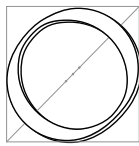


---

# Modèle multirésolution hiérarchique

---



N A ÉTUDIÉ dans le précédent chapitre la formulation d'opérateurs différentiels efficaces et indépendants de la résolution. Leur utilisation dans un algorithme multirésolution va consister à utiliser simultanément plusieurs maillages de résolutions différentes. L'introduction de points *fantômes* permettra à ces maillages de communiquer et il ne restera qu'à choisir quelle résolution est la mieux adaptée à chaque zone de l'objet. Les méthodes décrites dans ce chapitre ont fait l'objet d'une publication à *Computer Animation* [DDCB00] (avec le modèle hybride, voir Chapitre 4) et à *Siggraph* 2001[DDCB01] (avec le tenseur de Green-Lagrange et l'intégration semi implicite).

## 1 Idée générale

L'algorithme expliqué dans ce chapitre s'appuie sur une découpe de l'objet en un maillage de tétraèdres. Ce choix profite de la simplicité et de la souplesse de ces maillages, tétraèdres. Ce choix profite de la simplicité et de la souplesse de ces maillages, seuls capables de mailler tout objet délimité par une surface triangulée. La notion de voisinage entre points se déduit du maillage, les points voisins étant ceux reliés par une arête de tétraèdre.

Nous supposons ici que chaque point est capable de calculer la force qui lui est appliquée en fonction des seuls déplacements de ses voisins. Outre les méthodes d'éléments finis explicites ou celles basées sur les régions de Voronoï vues dans le chapitre précédent, on pourrait donc utiliser l'algorithme que nous allons décrire avec les opérateurs de la première méthode de cette thèse (Chapitre 3), voire avec une méthode de type masses-ressorts si l'on est capable de la rendre suffisamment indépendante du maillage.

De la qualité du maillage dépend grandement celle des résultats numériques obtenus, en particulier avec la méthode des éléments finis. Le tétraèdre idéal est régulier, toutes ses arêtes ayant la même longueur. Il est en pratique impossible de mailler un volume en utilisant uniquement de tels tétraèdres réguliers, mais on peut avec une méthode de relaxation essayer de s'en approcher.

L'idée générale de la méthode présentée ici est qu'il serait à la fois coûteux en temps de calcul et insatisfaisant au niveau de la simulation numérique d'essayer d'*adapter* le maillage en fonction de la simulation.

Il faudrait en effet subdiviser les tétraèdres existants lorsque l'on cherche à ajouter des points [HPH96, OH99]. Il n'existe malheureusement pas de découpe *régulière* d'un tétraèdre, et l'on est forcément amené à dégrader la qualité du maillage (en terme d'angles diédriques et de rapport des longueurs d'arêtes). Des études ont été faites sur le type de subdivision possible et l'optimisation de la position des points ajoutés [PLL98]. Il

n'existe pas de solution optimale et l'on ne peut assurer la qualité du maillage généré.

On pourrait envisager de remailler complètement l'objet lorsque les tétraèdres sont de trop mauvaise qualité, mais cette opération est coûteuse et incompatible avec le temps-réel. Il faut également s'assurer que le maillage reste *conforme* à chaque instant. Enfin, on a peu de chances de retrouver un maillage de grande qualité lorsque l'on revient dans la position non déformée, suite aux regroupements successifs.

D'ailleurs, la simplification du maillage peut elle-même poser problème. Regrouper localement plusieurs tétraèdres en un seul peut amener à des maillages de mauvaise qualité dans lesquels des tétraèdres de tailles très différentes cohabitent et ne peuvent se simplifier. On peut éviter ce problème en imposant que seuls les tétraèdres issus d'une *même* subdivision puissent se regrouper, ce qui restreint la souplesse du procédé. Conserver un historique des subdivisions peut alors permettre de revenir au maillage initial, supposé de bonne qualité.

En pratique, on constate une divergence de la simulation, liée à la dégradation du maillage, après deux ou trois subdivisions successives des tétraèdres. C'est cette constatation qui nous a décidés à développer une méthode à base de maillages fixes, de grande qualité, pouvant ensemble approximer à plusieurs échelles un même mouvement.

## 2 Cohabitation de différents maillages

### 2.1 Création des maillages

Les différents maillages utilisés sont totalement indépendants les uns des autres et n'ont en particulier pas à *partager* de sommets. Ils représentent le même objet à des résolutions différentes, leur création pouvant comporter une phase d'optimisation visant à rendre les tétraèdres aussi réguliers que possible.

Nous utiliserons dans nos simulations entre 2 et 7 maillages différents pour un même objet. Il est à noter que quelques dizaines de points suffisent à générer le mouvement général d'un objet. On n'utilisera les maillages fins que pour simuler le mouvement de détail de petites zones, souvent proches de l'outil.

En pratique, nous avons utilisé les logiciels `qslim` [GH97] et `GHS3D` [SIM] pour créer nos maillages. L'objet initial est donné par sa représentation surfacique détaillée, sous forme d'une surface triangulée. `qslim` permet de simplifier ce maillage pour obtenir des représentations avec plus ou moins de triangles du même objet. Une option du programme permet d'imposer la création de triangles aussi réguliers que possible.

Ces différents maillages surfaciques sont ensuite maillés volumiquement par `GHS3D` qui infère la taille à donner aux tétraèdres internes de celle des triangles surfaciques. Le maillage volumique comprend une optimisation de la position des points *internes* visant à égaliser la longueur des arêtes (voir Fig. 5.1). On dispose après maillage de statistiques sur la qualité des tétraèdres créés.

