



**HAL**  
open science

## Animation multirésolution interactive d'objets déformable

Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani

► **To cite this version:**

Gilles Debunne, Mathieu Desbrun, Marie-Paule Cani. Animation multirésolution interactive d'objets déformable. Journées AFIG, 1999, Reims, France. 10 p. inria-00510067

**HAL Id: inria-00510067**

**<https://inria.hal.science/inria-00510067>**

Submitted on 19 Aug 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

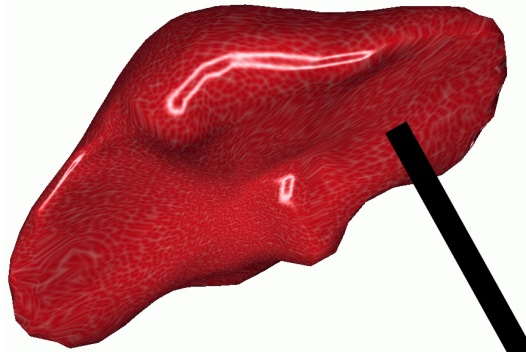
# Animation multirésolution interactive d'objets déformables

Gilles Debunne<sup>†</sup> Mathieu Desbrun<sup>‡</sup> Marie-Paule Cani<sup>†</sup>

<sup>†</sup>iMAGIS<sup>1</sup> <sup>‡</sup>Caltech

{Gilles.Debunne|Marie-Paule.Cani}@imag.fr

**Résumé :** C et article présente une approche pour animer des matériaux élastiques déformables en temps interactif en utilisant une résolution adaptative en temps et en espace. Nous proposons un nouveau modèle algorithmique, basé sur l'élasticité linéaire qui comprend le calcul d'opérateurs différentiels discrets sur une grille irrégulière. Ce modèle autorise un raffinement ou une simplification de l'échantillonnage en fonction d'un critère local d'erreur. Le résultat est une réduction des calculs tout en garantissant un comportement réaliste et indépendant de la résolution à un seuil d'erreur fixé près. Nous validons cette technique par une application de simulateur médical temps-réel.



## 1 Introduction

Bien que des techniques interactives d'animation existent et sont utilisées en réalité virtuelle par exemple, elles simulent principalement des objets rigides. En utilisant des lois simplifiées de la physique du solide, elles se concernent principalement la détection de collision et la modélisation des contacts, où les approches naïves sont trop chères en calcul [Bar96, Fau98]. Quelques travaux portent sur la simulation des objets déformables. La majorité des techniques existantes sont calculées hors-line et ne peuvent être utilisées dans le cadre d'un environnement virtuel et du temps-réel.

Néanmoins, des travaux récents montrent de possibles très faibles coûts de calcul sur des objets ou des surfaces déformables [BW98, DSB99]. Ces approches utilisent l'intégration implicite pour intégrer temporellement avec peu ou pas de soucis liés à la stabilité, même pour de grands pas de temps.

Les techniques d'animation actuelles utilisent une résolution spatiale constante (un nombre fixe de points). On doit pouvoir être plus efficace en adaptant la discrétisation en fonction de la déformation. Un objet soumis à une forte déformation locale doit pouvoir être raffiné dans cette région, pour assurer à la fois une précise description géométrique et une précision donnée. De nombreux calculs peuvent être évités en utilisant une telle technique adaptative, comme c'est maintenant largement fait dans le domaine de la radiosité en simulation d'éclairage. Malheureusement, assurer un comportement global similaire quelle que soit la discrétisation reste un problème ouvert.

### 1.1 Travaux précédents

Le premier modèle en image de synthèse pour animer des objets déformables fût présenté par Terzopoulos *et al.* [TPBF87], et utilisait des différences finies ou des éléments finis pour intégrer une équation de Lagrange basée sur l'énergie. Ce modèle initial, basé sur la loi de Hooke des objets parfaitement élastique a été notablement amélioré pour prendre en compte la plasticité et les fractures [TW88, TF88]. Les techniques d'éléments finis ont aussi été proposées [GMTT89], avec des applications en simulation temps réel [BNC96, DCA99], mais avec un modèle quasi-statique, perdant ainsi l'aspect dynamique de la simulation. Les modèles physiques demandant de gros calculs, d'autres approches sont apparues, permettant l'animation rapide de la dynamique simple en ne prenant en compte seulement quelques

<sup>1</sup>iMAGIS-GRAVIR/IMAG est un projet commun CNRS,INRIA,UJF,INPG. Adresse postale : INRIA Rhône-Alpes, 655 avenue de l'Europe, 38330 Montbonnot Saint-Martin

déformations ou modes vibratoires [PW89, WW90, MT92]. Malheureusement de telles restrictions sur le comportement affectent sensiblement le réalisme de la simulation.

Comme dit précédemment, toutes ces techniques utilisent une discrétisation spatiale fixe, et généralement un pas de temps également fixé. Récemment, un modèle utilisant une résolution adaptative a été développé pour la simulation de tissus [HPH96].

Le modèle masse-ressort modélisant un morceau de tissus se raffine localement dès que deux ressorts adjacents forment un angle supérieur à un seuil donné. Cette idée permet au modèle de converger vers un équilibre statique plus rapidement en limitant le nombre de masses utilisées dans la simulation. Malheureusement, un modèle aussi simple ne saurait garantir un comportement identique durant la simulation. De plus, la masse totale de l'objet augmente au cours de la simulation, ce qui pose problème si on doit ensuite tirer sur le tissus par exemple.

