Numéro d'ordre : 58-99
Année 1999

THÈSE

présentée devant

L'UNIVERSITÉ CLAUDE BERNARD - LYON 1

pour l'obtention

du DIPLÔME DE DOCTORAT
(arrêté du 30.03.92)
spécialité : INFORMATIQUE

par

Fabrice JAILLET

Contribution à la reconstruction et à l'animation d'objets déformables définis à partir de données structurées en sections

Soutenue le 05 février 1999, devant le jury composé de :

Président :	Denis VANDORPE
Rapporteurs :	Marc NEVEU
	Claude PUECH
Examinateurs :	Serge MIGUET
	Chris MOORE
	Behzad SHARIAT

Remerciements

Ce travail a été réalisé au sein du Laboratoire d'Informatique Graphique, Images et Modélisation de l'Université Claude Bernard LYON I.

Je tiens à exprimer toute ma reconnaissance à M. Denis VANDORPE, professeur à l'université LYON I et directeur du LIGIM pour m'avoir accueilli dans ce laboratoire. Je lui exprime ici ma profonde gratitude et ma considération pour ses critiques positives qui m'ont permis de mener à bien ce travail.

J'adresse mes plus vifs remerciements à M. Claude PUECH, professeur à l'université Grenoble I, ainsi qu'à M. Marc NEVEU, professeur à l'université de Bourgogne, pour m'avoir fait l'honneur d'être rapporteurs de mon mémoire de thèse, et surtout d'avoir accepté de le faire dans un très court délai. Les remarques et les conseils qu'ils m'ont adressés ont été vraiment très constructifs pour la suite de ma recherche.

Toute ma reconnaissance à M. Serge MIGUET, professeur à l'université LYON II, pour l'intérêt qu'il a porté a mon travail et pour avoir accepté de participer à mon jury. Je remercie également M. Chris J. MOORE, directeur de recherche au Christie Hospital de Manchester, qui a accepté d'être membre de mon jury, pour m'avoir accueilli chaleureusement lors de mes séjours à Manchester ainsi que pour toutes les discussions que nous avons eues sur mon travail.

Je tiens à exprimer toute ma gratitude à M. Behzad SHARIAT, maître de conférences HDR à l'université LYON I, qui a dirigé mes recherches pendant ma thèse. Il a su m'apporter tout le soutien dont j'avais besoin pour réaliser ce travail. Sa gentillesse et sa disponibilité n'ont jamais été démenties. Je le remercie aussi pour tout l'intérêt qu'il a porté à mes travaux mais aussi pour m'avoir toujours laissé libre de mes choix dans la conduite de cette thèse.

Je tiens à exprimer toute ma sympathie à tous les membres du laboratoire et du bâtiment 710. Notamment à Serge B. *Ponpon*, qui est le suivant sur la liste et qui devra faire des progrès à la pétanque, et à Raphaëlle, mademoiselle « par hasard » ! Merci à Yannick, sans qui l'interface n'existerait pas...
Merci enfin à tout ceux qui ont pris le temps de m'écouter et qui m'ont encouragés. Je pense particulièrement à Bilbo, Kanak, Steph, Flo et tous ceux du « tireur ».

Enfin, je voudrais remercier mes parents et mes sœurs pour toute l'affection qu'ils m'ont porté au cours de toutes ces années.

Table des matières

Table des figures

Notations

$a, \ldots, z$; $\alpha, \ldots, \zeta$: scalaires $\in \mathbb{R}$,
$\vec{a}, \ldots, \vec{z}$; $\vec{A}, \ldots, \vec{Z}$; $\vec{\alpha}, \ldots, \vec{\zeta}$: vecteurs $\in \mathbb{R}^3$,
$||\vec{a}||$: norme du vecteur $\vec{a}$,
$\vec{a} \cdot \vec{b}$: produit scalaire des vecteurs $\vec{a}$ et $\vec{b}$,
$\vec{a} \wedge \vec{b}$: produit vectoriel des vecteurs $\vec{a}$ et $\vec{b}$,
$(\vec{a}, \vec{b}, \vec{c})$: produit mixte des vecteurs $\vec{a}$, $\vec{b}$ et $\vec{c}$

$[A]$: matrice de dimension $n \times m$; le terme général est a_{ij},
$[A]^T$: matrice transposée de la matrice $[A]$,
$[A]^{-1}$: matrice inverse de la matrice $[A]$

u, v : paramètres descripteurs de courbes ou surfaces,
p, q : degré d'une courbe ou d'une surface,
$u_i, u_{i+1}, u_{i+2}, \ldots, u_{i+j}$: séquence nodale telle que les nœuds $u_i \leq u_{i+1} \leq u_{i+2} \leq \ldots \leq u_{i+j}$ forment une suite croissante,
$i!$: factorielle i,
$\left(\begin{array}{c} p \\ i \end{array} \right) = \dfrac{p!}{(p-i)!i!}$: coefficients du binôme de NEWTON pour un polynôme de degré p,
$B_i^p(u)$: $i^{\text{ème}}$ polynôme de BERNSTEIN de degré p,
$N_i^p(u)$: $i^{\text{ème}}$ fonction de base B-spline constituée de polynômes de degré p

$\vec{P}_i$: vecteur définissant le $i^{\text{ème}}$ sommet du polygone caractéristique d'une courbe,
$\vec{P}_{ij}$: vecteur définissant le sommet d'indice i et j du réseau caractéristique d'un carreau,
$\vec{P}_{ij}^{[k]}$: vecteur associé au $i^{\text{ème}}$ sommet du polygone caractéristique après k itérations d'un algorithme,

Introduction générale

En C.F.A.O.[1], les modeleurs surfaciques et volumiques constituent un excellent outil de conception, permettant de représenter et fabriquer les objets, à l'aide d'un modèle numérique. Le problème inverse, la reconstruction d'une représentation géométrique d'un objet physique existant, est une opération de plus en plus indispensable dans de nombreuses applications, comme le *reverse engineering*, le prototypage rapide, l'inspection ou tout simplement la visualisation. Les récents développements des technologies d'acquisition 3D permettent d'obtenir des nuages de points structurés ou non selon les domaines. La rapidité de ces technologies ainsi que la précision des données saisies ont été largement améliorées ces dernières années. Ceci justifie leur utilisation dans le domaine du *reverse engineering* [MPS+94].

Selon les domaines, diverses technologies peuvent être utilisées pour acquérir des représentations numériques des objets étudiés. La surface d'un objet peut être numérisée à l'aide d'un scanner utilisant un laser, le résultat étant un nuage de points sur-échantillonnés, regroupant éventuellement plusieurs vues. Des courbes de niveau peuvent être extraites d'images satellite en vue d'élaboration de cartes géographiques ou océanographiques. Dans le domaine médical, des informations sur les organes internes peuvent être obtenues grâce aux techniques de tomographie (CT) ou d'imagerie par résonance magnétique (IRM). La microscopie par émission d'électrons et les ultrasons sont deux autres techniques également utilisées. Un objet physique peut aussi être découpé en tranches plus ou moins épaisses pour être photographiées. Cette technique est le plus souvent utilisée en géologie, mais d'autres domaines peuvent y avoir recours, on citera notamment le *Visible Human Project* [Vis].

Nous nous sommes intéressés à la reconstruction d'un modèle géométrique à partir de données numérisées. Ce problème a suscité de nombreuses recherches ces dernières années, et les méthodes existantes peuvent être classifiées en deux grandes catégories. Dans la première, nous retrouvons celles qui n'exploitent pas la structuration des données ou qui travaillent sur des données de structure inconnues [HDD+92, Mal95, EH96], et la seconde regroupe celles qui traitent les données selon la structuration (grilles, contours, ...) [MSS92, PT95, PK96]. Le laboratoire LIGIM travaille depuis plusieurs années sur des projets dans le domaine médical. Ceci nous a naturellement amené à nous intéresser plus particulièrement à la reconstruction de formes anatomiques définies par un ensemble de sections. Cette représentation structurée des données initiales est très fréquente dans le domaine médical, mais aussi dans beaucoup d'autres domaines d'application.

Les propriétés particulières des formes anatomiques nous ont conduit à utiliser les sur-

1. Conception et Fabrication Assistée par Ordinateur

faces définies par produit tensoriel. Celles-ci permettent de modéliser des surfaces lisses ne comportant pas d'arêtes vives. Par ailleurs, les organes peuvent comporter des branches, ce qui pose des problèmes au niveau topologique et nécessite l'utilisation de plusieurs surfaces paramétriques, avec tous les problèmes de continuité que cela engendre.

En informatique graphique, la modélisation et l'animation sont des thèmes de plus en plus liés. Une représentation géométrique n'est pas seulement faite pour être visualisée, mais aussi pour être animée. Les perspectives de la simulation sont très vastes, dans des domaines aussi variés que la visualisation, les simulateurs ou la réalité virtuelle. Il est vite apparu que la simulation pouvait avoir un apport bénéfique dans le domaine médical. Ainsi, la réalisation de simulateurs d'opérations chirurgicales a fait l'objet d'un intérêt particulier, puisqu'ils permettent de remplacer avantageusement une formation en bloc opératoire.

Les formes anatomiques sont des objets déformables. De ce fait, leur représentation par des modèles paramétriques a vite été une limitation. Ceux-ci permettent de prendre en compte les mouvements rigides, mais les déformations sont plus difficiles à gérer, malgré quelques approches intéressantes [TF88b, NFD95, RBB97]. Pour obtenir des comportements réalistes, de nouveaux modèles d'objets déformables ont été développés. On distingue principalement deux grandes catégories qui sont les modèles adaptés à de petites déformations, et ceux permettant des déformations importantes, voire des changements de topologie. Nous nous sommes principalement intéressés à ceux de la seconde classe qui autorisent une très grande variété de comportements. Nous avons porté une attention toute particulière aux systèmes de particules qui sont des modèles très simples, adaptés à la simulation de formes anatomiques, déformables ou non.

Cette thèse se décompose en deux grandes parties.

La première est consacrée à la reconstruction de surfaces paramétriques sur un ensemble de sections. Après avoir positionné le problème et décrit les avantages et les limitations des méthodes existantes, nous présentons notre approche pour reconstruire des surfaces simples, ainsi que l'évolution vers la prise en compte des embranchements.

La seconde partie concerne la reconstruction et l'animation des objets déformables. Nous commençons par décrire les raisons qui nous ont amené à traiter ce thème. Nous exposons l'état de l'art dans ce domaine et consacrons un chapitre particulier aux systèmes de particules. Puis, nous présentons notre méthode de reconstruction d'objets déformables à l'aide de particules en plusieurs couches, ainsi que la phase d'animation des objets créés.

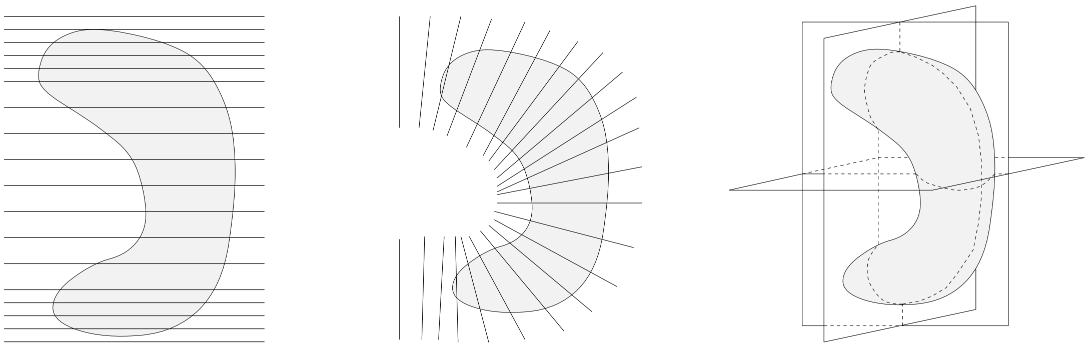
En conclusion, nous proposerons quelques perspectives et développements qui paraissent très prometteurs.

Première partie

Reconstruction de modèles de surface

1 Position du problème

La représentation des données par un ensemble de contours est de plus en plus utilisée dans de nombreux domaines, et en particulier dans le domaine médical. Les contours sont obtenus par l'intersection de la surface de l'objet avec une série de plans de coupe (ou sections). La figure 1.1 illustre différentes configurations. La plupart du temps, les différents plans de coupe ne s'intersectent pas dans la zone de l'objet (sous-figures (a) et (b)). La sous-figure (c) représente un cas de plans de coupe s'intersectant.



(a) Sections parallèles (b) Sections non parallèles (c) Sections orthogonales

Fig. 1.1 – *Configurations des sections*

Après la phase d'acquisition, un contour est souvent représenté par une ligne brisée qui est une approximation linéaire par morceaux du contour réel continu. Il peut arriver que la coupe par un plan définisse plusieurs contours, qui peuvent appartenir à un seul ou à différents objets (Fig. 1.2).

Par la suite, nous parlerons de topologie « simple » lorsque les données sont disposées en sections successives ne comportant qu'un seul contour. Nous avons des données de topologie « complexe » dès qu'une des sections comporte plus d'un contour. Généralement la topologie des données conditionne celle de la surface. Ainsi, des contours de topologie « simple » seront reconstruits par des surfaces « simples », c.-à-d. homéomorphes à un cylindre, une sphère ou un tore. De même, des contours « complexes » seront reconstruits par une surface de topologie « complexe ». Mais, ce n'est pas toujours le cas. La figure 1.3 illustre une telle

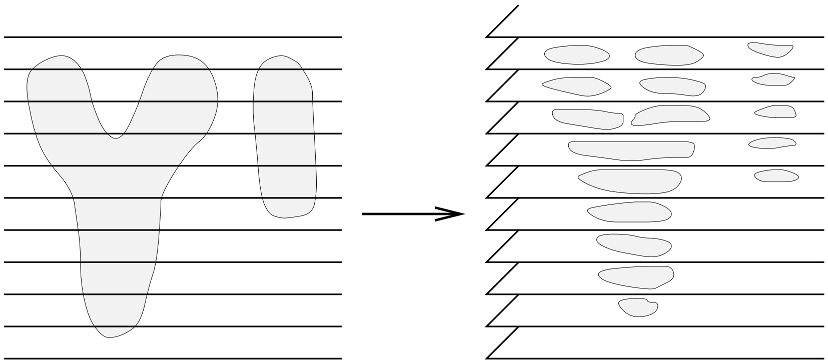


FIG. 1.2 – *Exemple de sections 2D d'un objet 3D*

configuration [ESS91].

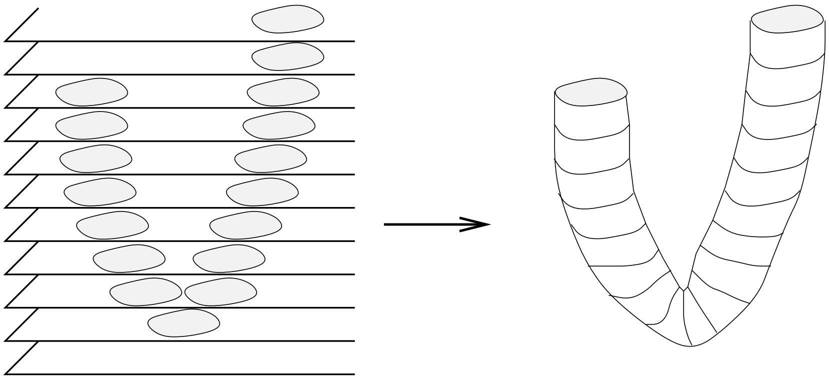


FIG. 1.3 – *Données « complexes » reconstruits à l'aide d'une surface « simple »*

Le but de la reconstruction est de fournir une description géométrique de l'objet numérisé, en se basant principalement sur le calcul d'un modèle de surface, qui sera une approximation de l'ensemble des données. Cette opération est classiquement décomposée en trois étapes [MSS92], plus ou moins liées selon la méthode employée :

1. mise en correspondance des contours ;
2. reconstruction de surfaces entre les contours ;
3. prise en compte des embranchements.

Dans le cas de données de topologie « simple », la première étape ne pose pas de problème. Par contre, dès qu'une section contient deux contours et plus, cela devient plus difficile. À l'issue de la première étape, les données sont séparées en plusieurs ensembles de sections de topologie « simple ». Il existe deux classes de solutions. La première ne tient compte que des branches simples et les embranchements sont mis de coté jusqu'à l'étape 3. La seconde classe modifie les contours ou en ajoute d'autres pour supprimer les zones d'embranchements et se ramener aux cas simples.

Chaque ensemble de contours est reconstruit par une surface « simple », ce qui constitue l'étape 2. Une approche largement répandue est de connecter les contours par un ensemble de triangles. Cette approximation planaire de la surface peut ensuite être lissée. Une autre approche consiste à reconstruire directement une surface lisse en utilisant par exemple des surfaces paramétrées. La reconstruction de la surface est un problème facilement résoluble lorsque les contours sont de formes simples. Par contre, cette tâche peut être compliquée par des dissimilarités dans la forme ou l'orientation des contours, comme cela arrive fréquemment dans les données réelles.

L'étape 3 consiste à assembler les différentes surfaces pour fournir une représentation finale complète. Si nécessaire, il faut construire les surfaces de raccordement dans les zones d'embranchement. Il s'agit de mettre une ou plusieurs surfaces entre les m contours d'une section et les n (très souvent $n \neq m$) autres contours de la section suivante appartenant à un même objet. Éventuellement, une étape de raccord peut être nécessaire pour obtenir un modèle continu ou même lisse. Les différents cas de connexion sont répertoriés sur la figure suivante (Fig. 1.4) :

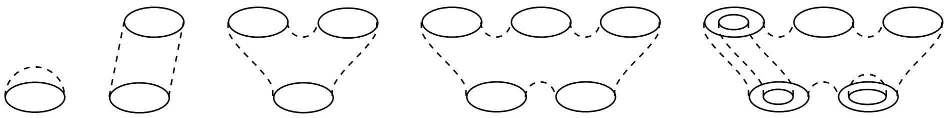


FIG. 1.4 – *Différents types d'embranchements, de gauche à droite : pas de connexion, connexion simple, branchement simple, branchement multiple et branchement multiple avec trous*

Les solutions proposées peuvent être classifiées dans deux catégories : celles utilisant un maillage polygonal [MSS92, OPC96] ou polyédrique [BG92], et celles s'appuyant sur des carreaux de surface paramétrique [ESS91, HLLK94, PK96]. On peut aussi distinguer pour chacune d'elles, celles dont la technique d'embranchement est indépendante du type de surface utilisé pour chaque branche et celles qui en sont dépendantes. Dans la première classe, nous retrouvons la plupart des méthodes polygonales. La construction de la surface entre les branches ne tient pas compte de la façon dont sont reconstruites les branches. Par contre, pour la seconde classe, où l'on trouve essentiellement les méthodes paramétriques, la surface d'embranchement est construite en fonction des branches.

2 État de l'art

Dans ce chapitre, nous exposons l'état de l'art dans le domaine de la reconstruction de surfaces à partir de contours. La première section est consacrée aux méthodes permettant la mise en correspondance des contours. La seconde présente les avantages et les inconvénient des approches existantes pour construire les surfaces. Pour chaque grande classe de méthodes, nous distinguons le cas des surfaces sans embranchements, et le cas des contours multiples (avec branches) lorsque cela est possible.

2.1 Mise en correspondance

Le problème de la mise en correspondance peut apparaître dès qu'une section contient plusieurs contours. Il faut alors déterminer les connexions qui existent entre les contours des sections adjacentes. La plupart du temps, l'espacement entre les sections est trop important pour pouvoir lever toute ambiguïté. La figure 2.1 représente différentes topologies possibles à partir du même jeu simple de contours, et illustre toute la difficulté du problème. Deux contours appartenant à deux sections adjacentes peuvent former une liaison simple, un embranchement ou faire partie de deux objets différents. Des informations complémentaires sur

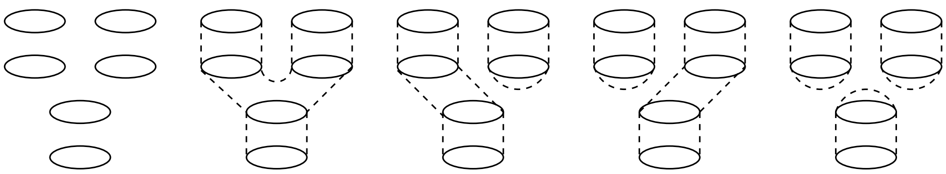


FIG. 2.1 – *Contours initiaux et exemples de topologies possibles*

la nature des objets à reconstruire sont souvent employées pour fournir une correspondance acceptable.

2.1.1 Recouvrement des contours

Quand les sections sont suffisamment denses, l'espacement relatif permet de résoudre la mise en correspondance en examinant le recouvrement des contours de sections adjacentes [ZJH87]. Le taux de recouvrement est calculé à partir des boîtes englobantes de chaque contour. Cette information est couplée à la notion de contour extérieur ou intérieur pour déterminer des connexions cohérentes. La limitation principale de cette méthode est son incapacité à traiter les cas où deux contours successifs de la même branche sont suffisamment décalés pour ne pas se recouvrir. Cela se produit par exemple lorsque l'axe de la branche est très différent de l'orientation des sections, ou que l'échantillonnage des sections est peu dense. Dans ce cas, la méthode trouvera plusieurs objets séparés à la place d'un seul et même objet.

2.1.2 Cylindres généralisés

Meyers et al. [MSS92] proposent une extension de la méthode de Soroka (1978) pour résoudre la mise en correspondance des contours. Chaque contour est reconstruit à l'aide d'une ellipse. En calculant la déviation standard, les contours sont classés comme elliptiques (simples) ou complexes, et seront traités différemment par la suite. Le processus peut être décomposé en trois étapes :

1. Assembler les cylindres elliptiques à partir des contours
   Si un contour est elliptique et qu'il satisfait aux conditions de variation linéaire des paramètres d'un cylindre, alors il est ajouté à ce cylindre, sinon un nouveau cylindre est créé.
   Si le contour est complexe, on recherche un sous-contour elliptique qui pourra être ajouté à un cylindre.
2. Assembler les objets à partir des cylindres
   Il s'agit à cette étape de trouver toutes les connexions entre les cylindres.
3. Analyser les connexions pour déterminer les branches.
   Il y a trois cas de connexion : deux extrémités de cylindres fusionnent en une autre extrémité, une extrémité est connectée à une autre, et une extrémité fusionne avec le corps d'un cylindre.

La mise en correspondance finale dépend beaucoup de l'ordre de traitement des contours, à cause des petits cylindres constitués de un ou deux contours. Une erreur peut facilement être propagée et les résultats obtenus sont peu satisfaisants.

2.1.3 Arbres

Les restrictions de la méthode basée sur les cylindres généralisés ont conduit les mêmes auteurs à proposer une solution plus globale [MSS92]. Comme précédemment, les contours sont approximés par une ellipse. Ensuite, un arbre couvrant minimum, est construit en liant chaque contour à tous les autres contours des sections adjacentes. Le coût associé à chaque arête est calculé en fonction de la position des centres et de la longueur des axes des ellipses. Ensuite, cet arbre est décomposé en segments, un nouveau segment étant créé pour chaque embranchement de l'arbre. Les contours d'un même segment forment un tube et sont reliés ensembles. Les informations de coût des arêtes de l'arbre permettent de déterminer les connexions entre les tubes. Comme la recherche de l'arbre couvrant minimum est une opération globale, la méthode n'est pas sujette à la propagation d'erreurs locales. Par contre, la structure d'arbre utilisée convient bien pour les objets à branches, mais elle n'est pas adaptée à des objets contenant des cycles (par exemples des tores), et dans ce cas un type de graphe plus général doit être employé.

2.2 Reconstruction de surface sur des contours

2.2.1 Maillage

Parmi toutes les approches de reconstruction de surfaces à partir de contours, le maillage triangulaire est celle qui a été la plus utilisée. Le résultat est un ensemble de triangles qui présente l'avantage d'être simple à manipuler et à visualiser. Deux catégories de méthodes sont à distinguer : celles qui forment une triangulation optimale en fonction de critères métriques comme l'aire des triangles ou le volume englobé, et celles qui prennent en compte la pertinence topologique de la triangulation. Une comparaison assez complète des différentes méthodes de triangulation de contours peut être trouvée dans la référence [Mey94].

2.2.1.1 Méthodes optimales

La construction d'une approximation triangulée optimale de la surface entre deux contours peut se réduire à un problème de recherche de chemin optimal dans un graphe. Fuchs et al. [FKU77] forment un graphe dont les sommets représentent l'ensemble des liaisons possibles des points entre les deux contours, et les arcs l'ensemble des triangles élémentaires. Toutes les surfaces acceptables entre deux contours peuvent être déterminées par association à des cycles dans ces graphes. Mais cela produit un nombre exponentiel de surfaces. Le problème est réduit en associant un coût à chaque arc et en cherchant des cycles de coût minimal. Pour faciliter cette recherche, chaque fois qu'un cycle, de coût minimal et passant par un nœud donné, est trouvé, il peut être utilisé pour réduire l'espace de recherche. L'algorithme est alors très efficace, et la complexité de la triangulation est $O(n^2 \log n)$ si n est le nombre de points.

Divers types de graphes ont par ailleurs été utilisés, notamment la généralisation du graphe discret en un graphe continu [SK91].

2.2.1.2 Méthodes heuristiques

À cause du coût des méthodes d'optimisation, des algorithmes basés sur des heuristiques simples à mettre en œuvre ont été proposés. Ces algorithmes ont un coût linéaire, mais produisent la plupart du temps des résultats assez décevants.

Le critère le plus simple est celui de la longueur de corde. Il est utilisé pour choisir à chaque étape entre les deux arêtes possibles (Fig. 2.2). Pour obtenir de meilleurs résultats, les distances peuvent être normalisées. Cependant, la méthode échoue dans des cas courants. Une amélioration a été proposée, elle consiste à modifier l'orientation de l'arête commune à deux triangles lors d'une phase d'optimisation locale (Fig. 2.3).

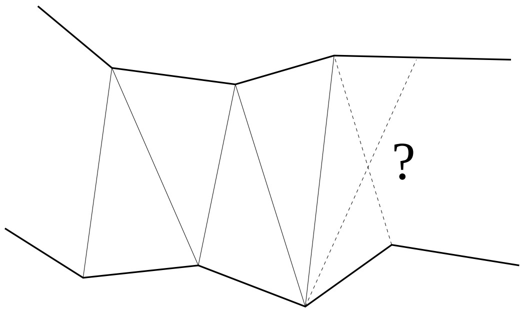


FIG. 2.2 – *Choix de l'arête suivante*

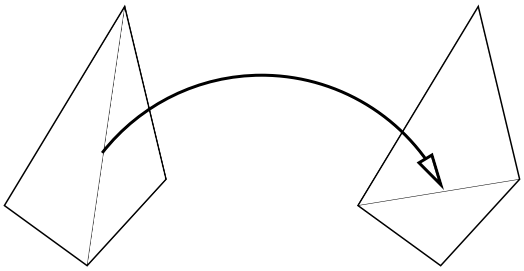


FIG. 2.3 – *Changement local d'orientation d'une arête*

Lorsque les contours ont des formes, des orientations ou des positions très différentes, ce genre d'algorithme produit des surfaces auto-intersectantes ou avec des torsions.

EKOULE et al. [EPO91] proposent une méthode de triangulation entre les contours concaves. Ils commencent par construire une surface de triangulation entre les enveloppes convexes des contours en minimisant un critère global de longueur de corde. Ensuite, ils mettent en correspondance les parties concaves du contour avec son enveloppe convexe, ce qui leur permet d'en déduire une triangulation entre les contours concaves. Cette méthode donne de bons résultats lorsque les contours présentent de grandes dissimilarités de forme, mais elle n'est pas satisfaisante quand les contours sont réguliers.

MEYERS et al. [Mey93] proposent un algorithme basé sur les techniques d'analyse multirésolution et d'ondelettes. Une description de ces outils peut être trouvée dans la référence [Coh92]. L'algorithme peut se décomposer en trois étapes :

1. Réduire la taille des données initiales en utilisant l'analyse multirésolution pour trouver une approximation basse résolution des contours ;

2. Construire une triangulation optimale sur ces contours réduits. L'algorithme utilisé est celui proposé par FUCHS et al. [FKU77] ;

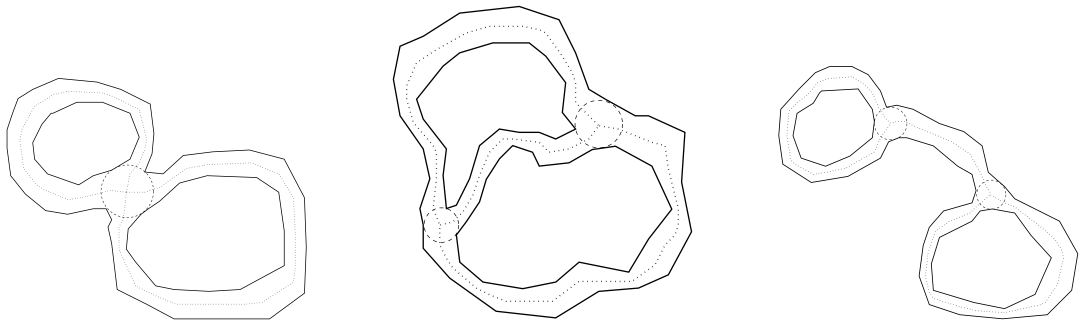
3. Ajouter les détails aux contours pour obtenir la résolution initiale. Pour chaque niveau de détail, des arêtes sont créées à partir des nouveaux points introduits. Les nouveaux triangles sont optimisés sur des critères locaux (inversion d'arête). Il est bien sûr possible de ne pas reconstruire tous les niveaux de détail pour obtenir un modèle plus compact.

La triangulation finale n'est plus optimale, mais elle en est très proche. Le fait d'appliquer la phase d'optimisation sur des contours réduits, mais qui conservent néanmoins l'allure du contour initial, permet de passer de $O(n^2 \log n)$ à $O(n \log n)$ en complexité, et de $O(n^2)$ à $O(n)$ en mémoire utilisée. Le gain est donc considérable pour un nombre élevé de points et autorise une reconstruction interactive pour corriger les erreurs de maillage qui pourraient apparaître. Car même si cette méthode semble être une des plus efficaces, elle ne résout pas tous les problèmes, dont certains sont inévitables et ne peuvent être résolus que manuellement.

2.2.1.3 Prise en compte des embranchements

La majorité des algorithmes de maillages d'objets définis par des contours n'est applicable qu'aux cas simples d'un contour vers un autre. La plupart des méthodes d'embranchement introduisent donc un contour composite arbitraire pour satisfaire à cette contrainte.

Ainsi, Ekoule et al. [EPO91] décomposent le problème de multibranches en un ensemble de sous-problèmes monobranches. Ils créent un contour simple au niveau intermédiaire qui sera lié à celui du dessus ainsi qu'à celui du dessous. Cependant, ce contour est une approximation médiocre des contours adjacents. Meyers et al. [MSS92] proposent une meilleure approche puisqu'ils étudient la forme des contours pour déterminer le contour composite. Ils identifient alors trois situations différentes (canyon, selle de cheval et crête) à l'aide de l'axe médian. Ils utilisent ensuite différentes méthodes de maillage pour reconstruire chacun des types d'embranchement en modifiant le contour initial de la section inférieure ou supérieure ou les deux (Fig. 2.4).



(a) Selle de cheval (b) Canyon (c) Crête

Fig. 2.4 – *Différentes dispositions des contours*

Oliva et al. [OPC96] proposent d'interpoler les contours avec un nombre arbitraire de contours intermédiaires. Ils calculent le squelette linéaire des aires de différences entre les contours adjacents à l'aide d'un « Angular Bisector Network » (ABN), dont les caractéristiques sont proches de celles du diagramme de Voronoï généralisé, excepté que tous ses éléments sont des segments linéaires. Cet ABN est ensuite décomposé en chaînes simples qui aideront à la triangulation. De plus, l'ABN peut être considéré comme un nouveau contour intermédiaire et le processus d'interpolation peut être appliqué à nouveau pour raffiner la forme finale de l'embranchement. Cette méthode permet de prendre en compte les irrégularités locales en ne tenant compte que de la topologie et de la morphologie des contours adjacents. Aucune autre information n'est nécessaire.

Boissonnat et Geiger [BG92] ont une approche différente du maillage. Ils construisent une triangulation de Delaunay tridimensionnelle entre les points des contours adjacents. La première étape est une triangulation 2D contrainte de chaque contour. Dans la deuxième étape, le diagramme de Voronoï associé est utilisé pour construire un ensemble de tétraèdres entre les contours. Les tétraèdres qui se trouvent à l'extérieur des contours sont supprimés. Cette méthode est employée pour chaque paire de contours consécutifs, qu'ils forment un embranchement ou non. Les faces externes forment la surface de visualisation, alors que la structure polyédrique interne peut être utilisée pour d'autres applications, comme pour la simulation de mouvement ou dans des méthodes par éléments finis.

2.2.1.4 Limitations des méthodes de maillage

Lorsque les contours sont très différents, il est difficile de fournir une topologie correcte. Gitlin et al. [GOS93] ont démontré que dans certains cas, il n'est pas possible de construire un maillage formant un seul polyèdre (non intersectant).

La plupart des méthodes de maillages considèrent la triangulation comme étant le résultat final. Cependant, cette représentation peut ne pas être satisfaisante pour différentes raisons.

Un inconvénient des méthodes de maillage est le grand nombre de triangles générés. C'est pourquoi des méthodes de simplification ont été proposées. Elles doivent répondre à deux critères : respect de la topologie et approximation correcte de la géométrie. Schroeder et al. [SZL92] proposent un algorithme en plusieurs étapes de simplification de maillage. Pour chaque passe, chaque point est évalué et classé. S'il correspond aux critères, il est enlevé ainsi que tous les triangles dont il était un des sommets. Le trou résultant est de nouveau triangulé, sans ce point, ce qui entraîne une diminution du nombre de triangles. Le processus est répété jusqu'à atteindre la précision souhaitée. Les résultats montrent qu'une réduction de 90% des données peut encore fournir un modèle acceptable.

Eck et Hoppe [EH96] partent d'une triangulation initiale dense. Ils joignent les triangles deux à deux à l'aide des méthodes d'optimisation de graphes et recherchent automatiquement un réseau de carreaux quadrangulaires ainsi qu'une paramétrisation des données appartenant à chaque carreau. Ceci leur permet d'obtenir un ensemble de carreaux B-spline jointifs, globalement G^1. Quand l'erreur d'approximation est supérieure à celle spécifiée par l'utilisateur, une subdivision adaptative est appliquée. Cette méthode permet de créer une surface lisse sur un ensemble de triangles. Elle est donc tout à fait adaptée au lissage des maillages obtenus à partir de contours, elle peut même être étendue aux objets avec branches ou trous puisqu'elle supporte les topologies complexes.

2.2.2 Surface paramétrique

Les surfaces paramétriques sont des outils très intéressants pour reconstruire des surfaces. En effet, leur formulation compacte est à opposer au nombre considérable de triangles générés par les méthodes de maillage pour obtenir une fine précision. De plus, leur caractère lisse est appréciable dans de nombreux cas, par exemple lors de la visualisation puisque les objets présentent un aspect plus naturel. C'est encore plus vrai pour les formes anatomiques.

2.2.2.1 À l'aide d'une surface triangulée

JOHNSTONE et SLOAN [JS95] reconstruisent des surfaces définies par un produit tensoriel, en se servant d'une triangulation optimale comme guide pour extraire les iso-paramétriques. La première étape de leur méthode consiste à générer une triangulation entre les contours initiaux horizontaux. Ensuite, ils placent des germes et les propagent orthogonalement aux contours pour former des lignes de flux. Ils maîtrisent leur nombre en espaçant correctement les germes et en fusionnant les flux lorsqu'ils deviennent trop proches ou en en créant de nouveaux. Lors de la deuxième étape, ils triangulent ces iso-paramétriques en v. Comme précédemment, ils placent des germes et les propagent pour former les iso-paramétriques en u, en les forçant cette fois à décrire une ligne fermée si les contours sont fermés. À ce stade, on dispose de points sur une grille régulière sur lesquels on peut facilement construire une surface. La troisième étape consiste donc en une interpolation ou une approximation de ces points par une surface définie par un produit tensoriel. Il est à noter que si la triangulation est optimale [FKU77], la surface B-spline ne l'est pas. Mais le gain en temps est conséquent par rapport à la qualité de la surface finale, qui peut être ajustée en faisant varier le nombre de lignes de flux. L'inconvénient est que chaque étape fournit un résultat approché à partir de données qui avaient déjà été approchées à l'étape précédente. Ce procédé appauvrit successivement les données jusqu'à obtenir le modèle final, et les erreurs commises sont facilement transmises à l'étape suivante de l'algorithme.

2.2.2.2 « Skinning »

Le « skinning » est une solution particulièrement intéressante au problème de la reconstruction de contours, puisque cette opération consiste à tendre une surface lisse, une peau, sur un ensemble de courbes.

L'opération de « skinning » se décompose en plusieurs étapes :

1. Définition d'un degré commun et d'un même vecteur nodal dans la direction u, mise en compatibilité des paramètres pour chaque contour à reconstruire ;
2. Reconstruction des contours par une courbe B-spline. Il est possible d'utiliser une méthode de correction adaptative des valeurs de paramètres pour optimiser l'approximation [RF89] ;
3. Définition d'un vecteur nodal commun et choix du degré, dans la direction v ;
4. Reconstruction de la surface par interpolation ou lissage des points de contrôle des courbes obtenues à l'étape 2.

Les étapes 1 et 2 sont éventuellement effectuées de manière itérative. Le résultat final est une représentation compacte, de continuité élevée. Elle est de plus parfaitement compatible avec la majorité des systèmes de C.A.O. .

Cette technique a été utilisée la première fois pour créer des surfaces sur des contours définis manuellement. WOODWARD [Woo87, Woo88] a présenté une approche de type C.A.O. pour construire des surfaces paramétriques sur des courbes porteuses ouvertes. Le contrôle de la forme est réalisé par la définition d'une courbe longitudinale (« spine curve »), qui va donner la valeur des dérivées le long de la surface. Elle permet de réduire le problème des surfaces tridimensionnelles au cas plus intuitif des courbes planaires.

Park et Kim [PK96] et Piegl et Tiller [PT96] ont récemment proposé des méthodes pour optimiser le nombre de points de contrôle lors de la mise en commun des vecteurs de nœuds. Ces processus sont itératifs. Ils commencent par définir un vecteur nodal commun minimum, qui est utilisé lors de la reconstruction de toutes les courbes. Si l'erreur commise entre une courbe et son contour est trop importante, alors le vecteur est raffiné à l'aide des techniques d'insertion de nœuds. Le même processus peut être utilisé dans la direction longitudinale, mais le gain est moins important car les contours ne sont généralement pas très nombreux. Le temps de calcul pour ces solutions optimisées est par contre très important, mais la compacité des solutions le compense largement si le modèle obtenu est utilisé après dans d'autres applications.

2.2.2.3 Prise en compte des embranchements

Les surfaces paramétrées permettent de reconstruire chaque branche, mais elles peuvent aussi être utilisées pour reconstruire la surface entre les branches.

Ebel et al. [ESS91] ont opté pour une approche de type C.A.O. , les parties tubulaires sont reconstruites par ajustement d'un cylindre généralisé dont la position initiale est fournie par l'utilisateur. Les points 3D, échantillonnés sur le modèle cylindrique obtenu, sont ensuite interpolés par une surface B-spline uniforme bicubique. Les embranchements sont reconstruits par prolongement de ces surfaces. Après avoir déterminé manuellement le sommet des carreaux de raccord, chaque demi-tube est joint à un autre par un réseau de courbes issues des isoparamétriques des surfaces B-splines . Ces courbes sont ensuite habillées par un carreau rectangulaire de Bézier /Gregory de degré 3×4 pour respecter la continuité G^1. Les espaces libres entre les carreaux rectangulaires sont comblés par des carreaux triangulaires de Bézier/Gregory de degré 5, toujours pour respecter la continuité. La figure 2.5 illustre la disposition des différents carreaux.

Cette méthode présente l'avantage de fournir une représentation lisse et compacte, avec peu de carreaux, d'objets définis par des contours. Cependant, la reconstruction est limitée à des structures tubulaires possédant des embranchements simples, nécessitant l'intervention active de l'utilisateur. Les restrictions sur le modèle sont justifiées par les auteurs qui remarquent que les éléments anatomiques qu'ils veulent traiter présentent souvent une régularité et une simplicité qui leur permettent de correspondre au modèle proposé.

Park et Kim [PK96] proposent une généralisation de leur méthode de « skinning » aux objets à branches. Les contours situés dans les zones d'embranchement, entre le tronc et les branches, sont tout d'abord triangulés à l'aide d'une des méthodes de maillage présentées au chapitre 2.2.1.3. Ensuite, pour réduire le nombre de triangles mais aussi pour obtenir une représentation plus lisse, ils construisent des carreaux de surface sur ce maillage [PK95]. Ils commencent par définir une surface grossière constituée de carreaux triangulaires de Bézier, qu'ils affinent ensuite jusqu'à obtenir la précision souhaitée. Cette méthode leur permet de prendre en compte n'importe quel type d'embranchement. Ils l'utilisent aussi pour reconstruire les extrémités de l'objet, ce qui leur procure un modèle complet d'objets à branches, ayant

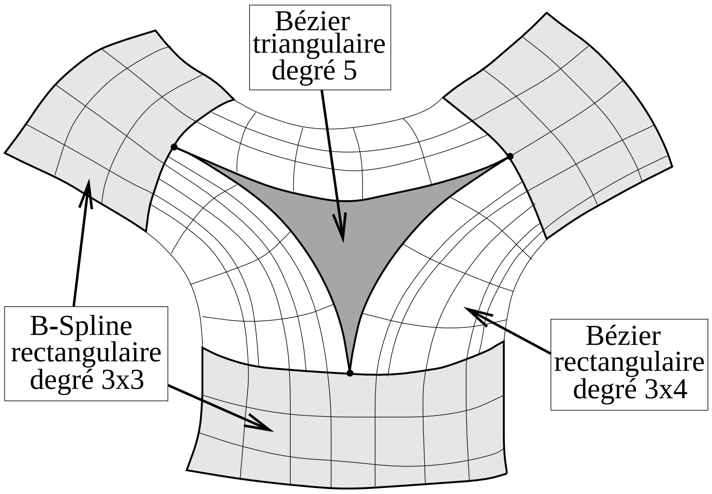


FIG. 2.5 – *Disposition des carreaux et des lignes de prolongement*

une continuité élevée C^2, mais au prix d'un grand nombre de petits carreaux de surface.

HAN et al. [HLLK94] créent un nouveau contour par interpolation, qui est une estimation de la position où les branches se séparent. Pour cela, ils calculent des vecteurs de déplacement entre le contour inférieur et les branches. Ces vecteurs sont calculés en fonction de la distance euclidienne, mais en tenant compte des zones concaves. Ensuite, de nouveaux contours sont générés de façon itérative. Chaque nouveau contour doit être testé pour déterminer s'il s'agit du contour exact ou les branches prennent naissance. Si c'est le cas, il est alors dupliqué. Le contour de branchement est séparé en deux contours tangents qui seront associés chacun à une branche. Le contour complet sera relié au tronc. Enfin, l'objet sera reconstruit en tendant une surface NURBS sur les trois groupes de contours. Cependant, au niveau du branchement, on dispose de trois contours distincts, et qui plus est de topologies différentes. Leurs approximations NURBS seront par conséquent différentes. La continuité C^0 sera difficilement obtenue, et plus particulièrement au point critique où les branches se séparent.

2.2.3 Représentation par fonctions

Les surfaces implicites ont aussi été utilisées pour la reconstruction de contours. PONTIER et al. [PSV98] ont proposé une méthode basée sur les modèles implicites à squelettes. La première étape est la détermination du squelette de chaque contour, à l'aide de la triangulation de Delaunay. Ce squelette peut être simplifié en enlevant les petits triangles non-significatifs. Ensuite, une mise en correspondance est effectuée entre ces squelettes 2D. Pour cela, les auteurs utilisent le diagramme de Delaunay 3D pour déduire le squelette 3D de l'objet à reconstruire. Il est alors possible d'approximer cet objet à l'aide d'une surface implicite basée sur des squelettes. Pour obtenir un modèle très simple, c.-à-d. utilisant un champ de potentiel uniforme, des poids sont affectés aux points du squelette 3D.

SAVCHENKO et al. [SPOK95, SP95] ont étudié les possibilités offertes par les représentations par fonctions pour interpoler un ensemble de contours, dans un but de visualisation mais aussi de transformation de l'objet obtenu. Le schéma de reconstruction est le suivant : dans un premier temps, un objet porteur est défini. Il s'agit d'une estimation initiale d'une description implicite du contour. Dans le cas le plus simple, cela peut être un disque. Ensuite, on calcule les valeurs de la fonction décrivant l'objet porteur, celles-ci sont approximées par une courbe B-spline. Entre les contours, l'objet peut être calculé par un mélange linéaire ou quadratique. Cette méthode peut être facilement généralisée aux contours multiples dans une même section. Cela est réalisé en générant autant de descripteurs F_i que de contours, puis en les fusionnant à l'aide de l'opération d'union $F = F_1 + F_2 + \sqrt{(F_1^2 + F_2^2)}$. La représentation par fonctions a de nombreux avantages. Elle permet notamment de s'affranchir des problèmes de concavités et d'embranchements difficilement résolus par d'autres méthodes. L'inconvénient reste bien sûr le temps de calcul. Les objets créés peuvent aussi être facilement modifiés ou sculptés en utilisant les opérateurs ensemblistes de différence ou de jointure.

2.2.4 Méthodes globales

Contrairement à la plupart des algorithmes publiés jusqu'à présent qui ne s'intéressent à la fois qu'à une voire deux des étapes de la reconstruction de surfaces sur des contours, quelques méthodes considèrent en même temps la mise en correspondance, le maillage et les embranchements.

C'est le cas de BAJAJ et al. dans [BCL95], qui définissent trois contraintes leur permettant d'obtenir un maillage triangulé qui corresponde à l'objet réel :

- le modèle doit être composé d'un ensemble fermé de surfaces ou de polyèdres, ceci pour éviter les erreurs de topologies comme l'auto-intersection par exemple ;
- les données ne doivent pas faire de plis ou d'oscillations, ce qui signifie qu'une verticale soit n'intersecte pas la surface, soit l'intersecte en un point ou selon un segment ;
- le ré-échantillonnage de la surface doit permettre de retrouver les données initiales. La surface doit donc interpoler ces données initiales.

À partir de ces critères, ils déterminent pour la première fois des règles précises pour résoudre le maillage et l'embranchement. Ils résolvent ainsi le problème des méthodes basées sur heuristiques, qui pouvaient échouer selon la topologie des données initiales. Une des contributions majeures de ce travail est l'algorithme multipasses de maillage. En effet, il est difficile de voir quand alterner entre le maillage normal entre deux contours et l'embranchement, et la plupart des méthodes échouent à cause de ça. C'est pourquoi les auteurs commencent par créer les triangles optimaux, laissant de coté toutes les zones à problèmes (embranchements, trous, zones dissimilaires), qui seront maillées lors des dernières passes. Grâce à cette méthode, des objets de n'importe quelle topologie peuvent être reconstruits de façon naturelle, particulièrement dans les zones complexes où les contours sont très différents.

2.2.5 Nuages non-organisés

Il est évident que les méthodes de reconstruction applicables à un nuage de points non-organisés, peuvent aussi être appliquées pour obtenir des surfaces à partir des contours. Dans ce cas la structuration des données n'est pas exploitée. L'étude de ces méthodes n'entre pas dans le cadre de ce travail. Un état de l'art peut être trouvé dans les références [BV91a] ou [EH96].

