

Simulation en Lagrangien Total Implicite à base de Repères

Julien Bosman¹, Christian Duriez²

¹LIFL/INRIA, ²INRIA

Résumé

A faire

Mots clé : Simulation, Méthodes sans maillage, Mécanique

1. Introduction

A faire

2. Etat de l'art

La simulation à base de méthodes sans maillage a été introduite en informatique graphique avec [DC96] et l'utilisation des *Smoothed Particle Hydrodynamics* (SPH) – reposant sur l'interaction réciproque d'un ensemble de particules – pour l'animation de modèles très déformables. Les SPH demeurent une méthodes populaires pour l'animation de fluides et d'objets déformables. Les travaux de [BIT09] apportent une réponse au problème des non-linéarités géométrique en proposant un modèle corotationnel dédié. La méthode des SPH permet également de simuler des matériaux non-newtoniens aux comportements plus complexes notamment dans [PPLT09] et [HMH07]. Parallèlement, d'autres méthodes d'animation sans maillage ont été développées. Des modèles basés sur la mécanique des milieux continus à l'instar de [KNP*04] et de [HWJM10] où les champs de déplacement sont approximés grâce aux *Moving Least Squares* (MLS). Des techniques d'animation, non basées sur la physique, ont également été développées. Dans [MHTG05], les objets sont déformés grâce à un système masse ressort en utilisant la méthode dite de *Shape Matching*, où les points de l'objet sont partitionnés en plusieurs groupes dont on estime une rotation optimale et une translation par rapport à la configuration non déformée. Cette idée est reprise dans [MC11] à la différence près que chaque possède une orientation qui est estimée au grâce à un *Shape Matching* généralisé.

3. Simulation

3.1. Cinématique

Les objets déformables sont discrétisés par un ensemble de nœuds et de points d'intégration. Les nœuds sont des repères à six degrés de liberté (DDL) :

$$\mathbf{q}_i = [x_1 \ x_2 \ x_3 \ r_1 \ r_2 \ r_3]^t = [\mathbf{x}_i \ \mathbf{r}_i]^t \quad (1)$$

Les points d'intégration ne possèdent que trois DDL. Dans la mesure où l'on utilise une formulation en lagrangien total, la position globale des points d'intégration est exprimée en fonction de leur position dans chacun des repères dans la configuration non déformée. Les coordonnées d'un point d'intégration s dans le repère i dans la configuration non déformée est donné par :

$$\tilde{\mathbf{p}}_s^{l,i} = [\mathbf{R}_i^{0,t}]^t \times (\tilde{\mathbf{p}}_s - \tilde{\mathbf{x}}_i) \quad (2)$$

où \mathbf{R}_i^0 et $\tilde{\mathbf{x}}_i$ sont la matrice de rotation et les coordonnées du repère i et $\tilde{\mathbf{p}}_s$ les coordonnées du point d'intégration s dans la configuration non déformée. Ainsi, les coordonnées du point d'intégration s dans la configuration déformée sont donnés par :

$$\mathbf{p}_s = \sum_i w_s(\mathbf{x}_i) \left(\mathbf{x}_i + [\mathbf{R}_i] \times \tilde{\mathbf{p}}_s^{l,i} \right) \quad (3)$$

où $w_s(\cdot)$ est une fonction de forme associée au point d'intégration s et q_i et \mathbf{R}_i les coordonnées et la rotation du repère i dans la configuration déformée. La fonction de forme $w_s(\cdot)$ peut être à support compact (du type *fonction de Wendland*) ou non (fonction de shepard par ex.

De la même façon, la cinématique des repères est influencée par le mouvement des points d'intégration qui lui sont associés. L'orientation du repère i dans la configuration dé-

formée et donnée par :

$$\mathbf{r}_i = \sum_s w_s(\mathbf{x}_i) \left(\bar{\mathbf{p}}_s^{l,i} \wedge \mathbf{u}_s^{l,i} \right) \quad (4)$$

où $\mathbf{u}_s^{l,i} = \bar{\mathbf{p}}_s^{l,i} - \bar{\mathbf{p}}_s^{l,i}$ est le déplacement subi par le point d'intégration s localement au repère i .