Un autre modèle, introduit pour des matériaux très déformables comme la boue ou la pâte propose une résolution adaptative en temps et en espace basé sur un modèle à base de SPH (Smooth Hydrodynamics particles) [DCG96, GCD<sup>+</sup>98]. Cette fois-ci une équation d'état représentant le comportement (la raideur par exemple) est définie par l'utilisateur. Les particules discrétisant la matière se divisent et se regroupent en fonction d'un critère local d'énergie. Cette méthode ne s'applique pas aux matériaux plus rigides. Nous gardons néanmoins ce même type d'approche dans cet article en simulant à différentes résolutions une même équation.

## 1.2 Approche

Dans cet article, nous gagnons en temps de calcul en optimisant le nombre de points d'échantillonnage nécessaires. En utilisant un modèle général d'élasticité, nous dérivons une équation aux dérivées partielles garantissant un même comportement global dans le chapitre 2. Nous proposons ensuite en 3 des opérateurs différentiels simples qui permettent d'intégrer l'équation précédente même sur des grilles modérément déformées. Une fois le seuil d'erreur fixé, nous prouvons au chapitre 4 qu'une intégration adaptative espace-temps utilisant des raffinements/simplifications est facile à mettre en place. Nous donnons quelques détails d'implémentation en section 5, et terminons par une démonstration de simulateur chirurgical hépatique basé sur notre méthode en section 6, et une conclusion en section 7.

## 2 Un modèle physique général

Notre modèle est globalement le même que celui de [TPBF87]. Néanmoins, l'aspect mathématique est différent. Cette section expose la physique simple qu'utilise notre modèle, des détails pouvant être trouvés dans [TG70] par exemple.

### 2.1 Notations

Nous utilisons une notation légèrement non standard pour plus de simplicité. Les vecteurs seront en gras :  $\mathbf{u} = (u_x \ u_y \ u_z)^T$ , les matrices en lettres calligraphiques :  $A = \mathbf{u} \mathbf{u}^T$ . Nous utiliserons également des notations compactes pour les dérivées. Par exemple,  $u_{x,y} = \partial u_x / \partial y$ , ou  $u_{z,xx} = \partial^2 u_z / \partial x^2$ .

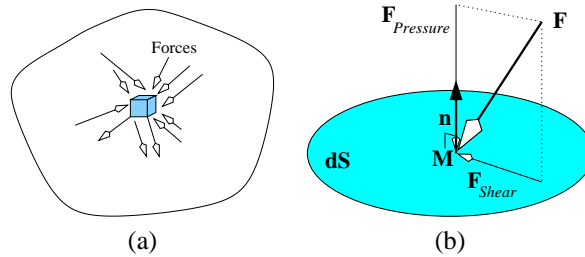


FIG. 1: (a) Forces autour d'un volume (b) Forces agissant sur un élément de surface donné  $dS$ , centré autour d'un point  $M$  et défini par sa normale  $\mathbf{n}$ .

### 2.2 Tenseur des contraintes

Un petit élément de matière  $M$  reçoit des forces de toutes les directions (Figure 1(a)). Une façon de décrire ces forces locales est d'évaluer la force surfacique (la contrainte) agissant sur un élément de surface donné centré en  $M$  et de normale  $\mathbf{n}$ . Cette force  $\mathbf{F}$  aura une composante le long de  $\mathbf{n}$ , analogue à une pression, et une composante orthogonale, créant un cisaillement (Figure 1(b)). Le *tenseur des contraintes* décrit alors l'application *linéaire* entre toutes les normales et leur contraintes associées.

Cette matrice  $3 \times 3$  symétrique, généralement notée  $\sigma$ , donne la force de contrainte  $\mathbf{F}$  sur un élément de surface de normale  $\mathbf{n}$  :

$$\sigma \mathbf{n} = \mathbf{F}.$$

De ce tenseur, nous déduisons la *force résultante par unité de volume* agissant sur un élément de matière comme étant la divergence de  $\sigma$ <sup>1</sup>. Si  $\rho$  est la densité volumique, de l'élément considéré,  $\mathbf{g}$  la gravité, et  $\mathbf{a}$  l'accélération de cet élément, nous pouvons utiliser le principe fondamental de la mécanique pour écrire :

$$\rho \mathbf{a} = \text{Div } \sigma + \rho \mathbf{g}. \quad (1)$$

Pour calculer le tenseur des contraintes, il nous faut connaître la déformation subie par le matériaux pour en déduire la force locale. Nous introduisons pour celà le tenseur des déformations.

### 2.3 Tenseur des déformations

Nous appelons  $\mathbf{d}$  le *déplacement* d'un élément de matière depuis sa position initiale. Ceci définit un champ de vecteur à l'intérieur de l'objet. Une rotation pure de l'objet créera un champ constant, alors qu'une déformation complexe peut créer un champ arbitraire comme montré figure 2.

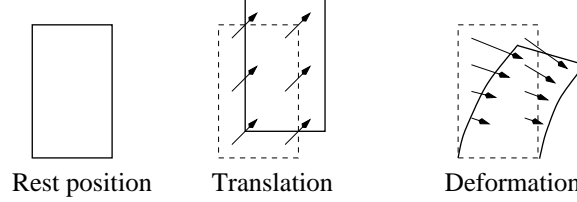


FIG. 2: Un objet et des champs de déplacement possibles définissant la forme courante.