Il peut arriver qu'une différence de quelques triangles sur la surface conduise à des maillages volumiques de qualités assez différentes. Aussi avons-nous utilisé un script qui, une fois le nombre approximatif de tétraèdres désirés donné, simplifie le maillage surfacique de l'objet pour obtenir plus ou moins de triangles et ne conserve parmi les maillages volumiques résultants, que celui correspondant aux meilleures statistiques.

### 2.2 Interface entre deux maillages

Supposons dans un premier temps que notre objet soit maillé avec deux maillages différents, l'un fin et l'autre plus grossier. Imaginons de plus qu'une moitié de l'objet doit être simulée avec le maillage fin, l'autre pouvant se contenter des calculs plus approximatifs du maillage grossier. La simulation de chacune des deux parties se fait très bien en utilisant les algorithmes et les opérateurs vus précédemment, et il ne reste qu'à gérer l'interface entre les deux maillages pour que l'objet forme un tout et ne soit pas la simple juxtaposition de deux simulations.

Pour ce faire, nous utilisons une technique qui peut être rapprochée de la méthode de décomposition de domaine<sup>1</sup>. Les deux maillages vont légèrement se superposer et certains de leurs sommets, que nous désignerons par le terme de *fantômes*, serviront alors à faire l'interface entre les deux maillages.

<sup>1</sup>Voir [www.ddm.org](http://www.ddm.org) pour de plus amples informations.

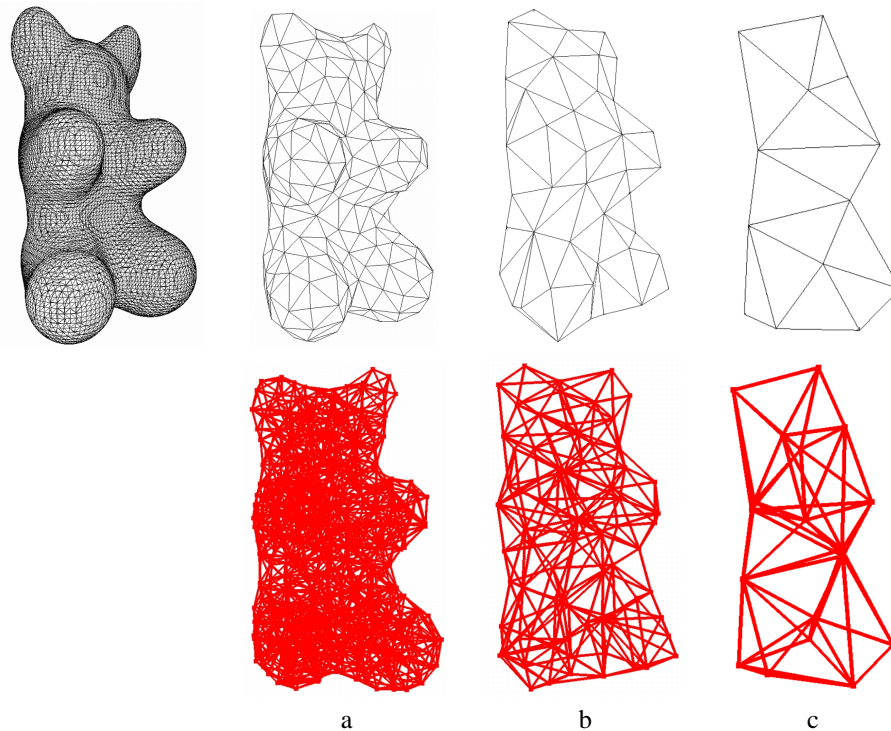


FIG. 5.1: Le maillage triangulaire original (34700 triangles) est dégradé à différents degrés (ici 550, 130 et 30 triangles), puis maillé volumiquement (1874, 172 et 28 tétraèdres, ligne du bas).

Pour cohabiter, les deux maillages ont seulement besoin de connaître la valeur du champ *déplacement* dans la zone située à l'interface. C'est en effet la seule donnée nécessaire aux points pour calculer leur mouvement. Les points fantômes d'un maillage porteront cette information de déplacement. Ils la déduiront de l'autre maillage et la communiqueront à leurs voisins du même maillage, servant ainsi d'interface.

### 2.3 Les points fantômes

Chaque point d'un maillage pourra avoir l'un des statuts suivants :

- Un point *actif* calculera la force qu'il subit à chaque pas de temps en fonction des déplacements de ses voisins. Il intégrera cette force pour trouver sa nouvelle position et donc son nouveau déplacement.
- Un point *inactif* ne sera simplement pas simulé.
- Entre les deux, nous trouverons les points *fantômes*. Ceux-ci ne calculent pas de force, mais auront un déplacement actualisé en fonction de celui du maillage actif de la même zone.

Un point actif aura donc pour voisins d'autres points actifs (aux déplacements réactualisés) ou des points fantômes (aux déplacements interpolés), qui tous lui indiqueront quelle est la déformation locale dans la zone qui l'entoure. Ce point réagira en sommant les influences de ses différents voisins, prenant ainsi, en particulier, en compte le déplacement de la zone d'interface (voir Fig. 5.2).

Un point fantôme déduira son déplacement de celui du tétraèdre du maillage actif dans lequel il est situé. On ne peut en effet avoir de meilleure estimation de ce qui se passe en ce point qu'en utilisant le tétraèdre qui le contient car c'est lui qui représente cette partie de l'espace pour ce maillage. Le point fantôme utilise ses coordonnées barycentriques dans ce tétraèdre pour calculer la valeur de son propre déplacement en fonction des valeurs du déplacement aux quatre sommets du tétraèdre englobant (les points  $Q_i$  de la Figure 5.2 pour l'exemple du point  $F$ ).