3.2. Dynamique

3.2.1. Expression des forces internes

La méthode décrite dans cet article utilise une formulation en lagrangien total. Dans une formulation en lagrangien total les déplacements sont évalués par rapport à la configuration non déformée de l'objet contrairement à la formulation en lagrangien updatée où ils sont évalués par rapport à la configuration de l'objet au pas de temps précédent. Le calcul des forces internes fait appel au principe des *travaux virtuels* :

$$\delta W = \delta q \cdot f = \int_{\Omega} \frac{\partial L}{\partial q} \{S\} d\Omega \quad (5)$$

où $L = \frac{1}{2}(F^t F - I)$ est le tenseur de déformation de *Green-Lagrange* où F est le gradient de déformation. S le second tenseur de contrainte de *Piola-Kirchoff*. L'expression de la dérivée du tenseur de Green-Lagrange par rapport au degrés de liberté des repères est donné par :

$$\frac{\partial L}{\partial q} = \frac{1}{2} \left(\frac{\partial F^t}{\partial q} F + F^t \frac{\partial F}{\partial q} \right) \quad (6)$$

L'équation (3) permet de calculer le tenseur de déformation en un point d'intégration s :

$$F_s = \frac{\partial \bar{\mathbf{p}}_s}{\partial \mathbf{X}} = \sum_i \left[\frac{\partial w_s(\mathbf{x}_i)}{\partial \mathbf{X}} \left(\mathbf{x}_i + [\mathbf{R}_i] \bar{\mathbf{p}}_s^{l,i} \right) + w_s(\mathbf{x}_i) [\mathbf{R}_i] [\mathbf{R}_i^0]^t \right] \quad (7)$$

Afin d'obtenir la dérivée du tenseur de *Green-Lagrange* de l'équation (6) il est nécessaire de calculer la dérivée du tenseur de déformation :

$$\delta F_{IJ}^s = \sum_i \left(\delta \mathbf{x}_i + [-\mathbf{R}_i(\bar{\mathbf{p}}_s^{l,i})^\wedge]^t \cdot \delta \mathbf{r}_i \right) \frac{\partial w_s(\mathbf{x}_i)}{\partial \mathbf{X}_j} + [-\mathbf{R}_i(r_{i,j}^0)^\wedge]^t \cdot \delta \mathbf{r}_i w_s(\mathbf{x}_i) \quad (8)$$

avec $\mathbf{R}_i(\bar{\mathbf{p}}_s^{l,i})$ la rotation de $\bar{\mathbf{p}}_s^{l,i}$ par \mathbf{R}_i , $[\mathbf{v}^\wedge]$ la matrice anti-symétrique 3×3 associée au produit vectoriel, $[\mathbf{M}]^i$ et $[\mathbf{M}]^j$ correspondent respectivement à la $i^{\text{ème}}$ ligne et à la $j^{\text{ème}}$ colonne de la matrice \mathbf{M} et $r_{i,j}^0 = [\mathbf{R}_i^0]^j$. En réécrivant les 9 valeurs de δF^s sous forme d'un vecteur colonne, l'équation (8) peut être exprimée sous la forme d'un produit matrice-vecteur :

$$\begin{bmatrix} \delta F_{xx}^s \\ \delta F_{xy}^s \\ \vdots \\ \delta F_{zz}^s \end{bmatrix} = \sum_i \mathbf{B}^{i,s} \cdot \delta \mathbf{q}_i \quad (9)$$

avec $\delta \mathbf{q}_i^t = [\delta \mathbf{x}_i \ \delta \mathbf{r}_i]^t$. La matrice B est donnée en annexe. De

la même façon, il est également possible d'écrire $\partial L / \partial q$ sous la forme d'un produit matrice-vecteur. En effet :

$$\begin{aligned} \delta L_{xy} &= \frac{1}{2} \left([\delta F^t]^{1,1} \cdot [F]^2 + [F^t]^{1,1} \cdot [\delta F]^2 \right) \\ &= \frac{1}{2} \left([\delta F_{xx} \ \delta F_{yx} \ \delta F_{zx}] \cdot [F]^2 + [F^t]^{1,1} \cdot [\delta F_{xy} \ \delta F_{yy} \ \delta F_{zy}] \right) \end{aligned} \quad (10)$$