Par définition, le gradient de ce champ de vecteur est :

$$A = \overline{\text{grad}(\mathbf{d})} = \begin{pmatrix} d_{x,x} & d_{x,y} & d_{x,z} \\ d_{y,x} & d_{y,y} & d_{y,z} \\ d_{z,x} & d_{z,y} & d_{z,z} \end{pmatrix} \quad (2)$$

La partie antisymétrique de cette matrice représente seulement la partie rotationnelle du déplacement, alors que la partie symétrique, appelée *tenseur du taux de déformation* et noté  $\varepsilon$ , exprime le taux intrinsèque de déformation agissant sur un élément de matière :

$$\varepsilon = \frac{1}{2}(A + A^T) = \frac{1}{2} \begin{pmatrix} 2d_{x,x} & d_{x,y} + d_{y,x} & d_{x,z} + d_{z,x} \\ d_{x,y} + d_{y,x} & 2d_{y,y} & d_{y,z} + d_{z,y} \\ d_{x,z} + d_{z,x} & d_{y,z} + d_{z,y} & 2d_{z,z} \end{pmatrix} \quad (3)$$

Comme pour le tenseur des contraintes, nous insistons sur le fait que ce tenseur n'est qu'une approximation au premier ordre et ne représente qu'une approximation linéaire des déformations.

### 2.4 Loi de déformation

Un modèle physique définit la façon dont un objet se déforme en fonctions des forces appliquées, et vice-versa. Nous avons donc à décrire une relation entre le tenseur des contraintes et le tenseur des déformations. Nous avons choisi la loi de Hooke car c'est l'une des plus simple, et elle décrit néanmoins suffisamment précisément une grande gamme de matériaux. Cette loi s'exprime par ( $I_3$  étant la matrice  $3 \times 3$  identité) :

$$\sigma = 2\mu\varepsilon + \lambda \text{trace}(\varepsilon) I_3 \quad (4)$$

Nous pouvons maintenant déduire des forces des déplacements, ou des déplacements des forces.

### 2.5 Équation de Lamé

En utilisant la loi de déformation précédente, l'équation globale du mouvement peut se réécrire (sans la gravité par simplicité) en (see [TG70]) :

$$\rho \mathbf{a} = \mu \Delta \mathbf{d} + (\lambda + \mu) \nabla(\text{div } \mathbf{d}) \quad (5)$$

en réécrivant la loi de Hooke dans l'équation fondamentale du mouvement  $\rho \mathbf{a} = \text{Div } \sigma$ . Cette formulation, due à Lamé, contient la relation contrainte/déformation dans une équation aux dérivées partielles qui donne une autre interprétation de la loi de Hooke. Un tel modèle est en effet la composition d'une propagation d'onde et d'une contrainte de préservation du volume. Puisque l'accélération est la dérivée seconde par rapport au temps du déplacement  $\mathbf{d}$  et que  $\Delta \mathbf{d}$  est la somme des dérivées secondes en espace, la première partie de l'équation  $\rho \mathbf{a} = \mu \Delta \mathbf{d}$  est en fait une équation hyperbolique, aussi appelée *équation d'onde*. La vitesse de propagation de cette onde est dans ce cas  $c = \sqrt{\mu/\rho}$ . L'autre partie,  $\rho \mathbf{a} = (\lambda + \mu) \nabla(\text{div } \mathbf{d})$ , représente un terme de préservation du volume.  $\text{div } \mathbf{d}$  est l'*expansion volumique*, suivant le gradient de l'expansion volumique et tendant à rétablir le volume initial. En fonction des valeurs de  $\lambda$  and  $\mu$ , on peut interpréter la loi de Hooke comme une onde de déformation avec plus ou moins de compressibilité.<sup>2</sup> Cette interprétation va nous aider à concevoir notre simulation numérique.

<sup>1</sup>L'opérateur divergence d'une matrice,  $\text{Div}$ , est le vecteur formé de la divergence  $\text{div}$  de chaque ligne, avec  $\text{div}(\mathbf{u}) = u_{x,x} + u_{y,y} + u_{z,z}$ .

<sup>2</sup>Bien qu'on ne puisse l'atteindre parfaitement avec cette formulation, la préservation du volume de l'objet est habituellement considérée comme bonne lorsque  $\lambda > 100\mu$ , ce que nous utilisons dans nos exemples.

### 3 Simulation à une résolution fixe

Dans cette section, nous présentons notre implémentation du modèle physique précédemment décrit pour une résolution donnée.

#### 3.1 Principe

Nous allons utiliser l'équation de Lamé (Equ. (5)), et calculer les opérateurs différentiels qui la composent en essayant de ne pas faire d'hypothèses sur la discrétisation de l'objet. L'animation d'un objet demandera alors une discrétisation, suivie d'une intégration des équations aux dérivées partielles (EDP) sur les points et dans le temps.

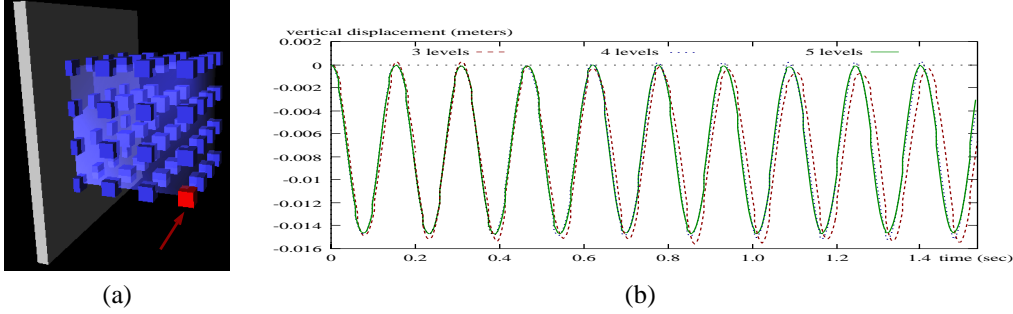


FIG. 3: Un cube de 10cm de côté oscillant avec la gravité, une de ses faces étant fixée. Nous mesurons le déplacement vertical d'un de ses coins (flèche) à différentes résolutions spatiales. Les niveaux sont composés de 64, 512 et 4096 particules. notez qu'aucun frottement n'a été utilisé dans cette simulation.

