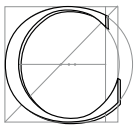
CHAPITRE 3

Premier modèle multirésolution



CALCULER le mouvement d'un objet à plusieurs résolutions est indispensable pour optimiser les calculs et accélérer la simulation. Nous avons décrit dans le chapitre précédent la théorie de l'élasticité qui permet de s'abstraire, au moins partiellement, de la résolution utilisée.

Ce chapitre va présenter un premier modèle d'algorithme multirésolution mettant en œuvre les équations de la mécanique des milieux continus vues dans le précédent chapitre. Il a fait l'objet d'une publication au 10 *ème* Workshop Eurographics sur l'animation et la simulation [DDBC99].

Plus précisément, nous sommes ici partis de l'équation de Navier, qui donne directement l'accélération subie par un point en fonction des déplacements des points voisins (Eq. 2.19) :

$$\rho\ \mathbf{a} = \mu\ \mathbf{div}(\mathbf{grad}\ \mathbf{u})\ +\ (\mu+\lambda)\ \mathbf{grad}(\mathrm{div}\ \mathbf{u}) \tag{3.1}$$

Le chapitre précédent a détaillé les hypothèses faites et l'interprétation de cette équation (voir Section 3.3). L'algorithme que nous allons développer (sans prendre en compte dans un premier temps l'aspect multirésolution) a la simplicité de ceux des modèles de particules.

Partant d'une discrétisation de l'objet, à chaque pas de temps on va :
- calculer en chaque point **div** (**grad u**) et **grad**(div **u**) ;
- en déduire l'accélération en utilisant l'équation de Navier ;
- intégrer cette accélération et les éventuelles forces extérieures (gravité) ;
- en déduire les nouvelles positions des points et donc leurs déplacements respectifs ;
- et ainsi de suite...

La seule difficulté réside dans le calcul des deux opérateurs différentiels. Des méthodes comme les *différences finies* permettent de calculer, grâce à des développements de Taylor, l'expression de dérivées partielles arbitraires. Elles nécessitent par contre de connaître la valeur du champ dont on veut calculer la différentielle en des points *équirépartis* sur une *grille régulière*. On veut ici pouvoir discrétiser un objet de topologie arbitraire. L'introduction de nouveaux points pour la multirésolution va de plus compliquer ensuite la discrétisation, pouvant amener à une grille quelconque, ce qui fait que cette méthode est mal adaptée.

C'est la raison pour laquelle nous avons cherché à définir de nouveaux opérateurs différentiels capables d'approximer les différentielles sur des points d'échantillonnage arbitrairement répartis. Le laplacien **div** (**grad u**)

est le premier des deux termes que nous avons calculés. Son expression a ensuite été reprise pour formuler **grad**(div **u**).

Une fois ces opérateurs exprimés, nous pourrons les utiliser dans un algorithme multirésolution, profitant de ce que l'échantillonnage puisse être arbitraire. Nous détaillerons la façon dont nous adaptons la résolution locale au cours de la simulation, élaborant ainsi l'une des premières méthodes de simulation multirésolution pour un matériau élastique.

1 Calcul du laplacien

Le laplacien $\Delta\mathbf{u}$ du champ **u**, qui est égal au **div** (**grad u**) est pour chacune de ses composantes la somme des dérivées secondes dans les trois directions :

$$(\Delta\mathbf{u})_i = u_{i,xx} + u_{i,yy} + u_{i,zz} \tag{3.2}$$

1.1 Dérivée seconde scalaire

Le but est donc ici de savoir calculer un des ces termes $u_{i,jj} = \frac{\partial^2 u_i}{\partial x_j^2}$, dérivée seconde du champ scalaire u_i dans une direction j quelconque.

S'inspirant des différences finies, on peut exprimer classiquement la dérivée seconde de f au point p comme :

$$f''(p) = \frac{f(p+h) \; - \; 2f(p) \; + \; f(p-h)}{h^2}$$

où h est la distance entre p et ses deux voisins situés en $p+h$ et $p-h$. Cette expression est valide à l'ordre 2, les termes négligés étant de l'ordre de $O(h^2)$.

Milne avait dans sa thèse [Mil95] montré combien la dérivée était sensible au bruit dès lors que l'on n'a plus affaire à une dérivée centrée. Fornberg [For88] avait proposé une extension de la formule précédente pour traiter le cas où les deux voisins de p ne sont plus situés à la même distance h, mais à des distances δ et ε. Son calcul revient à chercher la courbure du polynôme de degré deux passant par les trois points. La dérivée s'exprime alors par :

$$f''(p) = \frac{2}{\delta+\varepsilon} \left(\frac{f(p+\delta) - f(p)}{\delta} \; + \frac{f(p+\varepsilon) - f(p)}{\varepsilon} \right)$$

On notera que dans le cas où $\delta = \varepsilon = h$ on retrouve l'expression précédente, ce qui est normal puisque toutes deux s'obtiennent en regroupant les développements de Taylor de f faits en p avec chacun de ses voisins. Si les points ne sont pas à la même distance, on n'obtient par contre plus qu'une formule valable à l'ordre 1, les termes négligés étant de l'ordre de $O(\delta, \varepsilon)$.

1.2 Passage en trois dimensions

L'expression du laplacien que nous allons utiliser en 3D est une simple extrapolation de la formule précédente. En lieu et place des deux voisins précédents, nous avons maintenant n voisins répartis autour du point. Nous ne calculons plus la dérivée seconde le long d'un axe donné (celui sur lequel sont les points dans l'exemple 1D précédent), mais la somme de toutes ces dérivées secondes dans *toutes* les directions. On obtient donc en un point i une approximation du laplacien, somme de ces dérivées secondes :

$$(\Delta f)^i = \frac{2}{\sum\limits_{\text{voisins } j} l^{ij}} \sum_{\text{voisins } j} \frac{f^j - f^i}{l^{ij}} \tag{3.3}$$

f^j est la valeur du champ (scalaire) au voisin j et l^{ij} la distance avec ce voisin.

Cette expression a déjà été utilisée par Fujiwara dans le cadre de l'analyse d'image [Fuj95] et fut reprise par Desbrun *et al.* pour le lissage de maillages [DMSB99].

Il est à noter qu'on retrouve exactement l'expression des différences finies dès lors que l'on se place sur une grille régulière. On peut donc voir cette formule, avec ses coefficients dépendant des distances aux voisins

comme une extension de différences finies, les poids associés à la valeur f^j de la fonction dépendant de la configuration dans laquelle on se trouve.

Cette formulation peut également être vue comme un filtrage par analogie au noyau de filtrage W_h utilisé dans les SPH (voir Sec. 5, Chap. 1). L'importance d'un point (de la *valeur* de la fonction en ce point) est en effet inversement proportionnelle à la distance qui le sépare du point considéré. On tiendra donc davantage compte des voisins proches ce qui est naturel puisque c'est leur valeur qui influencera le plus l'aspect local de la fonction, et donc le laplacien.

1.3 Laplacien d'un champ vectoriel

On cherche le laplacien *vectoriel* du champ déplacement $\mathbf{u}$[1]. Puisque chacune de ses composantes est le laplacien de la composante associée de $\mathbf{u}$, on a simplement :

$$\boxed{(\Delta \mathbf{u})^i = \frac{2}{\sum\limits_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\mathbf{u}^j - \mathbf{u}^i}{l^{ij}}} \tag{3.4}$$

pour un point i donné, en fonction de ses voisins j situés à la distance l^{ij}.

2 Extension au grad (div)

Le **grad**(div **u**) que nous cherchons à calculer s'écrit, pour chacune de ses composantes :

$$[\mathbf{grad}(\text{div } \mathbf{u})]_i = u_{x,xi} + u_{y,yi} + u_{z,zi} \tag{3.5}$$

Nous allons, pour calculer cet opérateur, extrapoler l'Équation 3.4 du laplacien, les deux formulations étant donc développées dans un cadre unifié. Le **grad**(div **u**) est en effet lié au laplacien précédemment calculé par l'équation :

$$\Delta \mathbf{u} = \mathbf{grad}(\text{div } \mathbf{u}) \; - \; \mathbf{rot}(\mathbf{rot}\, \mathbf{u}) \tag{3.6}$$

Il faut donc, dans l'expression du laplacien, simplement parvenir à différentier les deux termes de cette somme pour isoler la contribution de **grad**(div **u**). Cela se fait en remarquant que le double rotationnel va précisément mesurer la rotation du champ **u** et que celle-ci peut être extraite du vecteur déplacement. Celui-ci peut en effet être considéré comme la somme d'une composante purement radiale due aux forces de pression et d'une composante tangentielle indiquant localement la rotation du champ **u** (voir Fig. 3.1).

