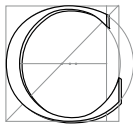


---

# Travaux antérieurs

---



CE CHAPITRE présente quelques unes des méthodes qui ont été développées dans le domaine de l’animation en synthèse d’images pour animer des objets déformables. Le but étant d’avoir un effet visuel, on pourra définir comme déformables des objets suffisamment mous pour que leur déformation soit visible et intéressante. Nous avons regroupé les différentes approches en fonction de la méthode de résolution employée, et nous en décrirons à chaque fois les avantages, les inconvénients et l’adéquation à nos buts.

Après une revue plus exhaustive des méthodes permettant des applications temps-réel ainsi que de celles mettant en œuvre des processus multirésolution, buts visés par cette thèse, nous justifierons le choix des méthodes employées par la suite.

D’autres descriptions des modèles déformables existants peuvent être trouvés dans l’état de l’art réalisé en 1997 par Sarah Gibson et Brian Mirtich [GM97] et dans les thèses de Mathieu Desbrun [Des97] et Stéphane Cotin [Cot97] qui portent sur le sujet. Nous ne traitons pas ici le cas des objets déformables purement surfaciques comme les tissus et le lecteur intéressé pourra trouver un état de l’art plus complet dans [NG96].

## 1 Déformations géométriques de la surface

Cette première classe d’algorithmes est un peu à part. Elle ne considère pas l’objet déformable comme un tout avec des propriétés physiques intrinsèques, mais considère plutôt la surface qui le délimite et que l’on va chercher à animer. Ces méthodes, particulièrement employées dans le design, sont néanmoins intéressantes pour nous car elles seront par la suite reprises dans certains algorithmes plus “physiques”.

### 1.1 Surfaces de forme libre

Historiquement, la conception géométrique assistée par ordinateur (CAO) fut l’un des premiers domaines à utiliser des techniques de déformation d’objets. Les designers voulaient pouvoir manipuler et affiner la représentation numérique de leur objet. De là sont nées les courbes de Bézier et la famille des *splines* : courbes d’interpolation, B-splines, splines rationnelles, non uniformes (NURBS), toutes définissant avec peu de valeurs une courbe lisse, en 1, 2 ou 3 dimensions (voir [BBB87, Far90] pour une étude plus complète). Ces courbes et les surfaces associées seront généralement définies par des *points de contrôle* dont le déplacement modifiera

l'aspect de la courbe de manière intuitive (voir Fig. 1.1). On peut aussi déplacer directement la courbe ou la surface B-spline plutôt que les points de contrôle [BB89].

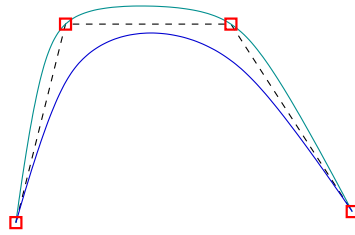


FIG. 1.1: Quatre points de contrôle, la spline d'interpolation qui passe par eux (à l'extérieur) ainsi que la courbe de Bézier qu'ils définissent, tangentes au départ et à l'arrivée au polygone de contrôle (à l'intérieur).

Des approches multirésolution à base d'ondelettes ont été développées pour permettre de modifier la courbe, ou par extension la surface, de manière précise et locale ou au contraire dans sa forme globale [FS94].

## 1.2 Les déformations de l'espace

Une autre façon de modifier globalement la forme d'un objet est de déformer l'espace dans lequel il réside. On peut par exemple remplacer la coordonnée  $x$  par  $-x$  pour effectuer une symétrie. Les transformations appliquées peuvent être bien plus complexes et ont l'avantage, en déformant tout l'espace, de pouvoir s'appliquer aussi bien à des surfaces polygonales, splines, paramétriques qu'implicites. La Figure 1.2 illustre cette idée.

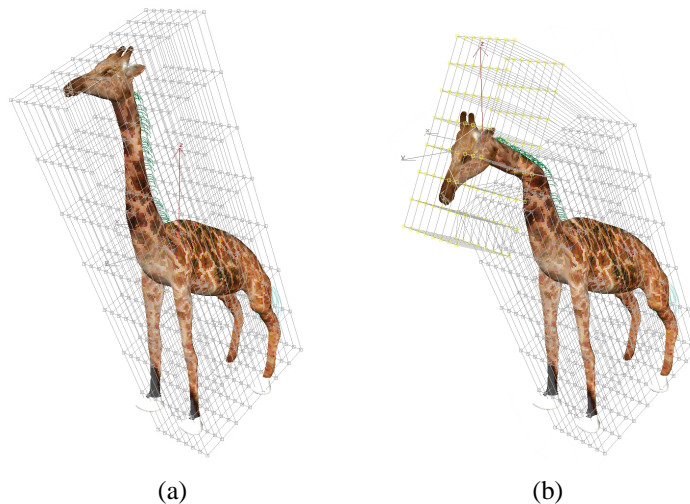


FIG. 1.2: Déformation de l'espace illustrée sur un modèle de girafe : la grille originale (a) et celle obtenue après une déformation localisée (b).

Ces travaux furent débutés par Al Barr [Bar84] et consistaient alors en des fonctions matricielles de  $\mathbb{R}^3$  dans  $\mathbb{R}^3$  (provoquant des déformations rigides, des pliages, des effilements) qui pouvaient être combinées pour donner des effets complexes.

Sederberg et Parry ont généralisé cette approche [SP86] par l'introduction des FFD, pour *Free-Form Deformation*. L'objet est plongé dans un maillage qui va pouvoir être déformé, chaque cellule du maillage entraînant avec elle les points qu'elle contient au moyen d'interpolations utilisant habilement les polynômes de Bernstein. Des extensions, tirant parti de la séparation en plusieurs sous-régions du maillage déformant [Coq90] ou utilisant la multirésolution [MJ96] (le maillages pouvant être de topologie arbitraire grâce à utilisation d'algorithmes de subdivision) furent présentées.

Citons aussi une méthode permettant d'inférer par l'utilisation de moindres carrés la modification du maillage directement de celle que l'on veut faire subir aux points de la surface [HHK92].

## 1.3 Surfaces implicites

La surface d'un objet peut également être représentée *implicitement*, plutôt qu'*explicitement* à l'aide de polygones ou de splines. On crée pour cela un champ scalaire dans tout l'espace, et la surface est définie

comme le lieu où ce champ a une valeur donnée. Ces champs dérivent généralement de potentiels et l'on parle de *surface équipotentielle* ou d'*iso-surfaces*. Ce potentiel est souvent une fonction dépendant de façon monotone de la distance à un squelette, qui peut être un point, un segment ou toute autre forme plus complexe [BW90].

Cette formulation a de nombreuses propriétés intéressantes : selon que la valeur du champ en un point de l'espace a une valeur inférieure ou supérieure à l'iso-valeur, on sait si l'on se trouve à l'intérieur ou à l'extérieur de l'objet, ce qui permet notamment une détection des collisions très rapide. La combinaison de plusieurs objets implicites peut être très simple en sommant les différents potentiels qui les génèrent, ce qui permet également la gestion d'objets à topologie variable, l'iso-surface se divisant naturellement lorsque les deux squelettes se séparent.

La visualisation de telles surfaces est par contre plus complexe, leur polygonalisation pouvant en particulier s'avérer difficile. On pourra utiliser l'algorithme du *marching cube* [WMW86, Blo87, Blo88], des maillages locaux [DTG96] ou la méthode à base de particules de Witkin et Heckbert [WH94] pour les afficher.

Ce formalisme définit la surface d'un objet comme issue de potentiels. Son animation va alors consister à animer les squelettes qui la génèrent, la surface se modifiant en conséquence avec d'intéressantes propriétés de continuité ou de changements de topologie. On ne modifiera néanmoins que rarement manuellement la position de ces squelettes, qui sont généralement des masses ponctuelles, préférant laisser cette tâche à des modèles générateurs, comme on le verra par la suite.

## 1.4 Conclusion

Ces méthodes géométriques ont été surtout présentées comme une introduction car certaines seront reprises par la suite. Elles ont l'avantage d'être généralement très rapides et offrent un grand contrôle sur la déformation produite. Elles ne peuvent par contre s'appliquer directement au cas qui nous préoccupe puisque le mouvement est directement créé par l'utilisateur et que c'est de son habileté que dépendra le résultat.

On utilisera donc plutôt des modèles générateurs, qui simuleront des lois de comportement, plus ou moins physiques, et généreront donc *automatiquement* le mouvement de l'objet. Il existe beaucoup de méthodes génératrices et nous allons en décrire les principales familles dans ce qui suit.

# 2 Déformations globales

Ces méthodes déforment l'objet *globalement* en le considérant comme un tout mais en utilisant maintenant des lois inspirées de la physique pour générer son mouvement.

## 2.1 Dynamique modale

L'algorithme proposé par Pentland et Williams tire parti de l'analyse des *modes vibratoires* de l'objet [PW89]. Les lois physiques de l'élasticité linéaire conduisent à une équation du second ordre classique du type :

$$M \frac{\partial^2 \mathbf{u}}{\partial t^2} + D \frac{\partial \mathbf{u}}{\partial t} + K \mathbf{u} = \mathbf{f} \quad (1.1)$$

où  $M$ ,  $D$  et  $K$  sont des matrices représentant respectivement la masse, l'amortissement et la raideur du matériau.  $\mathbf{u}$  mesure le déplacement des différents nœuds de discrétisation et  $\mathbf{f}$  est la force qui y est appliquée. Ces trois matrices sont symétriques, et sous certaines hypothèses ( $D$  doit en particulier être égale à  $M$  à un coefficient multiplicateur près), sont diagonalisables dans la même base. L'équation précédente se simplifie alors grandement et devient, en notant avec un tilde l'expression des variables dans la nouvelle base :

$$\tilde{M} \frac{\partial^2 \tilde{\mathbf{u}}}{\partial t^2} + D \frac{\partial \tilde{\mathbf{u}}}{\partial t} + K \tilde{\mathbf{u}} = \tilde{\mathbf{f}} \quad (1.2)$$

où les 3 matrices  $\tilde{M}$ ,  $\tilde{D}$  et  $\tilde{K}$  sont maintenant diagonales. On n'a donc plus ici qu'une liste d'équations différentielles *indépendantes*, chacune décrivant un mode vibratoire de l'objet. Chacun de ces modes a une fréquence donnée et la simplification consiste ici à ne garder que les plus basses fréquences, qui participent le plus au

rendu du mouvement. L'animation sera alors plus stable numériquement, les hautes fréquences nécessitant un faible pas de temps ayant été supprimées<sup>1</sup>.