#### 3.2 Approximation discrete des opérateurs

Afin d'intégrer l'équation Equ. (5), il nous faut simplement approximer deux opérateurs : le Laplacien du champ de déplacement,  $\Delta \mathbf{d}$ , et le gradient du divergence de ce même champ :  $\nabla(\text{div } \mathbf{d})$ . Les différences finies ne s'appliquent qu'aux grilles régulières et ne s'appliquent pas ici. Ce calcul est difficile et a fait l'objet de nombreuses recherches en mathématiques et en physique.

##### L'opérateur Laplacien

Dans sa thèse, Milne a montré combien l'approximation d'une dérivée seconde en 1D était sensible, avec des problèmes de bruit dès lors que les points voisins ne sont plus centrés. Heureusement l'extension des différences finies proposée par Fornberg [For88] résoud le problème du bruit. Elle consiste à faire passer une fonction quadratique entre le point et ses deux plus proches voisins. Avec trois points d'échantillonnage d'une fonction  $f$ , situés à respectivement  $\Delta$  and  $\delta$  du point central, nous obtenons :

$$f_i'' = \frac{2}{\delta + \Delta} \left( \frac{f_{i-1} - f_i}{\delta} + \frac{f_{i+1} - f_i}{\Delta} \right)$$

Une généralisation directe de cette formule en 3D donne un opérateur indépendant de l'échelle [DMSB99, Fuj95] (avec  $l_{ij} = \|\mathbf{l}_{ij}\| = \|\mathbf{x}_i - \mathbf{x}_j\|$ , distance entre les points  $i$  et  $j$ ) :

$$\Delta \mathbf{d}_i = \frac{2}{\sum_j l_{ij}} \sum_{\text{neighbors } j} \frac{\mathbf{d}_j - \mathbf{d}_i}{l_{ij}} \quad (6)$$

Cette formulation simple, récemment introduite en image de synthèse pour le lissage des maillages nous donne une bonne approximation du Laplacien, ne dépendant que peu de la distribution des voisins. Les résultats qualitatifs sont en section 3.3.

##### L'opérateur gradient du divergence

Il nous faut maintenant déduire de façon cohérente un opérateur gradient du divergence de celui, robuste, du Laplacien.

En développant les dérivées secondes du Laplacien, nous trouvons :

$$\Delta \mathbf{d} = \nabla(\text{div } \mathbf{d}) - \nabla \times (\nabla \times \mathbf{d}) \quad (7)$$

où  $\times$  est le produit vectoriel.

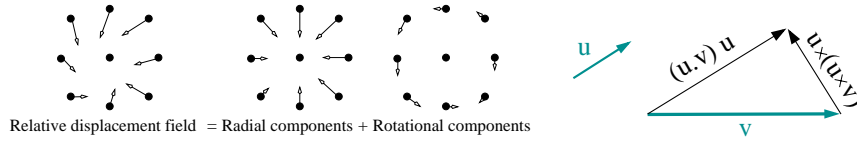


FIG. 4: Le champ de déplacement peut être décomposé en deux composantes : la radiale, créée par les forces de pression, et la rotationnelle, créée par les forces de cisaillement.

Nous savons que le divergence de  $\mathbf{d}$  est une mesure de l'expansion du volume comme dit en section 2.5. Une contrainte de cisaillement ne change pas le volume affecté par la seule composante radiale. Aussi proposons-nous de décomposer le Laplacien en ses deux composantes radiales et rotationnelles. Sachant qu'un vecteur  $\mathbf{v}$  peut s'écrire en utilisant un vecteur unitaire  $\mathbf{u}$ , en un vecteur aligné avec  $\mathbf{u}$  et un vecteur orthogonal à  $\mathbf{u}$  grâce à la relation :  $\mathbf{v} = (\mathbf{v} \cdot \mathbf{u})\mathbf{u} - \mathbf{u} \times (\mathbf{u} \times \mathbf{v})$  with  $\mathbf{u} = \mathbf{l}_{ij}/l_{ij}$  (voir Fig. 4), nous pouvons réécrire l'équation (6) en :

$$\begin{aligned} \Delta \mathbf{d}_i &= \frac{2}{\sum_j l_{ij}} \sum_{j \text{ neighbors}} \frac{\mathbf{d}_j - \mathbf{d}_i}{l_{ij}} \\ &= \frac{2}{\sum_j l_{ij}} \sum_{j \text{ voisins}} \frac{[(\mathbf{d}_j - \mathbf{d}_i) \cdot \frac{\mathbf{l}_{ij}}{l_{ij}}] \frac{\mathbf{l}_{ij}}{l_{ij}} - \frac{\mathbf{l}_{ij}}{l_{ij}} \times [\frac{\mathbf{l}_{ij}}{l_{ij}} \times (\mathbf{d}_j - \mathbf{d}_i)]}{l_{ij}} \\ &= \frac{2}{\sum_j l_{ij}} \sum_{j \text{ voisins}} \frac{[(\mathbf{d}_j - \mathbf{d}_i) \cdot \frac{\mathbf{l}_{ij}}{l_{ij}}] \frac{\mathbf{l}_{ij}}{l_{ij}}}{l_{ij}} - \frac{2}{\sum_j l_{ij}} \sum_{j \text{ voisins}} \frac{\frac{\mathbf{l}_{ij}}{l_{ij}} \times (\frac{\mathbf{l}_{ij}}{l_{ij}} \times (\mathbf{d}_j - \mathbf{d}_i))}{l_{ij}} \end{aligned} \quad (8)$$

En comparant alors (7) et (8), nous choisissons d'identifier la composante radiale comme étant le gradient du divergent :

$$\nabla(\text{div } \mathbf{d})_i = \frac{2}{\sum_j l_{ij}} \sum_{j \text{ voisins}} \frac{[(\mathbf{d}_j - \mathbf{d}_i) \cdot \frac{\mathbf{l}_{ij}}{l_{ij}}] \frac{\mathbf{l}_{ij}}{l_{ij}}}{l_{ij}} \quad (9)$$

Nous verrons dans la section suivante que ces deux opérateurs donnent de bons résultats en pratique.