2.3 Limitations des méthodes actuelles

L'étude des méthodes de reconstruction existantes, nous a montré que des travaux de recherche complémentaires sont nécessaires afin de résoudre les problèmes liés à la construction des surfaces de topologie simple, mais aussi des embranchements. Dans ce cadre, certains critères sont importants pour définir la qualité du modèle de surface reconstruit :

- la compacité du modèle, le taux de compression des données ;
- la continuité, C^0 ou plus, avec éventuellement des ordres plus élevés ;
- l'erreur commise, la distance entre le modèle et les contours ;
- le temps de calcul ;
- le niveau d'automaticité et la possibilité d'agir rétroactivement en cas d'erreur dans le modèle final.

Il est bien sûr très difficile de satisfaire tous ces critères en même temps, mais trouver une « bonne » surface peut correspondre à trouver un « bon » compromis entre ces contraintes, compromis qui dépend souvent de l'usage a posteriori du modèle. Les plus importants nous ont semblé être la compacité et la continuité, puisque la plupart du temps les modèles obtenus sont réutilisés dans d'autres applications, par exemple pour effectuer des mesures ou tout simplement pour le visualiser. Dans le contexte de notre travail (application médicale), les formes anatomiques étant des objets lisses, il est essentiel que le modèle présente un aspect agréable à l'œil sans avoir recours à une précision excessive. Même si le modèle géométrique est souvent précalculé, la détermination d'une surface optimale peut quelquefois être trop longue. Il est sans doute préférable de construire rapidement une surface proche de la surface optimale (au sens de l'erreur absolue), tout en contrôlant l'erreur commise. Cette approximation est généralement suffisante, elle se révèle d'ailleurs parfois meilleure que la solution optimale qui peut présenter des oscillations. Le dernier critère important est l'interactivité que permet la méthode. Lorsque le modèle est erroné, l'utilisateur doit pouvoir agir pour le corriger. Cela arrive fréquemment lorsque les contours sont très perturbés.

2.3.1 Limitations lors de la construction des surfaces

Les algorithmes de maillage sont les plus utilisés pour la reconstruction d'un objet défini par des contours. Cela se justifie par le fait que le modèle est très simple – un ensemble de triangles – et qu'il est très facile à visualiser. Un autre avantage est qu'il peut être transformé en un système masses-ressorts pour l'animation, ou utilisé dans une méthode de résolution par éléments finis. Par contre, pour obtenir une précision fine, un grand nombre de triangles est requis. Et quelle que soit la précision, la représentation géométrique n'est jamais lisse, ce qui peut devenir un problème pour certaines applications, l'inspection par exemple.

Il est bien sûr possible de lisser ce maillage, diverses méthodes existent pour cela. Mais le meilleur moyen d'obtenir une surface lisse est de la construire directement à partir des contours. Cependant, à ce niveau là, se pose le problème de la détermination d'un paramétrage. En effet, les méthodes de construction de surfaces paramétrées à partir de points digitalisés sont sensibles aux valeurs de paramètres affectées à ces points.

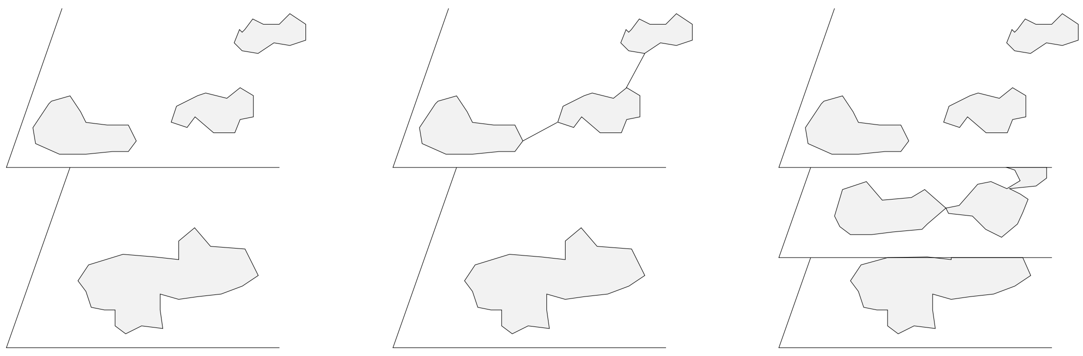
Un moyen de contourner ce problème est d'utiliser la méthode de « skinning » [HLLK94, PK96]. Elle consiste à tendre une peau sur une série de courbes porteuses. Le passage des contours polygonaux initiaux vers les courbes paramétriques puis vers la surface biparamétrique se fait naturellement. Le problème de la paramétrisation (mais aussi celui de la reconstruction) en 3D est réduit à un problème en 2D nettement moins complexe. La reconstruction par étapes permet à l'utilisateur de mieux contrôler la reconstruction. Deux tolérances différentes peuvent être définies, une pour chaque section. De plus, la réduction de dimension autorise des corrections beaucoup plus intuitives et surtout beaucoup plus simples à effectuer. Il est plus facile de gérer des interactions en 2D que dans le cas 3D.

Pour ces raisons, nous avons choisi la méthode de « skinning ». Notre méthode permet de générer une surface tout en respectant les contraintes de précision, et en réduisant le nombre de points de contrôle. Nous nous sommes démarqués des méthodes précédemment publiées dans ce domaine. Plutôt que de définir un vecteur nodal exactement compatible puis de réduire le nombre de nœuds, nous faisons une approximation du vecteur nodal commun et nous augmentons le nombre de nœuds si l'erreur qui en découle est trop importante. La surface finale peut comporter un nombre de points de contrôle très légèrement supérieur au nombre minimum requis pour satisfaire exactement le critère d'erreur, mais elle est obtenue beaucoup plus rapidement car elle évite la phase de mise en compatibilité exacte qui représente jusqu'à 90% du temps de calcul dans les autres méthodes d'approximation par « skinning » [PT95, PK96].

2.3.2 Limitations lors de l'embranchement

La plupart des méthodes de reconstruction n'est pas capable de prendre en compte les embranchements, car elles requièrent en entrée une série de contours, où chaque contour n'est connecté qu'à un seul autre dans la section suivante. Une technique couramment employée est de transformer les contours pour obtenir plusieurs sous-ensembles de contours formant une topologie « simple ». Cela peut être réalisé soit en regroupant les contours multiples en un seul contour composite [EPO91], soit en créant de nouveau contours intermédiaires [OPC96]. Mais ces contours modifiés sont souvent de piètres approximations des contours initiaux (Fig. 2.6).

De plus, il est difficile d'obtenir une continuité élevée (G^2, C^2 ou plus) entre les différentes surfaces représentant les branches. Certaines méthodes réalisent une continuité G^1 voire C^1 en collant des carreaux paramétriques triangulaires ou quadrangulaires [ESS91, PK95]. Mais cela se restreint à des degrés faibles, et cette méthode n'est plus utilisable dès que le degré d



(a) Contours initiaux

(b) Exemple de contour composite

(c) Exemple de contour intermédiaire

FIG. 2.6 – *Passage à un seul contour dans les zones d'embranchement*

carreau est augmenté (4 ou plus) car l'allure des carreaux serait alors trop difficile à contrôler et des oscillations voire des boucles pourraient apparaître.

Nous avons donc proposé une méthode originale de remplissage des zones d'embranchement basée sur l'utilisation des carreaux de COONS. Les propriétés de ce type de surface sont tout à fait intéressantes pour résoudre ce genre de problème. Ils consistent en une interpolation entre quatre courbes frontières. En choisissant deux des contours de part et d'autre de l'embranchement pour former des courbes frontières, nous réalisons une interpolation d'une partie de l'embranchement. Nous comblons ensuite les espaces libres entre les carreaux par un nouveau carreau de COONS. D'autre part, il est possible de définir des tangentes sur les bords du carreau. En les choisissant égales à celles des surfaces construites sur chaque branche, nous obtenons facilement une continuité G^1 dans la zone de raccord, et cela sans augmentation du degré. Peu de carreaux sont nécessaires, puisque pour un embranchement simple à deux branches, seulement trois carreaux permettent de remplir l'embranchement de façon lisse, et six carreaux pour trois branches.

3 Reconstruction à l'aide d'un modèle paramétrique

Au vu de l'état de l'art sur la reconstruction de surfaces à partir d'un ensemble de sections, il nous est apparu que les méthodes existantes n'étaient pas satisfaisantes. Pour résoudre un certain nombre de problèmes évoqués dans le chapitre précédent, nous avons donc développé une nouvelle approche. Nous avons choisi le modèle paramétrique, et plus précisément le modèle B-spline, qui est très largement répandu dans la domaine de la modélisation et de la reconstruction de courbes et surfaces gauches. Les surfaces obtenues possèdent des qualités de compacité et de continuité tout à fait adaptées à nos besoins. Le paramétrage, qui est une contrainte importante des modèles paramétriques, est résolu par l'utilisation d'une méthode dite de « skinning ». Dans ce chapitre, nous détaillons notre approche pour reconstruire une surface sur un ensemble de contours ouverts ou fermés. Après une brève description de la méthode de « skinning », nous présentons les différentes étapes de la construction de cette surface.

3.1 Étude du « skinning »

Le « skinning » est un nouveau terme pour « lofting ». Cette opération est utilisée depuis longtemps en construction navale, automobile ou aéronautique, et elle reste très adaptée pour de nombreuses applications. Le « skinning » consiste à tendre une peau (ou « skin » en anglais) sur un ensemble ordonné de courbes, qui sont aussi appelées courbes de section. Si en pratique ces courbes sont la plupart du temps des courbes planaires, il n'y a aucune restriction à ce qu'elles soient tridimensionnelles. Le résultat obtenu est une surface bi-paramétrique, la direction u est celle des sections, alors que la direction v permet de passer d'une section à l'autre, elle est aussi appelée direction longitudinale. Dans la plupart des cas, les sections sont interpolées, elles définissent alors des courbes iso-paramétriques de la surface. Cependant, l'approximation est aussi utilisée, notamment pour réduire le nombre de points de contrôle en v lorsque les sections sont nombreuses.

Le « skinning » est utilisé pour reconstruire une surface B-spline à partir d'un ensemble de points disposés selon des sections. Ce processus se décompose en plusieurs opérations consécutives :

1. Chaque section est reconstruite par une courbe B-spline
2. Ces courbes de section sont rendues compatibles
3. La surface B-spline finale est générée en tendant une peau sur ces courbes compatibles

La première étape a été largement étudiée, c'est maintenant une opération classique en C.A.O. . Elle est décrite dans l'annexe A. Les contours sont représentés par une suite de points ordonnés, qui peut être ouverte ou fermée. Même si, dans la plupart des applications, les contours sont obtenus par l'intersection de plans parallèles avec l'objet réel, nous n'imposons aucune restriction ni sur le parallélisme, ni sur la planarité des contours. Le passage d'une courbe ouverte vers une courbe périodique, nécessaire à la reconstruction d'objets fermés, est aussi décrite dans cette annexe.

La deuxième étape est la plus délicate. Elle est rendue nécessaire par la définition des surfaces B-splines par un produit tensoriel, qui impose un degré et un vecteur nodal identiques pour toutes les courbes de section. Si l'égalité des degrés peut être résolue par une simple élévation de degré, l'égalité des vecteurs nodaux est loin d'être triviale. Elle peut être résolue soit par moyennage [PK96], soit par fusion des différents vecteurs en utilisant la méthode d'insertion de nœuds de Boehm [Boe80]. Cependant, le nombre de points de contrôle devient vite considérable et il est nécessaire de le réduire pour obtenir des modèles utilisables, mais à un coût élevé [PT95, PK96].

Une fois que les courbes de section sont compatibles, la troisième étape consiste à tendre une peau sur ces courbes. Pour cela, les points de contrôle sont reconstruits par une courbe dans la direction longitudinale. Le regroupement des résultats de chaque colonne de points définit le réseau de contrôle de la surface finale. L'allure de la surface peut être contrôlée en introduisant une courbe longitudinale, appelée « spine curve », qui donne l'orientation des tangentes dans la direction v [Woo87]. Une méthode pour représenter de façon procédurale les surfaces de « skinning » a aussi été proposée [FB89].

3.2 Reconstruction de surfaces

Dans la suite de ce chapitre, nous décrivons l'aspect technique de la méthode de « skinning », ainsi que la façon dont nous utilisons cette méthode pour la reconstruction de surfaces. Comme nous l'avons vu au paragraphe précédent, la mise en compatibilité des courbes est le problème majeur à résoudre. Nous nous attarderons donc sur ce point.

3.2.1 Description de la méthode de « skinning »

Nous supposons dans ce chapitre que nous disposons d'un ensemble de s courbes $\{\vec{C}_k(u)$, pour $k = 0, \ldots, s-1\}$. Le « skinning » permet de lisser ou d'interpoler ces courbes pour former une surface B-spline. Une restriction à cette méthode est la nécessité d'avoir une compatibilité entre les courbes qui doivent remplir trois conditions :

- être de degré égal p ;
- posséder le même nombre de points de contrôle n ;
- être définies sur le même vecteur nodal U.

Ces conditions sont imposées par le fait que la surface B-spline finale de degré $p \times q$ est définie par un produit tensoriel (Éq. 3.1) :

$$S(u,v) = \sum_{j=0}^{m-1} \sum_{i=0}^{n-1} \vec{P}_{i,j} N_i^p(u) N_j^q(v) \text{ pour } u \in [0,1] \text{ et } v \in [0,1] \tag{3.1}$$

où $\vec{P}_{i,j}$ représente le sommet d'indices i et j du réseau caractéristique des points de contrôle et $N_i^p(u)$, $N_j^q(v)$ représentent les fonctions de base B-spline dans les deux directions. La direction u est celle des courbes à reconstruire et la direction v est celle du lissage ou de l'interpolation, que nous appelons aussi direction longitudinale.

Nous devons déterminer ce réseau pour reconstruire la surface. Si nous réécrivons l'équation 3.1 pour chaque courbe intermédiaire $\vec{C}_k(u)$, nous obtenons :

$$\begin{aligned} \vec{C}_k(u) &= \vec{S}(u, v_k) \\ &= \sum_{i=0}^{n-1} \left\{ \sum_{j=0}^{m-1} \vec{P}_{i,j} N_j^q(v_k) \right\} N_i^p(u) \\ &= \sum_{i=0}^{n-1} \left(\vec{P'}_k\right)_i N_i^p(u) \text{ pour } k = 0, \ldots, N-1 \end{aligned} \tag{3.2}$$

où les $\left(\vec{P'}_k\right)_i$ pour $i = 0, \ldots, n-1$ sont connus et forment le polygone caractéristique de la courbe $\vec{C}_k(u)$. D'après l'équation 3.2, nous déduisons que :

$$\sum_{j=0}^{m-1} \vec{P}_{i,j} N_j^q(v_k) = \left(\vec{P'}_k\right)_i \quad \text{pour} \quad \begin{aligned} k &= 0, \ldots, s-1 \\ i &= 0, \ldots, n-1 \end{aligned} \tag{3.3}$$

Pour déterminer l'ensemble des sommets $\vec{P}_{i,j}$ du réseau caractéristique de la surface, il suffit alors de reconstruire une courbe sur chaque colonne $i = 0, \ldots, n-1$ de points de contrôle $\left(\vec{P'}_k\right)_i$ des courbes intermédiaires. Ce problème peut être résolu soit par approximation (cf. annexe A.1), soit par interpolation (cf. annexe A.2). Lorsque le nombre de sections est réduit, une simple interpolation suffit. Par contre, dès que le nombre de sections augmente, il devient nécessaire d'utiliser l'approximation pour éviter de générer un nombre trop important de points de contrôle dans la direction longitudinale. En regroupant le résultat de chaque colonne, nous obtenons le réseau caractéristique complet qui définit la surface en produit tensoriel. Pour faire ce regroupement, et pour les mêmes raisons que dans la direction u, il faut s'assurer que les courbes reconstruites dans la direction v sont compatibles, c.-à-d. qu'elles satisfont les trois propriétés précédemment énoncées d'égalité de degré, du nombre de points de contrôle et de vecteur nodal.

3.2.2 Mise en compatibilité des courbes intermédiaires

Le « skinning » est une opération très générale puisqu'elle permet de reconstruire une surface sur des courbes de degrés quelconques et de vecteurs nodaux différents. Par contre, le revers d'une telle flexibilité est le coût élevé de la phase de mise en compatibilité des courbes.

Pour une courbe unique, le placement des nœuds est une tâche relativement facile. Par contre, pour un ensemble de courbes, cela devient plus ardu puisque chaque nœud agit sur toutes les courbes à la fois. Il serait possible de répartir les nœuds de façon uniforme, mais cela ne fournit pas de bons résultats. En effet, si pour une courbe, les valeurs de paramètres s'accumulent dans un intervalle de nodal et laissent vides certains autres intervalles, il peut se produire des recouvrements ou des oscillations. Dans les cas extrêmes, les équations de contraintes pour déterminer la courbe peuvent même définir un système singulier.

Idéalement, le vecteur nodal doit refléter parfaitement la distribution des paramètres le long de toutes les courbes, tout en évitant la formation d'intervalles vides. De bonnes valeurs pour les nœuds sont obtenues par un processus de moyennage. Ainsi, pour chaque courbe, nous calculons le vecteur nodal en fonction des paramètres (équation en annexe A.7 ou A.8 selon les cas, respectivement approximation ou interpolation). Nous prenons la moyenne de ces valeurs pour toutes les courbes :

$$u_i = \frac{1}{s} \sum_{j=0}^{s-1} u_i^j \text{ pour } i = 0, \ldots, n+p \tag{3.4}$$

où s est le nombre de contours et $(n+p+1)$ le nombre de nœuds. Le nombre n de points de contrôle est le même pour chaque courbe. Cependant, l'équation 3.4 n'évite pas la formation d'intervalles vides, même lorsque les points sont disposés de façon relativement équivalente sur tous les contours. De tels vides peuvent être comblés en échantillonnant de nouveaux points sur le contour, dans ces intervalles.

Il reste enfin à déterminer le nombre n de points de contrôle. Pour obtenir le nombre minimal de points nécessaire pour reconstruire l'ensemble des courbes avec une précision donnée, il faut,en théorie, résoudre un système non-linéaire d'optimisation dont les inconnues sont le nombre de points de contrôle, les points de contrôle, les valeurs de paramètres ainsi que les vecteurs nodaux et ceci pour chaque courbe. Étant donné le nombre important d'inconnues, cette résolution est beaucoup trop coûteuse. Il est cependant possible de réduire facilement le nombre de points de contrôle. L'algorithme 3.1 applique le principe de la dichotomie [PK96]. Même si le nombre de points de contrôle obtenu n'est pas optimal, il reste très proche pour un coût nettement inférieur à celui de l'optimisation des solutions du système non-linéaire. Cependant, cette recherche reste une opération onéreuse, qui peut représenter jusqu'à 90% du temps total pour tout le processus de « skinning » [PK96].

Dans la plupart des cas, un simple moyennage des vecteurs de nœuds calculés pour chaque courbe est largement suffisant, dans la mesure où les contours sont relativement identiques, ce qui implique que les points soient disposés de façon similaire le long des contours. Cela est d'autant plus vrai quand les sections sont, en plus, régulièrement espacées, mais surtout si elles sont parallèles. Ainsi, les points de contrôle de chaque courbe se trouvent tous dans un même plan, celui de la section. Lors de la mise en commun des vecteurs nodaux dans

Algorithme 3.1 Réduction du nombre de points de contrôle

1 initialiser l'intervalle pour le nombre de points de contrôle

$\triangleright$ parmi tous les contours déterminer celui qui contient le plus grand nombre de points n_{max}

$\triangleright$ n_{min} est le nombre de points nécessaires pour reconstruire ce contour selon la précision donnée

$\triangleright$ $n_{inf} \leftarrow n_{min} - 1$, $n_{sup} \leftarrow n_{max}$, $n \leftarrow n_{min}$

2 déterminer un vecteur de nœuds commun en fonction n

3 reconstruire le contour avec ce vecteur de nœuds

si la courbe satisfait la précision requise

alors passer au contour suivant

sinon aller à l'étape 5

si toutes les courbes satisfont la précision

alors aller à l'étape 4

4 succès

si $n - n_{inf} \leq 1$

alors FIN

sinon $n_{sup} \leftarrow n$, $n \leftarrow (n_{inf} + n_{sup})/2$

aller à l'étape 3

5 échec

si $n_{sup} - n \leq 1$

alors FIN, il n'est pas possible de faire mieux

sinon $n_{inf} \leftarrow n$, $n \leftarrow (n_{inf} + n_{sup})/2$

aller à l'étape 3

la direction v, les points de contrôle sont répartis de façon similaire pour chaque colonne. Il est possible alors de se limiter à une approximation du vecteur de nœud commun, de type *moyenne*. Ce n'est pas la peine de chercher à calculer un vecteur nodal exact, en employant des techniques plus lourdes, telles que l'insertion de nœud (voir paragraphe suivant 3.2.3).

3.2.3 Problèmes liés à la mise en compatibilité

Lorsque les courbes sont « très différentes », la méthode précédente consistant à faire la moyenne de nœuds, n'est plus adaptée. Cela peut être dû à plusieurs facteurs dont quelques-uns sont énumérés ci-après :

- le nombre de points est très différent d'un contour à l'autre ;
- la forme des contours varie beaucoup ;
- les points ne sont pas répartis uniformément.

Il a été proposé une formulation pour évaluer la complexité de forme d'un contour, qui reprend plus ou moins les critères énoncés ci-dessus [PK96] :

$$SC = \alpha \frac{r_c}{r} + (1-\alpha)\left\{1 - \frac{A}{2A_c} - \frac{D}{2D_c}\right\}$$

où $0 \leq \alpha \leq 1$. r et r_c sont le nombre de points du contour et de son enveloppe convexe, A et A_c sont les aires respectives, D et D_c sont les périmètres respectifs. Le contour est donc comparé à son enveloppe convexe selon différents critères et avec différents coefficients. Le résultat SC est une valeur entre 0 et 1, qui est d'autant plus grande que le contour est complexe. Cet indice permet de déterminer la complexité d'un contour, mais elle peut être utilisée pour comparer les contours entre-eux car deux contours qui ont des complexités différentes ont peu de chance d'être similaires en forme. Cela dit, le contraire n'est pas vrai, deux contours très différents peuvent avoir des complexités équivalentes. L'étude de la variation des indices SC de chaque contour permet de définir une complexité de forme globale pour l'objet à reconstruire.

Lorsque les contours d'un objet ne satisfont pas aux critères de complexité, les courbes construites sur ces contours risquent de ne pas pouvoir être mises en compatibilité en utilisant un simple moyennage. Il faut alors calculer le vecteur nodal commun de façon exacte. La fusion entre deux vecteurs de nœuds U_1 et U_2 se fait en deux étapes :

1. Fusion de U_1 et U_2 pour former U
   u_i appartient à U si il appartient soit à U_1 soit à U_2
   La multiplicité maximale de u_i dans U_1 ou U_2 est reportée dans U
2. En utilisant U, appliquer aux deux courbes le raffinement de nœuds qui consiste à insérer plusieurs nœuds en même temps

L'algorithme du raffinement de nœuds est dérivé de l'insertion de nœuds de Boehm :

Algorithme 3.2 Raffinement de nœuds

1 Déterminer les indices a et b tels que $U_a \leq u_i < u_b$, si u_i est le nœud à insérer, pour tout $i = 0, \ldots, w-1$

2 Les points de contrôle $P_0, \ldots, P_{a-p}$ et $P_{b-1}, \ldots, P_{n-1}$ ne changent pas, seuls les $(w + p + b - a - 2)$ points intermédiaires sont modifiés

3 Calculer successivement les nouveaux points de contrôle de P_{a-p+1} à P_{b-2}, [PT96]

La fusion doit se faire sur tous les vecteurs nodaux. À chaque fusion, le nombre de points augmente. Par conséquent, pour un grand nombre de courbes et si les nœuds sont différents d'un vecteur à l'autre, ce qui est le plus courant, le nombre de nœuds devient vite exorbitant. Il n'est pas rare que pour interpoler une centaine de contours, le nombre de points de contrôle atteigne quelques milliers.

Une méthode a été proposée pour réduire le nombre de points de contrôle générés lors de la phase de mise en compatibilité précise. Elle consiste simplement à calculer le vecteur nodal commun puis à supprimer autant de nœuds que possible tout en respectant la tolérance [PT96]. L'algorithme est le suivant :

Algorithme 3.3 Élimination de nœuds

1 Calculer le vecteur nodal commun

2 Calculer l'erreur commise lorsque l'on supprime un nœud, ceci pour chaque nœud et pour chaque courbe

3 **répéter**

  ▷ pour le nœud qui a la plus petite erreur

  ▷ calculer les erreurs commises sur chaque courbe si l'on supprime ce nœud

  si la nouvelle erreur est acceptable

    alors supprimer ce nœud et mettre à jour les erreurs

    sinon passer au prochain nœud

Cet algorithme est essentiellement dérivé de la suppression d'un nœud sur une courbe, mais à la différence que le nœud est supprimé sur toutes les courbes en même temps. Il peut aussi être adapté pour réduire le nombre de points de contrôle dans la direction longitudinale mais dans ce cas le calcul de l'erreur doit se faire avec la surface approximée, ce qui est très coûteux. Le gain n'est vraiment intéressant que pour un grand nombre de contours. Cet algorithme peut ainsi être utilisé dans les deux directions simultanément, avec la possibilité de définir deux tolérances distinctes en u et en v.

Les résultats sont très probants puisque pour de faibles précisions, plus de 90% des points de contrôle sont supprimés. Pour des précisions très élevées, l'algorithme proposé par Piegl et Tiller permet de réduire encore un peu le nombre de points, de l'ordre de 1 à 5%, ce qui montre combien la mise en compatibilité précise est une opération redondante, puisque de nombreux nœuds du vecteur nodal commun sont inutiles. Le temps de calcul requis pour faire la mise en compatibilité est très importante, mais au final, la surface obtenue est plus compacte, ce qui est un avantage lorsque l'on veut travailler sur le modèle dans un système de C.A.O. .

3.3 Conclusion

Dans ce chapitre, nous avons présenté une méthode de reconstruction de contours, basée sur la technique de « skinning ». L'apport principal de notre approche, par rapport aux méthodes existantes qui utilisent le « skinning », est qu'elle génère un nombre restreint de points de contrôle en peu de temps. C'est une simplification des méthodes existantes, qui est justifiée lorsque la surface finale ne nécessite pas une très grande précision. En effet, lors de la phase de mise en compatibilité des courbes, nous réduisons fortement le nombre de nœuds, et donc de points de contrôle, en calculant une approximation du vecteur nodal. Cette approximation se fait en gardant un contrôle de l'erreur, ce qui justifie la validité du modèle paramétrique obtenu.

Cependant, la représentation de la surface par un produit tensoriel pose quelque problèmes. Les formes à modéliser représentent des volumes. Nous devons donc apporter quelques modifications au modèle B-spline classique pour prendre en compte cette contrainte. Dans la direction u, nous utilisons les B-splines périodiques pour reconstruire des courbes fermées lorsque les contours les sont aussi. Mais, la surface reste ouverte dans la direction longitudinale v (en haut et en bas), et l'utilisation des splines périodiques dans cette direction est très difficile à réaliser techniquement. Nous créons donc des sections fictives aux deux extrémités, en imposant des contraintes de continuité. Cette solution est présentée au chapitre 4.

Le deuxième problème est la prise en compte des embranchements. Il n'est pas possible de créer une surface en produit tensoriel qui se sépare en plusieurs morceaux. Il faut donc recourir à plusieurs surfaces qui seront raccordées entre-elles pour conserver la continuité (G^1, G^2...) que les splines nous permettent d'obtenir. Nous avons développé une solution simple, que nous présentons au chapitre 5.

4 Évolution vers des surfaces fermées

Ce chapitre traite des problèmes liés à la structuration des données lors de la reconstruction de surfaces B-splines fermées à partir d'un ensemble de contours. Dans la première section, nous décrivons les traitements à appliquer aux données pour obtenir une fermeture correcte dans la direction u, celle des contours. La seconde section concerne l'approche utilisée pour réaliser une fermeture aux extrémités, c.-à-d. dans l'autre direction.

4.1 Alignement des origines des contours

Un des problèmes qui apparaît lorsque nous voulons faire un « skinning » sur des courbes fermées est le décalage des origines entre les courbes successives. Pour obtenir une surface finale qui soit lisse et qui ne présente pas de torsion, il est nécessaire d'aligner les origines des courbes initiales. Cela signifie qu'un point de départ adéquat doit être trouvé sur chaque contour initial. Nous appelons ligne de base la ligne qui relie l'ensemble de ces points.

Intuitivement, cette ligne doit être la plus courte et la plus rectiligne possible. Elle doit aussi tenir compte de la forme des contours. De manière idéale, il faudrait chercher parmi tous les chemins possibles allant d'un contour à l'autre, celui qui correspond le mieux aux critères. Cependant, cette recherche peut s'avérer fastidieuse dans la cas où les données comportent un grand nombre de points et de sections.

Nous proposons une méthode qui permet de trouver une ligne de base acceptable, avec un coût réduit en ne testant qu'un nombre restreint de chemins. Nous procédons de la manière suivante : une ligne de base est construite à partir de chaque point du premier contour. Chaque ligne est ensuite prolongée contour par contour. Si P_i est le $i^{\text{ème}}$ point de la ligne de base, il appartient au $i^{\text{ème}}$ contour, alors le $(i+1)^{\text{ème}}$ point P_{i+1} appartient au contour $i+1$ et est tel que la distance avec P_i est minimale. Parmi toutes ces lignes de base, nous sélectionnons celle qui est la plus courte. L'inconvénient de cette méthode est que le résultat, bien que correct localement, ne reflète pas la forme globale des contours. Pour remédier à cela, nous considérons plusieurs lignes à la fois. Ces lignes doivent être régulièrement positionnées, afin de ne privilégier aucune partie du contour, dont la forme n'est pas connue à priori. La figure 4.1 montre la répartition des lignes le long du contour. Ainsi, pour chaque point du premier

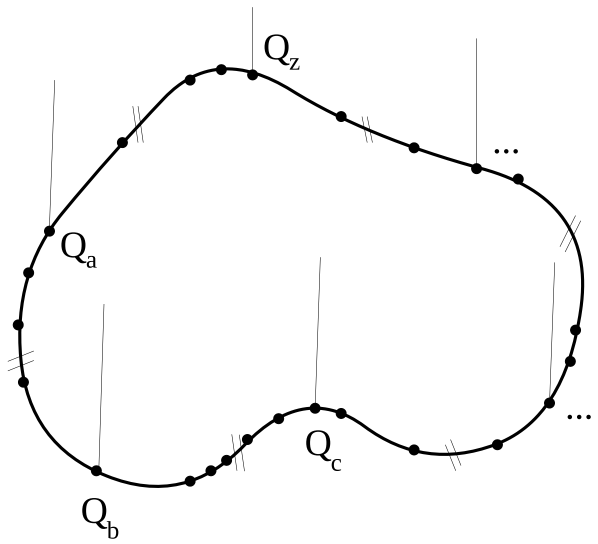


FIG. 4.1 – *Répartition régulière des lignes*

contour, nous calculons la somme des longueurs de la ligne de base issue de ce point et de quelques autres lignes issues de points régulièrement espacés. Cela signifie que la distance le long du contour entre chacun de ces points est régulière. Si d est la longueur totale du contour, et si Q_a est le point de départ de la ligne de base, alors le point de départ de la ligne suivante sera le premier point tel que $b > a$ et :

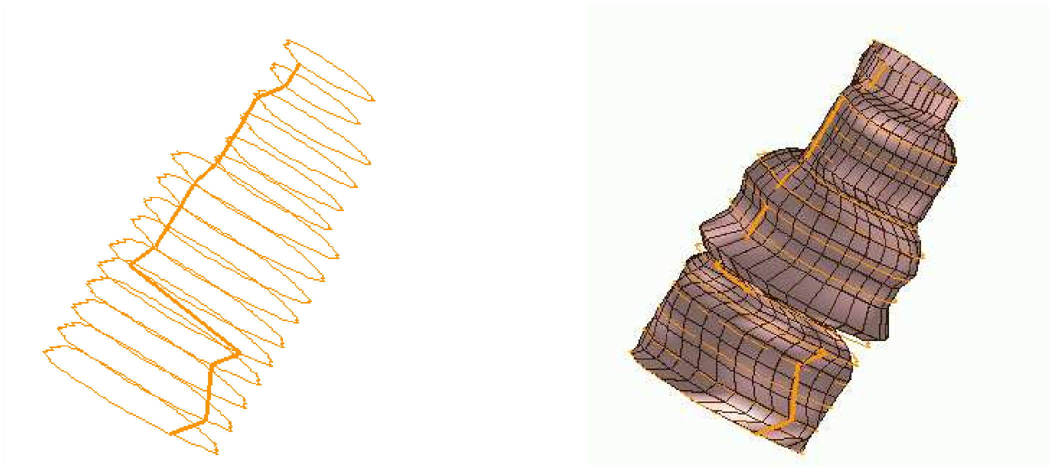
$$\sum_{i=a+1}^{b} \left|\vec{Q}_i - \vec{Q}_{i-1}\right| \geq \frac{d}{nb} \text{ pour } nb \geq 1 \tag{4.1}$$

où nb est le nombre de lignes considéré. Si $nb = 0$, une seule ligne est prise en compte, et cela n'est pas significatif. Si $nb = 1$, nous prenons en compte une zone du contour et la zone à l'opposé. Nous avons constaté expérimentalement que $nb = 4$ lignes fournissaient des résultats visuellement satisfaisants.

La ligne de base est alors considérée comme la meilleure si sa longueur et celle de ses lignes associées est la plus courte. Il ne reste plus qu'à re-numéroter les points de chaque contour.

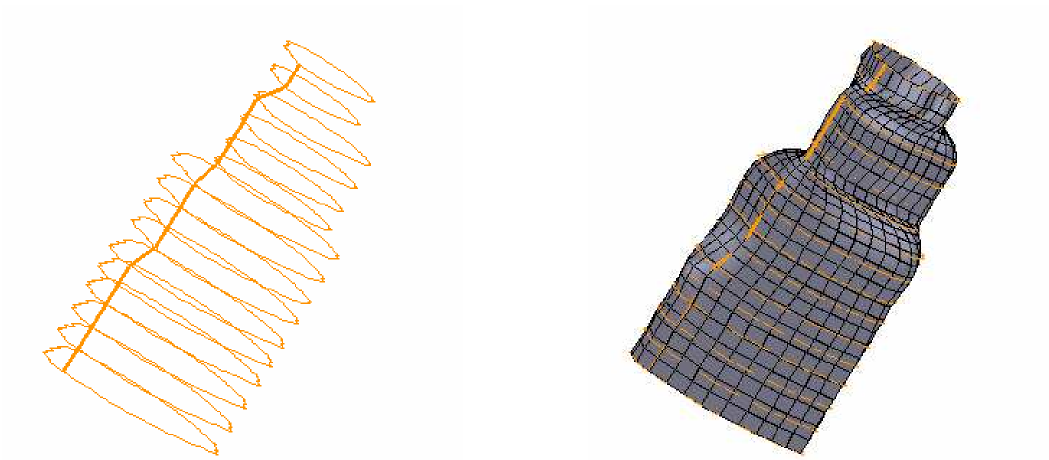
La figure 4.2 montre comment des points de départ disposés de façon quelconque peuvent conduire à des zones de torsion sur la surface. Par contre, lorsque les points sont alignés correctement, ce défaut disparaît (Fig. 4.3).

HAN et al. [HLLK94] ont proposé une méthode qui prend en compte l'orientation des contours. Les points de départ de chaque contour sont tous localisés dans la même direction par rapport au centroïde de ce contour. Cette méthode est bien adaptée pour traiter les décalages d'un contour à l'autre, mais elle ne tient pas compte de la forme des contours. Il serait intéressant de combiner cette approche avec la nôtre, pour obtenir des résultats plus satisfaisants.



(a) Contours initiaux (b) Surface B-spline

FIG. 4.2 – *Reconstruction d'une surface à partir de contours dont les origines ne sont pas alignées*



(a) Contours initiaux (b) Surface B-spline

FIG. 4.3 – *Reconstruction d'une surface à partir de contours dont les origines sont correctement alignées*

4.2 Fermeture

Dans le cas où les contours sont fermés, ils sont reconstruits à l'aide de courbes B-splines périodiques fermées (cf paragraphe A.3). La surface finale est alors fermée dans la direction u, mais elle est ouverte dans l'autre direction. Cela peut ne pas être acceptable si l'objet à reconstruire décrit un volume. Il est alors nécessaire de fermer la surface en haut et en bas.

La méthode la plus simple est de couvrir les extrémités par un plan. Il est aussi possible d'effectuer une triangulation du dernier contour. Le principal inconvénient de ces méthodes est qu'elles génèrent des discontinuités au niveau des extrémités. Nous avons donc envisagé d'autres moyens de fermeture. Pour chaque méthode, nous décrivons le cas d'une fermeture au sommet, la fermeture en bas de l'objet s'obtenant facilement de manière analogue. Nous pouvons bien évidemment réaliser la fermeture en bas et en haut, ce qui conduit à un objet totalement fermé.

4.2.1 Extension vers un point

L'idée de la méthode est d'étendre la surface de manière à ce que toutes les iso-paramétriques en v convergent vers un même point situé au sommet de l'objet. Cela permet de modéliser tout l'objet à l'aide d'une seule surface B-spline. Nous évitons ainsi d'avoir recours à des carreaux de raccord pour joindre les différentes surfaces en préservant la continuité.

Le choix du point de convergence est crucial, puisque de sa position dépend la forme de l'objet au sommet. Il doit se trouver dans le prolongement des courbes longitudinales pour que la surface finale paraisse réaliste. Pour une première estimation, nous considérons que cette position ne dépend que de la position relative des derniers contours. Si ceux-ci sont décalés, le point de convergence sera décalé. Soit $\vec{G}_{m-1}$ et $\vec{G}_{m-2}$ les centres de gravité respectifs du dernier et de l'avant dernier contour, si m est le nombre de contours. Nous choisissons le point de fermeture sur la droite reliant ces deux points. Nous pouvons d'ores et déjà estimer que ce point se trouve à une distance de $\vec{G}_{m-1}$ inférieure à celle existant entre les deux centres de gravité, car sinon il est probable qu'un contour supplémentaire aurait été saisi. Si le point est éloigné du dernier contour, nous obtiendrons une surface pointue, par contre s'il est proche, la surface risque de paraître écrasée. Une bonne estimation de la position de $\vec{F}$ est donnée par la formule suivante (voir aussi la figure 4.4) :

$$\overrightarrow{FG_{m-1}} = \alpha \cdot \frac{S_{m-1}}{S_{m-2}} \cdot \overrightarrow{G_{m-1}G_{m-2}}$$

où S_{m-1} et S_{m-2} sont les aires respectives du dernier et de l'avant-dernier contour. α est un coefficient multiplicateur qui permet de moduler suivant le type de fermeture que l'on désire obtenir. Le ratio des aires des contours permet de ne privilégier aucune direction, contrairement à ce qui aurait pu se passer si l'on avait tenu compte uniquement de la largeur des contours. Ainsi, avec cette formule, plus la taille du dernier contour est petite par rapport au précédent et plus le point de convergence sera proche. Inversement, si les contours sont de même taille, cela signifie que nous sommes loin du sommet potentiel.

Une fois que le point de convergence a été choisi, il doit être intégré avec les autres courbes modélisant les contours initiaux. Pour permettre la phase de « skinning », ces courbes doivent

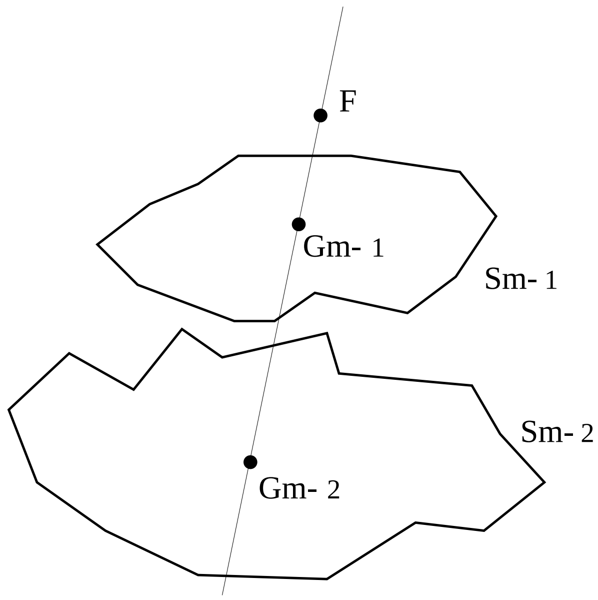


FIG. 4.4 – *Détermination du point de fermeture*

être compatibles (cf. paragraphe 3.2.2), y compris celles des extrémités. Nous créons donc une courbe B-spline de même degré et définie sur la même séquence nodale que les autres. Les points de contrôle sont une répétition du point de convergence autant de fois que nécessaire pour correspondre au vecteur de nœuds. Nous avons donc généré une courbe réduite à un point et compatible avec celles calculées à partir des données des contours. Nous pouvons alors procéder au « skinning » et obtenir ainsi une surface totalement fermée.

Cette méthode donne de bons résultats si le centre de gravité se trouve au centre du contour. Par contre, si les contours sont plus ou moins concaves, leur centre de gravité ne sera pas forcément inclus dans ce contour, et le point de convergence risque d'être erroné. La solution peut alors être de remplacer le centre de gravité par le centre du squelette du contour [Mal95].

4.2.2 Tangence aux extrémités

Pour assurer la continuité au sommet, nous introduisons des contraintes supplémentaires lors de la construction des courbes dans la direction longitudinale. La continuité des plans tangents G^1 est largement suffisante dans la plupart des cas, c'est d'ailleurs celle qui est le plus fréquemment utilisée dans les systèmes de C.A.O. lors des raccords entre les surfaces. Nous mettons donc l'accent sur la description de la méthode à l'ordre 1, mais nous indiquons comment obtenir des contraintes d'ordre plus élevé.

Le processus de reconstruction est essentiellement le même que celui utilisé pour les courbes sans fermeture, excepté le fait que chaque dérivée introduit un point et un nœud supplémentaire, et en conséquence une équation de plus dans le système linéaire. Les dérivées premières des bases B-spline peuvent se déduire des fonctions de base à l'aide de l'équation 4.2.

$$N'^{p}_{i}(u) = \frac{p}{u_{i+p} - u_i} N_i^{p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u) \tag{4.2}$$

Cette équation peut se généraliser à toutes les dérivées telles que $k \leq p$ si l'on désire une continuité d'ordre plus élevé que G^1 (Éq. 4.3).

$$\begin{aligned}
N^{(k)p}_{\ i}(u) &= \frac{p!}{(p-k)!} \sum_{j=0}^{k} -a_{k,j} N_{i+j}^{p-k}(u) \\
a_{0,0} &= 1 \\
a_{k,0} &= \frac{a_{k-1,0}}{u_{i+p-k+1} - u_i} \\
a_{k,j} &= \frac{a_{k-1,j} - a_{k-1,j-1}}{u_{i+p+j-k+1} - u_{i+j}} \text{ pour } j = 1, \ldots, k-1 \\
a_{k,k} &= \frac{-a_{k-1,k-1}}{u_{i+p+1} - u_{i+k}}
\end{aligned} \tag{4.3}$$

De l'équation 4.2, nous pouvons facilement déduire l'équation de la dérivée d'une courbe B-spline (Éq. 4.4) :

$$\vec{C}'(u) = \sum_{i=0}^{n-2} \frac{(\vec{P}_{i+1} - \vec{P}_i)}{u_{i+p+1} - u_{i+1}} N_i^{p-1}(u) \text{ pour } u \in [0,1] \tag{4.4}$$

qui est une courbe B-spline de degré $p-1$, définie sur la même séquence nodale que $\vec{C}(u)$ à laquelle on enlève les premier et dernier nœuds, nous obtenons $m-2$ nœuds (cas de l'interpolation) :

$$U' = \{\underbrace{0, \ldots, 0}_{p \text{ fois}}, u_{p+1}, \ldots, u_{m-p-3}, \underbrace{1, \ldots, 1}_{p \text{ fois}}\}$$

Nous voulons imposer les dérivées aux extrémités, ce qui conduit à l'ajout de deux équations dans le système à résoudre pour obtenir les $(n+2)$ points de contrôle, ce qui peut s'exprimer ainsi pour une interpolation :

$$\begin{aligned}
\vec{Q}_k &= \sum_{i=0}^{n+1} \vec{P}_i N_i^p(\bar{u}_k) \\
-\vec{P}_0 + \vec{P}_1 &= \frac{u_{p+1}}{p} \vec{C}'(0) = \vec{D}_0 \\
-\vec{P}_n + \vec{P}_{n+1} &= \frac{u_{m+p-2}}{p} \vec{C}'(1) = \vec{D}_1
\end{aligned} \tag{4.5}$$

Ces deux équations supplémentaires sont insérées dans le système en seconde et en avant-dernière positions, ce qui conduit aux systèmes linéaires suivants :

$$
\begin{array}{ccccccccc}
N_0^p(\bar{u}_0)\vec{P_0} & + & N_1^p(\bar{u}_0)\vec{P_1} & + & \ldots & + & N_n^p(\bar{u}_0)\vec{P_n} & + & N_{n+1}^p(\bar{u}_0)\vec{P}_{n+1} & = & \vec{Q_0} \\
-\vec{P_0} & + & \vec{P1} & & & & & & & = & \vec{D_0} \\
N_0^p(\bar{u}_1)\vec{P_0} & + & N_1^p(\bar{u}_1)\vec{P_1} & + & \ldots & + & N_n^p(\bar{u}_1)\vec{P_n} & + & N_{n+1}^p(\bar{u}_1)\vec{P}_{n+1} & = & \vec{Q_1} \\
\vdots & & & & & & & & & & \vdots \\
\vdots & & & & & & & & & & \vdots \\
N_0^p(\bar{u}_{m-2})\vec{P_0} & + & N_1^p(\bar{u}_{m-2})\vec{P_1} & + & \ldots & + & N_n^p(\bar{u}_{m-2})\vec{P_n} & + & N_{n+1}^p(\bar{u}_{m-2})\vec{P}_{n+1} & = & \vec{Q}_{m-2} \\
 & & & & & - & \vec{P_n} & + & \vec{P}_{n+1} & = & \vec{D_1} \\
N_0^p(\bar{u}_{m-1})\vec{P_0} & + & N_1^p(\bar{u}_{m-1})\vec{P_1} & + & \ldots & + & N_n^p(\bar{u}_{m-1})\vec{P_n} & + & N_{n+1}^p(\bar{u}_{m-1})\vec{P}_{n+1} & = & \vec{Q}_{m-1}
\end{array} \quad (4.6)
$$

où m est le nombre de contours, et donc de points à reconstruire (nous sommes dans la direction longitudinale), et $n = m+2$ est le nombre de points de contrôle puisque l'on a ajouté deux points pour prendre en compte les dérivées. Ce système linéaire à bande peut être résolu comme précédemment, par inversion de matrice A. L'approche est tout à fait identique pour un lissage. Dans ce cas, $n < m + 2$, et le système doit être résolu par approximation au sens des moindre carrés.

Le schéma 4.5 montre en projection comment nous fixons la valeur des dérivées au sommet pour une courbe. Le même type d'opération doit bien sûr être effectuée pour les dérivées en-dessous de l'objet.

