ses hauteurs a une longueur inférieure ou égale à ε .

La démonstration procède ainsi :

1. Montrons d'abord qu'aucun triangle *idéal* n'est de taille négligeable si les H_i sont respectées. Ceci sert à démontrer ensuite que :
2. Aucun triangle *effectivement construit* n'est de taille négligeable si les H_i sont respectées.
3. Montrons que les H_i sont respectées à chaque itération.

C.2.4.a Taille des triangles idéaux

On suppose dans ce paragraphe et dans le suivant, qu'il est possible de mailler en respectant l'écart nodal et donc on est dans la situation où un triangle idéal va être construit. Ses dimensions sont calculées en se basant sur la distance entre une arête de départ et une arête en vis-à-vis (qui existe toujours car les zones à remailler sont fermées).

Par construction, la méthode assure que le triangle idéal qu'on cherche à construire est isocèle et possède une base de longueur supérieure à ε d'après H1. Montrons que

- a. Quelle que soit la situation, le triangle idéal ne possède pas de côté inférieur à ε dans le cas où on suppose que l'agrandissement (- resp. la réduction) nécessaire ne dépasse par l'agrandissement (- resp. la réduction) maximale.
- b. Le triangle idéal n'est pas de taille négligeable.

Preuve de a. : La situation est la suivante :

$[S_i S_{i+1}]$ est un segment de départ et $[P_i P_{i+1}]$ est le segment coupé par la médiatrice de $[S_i S_{i+1}]$. On construit un triangle à partir de $[S_i S_{i+1}]$. On montre que la différence des écarts nodaux

entre le segment de départ et celui d'arrivée n'engendre pas à un triangle idéal ayant un côté de longueur inférieure à ε .

Comme H_1 est vérifiée, on peut poser

$$\|\overline{S_i S_{i+1}}\| = \varepsilon + \eta \text{ avec } \eta > 0, \text{ et } \|\overline{P_i P_{i+1}}\| = \varepsilon + \sigma \text{ avec } \sigma > 0,$$

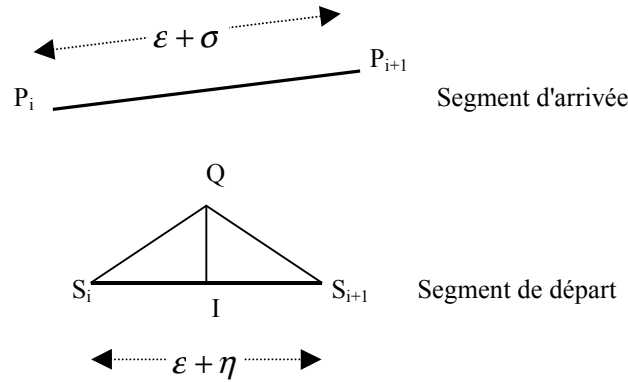


Figure B.2 Définition de ε , η et σ

Si $\varepsilon + \eta \leq \varepsilon + \sigma$ alors le triangle va être agrandi par rapport au triangle équilatéral, donc ses côtés aussi (sauf la base) et ils ne pourront donc être de longueur inférieure à ε . Dans le cas contraire ($\sigma < \eta$), le triangle va être réduit. Montrons alors que $QS_i > \varepsilon$.

La réduction nécessaire et suffisante pour passer d'un écart nodal à l'autre est (voir paragraphe C.1):

$$\Delta = \sqrt[r]{\frac{\varepsilon + \sigma}{\varepsilon + \eta}} \text{ avec } r \geq 1$$

Calculons $\|\overline{QI}\|$ afin de déterminer $\|\overline{QS_i}\|$:

$$\|\overline{QI}\| = \frac{\sqrt{3}}{2} \Delta \|\overline{S_i S_{i+1}}\|$$

$$\|\overline{QI}\| = \frac{\sqrt{3}}{2} \sqrt[r]{\frac{\varepsilon + \eta}{\varepsilon + \sigma}} (\varepsilon + \eta)$$

Or

$$\|\overline{QS}_i\|^2 = \|\overline{QI}\|^2 + \left(\frac{\|\overline{S}_i \overline{S}_{i+1}\|}{2} \right)^2$$