Une moyenne pondérée par les coefficients barycentriques des déplacements est équivalente à une interpolation linéaire de la valeur du champ à l'intérieur du tétraèdre. Cette interpolation est cohérente avec la méthode de calcul choisie pour les opérateurs différentiels, qui elle aussi suppose que le champ est linéaire par morceaux sur chaque élément. On n'ajoute donc pas ici d'hypothèses de continuité, se contentant d'être en accord avec les hypothèses faites précédemment.

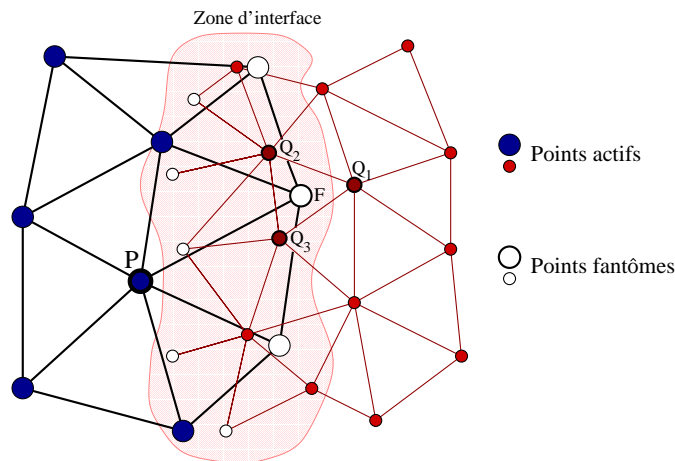


FIG. 5.2: Le voisin  $F$  de  $P$  est fantôme et sert à  $P$  à savoir ce qui se passe à sa droite. Il interpole son déplacement d'après celui des points  $Q_i$  du maillage fin.

En pratique, le point fantôme aura évalué ses coordonnées barycentriques dans la position de repos et les utilisera comme poids durant la simulation. L'évaluation du déplacement d'un point fantôme ne nécessitera donc que quatre multiplications scalaire-vecteur et trois additions de vecteurs, ce qui est beaucoup plus rapide que l'évaluation et l'intégration de la force subie que réalise un point actif.

Il est normal de considérer ces coefficients pondérateurs comme constants au cours de la simulation. On peut en effet voir le point fantôme comme "pris" dans un tétraèdre gélatineux, dont il va suivre les déformations. Ses coordonnées intrinsèques à l'intérieur de ce tétraèdre ne varient donc pas au cours de la simulation.

Il peut arriver, au bord de l'objet, qu'un point fantôme d'un maillage fin ne soit pas à l'intérieur d'un tétraèdre du maillage grossier. Nous le lierons alors avec le tétraèdre le plus proche, avec une de ses coordonnées barycentrique négative, de sorte qu'il extrapolera néanmoins le déplacement du tétraèdre.

Ce que nous venons de décrire pour deux maillages peut se généraliser, à condition de bien prendre soin à l'apparition des zones d'interface. Il faudra en particulier veiller à ce que les points actifs aient bien des voisins actifs ou fantômes et que les fantômes soient capables d'interpoler leur déplacement.

Les conditions et la mise à jour de la structure lors du changement de statut d'un point sont l'objet de la section suivante, qui va permettre de choisir localement les maillages les mieux adaptés au cours de la simulation.

### 3 Adaptation de la simulation

#### 3.1 Structure de données hiérarchique

Les sommets des différents maillages vont être reliés, dans la structure de données, avec les tétraèdres des deux maillages de résolutions directement inférieure et supérieure. Chaque point stockera son tétraèdre parent (maillage plus grossier) et fils (maillage fin), ainsi que ses coordonnées barycentriques dans ces deux tétraèdres. Il pourra ainsi, s'il devient fantôme, déduire son déplacement de celui des autres maillages (Fig. 5.3).

Chaque point stockera également une liste de points *fils*, définis comme étant ceux du maillage immédiatement inférieur (plus fin) situés à l'intérieur de sa région de Voronoï, définie par les voisins du point de même niveau que lui (voir Fig. 5.4).

Puisque situés à l'intérieur des régions de Voronoï, les points d'un maillage fin ne peuvent être fils que d'un point du maillage grossier. Cette relation permet donc de définir une structure d'arbre couvrant tous les sommets des maillages, chaque père étant lié à ses fils (voir Fig. 5.4).

On stocke aussi dans chaque point l'ensemble des premiers voisins de ses fils (hormis les fils eux mêmes). Ces points seront en effet nécessaires aux fils lorsqu'ils deviendront actifs pour pouvoir calculer une force (Fig. 5.4).

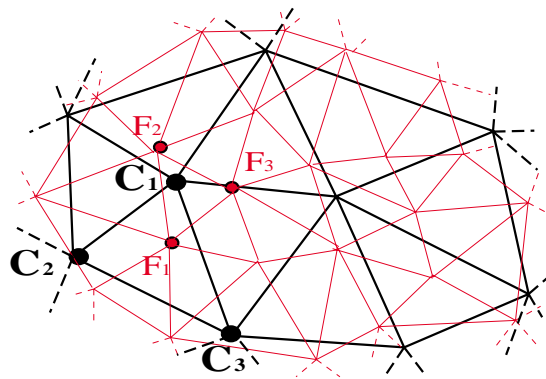


FIG. 5.3: Liaison des maillages (en 2D) : en fonction des maillages effectivement simulés, le sommet  $C_1$  pourra calculer sa position à partir de celles des points  $(F_1, F_2, F_3)$  du maillage fin. Les déformations pourront aussi être propagées aux noeuds du maillage fin :  $F_1$  pourra déduire sa position de celle des points  $(C_1, C_2, C_3)$  de son triangle parent.