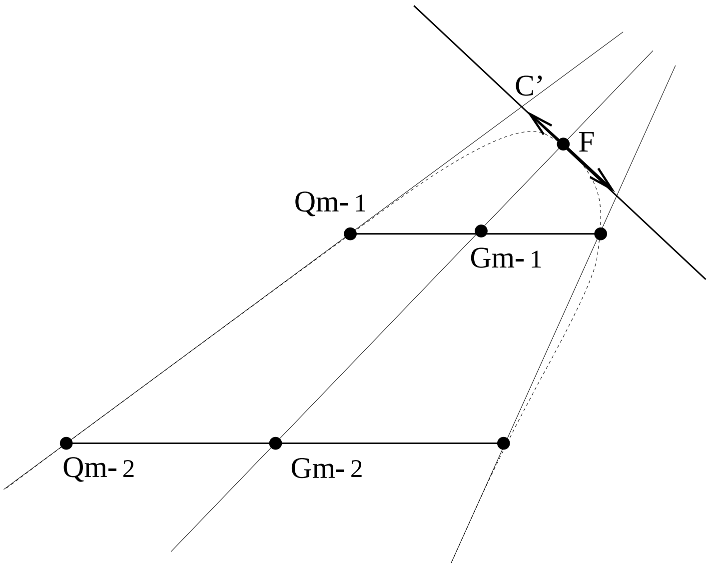
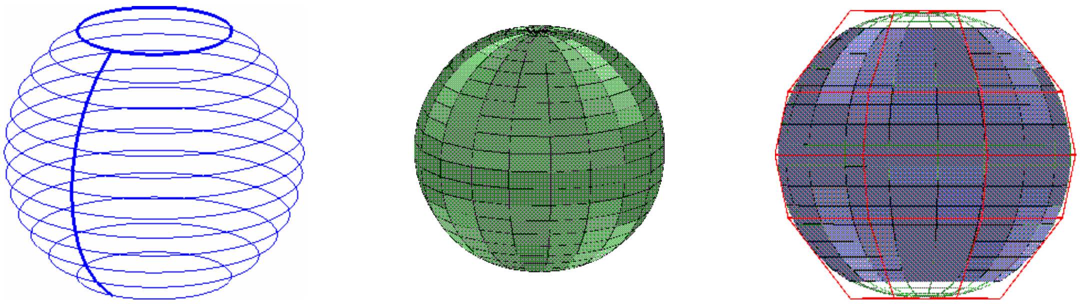


FIG. 4.5 – *Détermination des dérivées au sommet*

Si $\vec{Q}_{m-1}$ et $\vec{Q}_{m-2}$ sont respectivement le dernier et l'avant-dernier points à reconstruire pour la i$^{\text{ème}}$ courbe en v, et $\vec{F}$ est le point de convergence, alors la dérivée est définie par le vecteur $\vec{D_1} = \alpha\overrightarrow{FC'}$ tel que $\vec{C'}$ est l'intersection de la droite $(\vec{Q}_{m-2}\vec{F}_{m-1})$ avec le plan passant par $\vec{P}$ et orthogonal à la droite $(\vec{G}_{m-2}\vec{G}_{m-1})$ reliant les centres de gravité des deux derniers contours. α est un coefficient qui agit sur la norme de la dérivée, et permettant d'obtenir un sommet plus ou moins plat. Nous procédons de même pour tous les points de contrôle des courbes modélisant les deux derniers contours. Nous obtenons ainsi une continuité G^1 puisque le plan tangent ne varie pas. De plus, pour chaque courbe en v, la dérivée tient compte de l'allure des contours aux extrémités.

4.2.3 Résultats

Dans ce paragraphe, nous présentons quelques résultats significatifs de notre méthode de fermeture. La première figure illustre la reconstruction d'un ensemble de contours échantillonnés sur une sphère 4.6(a). La sous-figure (b) représente la surface d'approximation, bicubique fermée, comportant 7×7 points de contrôle. Le paramètre α défini à la fin du paragraphe 4.2.2 a une valeur égale à 1, ce qui permet d'obtenir un rayon de courbure constant dans la zone de fermeture. Sur la sous-figure (c), la surface fermée est superposée à la surface ouverte approximée avec les mêmes paramètres. Nous observons alors que la surface de fermeture se situe dans le prolongement de celle qui est ouverte. L'affichage des points de contrôle permet de vérifier que l'égalité des plans tangents est respectée au point de fermeture.



(a) Contours initiaux

(b) Surface B-spline fermée

(c) Superposition des surfaces ouverte et fermée

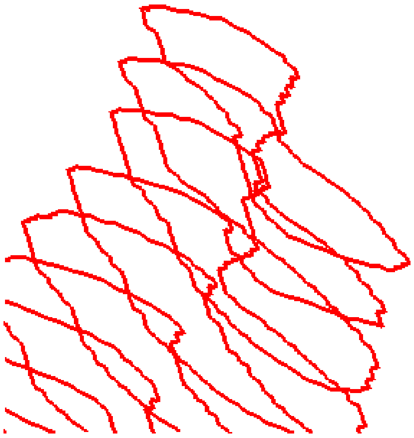
FIG. 4.6 – *Illustration de la méthode de fermeture aux extrémités*

La figure 4.7 illustre le résultat de fermeture dans le cas de contours moins simples. Les sous-figures de (a) à (b) représentent les derniers contours d'objets que l'on désire reconstruire à l'aide de surfaces entièrement fermées. Les sous-figures de (c) à (d) présentent les surfaces B-splines respectives.

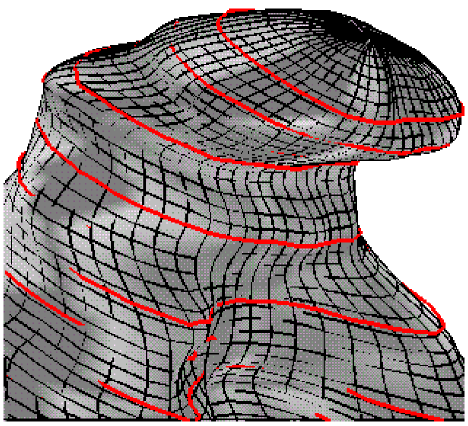
Sur la sous-figure (d), lorsque les derniers contours ne sont pas étoilés, la fermeture vers un seul point risque de provoquer des plis sur la surface reconstruite. Pour éviter ce problème, il peut être envisagé d'étendre la surface B-spline vers une courbe ouverte s'approchant du squelette du contour.

Dans les cas extrêmes, la solution serait de fermer la forme à l'aide d'une nouvelle surface B-spline. Cette surface aurait le rôle d'un chapeau. Pour assurer la continuité entre le corps de l'objet et la fermeture, il est nécessaire de créer une surface de raccord. Une méthode permettant de réaliser de tels carreaux à été développée au sein du LIGIM [Per92]. L'inconvénient

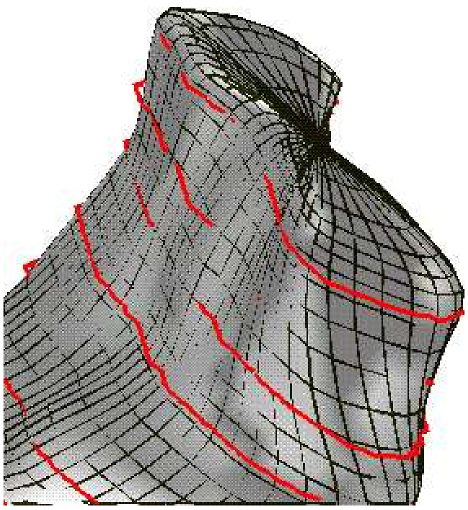
(a) Contours de l'objet 1



(b) Contours de l'objet 2



(c) Objet 1 : surface fermée



(d) Objet 2 : surface fermée

FIG. 4.7 – *Quelques exemples de fermeture*

de cette méthode est qu'elle rajoute deux surfaces B-splines à chaque extrémité si l'on ferme le haut et le bas, ce qui alourdit la gestion du modèle de l'objet.

5 Évolution vers l'embranchement

La reconstruction de surfaces comportant des embranchements est un problème difficile à résoudre dans le cas général. Nous présentons une méthode qui traite le cas des bifurcations (restriction à deux branches). La zone de raccordement est reconstruite à l'aide de carreaux de COONS, tout en respectant la continuité avec chaque branche. Nous distinguons plusieurs cas selon la configuration de l'embranchement, et chaque cas fait l'objet d'un traitement spécifique, en particulier au niveau de la disposition des carreaux. Le cas des bifurcations est une bonne base pour étendre la méthode aux embranchements de type $m \to n$, et nous proposons une approche pour prendre en compte les embranchements quelconques.

5.1 Méthodologie pour les bifurcations

Dans cette section, nous proposons une méthode générale pour effectuer un raccordement entre un tronc et deux branches ($1 \to 2$). Ce cas a été plus particulièrement développé car c'est celui qui se retrouve le plus couramment lorsque l'on veut reconstruire des objets de topologie complexe. Nous disposons donc de trois surfaces fermées qui ont été modélisées selon la méthode dite de « skinning » décrite au chapitre 3. Ces trois surfaces B-splines périodiques peuvent bien sûr être décrites sur des vecteurs de nœuds différents et comporter un nombre différent de points de contrôle. Le but, ici, est de remplir l'espace entre ces surfaces (Fig. 5.1). Pour cela, nous proposons d'effectuer le raccordement à l'aide de carreaux de COONS, puisque

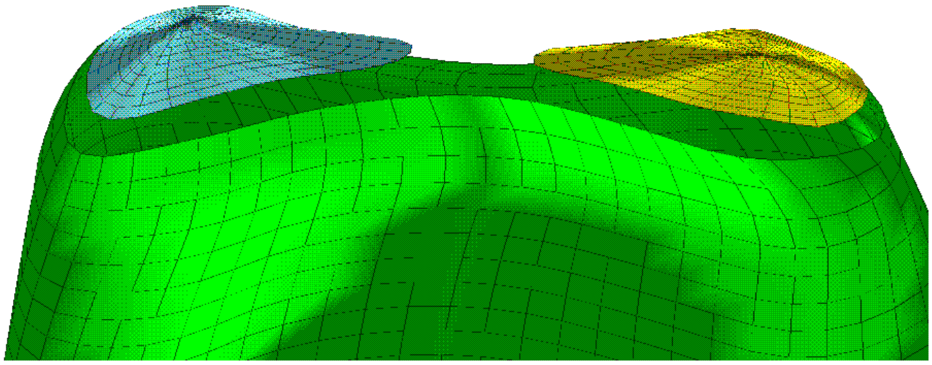


FIG. 5.1 – *Surfaces avant raccordement*

nous disposons des courbes frontières de la zone d'embranchement. Ce type de carreau fournit une surface de qualité en interpolant les contours qui le définissent. L'annexe B regroupe les équations nécessaires à la création de tels carreaux.

Pour obtenir un résultat satisfaisant tout en gardant une solution techniquement réalisable, nous avons choisi de raccorder à l'aide de trois carreaux de Coons, disposés comme cela est illustré sur la figure 5.2. Les deux premiers carreaux font la jointure entre le tronc et les

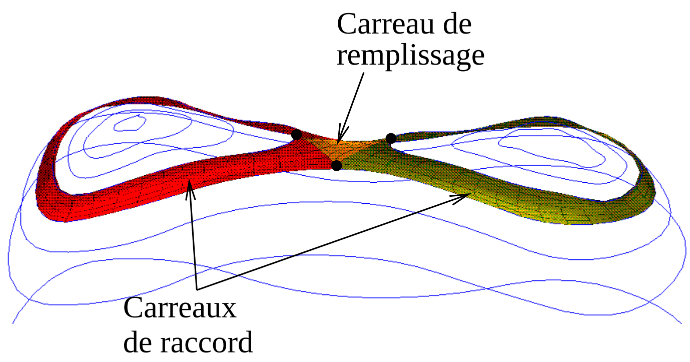


Fig. 5.2 – *Deux carreaux de raccord et un de remplissage*

branches, ce sont les carreaux de raccord. Le troisième carreau, appelé carreau de remplissage, comble l'espace entre les deux carreaux précédents afin d'obtenir une surface gauche de bonne qualité, qui repose sur les contours existants.

Le choix des sommets des carreaux et des contours intérieurs est déterminant pour obtenir une surface d'embranchement correcte. La position des quatre sommets est dépendante de la forme et du placement relatif du tronc et des branches. Une autre difficulté est le raccordement de surfaces suivant leur frontière définie par des courbes B-splines avec des bases nodales et des paramétrisations différentes. En effet, la moitié du tronc se raccorde sur la totalité d'une branche, et l'autre moitié sur l'autre branche. Il faut donc mettre en correspondance des courbes ouvertes avec des courbes B-splines périodiques.

La méthode proposée fournit donc un embranchement avec une continuité C^0, puisque les carreaux de Coons possèdent naturellement cette propriété. Cependant, une continuité d'ordre 0 n'est pas toujours satisfaisante, une arête vive au milieu d'une surface lisse n'étant pas très agréable visuellement. Il devient indispensable de réaliser un embranchement avec égalité des plans tangents, et nous proposons une solution pour obtenir une continuité G^1 lors du raccordement.

D'autre part, nous détaillons le cas de deux branches mais il arrive d'avoir à effectuer des embranchements multiples. Notre méthode est extensible dans certains cas, les plus simples, lorsque les branches sont disposées régulièrement. Cependant, d'autres solutions sont envisagées pour répondre au problème général d'embranchement (paragraphe 5.3.2).

5.2 Création des carreaux

Dans cette section, nous décrivons la génération des différents carreaux pour effectuer le remplissage de la zone de bifurcation, ainsi que les outils nécessaires pour obtenir une surface lisse.

5.2.1 Jointure C^0 entre le tronc et les branches

Nous utilisons donc des carreaux de COONS pour raccorder le tronc et les branches. Nous disposons des deux courbes extrémités des surfaces à raccorder. Sur le tronc, la courbe frontière est une restriction de la courbe totale sur un intervalle défini de manière à obtenir un carreau de raccord le plus lisse possible (cf. 5.2.2). Il reste à définir les deux autres courbes frontières (Fig. 5.3). Nous choisissons des courbes B-splines uniformes de degré 2 ce qui per-

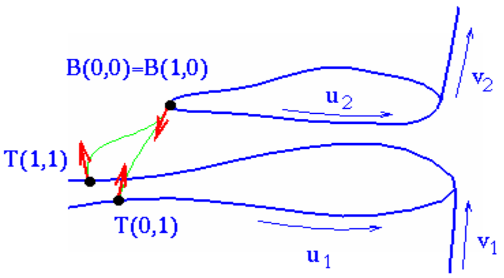


FIG. 5.3 – *Détermination des courbes frontières des carreaux de raccord*

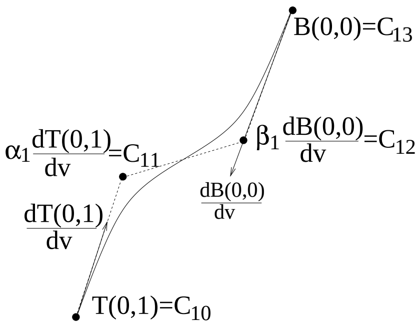


FIG. 5.4 – *Points de contrôle de la première courbe*

met d'assurer une continuité G^1 entre les deux surfaces au moins le long de ces courbes. Les points de contrôle sont définis ainsi (Équation 5.1) :

$$
\begin{aligned}
C_{10} &= T(0,1) & C_{20} &= T(1,1) \\
C_{11} &= \alpha_1.\frac{dT(0,1)}{dv_1} & C_{21} &= \alpha_2.\frac{dT(1,1)}{dv_1} \\
C_{12} &= \beta_1.\frac{dB(0,0)}{dv_2} & C_{22} &= \beta_2.\frac{dB(1,0)}{dv_2} \\
C_{13} &= B(0,0) & C_{23} &= B(1,0)
\end{aligned} \tag{5.1}
$$

Nous choisissons les coefficients α et β pour obtenir une répartition « uniforme » des points de contrôle lorsque cela est possible (Fig. 5.4).

Nous agissons de manière analogue pour raccorder le tronc avec l'autre branche. La configuration globale des courbes frontières est représentée sur la figure 5.5. La figure 5.6 illustre le résultat du raccordement entre le tronc et les branches.

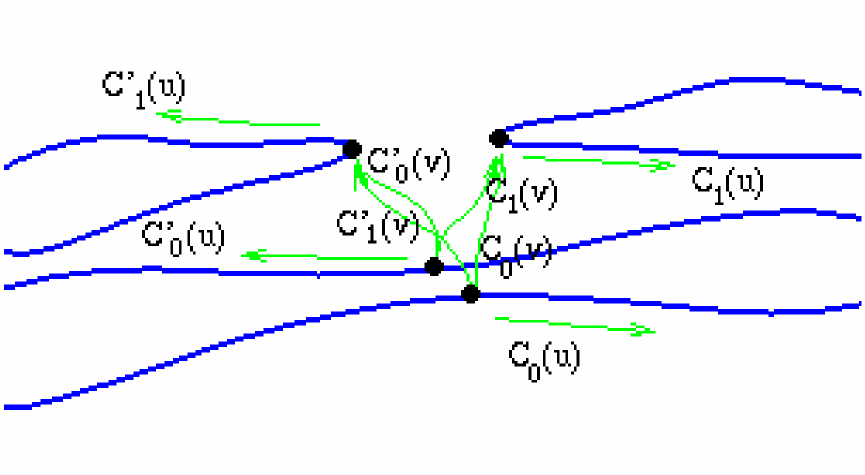


FIG. 5.5 – *Orientation des courbes frontières*

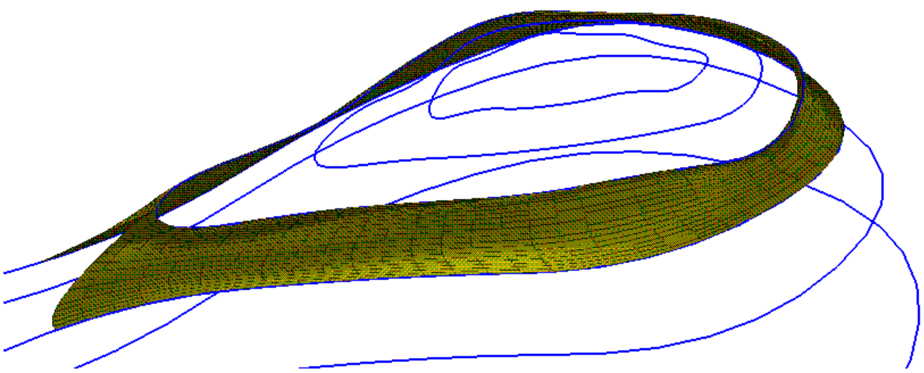


FIG. 5.6 – *Carreaux de raccord entre le tronc et les branches*

5.2.2 Reparamétrisation

Un des problèmes soulevés par la méthode précédente de raccordement entre un tronc et des branches, est la paramétrisation des courbes frontières. C'est une phase critique dans la construction des carreaux de COONS. Cela est dû à la formulation même du carreau, qui est une fonction de mélange entre les points des frontières. Lorsque la forme de ces courbes est très différente ou que ces courbes sont définies sur des séquences nodales très différentes, il peut apparaître des torsions, voire des plis sur le carreau reconstruit (Fig. 5.8(a)). Pour éviter ces désagréments, il est nécessaire de paramétrer correctement, c.-à-d. non-uniformément, les points sur les courbes frontières lors du calcul de l'intérieur du carreau de raccord.

Le premier problème est que l'on veut raccorder une portion de la courbe C_0 (le tronc) décrite sur un intervalle réduit $[a, b]$, avec une courbe complète C_1 (une des branches) décrite sur l'intervalle $[0, 1]$. La question est donc comment paramétrer pour avoir la correspondance entre la portion du tronc et le contour entier de la branche. Comme le point $C_0(0)$ coïncide rarement avec le sommet du carreau de raccord $C_0(a)$, pas plus que les points $C_0(1)$ et $C_0(b)$ du coté opposé, il est nécessaire de reparamétrer la portion de courbe du tronc sur laquelle s'appuie le raccord. Dans ce cas, le reparamétrage est simplement une bijection entre $[0, 1]$ et l'intervalle réduit $[a, b]$.

D'autre part, il faut trouver un paramétrage pour éviter la torsion entre les contours, voire pour éviter l'auto-intersection de carreaux de COONS dans la zone d'embranchement. Le paramétrage doit suivre parfaitement un contour par rapport à l'autre. Les deux contours opposés doivent être mélangés lors du raccordement en tenant compte de leur forme et non plus de façon uniforme. L'une des approches possibles pour mettre en correspondance les points d'un contour à l'autre peut s'inspirer de celle de MEYERS [MSS92]. L'information des points correspondants est utilisée pour définir un paramétrage adapté aux différences entre les courbes frontières [PT95]. La figure 5.7 illustre, sur un exemple, la reparamétrisation des courbes, où h_0 et h_1 sont les fonctions de reparamétrisation associés aux courbes frontières C_0 et C_1, respectivement (ici $a = 0$ et $b = 1$ pour simplifier l'écriture).

L'équation B.3 est alors modifiée comme suit :

$$\begin{aligned} S(u,v) \quad = \quad & C_0(h_0(u)).g_0(v) + C_1(h_1(u)).g_1(v) \\ & +C_0(v).f_0(u) + C_1(v).f_1(u) \\ & -S(0,0).f_0(u).g_0(v) - S(0,1).f_0(u).g_1(v) \\ & -S(1,0).f_1(u).g_0(v) - S(1,1).f_1(u).g_1(v) \end{aligned} \tag{5.2}$$

Ainsi, $u \in [0, 1]$ et $h_0(u)$ ou $h_1(u) \in [a, b]$. Les fonctions h_0 et h_1 permettent d'avoir deux systèmes de paramètres selon la courbe sur laquelle on se trouve. Comme ces deux fonctions sont calculées en tenant compte simultanément de la forme de chaque contour, leur utilisation empêche la formation de plis ou d'auto-intersection sur le carreau de raccord. Il est a noter que la courbe initiale et la courbe reparamétrée sont identiques géométriquement, mais différentes paramétriquement.

La figure 5.8(b) montre le nouveau carreau de raccordement, illustrant l'efficacité du reparamétrage, par rapport à un paramétrage uniforme comme illustré sur la figure 5.8(a).

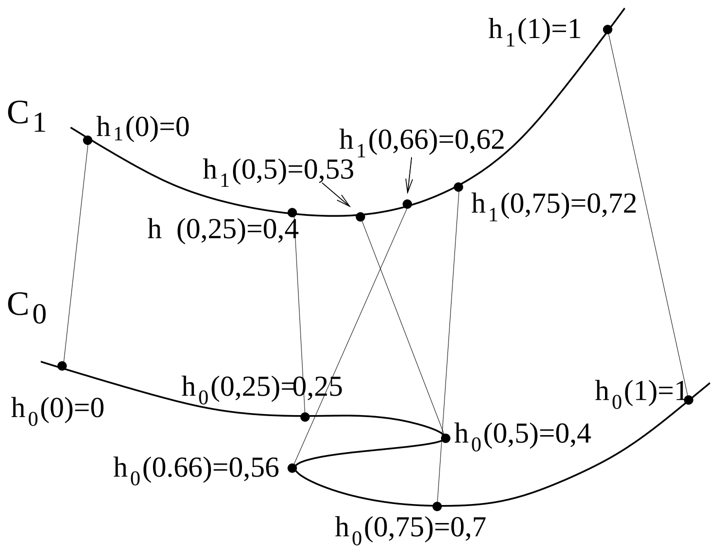
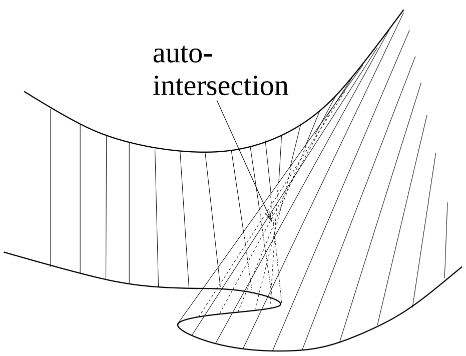
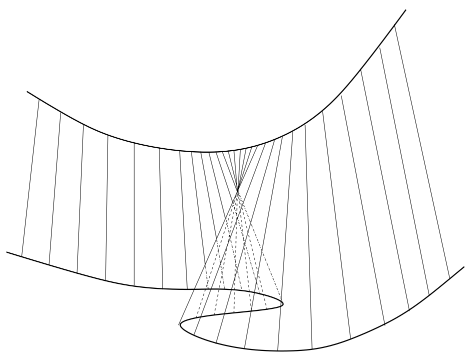


FIG. 5.7 – *Exemple de paramétrisation*



(a) Problèmes dus à la paramétrisation uniforme : auto-intersection de la surface



(b) Reparamétrisation des courbes, la surface est correcte

FIG. 5.8 – *Problèmes et solution au paramétrage*

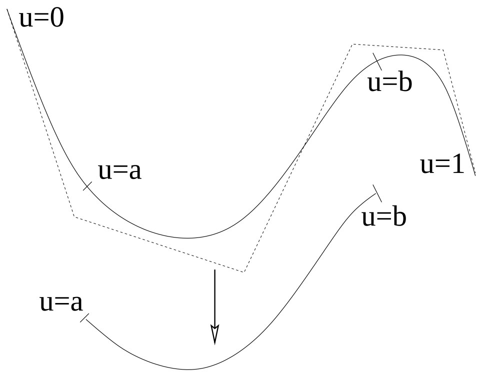


FIG. 5.9 – *Restriction de l'intervalle utile d'une courbe*

5.2.3 Problème de la continuité

L'utilisation des carreaux de COONS classiques, c.-à-d. par mélange polynômial, fournit en général une continuité C^0 avec les carreaux adjacents. Pour obtenir une continuité plus élevée, nous utilisons un autre type de carreau, lui aussi développé par COONS. Il s'agit d'un mélange de surfaces de BÉZIER, par combinaison des points de contrôle (cf. section B.2 en annexe).

En pratique, il n'est pas facile de construire un carreau de COONS bicubique. Dans notre cas, la création du carreau de raccord, les champs dérivés ne sont pas disponibles le long de la frontière dans la direction longitudinale (en v). Nous utilisons donc une version simplifiée de ce modèle, puisque nous ne tenons compte des dérivées que dans une seule direction. D'autre part, comme nous voulons raccorder des portions de courbes B-splines, il est nécessaire de redéfinir ces courbes sur l'intervalle correct pour pouvoir en déduire les points de contrôle des surfaces cubiques S_1 et S_2 (Équations B.7 et B.8 en annexe). Nous voulons donc construire une nouvelle courbe, correspondant exactement à la courbe initiale, mais sur une partie uniquement de son intervalle de définition, $[a, b]$ avec $a \geq 0$ et $b \leq 1$. Nous utilisons un procédé de restriction de l'intervalle utile d'une courbe B-spline [Leo91], qui se base sur l'algorithme d'insertion de nœuds de BOEHM [Boe80]. La figure 5.9 illustre l'extraction d'une courbe limitée par $u = a$ et $u = b$.

De plus, lorsque les informations sont obtenues à partir des surfaces adjacentes au carreau, ce qui est notre cas, la compatibilité entre les vecteurs de torsion (Éq. B.6 et fig. B.3 en annexe) aux sommets du carreau est rarement vérifiée. Cela implique pour le premier sommet :

$$\frac{dD_0(u=0)}{du} \neq \frac{dD_0(v=0)}{dv}$$

Il en est de même pour les trois autres sommets. Nous pouvons alors construire une surface présentant une légère discontinuité dans la direction de la tangence (une légère torsion apparaît alors aux sommets du carreau), en spécifiant une tolérance angulaire entre les vecteurs de torsion dans chaque direction, et calculant les vecteurs finaux par une moyenne [PT95].

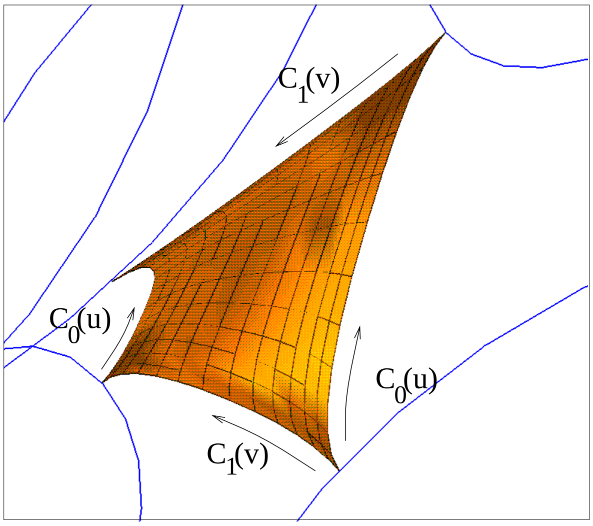


FIG. 5.10 – *Carreau de remplissage*

5.2.4 Remplissage entre les branches

Une fois que les raccords entre le tronc et les branches ont été effectués, il reste un espace à combler entre les deux carreaux précédemment crées. Nous utilisons là aussi les carreaux de COONS, qui permettent d'obtenir facilement une continuité entre les carreaux adjacents qui soit C^0 (ou G^1 si on utilise le modèle présenté au paragraphe précédent). Les frontières de ce nouveau carreau ont déjà été déterminées lors de la phase précédente, il suffit alors d'interpoler les quatre frontières pour obtenir une surface lisse (Fig. 5.10).

Les figures 5.11 et 5.12 représentent différentes vues de la zone d'embranchement.

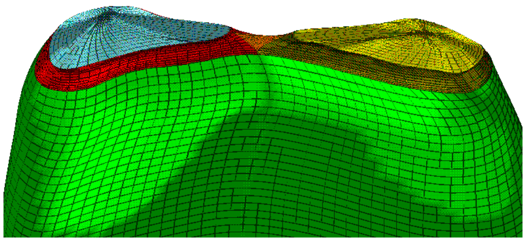


FIG. 5.11 – *Exemple de carreaux de raccordement*

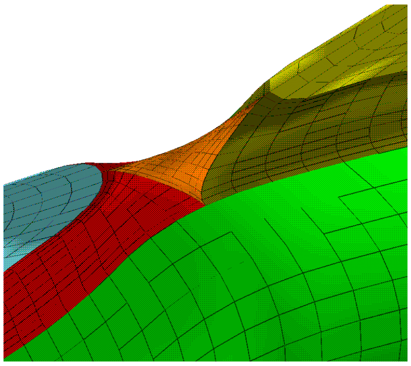


FIG. 5.12 – *Vue rapprochée sur la zone de raccordement*

5.3 Choix des courbes frontières

Le choix des courbes frontières est déterminant puisqu'il va conditionner la qualité du raccord. Il s'agit ici de trouver les points optimaux comme sommets des carreaux de raccord. Pour conserver la continuité C^0, ces points sont à prendre sur les bords des surfaces à raccorder. La solution la plus simple est de minimiser la longueur des courbes frontières du carreau intérieur de remplissage. Les sommets situés sur les branches sont ceux qui sont les plus proches entre les deux branches. Ensuite, les sommets sur le tronc minimisent la distance avec les deux points précédents de part et d'autre du contour. Les sommets obtenus sont corrects si les courbes frontières sont de forme simple et régulièrement placés. Par contre, dans le cas de contours plus complexes, le choix de tels sommets peut provoquer des plis ou des problèmes de paramétrage. Il est alors nécessaire de trouver des sommets plus adapté à la disposition des contours.

Au chapitre précédent, nous avons illustré notre approche sur un cas simple, nous nous intéressons maintenant à la généralisation de la méthode aux cas des embranchements plus complexes. La disposition relative ainsi que la géométrie des contours sont des facteurs importants, qui influent sur la forme de l'embranchement. Il est aussi intéressant de pouvoir gérer des embranchements multiples de type $1 \to n$, un tronc se sépare en plusieurs branches, et éventuellement $m \to n$, où un nombre quelconque $m > 1$ de pré-branches se combinent pour former un nombre différent $n > 1$ de post-branches. Ces configurations peuvent se produire dans le cas d'objets fortement non-convexes, ou quand les sections sont très espacées.

5.3.1 Disposition et forme

Les méthodes classiques d'embranchement produisent des résultats peu probants car elles ne tiennent pas compte de la géométrie du contour précédant l'embranchement, pas plus que de celles des suivants. Pour les embranchements simples, de type $1 \to 2$, il a été proposé de distinguer trois classes. Les embranchements de type plus complexe $1 \to n$ peuvent aussi être analysés selon ces termes. Les types sont déterminés en étudiant l'axe médian de la projection des contours sur un même plan. L'axe médian est un graphe planaire construit à partir du diagramme de Voronoï [Mon87], il contient exactement autant de boucles que de branches. La disposition des boucles permet de déterminer le type de branchements (Fig. 2.4) :

1. par défaut, ou si les boucles se joignent en un seul point, c'est une selle de cheval (Fig. 2.4(a)) ;
2. si elles partagent une séquence d'arêtes, c'est un canyon (Fig. 2.4(b)) ;
3. si elles sont jointes par un ensemble d'arêtes qu'elles ne partagent pas, c'est une crête (Fig. 2.4(c)).

La figure 5.13 montre un exemple de contours multiples contenant différents types d'embranchement, dont un canyon, une selle de cheval et deux crêtes.

Nous nous sommes inspirés de cette méthode pour définir les courbes frontières des carreaux de COONS servant à remplir la zone de l'embranchement. Une fois que le type d'embranchement a été déterminé, nous pouvons utiliser le graphe de l'axe médian pour déterminer

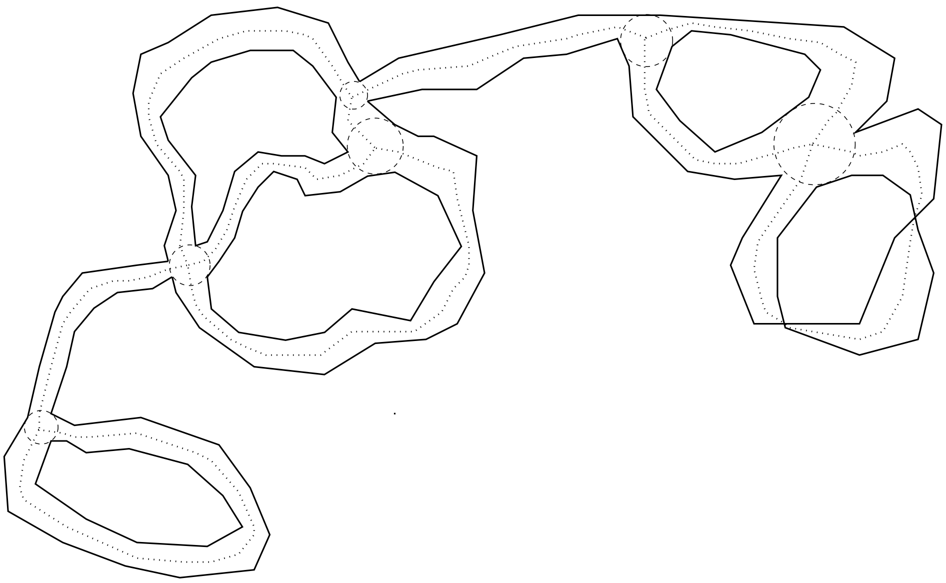


FIG. 5.13 – *Exemple de branchement complexe*

les sommets des courbes frontières, et en déduire celles-ci. Nous appliquons une méthode de reconstruction différente pour chacun des types d'embranchement.

5.3.1.1 Selle de cheval

Pour une selle de cheval, la meilleure solution est de connecter les branches au plus près. À partir de la jonction, nous considérons les points les plus proches sur chaque branche et sur le tronc, ce sont les sommets cherchés (Fig. 5.14). Nous utilisons la méthode présentée au chapitre 5 pour reconstruire l'embranchement. Le résultat d'un tel embranchement est illustré par la figure 5.15.

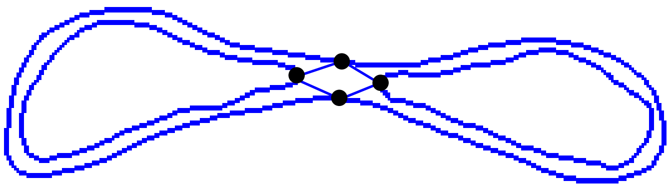


FIG. 5.14 – *Détermination des sommets des carreaux*

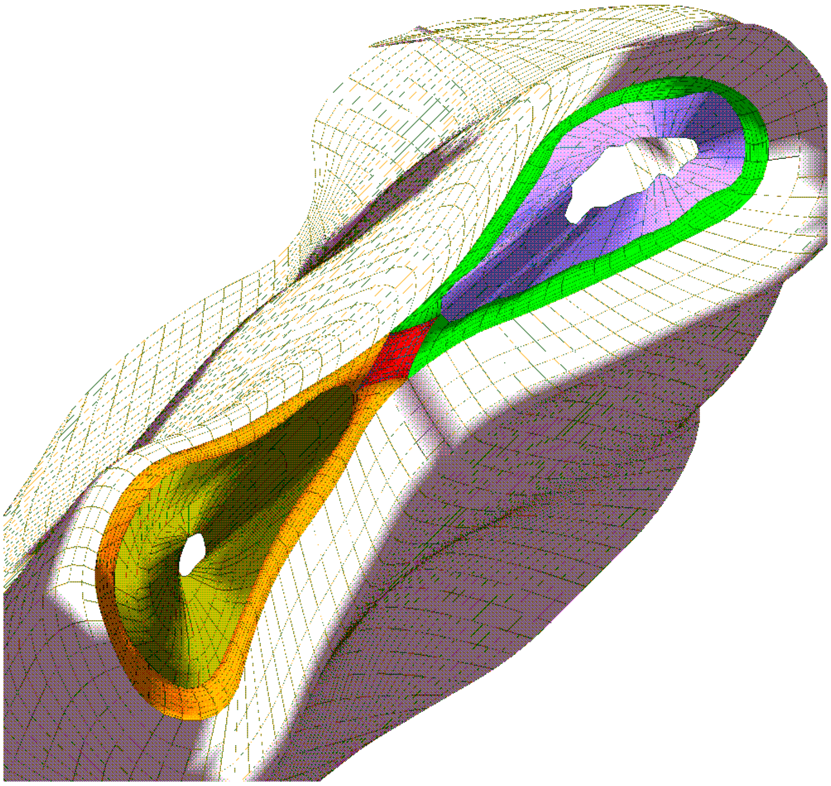


FIG. 5.15 – *Carreaux de* COONS *sur un embranchement de type selle de cheval*

5.3.1.2 Canyon

Pour un canyon, la méthode est un peu différente. Les deux jonctions permettent de définir les sommets des courbes frontières sur chaque contour (Fig. 5.16). Le remplissage est un peu plus complexe puisqu'il nécessite trois carreaux de Coons pour connecter les branches et deux carreaux supplémentaires pour boucher les trous et assurer la continuité G^1. Les deux premiers carreaux rectangulaires se situent entre la moitié du tronc et chacune

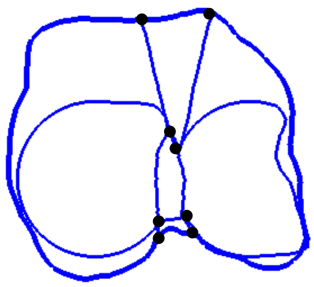


Fig. 5.16 – *Détermination des sommets des carreaux*

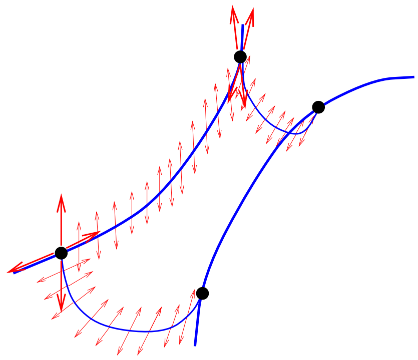


Fig. 5.17 – *Détermination des courbes frontières et des tangentes*

des deux branches comme pour une selle de cheval. Le troisième carreau rectangulaire permet de remplir le canyon. C'est un carreau en creux s'appuyant sur les parties des contours des branches qui sont en regard. Les deux autres courbes frontières sont définies pour fermer le carreau en tenant compte des tangentes au bord des branches (Fig. 5.17). Des problèmes de paramétrisation apparaissent. En effet, nous voulons interpoler des courbes qui ne sont pas définies sur le même intervalle de paramétrage, et sur lesquelles les paramètres ne varient pas uniformément. Une solution peut être trouvée en adaptant la méthode proposée à la section 5.2.2 traitant de la reparamétrisation. Enfin les espaces vides sont remplis à l'aide de carreaux de Coons de la même façon que pour la selle de cheval. Le résultat du remplissage d'un embranchement de type canyon peut être trouvé sur la figure 5.18.

5.3.1.3 Crête

Pour l'embranchement de type crête illustré par la figure 5.19, nous procédons de la même manière que pour le canyon, à l'exception du troisième carreau rectangulaire. Cette fois, il ne joint plus les deux branches mais deux parties du tronc situées de part et d'autre de la crête (Fig. 5.20).

5.3.2 Embranchements de type $m \to n$

Même si cela est assez rare dans la plupart des applications, plusieurs branches peuvent se rejoindre et immédiatement se séparer en un nombre quelconque de nouvelles branches. Il

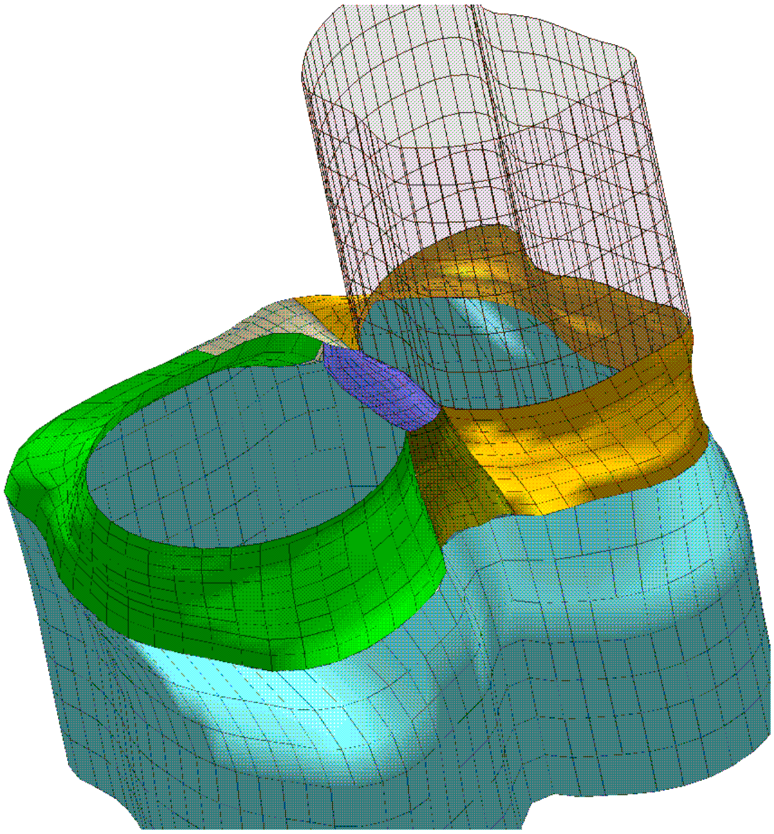


FIG. 5.18 – *Carreaux de* COONS *sur un embranchement de type canyon*

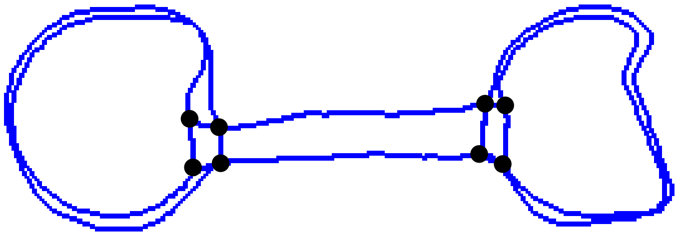


FIG. 5.19 – *Détermination des sommets des carreaux*

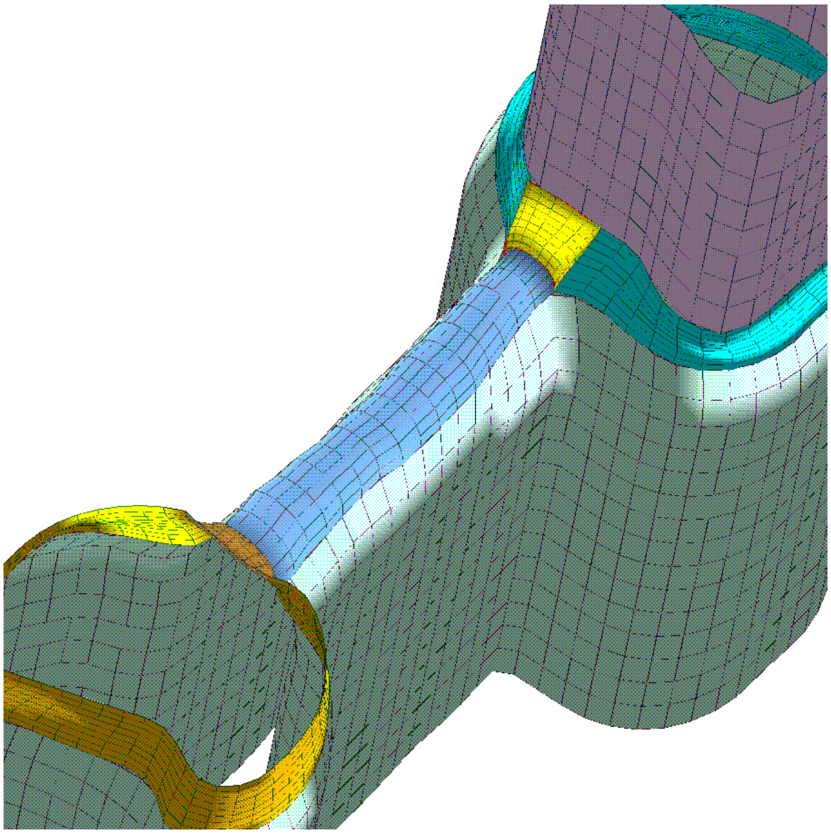


FIG. 5.20 – *Carreaux de* COONS *sur un embranchement de type crête*

est principalement dû à un échantillonnage trop épars pour un objet comportant beaucoup de branches. Il est souvent impossible de déterminer une topologie exacte pour ce genre de cas.

Nous avons vu au paragraphe précédent 5.3.1 qu'un embranchement de type $1 \to n$ peut facilement se décomposer en termes de canyon, selle de cheval et crête (qui sont de type $1 \to 2$). De même, pour les embranchements $m \to n$, il est possible de le décomposer en branchements $1 \to n$ reliés entre eux. Mais pour cela, nous avons besoin de la forme et de la disposition du contour de chaque pré-branche, qui peut aussi être une post-branche pour un autre branchement $1 \to n$. De ce fait, le cas général est très difficile à prendre en compte puisqu'il fait intervenir un grand nombre de paramètres et que de nombreux cas distincts apparaissent en fonction de m et de n. Il n'existe pas de méthode automatique qui permette de résoudre correctement ce problème. Il est d'ailleurs relativement difficile de produire interactivement une solution acceptable. Une solution pratique serait de calculer un voire plusieurs contours intermédiaires jusqu'à obtenir une suite de branchement $1 \to n$, en remplacement de celui de type $m \to n$. Malheureusement, ce n'est pas toujours possible.

6 Conclusion

Dans cette partie du mémoire, nous avons présenté une méthode de reconstruction d'objets définis par des sections, et ceci à l'aide d'un ensemble de surfaces paramétriques. Nous résumons ici les principaux apports de cette méthode. Nous constatons aussi que certaines limitations des modèles paramétriques, notamment au niveau de la modélisation d'objets déformables, sont des handicaps lorsque l'on veut animer les objets issus de la reconstruction.

6.1 Principales contributions

6.1.1 « Skinning »

Dans un premier temps, nous nous sommes restreints à des formes homéomorphes à des cylindres ou des sphères. Nous avons reconstruit ces formes à l'aide de la technique de « skinning ». Nous avons proposé une nouvelle méthode de définition du vecteur nodal lors de la mise en compatibilité des courbes. Nous n'effectuons pas la mise en compatibilité exacte mais proposons une approximation basée sur une moyenne. Cela évite de générer un trop grand nombre de points de contrôle sur la surface finale, qu'il faudra , de toute façon, réduire par la suite. Dans notre cas, la réduction se fait avant le calcul des positions des points de contrôle, ce qui accélère la reconstruction. Nous nous dispensons ainsi de la phase de réduction, qui est la plus coûteuse dans la méthode traditionnelle, qui consiste à évaluer l'erreur engendrée par la suppression de chaque nœud,et ceci pour toutes les courbes. Bien sûr, nous ne pouvons réaliser ce moyennage que si les courbes ont des caractéristiques géométriques similaires. Cela suppose que la forme et la disposition des contours varie peu de l'un à l'autre. Dans le cas contraire, une compatibilité exacte entre les courbes est effectuée. Ainsi, en conclusion, nous commençons par reconstruire une surface relativement simple. Ensuite, à l'aide des algorithmes de subdivision, nous rajoutons autant de points de nœuds que nécessaire dans les zones où la précision est médiocre. Cette approche est à l'opposé des autres méthodes de « skinning », qui commencent par une surface beaucoup trop précise pour arriver à une surface acceptable.