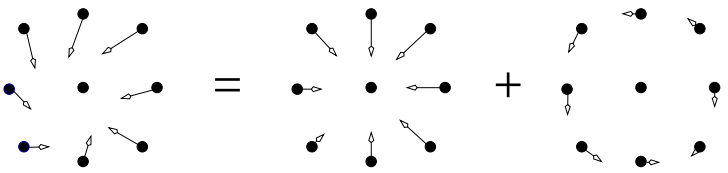


FIG. 3.1: Le champ de déplacement peut être séparé en deux composantes : la radiale, créée par les forces de pression, et la rotationnelle, créée par les forces de cisaillement.

Ce ne sont pas directement les valeurs $\mathbf{u}^j$ du déplacement des voisins qui nous intéressent, mais plutôt leur *déplacement relatif*, $\mathbf{u}^j - \mathbf{u}^i$, par rapport au point i où l'on cherche à calculer la différentielle. On se place en quelque sorte ici dans un repère lié au point où l'on fait les calculs et l'on observe comment se sont déplacés les voisins en supposant notre position fixe.

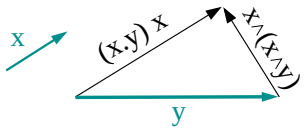


FIG. 3.2: **y** s'écrit comme la somme de deux vecteurs orthogonaux en fonction du vecteur unitaire **x**.

2.1 Projection le long de l'axe

Comme on le voit sur la Figure 3.2, un vecteur **y** peut s'écrire un utilisant en vecteur unitaire **x**, comme la somme d'un vecteur aligné avec **x** et d'un vecteur orthogonal à **x** grâce à la relation : $\mathbf{y} = (\mathbf{x}\cdot\mathbf{y})\mathbf{x} - \mathbf{x}\wedge(\mathbf{x}\wedge\mathbf{y})$, $\wedge$ désignant le produit vectoriel et $\cdot$ le produit scalaire.

On notera $\mathbf{l}^{ij}$ le vecteur reliant le point i à son voisin j et l^{ij} sa norme. Nous pouvons alors réécrire en la décomposant l'équation du laplacien 3.4 en prenant pour **x** le vecteur unitaire $\mathbf{x} = \mathbf{l}^{ij}/l^{ij}$ et pour **y** le vecteur déplacement relatif $\mathbf{u}^j - \mathbf{u}^i$:

$$
\begin{aligned}
(\Delta\mathbf{u})^i &= \frac{2}{\sum_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\mathbf{u}^j - \mathbf{u}^i}{l^{ij}} \\
&= \frac{2}{\sum_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\left[(\mathbf{u}^j - \mathbf{u}^i)\cdot\frac{\mathbf{l}^{ij}}{l^{ij}}\right]\frac{\mathbf{l}^{ij}}{l^{ij}} - \frac{\mathbf{l}^{ij}}{l^{ij}}\wedge\left[\frac{\mathbf{l}^{ij}}{l^{ij}}\wedge(\mathbf{u}^j - \mathbf{u}^i)\right]}{l^{ij}} \\
&= \underbrace{\frac{2}{\sum_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\left[(\mathbf{u}^j - \mathbf{u}^i)\cdot\frac{\mathbf{l}^{ij}}{l^{ij}}\right]\frac{\mathbf{l}^{ij}}{l^{ij}}}{l^{ij}}}_{\textit{Composante radiale}} - \underbrace{\frac{2}{\sum_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\frac{\mathbf{l}^{ij}}{l^{ij}}\wedge\left(\frac{\mathbf{l}^{ij}}{l^{ij}}\wedge(\mathbf{u}^j - \mathbf{u}^i)\right)}{l^{ij}}}_{\textit{Composante rotationnelle}}
\end{aligned}
\tag{3.7}
$$

En comparant cette équation avec 3.7, nous choisissons d'identifier la composante radiale comme étant le gradient de la divergence :

$$
\boxed{[\mathbf{grad}(\operatorname{div}\mathbf{u})]^i = \frac{2}{\sum_{\text{voisins j}} l^{ij}} \sum_{\text{voisins } j} \frac{\left[(\mathbf{u}^j - \mathbf{u}^i)\cdot\frac{\mathbf{l}^{ij}}{l^{ij}}\right]\frac{\mathbf{l}^{ij}}{l^{ij}}}{l^{ij}}} \tag{3.8}
$$

Cette extrapolation à partir de la formule du laplacien possède une autre justification, plus intuitive. Comme il l'a été dit dans la discussion de l'équation de Navier chapitre 2, le terme $\mathbf{grad}(\operatorname{div}\mathbf{u})$ va chercher à préserver le volume du matériau. Localement, la variation de ce volume peut-être représentée comme celle du volume entourant un point, défini par ses plus proches voisins. Il est donc normal de ne considérer que le déplacement *radial* des voisins, car c'est cette composante uniquement qui affecte le volume.

Nous reviendrons à la fin de ce chapitre sur les inconvénients de cette assimilation.

3 Simulation à résolution fixe

3.1 Cas d'école

Nous avons maintenant grâce aux Équations 3.4 et 3.8 le moyen de calculer l'accélération de tout point pour appliquer l'algorithme décrit au début de ce chapitre. Cette accélération est une fonction du champ de déplacement, échantillonné sur ses points voisins. Nous allons mesurer la qualité de ces opérateurs différentiels en simulant un ensemble de points. L'exemple choisi est celui d'un cube de matière uniforme, dont l'une des faces est collée sur un mur. Partant de sa position au repos avec une vitesse nulle, nous allons simuler ses oscillations sous l'effet de la gravité.

Nous avons volontairement choisi de ne faire intervenir aucune force dissipative (frottement de l'air, viscosité) de manière à mettre en évidence le comportement du modèle sur un cas d'école.

[1] Voir Annexe A.

Les résultats visuels obtenus sont assez satisfaisants. Ils sont conformes au modèle de Cauchy présenté au Chapitre 2. Le cube se déforme sans trop plier, oscillant principalement verticalement. Augmenter la raideur du matériau crée des oscillations plus rapides et de plus faible amplitude.

La Figure 3.3 représente l'altitude de l'un des coins du cube en fonction du temps. Les différentes courbes ont été obtenues avec plus ou moins de points simulés correspondant à des maillages réguliers de respectivement $4^3 = 64$, $8^3 = 512$ et $16^3 = 4096$ particules.

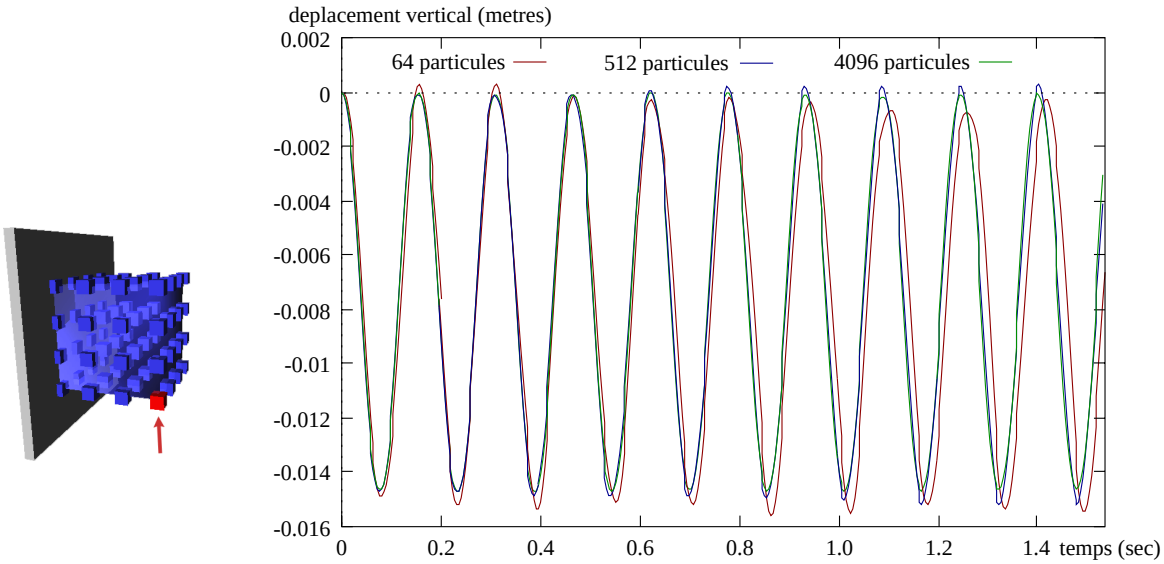


FIG. 3.3: L'altitude de l'un des coins du cube (indiqué par une flèche) lors de la simulation, pour différents maillages réguliers. Aucun frottement n'a été ajouté dans cette simulation.