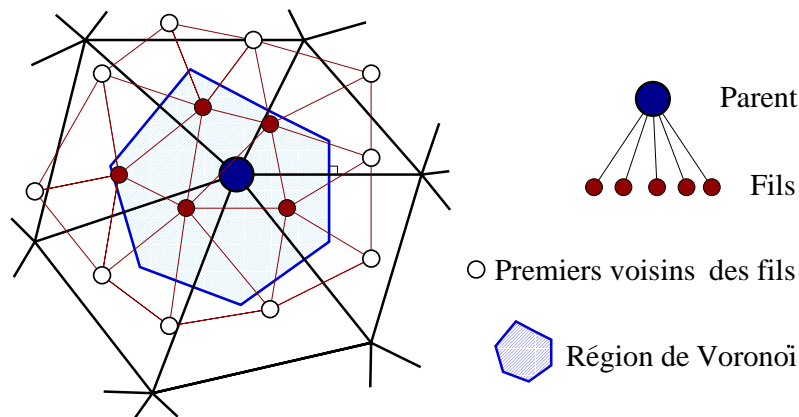


FIG. 5.4: Les fils d'un point sont les sommets du maillage fin situés dans sa zone de Voronoï. Ce sont eux qui remplaceront le point lorsqu'il se subdivisera. On stockera également la liste des premiers voisins de ces fils.

### 3.2 Subdivision et regroupement d'un point

Lorsqu'un point actif de l'un des maillages devient trop grossier pour représenter les déformations de cette zone, il se subdivise et est remplacé par ses fils. Il acquiert alors le statut de fantôme : il déduira son déplacement de celui de son tétraèdre fils. Sa position pourra être utilisée par ses voisins, qui peuvent eux être encore actifs. Ses fils ont quant à eux le statut d'actifs, et ils simuleront plus précisément les déformations de la région de Voronoï du point subdivisé.

Pour pouvoir se simuler, les points fils créés ont besoin d'avoir des déplacements actualisés en chacun de leurs voisins. C'est à cela que vont servir les premiers voisins des fils, qui deviendront fantômes (ou resteront actifs s'ils l'étaient déjà) lors de la subdivision.

Ces premiers voisins fantômes vont donner aux fils les déplacements des zones voisines, soit directement s'ils sont actifs (leur particule parente s'étant divisée), soit par moyenne, en utilisant une interpolation sur leur tétraèdre parent.

Inversement, on pourra simplifier une zone de l'espace en remplaçant un ensemble de fils par leur parent, celui-ci redevenant actif. Les points fils ne disparaissent pas forcément de la simulation car ils peuvent être nécessaires comme fantômes, voisins d'un point encore actif. Pour le savoir, un compteur indique combien de fois chaque point a été sollicité pour être premier voisin actif. Ce compteur est mis-à-jour lors de la subdivision ou du regroupement d'un point, et l'on sait quand il est à zéro que le point fils peut être supprimé lors du regroupement.

### 3.3 Critères de subdivision-regroupement

Les critères de subdivision et de regroupement des particules sont également les mêmes que ceux décrits au Chapitre 3, Sec. 4.5, page 54. Ils sont à nouveau basés sur la variation d'amplitude de la norme du laplacien, qui représente la dérivée seconde locale du champ, et donc son inadéquation avec le modèle linéaire utilisé pour calculer les opérateurs.

Rappelons que contrairement à ce qui est décrit au Chapitre 3, on n'a pas ici à chercher à conserver un octree restreint.

## 4 Raffinements du modèle

### 4.1 Utilisation d'un octree non restreint

Contrairement à ce qui avait été fait dans le premier modèle de cette thèse (Chapitre 3), nous avons ici choisi de ne pas nous limiter à la création d'un octree restreint. Une zone simulée avec des particules actives d'un certain niveau pourra donc côtoyer des régions simulées à n'importe quel niveau de la hiérarchie (et non plus seulement les niveaux directement inférieurs et supérieurs).

Ce choix simplifie grandement les procédures de subdivision et de regroupement de particules qui n'ont plus à chercher à conserver la structure d'octree restreint. Outre l'économie faite en temps de calcul par les vérifications supprimées, on gagne en simplicité et en lisibilité algorithmique, la conservation d'un octree restreint étant en pratique très subtile à programmer efficacement.

#### Justification

La conservation d'un octree restreint n'est plus vraiment nécessaire avec notre méthode d'animation puisque les résolutions ne communiquent plus vraiment entre elles comme c'était le cas dans la première méthode du Chapitre 3. L'utilisation de points fantômes fait que les déplacements d'une zone d'interface peuvent être déduits récursivement de n'importe laquelle des résolutions actives de cette zone. S'il y a plus d'un niveau d'écart entre une zone active et les points actifs d'une zone d'interface voisine, les différents points fantômes vont permettre en transférant l'information d'un maillage au suivant de la propager jusqu'à la résolution adéquate.

Les résultats de la première méthode ont également montré qu'un critère de subdivision bien choisi, en l'occurrence fonction de la courbure du champ déplacement (voir Chapitre 3, Sec. 4.5), permet d'obtenir en pratique une discrétisation intuitive du matériau, proche de celle d'un octree restreint (voir Sec. 3.3). Nous espérons donc que la structure conservera d'elle-même son caractère d'octree (quasi) restreint.