6.1.2 Embranchements

Une contribution importante de ce travail, dans le domaine de la reconstruction de formes lisses, a été l'évolution vers des formes possédant des embranchements. Nous nous distinguons des méthodes précédemment publiées par l'utilisation des carreaux de Coons. Ces surfaces paramétriques permettent d'interpoler des courbes frontières. Il est aussi possible d'ajouter des contraintes de tangence sur les bords, ce qui rend ce modèle intéressant pour combler les espaces restés vides entre les branches. L'utilisation des carreaux de Coons nous a permis de raccorder les branches à l'aide d'une seule surface bi-paramétrique, tout en évitant l'utilisation d'un grand nombre de surfaces dans la zone de raccordement. Nous avons décidé d'en mettre un petit nombre, le plus faible possible, mais sans perdre de vue la continuité. Cela nous a conduit à un carreau par branche, plus ceux nécessaires pour les raccorder entre eux. En utilisant une classification déjà établie, nous avons proposé une méthode pour reconstruire chaque type de bifurcation (embranchement de type $1 \to 2$), puis nous avons montré comment généraliser la reconstruction aux objets comportant des des embranchements de type $1 \to n$, voire $m \to n$ dans certains cas.

Le résultat est une surface composée de carreaux bi-paramétriques B-spline et de carreaux de Coons. Tous ces carreaux sont raccordés entre eux avec une continuité G^1, sans pour autant provoquer d'élévation exagérée du degré.

6.2 Application

Depuis quelques années, le laboratoire LIGIM, et le groupe Reconstruction 3D en particulier, est impliqué dans des projets européens dans le domaine médical. Le projet BIOMED II Infocus a débuté début 1996. L'objectif principal est de fournir aux professionnels du médical des outils radicalement nouveaux pour les phases de préparation, de traitement et de suivi lors d'opérations en radiothérapie.

Dans le cadre d'un traitement radiothérapique, une dose létale de radiations est administrée dans la zone des cellules cancéreuses, avec un minimum de dommages aux tissus sains environnants. L'amplitude des radiation ainsi que leur pénétration à l'intérieur des tissus sont connues. Ceci a permis le développement de la radiothérapie conformationnelle, pour laquelle le médecin doit définir pour chaque patient les doses de radiations au cours du temps et en fonction des déplacements de la zone à traiter.

Le projet peut se découper en plusieurs points, qui seront traités par les différents partenaires :

1. Développer un capteur optique sans contact pour déterminer la position du patient, dans le but d'obtenir une reconstruction tridimensionnelle précise de la surface externe ;
2. Développer un outil pour améliorer la qualité des images (megavoltage image) issues de la machine de traitement radiothérapique ;
3. Développer des techniques pour la modélisation des organes internes, à partir des acquisitions par tomographie ;
4. Amélioration des techniques de calcul dosimétrique et de préparation du traitement

pour aider le médecin à utiliser la quantité d'informations nouvelles apportées par le projet.

Notre travail concerne les points 1 et 3 du projet. Le principal problème est un problème géométrique concernant les mouvements du patient et les changements de forme et de position des organes internes. Pour le résoudre, il est nécessaire de créer un modèle géométrique de la surface externe du patient ainsi que des volumes internes à traiter, puis de corréler ces deux modèles dans un même référentiel. Dans ce cadre, nous avons développé un ensemble d'outils permettant la reconstruction et la manipulation des formes anatomiques externes (le corps du patient) et internes (les organes). Nous avons développé une interface graphique, qui propose diverses fonctionnalités pour manipuler et traiter des contours. Il permet aussi de reconstruire des courbes et des surfaces à partir de ces contours, ainsi qu'un grand nombre de fonctions d'affichage des modèles créés.

Cette interface est une réalisation commune, et elle se veut être une interface utilisée par tous les membres du groupe *Reconstruction 3D* du laboratoire. La programmation a été faite en langage *C*. L'interface graphique a été développée par Y. PERRET. La base de l'interface (fenêtres et boutons) a été créée à l'aide de la librairie graphique *libforms*, elle même reposant sur la librairie *Xlib* pour les systèmes *X Window*. La partie affichage 3D, quant à elle, a été écrite en utilisant la librairie graphique *Mesa/GL*. Le total représente environ 25 000 lignes de code. L'ensemble des fonctions de traitement des contours et des surfaces a été implémentée en *C* de façon indépendante, puis a été rajouté comme un module de l'interface. Cela constitue environ 11 000 lignes de *C*.

La figure 6.1 présente quelques fenêtres de cette interface. Elle représente la zone d'affichage, dans laquelle figurent trois organes (vessie, prostate et rectum) reconstruits à l'aide de surfaces fermées. Dans une seconde fenêtre sont affichés un contour et la courbe B-spline correspondante avec les points de contrôle. La dernière fenêtre est celle de contrôle des objets de l'interface.

6.3 Handicaps des modèles paramétriques

Les modèles paramétriques sont intéressants pour obtenir une représentation compacte et lisse d'un objet. Cependant, la prise en compte de l'aspect déformable des objets « mous » est difficilement maîtrisable à l'aide de ces surfaces. Les travaux sur l'animation des NURBS [PH94, RBB97] ont permis d'aborder ces problèmes, mais le temps de calcul élevé rend leur utilisation limitée dans notre application médicale. Ceci nous a amené à généraliser le concept de la modélisation géométrique en nous orientant vers de nouveaux types de modèles pour représenter et animer des objets déformables. Cette étude fait l'objet de la seconde partie de la thèse.

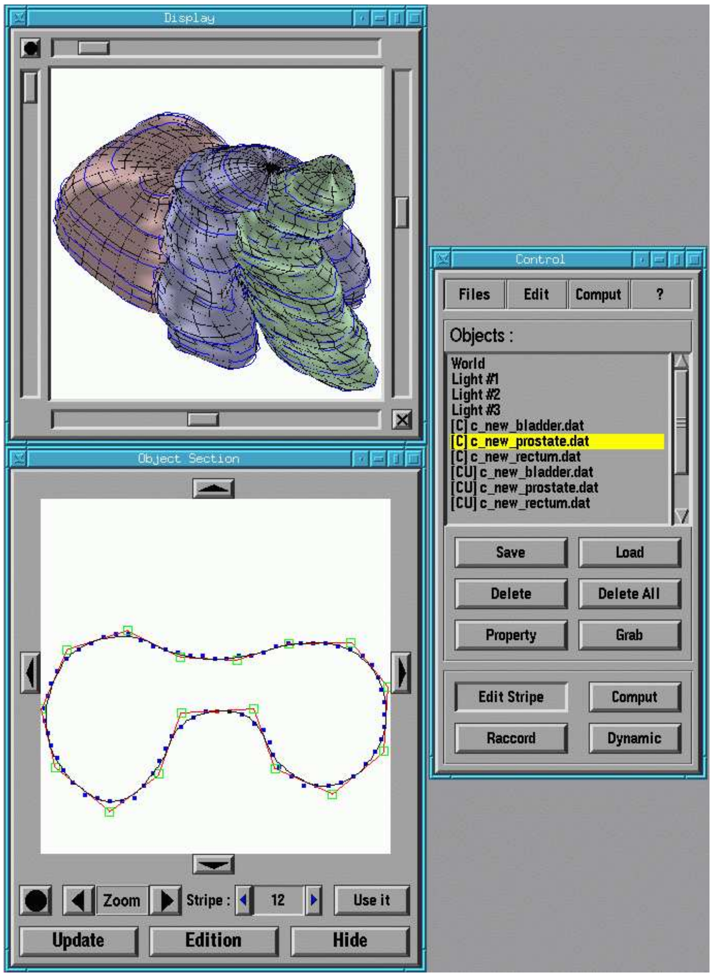


FIG. 6.1 – *Capture d'écran sur des fenêtres de notre interface*

Deuxième partie

Généralisation vers des modèles déformables

1 Position du problème

En informatique graphique, l'animation est un domaine de recherche assez récent. La première utilisation fut le calcul des positions intermédiaires. Ces images étaient générées par des techniques d'interpolation entre plusieurs positions clefs. Cette approche est encore très utilisée, notamment dans la production de dessins animés. Cependant, elle reste limitée en cas de scènes complexes. Mais surtout, le réalisme de l'animation dépend du savoir-faire du graphiste. Les modèles générateurs ont permis le réel développement de l'animation par ordinateur. Ces modèles utilisent les lois physiques pour calculer le mouvement des objets. Il suffit alors de décrire les contraintes de l'animation et la trajectoire de chaque objet est automatiquement calculée, de façon beaucoup plus réaliste qu'avec les modèles descriptifs.

Les applications de l'animation ne se limitent pas du tout aux dessins animés. Dans le domaine médical, la simulation des déformations est devenu un outil précieux, par exemple en chirurgie pour simuler une opération délicate [CDA96, PBP97, MC96, CEO$^+$93]. De nombreux modèles ont été développés pour autoriser des déformations ou des découpes. Ces modèles ont été repris pour l'animation de matériaux déformables voire très déformables, comme la pâte à modeler, qui nécessitent de prendre en compte des séparations de matière.

De tels matériaux peuvent être utilisés lors de la conception de prototypes ou pour toute autre application de design. La simulation d'avalanches ou de coulées de lave requiert des modèles situés à mi-chemin entre les solides et les liquides. Les modèles sont difficiles à manipuler car ils font intervenir des notions de mécanique du solide mais aussi des fluides. De ce fait, l'ordinateur peut être d'un grand recours.

Le but de nos travaux, dans cette partie, est de proposer un modèle d'objet très déformable (c.-à-d. autorisant les changements de topologie). L'application immédiate est la simulation des mouvements des organes internes lors d'un traitement en radiothérapie, dans le cadre du projet BIOMED II Infocus.

Cette seconde partie se décompose ainsi : après avoir brièvement introduit le problème, nous étudions, au chapitre 2, l'état de l'art en modélisation et animation d'objets déformables, en donnant les limitations et les avantages de chacune des techniques existantes. Le chapitre 3 présente, de façon détaillée, le modèle de particules que nous avons jugé particulièrement intéressant. Ceci a permet la reconstruction d'un objet (volume) à l'aide d'un ensemble de particules. Le chapitre 4 est l'optimisation des méthodes présentées dans le chapitre 3. Nous décrivons ici notre approche de reconstruction multi-couches. Le chapitre 5 concerne l'animation à proprement parler. Nous expliquons les solutions choisies pour résoudre un certains nombre de problèmes courants dans le domaine. Nous présentons l'implémentation et les résultats de l'application médicale sur laquelle nous avons travaillé, avant de conclure cette partie.

2 État de l'art : modèles déformables

L'intérêt grandissant pour la simulation de matière déformable a vu l'apparition d'un grand nombre de méthodes. Il s'agit de simuler des comportements réalistes à l'aide paramètres physiques tels que l'élasticité ou la pression interne. Nous pouvons classer celles-ci en deux grands groupes, en fonction des types de déformations qu'elles permettent de gérer. Ainsi, nous distinguons l'ensemble des modèles dits « structurés », qui peuvent subir de petites déformations, mais qui ne peuvent pas changer de topologie. Ce premier ensemble est très vaste et assez varié. L'autre groupe contient les méthodes qui utilisent des modèles « non-structurés » et qui permettent de grandes déformations. En particulier, cette absence de topologie fixe autorise des séparations ou des regroupements de matière qui sont très utiles dans certaines applications.

Dans ce chapitre, nous étudions les différentes méthodes existantes dans le domaine de la reconstruction et de l'animation d'objets déformables. Nous tenterons de cerner les intérêts et les limites de chacune des méthodes. On pourra aussi se référer à l'article [GO95] qui présente différentes approches.

2.1 Modèles structurés

Les objets déformables structurés ont été une première amélioration des modèles solides. Ils ne peuvent subir que des déformations légères, leur topologie ne pouvant pas changer au cours de l'animation. La représentation choisie pour modéliser de tels objets est celle des maillages : les déformations se propageront à tout l'objet par des déplacements des éléments discrétisés. Nous distinguons alors deux approches : la résolution d'équations différentielles de milieux continus et l'assemblage d'éléments discrétisés.

2.1.1 Approche continue

Le premier modèle d'objets élastiques a été proposé par TERZOPOULOS et al. [TF88b]. Le mouvement des points de l'objet est donné par les équations de Lagrange suivantes :

$$\mu \frac{\partial^2 \vec{p}}{\partial t^2} + \gamma \frac{\partial \vec{p}}{\partial t} + \delta_{\vec{p}}\, \varepsilon = \vec{f} \tag{2.1}$$

où μ est la masse volumique du point a de position $\vec{p}$, γ est la densité de frottement. Le premier terme traduit l'inertie de l'objet, le second représente les forces de frottement et le dernier la force élastique interne qui résiste à la déformation. $\vec{f}$ est la somme des forces extérieures. L'énergie de la déformation ε tend à redonner sa forme initiale à l'objet :

$$\varepsilon = \int_{volume} ||\Gamma - \Gamma^0|| da_x da_y da_z \tag{2.2}$$

où G est la matrice représentant le tenseur métrique associé à l'objet, dont les termes sont définis ainsi :

$$\Gamma_{ij}(a) = \frac{\partial \vec{p}}{\partial a_i} \cdot \frac{\partial \vec{p}}{\partial a_j}$$

Si Γ^0 est le tenseur de l'objet dans sa position initiale, donc $||\Gamma - \Gamma^0||$ augmente quand l'objet est déformé. Cette formulation permet de définir plusieurs types de comportements élastiques. Malheureusement, le critère reste peu intuitif (une norme de tenseur). De plus, à chaque étape, il faut résoudre un ensemble d'équations différentielles, ce qui revient à inverser la matrice d'un énorme système.

Ces limitations ont amenés les mêmes auteurs à développer un nouveau modèle [TF88a]. Ils proposent une formulation hybride dans laquelle un objet est défini par une composante de référence r et par une composante déformable e, telles que :

$$\vec{q}(a,t) = \vec{r}(a,t) + \vec{e}(a,t)$$

r représente la forme de l'objet au repos mais aussi sa position actuelle. e se déforme mais ne se déplace pas par rapport à la référence. Cela permet d'écrire l'énergie de la déformation de façon simple puisqu'il ne faut plus prendre en compte les déplacements solides. Pour améliorer ce modèle, le composant de référence peut lui aussi être déformable. Il absorbe peu à peu certaines déformations. De nouveaux types de comportements découlent de cette propriété :

- visco-élasticité à l'aide de la relation $\frac{\partial \vec{r}}{\partial t} = \frac{1}{\eta}\vec{e}$. L'objet s'écrase si les forces sont continues et rebondit quand les forces sont des impulsions ;
- semi-plasticité, la déformation devient irréversible quand elle est trop importante ;
- fracture.

L'animation consiste toujours à résoudre un système matriciel à chaque étape, mais cette fois la matrice à inverser ne dépend plus du temps. Cependant, la base mathématique nécessaire est complexe et elle entraîne des temps d'exécution importants malgré les nombreuses approximations pour rendre le système calculable. De plus, chaque comportement est décrit par ses propres équations, il n'y a donc pas de modèle unifié. Les modèles sont peu intuitifs et peu maniables, donc inutilisables de manière interactive. Il est aussi difficile d'appliquer des contraintes sur l'objet, par exemple un taux de compressibilité ou une valeur de seuil de rupture.

2.1.2 Modèles paramétriques

COVER et al.[CEO+93] ont développé une modélisation qui supporte les manipulations en temps réel. Le modèle choisit est celui du contour actif, et dans ce cas une courbe B-spline minimisant l'énergie :

$$E(\mathbf{p}) = E_{int}(\mathbf{p}) + E_{ext}(\mathbf{p}) \text{ avec } \mathbf{p} = (x, y, z)$$

L'énergie interne est composée d'un terme représentant l'élasticité et d'un terme correspondant à la flexibilité :

$$E_{int}(\mathbf{p}) = \frac{\alpha|\dot{\mathbf{p}}|^2 + \beta|\ddot{\mathbf{p}}|^2}{2}$$

où α et β sont des coefficients permettant de faire varier le comportement physique de l'objet. Cette formulation mathématique fournit une base pour créer des modèles structurés, qui sont régis par des lois physiques. Les déformations sont ensuite répercutées sur le modèle à l'aide d'une méthode d'éléments finis. La déformation est totalement élastique, puisque réalisée par modification d'un ressort idéal. Par contre, elle est isotrope et linéaire. La position des points dépend du frottement des voisins, des forces extérieures et de la position d'équilibre. Cette association entre éléments finis et minimisation d'énergie permet une manipulation interactive des objets, mais le comportement dynamique n'est malgré tout pas pris en compte.

2.1.3 Superquadriques

Une première approche de description globale des objets déformables fut celle de TERZOPOULOS et METAXAS [TM91]. Ils utilisèrent les superquadriques qui sont une extension des quadriques classiques. Elles varient selon des paramètres de taille a_1, a_2 et a_3, ainsi que de rotondité ε_1 et ε_2 :

$$s = \left\{ \begin{array}{lcll} x & = & a_1 \cos^{\varepsilon_1}(\eta) \cos^{\varepsilon_2}(w) \text{ avec} & \eta \in [\frac{-\pi}{2}; \frac{\pi}{2}] \\ y & = & a_2 \cos^{\varepsilon_1}(\eta) \sin^{\varepsilon_2}(w) & w \in [\pi; \pi] \\ z & = & a_3 \sin^{\varepsilon_1} & \end{array} \right. \tag{2.3}$$

PARK et al. [PMY94] ajoutent des primitives de déformations globales telles que la courbure ou la torsion. Ils obtiennent ainsi un modèle suffisamment complexe pour pouvoir étudier les variations des paramètres du mouvement d'un ventricule.

TERZOPOULOS et METAXAS définissent un vecteur caractéristique de cette surface s, tel que $q = \{q_c, q_\theta, q_s, q_d\}$, où q_c et q_θ représentent les positions globales, q_s est la coordonnée de déformation globale et q_d est la coordonnée de déformation locale. Le comportement de l'objet est régi par l'équation de Lagrange :

$$M\ddot{q} + C\dot{q} + Kq = G_q + F_q$$

où M est la masse pour prendre en compte l'inertie, C est un coefficient de dissipation et K la rigidité. G_q et F_q sont les forces d'inertie et externes. Le modèle est utilisé pour construire une représentation qui coïncide avec des données initiales. Il est donc simplifié en fixant une masse nulle, ce qui est justifié car seul l'état final est important. L'équation devient alors $C\dot{q} + Kq = F_q$.

Une extension du modèle original a été présentée par DeCarlo et al. [DKM+95]. Cette fois, l'objet rigide possède une masse qui est répartie sur chaque nœud du maillage. La force exercée sur chaque nœud est dépendante de la pression P :

$$\vec{F} = \frac{P}{l}\vec{n}$$

où l est la longueur de l'arête du maillage (2D), et $\vec{n}$ la normale à cette arête. Cela leur a permis de simuler des comportements cinématiques d'un poumon à l'intérieur de la poitrine. Lors de l'inspiration ou de l'expiration, la cavité pulmonaire change de taille. La différence de pression entre la paroi et la membrane entraîne la déformation.

Comme précédemment, il faut ensuite discrétiser l'objet pour résoudre cette équation à chaque pas de temps. L'inconvénient est que les méthodes de résolution par éléments finis font une interpolation de l'objet. Neveu et al. [NFD95] proposent de faire un lissage au sens des moindres carrés à la manière des B-splines pour réduire le nombre d'éléments discrets. Ils réécrivent alors les équations du mouvement en remplaçant les formes nodales d'un élément fini par les fonctions B-splines d'un morceau de surface. De cette façon, le calcul des déformations se fait à partir des points de contrôle de la surface et non plus pour chaque nœud, ce qui est un gain en temps et en mémoire.

2.1.4 Approche discrète

Au lieu de décrire l'objet de façon continue puis de le discrétiser pour l'animer, il a été proposé de modéliser directement l'objet par un assemblage d'éléments discrets. La plupart du temps, il s'agit de masses ponctuelles reliées entre elles par des lois d'interaction de type ressort. Il devient alors très aisé d'écrire les équations du mouvement de tels objets, qui sont tout simplement les équations de la dynamique appliquées indépendemment en chaque point discret.

$$\begin{aligned} \frac{d\vec{v}}{dt} &= \vec{a} = \frac{\sum \vec{F}}{m} \\ \frac{d\vec{p}}{dt} &= \vec{v} \end{aligned} \tag{2.4}$$

où $\vec{a}$, $\vec{v}$ et $\vec{p}$ sont respectivement l'accélération, la vitesse et la position de chaque masse.

Une approche générale des systèmes masses-ressorts a été présentée dans le cadre du projet CORDIS/ANIMA. Un objet est représenté par un ensemble de points matériels, reliés entre eux par des liaisons. Une description de ce modèle peut être trouvée au chapitre 3 sur l'étude des systèmes de particules. Ce type de modèle discret permet une grande diversité des comportements.

Le modèles discrets sont très bien adaptés aux applications interactives. Ainsi, ils ont été employés pour la simulation d'actes chirurgicaux. Cotin et al. [CDA96] ont proposé un modèle volumique qui consiste en une décomposition de l'objet en tétraèdres. Ils s'appuient, pour cela, sur une représentation en maillage « simplex », qui présente une dualité avec la triangulation. Le modèle physique est basé sur l'utilisation des éléments finis appliqués à la théorie de l'élasticité. Le problème revient à déterminer le champ des déplacements de toutes les particules, situées au sommet des tétraèdres:

$$[K][U] = [F]$$

où $[K]$ est la matrice de raideur, $[U]$ est la matrice inconnue des déplacements et $[F]$ est la matrice des forces. Un tel modèle leur permet de reproduire différentes caractéristiques anatomiques, comme l'élasticité des tissus, le contact entre les organes ou les ligaments qui restreignent les mouvements. La simulation se fait en temps réel grâce à un modèle physique approprié et à une bonne représentation géométrique. Cependant, le modèle est simplifié, pour gagner en temps de calcul, les déformations sont uniquement élastiques.

2.2 Modèles non structurés

Les modèles d'objets déformables présentés au paragraphe précédent proposent des comportements diversifiés. Cependant, les déformations apportées au modèle doivent être de faible amplitude, puisque la structuration empêche, la plupart du temps, les objets de se fracturer complètement. Quelques modèles structurés autorisent les découpes, voire la séparation en plusieurs parties. Mais, les morceaux, même séparés, continuent de faire partie d'un seul objet, alors que l'on s'attendrait à avoir de nouveaux objets indépendants.

2.2.1 Surfaces implicites

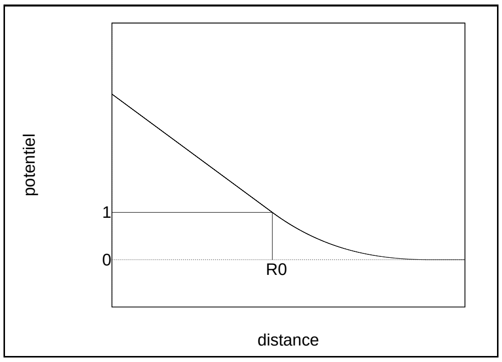
Dans [Gas93], une formulation implicite est présentée pour obtenir des surfaces de contact exactes lors de la collision de deux objets implicites. Cette formulation est valide pour des objets décrits par des squelettes S_i, associés avec des fonctions de potentiel f_i. La surface S de l'objet est alors donné par :

$$S = \{P \in \mathbb{R}^3 / f(P) = 1\} \text{ avec } f(P) = \sum_i f_i(P) \tag{2.5}$$

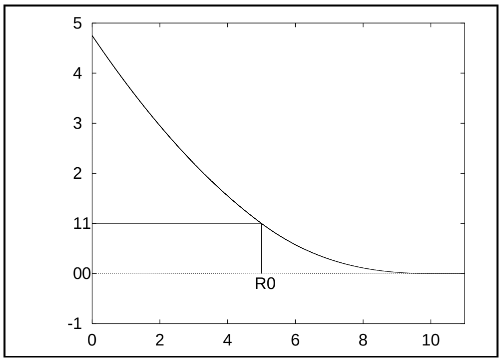
Les points internes à l'objet ont un potentiel supérieur à 1 et les points externes ont un potentiel inférieur à 1. Les squelettes sont n'importe quelle forme géométrique pour laquelle une distance à un point est définie. En un point de la surface de l'objet, la normale à la surface est donnée par le gradient de la fonction de potentiel. Les auteurs proposent de modéliser la raideur de l'objet avec la valeur de la dérivé de la fonction de potentiel. Pour un objet linéairement élastique, la dérivée de la fonction de potentiel doit être constante, alors que dans le cas d'un objet inélastique, la dérivée doit augmenter lorsqu'on se rapproche du squelette (Figure 2.2.1).

Lors d'une collision, la déformation des objets est réalisée en modifiant localement la fonction de potentiel les décrivant. Cette modification est réalisée sur un objet en ajoutant un potentiel de déformation calculé à partir de la fonction de potentiel de l'autre objet. Ce potentiel a pour but de créer une surface de contact exacte. Le potentiel de déformation ajouté au point P, de l'objet i entré en collision avec l'objet j est donné par :

$$g_{ji}(P) = 1 - f_j(P). \tag{2.6}$$



(a) fonction de potentiel d'un objet élastique



(b) fonction de potentiel d'un objet inélastique

De plus, pour avoir une animation visuellement réaliste, il est nécessaire de simuler la propagation des déformations sur les deux objets. Ceci est fait de manière purement géométrique en ajoutant un autre potentiel qui est défini en fonction de différents paramètres donnés par l'utilisateur. Après avoir déformé les objets, une force de réaction est calculée. Celle-ci est obligatoirement normale à la surface. Cette méthode donne un résultat visuellement correct, mais ne s'appuie pas sur des réalités physiques.

Cette méthode a été souvent utilisée en association avec un système de particules, les surfaces implicites étant utilisées pour avoir un meilleur rendu.

Dans [DG95], les auteurs ont développé une méthode d'échantillonnage leur permettant de contrôler les variations de volumes. Pour chaque squelette, un territoire leur est associé. Il suffit alors de calculer le volume de leur territoire pour détecter les variations de volume. Dans ce cas, la fonction de potentiel est localement modifiée. Afin de conserver les propriétés physiques stockées dans la fonction de potentiel (la raideur de l'objet est la dérivée de la fonction de potentiel), cette modification est une simple translation.

2.2.2 Systèmes de particules

De manière générale, une particule est représentée par une masse ponctuelle qui répond à un ensemble de lois de mouvement. Grâce à leur simplicité, les particules sont de loin les objets les plus faciles à utiliser. Cependant, elles permettent de modéliser une grande variété de comportement, et notamment celui d'objets très déformables sous contraintes. Ce modèle est celui que nous utilisons, il a donc fait l'objet d'une étude approfondie présentée au chapitre suivant.

2.3 Conclusion

Les modèles non structurés permettent de gérer la séparation et la réorganisation d'objets. Mais la perte de structure implique des problèmes de gestion de topologie, qui peut donc varier au cours de la simulation. Une autre difficulté est la constante variation du voisinage, ce qui complexifie de façon non négligeable les relations entre les éléments discrétisés. Un des seuls modèles permettant des variations importantes de topologie est le système de particules. À partir d'un élément très simple – la particule – le modèle offre des possibilités de comportement très variés et complexes. la seule variation des forces entre les particules permet toutes sortes de comportements physiques : élastique, inélastique, plastique, visqueux et même la séparation et la fusion. Les systèmes de particules proposent une vision unifiée des déformations, ce que ne permettaient pas les autres méthodes qui sont souvent optimisées pour une application précise. Ces dernières sont simplifiées pour pouvoir effectuer une simulation d'un certain type en un temps acceptable.

Ainsi, au vu des méthodes existantes, nous avons choisi le modèle particulaire pour représenter des objets très déformables.

3 Étude des systèmes de particules

3.1 Description des systèmes de particules

Un système de particules est constitué d'un ensemble de masses ponctuelles. Ces particules se déplacent sous l'action du monde extérieur, mais aussi du système lui-même. Les actions sont le plus souvent représentées par des forces, et les déplacement sont déduits à partir des lois de la dynamique. Ces forces externes agissent généralement de manière indépendante sur chaque particule. Les plus communes sont la gravité ou la collision avec un obstacle.

Au-delà de ce système simplifié, il est possible d'ajouter des interactions entre les particules. Selon le comportement simulé, ces forces internes peuvent agir de façon nominative entre deux ou quelques particules. C'est le cas par exemple des ressorts. Elles peuvent aussi avoir une action dépendant de la distance entre les particules. Dans ce cas, les particules sont dotées d'un rayon d'action. Ce peut être une simple répulsion pour simuler des matériaux granuleux. Ou alors, ce peut être une loi d'attraction à longue portée et de répulsion à courte portée pour maintenir la cohésion du système en le forçant à conserver un volume, ceci pour simuler un fluide ou une pâte.

3.2 Forces appliquées aux particules

Les particules possèdent toutes à peu près les mêmes caractéristiques géométriques. Par contre, la diversité des interactions entre particules et avec le monde extérieur, permet de modéliser des objets aux comportement variés. La même représentation d'un objet réagira différemment selon les forces employées. WITKIN [Wit92b] propose une classification des forces en fonction du nombre d'individus auxquelles elles s'appliquent. Chaque type de force est développé dans cette section.

3.2.1 Forces unaires

Les forces *unaires* sont les plus simples à mettre en œuvre. Elles sont principalement dues au milieu dans lequel évoluent les particules. Elles s'appliquent indépendemment sur chacune des particules du système, leur complexité est donc $O(n)$. Parmi elles, on retrouve la force de gravité $\vec{f} = m\vec{g}$ où $\vec{g}$ représente le vecteur gravité, qui est considéré comme constant dans la plupart des applications mais qui peut aussi dépendre de l'altitude par exemple. Le frottement visqueux est une autre forme de force unaire, dépendante de la vitesse. Elle peut s'écrire $\vec{f}_{frot} = -k\vec{v}$. Son effet est d'amortir le mouvement, en lui opposant une force proportionnelle à la vitesse. La particule atteindra graduellement l'équilibre en l'absence de forces extérieures. Pour modéliser un milieu sans frottement, il suffit d'annuler k. Par contre, en augmentant k, il est possible de simuler des déplacement dans des matériau de type magma.

3.2.2 Forces n-aires

On utilise couramment des forces n-aires, qui s'appliquent entre un nombre n fixé de particules. Dans les systèmes classiques masses-ressorts, les ressorts sont des forces entre deux masses. La force d'un ressort peut se représenter par l'équation suivante :

$$\vec{f} = -\left[k_r(r - r_0) + k_a\frac{\vec{v}\cdot\vec{r}}{r}\right]\frac{\vec{r}}{r}$$

où r_0 est la distance de repos du ressort, $\vec{r}$ est le vecteur entre les particules, et $\vec{v}$ est le vecteur vitesse relative. Le premier terme dépend de l'élongation et de la constante de raideur k_r. Le second terme exprime un amortissement en fonction de la vitesse entre les deux masses, avec un coefficient k_a. Cette force agit de façon réciproque sur chaque particule a et b, c.-à-d. $\vec{f}_a = \vec{f}_b$. L'amortissement agit différemment par rapport au frottement global, puisqu'il agit de manière symétrique sur les deux particules, sans avoir aucun effet sur le déplacement de leur centre de masse commun.

3.2.3 Forces d'interaction spatiale

Il existe des forces qui s'appliquent entre un nombre non fixé de particules. Ce sont les forces d'interaction spatiale. Les particules interagissent entre elles dès qu'elles sont proches, et cessent d'interagir quand elles s'éloignent. Ce type de force est souvent dérivé d'un potentiel, et comporte généralement deux termes : une répulsion à courte distance et une attraction à moyenne distance qui décroît pour s'annuler à longue distance. L'absence de voisinage fixe est très utile pour modéliser des changements de topologie, mais cela se fait à un coût élevé. Le calcul des forces est en $O(n^2)$. Il est cependant possible de réduire cette complexité en partitionnant l'espace.

Le potentiel de Lennard-Jones est très utilisé en animation. Il est issu de la cristallographie, pour modéliser les interactions d'attraction/répulsion entre les molécules. Il s'exprime en fonction de la distance r entre deux molécules :

$$\Phi(r) = \frac{A}{r^n} - \frac{B}{r^m}$$

A et B sont des constantes. Le premier terme correspond à une forte énergie de répulsion à courte distance, qui est due au principe d'exclusion de PAULI. Le second terme exprime l'attraction longue portée. Lennard-Jones a suggéré de prendre les valeurs $n = 12$ et $m = 6$ pour modéliser l'énergie entre deux molécules.

Quand deux particules sont en équilibre, le potentiel est minimal. La distance entre ces deux particules est alors égale à r_0. L'énergie requise pour dissocier les molécules est $\Phi(r_0) = \varepsilon$. Nous pouvons donc généraliser le potentiel de Lennard-Jones en fonction de ces valeurs à l'équilibre, ce qui donne une formulation très intuitive (Fig. 3.1).

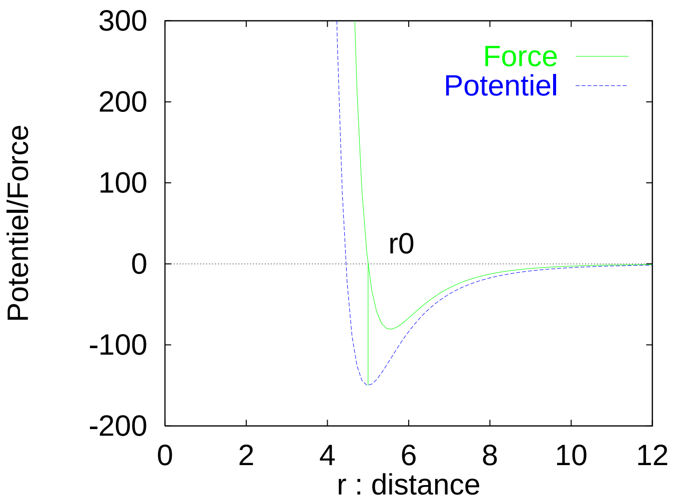


FIG. 3.1 – *Potentiel de Lennard-Jones*

$$\Phi(r) = \frac{-\varepsilon}{n-m}\left(m\left(\frac{r_0}{r}\right)^n - n\left(\frac{r_0}{r}\right)^m\right)$$

La force dérivée de ce potentiel peut s'écrire :

$$\vec{f}(r) = \overrightarrow{Grad}(\Phi(r))$$

soit sous une autre forme :

$$\vec{f}(r) = \frac{-mn\varepsilon}{(n-m)r_0}\left(\left(\frac{r_0}{r}\right)^{(n+1)} - \left(\frac{r_0}{r}\right)^{(m+1)}\right)\frac{\vec{r}}{||\vec{r}||} \tag{3.1}$$

Sous cette forme, la force est très facile à contrôler. Lorsque l'on fait varier n, c'est essentiellement le terme de répulsion qui est modifié. Si n augmente, les particules se repousseront plus, s'il diminue, elles se repousseront avec moins de vigueur. De la même façon, m permet d'agir sur l'attraction entre les particules. L'énergie de dissociation ε permet d'agir sur l'amplitude de la force. Pour sa part, r_0 permet de modifier le taux de recouvrement entre les particules. Si $r_0 = r_1 + r_2$, c.-à-d. qu'il est égal à la somme des rayons d'action des deux particules, alors celles si sont tangentes quand elles sont à l'équilibre. Par contre, si $r_0 < r_1 + r_2$, les particules pourront se chevaucher, ou si $r_0 > r_1 + r_2$, elles ne se toucheront pas lorsqu'elles sont à l'équilibre, ce qui permet de les forcer à occuper un plus grand espace.

Lors de l'animation, des oscillations peuvent apparaître entre deux particules quand elles sont séparées par une distance proche de la distance au repos. Pour éviter ce problème, une nouvelle force d'interaction a été proposée dans [LP95]. La principale modification par rapport à la force de Lennard-Jones est l'insertion d'un point d'inflexion au niveau de la distance au repos, pour limiter les oscillations. L'amplitude de cette force est donnée par :

$$F(r) = E_0 \frac{(r_0 - r)^3}{r} e^{-\alpha(r_0 - r)^2}$$

avec r_0 la distance au repos entre deux particules, E_0 un paramètre exprimant la raideur de l'objet (plus E_0 est grand, plus l'objet est raide) et α permet de contrôler la courbure. Sur la figure 3.2, nous avons diminué la courbure et nous avons ajusté la raideur pour obtenir une énergie de dissociation équivalente entre les courbes *Lombardo1* et *Lombardo3*. Autour

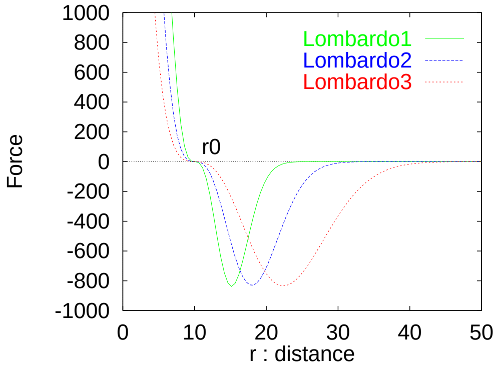


FIG. 3.2 – *Force de Lombardo pour différents paramètres*

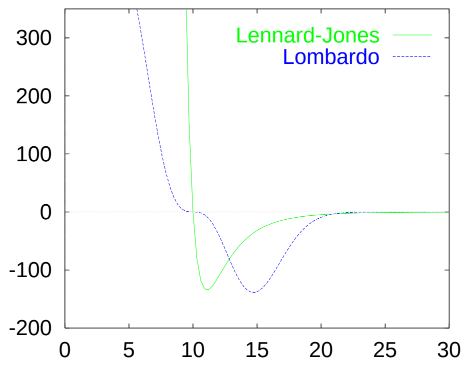


FIG. 3.3 – *Forces de Lennard-Jones et de Lombardo*

de la position d'équilibre, la tangente est nulle. La force exercée sur les particules est donc relativement moins importante par rapport au déplacement que pour Lennard-Jones. Ainsi les oscillations entre les particules sont réduites. Par contre, cette force est relativement difficile à manipuler, les paramètres étant beaucoup moins intuitifs que pour Lennard-Jones. La figure 3.3 montre une comparaison entre la force de Lennard-Jones et celle de Lombardo. L'énergie de cohésion n'est pas explicitement représentée dans la formulation, ce qui pose un problème lorsque l'on veut fixer l'amplitude maximale de la force d'attraction. D'autre part, une faible variation du paramètre α doit être corrigée par une grande variation de la raideur E_0 du fait de l'exponentielle.

3.3 Collision et contact avec des obstacles

3.3.1 Détection

Dans la plupart des cas, la détection des collisions entre des objets de type quelconque n'est pas évidente. Par contre, lorsqu'il s'agit de détecter une éventuelle collision entre un système de particules et un obstacle rigide, ce problème peut se ramener à un calcul d'intersection entre une sphère et une surface. Selon le modèle utilisé pour décrire l'obstacle, la détection de l'intersection peut se réduire à un calcul de distance. Si cette distance est supérieure au rayon de la particule, alors il n'y a pas de collision ; si elle est inférieure, il y a collision et si les deux valeurs sont égales, il y a contact.

Lorsqu'une collision est détectée, ce qui veut dire qu'une particule est à l'intérieur de l'obstacle, comme ceci n'est pas autorisé, il faut normalement déterminer l'instant précis où la particule est entrée en contact avec l'obstacle. Cela peut se faire par interpolation entre l'ancienne position et l'actuelle. Ensuite il faut ramener tout le système à l'instant du contact. Une autre méthode, plus approximative mais souvent largement suffisante, consiste à déplacer seulement le point qui est en collision.

3.3.2 Réponse

Pour décrire la réponse à la collision, la vitesse de la particule est décomposée en deux parties, une normale à la surface et l'autre tangente. Si $\vec{n}$ est le vecteur normal à la surface au point de collision, alors la composante normale de la vitesse est $\vec{v}_n = (\vec{n}.\vec{v})\vec{v}$ et la composante tangentielle est $\vec{v}_t = \vec{v} - \vec{v}_n$. Si l'on considère une collision élastique sans frottement, ce qui est le plus simple, il suffit d'inverser le signe de la composante normale de la vitesse pour obtenir la vitesse résultante $\vec{v}_r$, après collision. On obtient alors $\vec{v}_r = \vec{v}_t - \vec{v}_n$. Dans le cas d'une collision inélastique, la composante normale est multipliée par $-\rho$, ρ étant le coefficient de restitution. À $\rho = 0$, la particule ne rebondit pas du tout, à $\rho = 0.9$ c'est une super-balle, tandis qu'à $\rho = 1$, on retombe sur une collision totalement élastique. La figure 3.4 illustre le calcul de la vitesse résultante.

3.3.3 Cas du contact

Lorsque la particule est en collision avec une surface et si sa vitesse normale est nulle, alors la particule est en contact. Si la particule se rapproche de la surface, il faut corriger sa trajectoire pour qu'elle ne pénètre pas, mais qu'elle glisse le long de l'obstacle. On lui exerce alors une force qui annule exactement la composante normale de $\vec{f}$, soit $\vec{f_c} = -(\vec{n}.\vec{f})\vec{f}$. Par contre, si la particule s'éloigne du point de contact,il n'est pas besoin de corriger les forces qui lui sont appliquées, le contact sera rompu de lui-même. Il est possible d'ajouter un frottement au contact, qui doit agir tangentiellement à la surface. La force de frottement peut s'exprimer sous la forme $-k_f(-\vec{f}.\vec{n})\vec{v}_t$, son amplitude est proportionnelle à la force normale. Pour simuler une surface parfaitement non glissante, il suffit d'annuler $\vec{v}_t$. La figure 3.5 illustre le calcul de la force de contact.

3.4 Évolution des systèmes de particules

Depuis leur introduction par Reeves en 1983 [Ree83], les systèmes de particules ont été très largement utilisés. Leur succès grandissant peut s'expliquer par le fait de leur simplicité, mais aussi de leur capacité à modéliser des objets difficilement obtenus par les autres techniques existant en informatique graphique. Ils dépassent les limitations des modèles surfaciques classiques et permettent de créer des objets dont la surface n'est pas clairement définie, comme les nuages, la fumée ou l'eau. Il est aussi possible de modéliser des formes fractales à l'aide de systèmes de systèmes de particules.

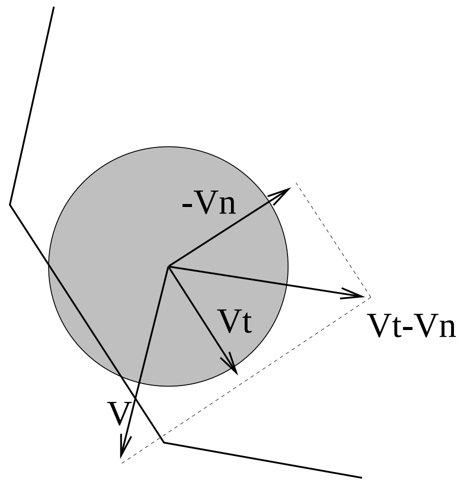


FIG. 3.4 – *Réponse à la collision*

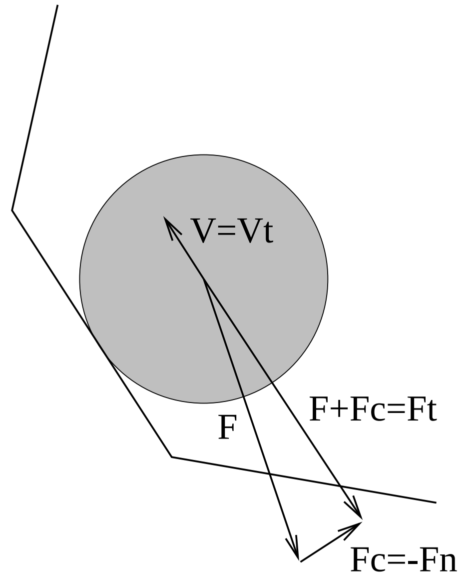


FIG. 3.5 – *Réponse au contact*

Les particules ne sont pas uniquement des objets statiques. Elles évoluent sous l'action des lois de la dynamique. Dans les premiers systèmes, le mouvement était gouverné par une approximation des lois de la physique. Mais des systèmes de plus en plus complexes sont apparus pour simuler des fluides ou des objets déformables.

3.4.1 Systèmes sans interactions

En 1983, REEVES [Ree83] introduit le premier système de particules pour la synthèse d'images. Le but était de proposer un modèle d'objets sans contours bien définis et qui supportent des changements dynamiques de forme et d'apparence. Une ensemble de particules forme un système qui représente l'objet. Les particules sont créées dans le système, elles bougent et se transforment, puis meurent. Elles sont générées par un processus stochastique . Leur nombre peut varier pour qu'un système s'accroisse ou décline, ou pour contrôler le niveau de détail. Pour paramétrer la simulation, des attributs décrivent la particule. Les principaux sont la position, la vitesse, la taille, la couleur, la forme, la durée de vie... Ces paramètres évoluent au cours du temps. Par exemple, la nouvelle position d'une particule est obtenue à partir de position translatée du vecteur vitesse. Les particules sont vues comme des points de lumière, ce qui permet de modéliser des explosions ou des feux d'artifices en combinant des systèmes de systèmes de particules. En affichant la trajectoire décrite par chaque particule, il est possible de générer des végétaux, qui peuvent être des touffes d'herbes ou des arbres plus complexes. Mais, la trajectoire des particules n'est pas conservée et les végétaux ainsi construits ne seront pas déformables. Les lois physiques sont assez grossières, le modèle est donc très simple. Par contre, il autorise le traitement d'un très grand nombre de particules.

3.4.2 Systèmes interagissants

Des lois physiques plus complexes ont été introduites dans les systèmes pour simuler les interactions entre les particules. Chaque particule réagit avec ses voisins et avec l'environnement. De cette manière, il est possible de simuler des comportements solides, fluides, granulaires ou gazeux.

Miller et Pearce [MP89] ont proposé une formulation générale de la force d'interaction entre particules voisines (Éq. 3.2) :

$$\vec{f} = \left(s_r \left(\frac{b_1}{||\vec{P}||^m} - \frac{b_2}{||\vec{P}||^n} \right) - s_d \frac{\vec{V} \cdot \vec{P}}{||\vec{P}||^2} \right) \vec{P} \tag{3.2}$$

où $\vec{P}$ et $\vec{V}$ sont respectivement les vecteurs position et vitesse relatives des particules. Le premier terme exprime une force d'attraction/répulsion dont m et n sont des coefficients de forme, alors que le second terme représente l'amortissement. En faisant varier les coefficients s_r et s_d, il est possible de simuler différents comportements. Une simple répulsion inélastique créera un matériau de type poudre. En augmentant le frottement, les particules auront tendance à s'agglutiner mais peuvent toujours se séparer facilement. On obtient un matériau fluide. Et si l'on ajoute une attraction de moyenne portée, les particules restent en contact prolongé pour former un solide plus ou moins compressible. Une valeur de température permet de passer de manière continue d'un comportement à l'autre. Des interactions avec le monde environnant sont introduites pour prendre en compte les collisions avec les obstacles. Les nouvelles positions sont calculées selon la méthode d'intégration d'Euler (cf. annexe C).

La notion de température a aussi été développée par Terzopoulos et al. [Ter89] pour modéliser des objets déformables capables de conduire la chaleur, et donc de fondre. Les mouvements dynamiques sont régis par les équations de Lagrange.

$$\mu \frac{\partial^2 \vec{P}}{\partial t^2} + \gamma \frac{\partial \vec{P}}{\partial t} + \delta_{\vec{P}} \varepsilon = \vec{F} \tag{3.3}$$

La position $\vec{P}$ est calculée à partir des la force $\vec{F}$ qui dépend de la force d'inertie due à la densité de masse μ, du frottement dû à la densité d'amortissement γ, et de la force élastique. Quant aux transfert thermiques, ils dépendent des équations de chaleur dans les milieux non-homogènes.

$$\frac{\partial}{\partial t}(\mu \sigma \theta) - \nabla \cdot (C \nabla \theta) = q \tag{3.4}$$

où q exprime la quantité de chaleur par unité de volume en fonction de la distribution de chaleur θ et de la matrice de conductivité thermique C. μ est la densité de masse et σ la température. ∇^2 représente le Laplacien. À l'état solide, l'objet est discrétisé par un maillage hexaèdrique composé de masses reliées par des connections thermo-élastiques. La raideur augmente quand la température baisse, et la connection commence à fondre quand la température dépasse le seuil de fusion. Elle est alors remplacée par des interactions de type fluide, une répulsion courte portée et une attraction longue portée qui décroît quand la température augmente, jusqu'à disparaître.