Une autre approximation consiste à remarquer que les modes vibratoires de basses fréquences d'un objet dépendent plus de ses dimensions que de sa forme précise. On précalcule donc des modes vibratoires pour un solide *parallépipédique*, et on les interpole aux dimensions de la boîte englobante de l'objet.

## 2.2 Déformation globale dynamique

Witkin et Welch ont présenté une technique d'animation et de contrôle globaux d'objets déformables [WW90]. Elle ajoute une composante temporelle aux déformations de l'espace de Barr [Bar84] vues plus haut, en faisant varier au cours du temps les coefficients de la matrice de déformation. Les énergies cinétiques et potentielles d'une configuration donnée peuvent être calculées et d'elles dérive une équation lagrangienne qui définit la dynamique du mouvement.

Cette méthode fut ensuite reprise dans [BW92] et par [MT92] dans un modèle à couches.

## 2.3 Conclusion

Associer des lois physiques aux déformations globales d'un objet le rend capable de mouvements autonomes. Ces méthodes, grâce aux approximations qu'elles mettent en œuvre, sont en outre très rapides à calculer. Mais c'est de ces mêmes approximations que provient leur faible adéquation avec nos objectifs, le mouvement simulé étant trop grossier pour être visuellement réaliste. Ces déformations globales conviennent en effet mal aux détails locaux recherchés dans un simulateur chirurgical. Parfois utilisées en synthèse d'images pour leur rapidité et leur aspect intuitif, nous pourrions les réserver à l'animation d'objets distants et secondaires.

À noter, dans une optique multirésolution, l'utilisation qui pourrait être faite des modes vibratoires du modèle de [PW89]. En fonction du temps disponible ou de la qualité de simulation désirée, on pourra en effet utiliser un plus ou moins grand nombre de modes vibratoires et ainsi régler aisément le compromis qualité - rapidité de calcul de l'animation, sans toutefois pouvoir l'adapter localement.

# 3 Les modèles masses-ressorts

De part leur simplicité et leur aspect intuitif, les modèles masses-ressorts ont eu un grand succès dans le domaine de l'image de synthèse. Une vision *continue* de l'espace étant difficilement compatible avec la modélisation informatique, on va le *discrétiser* pour ne plus le considérer que comme un ensemble de valeurs ponctuelles, se différenciant ainsi des méthodes *globales* de la section précédente.

## 3.1 Principe

Un objet va ainsi être représenté par un nombre fini de points, si possible équirépartis pour une meilleure représentation. Leur comportement les uns vis-à-vis des autres est intuitivement celui de petites masses ponctuelles reliées à leurs voisins par des relations de type ressorts (Fig. 1.3).

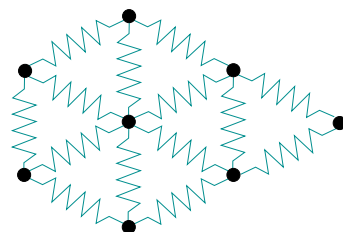


FIG. 1.3: Discrétisation d'un objet 2D à l'aide de masses-ressorts.

Avec des ressorts, il est facile de calculer la force subie par chaque masse à chaque instant en fonction des distances à ses voisins :

$$\mathbf{f}^i = \sum_{\text{voisins } j} k^{ij} (l^{ij} - l_{\text{initiale}}^{ij})$$

<sup>1</sup>Voir l'annexe B pour les problèmes liés à l'intégration.

où  $k^{ij}$  est la raideur du ressort entre le point  $i$  et son voisin  $j$ ,  $l^{ij}$  étant sa longueur.

Les ressorts peuvent être plus complexes et comporter par exemple des termes d'amortissement. Dans tous les cas, l'équation fondamentale de la dynamique pour chaque masse sera de la forme (voir Eq. 1.1) :

$$m \frac{\partial^2 \mathbf{x}}{\partial t^2} + d \frac{\partial \mathbf{x}}{\partial t} + k \mathbf{x} = \mathbf{f}_{ext}$$

Comme vu précédemment,  $m$ ,  $d$  et  $k$  sont trois scalaires désignant respectivement la masse, le coefficient de frottement et la raideur alors que  $\mathbf{x}$  est lié à la position du point.  $\mathbf{f}_{ext}$  désigne les forces externes appliquées à l'objet, comme la gravité. L'algorithme consistera donc à intégrer ces équations au cours du temps, et ce pour chaque point séparément<sup>2</sup>.

De tels modèles ont été très tôt appliqués par Platt et Badler (sans coefficient d'amortissement) [PB81] à l'animation de visage, par Luciani [Luc85] pour l'animation de marionnettes ou Chadwick [CHP89] (en utilisant un réseau de ressorts pour simuler le maillage déformant d'une FFD) pour animer la chair de personnages.

### 3.2 Raffinements du modèle

Miller anima des reptiles avec cette méthode [Mil88] en ajoutant des contraintes de préservation de volume ainsi que la propagation d'ondes sinusoïdales à travers le corps, simulant le déplacement. À noter que l'affichage utilisait des surfaces splines et donnait un aspect très lisse cachant le réseau cubique sous-jacent.

En mélangeant des ressorts de raideurs différentes, Terzopoulos et Waters [TW90] animent un visage composé de plusieurs couches de ressorts représentant l'épiderme, le derme et les muscles, ces derniers étant *actifs* ce qui signifie que leurs paramètres étaient modifiés au cours du temps. Des contraintes additionnelles viennent ici aussi chercher à préserver le volume. La technique fut ensuite améliorée pour s'adapter à des visages réels acquis par imagerie médicale, en optimisant la position des masses en fonction des caractéristiques du visage [LTW95].

En plus des classiques ressorts longitudinaux, on peut ajouter des ressorts angulaires et ainsi obtenir des comportements nouveaux et plus complexes.

Joukhadar propose une méthode de modification du pas de temps d'intégration pour l'optimiser en évitant les divergences [Jou96]. Lorsqu'est détectée une trop grande variation de l'énergie du système, synonyme de problème d'intégration, le pas de temps est diminué.

On peut contrôler l'anisotropie du matériau en ne plaçant les ressorts que dans des directions privilégiées, ceux-ci agissant alors sur les tétraèdres qui composent l'objet [BC00]. Il existe en effet une anisotropie intrinsèque et non contrôlée dans les modèles masses-ressorts liée à la direction des ressorts.

Imposer une distance maximale d'élongation du ressort permet de limiter les trop grands étirements dans une simulation de tissus [Pro95]. Boux de Casson propose également de casser certaines des liaisons masses-ressorts en fonction des contraintes, simulant ainsi les déchirures d'un tissu [BdCL00].

Dans ce domaine, citons les récents travaux de Baraff et Witkin [BW98] qui peuvent grâce à une intégration implicite<sup>3</sup> simuler des ressorts de très forte raideur et obtenir de superbes animations de vêtements, utilisées dans la production audiovisuelle et disponibles dans le logiciel *Maya*.

### 3.3 Conclusion

Les méthodes masses-ressorts ont de très nombreux avantages. Simples à implémenter, intuitives et rapides, elles donnent de bons résultats facilement. Elles sont de plus facilement parallélisables et peuvent être temps-réel sur des machines d'entrée de gamme avec suffisamment peu de masses simulées.

Tous ces avantages devraient en faire un modèle de choix dans notre cas, et de fait, de nombreux simulateurs chirurgicaux utilisent les réseaux masses-ressorts (voir Sec. 9). Il conviendra néanmoins de bien gérer les problèmes de stabilité numérique qui ne manquent pas de troubler le modèle pour de fortes raideurs. Un matériau organique demandera un pas de temps faible, limitant le nombre de masses simulables si l'on veut une simulation temps-réel. Nous avons pris le parti dans cette thèse d'explorer les intérêts d'une animation multirésolution et les masses-ressorts s'y prêtent par contre très mal.

<sup>2</sup>Voir l'annexe B.

<sup>3</sup>Voir l'annexe B.

S'ils permettent, en jouant sur les paramètres du système, d'obtenir un effet visuel donné à force d'essais successifs, il est impossible de prédire leur comportement dès lors que l'on modifie notablement la topologie ou la géométrie du maillage. Le *comportement dynamique* de maillages à différentes résolutions d'un même objet ne pourra donc être le même, et quand bien même on parviendrait à minimiser les différences en affinant les réglages, resterait le problème de la cohabitation des différentes résolutions. La simulation adaptative de tissus proposée par Hutchinson [HPH96] souffre ainsi de ce que le comportement du matériau évolue au cours de la simulation.

En pratique, même si l'on finit par avoir une intuition du comportement du système en fonction des paramètres qui le gouvernent, modifier la résolution du maillage pour l'affiner demande de nouvelles simulations pour obtenir l'effet désiré. Quelle raideur affecter alors à un ressort de telle longueur, dans telle configuration et avec tant de voisins ?

Ces problèmes d'*identification de paramètres* ont été abordés dans la littérature [DKT95, LPC95] en utilisant des méthodes, lentes, de recuit simulé ou des algorithmes génétiques. Mais les solutions ne cherchent à adapter les coefficients que sur des mouvements élémentaires, sans garantir le comportement de tout autre déplacement, ce qui ne saurait convenir.

## 4 Les systèmes de particules

### 4.1 Principe

Les systèmes de particules sont une extension du modèle masses-ressorts précédent. La différence réside dans le fait qu'on autorise les masses à interagir avec un ensemble de masses qui peut évoluer au cours de la simulation. Chaque masse interagit potentiellement avec toutes les autres, mais on seuilte généralement à un rayon maximal sa zone d'influence pour limiter les calculs.

Terzopoulos *et al.* [TPF89] simulent un matériau en train de fondre en faisant varier la raideur des ressorts en fonction de la température (soumise à l'équation de la chaleur). Le ressort disparaît lorsque le point de fusion est atteint et l'on passe alors à un modèle de particules. Tonnesen [Ton91] utilise le terme de *particule thermique* et simule lui aussi liquides et solides. Plus souple que les masses-ressorts, les systèmes de particules n'imposent pas de restriction topologique, offrant la possibilité de simuler des fluides [MP89, LHVD95].

Les forces inter-particules sont souvent des lois d'attraction-répulsion. Similaires aux forces de *Lennard-Jones* [MP89], elles s'apparentent à celles qui existent entre deux atomes (voir Fig. 1.4) et présentent, comme les ressorts, une distance d'équilibre où la force est nulle. La force diminue par contre ici après une certaine distance pour s'annuler ensuite.

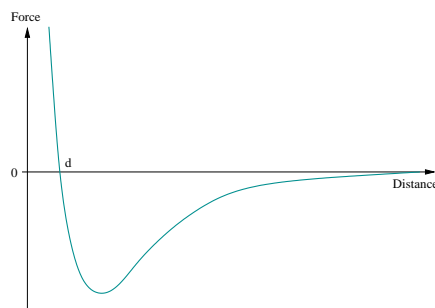


FIG. 1.4: Intensité de la force résultant d'un potentiel de Lennard-Jones en fonction de la distance.