### 3.3 Algorithme and validité

L'algorithme d'animation de notre modèle déformables est simple : à chaque pas de temps :

- Pour chaque particule, évaluer les forces internes en utilisant les opérateurs du Laplacien et du gradient du divergence,
- On en déduit une accélération avec Equ. 5,
- On intègre l'accélération sur le pas de temps  $dt$  en mettant à jour positions et vitesses.

Les opérateurs Laplacien et du gradient du divergence se comportent très bien à différentes résolutions. La figure 3 démontre qu'à trois différentes résolutions un objet subit les mêmes déformations dans le temps. Ceci valide nos opérateurs et nous autorise à passer à une discrétisation adaptative.

## 4 Simulation adaptative et temps et en espace

Le taux d'échantillonnage de la matière est généralement réglé par l'utilisateur et choisi *a priori*. Le nombre de points affectant souvent le résultat de l'animation, le réglage se fait à l'aide d'essais successifs. De plus, si un choc se produit durant l'animation, le pas de temps de toute la simulation (resp. le nombre de points) doit être pris suffisamment petit (resp. grand) pour éviter les instabilités numériques ce qui ralentit considérablement la simulation.

De nombreuses techniques utilisent un pas de temps adaptatif, mais nous proposons ici d'optimiser *automatiquement* le taux d'échantillonnage de notre modèle à la fois en temps et en espace. Les calculs sont alors concentrés où et quand c'est nécessaire. Cette section présente cette technique qui autorise un réglage du compromis précision/rapidité.

### 4.1 Adaptivité spatiale

Le modèle décrit dans la section précédente assurant un même comportement à différentes résolutions spatiales, nous pouvons l'adapter durant la simulation. Si un outil est utilisé sur l'objet, nous augmenterons l'échantillonnage dans cette zone pour obtenir une meilleure précision numérique et géométrique. Si l'outil est déplacé ailleurs, cette zone se simplifiera d'elle-même. Deux critères, contrôlant les seuils de raffinement et de simplification doivent être définis. Le premier assure la précision alors que le second permet d'optimiser l'efficacité des calculs.

## Critère d'adaptivité

Le modèle dépendant des dérivées secondes, ce sont elles qui indiqueront l'adéquation de l'échantillonnage. Lorsque la "fréquence" locale est trop élevée, l'approximation linéaire de nos opérateurs ne suffit plus. Ceci peut être testé par l'opérateur Laplacien, qui mesure la non linéarité de la fonction, grâce à la relation :

$$h^2 \|\Delta \mathbf{d}\| > \varepsilon_{max} \quad (10)$$

où  $h$  représente la plus petite distance entre cette particule et ses voisines. Nous avons trouvé cette grossière mais rapide approximation suffisante en pratique. Le raffinement apparaît où et quand nous l'espérons intuitivement.

De façon similaire, le critère de simplification indiquant quand des particules peuvent se regrouper est :

$$h^2 \|\Delta \mathbf{d}\| < \varepsilon_{min} \quad (11)$$

Une fois encore, les résultats montrent bien des simplifications dans les zones 'calmes' de l'objet.

## 4.2 Adaptivité en temps

Une fois la discrétisation spatiale choisie, le pas de temps doit être choisi en fonction pour éviter les instabilités. Trop grand la simulation ne serait pas stable ; trop petit, elle demande trop de temps. Nous voulons adapter le pas de temps localement, afin que les régions calmes n'aient pas à se mettre à jour aussi souvent que celles à mouvement rapide.

### Choix du pas de temps

La condition de Courant [DCG96] de l'équation de Lamé est :

$$dt < h \sqrt{\frac{\rho_0}{\lambda + 2\mu}} \quad (12)$$

où  $h$  représente encore la plus petite distance entre une particule et une de ses voisines, et  $\rho_0$  est la masse volumique du matériau. Ce pas de temps est le plus grand tolérable pour cette particule, mais nous assurons aussi qu'il est suffisamment petit pour prendre en compte les changements soudains. Ceci peut par exemple être fait en faisant en sorte que  $dt$  satisfasse :

$$\|\mathbf{a} dt\| = \|\Delta \mathbf{v}\| < \Delta \mathbf{v}_{max}$$

Ainsi nous n'aurons pas de soudain changement de vitesse, source d'instabilités.

Nous choisirons à chaque instant le plus grand pas de temps satisfaisant ces deux critères.

## 4.3 Validation

Ces critères donnent un comportement stable du modèle. Comme montré figure 5, un objet à une résolution grossière se divisera dans les régions déformées où l'utilisateur intervient.

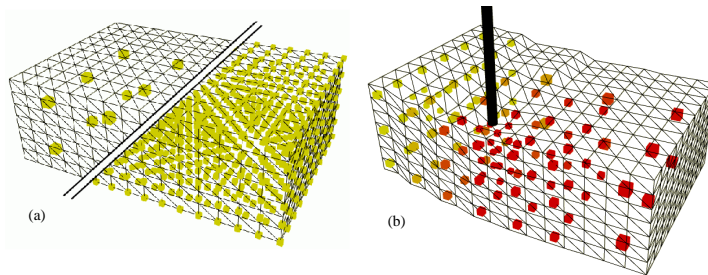


FIG. 5: Un parallélépipède déformé par un outil. La figure (a) montre les résolutions les plus grossières (24 particules) et les plus raffinées (1056 particules). La figure (b) montre la discrétisation intuitive qui a lieu durant la simulation.