#### Subdivision modifiée

Du fait de la non restriction à un octree restreint, la procédure de subdivision d'un point se trouve un peu modifiée. Les premiers voisins des points fils sont en effet toujours ajoutés comme fantômes à la simulation (s'ils n'étaient déjà actifs), mais doivent maintenant en plus s'assurer qu'ils seront capables de mettre à jour leur déplacement.

Il se peut en effet que les points composant leur tétraèdre parent ne soient pas encore actifs, cette région n'ayant pas été subdivisée. Si tel est le cas, ces points seront eux aussi ajoutés comme fantômes et iront chercher l'information au niveau supérieur. Cette procédure est récursive et des fantômes sont ajoutés tant que l'on n'est pas remonté au niveau réellement actif de cette zone.

#### Suppression des fantômes

Tous ces points fantômes qui peuvent être ajoutés doivent donc aussi être supprimés lors du regroupement de la particule. Il est assez lourd de conserver dans chacun de ces points fantômes un compteur indiquant le nombre de particules de niveaux inférieurs ayant encore besoin du déplacement de ce point pour mettre à jour leur propre déplacement par interpolation. Il faudrait alors à chaque regroupement parcourir récursivement les points fantômes des tétraèdres parents des premiers voisins pour voir s'ils ne peuvent être supprimés.

Il est de plus possible qu'un de ces points, juste après avoir été supprimé, soit de nouveau nécessaire comme fantôme, une nouvelle subdivision le réclamant. Aussi avons-nous mis en place un processus de "ramasse-miettes" qui servira à supprimer les points fantômes devenus inutiles.

Régulièrement, et en pratique à chaque affichage de la scène (soit généralement tous les 32, 64 ou 128 pas de simulation selon le pas de temps utilisé), l'ensemble des points fantômes *non subdivisés* est parcouru pour y supprimer ceux devenus inutiles.

Les points fantômes subdivisés ne sont pas testés car on les sait probablement utiles à la simulation. Il est possible, si le matériau est très largement subdivisé, qu'un point fantôme d'un niveau très grossier ne serve plus, tous ses voisins étant également fantômes. Ce cas apparaissant rarement et ne concernant que quelques points au plus, nous ne le traitons pas.

## 4.2 Mise à jour des points fantômes

Nous avons mis en évidence un problème avec cette méthode lorsque les maillages successifs de l'objet ne sont pas suffisamment imbriqués. Il peut en effet arriver que les quatre points du *tétraèdre fils* d'un point donné (ceux qui servent à calculer son déplacement une fois subdivisé) ne soient pas tous directement des fils de ce point (le point  $F$  dans la Figure 5.5).

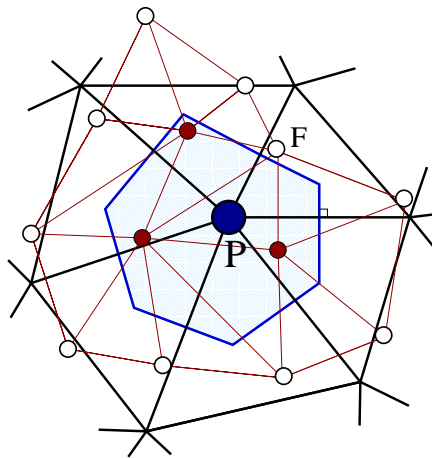


FIG. 5.5: Un des points ( $F$ ) du tétraèdre fils de  $P$  ne fait pas directement partie des fils de  $P$ .

Si l'un des quatre points  $F^i$  du tétraèdre fils ne fait pas partie des fils d'un point  $P$  donné, on obtient une configuration assez confuse lors de la subdivision de  $P$ . Sa position dépend alors en effet de celle d'un point  $F$  qui peut être également fantôme (si les zones voisines de  $P$  ne se sont pas subdivisées).

Si tel est le cas, on a alors une dérive systématique de la position du point  $P$ . Les positions des points fantômes  $F$  et  $P$  dépendent alors en effet mutuellement l'une de l'autre. Selon l'ordre dans lequel on calcule ces positions, on introduira alors forcément une dérive dans leurs positions, l'une étant mise à jour avec un pas de simulation de retard sur l'autre. Cela se traduit dans les simulations par un phénomène d'amplification des oscillations, une sorte d'inertie artificielle étant alors introduite.

Pour éviter ce problème nous avons choisi de déclarer comme actifs tous les sommets du tétraèdre fils d'un point qui se subdivise. On évite ainsi de créer cette boucle de dépendance. L'autre solution pourrait être d'utiliser des maillages qui évitent ce genre de problèmes, ce qui est faisable si l'on augmente la différence entre les résolutions de deux maillages successifs.

Plus généralement, et pour vraiment pouvoir considérer les points fantômes comme des points au déplacement correctement réactualisé, il faudra veiller à ce que leur mise à jour se fasse dans un ordre donné. Il faudra mettre leur position à jour en *remontant* dans l'arbre, des feuilles vers la racine, les points fantômes dépendant d'autres points fantômes étant mis-à-jour après ceux-ci. L'information la plus à jour contenue dans les feuilles (particules actives) sera ainsi bien propagée à tous les niveaux sans introduire de retards.

## 4.3 Intégration temporelle

Nous avons essayé d'utiliser différentes méthodes d'intégration temporelle sur ce modèle. Nous nous sommes limités à des méthodes ne demandant qu'une seule évaluation par pas de simulation<sup>2</sup>. Euler modi-

<sup>2</sup>Toutes ces méthodes sont décrites en Annexe B.

fié et Stoermer donnent rigoureusement les mêmes résultats. On préférera utiliser Euler modifié car il fournit une expression de la vitesse des points, nécessaire au calcul des forces de frottement internes. Newton-Cotes est, quand à lui, bien moins stable.