On peut imaginer des fonctions plus complexes : les lois d'interaction peuvent être arbitrairement complexes et définies algorithmiquement [LC86], avoir plusieurs états d'équilibre [LHVD95, Rey97] ou par exemple dépendre de l'historique de la simulation. Les particules peuvent de plus avoir des "tailles" différentes [Jim93, LJR<sup>+</sup>91] ou être *orientées*, subissant alors des couples supplémentaires fonctions de l'orientation de leurs voisines [ST92, LP95].

L'intérêt des systèmes de particules est qu'ils sont faciles à mettre en œuvre et que des lois d'interaction simples qui les gouvernent peuvent naître des comportements complexes, dits *émergents*. Reeves a ainsi très tôt simulé des phénomènes tels que le feu et la fumée [Ree83] sans se préoccuper de la physique très complexe qui les régit. Reynolds a lui simulé le comportement global de bancs de poissons ou de vols d'oiseaux, chaque individu ne réagissant pourtant que selon des lois simples [Rey87].

L'affichage du modèle est simpliste avec des réseaux masses-ressorts à topologie fixe et dans lesquels les mailles surfaciques peuvent être polygonalisées. Il est par contre plus complexe dans le cas où l'on ne dispose que d'un ensemble de particules, représentant qui plus est un objet à topologie variable. Les surfaces implicites, en utilisant les particules comme squelettes générateurs de potentiel, sont alors bien adaptées, mais plus coûteuses. Elles peuvent aussi modéliser les déformations locales de l'objet autour de la zone de collision [CGD97].

## 4.2 Conclusion

Les systèmes de particules sont une extension des réseaux masses-ressorts. Ils permettent de simuler une plus grande variété de comportements, comme celui des gaz ou des liquides. L'algorithme reste simple et l'on peut simplement modifier la topologie de l'objet ce qui autorise les découpes. Il conviendra par contre de bien gérer la complexité algorithmique qui peut apparaître avec beaucoup de particules lorsque l'on doit chercher celles qui interagissent entre elles. On utilisera pour cela un partitionnement de l'espace qui ramènera le coût quadratique à un coût quasi-linéaire en le nombre de masses.

Le problème avec ces méthodes est, tout comme avec les masses-ressorts, qu'on ne peut généralement pas garantir un comportement indépendant de la résolution. L'exception est le modèle SPH qui le permet grâce à l'utilisation de forces inter-particules bien choisies.

## 5 Le modèle SPH

Le modèle SPH, pour *Smoothed Particles Hydrodynamics*, est issu de l'astrophysique et a l'avantage sur les précédents de proposer une véritable *équation d'état* du matériau, conduisant à un comportement indépendant de la résolution [Mon92]. C'est une méthode à base de particules comme vu précédemment, mais elles sont maintenant vraiment considérées comme les *points d'échantillonnage* d'un petit volume les entourant et qu'elles représentent. C'est une méthode Lagrangienne, désignée comme méthode Lagrangienne *libre libre* par Rémi Cozot dans sa thèse [Coz96].

Simplifiée, elle a été employée dans le cadre de l'animation de phénomènes gazeux [SF93, SF95] et fut à la base d'un modèle décrit dans la thèse de Mathieu Desbrun [Des97], dont nous reprenons ici quelques explications et à laquelle on se référera pour des explications plus détaillées.

### 5.1 Passage du continu au discret

Un champ continu  $f$  va être approximé en un point  $\mathbf{x}$  par une valeur portée par particule et calculée comme une moyenne locale des valeurs du champ sur un petit volume  $V$  autour de la particule. Cette valeur moyenne locale, notée  $\langle f(\mathbf{x}) \rangle$ , peut être écrite comme la convolution de  $f$  par un noyau de filtrage :

$$\langle f(\mathbf{x}) \rangle = \int_V f(\mathbf{x}') W_h(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (1.3)$$

où  $W_h$  est un filtre de lissage dont le paramètre  $h$  contrôle l'étendue du filtrage. L'intégrale de  $W_h$  doit être égale à 1 quel que soit  $h$ , et  $W_h$  doit tendre vers l'impulsion de Dirac quand  $h$  tend vers zéro.

On passe d'une intégrale sur le *volume* à une intégrale sur la *masse* en utilisant la densité de masse  $\rho$  :

$$\begin{aligned} \langle f(\mathbf{x}) \rangle &= \int_V \frac{f(\mathbf{x}')}{\rho(\mathbf{x}')} W_h(\mathbf{x} - \mathbf{x}') \rho(\mathbf{x}') d\mathbf{x}' \\ &= \int_M \frac{f(\mathbf{x}')}{\rho(\mathbf{x}')} W_h(\mathbf{x} - \mathbf{x}') dm \end{aligned} \quad (1.4)$$

Une approximation discrète de l'équation précédente, obtenue par une méthode de Monte-Carlo grâce aux valeurs du champ pour les autres particules donne ( $f_j = f(\mathbf{x}_j)$  et  $m_j$  est la masse de la particule  $j$ ) :

$$\langle f(\mathbf{x}) \rangle \simeq \sum_j m_j \frac{f_j}{\rho_j} W_h(\mathbf{x} - \mathbf{x}_j) \quad (1.5)$$

En intégrant par parties, et en remarquant que  $W_h$  s'annule au bord du volume, on peut calculer le *gradient* du champ :

$$\int_V (\mathbf{grad} f)(\mathbf{x}') W_h(\mathbf{x} - \mathbf{x}') d\mathbf{x}' = \int_V f(\mathbf{x}') (\mathbf{grad} W_h)(\mathbf{x} - \mathbf{x}') d\mathbf{x}' \quad (1.6)$$

Ce qui se traduit en version discrétisée par :

$$\langle (\mathbf{grad} f)(\mathbf{x}) \rangle \simeq \sum_j m_j \frac{f_j}{\rho_j} (\mathbf{grad} W_h)(\mathbf{x} - \mathbf{x}_j) \quad (1.7)$$

Un ensemble, même désordonné, de points d'échantillonnage permet ainsi de calculer les valeurs de champs continus et de leurs dérivées de façon simple par filtrage local. Ceci va permettre de simuler les équations d'état qui gouvernent le comportement d'un matériau.

## 5.2 L'équation d'état du matériau

Les particules composant le matériau vont être soumises à un *champ de pression* qui va déterminer leurs déplacements. Les forces générées sont proportionnelles au gradient de la pression [Bat73], qui va pouvoir être calculé avec l'Équation 1.7 précédente.

Dans [Des97], l'expression du champ de pression  $P$ , qui constitue l'équation d'état a été choisie comme étant :

$$P = k(\rho - \rho_0)$$

Les forces générées chercheront ainsi à rétablir la densité  $\rho_0$  d'équilibre en créant une pression là où  $\rho$  est différent de  $\rho_0$ , pression réglable au travers du paramètre  $k$ .

Restent à choisir la forme du filtre de lissage  $W_h$  employé (qui modifiera l'animation), à ajouter des forces de dissipation visqueuses (pour stabiliser) et à faire en sorte que les forces créées satisfassent le principe d'action-réaction (voir [Des97]). On peut alors envisager la simulation à différentes échelles du matériau.

## 5.3 Simulation multirésolution des SPH

Ajouter ou supprimer des points d'échantillonnage est maintenant possible, tout en continuant d'approcher un comportement donné, qui est celui dicté par l'équation d'état. En fonction de seuils de continuité, Desbrun propose de diviser ou regrouper plusieurs particules lors de la simulation, optimisant ainsi l'utilisation du temps de calcul (Fig 1.5) [Des97, DC99b].

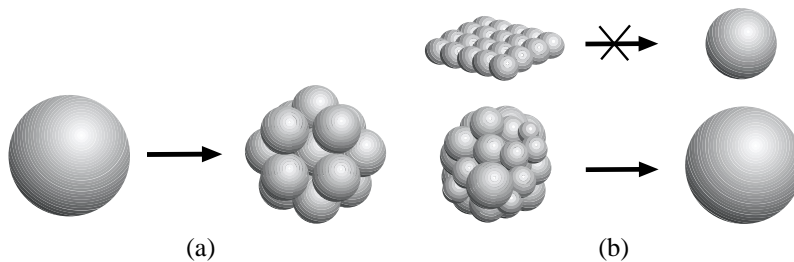


FIG. 1.5: Chaque particule peut se diviser en plusieurs autres (a). Le groupement est possible lorsque la forme est globalement sphérique (b).

Le pas de temps d'intégration peut également être modifié au cours de la simulation en fonction de critères physiques liés à la vitesse de propagation d'une onde dans le matériau<sup>4</sup>.

Hernquist et Katz introduisent la notion de *Tree SPH*, ou SPH hiérarchique [HK89]. Le partitionnement de l'espace en un arbre hiérarchique permet en effet, comme pour tout système de particules, d'accélérer les calculs de voisinage et d'influence. Cette découpe de l'espace n'est pas liée à une grille Eulérienne mais est Lagrangienne tout comme les SPH, et évolue avec la simulation. L'application est la simulation de très gros systèmes auto-gravitationnels en astrophysique.

<sup>4</sup>Voir l'annexe B.



## 5.4 Conclusion

Ce modèle possède beaucoup des avantages intrinsèques des modèles à base de particules décrits précédemment. À chaque instant, on est capable de calculer en un temps équivalent à celui d'un système de particules la force subie par chaque point pour ensuite l'intégrer au cours du temps. La différence est que le calcul plus subtil de la force en utilisant le formalisme SPH garantit le respect d'une équation d'état, autorisant un processus multirésolution. Dils a montré l'équivalence des SPH avec la méthode d'éléments finis de Galerkin appliquée en temps et en espace [Dil96].

On pourra toutefois reprocher aux SPH d'être mieux adaptées à la simulation d'un comportement gazeux qu'à celle d'un véritable matériau élastique. Augmenter la cohésion interne du matériau peut se faire en augmentant le rayon d'influence  $h$  du noyau  $W_h$  et le nombre de particules, au prix de calculs plus coûteux. Desbrun annonce ainsi des temps de calculs de plusieurs heures lors de la simulation de matériaux relativement peu rigides [DC99b].

## 6 Les modèles continus

Tout comme le formalisme SPH, les modèles décrits ici et dans la section suivante considèrent avant tout la matière comme un *milieu continu*, soumis à des *lois de comportement* globales. L'objet simulé ne sera finalement discrétisé que pour les besoins de la simulation informatique.