## 5 Implementation

Nous présentons ici la structure de données qui permet de bien tirer de notre modèle multirésolution.

## 5.1 Structure de donnée spatiale

Cette implementation n'est pas aussi générale que le modèle décrit dans la précédente section. Il est optimisé pour les faibles déformations pour gagner en efficacité. Nous nous sommes limité aux matériaux mous, tels que les organes humains. En supposant que durant la simulation le matériau ne se déforme pas trop, sa topologie va rester la même. En pratique, cela signifie que le voisinage d'un point donné restera le même et pourra donc être précalculé. Nous allons donc pouvoir stocker une représentation *hiérarchique* de notre discrétisation.

**Octree topologique** Toutes les particules (les points d'échantillonnage) sont stockées dans un octree. Les 8 filles d'une particule donnée seront celles qui discrétiseront la même zone de l'espace si cette particule se divise. La structure d'octree est bien appropriée à une discrétisation hiérarchique car un cube récursivement divisé en 8 nouveaux cubes est la seule structure offrant un échantillonnage régulier de l'espace à chaque niveau. L'octree est construit de bas en haut comme suit (voir figure 6) :

– (a) La grille cubique 3D est adaptée sur la taille de l'objet et régulièrement divisée jusqu'à son plus haut niveau.

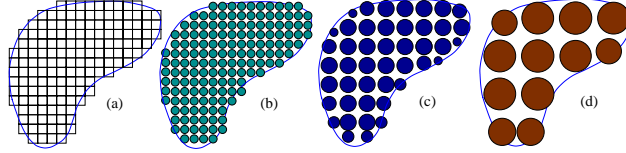


FIG. 6: Création récursive par remontée de l'octree. La taille représente la masse des particules.

– (b) On détermine ensuite lesquels de ces cubes sont à l'intérieur de l'objet en oubliant les autres. Ces cubes forment le niveau le plus haut de l'octree (les feuilles). Leur masse est celle de l'objet divisée par leur nombre.

– (c)&(d) Les niveaux suivants sont alors récursivement construits en regroupant par groupes de 8 (ou moins au bord) les particules entre elles en utilisant :

$$m = \sum m_i, \quad \mathbf{p} = \frac{\sum m_i \mathbf{p}_i}{m} \quad (13)$$

où  $m$  est la masse, et  $\mathbf{p}$  la position de la particule mère. Cette moyenne garantie une conservation de la masse et un bon échantillonnage de la matière.

Remarquez que c'est un octree *topologique* et non *spacial*. Il stocke les relations mère/fille mais n'est pas la classique division spatiale d'un octree, car il va se déformer durant l'animation.

**Structure des voisins** A un niveau donné de la hiérarchie, on définit le voisinage d'une particule comme l'ensemble des cellules adjacentes dans l'octree, que ce soit par une face, une arête ou un sommet. La figure 7 le montre en 2D.

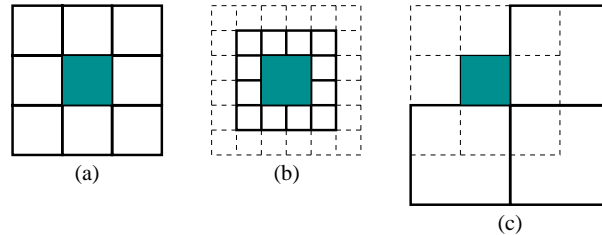


FIG. 7: Définition des voisins (en ligne épaisse) d'une particule donnée (grisée), en 2D. (a) au même niveau, (b) au niveau supérieur et (c) inférieur. En 3D, chaque particule a 89 voisins potentiels.

Afin de réduire le nombre de voisins et pour assurer un bon échantillonnage, nous forçons l'octree à rester restreint [VB87]. Durant l'animation deux particules actives voisines ne peuvent avoir plus d'un niveau d'écart (cf figure 8(a)). Ceci limite le nombre de voisins possible qui peuvent être précalculés et stockés pour chaque particule. Une particule a jusqu'à 89 voisins potentiels, mais comme ils ne peuvent être actifs en même temps, on a au maximum 56 (si tous les voisins sont divisés) et au minimum 14 voisins à un moment donné.

**Mise à jour de la structure** L'octree restreint impose certaines contraintes sur la division et le regroupement des particules. A chaque pas de temps, des listes de particules se divisant/regroupant sont



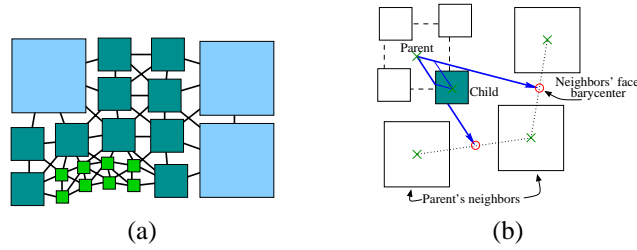


FIG. 8: (a) Durant la simulation, les voisins d'une particule ne peuvent appartenir qu'au même niveau de résolution, au niveau supérieur ou inférieur. (b) Le système de coordonnées locales (flèches), défini par les voisins de la mère assure un bon échantillonnage spatial, même lorsque l'objet est déformé.

calculées d'après les critères de la section 4.1. Ces listes sont parcourues, en favorisant la division dans un but de stabilité :

- Une particule peut se regrouper si et seulement si toutes ses voisines ont un niveau égal ou inférieur.
- Une particule peut se diviser si et seulement si toutes ses voisines ont un niveau égal ou supérieur. Si ce n'est pas le cas, les voisines de niveau inférieur sont insérées dans la liste des particules à diviser, afin que cette particule puisse se diviser par la suite.