Tout comme dans le Chapitre 3 (voir page 55) consacré à notre premier modèle, les particules bénéficient ici aussi d'une multirésolution temporelle, basée sur les mêmes critères (critère de Courant et limitation de la modification de la vitesse).

### Utilisation d'un W-cycle

Différentes listes contiennent à chaque instant la liste des particules utilisant un  $dt_i$  donné. Rappelons que ces  $dt_i$  sont des sous-multiples par une puissance de deux du  $dt$  d'affichage.

$$dt_i = \frac{dt_{\text{affichage}}}{2^i}$$

Un schéma de type *leap-frog*, aussi nommé *W-cycle*, permet d'éviter que certains pas de simulations aient à simuler *tous* les  $dt_i$  alors que d'autres pas ne simuleront que le niveau  $dt_{\text{max}}$ . Si tel est le cas, la durée de calcul de ces différents pas peut varier énormément, ce qui nuit à une simulation temps-réel.

Au lieu de cela, la simulation des différents  $dt_i$  sera régulièrement répartie, chaque pas de simulation nécessitant un temps de calcul plus constant, puisqu'il ne simule que deux (voire un)  $dt_i$  à chaque étape (Fig. 5.6).

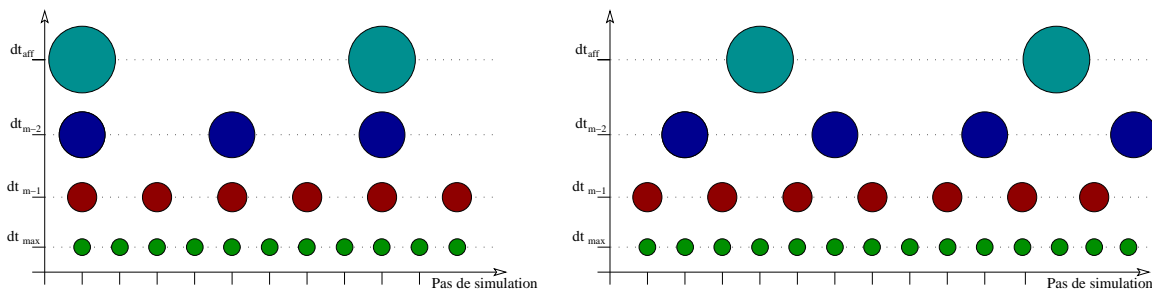


FIG. 5.6: La synchronisation décalée des simulations des différents pas de temps permet d'éviter les goulots d'étranglement en répartissant la charge sur toute la simulation.

### Intégration semi-implicite

Nous avons également implémenté une intégration *semi-implicite*, inspirée des méthodes de [DSB99]<sup>3</sup>. Cette terminaison n'est pas officielle et s'inspire de la différence entre éléments finis explicites et implicites. Plutôt que d'inverser à chaque pas de temps une matrice globale représentant tout le système comme le fait l'intégration implicite, nous allons une nouvelle fois approximer *localement* les effets de l'intégration implicite.

Nous allons précalculer pour chaque particule la matrice jacobienne de sa force  $J = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ . Cette matrice est une approximation au premier ordre de la force subie par cette particule lorsqu'elle se déplace. Pour l'estimer, nous calculons la force subie par chaque particule lorsqu'elle est déplacée dans les trois directions  $x, y$  et  $z$ , en supposant fixes et à leur position de repos toutes les autres particules.

L'amplitude de ce déplacement n'a pas d'importance dans le cas linéaire (tenseur de Cauchy, méthode hybride), un déplacement double créant une force double et donc un jacobien identique. Il n'en va pas de même lorsque l'on utilise le tenseur de Green-Lagrange, et nous choisissons alors de déplacer la particule avec une amplitude voisine de celle qu'elle subira lors de la simulation. En pratique nous avons fixé cette amplitude à 1/10ème de la distance inter-particulaire minimale du maillage auquel appartient la particule.

Comme dans [DSB99], nous allons supposer cette matrice  $J$  constante au cours du temps, ce qui permet d'approximer une intégration implicite pour un coût modique. L'intégration semi-implicite consiste alors simplement en la multiplication à chaque pas de temps de la vitesse par une matrice précalculée.

<sup>3</sup>Voir les détails en Annexe B.



$$\mathbf{v} += \frac{\mathbf{f}}{m} dt \quad \mathbf{v} *= (I_3 - \frac{dt^2}{m} J)^{-1} \quad \mathbf{x} += \mathbf{v} dt \quad (5.1)$$

L'application de ce procédé semi-implicite stabilise l'intégration temporelle et nous permet en pratique de doubler le pas de temps d'intégration sans qu'apparaissent d'instabilités.

Les forces de frottements additionnelles créées par cette méthode peuvent être compensées en réduisant les coefficients  $\phi$  et  $\psi$  qui quantifient la dissipation interne (Équation 2.22). Noter toutefois que ces frottements dissipent également les rotations rigides de l'objet, à l'inverse des frottements issus de l'Équation 2.22, ce qui peut être gênant dans certaines applications.

## 5 Simulation temps-réel

Nous allons ici expliquer comment obtenir un simulateur chirurgical temps-réel utilisant le modèle multi-résolution de ce chapitre.

### 5.1 Parallélisation

L'une des machines utilisée pour les démonstrations du prototype de simulateur comporte six processeurs. Nous avons donc cherché à utiliser cette puissance pour accélérer l'exécution du programme.