On va chercher à mesurer l'énergie potentielle de l'objet sous l'action de contraintes extérieures données. Cette énergie peut être calculée en fonction d'une variable qui décrit les déformations de l'objet. Elle correspond au *travail*<sup>5</sup> qu'il a fallu fournir pour amener l'objet dans cette position déformée.

Le travail étant l'intégrale du produit force  $\times$  déplacement, il est nécessaire de connaître les forces créés par la déformation pour le calculer et c'est une *loi de comportement* qui va donner l'expression de la force en fonction de la déformation, en faisant intervenir des coefficients propres au matériau simulé. Tout ce formalisme sera décrit plus en détails dans le chapitre suivant et il suffit ici de retenir que l'on peut calculer l'énergie potentielle en fonction des déformations.

### 6.1 Minimisation de l'énergie de déformation

Cette énergie potentielle de déformation a été directement utilisée dans une série d'approches. Terzopoulos *et al.* ont dès 1987 proposé un formalisme dans lequel une courbe, une surface ou un objet 3D se déforme afin de minimiser leur énergie de déformation [TPBF87]. Celle-ci est mesurée comme étant la norme d'une matrice, appelée *tenseur métrique* et qui est en fait pratiquement le tenseur des déformations de *Green-Lagrange* (dont on reparlera dans le prochain chapitre).

Les différentielles impliquées dans le calcul de l'énergie potentielle sont approximées par des différences finies. L'équation différentielle dynamique est intégrée dans le temps en utilisant une intégration *implicite*<sup>6</sup>. Des termes annexes, du même type, doivent être introduits dans l'évaluation de l'énergie potentielle pour des simulations d'objets 2D et 1D (voir [TPBF87, WFB87, PB88]). La méthode reste très lente car nécessitant une inversion matricielle à chaque pas de temps.

Le même type d'approche a été utilisé dans la méthode des *snakes*, ou *contours actifs* de [KWT87] dans un contexte d'analyse d'image et de recherche de contour. L'énergie de déformation mesure alors la courbure du contour, qui cherche à entourer les zones d'intérêt de l'image ou du volume.

L'approche précédente fut améliorée l'année suivante en définissant l'objet comme la somme de deux composantes, l'une rigide, dite *de référence*, et l'autre déformable attachée à la première [TW88]. Ce modèle hybride permet de simplifier l'expression de l'énergie potentielle qui n'a plus à gérer les transformations rigides (translation et rotation globales). Dans l'hypothèse des petites déformations, on peut même considérer cette matrice comme constante et l'inverser une fois pour toutes au début. Des forces de pénalité générées par les obstacles viennent ensuite déformer l'objet.

Dans [TF88], la composante de référence n'est plus rigide et peut "absorber" une partie des déformations, conduisant à des objets *plastiques*. Cette composante peut également subir des forces de frottement visqueuses

<sup>5</sup>Au sens physique, propre et mathématique du terme.

<sup>6</sup>Voir annexe B.

et les auteurs décrivent la possibilité de fractures en des points préétablis du maillage en cas de trop forte déformation.

## 6.2 Conclusion

Ces méthodes, assez anciennes, ont permis les premières animations de matériaux déformables. Leur inconvénient majeur est leur coût de calcul élevé dû à l'inversion matricielle qu'ils requièrent, ce qui les rend incompatibles avec une application temps-réel.

L'utilisation des différences finies dans [TPBF87] limite le champ d'utilisation à des maillages réguliers, les objets ayant forcément une forme liée à la grille utilisée. La méthode des éléments finis que nous décrivons dans la section suivante est en fait une généralisation de ces méthodes, basées sur une vision continue de la matière. On va maintenant être capable d'évaluer les équations différentielles sur un maillage arbitraire composé d'éléments.

# 7 Les éléments finis

Les éléments finis sont *la* méthode de prédilection des physiciens pour simuler un matériau déformable. Indispensables dans l'industrie et faisant l'objet de publications en mécanique et en mathématiques appliquées depuis des dizaines d'années, ils utilisent également le modèle continu décrit dans la section précédente.

La littérature sur le sujet est souvent trop pointue alors que le résultat visuel recherché en animation ne demande pas un modèle si précis. Elle est de ce fait d'une compréhension difficile, la méthode générale pouvant être expliquée plus simplement.

Cette technique peut également sembler être réservée à l'étude des positions d'équilibre *statiques* et non à l'évolution *dynamique* des objets au cours du temps, les calculs impliqués étant qui plus est souvent très longs. Toutes ces raisons expliquent le relativement faible engouement pour la méthode dans le domaine de l'image de synthèse, alors qu'elle a d'évidents avantages, comme nous allons essayer de le montrer dans cette section.

Cette description est très simplifiée et l'on se reportera à [Seg84, Bat96, Kas97] ou [ZT91] pour plus de détails.

## 7.1 Principe

On part ici aussi de l'expression de l'énergie potentielle de déformation de l'objet. La position d'équilibre recherchée sera celle qui *minimise* cette énergie, et donc où la dérivée de l'énergie par rapport à la position s'annule. Le but est donc de trouver la valeur de la déformation qui satisfait cette équation différentielle d'équilibre.

Chercher la valeur de la déformation comme une fonction *continue* dans le matériau signifie résoudre analytiquement cette équation, ce qu'on ne sait faire que dans des cas très simples. C'est donc ici qu'intervient la discrétisation en éléments finis.

### Discrétisation en éléments

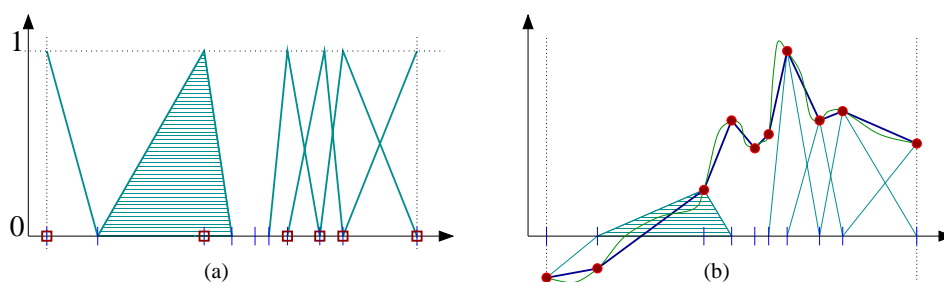
La fonction continue recherchée va être représentée par une approximation discrète. On va découper l'objet en *éléments*, sortes de cellules de matière possédant chacune un faible nombre de variables qui décriront l'allure de la fonction à l'intérieur de l'élément. Ces variables seront en fait assignées à chacun des *sommets* de l'élément, chacun décrivant plus précisément par ses variables ce qui se passe autour de lui dans l'élément. La fonction étant entièrement décrite par ces quelques coefficients, on restreint grandement l'espace des fonctions que l'on peut représenter.

Des raisons de continuité sur la fonction recherchée vont faire que l'on va souvent imposer à deux éléments juxtaposés d'avoir les mêmes variables aux sommets qu'ils partagent. Cela assure que l'on n'aura pas de discontinuité de la fonction quand on l'exprimera d'un côté et de l'autre de la frontière entre les éléments. La fonction est donc maintenant représentée par un ensemble de variables, définies en chaque sommet des éléments créés.

La description de la fonction par ces variables se fait au travers des *fonctions de base*. Elles décrivent pour chaque sommet comment les variables qu'il contient influencent la fonction générée. Ces fonctions de base seront généralement de la même forme pour tous les sommets et de leur complexité dépendra l'espace des fonctions représentables et donc la qualité de la solution.

Soyons concrets. La fonction cherchée sera le plus souvent approximée par une *fonction linéaire par morceaux*, linéaire sur chacun des éléments (plus rarement par un polynôme de plus haut degré). Pour la définir il suffit d'une valeur à chaque sommet<sup>7</sup> et les fonctions de base sont alors simplement les fonctions linéaires par morceaux qui valent 1 au sommet considéré et 0 sur tous les autres sommets (des fonctions triangle en 1D). Toute fonction linéaire par morceaux sur les éléments peut en effet être représentée comme une somme pondérée de ces fonctions de base, les poids associés à chaque sommet étant les valeurs prises par la fonction en ces sommets. Ce seront les variables qui décrivent la fonction.

La Figure 1.6 explicite cette représentation en une dimension. Chaque élément est un segment de l'axe des abscisses, sur lequel les fonctions de base ont une forme triangulaire (Fig. 1.6a). La fonction continue est approximée par une fonction linéaire par morceaux, somme pondérée de ces fonctions triangles (Fig. 1.6b).



**FIG. 1.6:** Les fonctions de base ont une forme triangulaire en 1D (seules celles associées aux sommets encadrés sont représentées) (a). Leur somme pondérée crée la fonction linéaire par morceaux qui approxime la fonction continue (b).

Noter que la forme des éléments (leur longueur en 1 dimension) n'a pas à être uniforme et qu'il est intéressant de l'adapter à la courbe approximée comme c'est le cas ici. Cela demande de connaître *a priori* l'allure de la solution cherchée.

En deux dimensions, sur un maillage d'éléments triangulaire, les fonctions de base auront la forme de pyramides s'appuyant sur les frontières du maillage.

En résumé, on a par passage aux éléments finis représenté la fonction cherchée (les déformations de l'objet) par un ensemble de variables aux sommets du maillage.

### Résolution du système

L'équation d'équilibre du système doit être vérifiée sur chacun des éléments, ce qui va se traduire par des équations sur chacun des sommets. Elle comporte des dérivées partielles qui s'expriment analytiquement en chaque sommet comme une combinaison linéaire des valeurs prises par la fonction sur les autres sommets.

En chaque sommet, on peut donc écrire une équation linéaire faisant intervenir la valeur de la fonction dans cette zone (définie par la valeur en ce sommet et aux sommets proches). En pratique, chaque sommet satisfait une, deux ou trois équations selon que l'on est en 1, 2 ou 3 dimensions, car chaque sommet comporte alors les 1, 2 ou 3 composantes de la fonction déformation, qui chacune doit avoir une dérivée nulle par rapport à la position du sommet.

En regroupant toutes ces équations linéaires dans une matrice on obtient un système matriciel de  $n$  (resp.  $2n$ ,  $3n$ ) équations en les  $n$  (resp.  $2n$ ,  $3n$ ) inconnues que sont les valeurs de la fonction aux sommets du maillage :

$$K\mathbf{U} = \mathbf{F} \quad (1.8)$$

où  $K$ , appelée *matrice de rigidité* du système, et où  $\mathbf{U}$  et  $\mathbf{F}$  sont les vecteurs rassemblant respectivement les déplacements et les forces externes appliquées aux  $n$  sommets.

Il ne reste qu'à résoudre ce système pour obtenir l'approximation de la fonction déformation sur le maillage, dont on pourra ensuite par exemple déduire les contraintes internes grâce à la loi de comportement.