Donc

$$\|\overline{QS}_i\|^2 = \left[\frac{\sqrt{3}}{2} (\varepsilon + \eta) \sqrt{\frac{\varepsilon + \sigma}{\varepsilon + \eta}} \right]^2 + \left(\frac{\varepsilon + \eta}{2} \right)^2$$

$$\|\overline{QS}_i\|^2 \geq \left[\frac{\sqrt{3}}{2} (\varepsilon + \eta) \sqrt{\frac{\varepsilon + \sigma}{\varepsilon + \eta}} \right]^2 + \left(\frac{\varepsilon + \eta}{2} \right)^2$$

car $\frac{\varepsilon + \sigma}{\varepsilon + \eta} < 1$ et la fonction $\sqrt[k]{x}$ avec k constante dans $]0,1]$ est décroissante pour $x > 0$

$$\|\overline{QS}_i\|^2 \geq \frac{3}{4} (\varepsilon + \eta)^2 \cdot \frac{\varepsilon + \sigma}{\varepsilon + \eta} + \left(\frac{\varepsilon + \eta}{2} \right)^2$$

$$\|\overline{QS}_i\|^2 \geq \frac{3}{4} (\varepsilon + \eta)(\varepsilon + \sigma) + \left(\frac{\varepsilon + \eta}{2} \right)^2 \text{ avec } \frac{3}{4} (\varepsilon + \eta)(\varepsilon + \sigma) > \frac{3}{4} \varepsilon^2 \text{ et } \left(\frac{\varepsilon + \eta}{2} \right)^2 > \frac{1}{4} \varepsilon^2$$

$$\text{Donc } \|\overline{QS}_i\|^2 > \varepsilon^2$$

$$\text{Donc } \|\overline{QS}_i\| > \varepsilon, \forall \eta, \forall \sigma : \sigma < \eta . \text{ cqfd (a.1)}$$

Preuve de b. : Montrons que le triangle idéal n'est pas de taille négligeable.

On dit d'un triangle qu'il est de taille négligeable si l'une de ses hauteurs a une longueur inférieure ou égale à $\frac{\sqrt{3}}{2} \varepsilon$, ce qui correspond à la hauteur d'un triangle équilatéral de côté ε .

Dans le cas de la construction d'un triangle idéal, la hauteur la plus petite est celle qui est contrôlée par une éventuelle réduction, c'est-à-dire QI (Figure B.2).

$$\begin{aligned} \|\overline{QI}\| &= \frac{\sqrt{3}}{2} \Delta \|\overline{S_i S_{i+1}}\| \\ \|\overline{QI}\| &= \frac{\sqrt{3}}{2} \sqrt{\frac{\varepsilon + \sigma}{\varepsilon + \eta}} (\varepsilon + \eta) \\ \|\overline{QI}\| &\geq \frac{\sqrt{3}}{2} \sqrt{\frac{\varepsilon + \sigma}{\varepsilon + \eta}} (\varepsilon + \eta) \\ \|\overline{QI}\| &\geq \frac{\sqrt{3}}{2} \sqrt{\varepsilon + \sigma} \sqrt{\varepsilon + \eta} \\ \|\overline{QI}\| &\geq \frac{\sqrt{3}}{2} \sqrt{\varepsilon} \sqrt{\varepsilon} \\ \|\overline{QI}\| &> \frac{\sqrt{3}}{2} \varepsilon \end{aligned}$$

Donc aucun triangle idéal ne peut être plat.

C.2.4.b Taille des triangles construits

Il y a plusieurs cas de construction d'un triangle : soit il est construit en prenant le triangle idéal et dans ce cas il n'est pas de taille négligeable, soit il est construit en prenant un voisin du sommet créé pour former un triangle idéal (C.2.4.b.1), soit un nouveau sommet est créé en fonction de la configuration locale (C.2.4.b.2).

C.2.4.b.1 *Le triangle est construit par choix d'un sommet voisin*

On montre dans ce cas, que le candidat choisi n'induit pas de triangle de taille négligeable.

a. Montrons que les côtés du triangle construit sont inférieurs à ε .

Raisonnons par l'absurde. Supposons qu'il existe un sommet S_p dans le voisinage de Q , tel que $\|\overline{S_p S_i}\| \leq \varepsilon$ (la preuve pour $\|\overline{S_p S_{i+1}}\|$ est identique).

Alors il y a 4 situations possibles selon l'origine de la construction des 2 sommets.

Cas 1 : S_p et S_i appartiennent à des fronts de la configuration initiale lors du lancement de l'algorithme, ce qui est contraire à l'hypothèse H1.

Cas 2 : S_i appartient à un des fronts de la configuration initiale et pas S_p . Alors S_p a été construit pour former un triangle idéal. Cette situation est impossible car au moment où a été construit S_p , le voisinage de S_p a été observé donc on aurait découvert la présence de S_i .