Chaque particule a des pointeurs vers toutes ses voisines potentielles et garde une liste de celles qui sont actives (i.e. réellement simulées) parmi elles. Lorsqu'une particule se divise/regroupe, des tables précalculées permettent de mettre à jour la liste de ses voisines actives, ainsi que celles de ses voisines, pour un coût nul.

**Position** Lorsque des particules se regroupent, leur mère moyenne leurs différentes valeurs (position, vitesse, déplacement...) pour calculer les siennes. Le problème est plus complexe lors de la division. Comme nous voulons que notre discrétisation spatiale soit la plus uniforme possible, les positions des nouvelles particules doivent être bien choisies. Nous avons choisi de définir la position d'une particule dans un repère centré sur sa mère (voir figure 8(b)).

Les axes de ce repère sont définis par les voisins de la mère (grâce à des centres de masse de ces voisines dans chaque direction). Ce repère local permet de bien placer les filles même lorsque l'objet est déformé. Un traitement spécifique est effectué pour les particules du bord. Lors du regroupement les coordonnées des particules dans le repère sont remises à jour de sorte qu'elles reviendront à la même place lors de la division suivante.

## 5.2 Structure de données temporelle

Comme chaque particule discrétise un volume 8 fois plus petit que sa mère, le critère de Courant (voir §4.2) stipule que son pas de temps doit être au moins deux fois plus faible que celui de sa mère. Pour cette raison et pour pouvoir synchroniser les différents pas de temps, on a choisi de fixer les différents pas de temps comme étant des puissances de 2 fois le pas de temps minimum, qui est lui déterminé par (12).

En pratique les particules actives sont stockées dans des listes correspondant à leur pas de temps qui sont parcourues au besoin. Changer le pas de temps d'une particule signifie simplement la changer de liste.

# 6 Résultats

Nous avons essayé notre modèle sur différents exemples simples et dans une application médicale.

## 6.1 Exemple d'école

Nous travaillons sur une tige, dont une des faces est maintenue fixe, qui se plie sous l'effet de la gravité, dans un milieu visqueux. La figure 9 montre les états initiaux et finaux, avec des coefficients de Lamé de  $\mu = 5000$  et  $\lambda = 1000000$ .

level	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

Ce tableau montre le temps moyen de simulation 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 éeement simulées varie entre 4 (début) et 88, se stabilisant à 32 (niveau 3) quand la barre atteint sa position d'équilibre. (voir Figure 9b pour des images de cette animation)

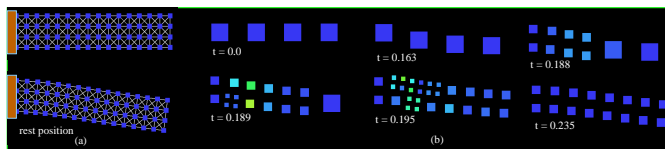


FIG. 9: Une tige oscillant sous la gravité. (a) Simulation de référence faite avec 256 particules. (b) simulation adaptative.

## 6.2 Une application temps-réel

Donner des simulateurs aux apprentis chirurgiens a de nombreux avantages, éthiques et financiers : on n'utilise plus de cadavres ou d'animaux et le chirurgien peut s'entraîner aussi longtemps qu'il le souhaite. Dans le contexte de la chirurgie laparoscopique <sup>3</sup>, les opérations du foie sont un bon exemple d'école, étant l'une des applications les plus courantes.

Nous utilisons la technique de cet article pour créer un simulateur laparoscopique. Le foie virtuel, issu de données anatomiques, peut éeagir en temps-réel aux actions de l'utilisateur, comme le montre la figure 10. Le caractère multirésolution de notre méthode est ici crucial car une simulation à discrétisation fixe aurait été trop lente. Nous pouvons garantir une fréquence d'affichage de 12 Hz sur une Silicon Graphics Onyx2 avec un processeur R10K. La fréquence d'affichage est garantie par une limitation du coût de calcul (qui varie linéairement avec le nombre de particules et leur pas de temps et peut donc être mesuré) qui interdit une division tron importante de modèle.

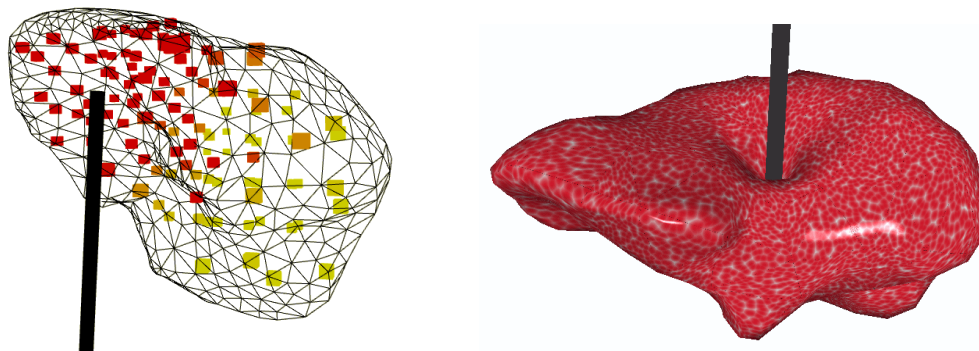


FIG. 10: Bien que les contraintes imposées dépassent parfois le formalisme des petites déformations, notre modèle présente néanmoins une bonne réponse à l'utilisateur. L'ajout de textures augmente sensiblement le réalisme du simulateur.