Plusieurs processus différents vont pouvoir être lancés simultanément. Ils partageront la même zone de mémoire (à l'exception du processus gérant la boucle de simulation du *Phantom*) et seront synchronisés à l'aide de sémaphores.

Le premier de ces processus gère donc le retour haptique utilisé par le bras à retour d'effort *Phantom*. Ce processus bénéficie d'une priorité *real-time* qui assure qu'il sera mis-à-jour à la fréquence de 1000Hz. Cette fréquence de rafraîchissement de la force est en effet imposée par le *Phantom* pour maintenir une sensation de force cohérente.

Notre simulation ne met pas forcément à jour la force à cette fréquence (en pratique ce sera fait à une fréquence variant entre 400 et 3700Hz) et ce processus se contente d'envoyer à nouveau la dernière force imposée en attendant la suivante. Il met également à jour les variables indiquant la position du bras. Récupérer les positions et envoyer des forces se fait donc de manière asynchrone par le programme principal de la simulation.

Les différentes étapes de la simulation proprement dites sont les suivantes (le pourcentage indique grossièrement la part du temps de la simulation consacrée à chaque partie) :

- simulation du modèle interne (60%) ;
- mise-à-jour des positions des points de la surface (5%) ;
- détection des collisions entre la surface et l'outil (20%) ;
- affichage de la scène (15%) ;
- attente éventuelle pour se synchroniser avec l'affichage.

La version mono-processus exécute ces tâches, dans cet ordre, entre deux affichages successifs.

La version multi-processus va séparer la simulation du modèle interne, partie la plus coûteuse de l'algorithme, dans un processus à part. Apparaissent alors des dépendances entre les processus : la simulation ne peut débuter qu'*après* la détection de collision et doit se terminer *avant* la mise-à-jour des points de la surface.

Nous avons décidé de décaler d'un affichage la détection et la réponse aux collisions afin de désynchroniser ces deux étapes. L'affichage prendra en compte la collision qui a été faite au pas de simulation *précédent* en non plus à celui qui vient d'avoir lieu. Ce décalage de 1/25ème de seconde n'est pas visible en pratique et la surface semble toujours bien repoussée hors de l'outil.

Le graphe représentant la simulation devient alors :

Noter qu'il est alors possible que la simulation du pas suivant commence *avant* l'affichage courant. On optimise ainsi le temps de calcul utilisable par la simulation qui peut aller jusqu'à celui séparant deux affichages moins le temps (court) nécessaire à la mise-à-jour de la surface.

Enfin, un autre processus a été utilisé pour gérer l'affichage de textures dynamiques gourmandes en calcul sur la surface du foie (gouttes de sang, blanchiment, brûlures). Nous ne détaillerons pas davantage ce procédé lié au rendu de l'objet et développé dans le cadre de l'action incitative *AISIM*.

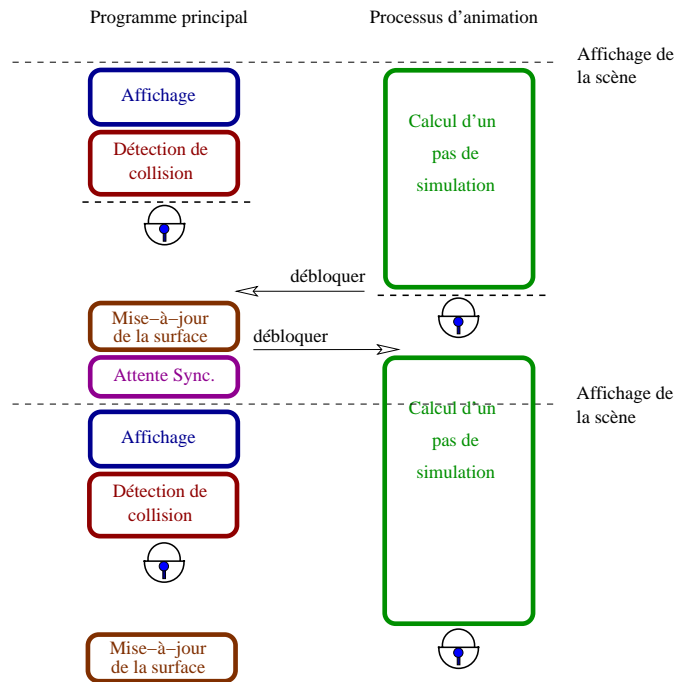


FIG. 5.7: Le programme principal et le processus chargé de l'animation sont liés par un système de sémaphores assurant leur synchronisation.

## 5.2 Temps-réel vrai

Un petit processus annexe sert uniquement à compter le nombre de rafraîchissements d'écran effectués depuis le dernier affichage. Il sert à assurer une simulation en temps-réel vrai en faisant en sorte que le programme attende éventuellement pour afficher une nouvelle image (étape "Attente Sync" de la Figure 5.7). On garantit ainsi que, même si l'on peut calculer plus vite que nécessaire, on attendra pour respecter le temps-réel.

Cette attente, même infime est nécessaire, et il faudra veiller à ne pas *dépasser* le temps imparti, ce qui décalerait l'affichage d'une *frame*. Pour ce faire, on peut limiter le nombre de particules de la simulation en limitant la subdivision.

Une variable indique en permanence le pourcentage de temps passé à la simulation pour le pas précédent, par rapport au temps total disponible entre deux affichages. Un seuil à ne pas dépasser (95% dans nos exemples) détermine s'il est encore possible de subdiviser. On garantit ainsi un affichage régulier de la scène.