Cas 3 : S_p appartient à un des fronts de la configuration initiale et pas S_i . Cette situation est symétrique à la 2^{ème} situation

Cas 4 : Ni S_i , ni S_p n'appartiennent à des fronts de la configuration initiale. Donc ils ont été construits pour former chacun un triangle idéal. Le second point construit aurait donc dû détecter la présence de l'autre lorsqu'on a examiné son voisinage. Donc c'est impossible.

Par conséquent, s'il existe un point dans le voisinage de Q , il ne peut pas être à une distance inférieure à ε de S_i ou de S_{i+1} . **cqfd (a)**

b. Montrons que le triangle $S_i S_{i+1} Q$ n'est pas plat.

Pour que le triangle $S_i S_{i+1} Q$ soit plat, il faut que Q soit à une distance ε de la droite $(S_i S_{i+1})$. Il y a deux possibilités: soit Q appartient (à ε près) au segment $[S_i S_{i+1}]$, soit il est à l'extérieur.

Cas 1 : Q est à l'extérieur du segment. Comme le triangle idéal est isocèle, cela signifie que Q est un point du voisinage du triangle idéal basé sur $[S_i S_{i+1}]$. Ceci implique que la taille du voisinage est supérieure à $\|\overline{QS_i}\|$, ce qui est contradictoire avec la méthode.

Cas 2 : Q est à l'intérieur de $[S_i S_{i+1}]$. Donc soit Q appartient au front initial, soit Q a été construit pour former un triangle idéal (Figure B.3).

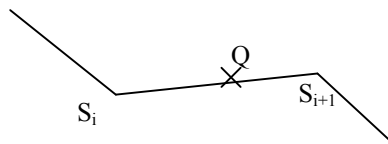


Figure B.3 Q est situé sur le segment (à une tolérance près)

Cas 2.1 : Q appartient au front initial (à ε près). Alors si $[S_i S_{i+1}]$ appartient au front de départ, cela signifie que le front de départ n'était pas conforme, ce qui est contraire à l'hypothèse H2. Sinon la situation aurait été détectée auparavant lors des tests d'intersection lors de la construction du triangle dont $[S_i S_{i+1}]$ fait partie. **Impossible**

Cas 2.2 : Q a été construit pour former un triangle idéal (pas forcément basé sur $[S_i S_{i+1}]$). Dans ce cas, on aurait dû détecter le fait que Q appartienne à une arête du front lors du calcul d'intersection du nouveau triangle avec le front. **Impossible**.

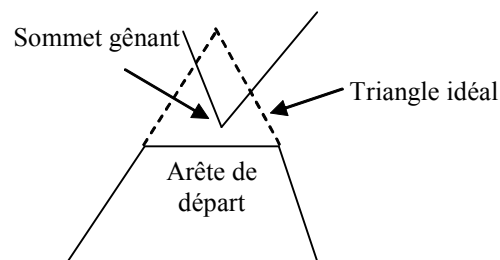
Le cas 2 n'est donc pas possible. **cqfd (b)**

Par conséquent la construction d'un triangle à partir d'un sommet voisin n'induit pas de triangle de taille négligeable.

C.2.4.b.2 Création d'un nouveau point

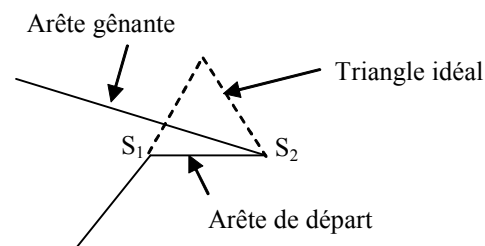
Les autres triangles candidats dépendent du type d'intersection du triangle idéal avec les fronts. Il se peut que le triangle idéal soit coupé plusieurs fois. On observe alors l'entité partiellement ou entièrement à l'intérieur du triangle idéal, qui est la plus proche de l'arête de départ. On distingue trois types d'intersections.

- 1^{er} type d'intersection : un sommet des fronts est dans le triangle idéal.