De la même façon, Tonnesen [Ton92] prolonge les travaux précédents et présente un modèle unifié pour les comportements liquides et solides en fonction de la température. Le potentiel appliqué entre deux particules est composé d'un terme d'attraction/répulsion plus une énergie thermique. Ils modifient l'équation générale de Lennard-Jones pour créer une famille de courbes d'énergie de potentiel inter-particule :

$$\Phi_\psi(r) = \frac{-\varepsilon_m + \psi}{n - m}\left(m\left(\frac{r_0}{r}\right)^n - n\left(\frac{r_0}{r}\right)^m\right) \tag{3.5}$$

où n et m sont les coefficients de forme de l'équation de Lennard-Jones, ε_m est un coefficient constant d'attraction/répulsion, et ψ est l'énergie thermique de dissociation. Ils définissent aussi un modèle discret du transfert thermique $\psi_i = \mu_i \sigma_i \theta_i$ entre deux particules i et j :

$$\frac{\psi_i(t+dt) - \psi_i(t)}{dt} = \sum_j \frac{\frac{1}{n}k_{ij}(\theta_i - \theta_j)}{\frac{1}{4}r_{ij}^2} \tag{3.6}$$

où k_{ij} est la conductivité thermique et r_{ij} la distance entre chaque paire de particules. Enfin, ils introduisent une force de collision qui est similaire à un potentiel de répulsion. Ainsi, ils obtiennent un modèle d'objets déformables ou fluides en fonction de la température, dont les mouvements sont produits à partir de la dynamique moléculaire et Newtonienne. Ce modèle devrait aussi permettre d'introduire de la friction, puis de convertir les frottements en perte de chaleur.

Reynolds [Rey97] propose de remplacer le potentiel standard de Lennard-Jones par un potentiel moins connu, qui est dû à Boscovich en 1765. Il s'agit d'un modèle possédant plusieurs points de repos. L'objet a donc un comportement élastique tant que la force exercée n'est pas trop importante, qui devient plastique dès qu'un des seuils de rupture est atteint. Les particules s'écartent en passant d'un point d'équilibre à un autre. Le nombre de ces point étant fini, l'objet est susceptible de se fracturer. Des solutions sont proposées pour répondre aux problèmes délicats de la fatigue du matériau et de la friction. Un des problèmes pour utiliser ce potentiel est qu'il n'existe pas de formulation mathématique, Boscovich n'ayant donné que des solution géométriques. L'auteur utilise les splines pour représenter cette force, et il inclut un palier de stabilité pour chaque position d'équilibre de la même façon que dans [Lom96]. La figure 3.6 illustre la fonction utilisée pour le calcul des forces d'interaction entre les particules.

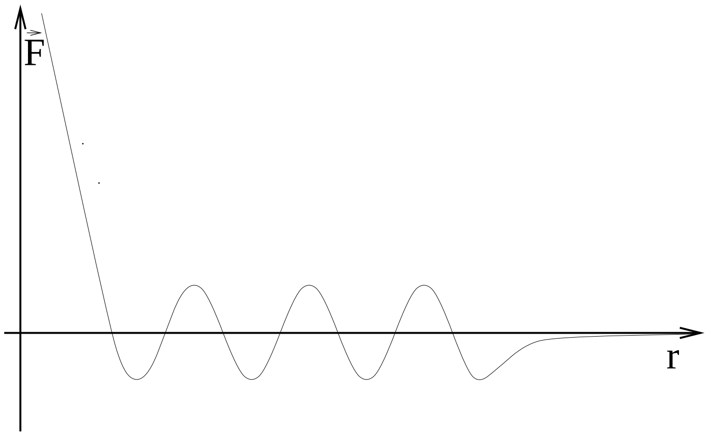


Fig. 3.6 – *Force dérivée du potentiel de* Boscovich

3.4.3 Modèles de mouvement

WEJCHERT et HAUMANN [WH91] ont développé un modèle aérodynamique avec contrôle du mouvement dans des flux en deux parties : le flux, appliqué à tout l'espace, et localement la frontière de l'objet. Cela permet de simuler des objets en mouvement dans l'air ou dans l'eau. Ils définissent quatre types de flux :

- uniforme, les courbes de vitesse sont parallèles ;
- concentrationnaire, les lignes se rejoignent en un seul point ;
- expansif, toutes les lignes partent d'un même point ;
- vortex, les lignes sont concentriques.

La combinaison de ces primitives par simple somme permet de créer des flux beaucoup plus complexes. Pour les obstacles, ils créent des champs très forts, ce qui est plus simple que des collisions, mais moins précis surtout en cas de contact prolongé. La force exercée par un flux sur une particule est donnée par la formule de Stocke $\vec{F} = 6\pi a\eta\vec{V}$. Si la viscosité η est importante, la particule suivra précisément le flux, si elle est nulle, elle ne sera pas affectée par ce flux. Pour obtenir de rotations, les particules sont connectées entre elles par des ressorts, et les flux s'exercent sur la surface entre les ressorts. La force subie par un objet dans un flux sera donc de la forme $\vec{F} = \rho A\vec{V}$, où ρ est la densité et A l'aire de l'objet. Ce modèle est utilisé pour simuler le déplacement sous l'action du vent de feuilles modélisées par un système masses-ressorts. À long terme, les auteurs pensent pouvoir simuler le vol aérodynamique des avions et des oiseaux.

MA et SMITH [MS95] ont développé un modèle pour la visualisation des processus de dispersion stochastique et de mélange des gaz. Le mélange des particules sans masse ou inertes est visualisé par la transformation du champ de vecteurs d'une référence Eulerienne vers une référence Lagrangienne. Des ensembles de particules sont suivis dans le champ de vecteurs par leur chemin moyen mais aussi par leur dispersion statistique, définie à l'aide d'informations scalaires ajoutées.

3.4.4 Particules orientées

En 1991, SZELISKI et TONNESEN [ST91, ST92] ont introduit un nouveau type de système de particules : les particules orientées. Sans intervention extérieure, les particules classiques ont tendance à s'agglutiner sous forme de solide plutôt que de surface. Les auteurs ont donc défini des particules représentées par un élément de surface (et non un point) et par une normale. Ils ont aussi défini de nouvelles forces de potentiel pour obliger les particules à former des arrangements surfaciques. Les notations sont illustrées par la figure 3.7. $\Psi(r)$ est une fonction monotone décroissante qui limite la portée de la force.

- Le potentiel de co-planarité garde les particules dans le même plan :

$$\Phi_P = (\vec{n}_i.\vec{r})^2\Psi(r)$$

- Le potentiel de co-normalité contrôle la torsion de la surface :

$$\Phi_N = ||\vec{n}_i - \vec{n}_j||^2\Psi(r)$$

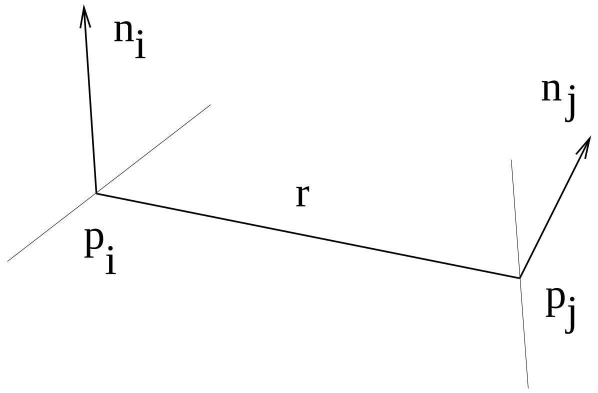


FIG. 3.7 – *Particules orientées*

– Pour obtenir des surfaces de courbure constante, le potentiel de co-circularité :

$$\Phi_C = ((\vec{n}_i + \vec{n}_j).\vec{r})^2\, \Psi(r)$$

De nombreuses opérations sur les surfaces sont simplifiées par l'emploi des particules orientées. Des objets 3D peuvent être créés et manipulés interactivement grâce à la définition d'outils d'étirement, de découpe ou de modélisation de forme. La portée des interactions est limitée par une notion de dégel des particules proches de l'outil et de gel lorsque celui-ci s'éloigne. Il est aussi possible de reconstruire automatiquement des surfaces sur un ensemble de points. Une interpolation est obtenue en faisant croître un ensemble de particules sur la surface.

En généralisant cette notion de particules orientées,LOMBARDO [Lom96] a proposé de nouvelles d'interactions, afin de modéliser des objets déformables et dynamiques, contrairement à ceux produits par le modeleur de SZELISKI et TONNESEN qui sont statiques. Il a notamment réécrit le potentiel co-circulaire, pour obtenir une position de repos, lorsque les particules sont situées sur une sphère de rayon R :

$$\Phi_C = ||R(\vec{n}_i - \vec{n}_j) + \vec{r}||^2 \Psi(r)$$

Cela permet de modéliser des objets à mémoire de forme. Par contre, ils doivent être composés de morceaux de plans ou de sphères. Pour pouvoir modéliser des comportements complexes, deux classes de particules sont définies : charnière ou standard. Le principe est de subdiviser l'objet en parties reliées par des charnières, qui sont limitées à des interactions de type attraction/répulsion. Ce modèle est appliqué à la modélisation de muscles qui se contractent. La conservation du volume du muscle est obtenue en faisant varier la distance de repos entre les particules. Les particules orientées sont aussi utilisées pour reconstruire des objets à partir de données 3D. Un échantillonnage est obtenu en générant automatiquement des particules proches de la surface. L'application des potentiels permet de positionner régulièrement des particules sur la surface de l'objet, notamment en alignant la normale de la particule avec celle de la surface. La répartition uniforme est obtenue facilement est ajoutant un potentiel de type attraction/répulsion entre les particules.

3.4.5 Réseaux masses-ressorts

3.4.5.1 Présentation du modèle

Les systèmes masses-ressorts sont un cas un peu particulier des systèmes de particules. Une particule est simplement reliée par des ressorts aux particules de son voisinage. Cette représentation est très utilisée pour la modélisation des tissus ou des surfaces déformables en général [Hou92]. Certain modèles sont capables de gérer des contraintes d'auto-intersection [WTMT95], ou d'adapter les liaisons en fonction des collisions [HPH96]. Les réseaux masses-ressorts peuvent aussi être utilisés pour modéliser des volumes. Ils sont très souvent dérivés d'une triangulation ou d'une tétraèdrisation de l'objet déformable. Il sont aussi utilisés en complément avec d'autres méthodes de modélisation, qui exploitent la simplicité de ces réseaux [Gas93, MC96].

3.4.5.2 Comparaison entre systèmes de particules et réseaux masses-ressorts

La principale caractéristique, outre leur simplicité, des systèmes à particules est l'absence de topologie fixe. À l'intérieur du système, chaque particule peut interagir avec toutes les autres. Deux particules en interaction à un moment donné, peuvent tout à fait la rompre cette interaction à l'instant suivant, et la retrouver un peu plus tard. Cela permet à un objet de subir des déchirures, de se fractionner en plusieurs morceaux et même de se recoller au cours de la simulation. Pour le cas particulier des systèmes masses-ressorts, il est tout à fait possible de gérer ces fractures, il suffit de couper le ressort quand la tension est trop forte par exemple. Le recollement est plus problématique puisqu'il faut tester tous les couples de masses. Par contre, le recollement n'est pas toujours souhaitable. Dans le cas d'une fracture, nous attendons des morceaux qu'ils se comportent comme des objets séparés et non pas comme des parties d'un seul objet.

Le voisinage d'une particule n'est en général pas fixé. Le calcul des forces internes du système est donc en $O(n^2)$ si n est le nombre de particules. De nombreuses méthodes d'optimisation existent. Nous citerons notamment la limitation de la portée des interactions (deux particules trop éloignées ne s'attirent plus), ce qui permet de limiter la recherche à un espace restreint autour de chaque particule, en partitionnant l'espace. Il est évident que l'utilisation des ressorts permet de réduire la complexité du calcul des forces, puisque chaque particule n'agit qu'avec celles auxquelles elle est connectée. Mais, il faut décrire la connection pour chaque couple de masses, ce qui conduit à une occupation mémoire en $O(n^2)$. Ainsi, au niveau complexité, les systèmes de particules optimisés et les réseaux masses-ressorts sont à peu près équivalents, même si l'utilisation des systèmes de particules est plus difficile.

L'emploi de modèles hybrides peut être un avantage. Le modèle particulaire développé dans le système CORDIS/ANIMA [Jim93] est un réseau masses-ressorts, avec des connections pouvant évoluer au cours du temps en fonction par exemple de la distance entre les masses. Cela permet un plus grand nombre de comportement puisque des ressorts peuvent devenir actifs ou inactifs.

3.4.6 Systèmes en couches

Un des gros problèmes des systèmes de particules est le temps de calcul. La réduction du nombre de particules, ainsi que celui des interactions entre les particules et avec le monde extérieur est un moyen efficace pour réduire les coûts de simulation. Mais, ces diminutions ne doivent pas aboutir à une perte de précision sur le modèle, ni sur le réalisme du comportement.

Jimenez [Jim93] a proposé un modèle structuré en couches, qui ont chacune leurs spécificités (Fig. 3.8). Une particule est une masse ponctuelle. Pour faciliter les interactions entre les particules et pour les forcer à occuper un espace non vide, elles sont aussi munies d'un rayon d'action sphérique, qui correspond à la zone de non interpénétration.

- Un noyau, composé de grosses particules. Il est plutôt peu déformable. Il donne l'allure générale et le comportement global de l'objet ;
- Un derme, qui définit les propriétés de déformabilité de l'objet. Il est composé de particules de rayon moins important que celles du noyau ;
- Un épiderme, composé de petites particules qui assurent la cohésion en exerçant une tension superficielle. Il est chargé des interactions avec l'extérieur.

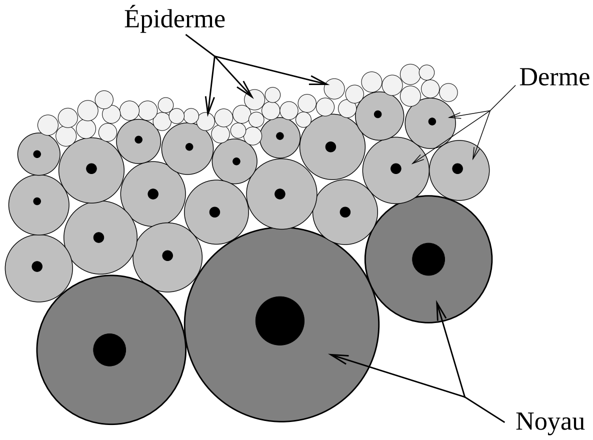


Fig. 3.8 – *Structuration des particules en couches*

La représentation en couches, non nécessairement toutes présentes, permet de structurer les interactions internes et externes, et d'en réduire le nombre. En effet, les particules d'une couche seront en interaction avec celles de la même couche et celles des couches adjacentes. Ainsi, le noyau ne sera jamais en interaction avec l'épiderme. Dans le cas d'interactions entre objets, seules les particules de l'épiderme (lorsqu'il est présent), sont concernées. Lors d'une collision ou d'un contact, un point virtuel est ajouté sur chaque surface, au niveau des points de contacts. Les forces seront alors calculées sur ce point qui disparaîtra dès que les objets s'éloigneront.

Cela permet aussi de réduire le nombre de particules. À la périphérie de l'objet, qui assure la majorité des interactions, une grande précision est requise. La proximité géométrique du modèle à l'objet dépend essentiellement de la qualité de l'épiderme. Par contre, au centre, qui est en général peu déformable, de grosses particules sont largement suffisantes pour donner l'allure de l'objet. Pour un objet rigide, le nombre de particules sera celui nécessaire pour reconstruire l'objet avec la précision donnée. À l'inverse, un objet totalement déformable et capable de se fracturer sera composé de particules qui seront toutes concernées par les interactions. Il serait donc intéressant, en plus de la structuration en couches, d'avoir un modèle adaptatif : de grosses particules lorsque l'objet n'est pas soumis aux actions extérieures, qui seraient raffinées localement pour répondre à la déformation.

Cette structuration a été implantée dans le système CORDIS/ANIMA, qui est un modeleur/simulateur d'objets physiques audibles, visibles et manipulables, développé à l'ACROE (Association pour la Création et la Recherche sur les Outils d'Expressions). Dans ce système, tous les objets sont définis à l'aide de points matériels connectés par des éléments de liaison. De nombreux comportements sont modélisés par la combinaison de liaisons définies à partir de modèles rhéologiques simples (élastique, visqueux et plastique). Une liaison conditionnelle permet de modéliser des changements d'états, de la même manière que les liaisons thermiques existant dans d'autres modèles. Il est possible d'imposer un comportement similaire à un ensemble de particules, grâce à la notion d'agglomérats, qui sont des points matériels auxquels est associé un module d'interaction mutuelle. Le plus souvent, il s'agit d'une force de type attraction/répulsion.

Toujours au sein du système CORDIS/ANIMA, Luciani et al. [LHM95] ont développé une modélisation multi-échelle de matériaux granulaires. Les phénomènes typiques des matériaux granulaires tels que le tas, l'arche, ou l'avalanche, sont réalisés à partir de modèles très simples : les réseaux de connections non-linéaires visco-élastiques et de ressorts avec frottement. Pour les autres phénomènes, comme le flux, ou pour la visualisation, une discrétisation très fine est requise. L'utilisation de deux modèles physiques permet d'éviter la discrétisation fine des phénomènes non-linéaires. Chaque grain de niveau intermédiaire est remplacé par un ensemble de grains plus fins, dont le nombre et la forme dépendent du mouvement de l'ensemble. L'espace est discrétisé en une grille de *phyxel* (pour « physical element »), qui sont caractérisés par une variable dynamique dépendant des coordonnées de tous les grains du niveau intermédiaire à l'instant t. La simulation des comportements est basée sur une modélisation très précise des phénomènes dynamiques, des oscillations et même des micro-oscillations durant le contact. Cela permet d'analyser les propriétés de ce type de matériau et d'en comprendre les causes.

Meseure et Chaillou [MC96] ont proposé un modèle dynamique de type masses-ressorts, qui s'appuie sur l'idée d'un corps déformable ayant une composante rigide. Les deux composantes sont liées lors d'un déplacement quelconque, mais elles sont dissociées dès qu'il y a déformation. La composante rigide donne le comportement volumique du maillage surfacique. Pour gagner du temps lors de l'animation, un minimum de masses décrit l'objet initial et de nouvelles masses sont ajoutées lorsque cela est nécessaire (sans pour autant perturber la dynamique du système). Pour simuler des découpes, lorsque les forces deviennent trop importantes en un point, celui-ci est dupliqué ainsi que les ressorts qui lui sont attachés. L'animation s'effectue en faisant un bilan des forces extérieures et internes (dues aux ressorts), et en intégrant les équations de la dynamique à l'aide de la méthode d'Euler.

Desbrun [Des97] a lui aussi défini un modèle composite. Le modèle particulaire est le plus adapté pour la modélisation d'objets déformables, il a donc logiquement été choisi pour former l'ossature. Il s'agit ici d'un ensemble de particules munies d'une loi d'attraction/répulsion de type Lennard-Jones. Cette première couche est enrobée par une couche de matière implicite, qui correspond au derme et à l'épiderme du modèle précédent. Les particules sont utilisées comme squelettes pour générer la surface implicite. Les propriétés élastiques et de tension de la peau lui permettent de gérer finement les contacts et les déformations locales. Les surfaces implicites sont capables de se fracturer, lorsque les squelettes se séparent. Ainsi les changements de topologie peuvent être pris en compte grâce au système de particules et aux surface implicites.

D'autre part, il a proposé un modèle adaptatif pour contrôler le nombre de particules au cours de la simulation. L'idée est de raffiner le modèle à l'endroit et au moment où l'on en a le plus besoin. Les particules sont subdivisées en un ensemble de plus petites particules, lorsque la différence de pression avec une de leurs voisines devient trop importante par rapport au volume.

$$\frac{|\rho_i - \rho_j|}{v_i} > seuil$$

où ρ désigne la pression et v le volume de la particule. Pendant cette opération, il est nécessaire de conserver un certain nombre de propriétés comme le volume couvert et la masse. À l'inverse, lorsqu'une région est stable, il faut économiser le nombre de particules et les regrouper en une seule grosse, selon un critère de densité associé à un critère de sphéricité. Si la détermination des particules à décomposer est immédiat, la recherche d'un groupe sphérique de particules stables est beaucoup plus délicat. L'adaptation automatique du modèle dans le temps et dans l'espace résout le problème du choix de la résolution, puisque la précision du modèle dépend de son environnement.

3.4.7 Applications diverses

Depuis leur création pour modéliser des explosions et d'autres phénomènes similaires, les systèmes de particules ont été employés pour des applications très variées. Même si l'application principale est l'animation d'objets déformables en informatique graphique, il existe de nombreux autres domaines d'utilisation.

Ainsi, les particules sont utilisées pour échantillonner des surfaces implicites. WITKIN et HECKBERT [WH94] se servent de contraintes pour forcer les particules à rester sur la surface pendant que les deux sont en mouvement. Le système est défini par les vitesses plutôt que par les positions et son comportement est gouverné par des équations différentielles de la vitesse par rapport au temps. Ils peuvent ainsi contrôler la forme de la surface en déplaçant les particules de façon interactive, qui agissent comme des points de contrôle. Pour l'échantillonnage, ils procèdent de manière différente. Les particules peuvent se déplacer librement le long de la surface. Un échantillonnage uniforme peut être obtenu par simple répulsion des particules. En fonction de la densité locale, les particules fusionnent ou se divisent jusqu'à obtenir un équilibre. La méthode proposée permet de visualiser et de déformer rapidement les surfaces implicites, mais elle ne n'est pas restreinte aux surfaces algébriques comme beaucoup d'autres méthodes. Une autre de ces applications peut être la création d'un maillage polygonal de la surface, même si cela nécessite de connaître la topologie. Un reproche que l'on peut faire à cette méthode est que l'échantillonnage est uniforme. Il serait intéressant d'ajouter des critères liés à la courbure par exemple, pour contrôler la densité.

De la même façon, FIGUEIREDO et GOMES [FG96] se servent des particules pour produire un échantillonnage des surfaces implicites. L'étude des trajectoires d'un ensemble de particules placées aléatoirement dans un champ de potentiel permet d'éviter la fastidieuse phase de recherche de points de la surface par un parcours de tout l'espace. La simplicité du modèle (les particules suivent les lignes de gradient) permet d'obtenir rapidement un échantillonnage, malheureusement assez grossier, des surfaces.

Cependant, les deux méthodes précédentes sont restreintes aux surfaces. Elles n'utilisent pas complètement les propriétés des particules qui sont naturellement adaptées à la modélisation de volumes. Un des rares travaux de construction automatique d'objets volumiques a été présenté par SHIMADA et GOSSARD [SG95]. Ils ont remarqué qu'un empilement de sphères était étroitement lié à un diagramme de Voronoï. La construction d'un maillage triangulé est décomposée en deux étapes : un remplissage avec des sphères suivie d'une triangulation en connectant les centres de ces sphères. La méthode, appelée « bubble-meshing », traite des représentations non-manifold en remplissant l'objet par ordre croissant de dimension : sommets, arêtes, faces et enfin volumes. Le processus est identique pour chaque dimension. Des particules sont initialement placées par subdivision hiérarchique. Une force de type attraction/répulsion est définie pour obtenir un maillage régulier. Cette force est utilisée lors de la simulation dynamique, en plus des forces qui contraignent les sphères à rester sur l'objet. De la même manière que dans la méthode présentée dans la référence [WH94], la densité des sphères est contrôlée par un critère dépendant du taux de recouvrement des sphères :

$$\alpha_i = \frac{2}{R_i} \sum_{j=0}^{n} \left(R_i + \frac{R_j}{2} - d_{ij} \right)$$

où R_i représente le rayon d'influence et d_{ij} la distance entre les particules i et j. Pour une courbe, le taux de recouvrement optimal est $\alpha = 2$, il est de $\alpha = 6$ pour un surface et de $\alpha = 12$ pour un volume. Ces valeurs correspondent au nombre de voisins de chaque sphère. En cas de valeur inférieure, la densité est faible autour de la sphère, et il faut créer de nouveaux voisins. Si la valeur est plus élevée, la particule est simplement détruite. Cela permet d'obtenir un partitionnement de l'objet par connexion des centres des sphères. La qualité du maillage finale est contrôlée par deux critères : l'un est topologique et dépend du nombre moyen de voisins, l'autre est géométrique et tend vers zéro si les mailles sont équilatérales.

3.5 Limitations des méthodes actuelles

Il existe plusieurs techniques de modélisation d'objets déformables à l'aide de particules. Elles peuvent être classifiée en deux catégories. Dans la première, nous retrouvons les méthodes de modélisation de surface. C'est le cas des particules orientées ou de l'échantillonnage. Des forces ont été ajoutées pour obliger les particules à s'arranger sous forme de surfaces. Les propriétés physiques de l'objet doivent donc être obligatoirement décrites sur la surface ; tension superficielle, déformation à surface constante... Si nous voulons obtenir des comportements qui seraient ceux d'un objet « plein », il faut passer à une reconstruction volumique. Pour cela, il faut enlever les forces qui maintiennent les particules sur la surface pour les autoriser à remplir tout le volume. Mais, dans une telle éventualité, tous les avantages liés aux surfaces seraient perdus.

Une des solution est de garder la surface (la peau) et de remplir l'intérieur par des particules. Une généralisation d'une telle technique peut être la structuration en couches proposée par Jimenez. Une évolution des systèmes à couches est le modèle hybride combinant les particules et les surfaces implicites, modèle défini par Desbrun. Ces deux techniques présentent des propriétés tout à fait intéressantes. Chaque couche a une fonction bien particulière, ce qui autorise une grande variété de comportement physiques de l'objet. Le nombre restreint de données permet d'animer les objets dans un temps correct par rapport à un modèle mono-couche : la réduction du nombre de particules entraîne une diminution du nombre d'interactions internes mais aussi externes.

Cependant, la création d'objets à l'aide de ces modèles se fait jusqu'à présent de manière interactive. Il n'y a pas de méthode automatique de reconstruction d'un volume défini par ses propres frontières surfaciques. La méthode de « bubble-meshing » peut à la limite être considérée comme produisant un modèle à deux couches, puisque la surface est remplie indépendemment de son intérieur. D'autre part, l'initialisation se fait à partir d'une subdivision hiérarchique, ce qui permet d'avoir de grosses particules dans les zones uniformes et de plus petites ailleurs. Cependant, au final, les couches ne sont pas indépendantes et n'ont pas un rôle bien séparé. Ce n'était d'ailleurs pas le but de cette reconstruction qui était destinée à fournir un maillage pour des objets solides et non pas des corps déformables.

Les limitations de ces méthodes nous ont amené à créer une nouvelle méthode de reconstruction d'objets déformables et susceptibles de se fracturer, qui reprenne les avantages de chacune des méthodes précédentes. L'idée est très simple : il s'agit de remplir un volume défini par ses frontières à l'aide de plusieurs couches de particules. Chaque couche est générée successivement ce qui réduit le temps de la reconstruction. Les outils de simulation dynamique des particules peuvent être ensuite exploités pour l'animation et la déformation des modèles. Cela présente l'intérêt de n'avoir qu'un seul modèle d'objet muni de lois dynamique qui est unique pour la reconstruction et l'animation. Il est alors possible de modifier le modèle au cours de l'animation, par exemple recréer une peau dans les zones de fractures.

4 Reconstruction à l'aide de systèmes de particules

4.1 Problématique de la reconstruction d'objets déformables

Comme nous l'avons vu aux chapitres 2 et 3, les systèmes de particules sont des modèles très intéressants pour représenter et animer les objets susceptibles de se déformer. Cependant, dans la grande majorité des travaux présentés à ce jour, les objets sont supposés déjà existants, ou alors ils sont modélisés de façon interactive. Très peu de travaux font référence à la façon dont sont construits leurs modèles. Nous pouvons citer la reconstruction d'une surface échantillonnée à l'aide d'un système de particules orientées [ST92], ou alors, les méthodes d'échantillonnage de surfaces implicites [Tur92, WH94]. Les méthodes précédemment citées ne s'appliquent qu'aux cas des surfaces.

Dans le cas des volumes, le « bubble-meshing » de SHIMADA et la méthode décrite par LOMBARDO sont assez intéressantes. Nous nous sommes inspirés de ces méthodes pour proposer une nouvelle approche de la reconstruction d'objets déformables. Nous désirons obtenir un échantillonnage du volume délimité par un ensemble de surfaces fermées, dont les frontières sont connues. Pour être le plus général possible, nous ne faisons pas de supposition sur la topologie de l'objet à reconstruire. Ce volume peut être composé de plusieurs morceaux distincts, peut comporter des trous et des îlots, et peut se séparer en un nombre quelconque de branches. La méthode doit pouvoir s'appliquer à tous les objets comportant un extérieur et un intérieur (nous excluons donc les objets du type bouteille de KLEIN).

Nous voulons obtenir une représentation de ces frontières par un système de particules. Il serait logique d'utiliser les particules orientées qui génèrent une surface facilement déformable. Cependant, en animation, les surfaces ne sont pas suffisantes pour décrire correctement un certain nombre de comportements. Ainsi, lorsque l'on exerce une force sur une surface, la résistance à cette déformation ne pourra pas être traitée correctement uniquement à l'aide de la tension superficielle. Il est nécessaire d'avoir une structure interne qui assure le réalisme des déformations en profondeur. Cela peut être un système de ressorts entre un squelette et une surface implicite [Gas93], mais dans ce cas la résistance sera uniquement normale à la surface. Une amélioration est de réaliser un réseau interne tridimensionnel de masses-ressorts, plusieurs ressorts d'orientations différentes contribuent alors à la force de réaction [PBP96]. Mais, même ainsi, il est difficile de conserver le volume de l'objet. Une solution est de remplir l'espace interne par des particules. Lors de déformations, celles-ci résisteront à l'altération de la forme. La préservation du volume est obtenue en empêchant les particules de se superposer.

Afin de pouvoir mieux expliquer notre démarche de reconstruction, nous commençons par un premier modèle mono-couche très simple. Ensuite, pour améliorer les performances (cf. paragraphe 3.4.6), nous généralisons cette démarche mono-couche vers un système multi-couche.

4.2 Description de l'algorithme

Le principe de la reconstruction est de remplir le volume de l'objet par un ensemble de particules dynamiques. Nous initialisons un ensemble de particules à l'intérieur de l'objet, et de nouvelles particules sont créées autour des particules existantes. Toutes ces particules sont placées dans un environnement muni de lois physiques, ce qui va permettre au particules de parvenir à un état stable (sans mouvement). Les particules permettent d'obtenir facilement un échantillonnage uniforme de l'objet. Pour cela, nous définissons une loi de type attraction/répulsion entre les particules [Tur92, WH94]. Ainsi, les particules se repousseront et se répartiront uniformément en fonction de la zone d'influence de la force. Dans tout ce qui suit, par abus de langage, nous confondrons la limite de la zone d'influence avec le rayon de la particule.

Les grandes lignes de l'algorithme sont reprises dans le schéma suivant :

Algorithme 4.1 Algorithme de reconstruction

1 Définir les paramètres de la reconstruction, qui sont essentiellement les caractéristiques des particules et les forces qui leur sont appliquées
2 Créer régulièrement les particules initiales
répéter pour l'ensemble des particules
3   Détecter les collisions avec les obstacles de la scène, ou entre les différents systèmes de particules
4   Déterminer les forces externes et internes aux systèmes de particules
5   Utiliser les lois de la dynamique pour intégrer l'accélération et calculer les nouvelles positions des particules
6   Générer de nouvelles particules autour des particules déjà existantes
jusqu'à ce que le volume soit rempli (et le système soit à l'équilibre)

4.3 Génération des particules

Dans cette partie, nous décrivons la méthode de génération de particules, qui constitue le cœur de l'algorithme de reconstruction. Nous distinguons le cas bidimensionnel de celui des volumes. Pour des raisons de simplicité et dans un souci de clarification, nous présentons d'abord la reconstruction en deux dimensions, et proposons quelques améliorations de la méthode de base. Ensuite, nous montrons l'extension de la méthode au cas tridimensionnel.

4.3.1 Cas bidimensionnel

En deux dimensions, la forme est décrite par un ensemble de contours fermés. Les frontières peuvent être décrites indifféremment à l'aide de n'importe quel modèle de courbe planaire, (polygonal, paramétrique ou implicite). Nous désirons obtenir une partition uniforme de la surface.

Pour cela, nous créons des particules à l'intérieur de chacun des morceaux de la forme. Ces particules serviront de germe. Nous avons défini un ensemble de règles simples pour générer de nouvelles particules autour des particules initiales :

1. Les nouvelles particules ne doivent pas perturber le système, elles doivent donc être introduites autour de particules existantes dont le déplacement est faible ;
2. Pour ne pas faire exploser le système, les particules doivent être créées à proximité de la position d'équilibre entre les particules ;
3. Nous devons aussi vérifier que les nouvelles particules ne se trouvent pas sur la frontière, voire en dehors de la surface à reconstruire. Pour le premier point, il suffit de faire un calcul de distance signée entre la particule et la frontière.

Dans les paragraphes suivants, nous présentons les solutions pour satisfaire à ces règles.

4.3.1.1 Création des germes

C'est la phase d'initialisation de la reconstruction. Comme dans tout processus itératif, elle est très importante puisqu'elle conditionne le résultat final.

La première possibilité est de créer un seul, voire un petit nombre de germes, puis de laisser le mécanisme de création couvrir toute la surface avec des particules. Cette solution est utilisée lorsqu'il est difficile de connaître l'emplacement de la surface, par exemple dans le cas d'un échantillonnage d'une surface implicite. Il faut cependant s'assurer que l'on crée un germe dans chaque branche de la surface. En effet, si un couloir d'accès est de dimension inférieure à la taille des particules, la branche ne sera jamais reconstruite car les particules ne pourrons pas se propager à l'intérieur (Fig. 4.1). La deuxième solution est de générer le plus de particules en une seule fois, pour en avoir le moins possible à créer lors du processus itératif. Elle est employée lorsque l'on peut facilement connaître l'emplacement de la zone à reconstruire, par exemple l'intérieur d'un polygone du plan. Dans ce cas, il suffit de disposer des particules selon un état stable. On choisira de préférence l'arrangement le plus dense dans

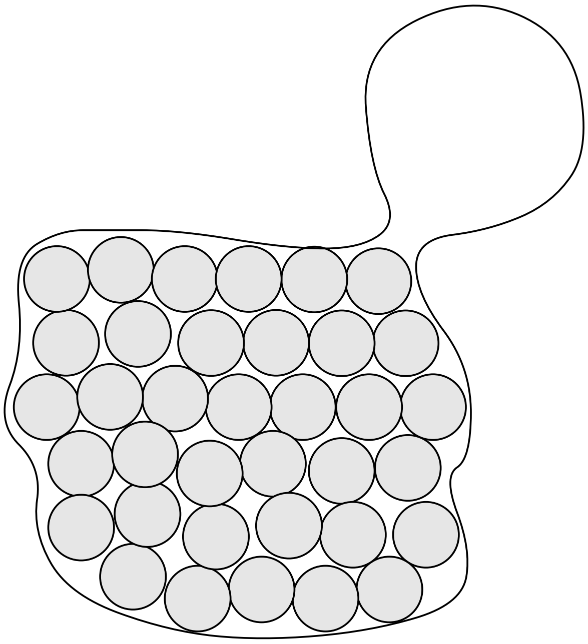


FIG. 4.1 – *Problème d'initialisation de la reconstruction*

le plan, le pavage hexagonal. La chose chose à faire est de vérifier que chaque particule se trouve bien à l'intérieur de la surface, et n'intersecte pas les frontières, quand il y en a.

4.3.1.2 Position des nouvelles particules

Les nouvelles particules créées doivent être proches de l'équilibre (règle 2). Pour nous conformer à cette règle, nous considérons le cercle de rayon $r = (1 + \alpha)R$, où R est le rayon des particules. α est le coefficient de proximité. Quand $\alpha = 1$, les particules sont tangentes. Pour $\alpha > 1$, les particules seront éloignées les unes des autres, l'échantillonnage initial sera peu dense. Enfin, pour $\alpha < 1$, elles sont superposées. Cela permet de créer rapidement beaucoup plus de particules mais cela rend le système un peu plus instable car les particules ne sont pas à l'équilibre lors de leur création. Le centre des nouvelles particules devra alors obligatoirement se trouver sur ce cercle. Il reste alors à déterminer la position de ce centre.

Une première méthode consiste à générer aléatoirement des points P sur le cercle, qui seront des positions potentielles. Il faut ensuite vérifier qu'aucune autre particule ne se trouve déjà à cet endroit, c.-à-d. calculer la distance entre le point P et le centre de toutes les particules susceptibles de se trouver dans son voisinage. Cette vérification est très onéreuse, puisqu'il faut tester toutes les particules existantes pour voir s'il n'y a pas de risque de superposition. Il est bien sûr possible de restreindre la recherche aux particules situées dans un voisinage proche de la position P, mais cela n'est toujours pas satisfaisant.

En fait, il est beaucoup plus efficace de rechercher directement une position valable, plutôt que de rejeter un grand nombre de positions potentielles. Pour cela, nous nous servons du voisinage de la particule existante. Sur un plan, un ensemble de particules homogènes (et notamment possédant le même rayon) se disposent selon un pavage hexagonal, qui est l'arrangement d'énergie minimale. Chaque particule doit normalement posséder six voisines, dont les centres sont les sommets de triangles équilatéraux. L'idée est de rechercher toutes les particules stables dont le voisinage est incomplet. Il y aura alors beaucoup plus de chances de créer une nouvelle particule dans cette zone, par rapport à une particule dont le voisinage est quasiment complet.

La première étape de cette seconde méthode est la recherche des particules voisines. Nous considérons que toute particule se trouvant à proximité d'une autre, est sa voisine. Nous déterminons un critère simple de proximité basé sur la distance entre les centres. Si celle-ci est inférieure au triple du rayon $distance(P, P') \leq 3R$, alors les particules sont voisines. Ensuite, les voisines sont triées selon l'angle entre le centre P' et l'axe Px, par exemple. Cela permet de déterminer l'écart angulaire entre deux voisins. Si cet écart est supérieur à un certain seuil, cela signifie qu'il y a un trou dans le voisinage. Nous créons alors une nouvelle particule sur la bissectrice de l'angle, à une distance de $(1 + \alpha)R$ de la particule P (Fig. 4.2).

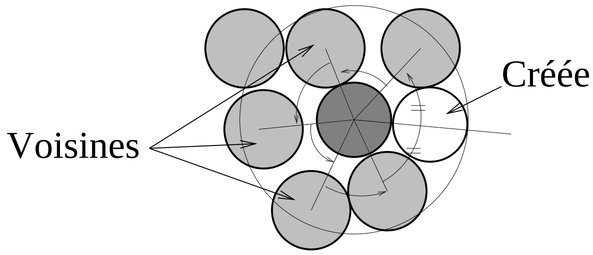


FIG. 4.2 – *Recherche de la position des nouvelles particules*

LOMBARDO, dans [Lom96] utilise une méthode équivalente pour échantillonner une surface implicite en 3D. À partir d'un germe proche de la surface, il crée des particules orientées qui seront attirées par la surface et qui se répartiront en fonction d'une loi d'attraction/répulsion. Il propose de traiter différemment les particules selon le nombre de voisins. Pour une particule isolée, il crée quatre voisines équidistantes, mais il n'en crée qu'une seule si il n'y a qu'une voisine. Dans les autres cas, des particules sont créées dans les trous. Ces règles empêchent la création d'un grand nombre de voisins dans les zones de forte courbure, ou le voisinage complet n'est plus forcément hexagonal comme dans le plan. L'idéal serait de déterminer le nombre de voisins en fonction de la courbure, et non plus arbitrairement (ici quatre particules).

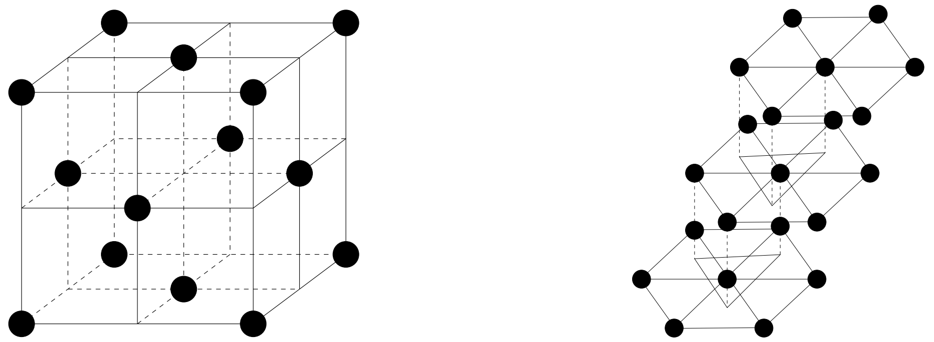
4.3.1.3 Distance au contour

Pour vérifier la règle 3, il ne faut pas créer de particules à l'extérieur ou sur la frontière. La distance entre une particule et un contour sera plus ou moins facilement calculée selon le modèle utilisé pour représenter les contours de l'objet à reconstruire. Mais, dans tous les cas, cela revient à évaluer la distance entre un point (le centre de la particule) et la courbe. Si cette distance est inférieure au rayon de la particule, $distance(contour, C) < R$, celle-ci est à cheval sur le contour, et elle ne sera pas créée.

Elle ne sera pas non plus créée si elle se trouve à l'extérieur du contour fermé. Sinon, l'espace va très vite se retrouver rempli de particules ! Pour de nombreuses représentations, il est nécessaire de discrétiser les contours. Il existe alors des algorithmes efficaces pour déterminer si un point est à l'intérieur d'un ensemble de polygones concaves.

4.3.2 Cas des volumes (3D)

Après avoir présenté le cas bidimensionnel, nous généralisons la reconstruction au objets volumiques. Dans le cas d'une reconstruction d'une forme décrite par une ou plusieurs surfaces fermées, la génération des particules devient un problème plus compliqué. Les particules se disposent toujours selon un arrangement d'énergie minimale : cubique face centrée ou hexaèdrique. Par contre, les informations d'angle ne sont plus exploitables. Le plan de référence



(a) Cubique face centrée (b) Hexaèdrique

FIG. 4.3 – *Mailles élémentaires des arrangements d'énergie minimale*

n'existe plus et il est alors impossible de classer les voisins d'une particule. Nous proposons alors d'adapter la première solution du cas bidimensionnel au cas tridimensionnel. Mais cette fois-ci, les positions potentielles sont tirées aléatoirement sur une sphère et non plus sur un cercle. Cela peut être réalisé simplement de la manière suivante :

$$\begin{aligned}
&\text{choisir aléatoirement } z \in [-1, 1] \\
&\text{choisir aléatoirement } t \in [0, 2\pi] \\
&r = \sqrt{1 - z^2} \\
&x = r * cos(t) \\
&y = r * sin(t)
\end{aligned}$$

Ensuite, comme pour le cas bidimensionnel, il faut regarder si la position est effectivement libre. La condition $distance(P, C) \leq (1+\alpha)*R$, permet d'éviter les superpositions de particules lors de la création. Nous préservons ainsi une certaine stabilité du système de particules. Cette vérification est très onéreuse, puisqu'il faut tester toutes les particules existantes pour voir s'il n'y a pas de risque de superposition. La complexité peut être réduite en partitionnant l'espace, et en restreignant la recherche au particules qui se trouvent dans le voisinage immédiat de la position P. Une autre possibilité est de disposer d'une liste des voisins de chaque particule.

Mais cela suppose que le voisinage ne varie pas trop ni trop souvent, car sinon la mise à jour des listes est beaucoup trop coûteuse.

De plus, pour obtenir une reconstruction correcte, et ne jamais laisser de zones vides, il faut générer un grand nombre de positions potentielles, dont beaucoup seront rejetées. Pour réduire le nombre de positions générées, nous définissons un critère qui correspond au taux de superposition d'une particule par rapport à ses voisins [SG95] :

$$\tau = \frac{1}{R} \sum_{i=1}^{n} (3R - d_i) \tag{4.1}$$

où R est le rayon des particules, n le nombre de voisines, et d_i est la distance entre la particule centrale et sa $i^{\text{ème}}$ voisine (Fig. 4.4). Cette mesure permet de déterminer si le voisinage

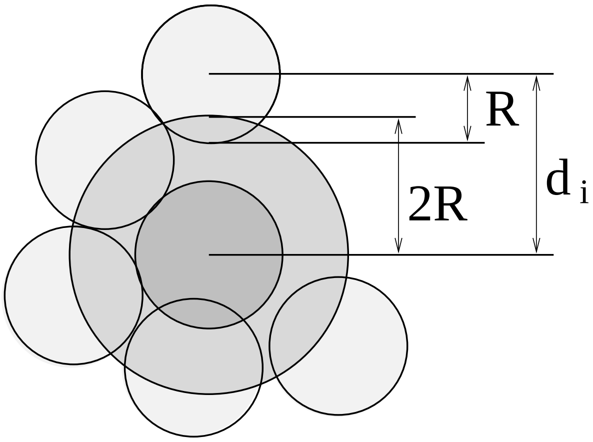


FIG. 4.4 – *Taux de superposition des particules*

d'une particule est complet. L'équation possède l'avantage d'être utilisable pour toutes les dimensions. Par contre, la valeur correspondant à un voisinage complet varie : elle est de 2 pour le long d'une courbe, de 6 dans un plan, et de 12 dans l'espace. On remarquera que, quelque soit la dimension, toutes ces valeurs correspondent au nombre de voisins. Cependant, le calcul de la valeur de τ ne demande pas vraiment plus de temps qu'un simple comptage, mais il est beaucoup plus précis. En effet, il se peut qu'une particule possède 12 voisines, alors que son voisinage n'est pas complet si ces dernières se trouvent à la limite de la zone d'influence (ici, deux fois le rayon).

4.4 Dynamique des particules

Dans cette section, nous décrivons brièvement les mouvements des particules lors de la reconstruction. Au départ, pour ne pas perturber le système, les particules sont créées avec une vitesse et une accélération nulles. Ensuite, chaque particule va interagir avec les autres particules et le monde extérieur. Dans le cas de la reconstruction, le monde extérieur se limite à la surface de l'objet, qui est une frontière fixe, infranchissable. Les particules évoluent dans un milieu sans gravité, et aucune autre force extérieur ne s'applique sur les particules, à part bien sûr le frottement. La reconstruction est un processus itératif. Après avoir calculé l'ensemble des forces appliquées aux particules à un instant donné t, nous déduisons la vitesse et la position de chaque particule à l'itération suivante, c.-à-d. à l'instant $t + dt$. Pour cela, nous intégrons les équations de NEWTON (Éq. C.1 et C.2) sur l'intervalle $[t; t + dt]$, à l'aide de la méthode d'EULER par exemple, ou d'une autre méthode de rang plus élevé (cf annexeC).