L'absence de forces de dissipation explique ces oscillations continues et régulières non amorties. Peu de méthodes sont suffisamment "propres" pour permettre ce genre de simulation sans divergence numérique (on intègre avec un schéma d'Euler modifié[2]). L'existence d'une position de référence servant d'attracteur y est probablement pour beaucoup.

La forte ressemblance des courbes obtenues confirme l'indépendance à la discrétisation obtenue grâce à l'utilisation de l'équation de Navier. Les opérateurs différentiels présentés fournissent bien un calcul relativement indépendant de la résolution. Cette propriété est un élément essentiel pour pouvoir passer à une méthode mélangeant à chaque instant les différentes résolutions.

3.2 Ajout de forces dissipatives

Le réalisme de la simulation peut être amélioré par l'ajout de forces dissipatives. Ceci aura pour effet de stabiliser plus ou moins rapidement les oscillations de l'objet, se rapprochant ainsi des matériaux réels où ces forces apparaissent naturellement. Ceci peut se faire à l'aide d'une force de frottement visqueuse classique, inversement proportionnelle à la vitesse :

$$\mathbf{F}_d = -k_d.\mathbf{v}$$

Nous avons également ajouté une force de viscosité dans notre modèle en modélisant les interactions des particules entre elles. Cette force va faire en sorte que chaque particule soit affectée par le mouvement de ses voisines et tente de suivre leur mouvement global, ajoutant ainsi une cohérence interne au matériau. Inspirée par la formulation SPH de [Mon92, DCG96], nous exprimons cette force comme :

$$\mathbf{F}_v^i = \frac{k_v}{\sum\limits_{\text{voisins } j} m_j} \sum_{\text{voisins } j} m_j(\mathbf{v}_j - \mathbf{v}_i) \tag{3.9}$$

[2] Voir Annexe B.

On peut voir cette équation comme une extension de celle utilisée dans les cas d'échantillonnage régulier de l'espace. L'influence d'un voisin a été choisie comme proportionnelle à sa masse.

Cette force est filtrée pour n'en conserver que son aspect dissipateur et on l'annule si elle accélère la particule ou lui fait changer la direction de sa vitesse.

Bien que n'offrant pas de garantie d'un comportement indépendant de la résolution, cette formulation inspirée de [Mon92] ne crée pas d'artefact trop visible. Nous utiliserons des coefficients assez faibles dans nos exemples, principalement pour obtenir un matériau plus rigide.

Nous aurions pu utiliser le formalisme d'élasticité linéaire décrit en Section 4 du chapitre précédent. Cela n'a pas été fait car au moment du développement de cette méthode, nous n'avions pas développé les formules qui y sont présentées, qui sont une extrapolation de celles trouvées dans d'autres articles de la littérature [OH99] au cas du tenseur infinitésimal de Cauchy.

4 Simulation multirésolution

La méthode multirésolution présentée ici va être *adaptative* selon la définition faite en Section 10 du chapitre 1. Nous allons précalculer plusieurs maillages volumiques, définir comment une zone de l'espace peut être simulée par l'un ou l'autre de ces maillages et comment elle interagit alors avec le reste de l'objet pour enfin voir comment et sur quels critères imposer un changement de niveau de détail.

4.1 Création des maillages

Les maillages que nous allons créer seront issus d'une division de l'espace en cubes. Le cube est en effet la seule figure géométrique *régulière* pouvant remplir l'espace. On peut mailler un volume à l'aide de tétraèdres, mais ceux-ci ne pourront avoir tous la même forme. La découpe d'un tétraèdre en de nouveaux tétraèdres plus petits dégrade par ailleurs largement leur qualité, plusieurs applications du processus conduisant à des tétraèdres très irréguliers. On garantit en utilisant des cubes une répartition la plus régulière possible des points d'échantillonnage.

Le maillage dont on va partir est régulier, aligné avec les axes et de même résolution dans les trois directions x, y et z. On ne considère que les nœuds de ce maillage qui se trouvent *à l'intérieur* de l'objet [3]. Ce seront les particules que nous allons simuler. Leur masse est fixée comme étant celle de l'objet divisée par le nombre de particules (si l'objet est supposé de densité uniforme, les points d'échantillonnage étant uniformément répartis).

On construit ensuite récursivement les différents maillages en partant de celui-ci, qui sera le plus fin. Chaque maillage se déduit du précédent en regroupant chaque bloc de 8 particules en une nouvelle particule, celles-ci formant le nouveau maillage (voir Fig. 3.4).

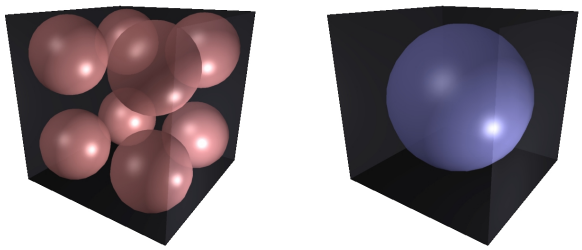


FIG. 3.4: Un même cube sera échantillonné par huit particules filles ou par leur mère.

Une telle particule échantillonnera ainsi un volume de taille huit fois moindre que celui échantillonné par les particules précédentes. On dénotera par les termes de *mère* et *filles* les particules ainsi liées. On va également implicitement classer ces maillages du bas vers le haut, en allant du plus fin vers le plus grossier. Une fille appartiendra ainsi au niveau *inférieur* de celui de sa mère.

Lorsqu'on construit un maillage à partir du précédent, on peut regrouper moins de 8 particules ensemble si on se trouve au bord (voir Fig. 3.5). Dans tous les cas, et pour conserver la masse totale et sa répartition,

[3] On pourrait choisir de déplacer les points situés près du bord pour les amener sur la surface, mais cette optimisation n'est pas nécessaire. Les maillages, même s'ils doivent être proches de la géométrie de l'objet, seront en effet décorrélés de la surface qui va être affichée (voir Chapitre 6), et seul leur mouvement global nous importe.

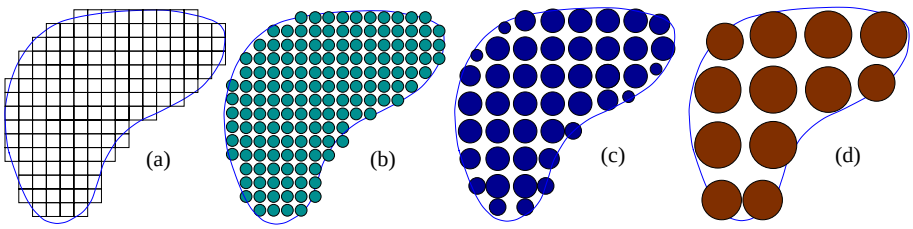


FIG. 3.5: Création récursive des maillages. La grille régulière qui sert à créer le maillage le plus fin (a). On regroupe ensuite les particules pour créer les maillages suivants (b), (c) et (d). Les tailles représentent la masse des particules.

la nouvelle particule aura pour masse la somme de celles de ses filles et une position définie comme leur barycentre pondéré par les masses :

$$m = \sum_{filles\ i} m_i \qquad \mathbf{p} = \frac{\sum_{filles\ i} m_i\, \mathbf{p_i}}{m} \tag{3.10}$$

où m est la masse, et $\mathbf{p}$ la position de la particule mère.