## 7 Conclusions et travaux futurs

Nous avons proposé dans cet article une technique d'animation multirésolution qui permet d'animer des objets déformables. Ce nouveau modèle bénéficie d'une discrétisation adaptative en temps et en espace pour minimiser les calculs. Basé sur l'élasticité linéaire, il utilise des opérateurs discrets de calcul des dérivées sur des grilles irrégulières. Des critères intuitifs permettent de raffiner ou de simplifier automatiquement la discrétisation tout en garantissant un seuil d'erreur.

Cette approche est très générale et peut facilement être améliorée. Une intégration implicite ou de meilleurs critères de raffinements peuvent être testés. Les opérateurs discrets introduits ici, bien que suffisants lorsque l'objet n'est pas trop déformé doivent être améliorés pour être plus généraux et robustes. L'animation multirésolution deviendra alors un outil puissant et efficace, après de nombreuses améliorations durant cette décennie.

<sup>3</sup>La chirurgie laparoscopique est une technique dans laquelle le chirurgien n'opère que de petits orifices dans le patient et travaille à l'intérieur à l'aide d'outils et d'une caméra.

## Références

- [Bar96] David Baraff. Linear-time dynamics using lagrange multipliers. In *SIGGRAPH 96 Conference Proceedings*, Computer Graphics Proceedings, Annual Conference Series, pages 137–146. ACM SIGGRAPH, Addison Wesley, August 1996. ISBN 0-201-94800-1.
- [BNC96] Morten Bro-Nielsen and Stéphane Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Eurographics*, pages 21–30, 1996.
- [BW98] David Baraff and Andrew Witkin. Large steps in cloth simulation. In Michael Cohen, editor, *SIGGRAPH 98 Conference Proceedings*, Annual Conference Series, pages 43–54. ACM SIGGRAPH, Addison Wesley, July 1998.
- [DCA99] Hervé Delingette, Stéphane Cotin, and Nicolas Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Computer Animation '99*, pages 70–81, may 1999.
- [DCG96] Mathieu Desbrun and Marie-Paule Cani-Gascuel. Smoothed particles : A new approach for animating highly deformable bodies. In Springer Computer Science, editor, *7th Eurographics Workshop on Animation and Simulation*, pages 61–76, Poitiers, France, September 1996.
- [DMSB99] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan H. Barr. Implicit fairing of arbitrary meshes using diffusion and curvature flow. In *SIGGRAPH 99 Conference Proceedings*, Annual Conference Series. ACM SIGGRAPH, Addison Wesley, 1999.
- [DSB99] Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Interactive animation of structured deformable objects. *To appear in Graphics Interface '99*, 1999.
- [Fau98] François Faure. Interactive solid animation using linearized displacement constraints. *9th Eurographics Workshop on Computer Animation and Simulation*, September 1998.
- [For88] Bengt Fornberg. Generation of finite difference formulas on arbitrarily spaced grids. *Math. Comput.*, 51 :699–706, 1988.
- [Fuj95] Koji Fujiwara. Eigenvalues of laplacians on a closed riemannian manifold and its nets. In *Proceedings of the AMS 123*, pages 2585–2594, 1995.
- [GCD<sup>+</sup>98] J.D. Gascuel, M.P. Cani, M. Desbrun, E. Leroy, and C. Mirgon. Simulating landslides for natural disaster prevention. In *9th Eurographics Workshop on Computer Animation and Simulation (EGCAS'98)*, September 1998.
- [GTT89] Jean-Paul Gourret, Nadia Magnenat Thalmann, and Daniel Thalmann. Simulation of object and human skin deformations in a grasping task. *Computer Graphics*, 23(3) :21–29, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).
- [HPH96] Dave Hutchinson, Martin Preston, and Terry Hewitt. Adaptive refinement for mass/spring simulation. In *7th Eurographics Workshop on Animation and Simulation*, pages 31–45, Poitiers, France, September 1996.
- [MT92] Dimitri Metaxas and Demetri Terzopoulos. Dynamic deformation of solid primitives with constraints. *Computer Graphics*, 26(2) :309–312, July 1992. Proceedings of SIGGRAPH'92 (Chicago, Illinois, July 1992).
- [PW89] Alex Pentland and John Williams. Good vibrations : Modal dynamics for graphics and animation. *Computer Graphics*, 23(3) :215–222, July 1989. Proceedings of SIGGRAPH'89 (Boston, MA, July 1989).
- [TF88] Demetri Terzopoulos and Kurt Fleischer. Modeling inelastic deformations : Viscoelasticity, plasticity, fracture. *Computer Graphics*, 22(4) :269–278, August 1988. Proceedings of SIGGRAPH'88 (Atlanta, Georgia).
- [TG70] S.P. Timoshenko and J.N. Goodier. *Theory of Elasticity*. McGraw-Hill, 1970.
- [TPBF87] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models. *Computer Graphics*, 21(4) :205–214, July 1987. Proceedings of SIGGRAPH'87 (Anaheim, California).
- [TW88] Demetri Terzopoulos and Andrew Witkin. Physically based model with rigid and deformable components. *IEEE Computer Graphics and Applications*, pages 41–51, December 1988.
- [VB87] Brian Von Herzen and Alan H. Barr. Accurate triangulations of deformed, intersecting surfaces. In Maureen C. Stone, editor, *Computer Graphics (SIGGRAPH '87 Proceedings)*, volume 21, pages 103–110, July 1987.
- [WW90] Andrew Witkin and William Welch. Fast animation and control for non-rigid structures. *Computer Graphics*, 24(4) :243–252, August 1990. Proceedings of SIGGRAPH'90 (Dallas, Texas, August 1990).