<sup>7</sup>Si l'on est en dimension 1. En dimension 3 la fonction cherchée a 3 composantes et sera donc représentée par 3 fonctions linéaires  $\mathbf{U}_x$ ,  $\mathbf{U}_y$  et  $\mathbf{U}_z$ . On aura alors 3 inconnues en chaque sommet.

En pratique, puisque seules les valeurs prises sur les sommets les plus proches interviennent dans l'expression des dérivées<sup>8</sup>, chaque équation n'impliquera qu'un petit nombre de sommets. La matrice  $K$  née de leur regroupement aura donc une faible largeur de bande, des méthodes adaptées pouvant être utilisées pour la résolution. Noter aussi que le conditionnement de la matrice<sup>9</sup> dépend directement des coefficients de raideur du matériau : plus il est rigide, plus l'inverse est difficile à calculer.

En chaque sommet la déformation et la force appliquée (généralement nulle) peuvent être connues ou être des variables du système. Rien n'empêche d'imposer une déformation à un sommet (s'il est en contact avec un autre objet par exemple) et d'y calculer la force créée, tout comme on peut lui faire subir une force donnée (nulle à la surface du matériau sauf en quelques points par exemple) et chercher la déformation résultante. On peut donc échanger les inconnues et les données du problème dans les valeurs de  $\mathbf{U}$  et  $\mathbf{F}$ , et modifier alors le système matriciel  $K$  pour ne plus avoir que des inconnues d'un côté et des données de l'autre.

### Simulation dynamique

En animation, on ne cherche généralement pas à trouver les positions d'équilibre d'un objet mais plutôt ses déformations au cours du temps. Il faut alors prendre en compte les inerties et les forces de frottement, ce qui transforme l'Équation 1.8 en une équation différentielle du second ordre classique :

$$M \frac{\partial^2 \mathbf{U}}{\partial t^2} + D \frac{\partial \mathbf{U}}{\partial t} + K \mathbf{U} = \mathbf{F} \quad (1.9)$$

où  $M$  et  $D$  sont des matrices liées à la masse et aux coefficients d'amortissement. Elles sont obtenues, tout comme  $K$ , en regroupant les équations de chaque sommet. La masse d'un sommet est ainsi calculée comme l'intégrale de la fonction densité au travers des fonctions de base (de même pour  $D$  obtenue à partir des coefficients de frottement de chaque élément).

Les différentes matrices  $M$ ,  $D$  et  $K$  peuvent dépendre de la géométrie des éléments et doivent donc être recalculées dès que l'objet se déforme, ce qui est coûteux. Cozot désigne par le terme de lagrangien *libre* de telles méthodes dans lesquelles un remaillage de l'objet suit chaque pas de simulation fait par éléments finis [Coz96].

On considère généralement que l'objet se déforme suffisamment peu pour considérer ces matrices comme constantes. Cette approximation signifie que l'on approxime un comportement purement linéaire de l'objet, ce qui n'est généralement vrai que pour les petites déformations. Il faudrait au delà d'une certaine déformation modifier les coefficients pour obtenir un comportement plus réaliste. Considérer ces valeurs comme constantes n'est néanmoins pas absurde, l'interaction entre deux points proches étant conditionnée par leur disposition géométrique (le nombre d'atomes les séparant), qui ne changera pas au cours du temps.

## 7.2 Applications

### Méthode classique

L'une des premières application des éléments finis à l'animation fut faite par Gourret *et al.* dans une simulation de contact entre une main et une balle déformables. La réponse aux collisions est quelque peu empirique et la modélisation des doigts simple, mais la simulation permet une déformation complexe [GMTT89]. Les matrices sont remises-à-jour quand les articulations des doigts ont bougé de plus de dix degrés. La simulation est longue à calculer et la dynamique en a été supprimée pour ne pas la ralentir davantage.

Chronologiquement vinrent ensuite Collier *et al.* avec un modèleur de tissus utilisant des éléments carrés en deux dimensions, inspirés de ceux utilisés pour simuler les membranes dans l'industrie [CCOS91].

Chen et Zeltzer cherchèrent quand à eux à animer les muscles humains [CZ92]. Leurs éléments sont très complexes (20 nœuds chacun) et il n'en utilisent que 2 par muscle. Ceux-ci se voient imposer des forces par l'intermédiaire des tendons qui les relient aux os. Bien qu'utilisant les principes de la dynamique modale décrits dans la Section 2.1, leur simulation de quelques dizaines de nœuds est très lente.

Bro-Nielsen et Cotin utilisèrent une idée intéressante en 1996 en remarquant tout simplement que seul le déplacement des nœuds de la surface est pertinent visuellement [BNC96]. Le système matriciel de l'Équation

<sup>8</sup>Les sommets ont une influence sur la fonction limitée au support de leur fonction de base. Avec des fonctions linéaires, l'expression des dérivées en un sommet ne fera intervenir que les sommets voisins qui lui sont directement liés par un élément.

<sup>9</sup>Sa facilité à être inversée.

1.8 peut être réécrit en séparant ces nœuds de ceux de l'intérieur. Cette méthode, dénommée *condensation* par les auteurs, aboutit à un système matriciel qui ne concerne plus que les nœuds de la surface, donc de taille bien moindre, mais qui a en contrepartie perdu sa structure de matrice creuse. À noter que cette manipulation, classique chez les physiciens, avait déjà été appliquée dans [GM89].

Le nouveau système, de type  $K\tilde{U} = \tilde{F}$  est alors pré-inversé, afin de pouvoir déterminer rapidement les déformations en fonction des forces par

$$\tilde{U} = K^{-1}\tilde{F}$$

On déformera l'objet en appliquant des forces non nulles à quelques endroits (seules quelques colonnes de  $K^{-1}$  seront donc à prendre en compte, ce qui accélère beaucoup le produit), et en en déduisant très rapidement les déformations générées. On obtient ainsi une position d'équilibre *statique* de l'objet. L'inconvénient de piloter les déformations en appliquant des forces et non des déplacements aux nœuds est qu'on ne peut alors garantir la forme des déformations à l'endroit où l'on voudrait appliquer un outil. Celui-ci peut ne pas repousser assez l'objet (si les forces appliquées sont trop faibles) ou au contraire sembler agir à distance (forces trop fortes), ce qui pose évidemment problème. On doit donc ici se mettre à la place du modèle physique et appliquer exactement les forces que celui-ci aurait générées pour avoir une simulation correcte, ce qui est impossible. On gagne par contre énormément en temps de calcul en imposant ainsi des forces arbitraires car la matrice peut-être pré-inversée une fois pour toutes.

Dans les applications classiques, l'outil va imposer une déformation à l'objet, les forces générées à ces nœuds devenant alors les inconnues du système. Si les points sur lesquels on agit ne sont pas toujours les mêmes au cours de la simulation, ce qui est le cas, le système matriciel change tout le temps et ne peut donc être ainsi pré-inversé.

On peut par contre effectuer une simulation dynamique, l'équation  $M\frac{\partial^2\mathbf{U}}{\partial t^2} + D\frac{\partial\mathbf{U}}{\partial t} + K\mathbf{U} = \mathbf{F}$  pouvant se réécrire à l'aide de différences finies pour calculer les dérivées temporelles. On obtient alors un système de la forme

$$K'\tilde{\mathbf{u}}_t = \tilde{\mathbf{f}}_{(t, t-1, t-2)}$$

où le vecteur  $\tilde{\mathbf{f}}$  dépend maintenant des déformations aux deux instants précédents. Il est plus cher à calculer et n'a plus comme précédemment quasiment toutes ses valeurs nulles ce qui ralentit beaucoup le produit matriciel avec  $(K')^{-1}$ .

Dans le même ordre d'idée, James et Pai [JP99] ont tout récemment proposé une méthode basée sur le même principe, mais dans laquelle ils *mettaient à jour* l'inverse de la matrice  $K$ . Lorsque l'utilisateur manipule un outil virtuel pour déformer l'objet, peu de nœuds passent du statut de déplacé à celui de libre, et inversement. Les auteurs ont alors fait remarquer que la matrice inverse n'était en fait que peu modifiée dans ces cas. L'influence d'un point étant limitée à une colonne de la matrice, une astuce mathématique permet de la remettre à jour rapidement à l'aide de lourds précalculs, l'animation restant temps-réel.

Dans sa thèse [Cot97], Cotin a proposé une méthode imposant cette fois-ci une contrainte en *déplacement* sur l'objet. Il utilise l'intéressante propriété de *superposition* des éléments finis en *élasticité linéaire*, qui dit que la déformation de l'objet résultant du déplacement  $\mathbf{d}$  d'un point est la somme de celles qui seraient produites par des déplacements  $d_x$ ,  $d_y$  et  $d_z$  appliqués séparément dans chacune des 3 directions  $x$ ,  $y$  et  $z$ . Ce résultat est dû au caractère linéaire du système d'équations. Qui plus est, un déplacement double va simplement créer une déformation double<sup>10</sup>.

Il suffit alors de précalculer, pour chaque point séparément, quelle va être la déformation de tout l'objet lorsque ce point est déplacé dans les trois directions. Ces informations, filtrées pour supprimer les déplacements trop faibles et économiser ainsi la mémoire, sont stockées dans chaque point *de la surface*, conduisant à une animation temps-réel [CDA99, CDA96, CDA98].

Le problème est qu'on ne peut connaître la déformation créée lorsqu'on manipule plus d'un point à la fois (ce qui est généralement le cas). Appliquer en les sommant chacune des déformations créées par les points déplacés va conduire à une déformation exagérée. Cotin n'en applique alors qu'une partie, en représentant dans une matrice l'influence mutuelle des nœuds déplacés, dont l'inverse va déterminer les déplacements à appliquer aux différents nœuds. On peut ainsi déplacer plusieurs sommets, mais au prix de l'inverse d'une matrice  $3m \times 3m$  où  $m$  est le nombre de nœuds déplacés.

<sup>10</sup>Ce comportement n'est pas très réaliste, mais la linéarité est une première approximation de la complexité des mouvements réels.

L'un des défauts de ces trois dernières méthodes ([BNC96, JP99, Cot97]) est que toute modification de la topologie de l'objet (une simple entaille ou une découpe plus profonde) est impossible car elle remettrait en cause toutes les matrices précalculées qui ne seraient plus adaptées. Mais leur inconvénient majeur est qu'elles sont *statiques*, ne donnant qu'une suite d'états d'équilibre de l'objet. C'est en particulier sensible lorsque l'outil quitte brusquement l'objet et que celui-ci revient alors immédiatement dans sa position d'équilibre sans aucune oscillation dynamique.