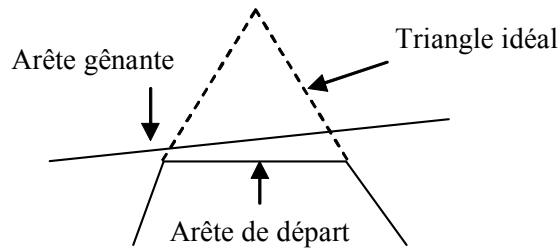
On cherche dans ce cas à construire un triangle avec le sommet gênant. Ce nouveau triangle ne peut pas couper le front car sinon cela voudrait dire qu'il y a une autre entité que ce sommet qui est encore plus proche de l'arête de départ, ce qui est impossible par hypothèse. Par les hypothèses H1 et H2, ce sommet se situe à une distance supérieure à ε de l'arête de départ. Le triangle construit n'est donc pas de taille négligeable.

- 2^{ème} type d'intersection : une arête voisine de l'arête de départ coupe le triangle idéal



Dans cette situation, on insère le projeté de S_1 sur l'arête gênante perpendiculairement à $[S_1S_2]$. Si ce nouveau sommet ne va pas à l'encontre des hypothèses H_i , un nouveau triangle est construit en se basant sur ce projeté. Sinon, on utilise l'extrémité de l'arête gênante pour construire le nouveau triangle. Les trois sommets, par hypothèse, respectent les H_i , donc le triangle construit n'est pas de taille négligeable.

- 3^{ème} type d'intersection : une arête non voisine coupe les deux côtés du triangle idéal.



Un sommet est ajouté au milieu des intersections du triangle idéal avec l'arête gênante. Si les H_i sont respectés, un nouveau triangle est construit, sinon on utilise une des extrémités de l'arête gênante.

Conclusion : tout triangle construit par la méthode réduit la taille de la zone à mailler de manière non négligeable.

C.2.4.c Respect de H_1 et H_2

L'hypothèse H_1 (tout sommet est à une distance inférieure à ε de tout autre) est assurée d'une part par le fait que l'insertion d'un nouveau sommet dans le maillage est précédée de l'observation du voisinage de ce sommet. Et d'autre part, l'ajout de nœuds ne crée pas d'arête de dimension inférieure à ε .

L'hypothèse H_2 (il n'y a ni intersection ni recouvrement d'arête), est maintenue par les tests d'intersection des arêtes créées avec les arêtes du maillage. Par conséquent si l'une s'approche à une distance inférieure à ε d'une autre, le test d'intersection le mettra en évidence.

C.2.5 Conclusion

Cette partie a démontré qu'à chaque itération, un nouveau triangle de taille non négligeable est construit. Par conséquent cela termine la preuve de la terminaison (C.2.1) et que l'aire à mailler est vide à la fin de l'exécution de l'algorithme (C.2.2). cqfd.

Annexe D : Critère pour la qualité en forme

Notation

On note par $\angle OAB$, l'angle orienté fait entre les vecteurs \overline{OA} et \overline{OB} .

D.1 Rappel

On dispose d'un triangle ABD et on souhaite lui adosser, le long de $[AB]$, un triangle ACB tel que l'angle en C ait une dimension supérieure ou égale à θ .

Soit DP le demi-plan limité par (AB) et ne contenant pas D . Pour que le triangle construit ne coupe pas le triangle de référence, C doit se trouver dans ce demi-plan.

Soit Me la partie de la médiatrice de $[AB]$ située dans le demi-plan DP . On appelle C_M le point de Me tel que $\angle AC_M B = \theta$. C_M n'étant pas aligné avec A et B , ces trois points définissent un unique disque. Nous appelons son centre O .

Soit Z l'intersection du disque avec DP . Pour tout point de DP , montrons que l'angle $\angle CAB$ est supérieur ou égal à θ si et seulement si C est dans Z .

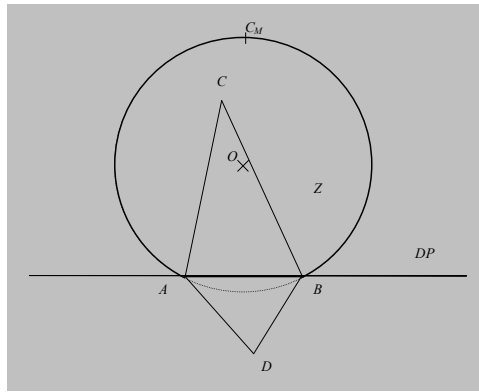


Figure C.2.5C.1 Géométrie relative à la contrainte de forme

D.2 Démonstration :

Dans un cercle, on a la propriété suivante : « Soit un cercle de centre O du plan. Si les points A , B , et C_M , deux à deux distincts, appartiennent au cercle, alors $\angle OAB = \angle C_MAB$ » (Figure C.2). (« Mathématiques, Probabilités – Algèbre – Géométrie », Éditions Bréal, Collection Cube, 1994).