4.4.1 Calcul des forces

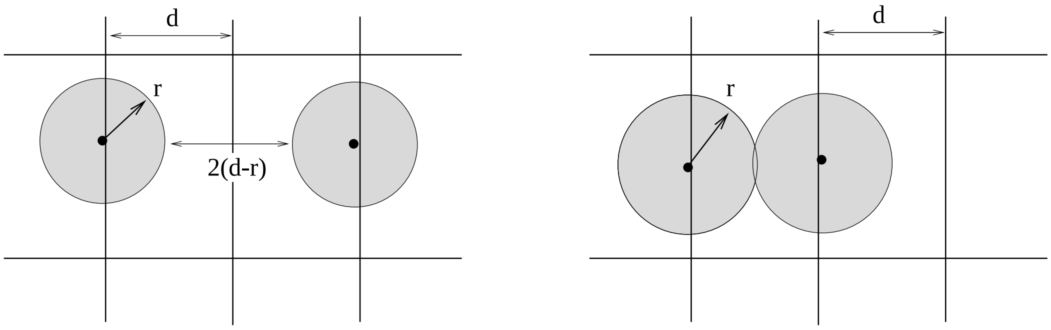
Les interactions entre les particules sont modélisées par une force d'attraction/répulsion. Nous utilisons la force dérivée du potentiel de Lennard-Jones (Éq. 3.1), dont les paramètres sont très intuitifs. La distance d'équilibre est égale à $2R$, les particules de rayon R sont alors tangentes. Si la distance entre deux particules est inférieure à ce seuil, elles se repoussent. Si elle est supérieure, elles s'attirent. Le calcul de cette force se fait en $O(n^2)$, car une particule est potentiellement en interaction avec toutes les autres particules du système. Pour réduire cette complexité, nous restreignons les interactions à une zone autour de la particule. Une particule en dehors de cette zone n'aura aucune influence. Il est possible de modéliser cette zone par une sphère autour de la particule, mais nous préférons utiliser une partition. La boîte englobante de la surface à remplir est découpée en cases, régulières ou non. Cette partition est effectuée une fois pour toutes puisque l'objet est statique. Nous maintenons à jour une liste des particules de chaque case de la partition. Les interactions seront alors réduites aux particules de la même case et étendues aux cases voisines. La taille et le nombre de cases est discuté au paragraphe suivant 4.4.2.

Lorsqu'une collision ou un contact entre une particule et la frontière est détectée, l'idéal serait de revenir en arrière jusqu'au moment exact de l'impact, ce qui implique que l'on ait mémorisé cette position. Cette méthode est très lourde, elle est difficilement applicable pour un grand nombre d'objet. Plus il y a de particules dans la scène, plus le risque de collision est élevé. Un grand nombre de collision peuvent alors se produire lors d'une itération. Il faut alors revenir au moment du premier impact, gérer la collision, puis reprendre jusqu'au second impact. Mais entre temps, de nouvelles collisions ont pu se produire, et il faut encore revenir en arrière, ce qui est une perte de temps. Une solution, moins précise mais plus facile à mettre en œuvre, peut être de simplement déplacer la particule pour qu'elle ne soit plus en collision.

4.4.2 Complexité du calcul des forces

Le calcul des forces est un grand consommateur de temps dans les systèmes de particules. L'évaluation des forces entre les n particules se fait en $O(n^2)$. Chaque particule peut potentiellement interagir avec toutes les autres. Il faut donc tester tous les couples de particules.

Pour réduire la complexité, une méthode très simple à mettre en œuvre est de partitionner l'espace. Si l'on dispose de la boîte englobant toutes les particules, il suffit de créer des cases de taille régulière. Ensuite, chaque particule est affectée à une case selon sa position. La recherche des particules en interaction avec une particule donnée sera restreinte aux cases voisines. Cela suppose que la force s'annule à partir d'une certaine distance. La figure 3.5(a) montre deux particules séparées d'une distance $2(d - r)$, qui est la plus petite distance possible entre deux particules n'appartenant pas à deux cases voisines. Si cette distance est plus petite que la distance d_a d'annulation de la force, alors les particules seront en interaction, mais elle ne sera pas détectée. Il faut donc que $d \geq \frac{d_a}{2} + r$. Il est évident que cette méthode ne peut pas être appliquée pour des particules de tailles différentes, puisque la plus petite dimension des cases doit au moins être égale au diamètre de la plus grosse particule. Sinon, une particule serait susceptible d'intersecter une particule située dans une case non contiguë, et la collision ne sera pas détectée puisqu'elle ne fait pas partie du voisinage immédiat (Fig. 3.5(b)). La



(a) En fonction de la force (b) En fonction du rayon

FIG. 4.5 – *Détermination de la taille minimum des cases*

complexité du calcul des forces se fait maintenant en $O(n)$, en moyenne. Mais dans le cas extrême où pratiquement toutes les particules sont regroupées dans une même case, le calcul est plus long que sans le partitionnement. Il faut aussi que le nombre de particules par cases soit assez réduit, car par exemple en trois dimensions, il y a 27 cases voisines, ce qui peut faire au total un grand nombre de particules à tester.

D'autres techniques existent. Elles créent un partitionnement autour de groupes de particules. À l'intérieur d'un groupe, les particules interagissent normalement. Par contre, une particule d'un groupe différent est influencée par une seule force commune. Cette méthode de regroupement est très utilisée pour simuler des interactions entre planètes d'un même système, ou l'influence des autres systèmes. L'approximation faite sur les forces entre les systèmes sont d'autant plus précises que les systèmes sont éloignés. Cela suppose que les particules sont disposées sous forme d'amas sans ou avec des interactions simplifiées, ce qui n'est pas applicable à la modélisation d'objets compacts.

4.4.3 Adapter le pas de temps

Dès qu'une particule a parcouru une distance supérieure à son rayon, les collisions risquent de ne plus être détectées (Fig. 4.6). De plus, cela rend le système extrêmement instable, car deux particules qui se rapprochent l'une de l'autre en parcourant une distance égale à leur rayon, risquent de se superposer exactement, ce qui va provoquer une force de répulsion infinie à l'itération suivante (Fig. 4.7. Pour éviter cela, il faut utiliser un échantillonnage temporel plus fin. Mais, nous encourrons le risque de calculer beaucoup de positions intermédiaires

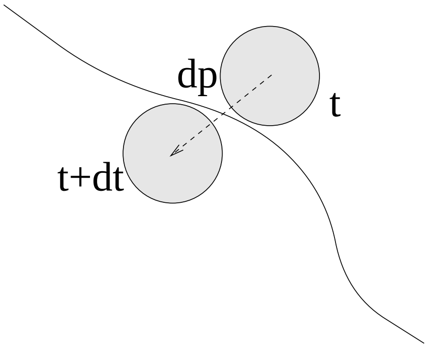


FIG. 4.6 – *La collision n'est pas détectée si le déplacement est trop important par rapport au rayon*

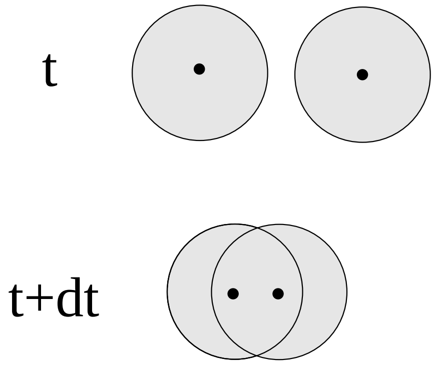


FIG. 4.7 – *Le déplacement relatif des particules est trop important, elles risquent de se superposer*

dont nous n'avons pas besoin. Il faut donc déterminer un pas de temps pour le calcul des nouvelles positions qui dépende du mouvement des particules. Pour réaliser cette condition, nous utilisons un pas de temps adaptatif. Ainsi, il faut prendre une valeur seuil suffisamment faible pour éviter les explosions. Cette valeur peut difficilement être estimée, car cela implique de pouvoir prédire comment va évoluer le système à l'itération suivante. Étant donné la pente importante de la force de répulsion, et pour restreindre les oscillations, en pratique, nous choisissons la valeur seuil ainsi : si pour une seule particule nous avons $déplacement > R/8$, alors nous réduisons de beaucoup le pas de temps : $dt = \alpha.dt$, où $\alpha = 0.5$ est le taux de réduction. Par contre, si aucune particule ne se déplace de trop, alors nous pouvons augmenter légèrement le pas de temps de la façon suivante : $dt = \beta.dt$, avec $\beta = 1.1$.

4.4.4 Quand s'arrêter ?

Il reste à savoir à quel moment arrêter la reconstruction. Le fait de ne jamais supprimer de particules, nous permet d'avoir un critère d'arrêt simple. En effet, tant que de nouvelles particules sont créées, nous continuons le processus. Lorsque l'on ne peut plus créer de particules, cela signifie que tous les trous ont été bouchés, la reconstruction peut être arrêtée. À partir de ce moment là, nous continuons la reconstruction jusqu'à obtenir un système en équilibre. Si nous enlevons toutes les actions extérieures (par exemple, la collision avec la frontière), le système ne doit plus évoluer. Cela permet d'obtenir un modèle d'objet déformable au repos. Lorsque nous animerons cet objet, il doit être parfaitement stable.

La stabilité des particules, dans un tel cas de figure, peut être quantifiée par les forces extérieures qui lui sont appliquées. L'algorithme s'arrête quand la somme des forces extérieures est inférieure à un seuil pour chaque particule. Cependant, ce critère ne permet pas de terminer la reconstruction dans tous les cas, certaines forces pouvant ne jamais s'annuler. Par exemple, si nous créons beaucoup de particules, celles-ci vont se repousser fortement et ne vont être bloquées que par les contours. Des forces importantes seront donc nécessaires pour les empêcher de traverser la frontière, et ces force ne s'atténueront jamais puisque nous ne détruisons jamais de particules. Ainsi, nous utilisons le déplacement des particules comme critère d'arrêt.

Si toutes les particules se déplacent d'une distance inférieure à un seuil, alors nous arrêtons la reconstruction. En pratique, nous choisissons un déplacement maximum dépendant du rayon :

$$d < \frac{R}{\alpha} \text{ avec } \alpha = 20$$

Ce critère est moins bon que le précédent, mais il garantit que l'algorithme s'arrêtera au bout d'un certain temps. Tant que nous ne créons pas de nouvelles particules, l'énergie du système décroît, donc les déplacements diminuent.

4.5 Limitations

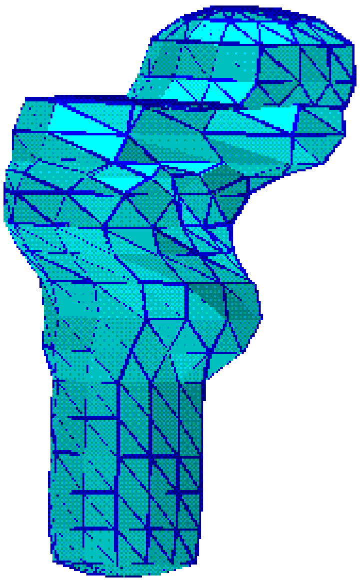
Un paramètre essentiel de la reconstruction est le rayon d'équilibre de la force d'attraction/répulsion, qui peut d'ailleurs souvent être rapproché du rayon des particules. En effet, plus ce rayon est grand, plus les centres des particules seront éloignés les uns des autres, et moins il faudra de particules pour recouvrir l'ensemble de l'objet. Nous obtenu alors un échantillonnage peu dense, mais les détails précis ne sont pas correctement reproduits. À l'inverse, si nous voulons voir les détails de l'objet, il faut définir un petit rayon de repos. Alors, l'échantillonnage sera fin, mais comportera un grand nombre de particules

La figure 4.8 illustre la reconstruction d'un objet selon différents niveaux de précision. L'image 4.8(a) représente un modèle polygonal d'une tête fémorale. Dans la figure 4.8(b), des particules de taille importante ont été utilisées. La reconstruction est grossière, seule la forme générale de l'objet est visible. Par contre, dans la figure 4.8(c), de plus petites particules permettent d'obtenir une représentation plus détaillée de l'objet. La figure suivante 4.8(d), offre encore plus de détails mais le nombre de particules nécessaire devient vraiment important.

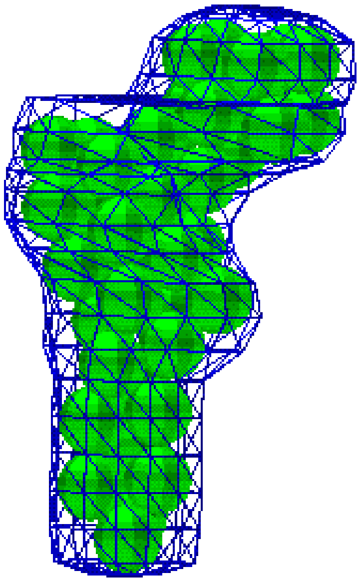
Nous nous apercevons alors que pour avoir un échantillonnage fin dans certaines zones, de nombreuses particules sont inutiles dans les parties où il y a peu de détails. En particulier, toutes les petites particules du centre pourraient être remplacée par quelques particules de rayon plus important, sans pour autant perdre de détails. Ainsi, un échantillonnage uniforme n'est pas très intéressant, et il est nécessaire d'adapter le rayon des particules en fonction du niveau de détail requis.

4.6 Reconstruction multi-couche

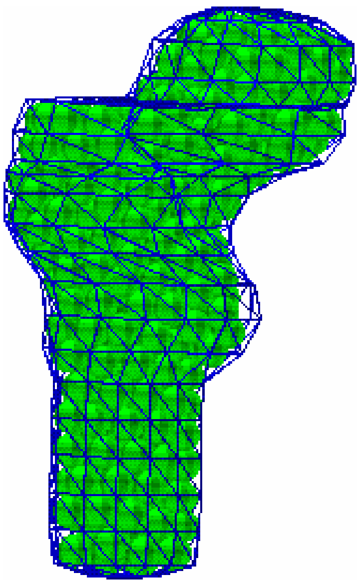
La reconstruction de volumes à l'aide de particules uniforme possède de nombreux avantages. Elle a le mérite d'être très simple, on fixe les paramètres de toutes les particules une fois pour toutes au début. Elles ont toutes le même rayon, réagissent à la même force d'attraction/répulsion. Cela permet évidemment de réduire l'occupation mémoire et de faciliter un certain nombre de traitements, tels que la gestion des partitions pour la recherche des voisins. Mais, très vite, on se trouve limité par cette uniformité, surtout lorsque l'on souhaite adapter l'échantillonnage en fonction des détails. Ainsi, nous avons étendu notre méthode de reconstruction à des particules ayant des caractéristiques différentes.



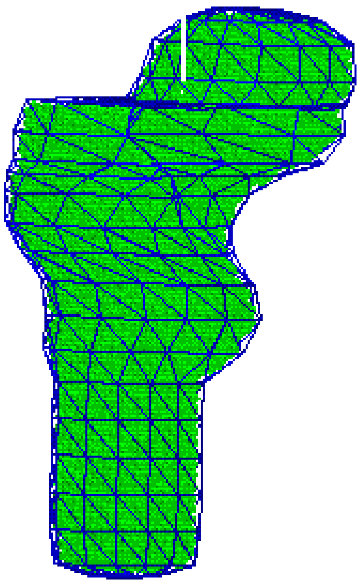
(a) Représentation polygonale initiale



(b) 41 particules de rayon 10



(c) 417 particules, rayon 5



(d) 12732 particules, rayon 1.6

FIG. 4.8 – *Différents niveaux de précision pour un même objet, lors d'une reconstruction mono-couche*

L'idée de départ est de placer des petites particules là où on a vraiment besoin de détails, et des grosses partout ailleurs. Nous reprenons la structuration en couches définie par JIMENEZ [Jim93] : un noyau, un derme et un épiderme. Nous nous sommes inspirés de sa méthode, que nous avons adaptée à notre cas. Nous détaillons notre approche dans les chapitres suivants.

4.6.1 Description de notre approche

Pour remplir une région à l'aide de particules de taille variable, nous commençons par créer de grosses particules au centre. Elles appartiendront au noyau. Nous appliquons alors le même processus de création et de positionnement que pour un rayon uniforme. C'est-à-dire que nous plaçons l'ensemble des particules sous l'action de lois physiques et que nous laissons évoluer ce système jusqu'à atteindre un certain équilibre. Pendant ce temps, nous essayons de créer des particules pour combler les vides dans ce noyau. Ensuite, nous procédons de la même façon pour le derme, à cette exception près que les particules des différentes couches peuvent interagir entre-elles. Et enfin, nous reconstruisons la couche superficielle de l'épiderme, en procédant de même.

Le schéma suivant reprend les grandes lignes de l'algorithme de reconstruction multi-couche :

Algorithme 4.2 Reconstruction multi-couche

pour chaque couche
1 Définir les paramètres de la couche, qui sont essentiellement les caractéristiques géométriques des particules et les forces qui leur sont appliquées
2 Créer régulièrement les particules initiales
répéter pour l'ensemble des particules
3 Détecter les collisions avec les obstacles de la scène, ou entre les différents systèmes de particules
4 Déterminer les forces externes et internes aux systèmes de particules
5 Utiliser les lois de la dynamique pour intégrer l'accélération et calculer les nouvelles positions des particules
6 Générer de nouvelles particules autour des particules déjà existantes
jusqu'à ce que le système soit à l'équilibre
fin

La différence majeure par rapport à la méthode de reconstruction mono-couche est la détermination des paramètres de la couche. Le calcul des forces internes est lui aussi un peu différent, puisque les interactions d'attraction/répulsion sont limitées aux particules appartenant à deux couches successives. Sinon, il s'agit essentiellement d'appliquer plusieurs fois algorithme mono-couche, c.-à-d. une fois pour chaque couche.

4.6.2 Réalisation pratique des couches

Avant de pouvoir matérialiser les différentes couches, il faut d'abord définir leurs caractéristiques géométriques et les relations qui existent entre elles, ou plutôt entre les particules qui les composent.

En premier lieu, il est important de rappeler que la couche du noyau se trouve au centre de l'objet. Le derme, quant à lui, entoure le noyau, et l'épiderme se situe à la limite extérieure, contre la frontière de reconstruction. Cela signifie que les grosses particules doivent être tenues éloignées de cette frontière, et que plus la taille des particules décroît, plus elles sont autorisées à se trouver près. Cette contrainte est réalisée en ajoutant une frontière fictive pour chaque couche, dont le rôle est d'empêcher les particules appartenant à cette couche de trop se rapprocher du contour. Elle doit aussi permettre aux particules d'une couche de ne pas se retrouver au milieu d'une couche voisine. La façon la plus simple de réaliser cette frontière est de créer une sorte d'« offset » vers l'intérieur du contour initial. Du point de vue de la particule, elle sera alors représentée par une distance minimum d_{min} au contour. Une particule ne pourra pas se trouver à une distance inférieure. Pour l'épiderme, cette distance est bien évidemment nulle.

Ainsi, la frontière fictive n'est jamais calculée explicitement, ce qui serait d'ailleurs assez long. De la même façon que les particules entrent en collision avec le contour initial. La détection est très simple. Lorsque la distance entre le centre C d'une particule de rayon R et le contour est trop faible, c.-à-d. $distance(C, contour) \leq d_{min} + R$, alors il y a collision. Nous lui appliquons les mêmes forces que lors d'une collision avec un obstacle fixe, la particule rebondit ou glisse le long de cette frontière qu'elle ne peut pas franchir.

Il est nécessaire de garder une structuration stricte des couches. Ainsi, de la même façon, nous créons une deuxième frontière fictive pour chaque couche. La première empêchait les grosses particules de se rapprocher du contour, celle-ci va empêcher les petites de s'en éloigner. Nous évitons ainsi que les couches ne se mélangent entre elles, ou que des particules de l'épiderme ne se retrouvent au cœur du noyau, où elles ne servent à rien et qui plus est où elles peuvent jouer le rôle de grains de sable, empêchant les grosses particules de se déplacer librement. Ces nouvelles frontières fictives (une pour chaque couche) vont être matérialisées par une distance maximale d_{max} au contour. Bien sûr, nous avons $d_{max} > d_{min}$. Pour le noyau, cette distance sera infinie, et en pratique nous n'en tiendront pas compte. Cette fois, il y aura collision lorsque $distance(C, contour) \geq d_{max} - R$.

De plus, les couches ne sont pas complètement disjointes. Cela signifie que la distance minimum d'une couche ne correspond pas exactement la distance maximum de la couche suivante. Pour assurer une reconstruction correcte dans les zones situées à la limite entre deux couches, il faut que celles-ci se recouvrent partiellement. Cela signifie que $d_{min\_noyau} < d_{max\_derme}$ et $d_{min\_derme} < d_{max\_épiderme}$. L'ensemble des relations géométriques entre les couches est représentée sur la figure 4.9.

Enfin, pour limiter le nombre d'interactions entre les particules des différentes couches, nous imposons qu'une particule soit en interaction uniquement avec les particules de la même couche ou des couches immédiatement voisines. Cela signifie qu'une particule appartenant au noyau interagit avec celles du noyau et du derme, mais jamais avec celles de l'épiderme. Nous

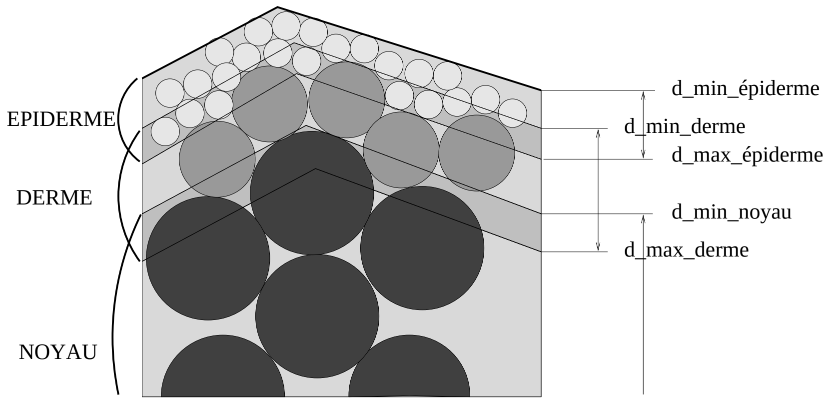


FIG. 4.9 – *Structuration en couches*

en déduisons donc $d_{min\_noyau} \geq d_{max\_épiderme}$. Cela signifie aussi que seules les particules de l'épiderme sont en relation avec le monde extérieur à l'objet lors de l'animation, d'où $d_{min\_derme} > 0$.

4.6.3 Passage d'une couche à l'autre

Dans ce paragraphe, nous présentons les données et la méthode qui permettent de gérer la reconstruction des différentes couches.

4.6.3.1 Structure des données

Une particule contient, entre autres, les informations suivantes :

- **p**, **v** et **f**, qui représentent respectivement la position, la vitesse et l'ensemble des forces appliquées à la particule. Ces paramètres permettent de décrire les mouvements ;
- m et R, la masse et le rayon, qui sont les caractéristiques physiques de la particule ;
- n le numéro de la couche à laquelle elle appartient, ce qui permet d'accéder aux distances d_{min} et d_{max}.

La masse et le rayon doivent être présents dans chaque particule, car rien ne nous empêche d'avoir des particules de taille et de masse différentes à l'intérieur d'une même couche. Cela permet au contraire de modéliser un plus grand nombre de comportements non homogènes. Par contre, il n'y a aucun besoin de stocker les paramètres communs à toutes les particules de la couche. En particulier, ni d_{min} ni d_{max} ne sont présents dans les données de la particule. Ces distinctions sont importantes pour la détermination des valeurs de ces paramètres de façon automatique. Le paragraphe suivant en détaille le calcul.

4.6.3.2 Valeurs des paramètres

Le choix des paramètres est une opération critique de la reconstruction. En effet, si ceux-ci sont mal choisis, la reconstruction peut ne pas être adéquate. Il est difficile de définir automatiquement ces paramètres. Par exemple, si la largeur de la couche est inférieure au diamètre des particules de cette couche, alors aucune particule ne pourra être créée, et la couche sera vide. À l'inverse, si le rayon des particules est nettement inférieur à cette même largeur, alors un très grand nombre de particules seront générées. Tout le bénéfice des couches est perdu puisque l'on se retrouve avec beaucoup de petites particules à un endroit où elles ne sont pas nécessaires.

Il serait vraiment difficile de trouver tous les paramètres importants de la reconstruction (rayon, masse, distances à la frontière) de façon automatique. Pour cela, il faudrait analyser la forme à reconstruire, puisque les paramètres dépendent de la géométrie. Si un objet est de forme sphérique, alors le rayon des particules du noyau pourra être important par rapport à la taille de cette forme. Par contre, si la forme est aplatie, le noyau devra comporter des particules de plus petit rayon.

Notre méthode n'est pas complètement automatique, il faut initialiser interactivement la reconstruction. Pour simplifier la phase d'initialisation des paramètres, l'utilisateur doit en fixer les valeurs pour le plus petit nombre possible. Ainsi, nous avons choisi les plus significatifs, à savoir :

1. le rayon des particules de l'épiderme, qui peut être vu comme la précision souhaitée pour la reconstruction ;
2. la largeur des couches, qui est déterminé par un coefficient γ qui représente le nombre de particules de la couche qui tiennent dans la largeur. Cela signifie que la largeur de la couche dépend du rayon, elle est égale à γR.
3. la densité δ des particules (qui est la même pour toutes les particules) ;
4. et le taux α d'augmentation du rayon entre deux couches.

Les autres paramètres sont déduits à partir des quatre précédents, qui sont fixés par l'utilisateur. Nous procédons de la façon suivante. Pour chaque couche, nous déterminons le rayon, soit interactivement (pour l'épiderme), soit automatiquement en fonction de la couche précédente. Puis, nous calculons les autres valeurs de la couche :

$$
\begin{aligned}
R' &= \alpha R \\
{d'}_{min} &= d_{max} - R \\
{d'}_{max} &= d'_{min} + \gamma R' \\
m' &= \delta / \left(\frac{4}{3} \pi {R'}^3 \right)
\end{aligned}
\tag{4.2}
$$

Ces équations permettent de vérifier toutes les relations entre les couches énoncées au paragraphe précédent. Le coefficient α est très important puisqu'il permet de déterminer le rayon des particules des couches internes. Nous verrons plus tard, au paragraphe 5.5.2 sur la subdivision des particules, quelle valeur donner à ce coefficient. La seconde équation assure un « mélange suffisant » entre les couches, en autorisant les particules d'une couche à pénétrer dans la couche voisine intérieure, dans une bande de la largeur du rayon. La largeur de la couche dépend du rayon, ce qui évite d'avoir des couches vides où trop pleines. Le coefficient γ permet de définir approximativement le nombre de particules par couche. Si $\gamma = 6$, cela correspond à trois particules côte à côte dans la largeur. Et enfin, m' est calculé en fonction de R', de telle façon que la densité volumique soit respectée.

Nous verrons dans le chapitre suivant (5.3), comment ces paramètres de la reconstruction vont influer sur le comportement de l'objet lors de l'animation. Cette corrélation n'est pas négligeable, car il ne faut pas perdre de vue que la reconstruction se fait dans le but de produire des objets dynamiques et déformables.

4.6.4 Exemple de reconstruction

Dans cette section, nous illustrons la méthode de reconstruction d'objet tridimensionnel à l'aide de plusieurs couches. Les figures 4.10(a) et (b) représentent la première étape : la reconstruction du noyau à l'aide de grosse particules. Ensuite, sur les figures 4.10(c) et (d), nous pouvons observer le derme. Le rayon a diminué et les particules sont autorisées à venir plus près des contours. Enfin, les figures 4.10(e) et (f) montrent la reconstruction de l'épiderme avec de petites particules qui entourent la couche précédente. La dernière image est à comparer avec celle de la figure 4.8(d). Nous obtenons la même qualité de reconstruction mais avec pratiquement quatre fois moins de particules.

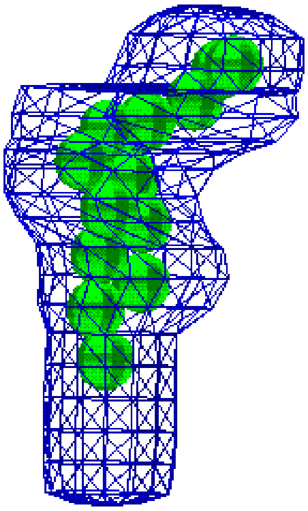
4.7 Conclusion

Nous avons présenté une nouvelle méthode de reconstruction d'objets déformables. À partir d'une courbe ou d'une surface fermée, nous calculons un échantillonnage de la surface intérieure, respectivement du volume intérieur. Nous n'imposons aucune restriction sur le modèle de départ, il suffit de pouvoir calculer une distance entre un point quelconque et la représentation de l'objet. Nous nous sommes cependant intéressés plus précisément à la reconstruction des volumes à partir de surfaces B-splines fermées, qui sont les représentations que nous obtenons à partir des données initiales disposées en contours (cf. partie I).

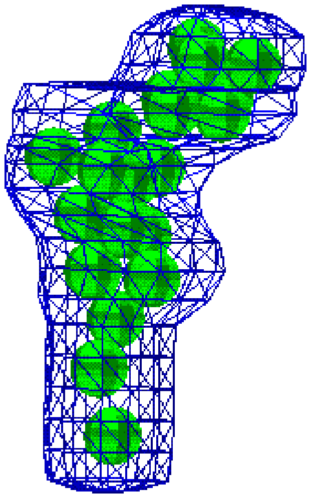
Le modèle de particules obtenu présente de nombreux avantages :

- Il est structuré en couches ;
- Il est dynamique ;
- Il permet de répondre simplement à des contraintes complexes ;
- Ses paramètres sont intuitifs, car basés sur des paramètres physiques.

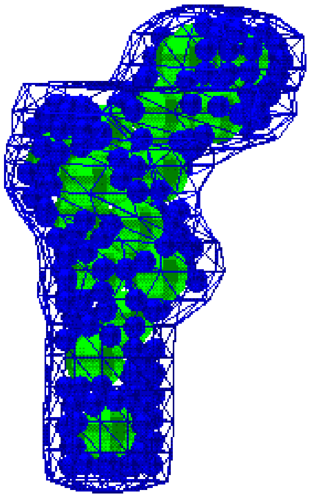
La structuration en couches est la plus grande innovation de la méthode de reconstruction. La difficulté vient du choix des caractéristiques des différentes couches, mais nous avons proposé une méthode de détermination qui permet de reconstruire de manière semi-automatique tous les types d'objets, quelque soit leur géométrie. À plus long terme, la reconstruction devra prendre en compte les caractéristiques géométriques de la forme. L'idéal serait de disposer d'un ensemble d'objets de base, pour lesquels les paramètres des différentes couches seraient prédéfinis. L'utilisateur n'aurait alors plus qu'à choisir la forme de base qui correspond le mieux à



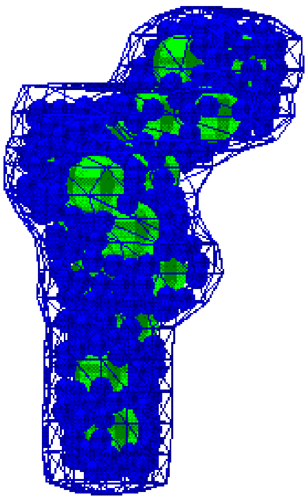
(a) Initialisation 1re couche : 11 particules



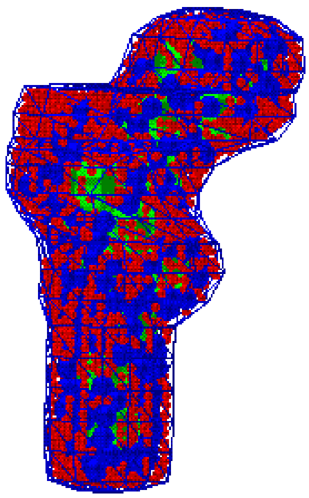
(b) Fin 1re couche : 20 particules



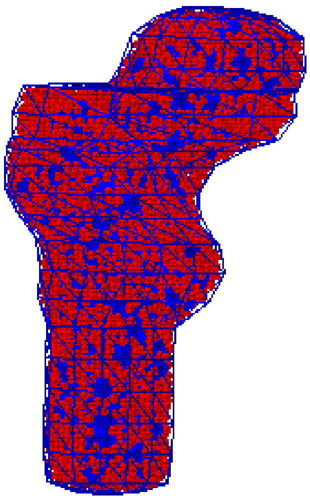
(c) Initialisation 2e couche : 235particules



(d) Fin 2e couche : 355particules



(e) Initialisation 3e couche : 2066 particules



(f) Fin 3e couche : 3175 particules

FIG. 4.10 – *Évolution de la reconstruction multi-couche d'un fémur*

l'objet qu'il doit reconstruire. Il serait aussi possible d'effectuer cette mise en correspondance de manière automatique, pour proposer un jeu de paramètres adéquat.

La structuration en couches apporte plusieurs points positifs. Elle permet une reconstruction plus rapide, puisque les couches sont traitées les unes après les autres. La représentation finale est plus compacte, car nous créons moins de particules au centre. Cela nous permet de nous concentrer sur les zones frontières, où une plus grande précision est requise. Ainsi, la plupart du temps, moins de particules sont nécessaires pour obtenir un meilleur modèle qu'avec une reconstruction mono-couche.

D'autre part, ce modèle est composé de particules régies par les lois de la physique. Leur animation ne posera pas de problème majeur. Un système de particules est une vue simplifiée de l'objet physique (qui lui aussi contient des particules), mais à une échelle différente. Les simulations de tels objets, qui sont basées sur les lois de la physique, seront donc assez réalistes. Ainsi, à partir d'éléments de base très simples, nous pouvons créer des comportements complexes proches de ce qui existe dans le monde réel.

La reconstruction est assez intuitive, tous les paramètres sont extrêmement simples. Ils correspondent à des distances ou à des grandeurs qui ont une signification physique, telles que la masse ou le rayon.

5 Animation de systèmes de particules

5.1 Position du problème

L'animation des objets déformables peut se décomposer en plusieurs grandes composantes [Jim93], qui sont :

- la description des caractéristiques physiques de la matière composant l'objet ;
- la prise en compte des interactions entre les différents objets d'une scène ;
- et le contrôle du mouvement et des trajectoires.

Le premier point concerne le mode de caractérisation de la matière. Il s'agit essentiellement de transformer les paramètres réels de l'objet dont on veut simuler le comportement, en un ensembles de paramètres du modèle informatique [JGL97]. Les paramètres sont souvent définis lors de la modélisation ou de la construction de l'objet. Ainsi, selon le comportement souhaité, un objet ne sera pas représenté de la même façon. Par exemple, un comportement élastique sera plutôt représenté par des ressorts, alors qu'un comportement plastique ou inélastique sera modélisé de préférence par une liaison de type attraction/répulsion. A partir de là, les paramètres du modèle ne seront pas les mêmes puisque le modèle est différent.

Le second point concerne les relations entre les objets, collision ou contact. Il s'agit dans un premier temps de détecter les inter-pénétrations entre les corps [MW88, Kam93, BV91b] , pour pouvoir en déduire les modifications à apporter pour calculer les nouvelles positions. Il est souvent nécessaire de distinguer les contacts instantanés (les chocs), des contacts prolongés (les empilements) [Gas93, Bar98]. Dans les systèmes de particules, les interactions entre particules appartenant à différents objets sont gérés de façon similaire aux interactions internes aux systèmes, ce qui présente l'avantage de simplifier grandement la gestion des collisions.

Le dernier point consiste à décrire l'environnement de la scène. Il s'agit de définir l'ensemble des forces extérieures et des contraintes qui seront appliquées aux objets. Cela peut être tout simplement ajouter de la gravité, ou forcer un objet à se déplacer le long d'une trajectoire fixée, ou encore contraindre à rester en contact avec un autre [PB88, BB88, Wit92a].

Dans le chapitre 4, nous avons présenté notre méthode de reconstruction d'objets déformables. Dans ce nouveau chapitre, nous montrons comment nous animons de tels modèles.

Nous reprenons quelques points du chapitre précédent, puisque nous précisons certains paramètres du modèle qui permettent de définir le comportement physique des objets. Ce sont des paramètres spécifiques à la simulation, ils n'étaient donc pas nécessaires lors de la reconstruction, mais ils conditionnent une simulation correspondant au comportement souhaité. Nous présentons aussi un certain nombre de simulations sous contraintes, dont certaines, comme la préservation de volume ou la fracture, sont difficiles à reproduire avec des modèles autres que les systèmes de particules.

5.2 Description de notre méthode d'animation

Dans cette section, nous considérons que les objets déformables ont déjà été reconstruits. Ils sont placés dans la scène à animer, et peuvent être entourés d'obstacles qui peuvent être fixes ou non (généralement représentés par un modèle polyèdrique). L'algorithme d'animation peut alors se décomposer en plusieurs étapes qui sont :

Algorithme 5.1 Animation des systèmes de particules

répéter
1 **pour** tous les objets
Détecter les collisions entre les objets
pour chaque système de particules
2 **pour** chaque particule
3 Calculer les forces internes avec les particules du système
4 Calculer les forces de répulsion avec les particules des autres systèmes en collision
5 Gérer les collisions et les contacts avec les obstacles (forces externes)
6 Intégrer les forces pour obtenir la nouvelle position
pour chaque obstacle mobile
7 Gérer les collisions avec les autres obstacles
8 Calculer la translation et la rotation
9 Déterminer la nouvelle position
10 Incrémenter le temps
fin

Le principe de l'animation des objets de la scène est comparable à celui utilisé pour la reconstruction des systèmes de particules. Les objets sont dirigés par un ensemble de lois physiques. Nous calculons les forces qui entrent en jeu à chaque étape entre tous les éléments de la scène. Nous connaissons alors la position, la vitesse et les forces appliquées à l'instant t. Ensuite, l'intégration de ces forces nous donne la vitesse de chaque particule. Nous intégrons la vitesse pour obtenir la nouvelle position de la particule, à l'instant $t + dt$. Nous agissons de la même façon avec les obstacles rigides, mais cette fois en tenant compte de la rotation.

Le calcul des forces internes entre les particules d'un même système est une opération très coûteuse. Nous pouvons utiliser un partitionnement de l'objet, de la même manière que pour la reconstruction. Il y pourtant une difficulté supplémentaire, car la boîte englobante n'est plus fixe. Il faut refaire la partition (et son contenu) à chaque itération, ce qui n'est rentable que pour un très grand nombre de particules.

Lorsque deux systèmes de particules sont en collision, nous modélisons la non pénétration des objets à l'aide de forces de répulsion. Nous utilisons la force de Lennard-Jones, mais uniquement la composante de répulsion. Si deux particules sont situées à une distance r telle qu'elles se superposent, alors elles se repoussent. Si elles sont à une distance supérieure, rien ne se passe :

$$\begin{cases} \text{si } r < r_0 \text{ alors } \vec{f}(r) = \frac{-mn\varepsilon}{(n-m)r_0} \left(\left(\frac{r_0}{r}\right)^{(n+1)} - \left(\frac{r_0}{r}\right)^{(m+1)} \right) \frac{\vec{r}}{\|\vec{r}\|} \\ \text{sinon } \vec{f}(r) = 0 \end{cases}$$

où r_0 est le rayon de repos entre les deux particules (en pratique $r_0 = R_1 + R_2$), m, n et ε sont les paramètres classiques de la force de Lennard-Jones (cf. éq 3.1). Il serait là aussi très long de tester toutes les distances entre les toutes les particules des systèmes en collision, puisque la plupart du temps, seule une petite zone de chaque système est concernée par la collision. L'emploi des partition est encore très utile pour réduire la complexité. Nous ne testons que les particules appartenant à des partitions en collision.

La détection des collisions et la prise en compte de celles-ci lors de l'animation est une partie critique qui requiert beaucoup de temps de calcul. Le but de notre travail n'est pas d'optimiser la détection des collisions. De plus amples précisions peuvent être trouvées dans les références [BV91b, Kam93].

5.3 Définition des paramètres du modèle

Dans cette section, nous détaillons le choix des paramètres pour l'animation. Ils se repartissent en deux catégories. Les paramètres géométriques sont essentiellement des paramètres liés au type du modèle, dans notre cas un système de particules disposées en couches. Les paramètres de déformation sont ceux qui permettent de définir le comportement physique de l'objet (type de forces et leurs coefficients).

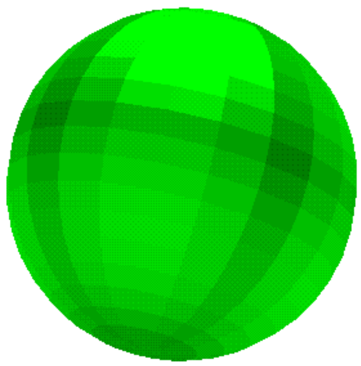
5.3.1 Paramètres géométriques

Les paramètres géométriques d'un objet sont déterminés au moment de sa modélisation ou de sa reconstruction. Ils dépendent grandement du comportement que l'on veut que l'objet soit capable de reproduire lors de la phase de l'animation.

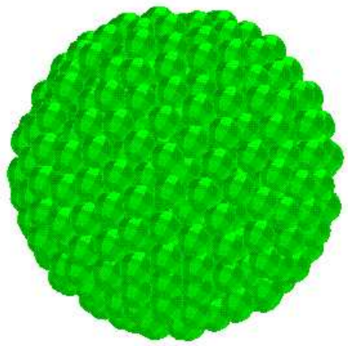
5.3.1.1 Nombre de couches

Dans le chapitre précédent, nous avons distingué trois couches distinctes pour modéliser les objets déformables. Chaque couche a ses propres fonctionnalités. Mais, pour certains objets, nous n'avons pas besoin de toutes les propriétés offertes par de tels modèles. Ainsi, les couches n'ont pas besoin d'être toutes représentées.

Par exemple, si un objet sphérique a un comportement totalement rigide, une seule particule est suffisante pour le modéliser. Par contre, si cette même sphère est légèrement déformable, alors il faudra trois couches pour la modéliser. Si la sphère est complètement déformable, seules les couches de l'épiderme et du derme devront être présentes. Les grosses particules du noyau empêcherait d'écraser l'objet pour qu'il atteigne une épaisseur inférieure au rayon des particules. À l'extrême, si la sphère est capable de se fracturer, il ne doit rester qu'une seule couche, celle de l'épiderme. C'est elle qui est en contact avec le monde extérieur, et aucune autre. Lors de déformations trop importantes, il se peut que des particules intérieures se retrouvent à l'extérieur. Cela poserait donc des problème lors de la détection des collisions si des particules du derme voire du noyau étaient en contact avec d'autres objets. La figure 5.1 illustre différentes représentations de la sphère selon son comportement potentiel.



(a) Sphère rigide composée d'une seule particule



(b) Sphère déformable : un derme et un épiderme

FIG. 5.1 – *Une sphère, deux modélisations possibles*

Le nombre de couches est donc un premier point à déterminer quand nous voulons simuler le comportement dynamiques d'un objet. Mais il faut aussi définir les caractéristiques de chaque couche : doit-on garder l'épiderme et le noyau, ou alors l'épiderme et le derme. Nous verrons, à la section 5.5, comment résoudre un certain nombre des problèmes liés à la détermination des couches.

5.3.1.2 Largeur des couches et rayon des particules

Deux autres paramètres importants pour caractériser le comportement des objets déformables sont l'épaisseur des couches γ et la taille R des particules les composant (cf. équation 4.2).

Le noyau donne l'allure générale de l'objet. La taille des particules le composant n'est donc pas très importante. Plus le rayon est grand, plus le nombre de particules est réduit. Mais, en contrepartie, elles laisseront de grands vides qui devront être compensés par les particules du derme, donc le gain n'est pas toujours conséquent.

La fonction essentielle du derme est d'assurer la déformation. Ainsi, la largeur de cette couche influera sur les possibilités de déformation de l'objet. Si cette couche est peu épaisse, seule des déformations légères seront autorisées. Les déformations plus en profondeur seront empêchées par les grosses particules du noyau. Par contre, si le derme est plus large, l'objet pourra subir des altérations de forme beaucoup plus prononcées. En général, une particule est indéformable ce qui peut être gênant comme nous venons de le voir. Mais elle peut être subdivisée en plusieurs particules appartenant au derme [Amr97] (paragraphe 5.5.2).

L'épiderme assure, entre autres, toutes les relations avec les objets extérieurs. Il faut donc qu'il recouvre tout l'objet. Cette couche doit donc être suffisamment épaisse pour ne pas laisser affleurer des particules du derme, voire du noyau quand le derme est absent. De plus, les particules de l'épiderme doivent être rapprochées pour éviter tout problème au moment du calcul des collisions. En effet, un petit objet risque de se glisser entre les particules de l'épiderme et de se retrouver à l'intérieur de d'un autre objet. Le rayon des particules situées vers l'extérieur est donc dépendant des autres objets de la scène (ainsi que de la précision de reconstruction).

5.3.2 Paramètres de déformation

En plus des caractéristiques liées aux couches, d'autres paramètres de déformation entrent en jeu dans la modélisation des comportements des objets déformables.

En faisant varier uniquement le type de la force entre les particules (Lennard-Jones ou ressorts) ou en modifiant les coefficients caractérisant cette force, il est possible de modifier complètement le type de comportement de l'objet. L'altération de forme est introduite par la rencontre avec des obstacles. Il en résulte un déplacement des particules de l'épiderme. Ensuite, le mouvement est propagé de proche en proche à l'intérieur de l'objet. Les forces internes entre les particules permettent de contrôler le mode de propagation des déformations. Si les interactions sont modélisées par des ressorts (cf. paragraphe 3.2.2), les objets peuvent avoir des comportements variés : élastique, inélastique ou plastique. Le calcul des forces est simplifié par le fait que le voisinage de chaque particule est pratiquement fixe. Celui-ci ne change que lorsque l'on détruit une liaison, par exemple pour simuler une fracture. Le voisinage fixe limite les actions que l'on peut appliquer à un objet. Les changements de topologie sont très lourds à gérer avec un tel formalisme. Pour pouvoir prendre en compte plus facilement les variations de topologie pendant l'animation, nous pouvons modéliser les interactions internes entre les particules par une force d'interaction spatiale (cf. paragraphe 3.2.3).

La modification des paramètres d'amplitude des forces permet de faire varier les caractéristiques du matériau. En effet, plus l'amplitude de la force est grande, plus l'objet modélisé sera résistant à la déformation.

5.4 Passage des paramètres physiques aux paramètres du modèle

Au chapitre précédent, nous avons présenté les paramètres de notre modèle qui influent sur le comportement des objets déformables. Pour obtenir une simulation réaliste, il faut établir une corrélation entre ces paramètres du modèle et les paramètres de l'objet réel. Cette tâche n'est pas aisée, car bien souvent il n'existe pas de relation directe entre tous ces paramètres, ou alors les paramètres de l'objet réel ne sont pas disponibles. C'est le cas des organes internes humains, pour lesquels peu de mesures sont effectuées en ce qui concerne l'élasticité ou les contraintes de déplacement relatives à leur environnement. L'identification des paramètres se fait en deux étapes: la détermination des propriétés d'inertie, et la détermination des propriétés physiques.

5.4.1 Identification des propriétés d'inertie

La conservation des propriétés d'inertie consiste à déterminer la position et la masse de chaque élément du modèle, qui permettent d'obtenir le même comportement que l'objet réel. Dans le cas de système à particule, il suffit de répartir la masse totale sur chaque particule en fonction de son rayon et de la densité volumique. Dans le cas de systèmes masses-ressorts, Joukhadar et al. proposent de décomposer l'objet en un ensemble de tétraèdres [JGL97]. Ils placent ensuite une particule sur chaque sommet, ainsi qu'une particule au centre de tous les tétraèdres. Chaque particule d'un sommet reçoit 1/20 de la masse du tétraèdre, alors que la particule centrale en reçoit les 16/20. Cela leur permet de ne pas sous évaluer le centre d'inertie par rapport à la matrice d'inertie.

5.4.2 Identification des propriétés physiques

Cette étape est beaucoup plus difficile que la précédente, puisqu'il s'agit de déterminer les caractéristiques élastiques, plastiques, de viscosité, etc... de l'objet dont on simule le comportement.

Selon le modèle employé, ces propriétés ne se traduiront pas de la même façon. Ainsi, pour les surfaces implicites, l'élasticité peut être codée dans la dérivée première de la fonction potentielle $f(r)$. La raideur k en un point P sera donc définie par l'équation $k(P) = -f'(r(P))$, où r représente la distance au squelette [Gas93]. Dans le cas des systèmes masses-ressorts, l'élasticité se retrouve dans les coefficients des équations utilisées pour coder les ressorts (linéaires ou non). Pour les systèmes de particules, la distance d'équilibre et les coefficients de la force d'interaction entre les particules permet d'obtenir des comportements plus ou moins élastiques. Ainsi, selon la formulation, un même paramètre de l'objet réel ne seront pas traduits par le même paramètre du modèle, et pas forcement par le même nombre de paramètres. Cela illustre toute la difficulté du passage des paramètres réels aux paramètres du modèle.