### Éléments finis explicites

Toutes les méthodes décrites jusqu'à maintenant font usage d'éléments finis *implicites* et nécessitent l'inversion d'un système global. Les éléments finis *explicites*, au contraire s'apparentent plus à des systèmes de particules et profitent de leur flexibilité.

Le principe est le suivant : lorsque, sur chacun des éléments du système, on a écrit les équations d'équilibre, on est capable d'exprimer la force que subit chaque sommet en fonction des déplacements des sommets voisins. Plutôt que de chercher à trouver la position d'équilibre que ces forces vont créer en résolvant le système matriciel, on va simplement intégrer ces forces pour chaque point. On obtient ainsi de nouvelles positions dans lesquelles on va de nouveau calculer des forces, etc. Il suffit d'utiliser les lois de la dynamique du point pour créer l'animation, ce qui est plus simple que d'essayer de résoudre l'équation du second ordre 1.9.

On perd ici en précision ce que l'on va gagner en vitesse. La résolution globale du système garantissait que les positions trouvées étaient toutes compatibles entre elles, satisfaisant globalement toutes les équations. On n'a plus ici cette cohérence, mais plutôt une juxtaposition de solutions locales, bien plus rapides à calculer. On introduit par contre un retard dans la propagation, chaque point ne réagissant qu'à la position de ses voisins à l'instant *précédent*.

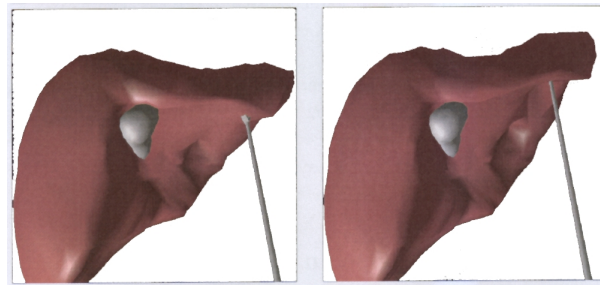


FIG. 1.7: Simulation de laparoscopie hépatique utilisant les masses-tenseurs.

Cette méthode a été utilisée dans deux contributions récentes de la littérature. De part sa ressemblance avec les masses-ressorts (on calcule une force en chaque point d'après ses voisins puis on l'intègre), Cotin a dans sa thèse dénommé cette méthode les *masses-tenseurs*, le calcul de la force pouvant s'exprimer par un tenseur<sup>11</sup> (voir Fig 1.7) [Cot97, DCA99].

Plus récemment, O'Brien et Hodgins ont proposé une méthode simulant les fractures et leur propagation à l'intérieur d'un matériau en utilisant également eux aussi une méthode explicite [OH99]. La discrétisation du modèle en éléments finis évoluait au cours du temps pour suivre l'apparition des craquelures dans le matériau. Celles-ci apparaissaient à l'endroit où les contraintes internes dépassaient un seuil donné.

Ces deux approches sont bien plus rapides que celles nécessitant une inversion globale, même si dans le cas de [OH99] les très fortes raideurs des matériaux employés demandaient un très faible pas de temps, nécessaire également au suivi de l'évolution des fractures. Cotin propose ainsi un simulateur chirurgical interactif.

## 7.3 Conclusion

Une fois le formalisme compris (on n'a en particulier pas décrit ici la façon dont sont calculées les dérivées partielles sur les éléments), utiliser les éléments finis revient dans la méthode classique à résoudre un gros système matriciel. Trop lente, cette opération interdit les applications temps-réel.

Pré-inverser les matrices en les supposant constantes au cours du temps limite les applications à de faibles déformations. Cela demande de fixer une fois pour toutes le statut (déplacé par l'outil ou libre) de chacun des

<sup>11</sup>On peut voir un tenseur comme une matrice symétrique.

points. Il faut employer des mathématiques subtiles pour pouvoir comme dans [JP99] mettre à jour cette inverse en fonction du statut de chaque point et pouvoir ainsi déplacer des points arbitraires avec un outil. Encore dans ce cas là n'obtient-on que des états d'*équilibre* du système, la dynamique de l'animation faisant cruellement défaut.

Les éléments finis explicites sont quand à eux bien plus rapides, et devraient permettre le temps-réel. Ils procurent de meilleures conditions que les masses-ressorts, le temps de calcul de la force en chaque point pouvant en effet être du même ordre de grandeur alors que moins de points simulés suffisent à obtenir le même résultat visuel.

Les changements de topologie sont impossibles avec les éléments finis implicites, mais mis en œuvre par Cotin avec une méthode explicite. Le système matriciel doit en effet alors être complètement recalculé, les interactions entre points étant modifiées. Ce formalisme n'est en outre pas approprié à la simulation de matériaux trop déformables. Les modèles physiques décrivant les matériaux sont faits pour des métaux et tolèrent une déformation de l'ordre de 1%, incompatible avec les tissus biologiques.

On peut considérer les matrices comme constantes pour de très faibles déformations, mais il faudrait pour des matériaux biologiques les remettre à jour régulièrement. Les résultats de Cotin avec une méthode explicite semblent néanmoins montrer que l'on doit pouvoir dépasser ces limites, aux dépens du réalisme physique de la simulation.

## 8 Modèles à couches

### 8.1 Principe

Les modèles que nous allons maintenant décrire vont essayer de mélanger plusieurs des approches présentées pour s'en enrichir en tirant parti de leurs avantages respectifs. Tout comme les modèles de Terzopoulos vu précédemment [TW88, TPF89], c'est généralement des *couches* constituées de différents modèles qui vont être superposées.

Dès 1976, Burtnik [BW76] animait le squelette d'un personnage et en déduisait le mouvement de la couche de muscles et de peau qui y était attaché, le travail de l'animateur s'en trouvant simplifié. La méthode fut améliorée par Chadwick *et al.* [CP88, CHP89] par la séparation des muscles, animés par une FFD contrôlée par un réseau masses-ressorts, et de la peau purement géométrique. Ce type de modèle est repris par Cani-Gascuel *et al.* dans [GVP90, GVP91, Gas90], les interactions entre les couches étant plus complexes. [CGD97] reprendra certaines des idées du modèle de [GVP91] en remplaçant la surface spline par une surface implicite.

Le modèle de Turner [TT93, Tur95] d'animation de personnages mélange une animation cinématique du squelette par cinématique inverse, des muscles modélisés par objets implicites attachés au squelette par des ressorts, et enfin une peau sur laquelle se concentrent les calculs et qui utilise la méthode de [TPBF87]. Shen et Thalmann [ST95] se concentreront sur la modélisation des muscles par *metaballs*, la peau purement géométrique étant constituée de B-splines.

Citons également en 1992 les travaux de Metaxas et Terzopoulos [MT92] qui reprennent ceux de [TF88] et [WW90] dans des simulations utilisant plusieurs couches.

### 8.2 Conclusion

Ces modèles sont souvent rapides, car ils décomposent le problème et simulent avec une méthode adaptée chaque partie. Ils offrent ainsi un paramétrage séparé de chaque couche, ce qui peut être intéressant car parfois plus intuitif.

Ce genre d'approches se rapproche de la multirésolution. Celle-ci ne consiste néanmoins pas simplement en un empilement de couches. On doit plutôt la voir comme une façon de générer automatiquement des couches pouvant interagir et se modifier. Or les méthodes proposées offrent très rarement une interaction réelle entre les couches, chacune se contentant plutôt de contrôler seulement la suivante. Dans [GVP91, CGD97] ou [TW88] l'interaction se fait dans les deux sens, les déformations de la peau pouvant à leur tour influencer le modèle interne.

Les sections précédentes classaient les modèles par grandes familles de méthode. Nous allons maintenant dépasser ce classement et nous intéresser à deux types d'algorithmes, plus spécifiquement visés par cette thèse : ceux autorisant une simulation interactive et ceux tirant parti de la multirésolution.

## 9 Applications interactives

Cette section détaille quelles sont, parmi les méthodes présentées, celles qui permettent une véritable animation interactive. On ne parle pas ici de *temps-réel* car nous réserverons ce terme à des applications bien précises (voir Sec. 11.2). Les applications présentées permettent néanmoins une réaction interactive du matériau aux sollicitations de l'utilisateur, accompagnées d'un affichage de plusieurs images par seconde.

L'application typique des modèles déformables temps-réel est le simulateur médical. Très utile pour la formation des chirurgiens, il est actuellement sujet de recherche pour de grands groupes industriels :

MUSE	<a href="http://www.musetech.com/html/muse2000/apps/medical.html">www.musetech.com/html/muse2000/apps/medical.html</a>
HT Medical System	<a href="http://www.ht.com">www.ht.com</a>
Teneo	<a href="http://www.teneocomp.com">www.teneocomp.com</a>

et universitaires :

MIT	<a href="http://touchlab.mit.edu">touchlab.mit.edu</a>
Penn State	<a href="http://cs.millersv.edu/haptics">cs.millersv.edu/haptics</a>
EPFL	<a href="http://imt.dmt.epfl.ch">imt.dmt.epfl.ch</a>
Karlsruhe	<a href="http://www-kismet.iai.fzk.de/TRAINER/mic_trainer1.html">www-kismet.iai.fzk.de/TRAINER/mic_trainer1.html</a>
LIFL	<a href="http://www.lifl.fr/GRAPHIX/">www.lifl.fr/GRAPHIX/</a>
AISIM	<a href="http://www-sop.inria.fr/epidaure/AISIM/">www-sop.inria.fr/epidaure/AISIM/</a> [INR]

Les simulateurs existant sur le marché se contentent néanmoins généralement d'animer des objets *rigides* et apprennent principalement au chirurgien le maniement des outils dans l'espace.

De par leur simplicité et leur rapidité, les masses-ressorts ont été largement utilisés, notamment par Meure [MC96a, MC96b, Mes97] et Cover *et al.* [CEO93]. Ces approches utilisent peu de ressorts pour assurer l'interactivité et ajoutent des ressorts de longueur à vide nulle entre chaque point et la position qu'il occupait dans sa position de repos. Il définissent ainsi une forme au repos vers laquelle la surface va chercher à revenir. Sans cette surface, le maillage purement surfacique de l'objet n'aurait aucune consistance *volumique* et pourrait être aplati sans résister.

Basdogan *et al.* [BHS<sup>+</sup>98] fournissent un retour d'effort à l'utilisateur au travers de l'utilisation d'un *Phantom* [Tec] en appliquant le même genre de méthode. La force calculée est celle subie par quelques ressorts attachés à un objet fixe et simulant le comportement élastique très local de l'objet, autorisant ainsi des calculs à 1000Hz nécessaires au retour d'effort. L'affichage du modèle utilise localement des courbes splines et globalement une méthode FFD.