Les maillages ainsi créés ne seront donc plus exactement réguliers, mais la position de chacune des nouvelles particules est optimale au sens de la répartition de masse et l'on crée donc un maillage adapté à la morphologie de chaque objet.

En regroupant ainsi par huit les particules entre elles, on crée un *octree* topologique de particules. Ce n'est pas un octree au sens classique (des cases cubiques régulièrement divisées en huit autres cases) puisque les maillages ne sont plus réguliers et vont pouvoir se déformer au cours de l'animation, mais ça l'est au sens topologique de *voisinage*. Les relations de voisinages entre les particules sont quant à elles bien organisées dans une structure d'octree, des voisines pouvant être définies dans chaque direction (bien que n'existant pas forcément au bord).

4.2 Définition des voisines

Voisines de même niveau

Nous avons choisi qu'une particule se serve, pour calculer les opérateurs différentiels, de ses voisines immédiates. Ce sont les 26 particules (il peut y en avoir moins si on est au bord de l'objet) qui touchent le cube associé à la particule par une face (6), une arête (12) ou un coin (8), et qui sont représentées Figure 3.7 b.

Voisines appartenant aux autres maillages

Une particule pourra également, et c'est la clef des algorithmes multirésolution, prendre en compte des particules se trouvant à côté d'elle mais ne faisant pas partie du maillage de ce niveau. Nous avons choisi de limiter cette interaction entre deux niveaux de détail aux seuls maillages précédant et suivants : parmi les maillages successifs de l'objet, le maillage du niveau n ne pourra utiliser que les niveaux $n+1$ et $n-1$, s'ils existent (voir Fig. 3.6).

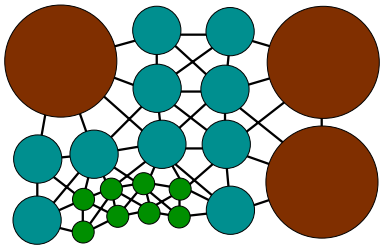


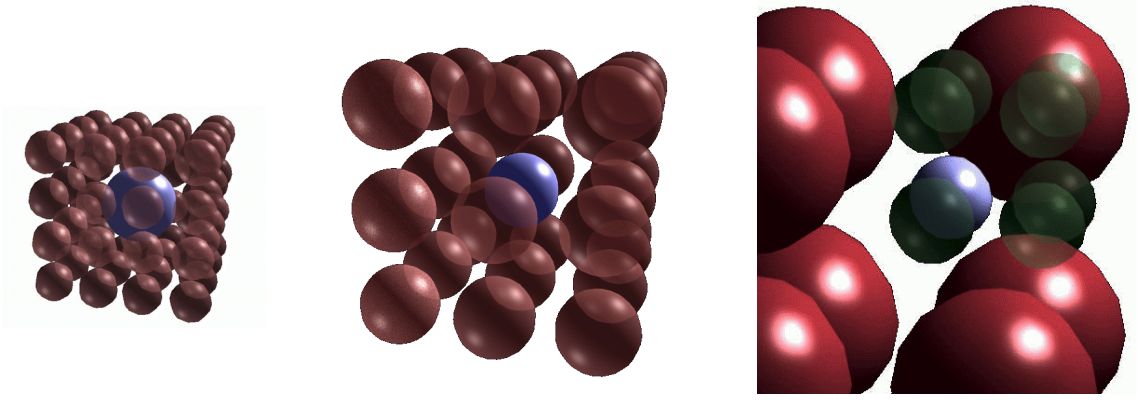
FIG. 3.6: Durant la simulation, les voisins d'une particule ne peuvent appartenir qu'au même niveau de résolution, au niveau supérieur ou au niveau inférieur.

Ce choix limite la souplesse de la méthode, mais créer un tel *octree restreint* (définition de [VB87]) a deux avantages :

– La structure de voisinage est limitée et plus simple. On va faire l'hypothèse que l'objet se déforme suffisamment peu pour que les voisines d'une particules ne changent pas au cours de la simulation. Elles pourront donc être définies et toutes stockées au début de la simulation et on évitera les coûteuses remises à jour de cette structure.
– Imposer des niveaux de résolution proches entre deux particules voisines va garantir que la discrétisation du matériau sera à tout instant continue. Il ne serait pas naturel que deux zones juxtaposées aient deux résolution trop différentes. Si on a besoin de beaucoup de précision à un endroit et de très peu juste à côté, il vaut mieux passer continuement d'une résolution à l'autre.

Il faudra donc garantir au cours de la simulation que chaque particule ne soit entourée que par des particules de niveau directement inférieur ou supérieur

En ayant ainsi limité les maillages parmi lesquels une particule va avoir des voisines, on peut stocker dans des listes toutes les voisines possibles, parmi lesquelles toutes ne seront pas forcément activées à un instant donné. Outre les 26 particules du même niveau déjà décrites, cette liste comporte 7 particules au niveau supérieur (les mères des précédentes) et 56 particules appartenant au niveau inférieur (les filles des voisines du même niveau qui jouxtent la particule considérée). Ce sont en fait toutes les particules qui touchent le cube associé à la particule (Fig. 3.7). Chaque particule aura donc ainsi 89 voisines potentielles.



(a) (b) (c)

FIG. 3.7: Les 56 voisines de niveau inférieur (a), les 26 voisines de même niveau (b) et les 7 voisines de niveau supérieur (et les "sœurs") (c) d'une particule donnée.

4.3 Simulation adaptative

À tout instant, chaque zone du matériau va être représentée par une et une seule particule. Cette particule aura une "taille" plus ou moins grosse (i.e. sera issue d'un maillage plus ou moins grossier) en fonction de la simulation. En pratique, un objet au repos sera représenté par les quelques particules du niveau le plus grossier que l'on aura créé. On appellera les particules servant ainsi effectivement à la simulation les particules *actives*. Toutes les particules des autres résolutions ne serviront pas à cet instant et seront qualifiées d'*inactives*.

S'il se passe quelque chose dans une région, une particule de cette région va pouvoir se *diviser* et être remplacée par ses 8 filles (encore une fois il peut ne pas y avoir 8 filles mais moins, si on considère une particule située au bord de l'objet), qui échantillonneront alors mieux la région. La particule mère qui s'est divisée devient *inactive*, ses filles accédant au statut d'*actives*. La simulation ne prendra plus alors en compte que les filles créées, qui cohabiteront avec les particules non divisées des zones voisines.

On peut avoir à diviser encore une ou plusieurs des filles créées, en les désactivant pour les remplacer également par leur filles, devenues alors actives, et ainsi de suite.

Inversement, lorsque cette zone est soumise à des contraintes moindres, les 8 filles peuvent se *regrouper* et laisser leur place à leur mère. Celle-ci redevient alors active et ses filles inactives. Ces différents processus de division et regroupement de particules devront néanmoins veiller à conserver la structure d'octree restreint de l'arbre. Ainsi, la nécessité de division d'une particule pourra entraîner d'autres divisions dans son voisinage.

La simulation ne va réellement utiliser à chaque instant que les particules actives. Celles-ci vont calculer des forces en fonction des déplacements de celles parmi leurs voisines qui sont également actives, forces qui seront intégrées comme précédemment. La différence avec l'algorithme à résolution fixe donné au début de ce chapitre est que des critères vont pouvoir à tout instant venir modifier la liste des particules actives, et donc les relations de voisinage entre particules, en effectuant des divisions et des regroupements.

4.4 Position relative mère-fille

Repère local

Lorsqu'une particule se divise, elle fait apparaître 8 filles qui vont la remplacer. La position de ces nouvelles particules doit être bien choisie. Si on se contente de placer les 8 filles dans la même position que celle qu'elles occupaient par rapport à leur mère au moment où l'on a construit les maillages, on va assister à des instabilités. En effet, le matériau étant déformé, les filles apparaîtront à des positions non déformées peu intuitives, source d'instabilités que l'on veut minimiser lorsque l'on change de résolution.