Il n'est pas possible de déduire directement les paramètres du modèle à partir des paramètres réels, il faut donc procéder différemment. La méthode la plus utilisée consiste à enregistrer des séquences durant lesquelles un objet réel va se déformer. On dispose alors d'un ensemble de positions successives et de déformations. On simule ensuite les mêmes séquences à l'aide d'un modèle déformable. On mesure l'écart entre l'enregistrement réel et la simulation et l'on ajuste les paramètres jusqu'à obtenir des comportements équivalents. Cette méthode peut être appliquée aux superquadriques [TM91]. Une superquadrique est discrétisée en éléments finis. L'élasticité est alors codée dans une matrice de déformation pour chacun des points discrétisés. Les valeurs des coefficients de la matrice sont calculées à l'aide de résultats expérimentaux. Cette méthode est étendue aux fonctions paramétriques [PMY94]. Cette fois, l'étude de la variation des paramètres permet de comprendre le mouvement de l'objet réel. Joukhadar et al. [Jou96, JGL97] ont proposé une approche générale permettant d'identifier les paramètres physiques. d'un objet. Cette identification consiste à résoudre l'équation suivante :

$$(P_1, P_2, \cdots, P_i, \cdots, P_n) = f^{-1}(\vec{C}_t^d, \vec{F})$$

où les P_i sont les paramètres recherchés, $\vec{C}_t^d$ est l'état désiré (position, vitesse, déformation) et $\vec{F}$ sont les forces extérieures. Il est évident que la fonction f ne peut pas être inversée directement. Ils utilisent donc un algorithme génétique pour déterminer les paramètres P_i qui minimisent $|\vec{C}_t^d - \vec{C}_t|$. Étant donné un comportement désiré, décrit par un ensemble de contraintes (position, orientation, déformation maximale, vitesse, accélération), l'algorithme génétique leur permet de trouver les paramètres qui satisfont ces contraintes. Nous pouvons nous inspirer de cette méthode pour déterminer les paramètres des systèmes de particules.

5.5 Animation adaptative

Les modèles adaptatifs sont de plus en plus utilisés en animation. Ils permettent de raffiner ou de simplifier une représentation en fonction des besoins. Ainsi, on gagne en précision tout en évitant de nombreux calculs inutiles. Witkin et Heckbert ont proposé une méthode d'échantillonnage de surface implicite à l'aide de points se répartissant grâce à des rayons de répulsion adaptatifs [WH94]. Une autre méthode adaptative à été proposé pour simuler des tissus [HPH96]. Un réseau de masses-ressorts est raffiné selon certaines contraintes. Par contre, la méthode ne garantit pas la conservation du comportement physique puisque de nouvelles masses sont créées, alourdissant de fait le tissu. Le raffinement doit se faire en fonction d'un certain nombre de contraintes, qui doivent permettre de conserver les propriétés de l'objet tout au long de l'animation. La suite de cette section est la description de notre approche, qui satisfait ces contraintes.

5.5.1 Systèmes de particules non structurés

Un modèle adaptatif intéressant est celui de DESBRUN [Des97], qui permet des subdivisions ou des simplifications pour des objets hautement déformables. L'idée est de subdiviser automatiquement les particules se trouvant dans les zones de forte variation de pression, et de regrouper celles se trouvant dans les zones de stabilité. Ainsi, une particule i est remplacée par plusieurs petites lorsque la différence de pression avec l'une de ses voisines j est trop grande :

$$|\rho_i - \rho_j|\frac{m_i}{m_j} > \Delta$$

où ρ désigne la densité, et m la masse. La figure 5.2 illustre l'intérêt de la subdivision. De

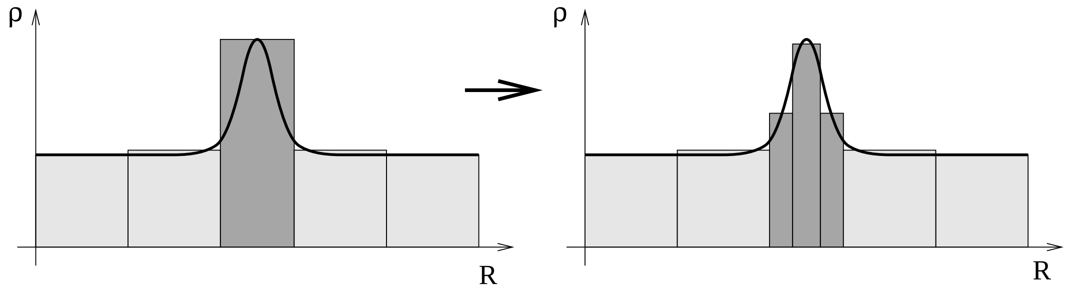


FIG. 5.2 – *Le raffinement est effectué en divisant une particule dans les régions à forte variation de densité*

la même façon, il simplifiera la discrétisation si une région est stable, c.-à-d. si toutes les particules j du voisinage de i vérifient le critère :

$$|\rho_i - \rho_j|\frac{m_i}{m_j} < \delta$$

Il faut aussi vérifier le critère de sphéricité des particules que l'on remplace (Fig. 5.3). Les arrangements sphériques de particules sont détectés grâce au critère suivant :

$$\left| det(I) - \left(\frac{trace(I)}{3}\right)^3 \right| < \varepsilon$$

où I est la matrice d'inertie locale de l'ensemble de particules. Le critère est vérifié quand les trois valeurs propres sont quasiment identiques. Ils présentent donc un algorithme qui permet une adaptation non seulement temporelle mais aussi spatiale. L'utilisateur n'a plus à se soucier du niveau de détail. Mais surtout, cela conduit à une optimisation des calculs qui garantit une meilleure précision en moins de temps.

5.5.2 Systèmes de particules en couches

Lorsque le système de particules est structuré en couches, nous ne pouvons pas nous contenter de subdiviser les particules sur lesquelles s'exercent de trop fortes pressions. En effet, le déplacement des particules, sous l'action de force extérieures ou lors d'une fracture,

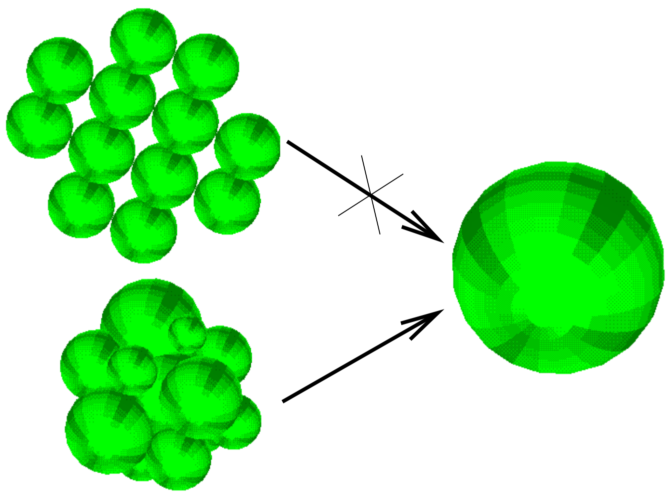


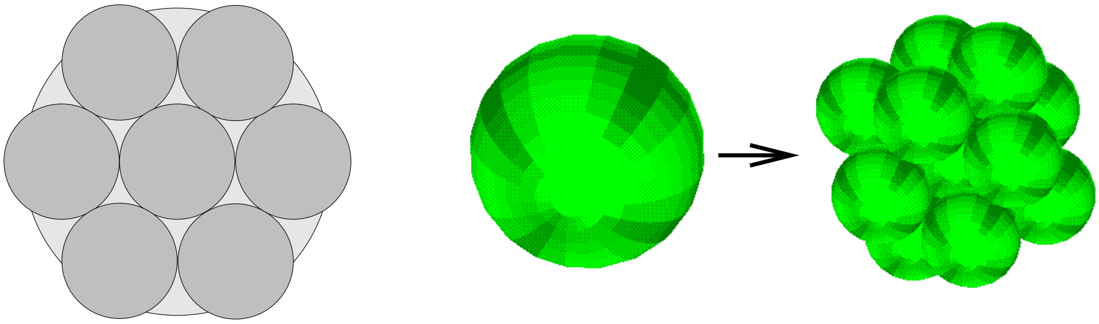
FIG. 5.3 – *Un ensemble de particules est simplifié si il répond au critère de sphéricité*

va engendrer les déformations. Si ces dernières sont suffisamment importantes, il est possible que des particules de l'épiderme entrent en contact avec le noyau, ou que des particules du derme soient en collision avec l'extérieur. La structure de couches va donc être perturbée et sa cohérence ne sera plus maintenue, puisque nous aurons un mélange entre les différentes couches. De plus, les couches internes n'ayant pas les mêmes propriétés ni les mêmes fonctions que l'épiderme, les caractéristiques du système de particules vont être modifiées, ce qui n'est pas acceptable. Nous traitons ce problème en subdivisant certaines grosses particules en de plus petites. Nous conservons ainsi une structure cohérente de couches.

5.5.2.1 Subdivision des particules

La première idée est d'utiliser la méthode de reconstruction vue au chapitre précédent 4. Nous remplissons l'intérieur de la grosse particule par un système de particules de taille inférieure, que nous laissons évoluer. La reconstruction est terminée lorsque nous ne pouvons plus créer de particules et que le système s'est stabilisé. Cependant, cette méthode est très coûteuse en temps de calcul.

Nous préférons donc utiliser une méthode plus géométrique, qui consiste à supprimer la particule et à créer régulièrement n particules à la place [Amr97]. La figure 5.4(a) illustre la nouvelle disposition dans le cas bidimensionnel, avec un pavage hexagonal. La figure 5.4(b) représente la même chose mais dans le cas tridimensionnel. Les petites particules adoptent la position d'énergie minimale dite *cubique face centrée*. Il faut prendre en compte les particules voisines, car il se peut que les nouvelles particules se superposent avec des particules déjà existantes, ce qui entraînerait une instabilité dans le système. Cette méthode présente l'avantage d'être très rapide, il suffit de tester si les nouvelles particules ne sont pas en collision avec les voisines (dont nous connaissons la liste). De plus, elle permet de conserver la stabilité du système.



(a) en 2D

(b) en 3D

FIG. 5.4 – *Ré-échantillonnage d'une particule*

5.5.2.2 Conservation des propriétés physiques et dynamiques

La subdivision doit répondre à certaines contraintes. Il faut par exemple que le système conserve le même centre de gravité et la même densité. Ces conditions garantiront un minimum de perturbation dans le mouvement du système.

Ainsi, les n petites particules et la grosse particule qu'elles remplacent doivent occuper le même volume (ou la même surface en 2D). Nous pouvons facilement déduire le rayon R' des nouvelles particules en fonction de R, celui de l'ancienne :

$$R' = \frac{R}{n^{\frac{1}{D}}}$$

où D est la dimension (2 ou 3). La densité, de surface ou de volume selon la dimension, est conservée en répartissant uniformément la masse m de la particule supprimée entre les n nouvelles, dont la masse est $m' = \frac{m}{n}$. En utilisant ces paramètres pour la subdivision, nous risquons d'introduire une série de particules de tailles différentes des particules déjà existantes dans une des couches. Pour remédier à cela, il faut construire les couches en tenant compte des ces paramètres.

La vitesse $\mathbf{v}$ et l'accélération $\mathbf{a}$ de chaque particule est la même que celle qu'elles remplacent. Nous conservons ainsi toutes les propriétés dynamiques du système, et notamment la quantité de mouvement :

$$m * \mathbf{v} = \sum_{i=1}^{n} (m' * \mathbf{v}) = \mathbf{v} * \sum_{i=1}^{n} m' = \mathbf{v} * m$$

5.5.2.3 Quand subdiviser ?

Maintenant que nous avons défini comment subdiviser, il nous faut déterminer à quel moment le faire. Pour cela nous utilisons principalement les critères suivants :

– en cas de collision d'une particule avec un objet extérieur ;

– si la distance qui sépare deux particules appartenant à des couches différentes franchit un seuil défini par l'utilisateur (quand elles se superposent).
– si deux particules de couches non-consécutives (par exemple le noyau et l'épiderme) sont situées à une distance telle que la couche intermédiaire est trop fine.

De plus, nous imposons qu'une particule ne peut être subdivisée qu'en particules de la couche qui l'entoure. Ainsi, une particule du noyau ne peut pas être directement transformée en particules appartenant à l'épiderme. Il faut auparavant passer par des particules du derme. Cela permet de conserver une structure cohérente.

5.5.2.4 Résultats de la subdivision

La figure 5.5 illustre la méthode présentée pour subdiviser le système de particules tout en conservant la structuration en couches. Un objet multi-couche tombe sous l'action de la gravité sur un obstacle fixe. Les grosses particules se subdivisent, lorsque cela est nécessaire, en plus petites particules qui sont intégrées à la couche inférieure.

5.6 Déformations sous contraintes

Les systèmes de particules sont très simples à mettre en œuvre. Cependant, cette simplicité n'empêche pas la réalisation d'un grand nombre de comportements complexes, qui seraient lourds à gérer avec d'autres types de modélisation. Dans cette section, nous présentons les principales avancées apportées par notre méthode. EN particulier, nous montrons comment respecter la cohérence des couches pour un certain nombre de comportements classiques.

5.6.1 Fracture et recollement

La modélisation des objets déformables à l'aide de systèmes de particules permet de simuler facilement les fractures d'un objet en plusieurs morceaux et les recollements.

Dans le cas où les forces entre les particules sont représentées par des liaisons de type ressort, la déchirure peut être obtenue en détruisant le ressort lorsque l'élongation dépasse un certain seuil. Si la déchirure se propage, l'objet peut se séparer en plusieurs morceaux. Si la déchirure est une opération facile à réaliser, il n'est est pas de même du recollement. Lorsque deux morceaux d'un même objet se rejoignent après s'être séparés, il est quelquefois souhaitable qu'ils fusionnent. Dans ce cas, il faut tester tous les couples formés entre les particules de l'un est l'autre morceau. Parmi tous ces couples candidats, il faut alors choisir ceux qui seront reconnectés en créant un nouveau ressort (et en respectant la structure de couches lorsqu'elle est présente).

Dans le cas d'interactions spatiales de type Lennard-Jones, la fracture et le recollement se font automatiquement. Pour qu'il y ait fracture, il suffit que la force exercée entre deux particules soit suffisamment forte pour les éloigner. De plus, dès qu'elles sortent du champ

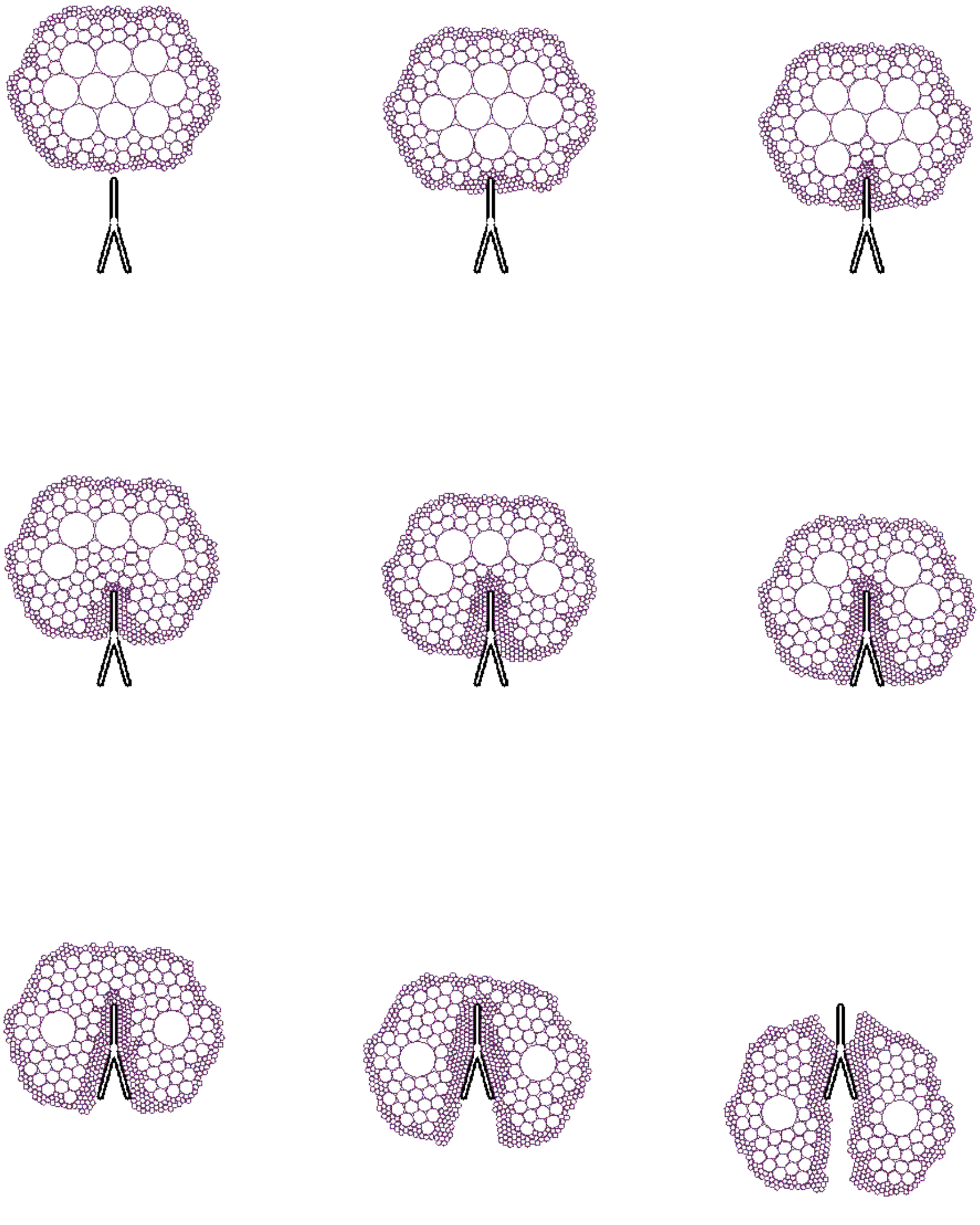


FIG. 5.5 – *Subdivision des particules lorsqu'un objet se sépare en deux*

d'action de l'autre particule, elles ne sont plus en interaction. Les deux morceaux deviennent alors indépendants. Par contre, sitôt qu'un particule d'une partie entre dans le champ d'une particule d'une autre partie, elles s'attirent, et éventuellement elles fusionnent. La fracture et le recollement sont donc réalisés sans aucune opération supplémentaire, à part le regroupement des particules pour recomposer les couches. Par contre, nous pouvons vouloir que deux morceaux ne fusionnent jamais après s'être séparés, mais qu'ils agissent comme deux nouveaux objets et puissent entrer en collision. Pour cela, dès que nous détectons une séparation en une ou plusieurs parties, nous créons un nouvel objet indépendant par partie. Cela peut aussi être réalisé modélisant des liaisons dépendant de la température, et en abaissant celle-ci pour solidifier l'objet et éviter la fusion [Ter89, Ton92].

La figure 5.6 présente différentes itérations d'une barre mono-couche déformable se cassant en tombant sur un obstacle fixe. Les liaisons sont modélisées par une force d'attraction/répulsion.

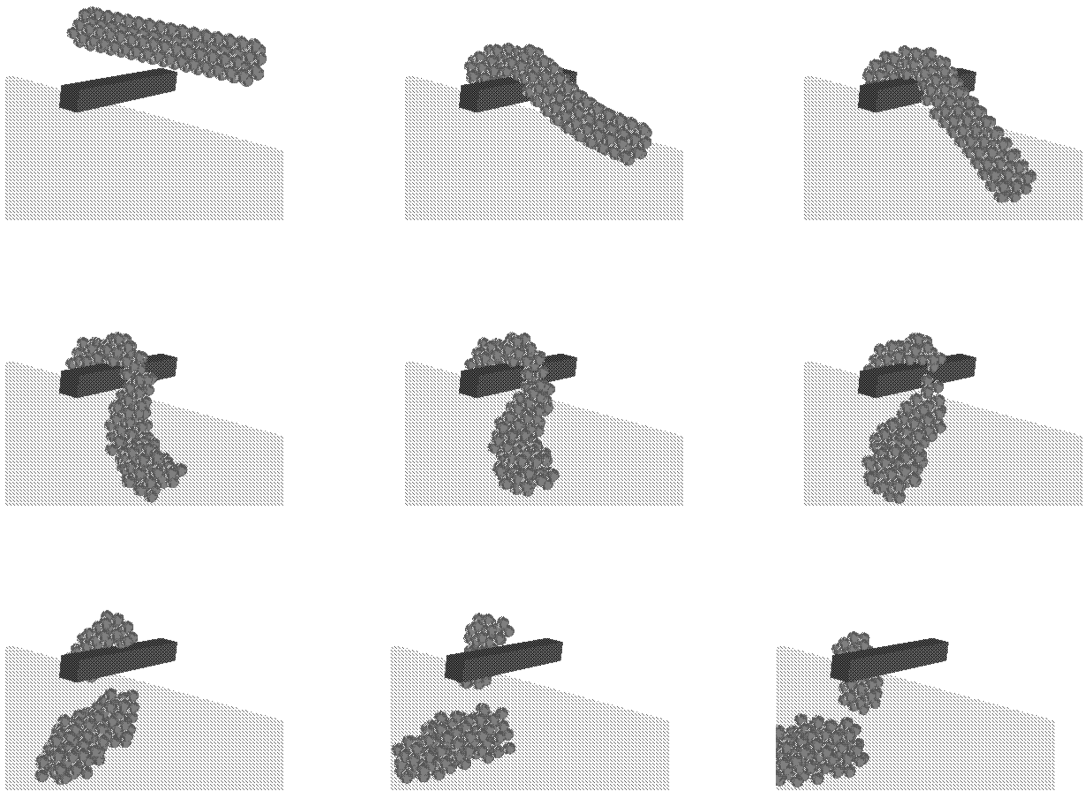


FIG. 5.6 – *Simulation d'une fracture*

5.6.2 Remplissage

Le remplissage et le vidage d'un objet défini par sa surface fermée sont des opérations faciles à mettre en œuvre avec les particules. Il suffit d'ajouter ou d'enlever des particules à l'intérieur de l'objet. La peau peut être modélisée à l'aide de particules orientées [ST92]. Nous pouvons définir de nouvelles contraintes sur la peau, pour obtenir une déformation élastique ou à surface constante. Lors du vidage, nous détruisons des particules. Les particules restantes ont plus de place pour évoluer, des vides apparaissent entre elles. La tension qui était exercée sur la surface diminue, et celle-ci se contracte. Les particules sont repoussées sous l'action de la surface, et le volume occupé diminue. Dans le cas du remplissage, les nouvelles particules repoussent les anciennes, déforment la surface et augmentent le volume. Nous pouvons ainsi créer des objets contenant différentes matières. La figure 5.7 illustre cette opération dans le cas bidimensionnel (et pour une seule couche).

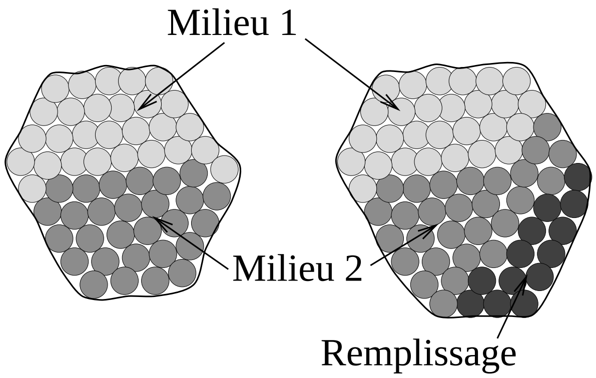


FIG. 5.7 – *Remplissage d'un objet 2D*

5.6.3 Déformation à volume constant

La contrainte de conservation de volume est difficile à résoudre lors de déformation d'un objet. Plusieurs méthodes ont été proposées dans ce cadre [PB88, She95]. PROMAYON et al. [PBP96] ont présenté une approche pour les objets polygonalisés. Les nouvelles positions X sont d'abord calculées sans contraintes. La conservation du volume est assurée en ajoutant une contrainte, ce qui revient alors à résoudre le système :

$$\begin{cases} X_c = X + \varphi \cdot \frac{\partial V(X)}{\partial X} \\ V(X_c) = V^0 \end{cases}$$

où X_c représentent les positions avec contraintes, V le volume et le vecteur projection $\varphi \cdot \frac{\partial V(X)}{\partial X}$ peut être comparé à une force de pression. L'erreur commise à l'aide de cette méthode est très faible, par contre elle nécessite de résoudre un important système de contraintes. DESBRUN et GASCUEL [DG95] proposent une solution pour les surfaces implicites. Ils décomposent l'objet en plusieurs parties et détectent les variations de volume de chaque partie. Ils corrigent ces variations en ajustant localement la fonction implicite.

L'utilisation des systèmes de particules simplifie beaucoup la solution de la contrainte de conservation du volume. En effet, il suffit de modéliser les interactions entre les particules d'un système par une force d'attraction/répulsion pour obtenir une déformation à volume constant. La force de répulsion empêche les particules de se superposer, elles glissent l'une contre l'autre.

La force d'attraction permet de conserver la compacité de l'objet, les particules restent les unes contre les autres. De cette façon, le volume global est conservé. Cette méthode est très simple, elle ne requiert pas de calculs supplémentaires. Notamment, il n'est pas nécessaire de calculer le volume ni de résoudre un système d'équations. Par contre, la conservation du volume n'est pas exacte. Si la somme des volumes occupés par chacune des particules ne varie pas, le volume de l'ensemble des particules est modifié, car l'espace entre celles-ci n'est pas conservé. La figure 5.8 illustre ce problème dans un cas 2D très simple. Bien que les deux formes soient composées du même nombre de particules, ces dernières étant pourtant tangentes, elles n'occupent pas la même surface. Par contre, en pratique, lors de la déformation

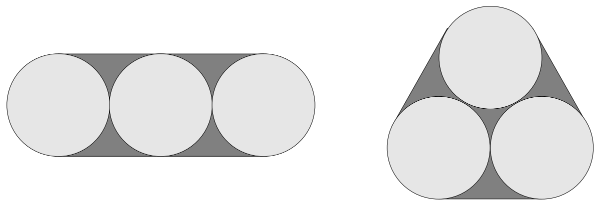


FIG. 5.8 – *Problème de conservation de la surface*

d'un objet comportant un grand nombre de particules, la variation de volume est assez faible. Lorsque les particules se déplacent, elles créent des espaces libres entre-elles et pour compenser d'autres espaces vides disparaissent.

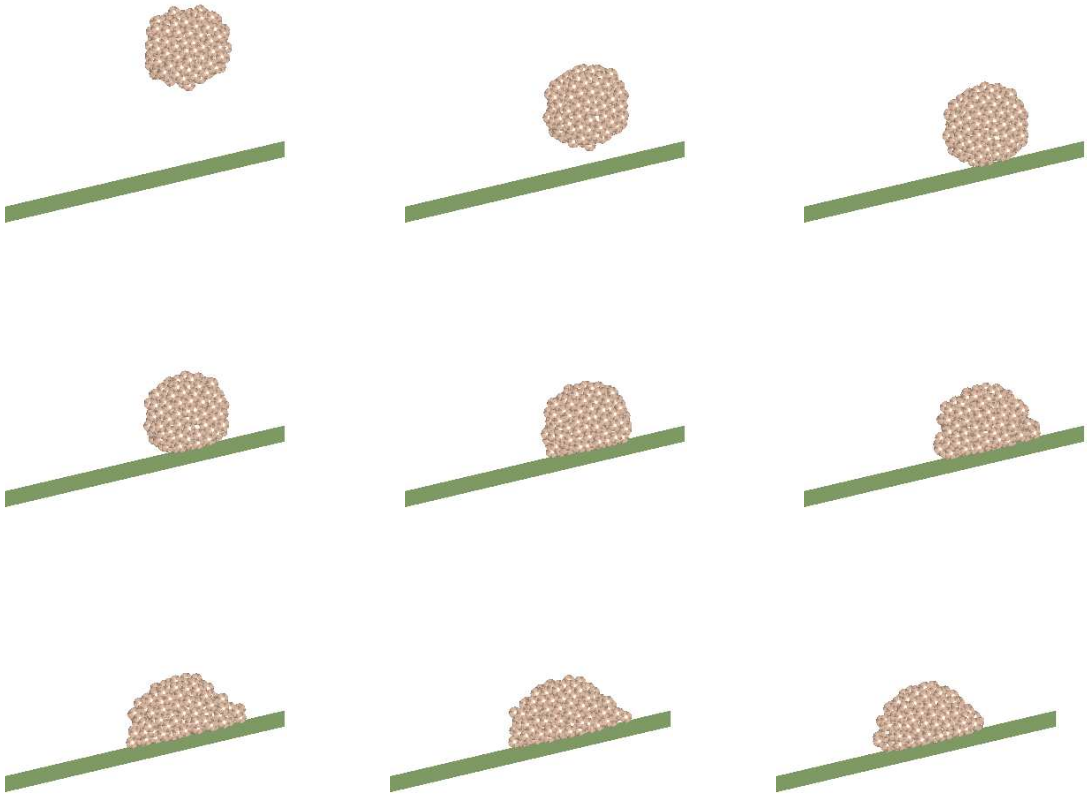


FIG. 5.9 – *Simulation d'une déformation avec conservation de volume*

6 Contributions

Dans cette partie, nous avons présenté une méthode de modélisation et d'animation d'objets déformables. Nous résumons ici les principaux apports de notre méthode.

A partir d'une description surfacique d'un objet, nous reconstruisons automatiquement un modèle de particules volumiques constitué de plusieurs couches. Les avantages d'un tel modèle sont multiples. Tout d'abord, il est préférable de disposer d'une représentation du volume plutôt que de la surface de l'objet. Cela simplifie le traitement des comportements de volume, tels que la résistance interne (par rapport à la résistance superficielle). Cela permet de générer des déformations à volume quasi constant, de façon très simple, sans aucun calcul supplémentaire. Ensuite, les systèmes de particules sont un outil de modélisation parfaitement bien adapté aux déformations. La simplicité des relations entre les particules permet de décrire aisément et intuitivement le type de matière que nous allons animer.

Il est à noter que la reconstruction ainsi que la simulation utilisent les mêmes outils algorithmiques de gestion des particules. Le calcul des forces entre les particules ou avec des obstacles, ou bien la détermination des nouvelles positions des objets, se font de manière similaire. Cela implique une grande compatibilité. Les objets reconstruits peuvent être immédiatement animés, sans aucun autre pré-traitement. D'autre part, un certain nombre d'objets peut être transformé en sa représentation particulaire, ce qui facilite la gestion des interactions entre les divers objets de la scène. Mais, bien évidemment, il est possible de prendre en compte l'interaction entre un système de particules et d'autres types de représentation, tels que implicite ou paramétrique.

Les systèmes de particules mono-couches se sont avérés très coûteux en temps de calcul. Nous avons donc évolué vers un système en plusieurs couches. L'intérêt est de réduire le nombre de particules, en spécialisant chaque couche selon son rôle, et en conservant une tolérance d'erreur imposée par l'application. Nous conservons la même précision, car les efforts sont répartis là où ils sont vraiment nécessaires. Nous réduisons ainsi considérablement le nombre d'interactions internes puisque chaque couche n'est en relation qu'avec les couches voisines. Le interactions entre les objets diminuent elles aussi, car seule la couche la plus externe est prise en compte. Durant l'animation, pour préserver la cohérence de notre système multi-couches, nous avons proposé une approche adaptative permettant la subdivision ou le regroupement de particules selon l'état de leur voisinage. Cette approche adaptative est encore peu développée en animation, mais devrait rapidement se généraliser.

Conclusion générale

Dans cette thèse, deux approches de modélisation sont présentées :

- surfacique en produit tensoriel pour les objets rigides ;
- à l'aide de systèmes de particules pour les objets déformables.

Dans ce cadre, nous avons proposé des algorithmes de reconstruction et d'animation. Ceux-ci ont été appliqués avec succès dans un contexte médical : nos programmes ont été intégrés dans un logiciel de traitement par radiothérapie au Christie Hospital de Manchester (GB).

Dans la première partie de ce mémoire, nous nous sommes donc intéressés à la reconstruction de surfaces sur un ensemble de contours. Nous nous sommes inspirés de la méthode du « skinning » appliquée au modèle B-spline, bien adaptée à notre application. Malgré les avantages indéniables des modèles paramétriques et de leur aspect lisse, ceux-ci posent de nombreux problèmes auxquels nous avons apporté une solution originale dans les domaines suivants :

- les approches classiques engendrent un grand nombre de nœuds lors de la phase de mise en compatibilité. Cela implique un processus de suppression de nœuds qui est très coûteux en temps de calcul. Nous avons proposé, en échange, une approche différente qui permet de générer un nombre de nœuds restreint durant cette phase de mise en compatibilité ;
- les objets créés à l'aide des surfaces B-splines sont généralement des surfaces ouvertes. Le deuxième apport a consisté en un ensemble de traitements pour la fermeture de ces surfaces et pour améliorer leur qualité, en ajoutant des contraintes de continuité. Notamment, les surfaces B-splines sont étendues aux extrémités de l'objet pour pouvoir modéliser des formes homéomorphes à une sphère (très courantes dans les applications médicales). Ainsi, nous obtenons rapidement, dans la plupart des cas, une représentation compacte et lisse des objets ouverts ou fermés ;
- nous avons aussi proposé une méthode pour prendre en compte les surfaces possédant des branches, qui posent de sérieux problèmes aux méthodes paramétriques classiques. Notre idée est de reconstruire chaque branche avec la méthode du « skinning ». La zone d'embranchement est ensuite remplie à l'aide de carreaux de COONS. Nous nous basons sur l'étude topologique de la zone d'embranchement pour déterminer le nombre de carreaux ainsi que les frontières de chacun d'eux. Trois types d'embranchements sont pris en compte : crête, selle de cheval et canyon. Grâce à un reparamétrage non-uniforme des courbes frontières, les surfaces de raccordement possèdent une continuité G^1, mais

surtout elles ne présentent pas de plis, comme cela peut se produire avec les carreaux de COONS.

Cependant, les représentations paramétriques sont très limitées dès que nous voulons prendre en compte des formes complexes ou les comportements dynamiques des objets modélisés.

Dans la seconde partie de ce mémoire, nous présentons donc la généralisation de la reconstruction aux objets déformables. Après une étude des modèles existants pour représenter et animer de tels objets, nous avons évolué vers les systèmes de particules. La simplicité de ce modèle, ainsi que son utilisation intuitive (dérivée des lois physiques), en font un outil adapté pour la simulation dynamique. La richesse de ces modèles pour prendre en compte différents comportements physiques des objets, et notamment les changements de topologie lors de l'animation, est un autre atout qui nous a orienté dans le choix de ce modèle. Un des apports de cette thèse a été de fournir une méthode de reconstruction entièrement automatique des objets déformables, alors que la plupart des méthodes existantes comportent toujours une étape interactive. Cependant, obtenir une représentation d'un objet avec une grande précision requiert un nombre élevé de particules. Même si les algorithmes développés pour l'animation sont de complexité $o(n.log(n))$, le grand nombre de particules n a pour conséquence un temps de calcul important pour l'animation. Nous avons donc proposé un modèle de particules structuré en plusieurs couches. Cette structuration est une des contributions majeures de notre travail :

- dans le domaine de la reconstruction, les systèmes de particules multicouches ont permis de réduire grandement le nombre de particules nécessaires pour représenter les objets. Le gain est estimé à 75% par rapport aux systèmes monocouches, pour une précision équivalente ;
- dans le cadre de l'animation, les couches sont séparées et spécialisées, chacune a un rôle bien déterminé, ce qui contribue à réduire la complexité. Les interactions intérieures se font uniquement entre particules appartenant à des couches voisines. De plus, seules les particules de la peau peuvent être en interaction avec les objets extérieurs. Nous limitons ainsi le nombre de tests pour détecter et gérer les collisions.

Cependant, la structuration en couches va engendrer des problèmes supplémentaires. Notamment, lors de l'animation, la cohérence du système doit être conservée, c'est-à-dire qu'il faut éviter les mélanges entre les couches, ainsi que l'amincissement d'une couche jusqu'à sa disparition. Nous avons donc proposé un certain nombre d'algorithmes nouveaux, qui assurent la conservation des couches lors d'une collision entre objets déformables, d'un remplissage ou encore d'une fracture ou d'une fusion.

Perspectives

Le domaine de la reconstruction d'objets possédant des branches à l'aide des surfaces en produit tensoriel, reste encore très ouvert. Si une solution au cas des bifurcations (2 branches) a été présentée dans ce mémoire, il n'en est pas de même pour les embranchements plus complexes qui nécessitent des études complémentaires. Nous avons présenté quelques idées pour généraliser notre méthode, mais certains problèmes restent non résolus. Par exemple, la détermination automatique des différents types (selle de cheval, crête ou canyon) n'est pas une tâche aisée dans le cas de branches multiples.

Une autre lacune des modèles paramétriques est la fermeture aux extrémités. Dans bien des cas, une simple extension de la surface vers un seul point n'est pas suffisante. Des plis se forment et quelquefois la surface peut même s'auto-intersecter lorsque les contours sont fortement convexes ou dissimilaires. Une des solutions est d'étudier la forme géométrique des contours aux extrémités, et de fermer la forme, non plus sur un seul point, mais sur une courbe telle que les contours soient étoilés par rapport à elle, qui peut être proche du squelette au sens de Voronoï. Cela pose le problème de l'égalité des plans tangents le long de la courbe pour garantir une continuité G^1 aux extrémités.

D'autre part, les systèmes de particules permettent de modéliser des formes complexes aisément. Cependant, le temps de calcul est considérable. Nous avons réduit la complexité par l'utilisation de boîtes englobantes et de partitions de l'espace. Néanmoins, le temps de calcul reste encore élevé, le recours au calcul parallèle paraît nécessaire. Dans le cadre d'un projet de fin d'études, nous avons abordé la parallélisation de l'algorithme de reconstruction multi-couche. Cela n'est pas une tâche aisée, même si la distribution spatiale des particules permet d'affecter un ensemble de particules à chaque processeur. Les premiers résultats semblent prometteurs, il est cependant nécessaire de prévoir un transfert de particules entre différents processeurs et un équilibrage des charges, afin d'obtenir une animation en temps réel.

La structuration en couche des systèmes de particules que nous utilisons permet de simplifier certains processus de la simulation. La conservation de l'intégrité des couches est une contrainte importante lors de la simulation des déformations. Nous avons proposé un certain nombre de solutions, et les résultats sont tout à fait satisfaisants, notamment grâce à la subdivision. Les particules ne sont plus des objets rigides, mais sont capables de se diviser de façon adaptative. Ce mécanisme est très intéressant, et il mérite d'être approfondi : il s'agit de le rendre plus efficace (en temps de calcul), mais aussi de développer l'opération inverse, le regroupement de petites particules en une plus grosse, opération pour laquelle nous avons seulement donné une ébauche de solution.

A Reconstruction de courbes

Dans cette annexe, nous décrivons les formules et les méthodes nécessaires à la reconstruction des courbes B-splines, ouvertes ou fermées, à partir d'un ensemble de points disposés sur un contour. De plus amples détails peuvent être trouvés dans de nombreux ouvrages, dont [PT96, Leo91, Béz87].

A.1 Approximation au sens des moindres carrés

Nous supposons que nous avons un ensemble de r points $\{\vec{Q}_k\}$, $k = 0, \ldots, r-1$, que nous voulons approximer à l'aide d'une courbe B-spline de degré $p \geq 1$. Nous recherchons donc l'ensemble $\{\vec{P}_i\}$, $i = 0, \ldots, n-1$ des points de contrôle de cette courbe qui admet pour équation :

$$\vec{C}(u) = \sum_{i=0}^{n-1} \vec{P}_i N_i^p(u) \text{ pour } u \in [0,1] \tag{A.1}$$

Nous pouvons réécrire l'équation A.1 pour chaque point $\vec{Q}_k$, cela de manière matricielle $[Q] = [N][P]$ où $[Q]$ est un vecteur de dimension r représentant les données initiales, $[P]$ est le vecteur de dimension n contenant les points de contrôle recherchés, et $[N]$ est la matrice de dimension $r \times n$, qui peut s'écrire ainsi :

$$[N] = \begin{bmatrix} N_0^p(\bar{u}_0) & \ldots & N_{n-1}^p(\bar{u}_0) \\ \vdots & \ddots & \vdots \\ N_0^p(\bar{u}_{r-1}) & \ldots & N_{n-1}^p(\bar{u}_{r-1}) \end{bmatrix} \tag{A.2}$$

avec :

$$\begin{aligned} N_i^0(u) &= \begin{cases} 1 & \text{si } u_i \leq u < u_{i+1} \\ 0 & \text{sinon} \end{cases} \\ N_i^p(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_i^{p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1}^{p-1}(u) \end{aligned} \tag{A.3}$$

En général, la matrice $[N]$ n'est pas carrée ($r > n$), et le système doit être résolu à l'aide de la méthode standard de l'approximation au sens des moindres carrés. Ce qui conduit aux conditions suivantes pour les valeurs de paramètres affectées aux $\{\vec{Q}_k\}$:

$$\sum_{k=0}^{r-1} \left| \vec{Q}_k - \vec{C}(\bar{u}_k) \right|^2 \text{ est minimum} \tag{A.4}$$

La courbe obtenue ne passe, en général, pas exactement par les points $\vec{Q}_k$, et $\vec{C}(\bar{u}_k)$ n'est pas le point de $\vec{C}(u)$ le plus proche de $\vec{Q}_k$. Le nouveau système peut alors s'écrire comme suit :

$$[P] = \left[[N]^T [N]\right]^{-1} [N]^T [Q] \tag{A.5}$$

Ce système est quelquefois noté $[P] = \left[[N]^T [N]\right]^{-1} [R]$, avec $[R] = [N]^T [Q]$. La résolution de l'équation A.5 fournit les points de contrôle de la courbe. Pour calculer $[N]$ (Eq. A.2), il faut fournir un vecteur nodal $U = \{u_0, \dots, u_{n+p}\}$ et les paramètres $\{\bar{u}_k\}$ affectés aux données $\{Q_k\}$. Leur choix affecte grandement la forme et la paramétrisation de la courbe. La méthode la plus utilisée, et généralement bien adaptée même quand les données sont réparties non uniformément, est la méthode de la longueur de corde.

$$\text{Si } d = \sum_{k=1}^{n-1} \left|\vec{Q}_k - \vec{Q}_{k-1}\right|$$

est la longueur totale de la corde, alors :

$$\begin{aligned} &\bar{u}_0 = 0 \text{ et } \bar{u}_{n-1} = 1 \\ &\bar{u}_k = \bar{u}_{k-1} + \frac{\left|\vec{Q}_k - \vec{Q}_{k-1}\right|}{d} \text{ pour } k = 1, \dots, n-2 \end{aligned} \tag{A.6}$$

Des méthodes d'optimisation existent pour paramétrer un ensemble de points aléatoirement disposés le long d'une courbe, en vue d'une approximation au sens des moindres carrés [MK95, SM91].

Les nœuds doivent aussi refléter la distribution des données. Ils sont définis de la façon suivante :

$$\begin{aligned} &u_0 = \dots = u_p = 0 \text{ et } u_n = \dots = u_{n+p} = 1 \\ &u_{i+p} = (1-\alpha)\bar{u}_j + \alpha\bar{u}_{j+1} \text{ pour } i = 1, \dots, n-p-1 \end{aligned} \tag{A.7}$$

où $\alpha = j * dist - i$ avec $dist = r/(n-p)$ et j est la partie entière de $i * dist$. Ceci garantit que chaque intervalle de nœud contient au moins un paramètre $\bar{u}_k$, et il a été prouvé [Boo78] que l'équation A.5 est positive et bien conditionnée. Nous pouvons alors la résoudre par une méthode de Gauss sans pivot.

A.2 Interpolation

Nous supposons cette fois que nous disposons d'un ensemble de n points $\{\vec{Q}_k\}$ que nous voulons interpoler, c.-à-d. que la courbe passera exactement par tous les points. Comme dans la section précédente concernant l'approximation, nous déterminons un paramétrage $\{\bar{u}_k\}$, ainsi qu'un vecteur nodal $U = \{u_0, \dots, u_{n+p}\}$. En réécrivant l'équation A.1 pour chaque point $\vec{Q}_k$, on obtient un système de n équations linéaires à n inconnues, qui sont les points de contrôle de la courbe. Ce qui peut s'exprimer sous forme matricielle par $[Q] = [N][P]$. Le choix du paramétrage et du vecteur nodal est là encore primordial. En utilisant l'équation A.6, nous affectons un paramétrage à chaque point initial $\vec{Q}_k$. Le vecteur de nœuds peut être obtenu en faisant une sorte de moyenne des paramètres de la façon suivante :

$$\begin{aligned} u_0 &= \ldots = u_p = 0 \text{ et } u_n = \ldots = u_{n+p} = 1 \\ u_{i+p} &= \frac{1}{p}\sum_{j=i}^{i+p-1} \bar{u}_j \text{ pour } i = 1, \ldots, n-p-1 \end{aligned} \tag{A.8}$$

Les nœuds sont ainsi définis en fonction de la distribution des données le long de la courbe. De plus, une telle répartition assure que le système $[Q] = [N][P]$ est totalement positif et qu'il peut être résolu par une méthode de Gauss sans pivot [Boo78].

A.3 Passage aux courbes fermées périodiques

Dans le cas où l'ensemble des points initiaux forme une courbe fermée, il est difficile, avec une courbe ouverte, d'obtenir une continuité élevée aux extrémités. Par exemple, pour une courbe de degré 2, obtenir une continuité C^1 au raccord impose d'avoir au moins trois points de contrôle alignés.

Nous utilisons donc les B-splines périodiques qui permettent la construction de courbes fermées avec une continuité C^{p-1} (si la courbe est de degré p) sur tout leur intervalle de définition. La périodicité est obtenue par transformation du vecteur de nœuds. La séquence nodale d'une courbe ouverte, qui est de la forme :

$$\underbrace{u_0 = \ldots = u_p}_{\text{multiplicité } (p+1)}, u_{p+1}, \ldots, u_{n-1}, \underbrace{u_n = \ldots = u_{n+p}}_{\text{multiplicité } (p+1)}$$

est transformée en une séquence nodale de période $(u_n - u_p)$, tous les nœuds étant de multiplicité égale à 1, à l'aide les relations suivantes :

$$\left\{ \begin{array}{lll} u'_{p-1-i} & = u_p - (u_n - u_{n-1-i}) & \text{pour } i = 0, \ldots, p-1 \\ u'_{i+p} & = u_{i+p} & \text{pour } i = 0, \ldots, n-p \\ u'_{n+1+i} & = u_n + (u_{i+p+1} - u_p) & \text{pour } i = 0, \ldots, p-1 \end{array} \right. \tag{A.9}$$

Cette séquence traduit les conditions de fermeture de la courbe, mais il est nécessaire de rajouter p équations sur les points de contrôle pour assurer une continuité C^{p-1} :

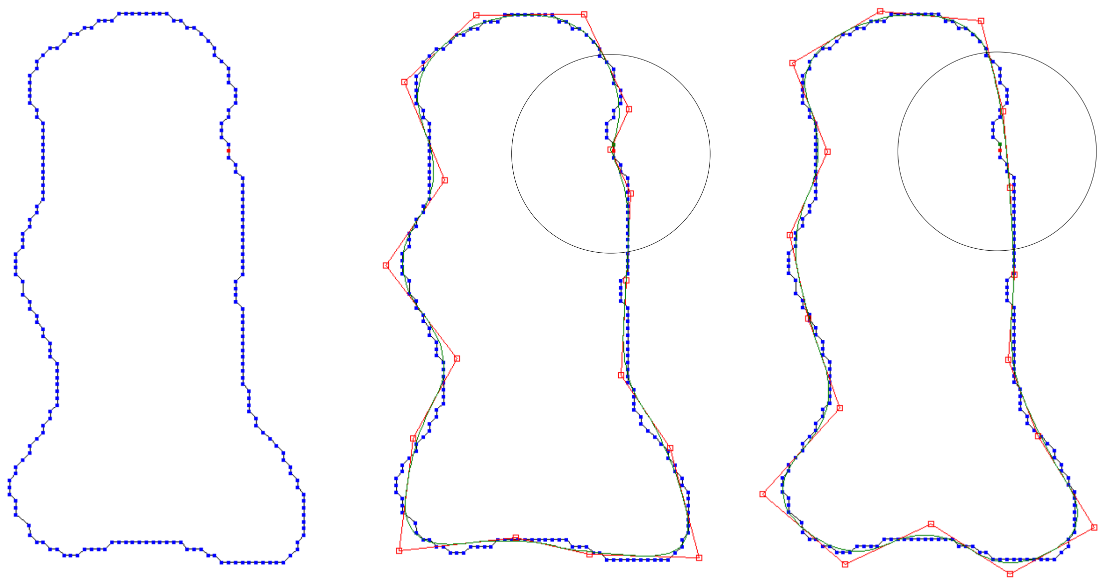
$$\vec{P}_i = \vec{P}_{i+n-p} \text{ pour } i = 0, \ldots, p-1$$

Ensuite, nous les insérons dans le système pour calculer les points de contrôle de la courbe à reconstruire.