Citons également l'utilisation temps-réel des masses-ressorts par Desbrun dans [DSB99] qui grâce à une méthode d'intégration semi-implicite, interprétation simplifiée de [BW98], propose une animation de tissus très rapide<sup>12</sup>.

L'algorithme de *ChainMail* de Gibson [Gib97] modélise l'objet comme un ensemble de maillons ne pouvant ni être dissociés ni être trop imbriqués (Fig. 1.8). À chaque déplacement imposé, un algorithme itératif déplace tous les maillons de façon à satisfaire ces contraintes. Purement géométrique, cette déformation est très rapide : elle est accompagnée d'une relaxation par masses-ressorts, faite uniquement lorsque l'on en a le temps, c'est-à-dire quand on ne manipule plus l'objet. L'intérêt de cette combinaison est que les déformations purement plastiques du réseau de maillons sont très rapides (125000 maillons déplacés à 3Hz sur une SGI Indy).

Parmi les méthodes utilisant les éléments finis, les travaux de Bro-Nielsen [BNC96] et Cotin [Cot97] ainsi que ceux de James et Pai [JP99] offrent une interface interactive. Notons qu'aucun de ces modèles n'est réellement dynamique, les masses-tenseurs de Cotin [Cot97] réinitialisant la vitesse à 0 avant chaque pas de simulation et les deux autres ne proposant que des états d'équilibre statiques.

<sup>12</sup>Voir Annexe B



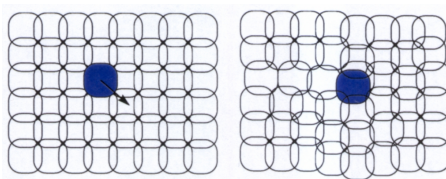


FIG. 1.8: Une juxtaposition de maillons imbriqués forme le *ChainMail*.

## 10 La multirésolution en animation

### 10.1 Plusieurs façons de faire de la multirésolution

Définissons plus précisément ce que l'on entend par multirésolution en animation. Il ne s'agit pas simplement de pouvoir avoir une résolution variable au cours de l'animation, comme c'est le cas dans [HPH96] ou [OH99] qui font évoluer la discrétisation de leur système en fonction du lieu des plis d'un tissu ou de la propagation de fractures.

Multirésolution signifie davantage avoir à *tout instant* à disposition différentes représentations du même mouvement et choisir en permanence celle qui convient le mieux, ce choix pouvant éventuellement dépendre de la partie de l'objet concernée. On se donne la possibilité de générer plusieurs mouvements, à des qualités et des vitesses de calcul différentes, pour n'utiliser à chaque instant que celui offrant le meilleur compromis.

Ces différentes représentations doivent néanmoins cohabiter suffisamment pour que l'utilisateur n'ait pas conscience de leur présence. Il doit avoir l'impression que le comportement reste identique au cours de la simulation, indépendamment des résolutions qui cohabitent et de leur possibles modifications.

Les différents niveaux de détail peuvent être établis au départ ou être calculés lors de l'animation en fonction de ce qui s'y déroule. Ce dernier cas est plus complexe et risqué, s'il prend trop de temps, de figer l'animation, ce qui est gênant dans un contexte interactif.

Ces différentes résolutions peuvent également être du même type de simulation, vues à des résolutions différentes, ou mélanger différentes techniques d'animation (dynamique, cinématique et ponctuelle par exemple dans [CH97]).

Les divers seuils qui décideront quelle résolution doit être utilisée à chaque endroit permettront de régler la vitesse de la simulation en autorisant très ponctuellement ou au contraire assez largement l'utilisation des résolutions les plus fines. Dans une approche de temps-réel, on pourra également *garantir* un délai fixe entre deux affichages en limitant l'utilisation des résolutions les plus coûteuses en fonction du temps de calcul disponible.

### 10.2 Difficultés

Il y a toujours deux problèmes principaux dans la mise en place des approches multirésolution :

- Les différentes résolutions que l'on veut mélanger doivent avoir, dans une certaine mesure, le *même comportement dynamique*, c'est-à-dire approximer le même mouvement au cours du temps. Si tel n'était pas le cas, la simulation serait chaotique, son comportement dépendant des résolutions employées alors que c'est précisément ce que l'on veut cacher à l'utilisateur.
- Les différentes résolutions doivent *pouvoir cohabiter*, c'est-à-dire que les *transitions* entre les niveaux ne doivent pas être visibles. Ces transitions sont d'une grande importance et de leur discrétion dépend le résultat. Cette notion est à différencier de la précédente, car même avec un comportement dynamique identique, deux résolutions ne peuvent pas donner un bon résultat si elles ne cohabitent pas. Deux mouvements, même similaires, doivent par exemple être *en phase* à l'endroit où l'on passe de l'un à l'autre. On devra aussi veiller à éviter les clignotements dues à une oscillation entre deux résolutions ainsi que les apparitions soudaines de détails qui ne faisaient pas partie de la résolution grossière précédente, ou tout autre indice visible dépendant de l'application.

Dans notre cas, simuler une série de points, chacun à sa propre échelle est aisé, mais encore faut-il que ces points puissent interagir les uns avec les autres lorsqu'ils n'appartiennent pas à la même échelle, afin que l'objet forme un tout et non une simple juxtaposition de parties à différentes résolutions.

### 10.3 Adaptatif versus Hiérarchique

Pour une même zone de l'objet, les différentes résolutions qui peuvent y être employées vont l'être séparément ou collectivement :

- Ou bien une résolution fine sera la seule réellement active dans cette zone et elle devra alors être capable de cohabiter avec les zones voisines qui n'ont pas forcément la même résolution.
- Ou bien cette résolution cohabitera avec les autres résolutions de cette même zone, en leur redonnant par exemple par moyenne les informations dont elle dispose. Un certain nombre de résolutions seront alors actives en même temps dans cette zone pour représenter le phénomène.

Nous désignerons les premières méthodes par le terme d'*adaptatives* (Fig. 1.9a) et les secondes par celui de *hiérarchiques* (Fig. 1.9b). L'adaptatif n'utilisera qu'une résolution à un endroit donné, devant l'interfacer avec les zones voisines. Le but sera alors d'*adapter* en chaque zone la résolution en fonction de la simulation.

Le hiérarchique devra par contre gérer plusieurs résolutions simultanément à chaque endroit et être capable de les faire communiquer. On gagne en généricité (les différentes zones peuvent communiquer directement au travers de la même résolution) au prix d'un coût de calcul légèrement plus élevé dû à la simulation de plusieurs résolutions à chaque endroit.

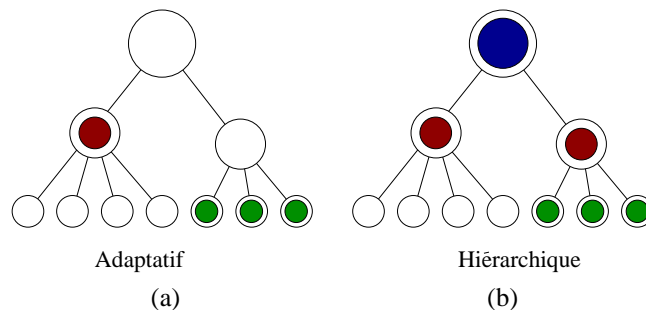


FIG. 1.9: Parmi toutes les résolutions possibles, l'adaptatif (a) ne simule que le niveau le plus fin (points) alors qu'en hiérarchique (b), les niveaux inférieurs continuent à être actifs et cohabitent avec le niveau fin.

Ces définitions ne sont pas académiques et sont introduites ici par commodité afin de pouvoir s'y référer dans la suite de ce document lorsque nous décrivons nos contributions.

### 10.4 Utilisations de la multirésolution

Différents articles ont tout d'abord cherché à formaliser quels étaient les types de mouvements que l'on pouvait générer et les critères permettant de choisir lequel est le mieux approprié.

Berka introduit dans [Ber97] la notion de niveau de détail *dynamique* (DLOD). Il énumère quelques critères de décision quand au choix du niveau de détail : l'amplitude observable à l'écran du mouvement, sa vitesse et le nombre des objets qui l'adoptent, permettent éventuellement de les regrouper. Il soulève, sans toutefois y répondre, les problèmes qui apparaissent alors : comment mesurer ces paramètres, quel sera le mouvement simplifié et comment gérer les transitions entre les niveaux ?

Chenney et Forsyth [CF97] vont eux étudier à quel point on peut simplifier le calcul de l'animation des objets temporairement non visibles. Il faut pour cela que l'état (position, posture) dans lequel les objets réapparaissent soit cohérent avec ce dont l'utilisateur peut se souvenir. Le but va donc être de reproduire la *statistique* décrivant les états de l'objet ; plus celle-ci est aléatoire, plus on pourra simplifier les calculs consacrés aux objets non visibles. Les gains de temps annoncés par les auteurs varient entre 1.5 et 5.

Dans sa thèse [Val99], Valton décrit un mécanisme de décomposition en niveaux de détail de modèles d'animation. Une structure d'arbre comportant différents modules découplés, pouvant être désactivés lorsqu'il faut dégrader le mouvement, permettent de régler la qualité de l'animation. Une grammaire permet de corréliser les changements de niveaux de détail dans l'animation et dans le modèle géométrique affiché.

Carlson et Hodgins ont proposé un modèle utilisant trois niveaux de détail dans l'animation par modèle physique de "pistons" bondissants [CH97]. Selon que chaque objet est proche ou lointain, ou utilisera une simulation dynamique, cinématique, voire simplement basée sur des particules en ne considérant l'objet que comme un point. Même si chaque niveau n'a pas tout à fait le même comportement, on obtient quand même ainsi un gain sur le temps de calcul.

Rappelons dans l'approche modale de Pentland et Williams [PW89] la possible utilisation multirésolution d'un nombre donné de modes vibratoires permettant de modifier le compromis vitesse-qualité.

En utilisant cette fois-ci un réseau masses-ressorts, Hutchinson *et al.* proposent une méthode adaptative de simulation de tissus [HPH96]. Celui-ci est tout d'abord maillé grossièrement, mais de nouvelles masses sont ajoutées en fonction de l'animation lorsque l'angle entre deux ressorts devient trop important pour affiner le résultat. On accélère ainsi le début de la simulation dans leur exemple d'une nappe tombant sur une table et se pliant au bord. Le problème principal de la méthode est que l'adjonction de ressorts modifie sensiblement le comportement du tissu, chaque point ajouté apportant par exemple dans leur implémentation une masse supplémentaire. Rien n'est de plus prévu pour resimplifier localement le maillage si l'objet change de position.