On va donc définir la position des filles par rapport à la mère dans un *repère local*, lié au matériau. Ce repère a pour origine la mère et pour axes trois directions définies comme suit. Pour chaque fille et pour chaque direction x, y ou z, il existe 4 particules issues du même maillage que la mère et formant la *face* la plus proche de la fille, orthogonale à la direction donnée. C'est le barycentre pondéré par les masses de ces faces qui définit la direction de l'axe.

La Figure 3.8 explicite cette définition en 2D. On n'a ici que deux axes, définis par deux faces. Les faces sont constituées des deux particules qui vont le mieux échantillonner la déformation dans chacune des directions. On calcule ensuite le barycentre de ces faces pour obtenir l'axe du repère.

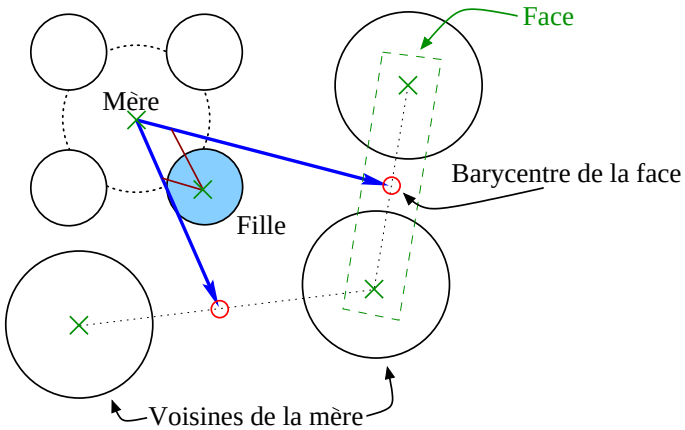


FIG. 3.8: Le système de coordonnées locales (flèches), défini par les voisines de la mère assure un bon placement des filles, même lorsque l'objet est déformé.

Les coordonnées de chacune des filles dans son repère local propre seront calculées dans la position non déformée et ne changeront pas. Ces axes, définis par rapport aux voisines de la mère, vont "suivre" la déformation du matériau et vont permettre de trouver une position intuitive pour les filles lorsqu'elles apparaissent.

Ces difficultés de positionnement n'apparaissent pas lorsque les filles se regroupent, la position de la mère se déduisant simplement grâce à une moyenne pondérée par les masses de celles de ses filles (Eq. 3.10).

Cas particuliers

La définition proposée n'est plus valable lorsqu'on l'applique aux particules du bord. Les faces que nous avons définies n'ont alors pas forcément 4 particules. Si elles en ont moins, c'est tout de même la seule source d'information que nous ayons sur les déformations dans cette direction et nous conservons la même définition, le barycentre des particules de la face comprenant simplement moins de 4 particules.

Il se peut quand on est tout au bord qu'il n'y ait *aucune* particule voisine de la mère pour composer une face dans une direction donnée. La meilleure approximation des déformations ayant eu lieu dans cette direction est alors donnée par la *face symétrique* de la face manquante par rapport à la mère.

Dans le cas ultime ou même cette face symétrique n'est pas définie on se rabat sur l'utilisation d'axes fixes dans les directions x, y ou z.

L'autre subtilité de la définition de ces axes est qu'au moment où la mère se divise, ses particules voisines qui définissent les faces ne sont pas forcément actives. Leur position n'a donc pas été mise à jour depuis le moment où elles se sont divisées[4].

On va dans ces cas là simplement mettre à jour leur position en fonction de celles de leurs filles par l'Équation 3.10. Cette remise à jour n'est pas très coûteuse, mais on pourrait décider ou non de la faire en fonction du temps écoulé depuis la division de la particule et donc de l'imprécision de sa dernière position. Il faut dans tous les cas savoir de quand date la dernière estimation de position pour éviter par exemple de faire ainsi plusieurs fois remonter la position depuis les filles lorsqu'une particule sert dans plusieurs des faces que l'on va utiliser, ce qui est souvent le cas.

4.5 Changement de résolution

Nous allons détailler les critères qui vont décider quelle résolution doit être appliquée en chaque zone de l'objet, ou plus précisément quelles sont les particules qui ont à se regrouper ou à se diviser à une étape donnée de la simulation. Ces choix sont faits en deux passes. La première fait une présélection des particules candidates (à la division ou au regroupement) et la deuxième filtre ces listes pour satisfaire le principe d'octree restreint.

Critère de continuité

On va décider de diviser (resp. regrouper) des particules en fonction de l'importance de la déformation dans la zone considérée. Ce n'est pas en fait de son amplitude, ni même de sa variation que l'on va se soucier, mais plutôt de sa dérivée seconde. Une amplitude constante voire même variant linéairement va être bien gérée par les opérateurs différentiels présentés plus haut car ils ont une précision du premier ordre. Les évolutions d'ordre plus élevé de la déformation, en commençant par la dérivée seconde, seront par contre mal approximées et il faudra pour mieux les gérer diviser les particules dans cette zone.

Il se trouve que le laplacien est justement une très bonne mesure du caractère non linéaire du champ déplacement puisqu'il mesure la somme des dérivées secondes principales. C'est sa norme, calculée en chaque point et multipliée par un terme correctif relatif à la résolution employée, qui va nous servir d'indicateur :

$$\overbrace{\varepsilon_{min} > h^2 \|\underbrace{\Delta \mathbf{u}}_{division}\| > \varepsilon_{max}}^{regroupement} \tag{3.11}$$

où h représente la plus petite distance entre cette particule et ses voisines. Le terme en h^2 vient pondérer la valeur du laplacien tolérable en fonction de la discrétisation locale.

Si elle est supérieure à un seuil ε_{max} donné, on va ajouter cette particule dans la liste de celles devant se diviser. Au contraire, si elle est inférieure à ε_{min} *pour les huit filles* d'une particule, celles-ci vont se placer dans la liste des particules à regrouper. Cette approximation rapide, basée sur la continuité, donne de bons résultats en pratique, le raffinement apparaissant où et quand nous l'espérions intuitivement.

L'écart entre les deux seuils ε_{min} et ε_{max} fait qu'il sera plus facile à une particule de se diviser qu'à ses filles de se regrouper. Cet effet *hystérésis* empêchera la zone d'osciller entre deux résolutions.

Critères liés à l'octree restreint

Les listes précédemment établies vont être filtrées pour garantir que la structure d'octree restreint soit conservée.

Nous passons d'abord en revue les particules désirant se diviser. Ce n'est possible que si toutes leurs voisines ont un niveau égal ou inférieur car on aurait sinon un écart de deux niveaux entre les filles créées et une particule voisine. Si ce n'est pas le cas, la particule qui désirait se diviser est supprimée de la liste, mais les voisines qui empêchaient sa division y sont ajoutées, afin que la division puisse se faire au pas d'après (ou à un pas suivant si la division des voisines a elle aussi demandé la division d'autres particules).

[4] Elles sont en effet forcément divisées si elles sont inactives pour satisfaire le critère d'octree restreint.

De façon similaire, il est possible qu'un regroupement pose problème. Pour des raisons de stabilité, la priorité est donnée aux particules subdivisées, même si elles empêchent un regroupement, et on se contente dans ce cas de supprimer ce regroupement de la liste.

Critères liés au respect du temps-réel

Pour respecter le temps-réel et assurer une fréquence d'affichage fixe, on peut être amené à ajouter d'autres conditions sur la division des particules. On va pouvoir *limiter* le coût de calcul entre deux affichages (qui varie linéairement avec le nombre de particules et leur pas de temps et qui peut donc être mesuré) en interdisant une division trop importante du modèle. On pourra à nouveau diviser pour améliorer la précision locale des calculs lorsque des regroupements auront eu lieu ailleurs dans l'objet.

Inversement, on va *ralentir* volontairement la simulation lorsque celle-ci est stable et pourrait aller plus vite que le temps-réel du fait du faible nombre de particules utilisées. Une simple boucle d'attente basée sur l'heure donnée par la machine permettra d'attendre la synchronisation avec l'affichage suivant.