La figure A.1 montre la différence entre une courbe ouverte et une courbe périodique fermée pour reconstruire le même ensemble de points.

A.4 Prise en compte de l'erreur

Dans le cas de l'approximation, une erreur est commise lors de la reconstruction au sens des moindres carrés. De manière générale, le nombre de points de contrôle requis pour obtenir



(a) Contour initial (b) Courbe ouverte (c) Courbe fermée

FIG. A.1 – *Différence entre courbes ouverte et fermée*

la précision voulue ne peut pas être calculé à l'avance. Une solution itérative doit donc être employée. Nous utilisons l'algorithme suivant [PT96] :

Algorithme A.1

1 commencer avec un nombre minimal de points de contrôle ;

2 approximer la courbe, résoudre l'équation A.5 ;

3 calculer l'erreur commise ;

4 **si** l'erreur est inférieure à la précision souhaitée, arrêter le processus itératif, **sinon** augmenter le nombre de points et retourner à l'étape 2.

Il est a noté que cette méthode ne converge pas obligatoirement, il qu'il faut en tenir compte lors de l'implémentation. Une solution peut être trouvée dans les splines de subdivision [FB95] ou dans les splines multirésolution [WSS+98].

Cet algorithme fournit le nombre optimal de points de contrôle, mais un grand nombre de courbes doivent être reconstruites et le calcul de l'erreur est souvent très long. En effet, la déviation entre la courbe et les données est mesurée à l'aide de l'une ou l'autre des formules suivantes (Eq A.10 et A.11) :

$$E_1 = \max_{k=0,\ldots,r-1} \left| \vec{Q}_k - \vec{C}(\bar{u}_k) \right| \tag{A.10}$$

$$E_2 = \max_{k=0,\ldots,r-1} \left(\min_{u\in[0,1]} \left| \vec{Q}_k - \vec{C}(\bar{u}) \right| \right) \tag{A.11}$$

L'équation A.10 correspond au maximum des distances entre chaque point initial et le point de la courbe pour le paramètre associé. Elle est beaucoup plus simple à calculer mais $E_1 \geq E_2$, et le nombre de points générés sera plus élevé. Si nous voulons obtenir le nombre optimal de points de contrôle, il nous faut déterminer E_2, ce qui revient à calculer la distance minimale entre chaque point $\vec{Q}_k$ et la courbe $\vec{C}(u)$. Nous utilisons la méthode itérative de NEWTON pour trouver le paramètre u_i tel que $\left|\vec{Q}_k - \vec{C}(u_i)\right|$ soit minimum. Soit la fonction (cf. fig. A.2) :

$$f(u) = \vec{C}'(u) \cdot \left(\vec{C}(u) - \vec{Q}_k\right) \tag{A.12}$$

La distance entre les points $\vec{Q}_k$ et $\vec{C}(u)$ est minimale lorsque $f(u) = 0$, c'est la condition d'orthogonalité (Fig. A.3). Le paramètre à l'itération suivante u_{i+1} peut donc se déduire

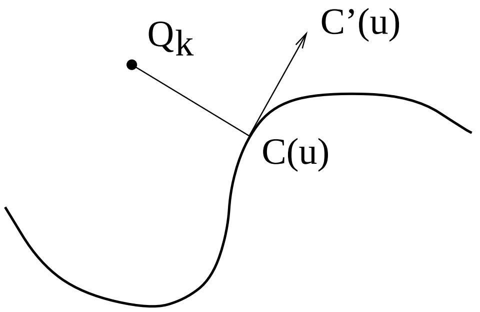


FIG. A.2 – *Notation pour le calcul de E_2*

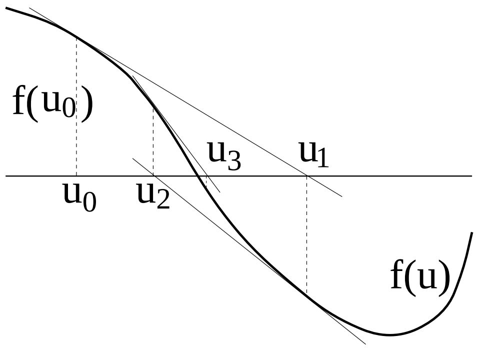


FIG. A.3 – *Méthode itérative de* NEWTON

selon l'équation A.13 :

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)} = \frac{\vec{C}'(u_i) \cdot \left(\vec{C}(u_i) - \vec{Q}_k\right)}{\vec{C}''(u_i) \cdot \left(\vec{C}(u_i) - \vec{Q}_k\right) + \left|\vec{C}'(u_i)\right|^2} \tag{A.13}$$

L'algorithme utilisé pour le calcul de E_2 est le suivant :

Algorithme A.2

1 prendre $\bar{u}_k$ la valeur de longueur de corde associée à $\vec{Q}_k$ comme paramètre initial ;

2 calculer l'erreur sur la distance entre les points $\left|\vec{Q}_k - \vec{C}(u_i)\right| \leq e_1$;

3 calculer l'erreur commise sur l'angle entre le projeté de $\vec{Q}_k$ sur la courbe et $\vec{C}(u_i)$

$$\frac{\left|\vec{C}'(u_i) \cdot \left(\vec{C}(u_i) - \vec{Q}_k\right)\right|}{\left|\vec{C}'(u_i)\right| \left|\vec{C}(u_i) - \vec{Q}_k\right|} \leq e_2$$

4 **si** une des conditions 2 ou 3 n'est pas satisfaite **alors** calculer u_{i+1} à l'aide de l'équation A.13 **sinon** sortir.

5 vérifier que la variation de paramètre est encore significative $\left|(u_{i+1} - u_i)\vec{C}'(u_i)\right| \leq e_1$;

6 **si** l'une des conditions 2, 3 ou 5 est remplie **alors** sortir, **sinon** retourner en 2.

e_1 est une mesure de distance, et e_2 est une mesure de nullité du cosinus. ce sont deux paramètres fixés par l'utilisateur. L'algorithme A.2 est exécuté pour chaque point initial $\vec{Q}_k$ et le maximum des erreurs commises donne l'erreur pour l'ensemble de la courbe.

B Carreaux de Coons

Dans cette annexe, nous décrivons les définitions relatives aux carreaux de Coons [Béz87, Far93, PT95]. Ils ont été développés par Coons en 1966, pour résoudre le problème suivant : sur un objet, on a tracé un quadrillage, définir les points intérieurs des carreaux pour que soit assuré entre ceux-ci un raccordement tangentiel (la continuité des courbures sera obtenue par la suite).

B.1 Définitions

Les carreaux de Coons sont des surfaces bi-paramétriques qui permettent de remplir l'intérieur d'un carreau définit à l'aide de quatre courbes frontières, définies comme :

$$C_k(u) = \sum_{i=0}^{m} N_i^p(u) P_{ki} \quad k = 0,1 \;\; u \in [0,1]$$

$$C_l(v) = \sum_{j=0}^{n} N_j^q(v) P_{lj} \quad l = 0,1 \;\; v \in [0,1] \tag{B.1}$$

Ces quatre courbes doivent respecter certains critères de compatibilité, notamment à leurs extrémités (Figure B.1) :

$$\begin{aligned} S_{00} = \; C_0(u=0) &= C_0(v=0) \\ S_{10} = \; C_0(u=1) &= C_1(v=0) \\ S_{01} = \; C_1(u=0) &= C_0(v=1) \\ S_{11} = \; C_1(u=1) &= C_1(v=1) \end{aligned} \tag{B.2}$$

Remarque : Nous ne restreignons pas la définition des deux courbes $C_k(u)$ ou $C_l(v)$ sur la même base nodale.

La position d'un point à l'intérieur du carreau de Coons peut se calculer à l'aide de l'équation B.3 :

$$\begin{aligned} S(u,v) = \; & C_0(u).g_0(v) + C_1(u).g_1(v) \\ & + C_0(v).f_0(u) + C_1(v).f_1(u) \\ & - S(0,0).f_0(u).g_0(v) - S(0,1).f_0(u).g_1(v) \\ & - S(1,0).f_1(u).g_0(v) - S(1,1).f_1(u).g_1(v) \end{aligned} \tag{B.3}$$

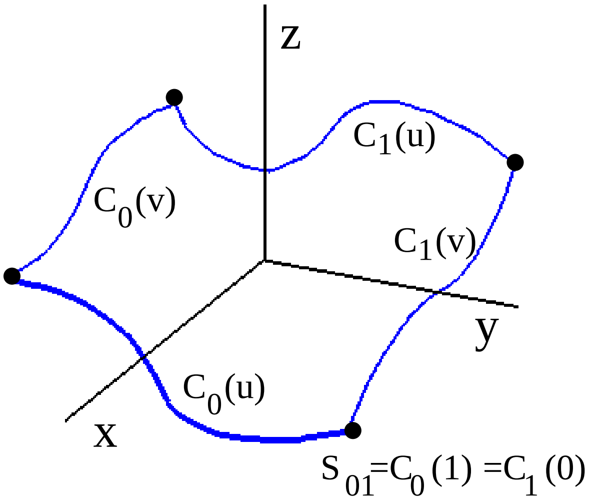


FIG. B.1 – *Courbes frontières d'un carreau de* COONS

où f_0, f_1 et g_0, g_1 représentent les fonctions de mélange. Nous choisissons les fonctions $f_0(t) = g_0(t) = t$ et $f_1(t) = g_1(t) = 1 - t$ mais n'importe quelles fonctions respectant les critères suivants peuvent convenir :

- $f_0(t) = 1 - f_1(t)$ et $g_0(t) = 1 - g_1(t)$ qui permet de générer des combinaisons de points barycentriques ;
- $f_0(0) = g_0(0) = 1$ et $f_0(1) = g_0(1) = 0$ pour obtenir une véritable interpolation.

Nous pouvons aussi définir le carreau de COONS comme la composition de trois surfaces :

$$S(u, v) = R_1(u, v) + R_2(u, v) - T(u, v)$$

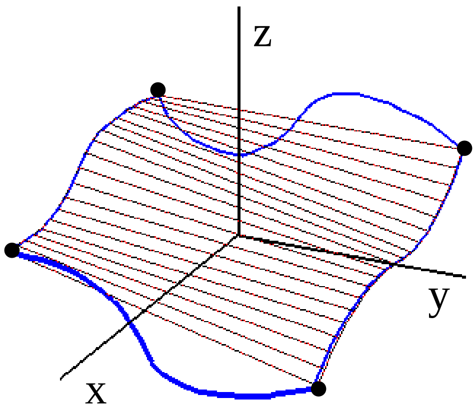
où $R_1(u, v)$ et $R_2(u, v)$ sont des surfaces réglées entre $C_k(u)$ et $C_l(v)$, respectivement. $T(u, v)$ est la surface obtenue par le produit tensoriel suivant :

$$T(u, v) = \begin{bmatrix} 1 & u \end{bmatrix} \begin{bmatrix} S_{00} & S_{01} \\ S_{10} & S_{11} \end{bmatrix} \begin{bmatrix} 1 \\ v \end{bmatrix} \tag{B.4}$$

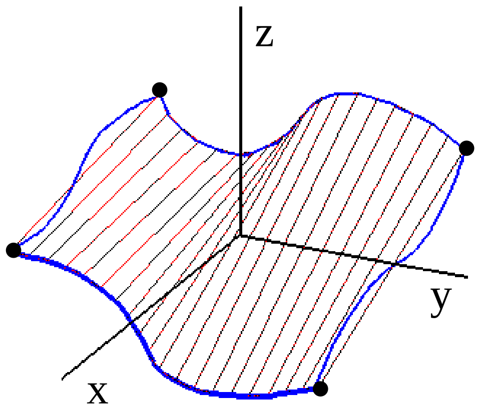
La figure B.2 montre la construction du carreau de COONS à partir des courbes de la figure B.1. Les figures B.2(a) et B.2(b) montrent la surface réglée dans les direction u et v, respectivement, alors que la figure B.2(c) montre la surface bilinéaire définie par les quatre sommets du carreau. Le résultat final, le carreau de COONS, est illustré figure B.2(d).

B.2 Problème de la continuité

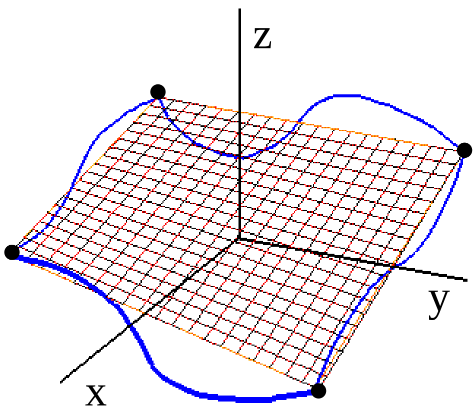
Dans le cas général, deux carreaux adjacents auront seulement une continuité C^0 le long de leur frontière commune. Pour obtenir une continuité plus élevée, COONS a développé des



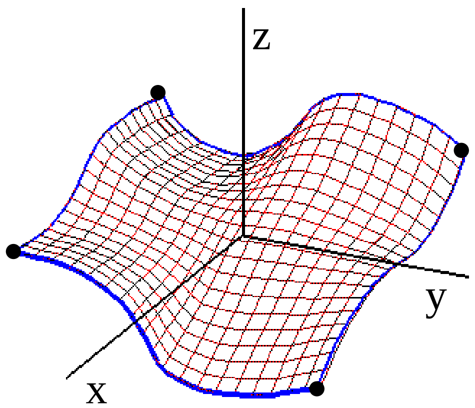
(a) Surface réglée en u



(b) Surface réglée en v



(c) Surface bilinéaire



(d) Surface finale

FIG. B.2 – *Reconstruction d'un carreau de* COONS carreaux d'ordre plus élevé en utilisant les champs dérivés (Fig. B.3) :

$$D_k(u) = \sum_{i=0}^{m} N_i^p(u) Q_{ki} \quad k = 0, 1 \;\; u \in [0, 1]$$

$$D_l(v) = \sum_{j=0}^{n} N_j^q(v) Q_{lj} \quad l = 0, 1 \;\; v \in [0, 1] \tag{B.5}$$

Le but est de construire une surface entre les courbes frontières, équation (B.1), et qui respecte

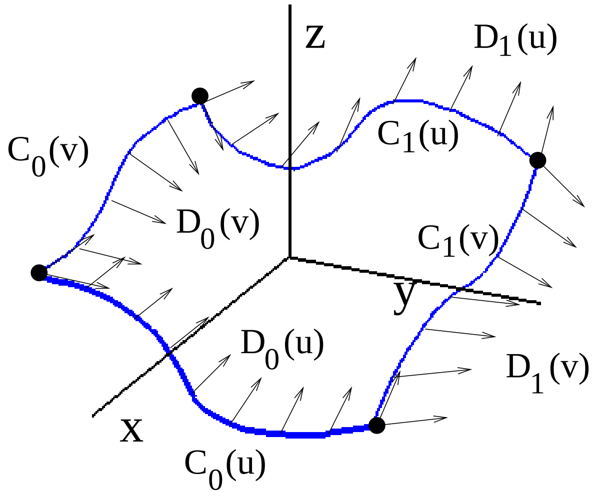


FIG. B.3 – *Courbes frontières et champs dérivés*

les dérivées premières définies à l'équation (B.5). Bien entendu, les quatre courbes frontières doivent respecter les conditions de compatibilité vues lors de la section précédente B.1. En plus, les vecteurs de torsion aux sommets du carreau doivent respecter les valeurs suivantes :

$$\begin{aligned}
T_{00} &= \frac{dD_0(u=0)}{du} &= \frac{dD_0(v=0)}{dv} \\
T_{01} &= \frac{dD_0(u=1)}{du} &= \frac{dD_1(v=0)}{dv} \\
T_{10} &= \frac{dD_1(u=0)}{du} &= \frac{dD_0(v=1)}{dv} \\
T_{11} &= \frac{dD_1(u=1)}{du} &= \frac{dD_1(v=1)}{dv}
\end{aligned} \tag{B.6}$$

Comme précédemment, on définit le carreau de COONS à l'aide de trois surfaces :

$$S(u, v) = S_1(u, v) + S_2(u, v) - T(u, v) \tag{B.7}$$

où $S_1(u, v)$ et $S_2(u, v)$ sont des surfaces de BÉZIER cubiques dans la direction u, respectivement v. Initialement, COONS avait utilisé des fonctions polynômiales de Hermite. Le choix des surfaces de BÉZIER est justifié car les frontières sont des courbes B-splines, et les carreaux adjacents (les branches à raccorder) sont aussi définis dans cette base. Des surfaces cubiques nous permettent d'obtenir une continuité G^1. S_1 est définie à partir de $C_k(u)$ et $D_k(u)$, les quatre rangées de points de contrôle sont calculés ainsi :

$$\begin{aligned}
P^1_{i0} &= P_{0i} \\
P^1_{i1} &= P_{0i} + \frac{1}{3}Q_{0i} \\
P^1_{i2} &= P_{1i} - \frac{1}{3}Q_{1i} \\
P^1_{i3} &= P_{1i} \qquad\qquad i = 0, \ldots, m
\end{aligned} \tag{B.8}$$

Les points de contrôle de S_2 sont calculés de manière analogue en fonction des données de $C_l(u)$ et $D_l(u)$. $T(u, v)$ est une surface bicubique de BÉZIER. Ses points de contrôle peuvent être calculés selon l'équation B.9 autour de S_{00}. Les douze autres points autour des autres sommets sont obtenus de manière analogue.

$$\begin{aligned}
P^T_{00} &= S_{00} \\
P^T_{10} &= \frac{1}{3}D_0(v = 0) + P^T_{00} \\
P^T_{01} &= \frac{1}{3}D_0(u = 0) + P^T_{00} \\
P^T_{11} &= \frac{1}{9}T_{00} + P^T_{10} + P^T_{01} - P^T_{00}
\end{aligned} \tag{B.9}$$

Ensuite, il faut rendre les surfaces S_1, S_2 et T compatibles. Pour cela, il est nécessaire de les élever au même degré mais aussi de rendre leurs vecteurs nodaux compatibles, ce qui peut poser certains problèmes.

Enfin, les points de contrôle de la surface de COONS sont calculés grâce à la formule suivante :

$$P_{ij} = P^1_{ij} + P^2_{ij} - P^T_{ij}$$

C Intégration des lois de la dynamique

Nous utilisons les lois de la dynamique pour calculer les nouvelles positions des objets lors d'une animation. Cela implique la résolution d'équations différentielles. Dans ce chapitre, nous présentons les méthodes de résolution les plus courantes pour résoudre une équation de la forme :

$$\frac{dy}{dt} = f(t)$$

Trouver une solution analytique à cette équation est souvent difficile, nous devons généralement passer par une solution numérique qui revient à évaluer l'intégrale :

$$y(t + \Delta t) - y(t) = \int_t^{t+\Delta t} f(u)du$$

où f est connue est différents points d'échantillonnage.

C.1 Équations dela dynamique

Les équations de Newton relient les déplacements aux forces extérieures $\vec{f}$ appliquées à l'objet.

$$\sum \vec{f} = m\vec{a} \tag{C.1}$$

où m est la masse d'inertie. L'accélération du centre de gravité $\vec{a}$ est obtenue en dérivant la position par rapport au temps.

$$a\vec{(t)} = \frac{d\vec{v}(t)}{dt} = \frac{d^2\vec{p}(t)}{dt^2} \tag{C.2}$$

où $\vec{p}$ est la position et $\vec{v}$ la vitesse.

C.2 Méthode d'EULER

Il s'agit de la méthode la plus simple et la plus employée. La position à l'instant $t + \Delta t$ se déduit de la position à l'instant t :

$$y(t + \Delta t) = y(t) + f(t)\Delta t$$

Cette méthode est exacte quand la fonction f est constante sur l'intervalle $[t, t + \Delta t]$. En intégrant les équations de la dynamique (Éq. C.1 et C.2), on obtient :

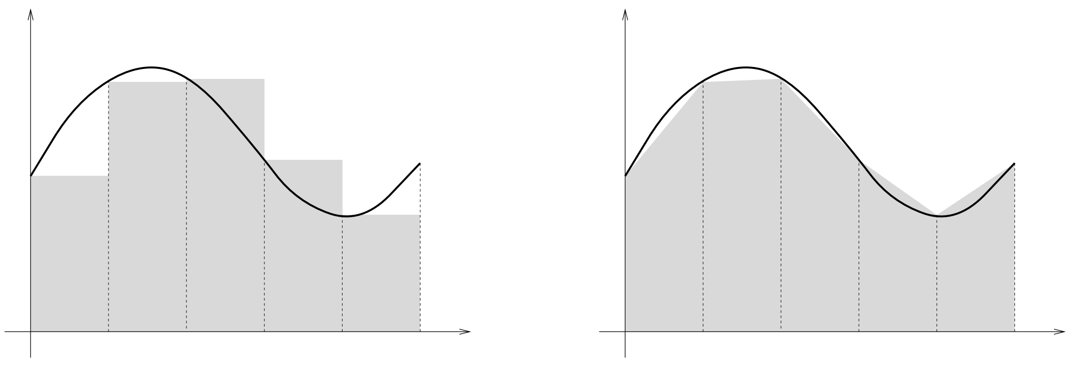
$$\begin{cases} \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t)}{m}\Delta t \\ \mathbf{p}(t + \Delta t) = \mathbf{p}(t) + \mathbf{v}(t)\Delta t \end{cases}$$

C.3 Méthode de NEWTON-COTES

La Méthode de NEWTON-COTES est une méthode un peu plus évoluée. La nouvelle valeur de y se déduit de la manière suivante :

$$y(t + \Delta t) = y(t) + (f(t) + f(t + \Delta t))\,\frac{\Delta t}{2}$$

Cette méthode est exacte lorsque la fonction f varie linéairement sur l'intervalle. C'est une meilleure approximation que la méthode d'EULER C.1. Elle n'est cependant pas utilisable



(a) Euler (b) Newton-Cotes

FIG. C.1 – *Comparaison entre deux méthodes d'intégration*

pour calculer les vitesses, puisqu'elle requiert l'état des forces à $t + \Delta t$, ce qui nécessite de connaître la position $\mathbf{p}(t + \Delta t)$. Ce qui est justement ce que l'on recherche. Nous aboutissons au système suivant :

$$\begin{cases} \mathbf{v}(t + \Delta t) = \mathbf{v}(t) + \frac{\mathbf{f}(t)}{m}\Delta t \\ \mathbf{p}(t + \Delta t) = \mathbf{p}(t) + (\mathbf{v}(t) + \mathbf{v}(t + \Delta t)(t))\,\frac{\Delta t}{2} \end{cases}$$

C.4 Méthode de Runge-Kutta

Il est possible d'évaluer la dérivée de y pour une valeur intermédiaire, pour corriger un peu l'intégration :

$$y(t + \Delta t) = y(t + \Delta t/2) + f(t + \Delta t/2)\frac{\Delta t}{2}$$

Et une approximation de y à la moitié de l'intervalle est obtenue par :

$$y(t + \Delta t/2) = y(t) + f(t)\frac{\Delta t}{2}$$

Cette méthode est plus précise que celle d'Euler, mais elle nécessite deux fois plus de calculs. Il est possible de faire des approximations à des ordres plus élevés en évaluant la dérivée en un plus grand nombre de points intermédiaires. Éventuellement, le pas d'échantillonnage peut être adaptatif, mais dans la complexité se trouve grandement augmentée.

C.5 Méthodes d'ordre plus élevé

Les méthodes d'Euler et de Newton-Cotes sont des méthodes d'ordre 1, elles n'utilisent que les informations aux instants t ou $t + \Delta t$. Obtenir un ordre plus élevé nécessite l'évaluation de f en des points intermédiaires. C'est le cas de la méthode de Runge-Kutta, qui utilise un point intermédiaire. Il existe d'autres méthodes, dont toute une famille issue de Newton-Cotes. Soit $f_i = f(a + ih)$ pour $i = 0 \dots n$, où n et h permettent une répartition régulière des points sur l'intervalle $[a, b]$.

$$\begin{array}{lrl}
n = 1 & \int_a^b f(u)du = \frac{h}{2}(f_0 + f_1) & \text{(Newton-Cotes)} \\
n = 2 & \int_a^b f(u)du = \frac{h}{3}(f_0 + 4f_1 + f_2) & \text{(Simpson)} \\
n = 3 & \int_a^b f(u)du = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) & \text{(Simpson } \frac{3}{8}\text{)} \\
 & \dots &
\end{array}$$

Ces méthodes sont beaucoup plus précises qu'Euler ou Newton-Cotes, mais elles engendrent un nombre plus important de calculs, qui ne sont pas vraiment nécessaires dans le cadre de l'animation. La simulation peut se décomposer en instants où les objets sont soumis à des forces constantes (contact prolongé ou vol balistique), ou sont en collision [Lom96]. Lors des collisions, les forces ne sont pas forcément continues et une approximation par un polynôme n'est pas recommandée. Lorsque les forces sont constantes, une intégration d'Euler ou de Newton-Cotes fournit des résultats tout à fait satisfaisants. L'utilisation de méthodes plus complexes n'est donc pas justifiée, du moins pour l'intégration des équations de la dynamique.

Bibliographie

[Amr97] M. AMRANI. « Contribution à la modélisation et à l'animation d'objets déformables ». Master's thesis, LIGIM, Université Claude Bernard LYON I, juin 1997.

[Bar98] D. BARAFF. « Rigid Body Simulation ». Dans *ACM SIGGRAPH'98 Courses Notes 13 : Physically Based Modeling*, pages D1–68, août 1998.

[BB88] R. BARZEL et A. H. BARR. « A modeling system based on dynamic constraints ». Dans *Proceedings of SIGGRAPH'88*, pages 179–188. Computer Graphics, août 1988.

[BCL95] C. L. BAJAJ, E. J. COYLE, et K.-N. LIN. « Arbitrary topology shape reconstruction from planar cross sections ». Rapport Technique, Purdue University, 1995.

[Béz87] P. BÉZIER. *Courbes et surfaces*, volume 4 de *Mathématiques et CAO*. Hermès, 2nde édition, 1987.

[BG92] J.-D. BOISSONNAT et B. GEIGER. « Three dimensional reconstruction of complex shapes based on the Delaunay triangulation ». Rapport Technique 1697, INRIA Sophia Antipolis, avril 1992. Programme 4 : robotique, image et vision.

[Boe80] W. BOEHM. « Inserting new knots into B-spline curves ». *Computer Aided Design*, 12(4) :199–201, juillet 1980.

[Boo78] C. De BOOR. *A practical guide to splines*. Springer-Verlag, 1978.

[BV91a] R. M. BOLLE et B. C. VEMURI. « On three-dimensional surface reconstruction methods ». *IEEE Transactions on PAMI*, 13(1) :1–13, janvier 1991.

[BV91b] W. J. BOUMA et G. VANĚČEK. « Collision detection and analysis in a physically based simulation ». Dans *2nd Eurographics Workshop 91*, pages 191–203, 1991.

[CDA96] S. COTIN, H. DELINGETTE, et N. AYACHE. « Volumetric deformable models for surgery simulation of non-rigid organs ». Dans *4e séminaire du groupe de travail "Animation et Simulation"*. GDR-PRC AMI (Algorithme, Modélisation et Infographie), Strasbourg, janvier 1996.

[CEO+93] S. A. COVER, N. F. EZQUERRA, J. F. O'BRIEN, R. ROWE, T. GADACZ, et E. PALM. « Interactively deformable models for surgery simulation ». *IEEE Computer Graphics & Applications*, 13(6) :68–75, novembre 1993.

[Coh92] A. COHEN. *Ondelettes et traitement numérique du signal*. Masson, 1992.

[Des97] M. DESBRUN. « *Modélisation et animation de matériaux hautement déformables en synthèse d'images* ». PhD thesis, Institut National Polytechnique de Grenoble, décembre 1997. Laboratoire iMAGIS/GRAVIR-IMAG.

[DG95] M. DESBRUN et M.-P. GASCUEL. « Animating soft substances with implicit surfaces ». Dans *Proceedings of SIGGRAPH'95*. Computer Graphics, 1995.

[DKM+95] D. DeCarlo, J. Kaye, D. Metaxas, J. R. Clarke, B. Webber, et N. Badler. « Integrating anatomy and physiology for behavior modeling ». Dans *Medicine meets virtual reality 3, San Diego*, 1995.

[EH96] M. Eck et H. Hoppe. « Automatic reconstruction of B-spline surfaces of arbitrary topological type ». Rapport Technique 1800, Dept. of Mathematics, Univ. of Darmstadt, Germany, janvier 1996. Submitted to SIGGRAPH'96.

[EPO91] A. B. Ekoule, F. C. Peyrin, et C. L. Odet. « A triangulation algorithm from arbitrary shaped multiple planar contours ». *ACM Transactions on Graphics*, 10(2):182–199, avril 1991.

[ESS91] R. Ebel, J. Sequeira, et F. Schmitt. « Reconstruction interactive d'arborescences 3D à l'aide de surfaces de forme libre ». Dans *Journées Graphiques GROS PLAN*, pages 31–38, décembre 1991.

[Far93] G. Farin. *Curves and surfaces for CAGD, a practical guide.* Academic Press, 3ème édition, 1993.

[FB89] D. J. Filip et T. W. Ball. « Procedurally representing lofted surfaces ». *IEEE Computer Graphics & Applications*, 9(6):27–33, novembre 1989.

[FB95] D. R. Forsey et R. H. Bartels. « Surface fitting with hierarchical splines ». *ACM Transactions on Graphics*, 14(2), avril 1995.

[FG96] L. H. De Figueiredo et J. Gomes. « Sampling implicit objects with physically-based particle systems ». *Computer & Graphics*, 20(3):365–375, 1996.

[FKU77] H. Fuchs, Z.M. Kedem, et S.P. Uselton. « Optimal surface reconstruction from planar contours ». *Communications of the ACM*, 20(10):693–702, octobre 1977.

[Gas93] M.-P. Gascuel. « An implicit formulation for precise contact modeling between flexible solids ». Dans *Proceedings of SIGGRAPH '93*, pages 313–320. Computer Graphics, août 1993.

[GO95] U. Güdükbay et Bülent Özgüç. « Animating deformable models: different approaches ». Dans *Computer Animation'95, Geneva*, pages 138–145, avril 1995.

[GOS93] C. Gitlin, J. O'Rourke, et V. Subramanian. « On reconstructing polyhedra from parallel slices ». TR 025, Smith College Department of Computer Science, Northampton, MA, mars 1993.

[HDD+92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, et W. Stuetzle. « Surface reconstruction from unorganized points ». Dans *Proceedings of SIGGRAPH'92*, pages 71–78. Computer Graphics, juillet 1992.

[HLLK94] Y. Han, K. Lee, S. Lee, et G. Khang. « Reconstruction of 3D objects from 2D medical images ». Dans *Proceedings of the 6th ICECGDG*, pages 380–384, Tokyo, Jp, août 1994.

[Hou92] D. House. « Coupled particles: theory ». Dans *ACM SIGGRAPH'92 Courses Notes: Particles system modeling, animation and physically based techniques*, pages 3.1–3.8, juillet 1992.

[HPH96] D. Hutchinson, M. Preston, et T. Hewitt. « Adaptative refinement for mass/spring simulation ». Dans *7e Eurographics Workshop on Animation and Simulation, Poitiers (F)*, pages 31–45, septembre 1996.

[JGL97] A. Joukhadar, F. Garat, et C. Laugier. « Parameter identification for dynamic simulation ». Dans *IEEE Int. Conf. on Robotics and Automation, Albuquerque (NM)*, pages 1928–1933, vol 3, avril 1997.

[Jim93] S. Jimenez. « *Modélisation et simulation physique d'objets volumiques déformables complexes* ». PhD thesis, Institut National Polytechnique de Grenoble, novembre 1993.

[Jou96] A. Joukhadar. « *Simulation dynamique et applications robotiques* ». PhD thesis, INRIA, décembre 1996.

[JS95] J. K. Johnstone et K. R. Sloan. « Tensor product surfaces guided by minimal surface area triangulations ». Dans *IEEE*, 1995.

[Kam93] V. V. Kamat. « A survey of techniques for simulation of dynamic collision detection and response ». *Computer & Graphics*, 17(4) :379–385, 1993.

[Leo91] J.-C. Leon. *Modélisation et construction de surfaces pour la CFAO*. Hermès, 1991.

[LHM95] A. Luciani, A. Habibi, et E. Manzotti. « A multi-scale physical model of granular materials ». Dans *Proceedings Graphics Interface'95*, pages 136–146, mai 1995.

[Lom96] J.-C. Lombardo. « *Modélisation d'objets déformables avec un système de particules orientées* ». PhD thesis, Université de Grenoble I, janvier 1996.

[LP95] J.-C. Lombardo et C. Puech. « Modélisation d'objets déformables avec un système de particules orientées ». *Revue Internationale de CFAO et d'Informatique Graphique*, 10(4) :337–354, 1995. Also in Deuxièmes Journées de l'AFIG, Toulouse, Dec. 1994.

[Mal95] I. Ould Dellahy Maloum. « *Reconstruction de surfaces gauches à partir de données non structurées - Paramétrisation par des transformations conformes* ». PhD thesis, Université Claude Bernard - LYON I, septembre 1995.

[MC96] P. Meseure et C. Chaillou. « Modélisation mécanique pour la simulation d'actes chirurgicaux ». Dans *4ème séminaire du groupe de travail "Animation et Simulation"*. GDR-PRC AMI (Algorithme, Modélisation et Infographie), Strasbourg, janvier 1996.

[Mey93] D. Meyers. « Multiresolution tiling ». Rapport Technique 93-12-02, Department of Computer Science and Engineering, Univ. of Washington, décembre 1993. Appeared in Proceedings, Graphics Interface '94, Banff, Alberta, pp25-32.

[Mey94] D. Meyers. « *Reconstruction of surfaces from planar contours* ». PhD thesis, University of Washington, 1994.

[MK95] W. Ma et J. P. Kruth. « Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces ». *Computer-Aided Design*, 27(9) :663–675, septembre 1995.

[Mon87] A. Montanvert. « *Contribution au traitement de formes discrètes. Squelettes et codage par graphe de la ligne médiane* ». PhD thesis, Université J. Fourier et INPG, Grenoble I, 1987.

[MP89] G. Miller et A. Pearce. « Globular dynamics : a connected particle system for animating viscous fluids ». *Computers & Graphics*, 13(3) :305–309, 1989.

[MPS+94] I. Ould Dellahy Malloum, E. Perna, B. Shariat, D. Vandorpe, S. Bouakaz, et M. Melkemi. « Surface Reconstruction from Range Images ». Dans *Interna-*

tional Conference on Engineering Computer Graphics and Descriptive Geometry, pages 370–374, août 1994.

[MS95] K.-L. Ma et P. J. Smith. « Visualization of particles and gas mixing ». *Revue de CFAO et d'Informatique Graphique*, 10(3) :273–289, 1995.

[MSS92] D. Meyers, S. Skinner, et K. Sloan. « Surfaces from contours ». *ACM Transactions on Graphics*, 11(3) :228–258, juillet 1992.

[MW88] M. Moore et J. Wilhelms. « Collision detection and response for computer animation ». Dans *Proceedings of SIGGRAPH'88*, pages 289–298. Computer Graphics, août 1988. Volume 22 (4).

[NFD95] M. Neveu, D. Faudot, et B. Derdouri. « Superquadriques-B-déformables pour la reconstruction 3D ». *Technique et Science Informatiques*, 14(10) :1291–1314, octobre 1995.

[OPC96] J.-M. Oliva, M. Perrin, et S. Coquillart. « 3D reconstruction of complex polyhedral shapes from contours using a simplified generalized voronoï diagram ». Dans *Eurographics'96, Poitiers (F)*, pages C397–C408, 1996. vol 15 (3).

[PB88] J. C. Platt et A. H. Barr. « Constraint method for flexible models ». Dans *Proceedings of SIGGRAPH'88*, pages 279–288. Computer Graphics, 1988.

[PBP96] E. Promayon, P. Baconnier, et C. Puech. « Physically-based deformations constrained in displacements and volume ». Dans *Eurographics'96, Poitiers*, pages C155–164, août 1996. Computer Graphics Forum, vol 15(3).

[PBP97] E. Promayon, P. Baconnier, et C. Puech. « Physically-based model for simulating the human trunk respiration movements ». Dans *CVRMED-MCAS First Joint Conference '97, Grenoble*, mars 1997.

[Per92] E. Perna. « *Modèles de surfaces pour la CFAO, raccordement de carreaux définis par produit tensoriel* ». PhD thesis, Univ. Claude Bernard - Lyon I, octobre 1992.

[PH94] M. Preston et W. T. Hewitt. « Animation using NURBS ». *Computer Graphics Forum*, 13(4) :229–241, 1994.

[PK95] H. Park et K. Kim. « An adaptative method for smooth surface approximation to scattered 3D points ». *Computer Aided Design*, 27(12) :929–939, décembre 1995.

[PK96] H. Park et K. Kim. « Smooth surface approximation to serial cross-sections ». *Computer Aided Design*, 28(12) :995–1005, décembre 1996.

[PMY94] J. Park, D. Metaxas, et A. Young. « Deformable models with parameter functions : application to heart-wall modeling ». Dans *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Seattle, WA*, pages 437–442, juin 1994.

[PSV98] S. Pontier, B. Shariat, et D. Vandorpe. « Implicit surface reconstruction from 2D CT scan sections ». Dans IEEE Computer Society, éditeur, *Proceedings of CGI'98, Hannovre (D)*, pages 583–586. IEEE Computer Society, juin 1998.

[PT95] L. Piegl et W. Tiller. *The NURBS book*. Springer, 1995.

[PT96] L. Piegl et W. Tiller. « Algorithm for approximate NURBS skinning ». *Computer-Aided Design*, 28(9) :699–706, septembre 1996.

[RBB97] R. Ramamoorthi, C. Ball, et A. H. Barr. « Dynamic splines with constraints for animation ». Rapport Technique CS-TR-97-03, California Institute of Technology, Computer Science Department, 1997.

[Ree83] W. T. Reeves. « Particle systems : a technique for modeling a class of fuzzy objects ». Dans *Proceedings of SIGGRAPH'83*, pages 359–376. Computer Graphics, 1983.

[Rey97] H. Reynolds. « An alternative inter-particle force model for coupled system flexiblebody dynamics ». Dans *EUROGRAPHICS'97, 8th Eurographics Workshop on Animation and Simulation, Budapest (H)*, pages 99–110, septembre 1997.

[RF89] D. F. Rogers et N. G. Fog. « Constrained B-spline curve and surface fitting ». *Computer-Aided Design*, 21(10) :641–648, décembre 1989.

[SG95] K. Shimada et D. C. Gossard. « Bubble mesh : automated triangular meshing of non-manifold geometry by sphere packing ». Dans *Third Symposium on Solid Modeling and Applications, Salt Lake City, Utah*, pages 409–419, mai 1995.

[She95] A. Sheffer. « Volume-preserving free-form solids ». Master's thesis, Inst. of Comp. Sc., The Hebrew Univ. of Jerusalem, Ir., 1995.

[SK91] Y. Shinagawa et T.L. Kunii. « The homotopy model : a generalized model for smooth surface generation from cross sectional data ». *The Visual Computer*, 7(2-3) :72–86, mai 1991.

[SM91] B. Sarkar et C.-H. Menq. « Parameter optimization in approximation curves and surfaces to measurement data ». *Computer Aided Geometric Design*, 8(4) :267–290, octobre 1991.

[SP95] V. Savchenko et A. Pasko. « Reconstruction from contour data and sculpting 3D objects ». Dans *Proceedings of Second Int. Symposium on Comp. Aided Surgery, ISCAS'95, Tokyo, Japan*, pages 56–57, octobre 1995. Also appeared in Journal of Computer Aided Surgery, vol. 1 supl.

[SPOK95] V. Savchenko, A. Pasko, O. Okunev, et T. Kunii. « Function representation of solids reconstructed from scattered surface points and contours ». *Computer Graphics Forum*, 14(4) :181–188, 1995.

[ST91] R. Szeliski et D. Tonnesen. « Surface modeling with oriented particle systems ». Rapport Technique CRL 91/14, DEC, Cambridge Research Lab, décembre 1991.

[ST92] R. Szeliski et D. Tonnesen. « Surface modeling with oriented particle systems ». Dans *Proceedings of SIGGRAPH '92*, pages 185–194. Computer Graphics, juillet 1992.

[SZL92] W. J. Schroeder, J. A. Zarge, et W. E. Lorensen. « Decimation of triangle meshes ». Dans *Proceedings of SIGGRAPH '92*, pages 65–70. Computer Graphics, juillet 1992.

[Ter89] D. Terzopoulos. « Heating and melting deformable models (from goop to glop) ». Dans *Graphics Interface '89*, pages 219–226, 1989.

[TF88a] D. Terzopolous et K. Fleicher. « Modeling inelastic deformation : viscoelasticity, plasticity, fracture ». Dans *Proceedings of SIGGRAPH'88*, pages 269–278. Computer Graphics, 1988.

[TF88b] D. Terzopoulos et K. Fleisher. « Deformable models ». *The Visual Computer*, 4, 1988.

[TM91] D. Terzopoulos et D. Metaxas. « Dynamic 3D models with local and global deformations : deformable superquadrics ». *IEEE Transactions on PAMI*, 13(7) :703–714, juillet 1991.

[Ton92] D. Tonnesen. « Modeling liquids and solids using thermal particles ». Dans *ACM SIGGRAPH'92 Courses Notes : Particles system modeling, animation and physically based techniques*, pages 4.22–4.29, juillet 1992.

[Tur92] G. Turk. « Re-tiling polygonal surfaces ». Dans *Proceedings of SIGGRAPH '92*, pages 55–64. Computer Graphics, juillet 1992.

[Vis] « The Visible Human Project ». U.S. National Library of Medicine. http ://www.nlm.nih.gov/research/visible/visible_human.html.

[WH91] J. Wejchert et D. Haumann. « Animation aerodynamics ». *ACM SIGGRAPH 91*, 25(4) :19–22, juillet 1991.

[WH94] A. P. Witkin et P. S. Heckbert. « Using particles to sample and control implicit surfaces ». Dans *Proceedings of SIGGRAPH '94*, pages 269–277. Computer Graphics, juillet 1994.

[Wit92a] A. Witkin. « Constrained dynamics ». Dans *ACM SIGGRAPH'92 Courses Notes : An introduction to physically based modeling*, pages G1–12, juillet 1992.

[Wit92b] A. Witkin. « Particle system dynamics ». Dans *ACM SIGGRAPH'92 Courses Notes : An introduction to physically based modeling*, pages C1–C12, juillet 1992.

[Woo87] C. D. Woodward. « Cross-sectionnal design of B-spline surfaces ». *Computer & Graphics*, 11(2) :193–201, 1987.

[Woo88] C. D. Woodward. « Skinning techniques for interactives B-spline surface interpolation ». *Computer-Aided Design*, 20(8) :441–451, octobre 1988.

[WSS+98] A. W.F.Lee, W. Sweldens, Peter Schröder, L. Cowsar, et D. Dobkin. « MAPS : Multiresolution Adaptive Parameterization of Surfaces ». Dans *Proceedings of SIGGRAPH '98*. Computer Graphics, août 1998.

[WTMT95] Y. Wu, D. Thalmann, et N. Magnenat-Thalmann. « Deformable surfaces using physically based particle systems ». Dans R. Earnshaw et J. Vince, éditeurs, *CGI'95 conference. Computer Graphics : Developments in Virtual Environments, Leeds, UK*, pages 205–215, juin 1995.

[ZJH87] M. J. Zyda, A. R. Jones, et P. G. Hogan. « Surface reconstruction from planar contours ». *Computers & Graphics*, 11(4) :393–408, 1987.

Résumé

Grâce aux récents développements des systèmes d'acquisition, les données structurées en contours sont de plus en plus répandues. Elles comportent souvent un grand nombre de points et sont par ailleurs peu agréables à visualiser. Il était donc intéressant de proposer une méthode de reconstruction de surfaces 3D lisses et compactes à partir de tels contours. La première partie de la thèse présente une approche originale basée sur le « skinning » des contours. Nous commençons par construire une courbe B-spline sur chaque contour. Après une phase délicate de mise en compatibilité, nous interpolons ou approximons l'ensemble de ces courbes pour obtenir une surface B-spline définie par un produit tensoriel et possédant une forte continuité en tout point. Nous avons aussi étendu notre méthode pour fermer ces surfaces en respectant des critères de tangence. Nous avons aussi pris en compte les objets possédant des branches, en comblant les zones d'embranchement par des carreaux de Coons.

Cependant, dans notre application médicale, les corps reconstruits sont en mouvement et peuvent se déformer. Ce type de comportement étant difficile à prendre en compte avec les modèles B-splines, nous avons proposé l'utilisation des systèmes de particules. La seconde partie de la thèse concerne la reconstruction et l'animation des objets déformables. Nous décrivons notre méthode de remplissage d'un volume par plusieurs couches de particules : des particules de grande taille donnent la forme générale, elles sont entourées par des particules de taille intermédiaire qui contiennent les paramètres liés au comportement, qui sont elles-mêmes entourées par de petites particules assurant la cohésion de l'objet ainsi que les échanges avec le monde extérieur. Ce modèle permet de simuler simplement une grande variété de comportements (élasticité, plasticité, fracture...).

Mots-clés :
Informatique graphique, imagerie médicale, modélisation géométrique, reconstruction 3D, *skinning*, ensemble de contours, modèles déformables, systèmes de particules, animation.

Abstract

Cross-sections are increasingly widespread thanks to the recent developments in digitizing systems. They often comprise a great number of vertices and are in addition not very pleasant to visualize. It was thus interesting to propose a new reconstruction method to produce smooth and compact 3D surfaces from such input data. The first part of this PhD thesis presents an original approach based on skinning of contours. We start by fitting a B-spline curve on each contour. These curves have to be made compatible. Next, we interpolate or approximate the whole of these curves to obtain a tensor product surface having a strong continuity in any point. We also have extended our method to close this surface while respecting tangency criteria. The branching cases are also handled by fitting the holes between the branches with Coons patches.

However, in our medical application, reconstructed bodies are moving and can become deformed. This kind of behavior being difficult to take into account with B-spline modeling, we proposed the use of particles systems. The second part of this thesis deals with reconstruction and animation of deformable objects. We describe our volume filling method with a multi-layer system : large particles give the general shape, they are surrounded by medium particles which handle parameters related to the behavior of the object, which themselves are surrounded by little particles ensuring cohesion as well as the exchanges with the external world. This modeling permits to recreate easily a wide variety of behavior (elasticity, plasticity, fracture).

Keywords :
Computer graphics, medical imaging, CAD/CAM, 3D reconstruction, skinning, cross-sections, deformable objects, particles system, animation.