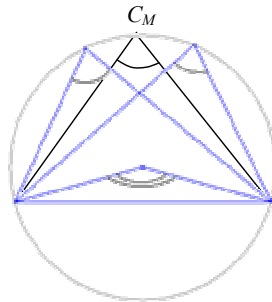


Figure C.2 Propriété des angles à la circonférence

Donc l'angle $\angle OAB$ est double de $\angle C_MAB$; il vaut $2x\theta$. Symétriquement, les angles formés par un quelconque point de l'arc AC_MB avec A et B mesurent tous θ .

Soit C_I un point de Z et (MC_I) la médiane en C_I du triangle ABC_I . C_O est l'intersection de la médiane avec l'arc AC_MB .

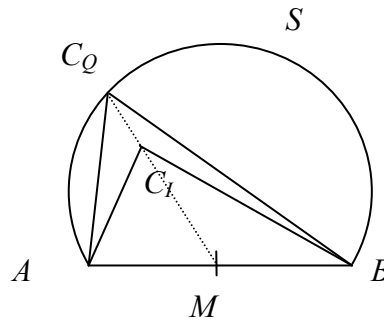


Figure C.3 Relation entre les angles en C_I et en C_Q

alors l'angle $\angle C_QAB$ vaut θ d'après ce qui précède.

L'angle $\angle C_IAM$ est supérieur à l'angle $\angle C_QAM$ car C_Q , C_I et M sont situées sur la même droite et C_I se trouve entre C_M et M . De même l'angle $\angle C_I MB$ est supérieur à l'angle $\angle C_Q MB$.

Par conséquent $\angle C_IAM + \angle C_I MB > \angle C_QAM + \angle C_Q MB = \theta$.

Donc $\angle C_IAB > \theta$. On a donc bien :

$$C_I \in Z \Rightarrow \angle C_IAB \geq \theta$$

On prouve de manière similaire que si C_I est extérieur à Z , alors $\angle AC_I B < \theta$ (en effet, C_I n'est plus entre M et C_M et, de ce fait, les inégalités sont inversées). Par contraposée, on déduit que :

$$\angle AC_I B \geq \theta \Rightarrow C_I \in Z$$

L'équivalence est ainsi démontrée.

Annexe E : Services géométriques

Les services suivant sont les services géométriques de bas niveau nécessaires aux fonctions de maillage. Tous les services ne sont pas nécessaires à toutes les fonctions de maillage.

- ❑ Test d'appartenance d'un point au domaine (2D plan, 2D surfacique, 3D).
- ❑ Test d'intersection droite/courbe, droite/surface.
- ❑ Les vecteurs tangent et normal d'un point d'une courbe ou d'une surface.
- ❑ Le rayon de courbure d'un point d'une courbe ou d'une surface.
- ❑ La distance d'un point à une courbe/surface.
- ❑ La distance maximale d'un segment à la portion qu'elle approche sur une courbe ou sur une surface (c'est-à-dire l'erreur d'approximation d'un segment).
- ❑ La distance maximale d'une maille à la portion qu'elle approche sur une surface (c'est-à-dire l'erreur d'approximation d'une maille).
- ❑ La distance minimale entre deux points : dans un plan, sur une courbe, sur une surface.
- ❑ Mesure de l'angle entre deux segments dans un espace plan muni d'une métrique.
- ❑ La création d'un point au barycentre d'un ensemble de points avec une distance calculée en restant sur une courbe ou sur une surface.
- ❑ La construction du cercle (-resp. la sphère) circonscrit à un triangle (-resp. un tétraèdre).
- ❑ La construction du cercle (-resp. la sphère) inscrit dans un triangle (-resp. un tétraèdre).
- ❑ La projection d'un point sur une courbe/surface.
- ❑ La construction d'un englobant de la courbe/surface.

- ❑ Le test d'appartenance d'un point à un voisinage (l'ensemble des points situés à une distance à un point central donné, d'une valeur inférieure à un seuil donné).
- ❑ Construction d'un point situé à une distance donnée d'un autre dans une direction donnée en restant sur la courbe ou sur la surface¹.
- ❑ Construction d'un segment perpendiculaire à un autre avec toutes les extrémités sur la surface¹.
- ❑ Construction d'un segment faisant un angle donné avec un autre en restant sur la surface¹

¹ Services généralement réalisés lorsque les surfaces sont des surfaces paramétriques régulières