4.6 Mise à jour de la structure

Les voisines qui vont réellement servir à une particule (leur déplacement servant à calculer la force) vont être celles parmi ses 89 voisines qui sont *actives* à ce moment là. Elles ne peuvent être toutes actives en même temps (elles sont liées par des relation mère-filles). Le nombre de particules actives est au maximum de 56 (si la particule est entourée de particules divisées (Fig. 3.7a) et au minimum de 14 (si tout est regroupé autour d'elle : 7 voisines du niveau supérieur et ses 7 "sœurs" (Fig. 3.7 c).

Les 89 voisines sont établies au départ et ne changeront pas durant la simulation. On mettra par contre à jour une liste dynamique de celles parmi elles qui sont actives (i.e. réellement simulées).

Lorsqu'une particule se divise (resp. regroupe), des tables précalculées permettent de mettre à jour la liste de ses voisines actives, ainsi que les listes des voisines actives de ses voisines. Toutes ces relations peuvent en effet être précalculées en fonction des position relatives des deux anciennes voisines. La mise à jour des listes de voisines actives ne demande ainsi aucun calcul.

5 Multirésolution temporelle

Tout comme la méthode propose une structure de multirésolution à l'aide d'échantillonnages différents de l'espace, nous allons introduire ici la notion de multirésolution temporelle. Le changement du pas de temps global d'intégration avait déjà été utilisé dans la littérature [Jou96, Arn88], et celui du changement du pas de temps individuel l'avait été par Desbrun [Des97], dont nous reprenons le principe ici.

Un pas de temps d'intégration différent pour chaque particule va permettre d'optimiser les calculs. Il permet de garantir en chaque endroit une stabilité numérique de l'intégration, tout en évitant de simuler à la fréquence la plus haute l'ensemble de l'objet.

L'algorithme de simulation est donc un peu plus complexe. Ce qui est décrit ici n'est pas lié au caractère de multirésolution spatiale décrit précédemment et pourrait être appliqué avec la méthode à résolution fixe de la Section 3.

5.1 Critère de Courant

Le critère de Courant (voir annexe B) impose un pas de temps d'intégration compatible avec la vitesse du son à l'intérieur du matériau. Ceci impose pour une particule donnée d'avoir un pas de temps satisfaisant :

$$dt < h \sqrt{\frac{\rho_0}{\lambda + 2\,\mu}}$$

La vitesse du son associée à l'équation de Navier est $\sqrt{\frac{\rho_0}{\lambda+2\,\mu}}$. Le paramètre h représente la plus petite distance entre une particule et l'une de ses voisines.

Puisque les résolutions d'échantillonnage sont divisées par deux entre un niveau et le suivant, la distance d'une particule à ses voisines grandit également d'un facteur deux. Le pas de temps d'un niveau grossier pourra donc être le double de celui du niveau fin qui le succède. En plus d'avoir environ huit fois moins de particules,

celles-ci doivent donc être simulées deux fois moins souvent, ce qui donne un rapport 16 entre les temps de simulation de deux niveaux successifs, si on les suppose intégralement actifs.

5.2 Synchronisation et mise en œuvre

Pour cette raison de facteur deux entre les pas de temps des particules des différents maillages, ainsi que pour des raisons de synchronisation, on n'utilisera en pratique qu'un nombre fini de pas de temps possibles, ceux-ci étant des inverses de puissances de deux du pas de temps d'affichage :

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

On classe les particules dans des listes en fonction de leur pas de temps d'intégration, choisi parmi les dt_i. Entre deux affichages successifs de la simulation, on va simuler de nombreux pas d'intégration. On divise le temps en intervalles de la taille du dt le plus faible, dt_{min}. Au $k^{ème}$ pas de la simulation, on parcourt les listes de particules correspondant à des dt_i sous-multiples du temps $t = k\ dt_{min}$. Le niveau dt_{min-1} sera ainsi effectivement simulé une fois sur deux, dt_{min-2}, une fois sur quatre et ainsi de suite. En pratique, choisir quels sont les dt_i qui doivent être simulés à chaque pas se fait très rapidement grâce à des opérations binaires sur le nombre de pas d'intégration déjà réalisés.

Pour chaque particule de la liste parcourue, on va calculer une force en fonction des derniers déplacements des voisines actives et ensuite l'intégrer pour trouver la nouvelle position. Cette intégration se fait sur un pas de temps correspondant à celui de la particule.

On pourrait aussi imaginer que toutes les intégrations se déroulent à la fréquence de dt_{min}. Les *forces* appliquées à toutes les particules actives seraient toujours calculées à la fréquence de leur dt_i, mais leur *position* serait remise à jour à chaque pas (tous les dt_{min}), en intégrant la *dernière force calculée*.

On aurait ainsi à chaque instant des positions actualisées pour toutes les particules actives, ce qui rendrait le calcul des forces plus précis car basé sur des positions plus récentes. Si on ne calcule pas plus souvent les forces, ce qui est l'opération chère, mettre à jour à chaque pas de temps la position de *toutes* les particules actives entraîne néanmoins un surcoût non négligeable qui ne vaut peut-être pas la peine d'être entrepris.

5.3 Critère de stabilité d'intégration

Le pas de temps déterminé par le critère de Courant est le plus grand tolérable pour une particule donnée. Nous lui avons ajouté un autre critère assurant que le pas de temps est suffisamment petit pour prévenir les problèmes d'intégration numérique. Ceci est obtenu en faisant en sorte que dt satisfasse :

$$||\mathbf{a}\ dt|| = ||\mathbf{v}_t - \mathbf{v}_{t-1}|| < \Delta\mathbf{v}_{max}$$

Ainsi nous réduisons dt lors des soudains changements de vitesse, sources d'instabilités.
Chaque particule adopte à chaque instant le plus grand pas de temps satisfaisant ces deux critères.

6 Résultats

6.1 Premiers essais

Les résultats de cet algorithme multirésolution sont assez intéressants. La simulation est très stable : même si l'on excède les limites de déformations réalistes, la surface commençant à se replier sur elle-même, il suffit de reculer l'outil pour que l'objet revienne très rapidement à sa position d'équilibre. Ce genre de comportement n'existe pas avec les réseaux masses-ressorts qui, une fois qu'ils ont été trop déformés, divergent inexorablement.

L'adaptativité du modèle en fonction des déformations subies est assez intuitive, comme le montre la Figure 3.9. Les différentes résolutions sont toutes employées, les plus fines étant adoptées à proximité de l'outil que l'on manipule, les régions lointaines se contentant de peu de particules.

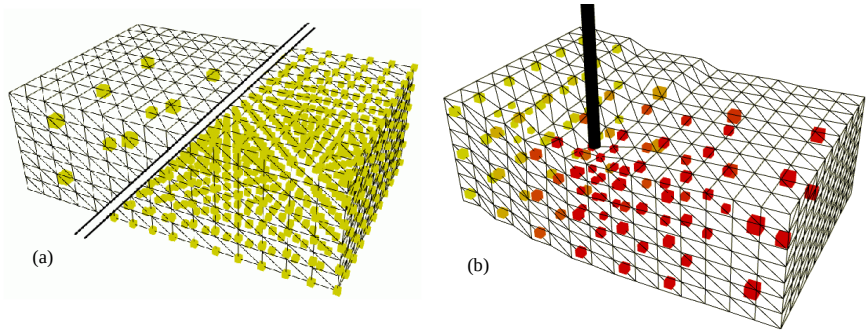


FIG. 3.9: Un parallélépipède déformé par un outil. Les résolutions comportent entre 24 et 1056 particules (a). La discrétisation qui a lieu au contact de l'outil durant la simulation est intuitive.

6.2 Cas d'école

Nous étudions maintenant l'animation d'une tige, dont une des faces est maintenue fixe, qui se plie sous l'effet de la gravité dans un milieu visqueux. La Figure 3.10 montre les états initiaux et finaux, avec des coefficients de Lamé de $\mu = 5000$ et $\lambda = 1000000$.