## 6 Résultats

Nous présentons ici quelques images issues de la simulation. La Figure 5.8 montre l'influence de l'augmentation du terme de préservation de volume sur la forme de l'objet déformé.

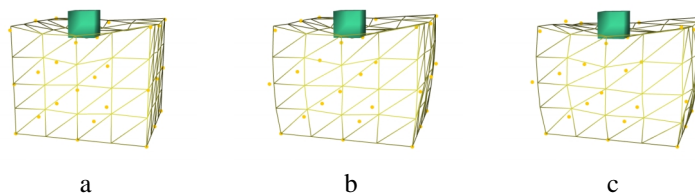


FIG. 5.8: Influence de l'augmentation de  $\lambda$  sur la préservation de volume. Les images représentent  $\lambda = 0$  (a),  $\lambda = 50000$  (b) et  $\lambda = 500000$  (c).

Lorsque l'outil vient appuyer trop fortement sur la surface, les maillages plus fins deviennent actifs dans cette zone, montrant ainsi l'adéquation du critère basé sur le laplacien pour définir les points à subdiviser ou à regrouper.

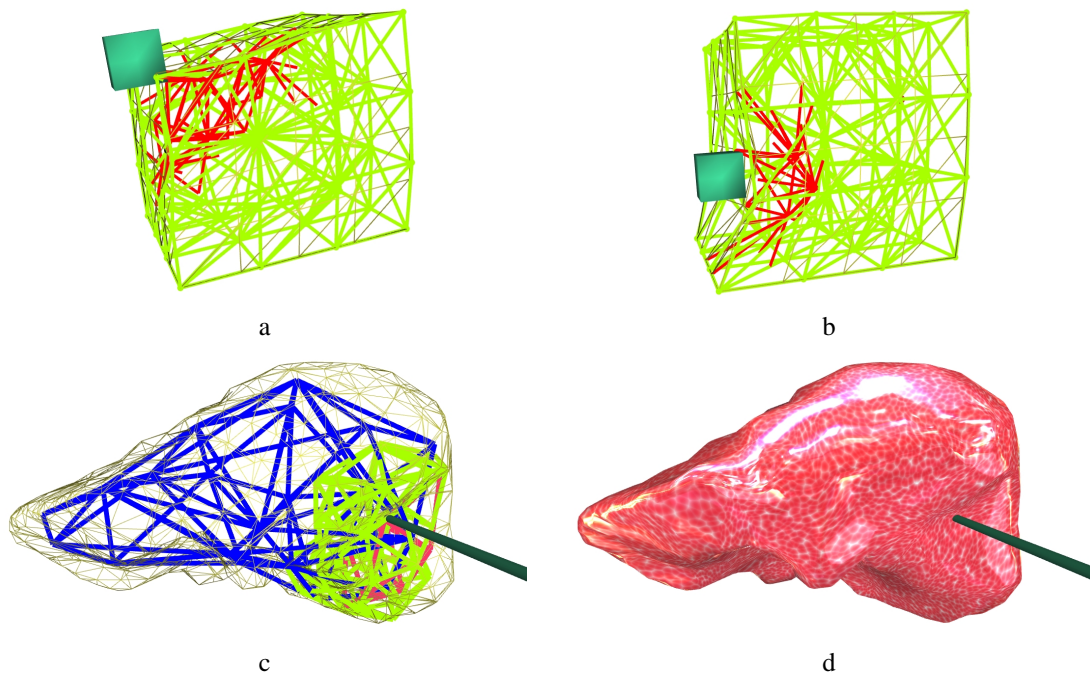


FIG. 5.9: L'utilisation des maillages fins, faite d'après des critères liés à la continuité, résulte en une discrétisation intuitive des zones proches de l'outil.

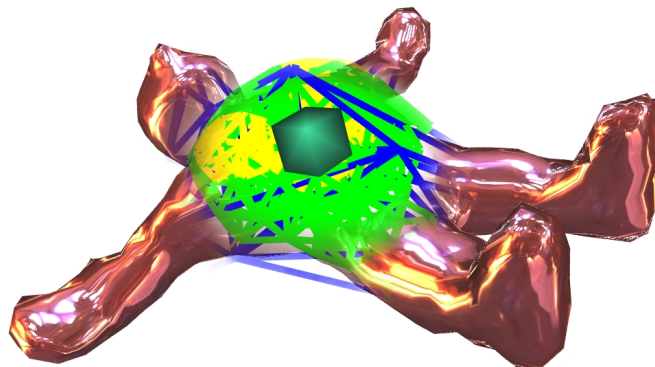


FIG. 5.10: Autre exemple d'objet déformable. Les secousses appliquées sur le ventre de ce bonhomme se répercutent sur ses bras qui oscillent légèrement.

## 7 Conclusion

Nous avons présenté dans ce chapitre un deuxième modèle multirésolution, utilisant les opérateurs décrits dans le chapitre précédent. L'originalité de cette approche réside dans l'utilisation de maillages *indépendants* pour représenter l'objet. Chaque zone de l'objet est réellement simulée en utilisant un et un seul des maillages, choisi en fonction des besoins de la simulation.

Ces différentes zones interagissent avec les zones voisines, simulées par d'autres maillages, grâce à l'utilisation de points fantômes. En propageant la déformation calculée d'un maillage à un autre, ces points fantômes permettent aux maillages de connaître la déformation de leur voisinage et peuvent ainsi simuler leur propre déformation. Bien que composé de différents maillages, l'objet forme donc un tout cohérent.

Ce modèle multirésolution étant mis au point, il reste à l'"habiller" en lui associant une surface. Ce lien entre la surface et le modèle interne est le propos du prochain chapitre.