Ganovelli a lui aussi développé des approches multirésolution à base de masses-ressorts. Ses premiers travaux [GCS99] cherchaient à déterminer la raideur équivalente à plusieurs ressorts en calculant leur résistance à un fort étirement (tous les ressorts ayant atteint leur longueur limite). Il a ensuite utilisé les raideurs de Van Gelder [Gel98] dans un modèle autorisant les découpes [GCMS00]. La faible indépendance à la résolution de ce modèle (voir Chapitre 4, Section 7) grève néanmoins cette méthode.

Desbrun développe lui aussi un modèle multirésolution (voir Sec. 5.3) en temps (le pas de temps d'intégration de chaque particule peut varier) et en espace (les particules peuvent se diviser et se regrouper) basé sur le formalisme SPH [DCG96, Des97, GCD<sup>+</sup>98, DC99b]. Cette méthode a été développée pour des matériaux hautement déformables, et les essais que nous avons faits d'application à des matériaux plus élastiques que plastiques n'ont pas été concluants. C'est néanmoins ce genre d'approche qui nous inspirera et que nous chercherons à appliquer.

Parmi les méthodes utilisant des éléments finis, citons les travaux de Hayward [AH97] qui, par une analogie avec les impédances équivalentes de circuits électriques, pouvait séparer l'objet simulé en 2 parties, l'une fine et l'autre plus grossière, chacune ne voyant l'autre que par un modèle équivalent simplifié au travers des noeuds situés à l'interface.

Zhuang décrit dans sa thèse [Zhu00] un formalisme à base d'éléments finis permettant de simuler en temps-réel les déformations d'objets déformables. La discrétisation, fixe au cours du temps, n'est pas uniforme : plus fine à proximité de la surface, elle permet d'obtenir plus de précision dans la zone d'interaction.

Nous avons déjà décrit le maillage adaptatif généré par les fractures dans [OH99]. Même si le maillage évolue pour s'adapter à la simulation, on ne peut toutefois ici, pas plus que dans les masses-ressorts de Hutchinson [HPH96], parler de multirésolution. Il faudrait pour cela que l'on puisse par exemple pour [OH99] à nouveau simplifier le maillage des morceaux qui se détachent de l'objet et ont un simple vol balistique ne nécessitant pas l'animation de tous les tétraèdres dont ils sont constitués.

## 11 Choix d'un modèle

### 11.1 Méthodes interactives

Il existe déjà des méthodes permettant de simuler interactivement des matériaux déformables, dans un contexte chirurgical par exemple. On pourrait dire que plus rien n'est à faire, certaines de ces méthodes "permettant une vitesse de rafraîchissement de 15 Hz". Cet argument n'a cependant aucune valeur, le taux de rafraîchissement ne signifiant strictement rien si l'on ne sait pas à quelle durée d'animation physique correspond l'intervalle entre deux affichages. Si les durées ne sont pas les mêmes, le mouvement apparaîtra simplement dans ces cas comme très ralenti, bien qu'affiché à 25 images par secondes.

### 11.2 Le temps réel vrai

Sont généralement désignées comme temps-réel toutes les méthodes permettant l'affichage de plusieurs images par seconde, définition bien trop imprécise que nous préférons restreindre. Une application temps-réel *vrai* synchronise à chaque image affichée le temps ressenti par la personne qui la regarde à celui correspondant à l'étape de simulation affichée. Il faut en d'autres termes que lorsque deux images sont affichées à une seconde d'intervalle, la machine ait effectivement calculé le comportement du matériau durant une seconde, ni plus, ni moins. Dans le cas contraire, le mouvement apparaîtra respectivement comme joué en accéléré ou au ralenti.

Ce genre de considération n'est que très rarement pris en compte [Luc85]. On choisit généralement de simuler le nombre maximal de points d'échantillonnage qui permette de conserver un affichage de plusieurs images par secondes. Dès que le matériau est un peu rigide, le pas de temps d'intégration doit être baissé pour limiter la divergence de l'intégration<sup>13</sup>. Les dix à cinquante images affichées par seconde (après chaque pas de simulation) n'ont alors plus rien à voir avec l'intervalle de un millième de seconde ou moins qui est supposé les séparer.

Que l'animation ne soit pas affichée à la bonne vitesse ne gêne pas trop si on ne cherche qu'un réalisme visuel. On pourra chercher à jouer sur les coefficients pour obtenir l'effet recherché (on aura tout de même du mal à créer un corps rigide et ayant beaucoup de particules). Il faut également garantir que chaque pas de simulation prendra le même temps machine à être calculé pour être sûr que l'animation se fera à une vitesse éventuellement fautive, mais au moins constante.

### 11.3 Nécessité de la multirésolution

Le temps de calcul d'une seconde d'animation dépend linéairement du nombre  $n$  de points simulés et du temps mis pour calculer une force en chaque point<sup>14</sup>. Si l'on admet que le temps de calcul d'une force en un point est une constante, on est limité par le nombre de points que l'on peut simuler. Il faut donc employer des méthodes multirésolution si on veut accroître le réalisme visuel et *optimiser* l'utilisation de ces  $n$  points, en les plaçant à chaque instant au bon endroit dans le matériau.

Dans des animations multirésolution encore plus que dans les autres méthodes, la charge de calcul va pouvoir varier au cours de l'animation, même si on cherchera à l'utiliser à son maximum. Il conviendra donc de bien synchroniser calcul et affichage pour obtenir ce que nous avons appelé le temps-réel vrai.

Mettre en place de telles méthodes nécessite un formalisme suffisamment indépendant de la résolution employée pour que l'on puisse espérer simuler des discrétisations différentes sans que cela n'apparaisse trop.

### 11.4 Modèle indépendant de la résolution.

Nous allons ici chercher, à la lumière de l'état de l'art qui précède, quelle sont les méthodes permettant une simulation indépendante de la résolution.

Le comportement des réseaux masses-ressorts est directement lié au maillage utilisé. On ne peut espérer faire facilement cohabiter des maillages trop différents ayant le même comportement dynamique. Van Gelder a démontré dans un article [Gel98] *l'impossibilité* de régler les raideurs d'un réseau masses-ressorts pour le faire coïncider avec un modèle continu de la matière et de le rendre indépendant de la résolution.

Van Gelder présente les deux méthodes dans un même formalisme matriciel (sur un réseau à trois points fait de masses-ressorts ou d'éléments finis) et cherche à déterminer les raideurs à donner aux ressorts pour les faire coïncider. Il fournit un contre exemple simple prouvant que c'est impossible. En simplifiant considérablement le modèle continu de la matière<sup>15</sup>, Van Gelder propose de choisir la raideur  $k$  de *chaque* ressort selon :

$$k = \frac{E V}{l^2}$$

pour minimiser la différence entre les deux méthodes.  $E$  est le module d'Young<sup>16</sup>, représentant la raideur globale que l'on souhaite donner au matériau.  $V$  est la somme des volumes des tétraèdres (des aires des triangles en 2D) adjacents à l'arête où est le ressort, alors que  $l$  est sa longueur au repos.

Ces considérations ne sauraient totalement supprimer le trop fort lien existant entre le maillage et le comportement d'un réseau masses-ressorts, ce qui les rend inutilisables en multirésolution.

Parmi les méthodes présentées, seules celles basées sur l'utilisation d'une équation d'état caractérisant le matériau garantissent un comportement indépendant (dans une certaine mesure) de la résolution employée. On se tournera donc vers les SPH et les méthodes de type élément finis.

<sup>13</sup>Voir annexe B

<sup>14</sup>On exclut donc d'emblée les méthodes matricielles de type éléments finis implicites dont la résolution est en  $O(\hat{n})$  au mieux.

<sup>15</sup>On supprime toute notion d'incompressibilité dans le matériau, ce qui revient à prendre  $v = 0$  avec les notations introduites au chapitre suivant.

<sup>16</sup>Voir le chapitre 2.

Nos premiers essais ont été réalisés avec une méthode à base de SPH. Ils n'ont pas permis la simulation de matériaux trop rigides, le modèle étant mieux adapté aux interactions inter-particules faibles. En effet, l'approximation d'une intégrale continue par une somme discrète telle qu'on le fait en Section 5.1 n'est possible que si le volume sur lequel on intègre est suffisamment bien échantillonné. Les simulations astrophysiques gèrent plusieurs millions de points qui font que cette approximation est valable. Dans notre cas, les contraintes de temps-réel font que l'on doit se contenter de quelques centaines de points au plus, les noyaux de convolution n'en englobant qu'une partie dans chaque intégrale, ce qui rend l'approximation bien plus grossière.

Libersky et Petschek simulent bien un matériau très rigide [DP90], mais là encore il utilisent beaucoup de points et les temps de calcul sont prohibitifs, même pour leur exemple 2D.

Les résultats visuels que nous avons constatés font état d'un matériau très mou, tendant plus vers le liquide ou le boueux que vers le solide, ce qui était bien adapté aux objectifs de Desbrun. Obtenir des matériaux rigides en augmentant la cohésion demande un grand nombre de points simulés et des pas de temps très faibles pour que l'animation ne diverge pas, ce qui rend cette méthode difficilement compatible avec notre objectif de temps-réel.

C'est vers des méthodes utilisant une loi de comportement globale, basées sur une vision continue de la matière que nous allons nous tourner. La solution semblait alors devoir passer par l'utilisation de méthodes de type éléments finis, assez complexes et très coûteuses puisque impliquant des résolutions matricielles. Les méthodes explicites nous étaient inconnues aux débuts de nos travaux et nous avons cherché à développer des méthodes similaires, impliquant le calcul direct d'une force en chaque point, intégrée séparément par la suite comme dans un système de particules. Nous comparerons par la suite les résultats obtenus avec ceux de certaines méthodes d'éléments finis. Seront alors élaborés des algorithmes permettant une utilisation multirésolution de ces calculs.

## 12 Conclusion générale

Ce chapitre a présenté les grandes classes de méthodes utilisées dans la simulation d'objets déformables ainsi que les principaux articles s'y rapportant. Des conclusions partielles explicitaient leurs avantages et leurs inconvénients, ainsi que leur adéquation avec nos buts de simulation temps-réel tirant parti de la multirésolution.

Nous avons vu qu'une solution pouvait être le mélange des méthodes à base de particules de type masses-ressorts (pour leur rapidité), et de forces issues de la description continue de la matière faite par les physiciens (pour leur indépendance à la résolution).

Nous allons en quelque sorte mettre au point un système de particules amélioré en cherchant une expression de la force entre deux points qui soit liée aux caractéristiques physiques du matériau. Nous allons pour cela étudier dans le chapitre suivant le formalisme utilisé par les physiciens pour décrire la matière comme un milieu continu. Nous verrons par la suite comment calculer ces forces et mettre alors en place les méthodes multirésolution qui en tireront parti.