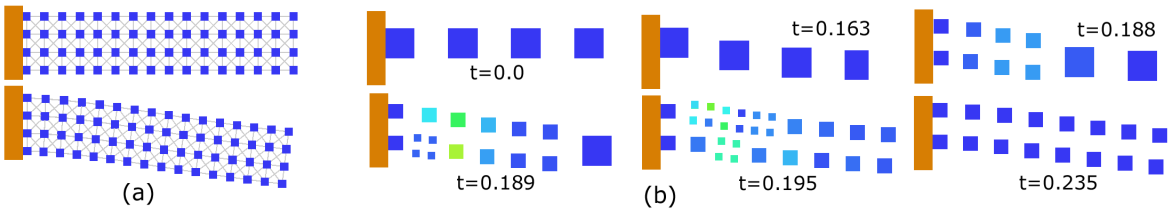


FIG. 3.10: Une tige oscillant sous la gravité. Simulation de référence faite avec 256 particules (a). Images de la simulation adaptative (b).

Le Tableau 6.2 montre le temps moyen de la simulation obtenu en utilisant différentes résolutions spatiales. Comme espéré, la simulation adaptative offre un bon compromis en donnant un temps de calcul proche de celui du niveau 3 (32 particules), tout en profitant des 256 particules potentielles du niveau 4. Le nombre de particules réellement simulées varie entre 4 (début) et 88, se stabilisant à 32 (niveau 3) quand la barre atteint sa position d'équilibre.

Niveau	2	3	4	adaptatif
Nb de particules	4	32	256	4-88
Temps de simulation (unités cpu)	0.87	4.29	38.90	5.27

6.3 Une application temps-réel

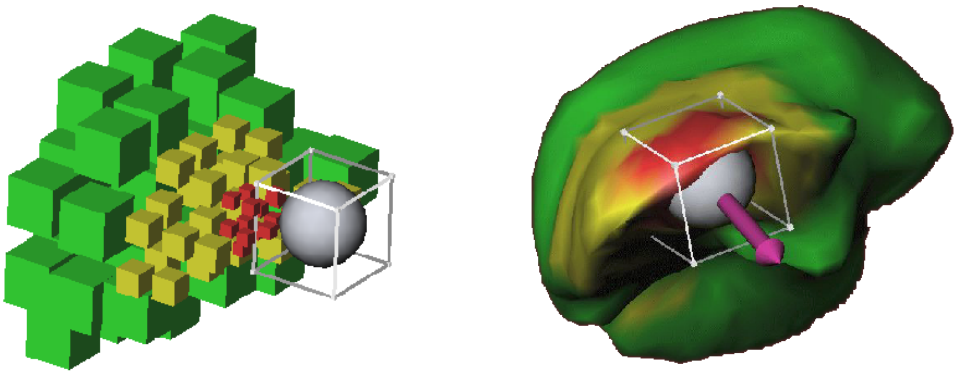
Nous avons réalisé avec cette technique un simulateur de chirurgie laparoscopique. Le foie virtuel, issu de données anatomiques, peut réagir en temps-réel aux actions de l'utilisateur, comme le montre la Figure 3.12.

Les coefficients rhéologiques λ et μ ont été choisis pour leur résultat visuel. Des données bio-médicales auraient dû être disponibles, mais les mesures faites ne sont pas suffisamment fiables car elle varient dans de grandes proportions en fonction de l'expérience et du caractère vivant ou mort de l'organe sur lequel elles sont faites.

Le caractère multirésolution de notre méthode est ici crucial car une simulation à discrétisation fixe utilisant seulement la résolution la plus fine aurait été trop lente d'un facteur dix par rapport au temps réel [5]. En combinant les résolutions, nous pouvons garantir que le temps-réel et la fréquence d'affichage seront respectés, celle-ci étant de 12 Hz dans notre cas sur une Silicon Graphics Onyx2 avec un processeur R10K. Le nombre de particules actives de la simulation varie entre 34 (résolution la plus grossière, utilisée lorsque le foie est au repos) et 130-140 qui est le nombre maximal de particules simulables en conservant le temps-réel avec cette raideur de matériau ($\mu = 4000$ et $\lambda = 5000$). La résolution des particules utilisées est intuitivement correcte :

[5] La simulation se faisant alors avec environ 770 particules.

élevée à proximité de l'outil et simplifiée à plus grande distance (voir Figure 3.11). Le pas de temps le plus faible utilisé était d'environ 0.002 sec (500 Hz) soit 32 pas d'intégration entre deux affichages.



a b

FIG. 3.11: Illustration de la répartition intuitive des particules lors d'une déformation. Le niveau des particules est représenté par la taille des cubes les symbolisant (a) ou par la couleur des nœuds de la surface auxquels elles sont reliées (b). La flèche représente la force renvoyée à l'utilisateur.

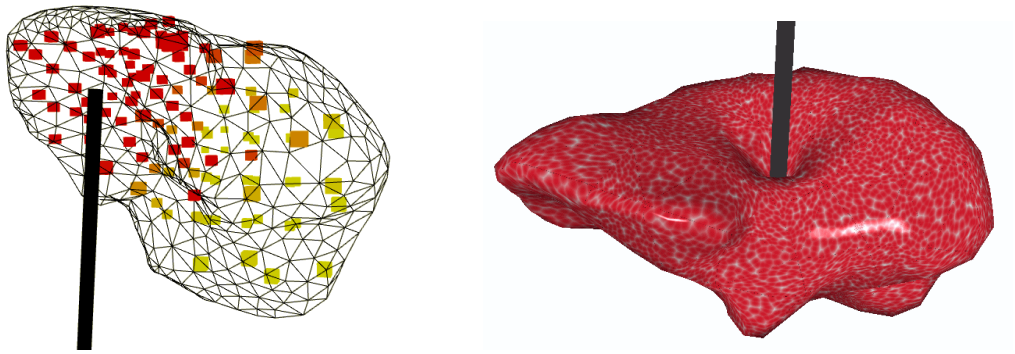


FIG. 3.12: Bien que les contraintes imposées dépassent parfois le formalisme des petites déformations, notre modèle présente néanmoins une réponse d'une bonne qualité visuelle à l'utilisateur. L'ajout de textures augmente sensiblement le réalisme du simulateur.

7 Discussion

7.1 Indépendance de la résolution

Les résultats obtenus sont assez convaincants et il est intéressant de noter que très peu de particules (quelques dizaines) suffisent souvent pour avoir un bon résultat.

Les simulations réalisées donnent un aspect un peu trop mou au foie et augmenter sa raideur se fait au détriment du nombre de particules que l'on peut simuler en conservant le temps-réel.

L'aspect multirésolution de l'animation est assez bien dissimulé à l'utilisateur (voir Chapitre 6), et même si les changements de niveau de détail internes peuvent se remarquer à la surface, le comportement global de l'objet reste cohérent.

Néanmoins des tests d'école réalisés sur l'exemple classique du cube collé à un mur et oscillant montrent que le comportement *dynamique* d'un objet où *se mélangent* les résolutions n'est pas le même que lorsque l'on simule les niveaux de détail séparément. La fréquence des oscillations est différente et le comportement global est beaucoup moins stable.

C'est un cas difficile où l'on ne mesure que le comportement dynamique et l'on peut comprendre que ces imperfections se remarquent moins sur l'exemple du foie où l'on se contente généralement d'appuyer et de maintenir la déformation, puis de relâcher.

Ces instabilités peuvent s'expliquer par le mauvais comportement des opérateurs différentiels lorsqu'on les applique sur des points trop éloignés d'une grille régulière (où ils rejoignaient les résultats des différences finies). Une particule entourée de voisines d'un niveau différent doit en effet utiliser des points d'échantillonnage assez mal répartis autour d'elle.

Il est possible de corriger ce comportement en passant d'une méthode adaptative à une méthode hiérarchique.

7.2 Une méthode hiérarchique

Puisque les particules semblent mal parvenir à prendre en compte leurs voisines issues d'une résolution différente, on va modifier l'algorithme pour que seules les voisines actives *du même niveau* soient prises en compte.

La modification principale réside dans le fait que lorsqu'une particule se divise, elle *reste* active (voir Fig. 3.13).

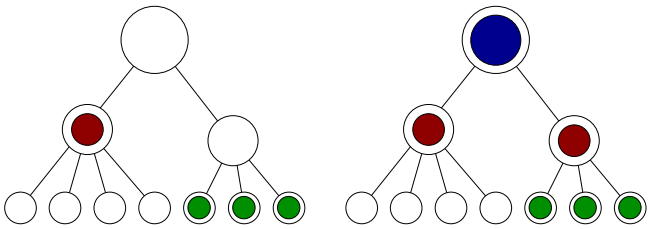


FIG. 3.13: L'adaptatif (a) ne simule que le niveau le plus fin choisi alors qu'en hiérarchique (b), les niveaux inférieurs continuent à être actifs et cohabitent.

Une particule divisée va continuer à calculer une force basée sur le déplacement de ses voisines, mais en ne considérant maintenant plus que celles *actives et non divisées*, du même niveau qu'elle. Elle ne prendra donc plus en compte que les influences des particules actives du même niveau qu'elle. La cohabitation entre les niveaux s'en trouve modifiée.

Descente des forces

La force calculée par la particule *divisée* ne va par contre pas être classiquement intégrée mais *transmise* aux filles. Celles-ci vont ainsi recevoir de la part de leur mère une force qui exprimera les interactions ayant eu lieu entre la zone qu'elles occupent et les zones actives voisines simulées à une résolution plus grossière.

Cette descente va se faire à chaque mise-à-jour de la force et va se propager à toute la descendance *active* de la particule (filles, petites filles, etc...). Toutes les particules actives de cette zone vont ainsi prendre en compte les contraintes générées par la mère et ses voisines de même niveau. Les interactions ayant eu lieu à l'intérieur de ce maillage entre cette zone et les zones voisines sont ainsi propagées aux filles des niveaux inférieurs.

Remontée des positions

Inversement, les mères vont pouvoir prendre en compte les interactions de leurs filles avec les niveaux *inférieurs* grâce à une mise à jour directe de leur position. Les particules divisées déduisent en effet leur position de celle de leur fille par barycentre (Eq. 3.10), reflétant ainsi ce qui a pu se passer dans cette zone aux niveaux inférieurs.

La particule *divisée* ne sert ainsi que en quelque sorte que d'indicateur grossier de ce que ses filles (ou petites-filles si celles-ci sont aussi divisées, etc...) ont simulé finement dans cette zone.

Sa position ainsi actualisée sert à calculer les forces intervenant avec les voisines non divisées pour pouvoir la communiquer à ses filles. On crée ainsi avec cette descente de forces et cette remontée de positions la liaison entre les maillages indispensable à tout processus multirésolution.

Algorithme

L'animation d'un niveau de résolution donné se passe ainsi :
- Pour les particules *divisées* seulement, remettre à jour la position en remontant récursivement celle de sa descendance active par barycentre (en descendant jusqu'aux particules actives non divisées).
- Calculer la force engendrée par les déplacements des voisines de même niveau, actives et non divisées grâce aux opérateurs différentiels.
- Pour les particules divisées, transmettre la force ainsi calculée à toute leur descendance active, qui la stockera.
- Pour les particules *non* divisées, ajouter à cette force celles reçues de tout le reste l'ascendance (mère, grand-mère, etc...) et intégrer la force totale pour déterminer la nouvelle position.

L'algorithme est un peu plus complexe si on autorise la multirésolution temporelle décrite en Section 5. On ne remet plus alors à jour les particules niveau par niveau, mais selon le pas de temps de chacune. Le principe reste le même : remontée de position, calcul et descente de force pour les particules divisées et calcul et intégration de la force cumulée pour les particules actives non divisées.

Avec un tel algorithme, on n'a plus d'interaction (*i.e.* de calcul de force en fonction des déplacements relatifs) qu'entre des particules de même niveau. On va ainsi limiter l'imprécision des opérateurs précédemment décrits qui nuisait aux résultats multirésolution.

Mesure du surcoût du hiérarchique

La contrepartie est une augmentation du nombre de particules actives simulées, et donc du coût de l'animation. Cette augmentation reste néanmoins très limitée. Supposons par exemple un matériau composé de 4 maillages, composés de respectivement 4, 32, 256 et 2048 particules. Prenons, pour simplifier, le cas où à un instant donné, la moitié des particules de chaque niveau est subdivisée (subdivision régulière).

En conservant les mères comme actives dans la méthode hiérarchique qui vient d'être proposée, on doit calculer la force subie par $4 + \frac{1}{2}32 + \frac{1}{2^2}256 + \frac{1}{2^3}2048 = 4 + 16 + 64 + 256 = 340$ particules. L'algorithme adaptatif n'aura pas à prendre en compte les particules divisées, ce qui donnera $\frac{1}{2}4 + \frac{1}{2}16 + \frac{1}{2}64 + 256 = 298$ particules actives.

On aura donc dans cet exemple une augmentation de 14% du nombre de particules actives et donc approximativement du temps de calcul (celui-ci étant principalement constitué du calcul de la force et de l'intégration des positions des particules actives) entre la version hiérarchique et celle adaptative de la méthode.

7.3 Problème avec le grad(div)

Nous n'avons pas poussé les tests avec cette nouvelle méthode, que nous avons décrite ici pour l'intérêt de son principe. En effet, nous nous sommes aperçus que l'opérateur servant à calculer le $\mathbf{grad}(\mathrm{div}\ \mathbf{u})$ donnait des résultats erronés.

Ceci s'est révélé lors de tests réalisés en 2D en étudiant les résultats des opérateurs lorsqu'on les appliquait à des champs *analytiques*. Les deux opérateurs s'expriment en 2D comme :

$$\Delta\mathbf{u} = \begin{cases} u_{x,xx} + u_{x,yy} \\ u_{y,xx} + u_{y,yy} \end{cases} \qquad \mathbf{grad}(\mathrm{div}\ \mathbf{u}) = \begin{cases} u_{x,xx} + u_{y,xy} \\ u_{x,xy} + u_{y,yy} \end{cases}$$

Notons tout d'abord que ces opérateurs sont totalement indépendants de la direction dans laquelle se trouvent les voisins, à partir du moment où ceux-ci sont sur une grille régulière, qui n'a donc pas besoin d'être alignée avec les axes. Ce résultat était prévisible dans la mesure où ces opérateurs ne font intervenir que les distances entre les points et non leur position absolue.

Si l'opérateur laplacien donne bien les résultats escomptés, l'opérateur **grad**(div **u**) tel que nous l'utilisons calcule en fait **grad**(div **u**) $= \begin{cases} u_{x,xx} \\ u_{y,yy} \end{cases}$, les dérivées croisées n'apparaissant pas.

Si au lieu de considérer la projection le long des directions radiales comme on le fait, on prend les directions tangentes, on crée un opérateur calculant **grad**(div **u**) $= \begin{cases} u_{x,yy} \\ u_{y,xx} \end{cases}$, ce qui est normal puisque la somme de ces deux composantes doit donner le laplacien.

L'assimilation faite dans la Section 2 entre le **grad**(div **u**) et la partie non rotationnelle du champ était donc trop simplificatrice. La version publiée de ce chapitre [DDBC99, DC99a] ne mentionnait pas ce problème, apparu lors de tests postérieurs. On a également pu mesurer sur cet exemple analytique que les résultats se dégradaient assez vite lorsqu'on s'éloignait de la grille régulière.

Le problème plus profond de cet opérateur est qu'il est incapable de calculer une *dérivée croisée*. On a beau modifier l'axe selon lequel on projette, elles ne peuvent apparaître avec cette méthode. Les approximations faites sont trop arbitraires et l'on ne voit pas où l'on pourrait retrouver les termes de dérivée croisée. D'autres pistes de recherche menées en parallèle semblant promettre de meilleurs résultats, nous avons préféré les privilégier pour obtenir une nouvelle expression des opérateurs.

8 Conclusion

Un calcul des opérateurs intervenant dans l'équation de Navier a été proposé. Donnant de bons résultats indépendants de la résolution, ils ont pu être utilisés dans une méthode multirésolution adaptative utilisant un octree.

Les résultats visuels sont assez convaincants et un premier modèle de simulateur chirurgical a été mis au point. Il permet de simuler le mouvement d'une centaine de points en temps-réel, la résolution s'adaptant automatiquement lors de la simulation.

Les tests plus formels que nous avons effectués sur ces opérateurs ont révélé des insuffisances numériques. Cette médiocre qualité nous a conduit à chercher une autre façon de calculer ces différentielles, en partant de considérations géométriques et mathématiques plus poussées, comme nous allons le décrire dans le prochain chapitre.