Or, d'après l'équation (9) :

$$[\delta F_{xx} \ \delta F_{yx} \ \delta F_{zx}] = [B_1, \cdot \delta q \ B_3, \cdot \delta q \ B_7, \cdot \delta q] \quad (11)$$

avec B_i , la $i^{\text{ème}}$ ligne de la matrice B d'où :

$$\delta L_{xy} = \frac{1}{2} [F_{xy} B_1, + F_{yy} B_4, + F_{zy} B_7, + F_{xx} B_2, + F_{yx} B_5, + F_{zx} B_8,] \cdot \delta q \quad (12)$$

On a donc :

$$\begin{bmatrix} \delta L_{xx}^s \\ \delta L_{xy}^s \\ \vdots \\ \delta L_{zz}^s \end{bmatrix} = \sum_i \mathbf{M}^{i,s} \cdot \delta \mathbf{q}_i \quad (13)$$

En substituant l'équation (13) à l'équation (5) et en divisant par δq on obtient :

$$f = \int_{\Omega} \mathbf{M}^t \{S\} d\Omega \quad (14)$$

ou, en version discrète :

$$f_i = \sum_s \mathbf{M}^{i,s} \{S^s\} w_s(\mathbf{x}_i) \quad (15)$$

avec S calculé grâce à la *loi de Hooke* :

$$\mathbf{S} = \lambda \text{tr}(\mathbf{L}) \mathbf{I} + 2\mu \mathbf{L} \quad (16)$$

en notation de Voigt où λ et μ sont les *coefficients de Lamé*

3.2.2. Formulation implicite

La formulation précédente (équation 15) est utilisable avec un solveur explicite. Cependant, bien qu'ils soient rapides, les solveurs explicites présentent certains désavantages, notamment l'instabilité. Les oscillations et l'introduction d'énergie supplémentaire conduisent la simulation à "exploser". Une solution couramment employée pour répondre à ce problème consiste à utiliser des pas de temps très court avec pour effet de ralentir la simulation. Pour ces raisons, on utilise un solveur implicite qui est inconditionnellement stable – une propriété intéressante dans le cadre de la simulation d'objets déformables. Les solveurs implicites permettent de résoudre un problème non-linéaire de manière itérative selon la méthode de *Newton-Raphson* [FDD*12]. Pour permettre au solveur de résoudre l'équation du mouvement, il est nécessaire de linéariser la force, c'est-à-dire dériver l'expression de la force par rapport aux repères. D'après [BLM00] on a :

$$\delta f = \int_{\Omega} \mathbf{M}^t \{\delta S\} d\Omega \quad (17)$$

avec :

$$\{\delta \mathbf{S}\} = \mathbf{C}\{\delta \mathbf{L}\} \quad (18)$$

ou la matrice \mathbf{C} donnée en annexe est la matrice des coefficients élastiques associés à la loi de Hooke. En substituant l'équation (13) à l'équation (17) on obtient :

$$\delta f = \int_{\Omega} \mathbf{M}^t \mathbf{C} \mathbf{M} \cdot \delta \mathbf{q} d\Omega \quad (19)$$

on obtient alors, en version discrète :

$$\delta f_i = \sum_s \mathbf{M}^{i,s} \cdot \Delta \mathbf{S}^s \quad (20)$$

avec :

$$\Delta \mathbf{S}^s = \sum_i \mathbf{C} \mathbf{M}^{i,s} \cdot \delta \mathbf{q}_i \quad (21)$$

3.3. Figures en couleur

Etant donné qu'il s'agit d'une publication électronique, il n'y a aucune censure à avoir sur les figures en couleur (cf figure 1.

4. Conclusion

A faire

Références

- [BIT09] BECKER M., IHMSEN M., TESCHNER M. : Corotated SPH for deformable solids.
- [BLM00] BELYTSCHKO T., LIU W. K., MORAN B. : *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.
- [DC96] DESBRUN M., CANI M. : Smoothed particles : A new paradigm for animating highly deformable bodies. *Proceedings of the Eurographics workshop on Computer animation and simulation* (1996), 61–76.
- [FDD*12] FAURE F., DURIEZ C., DELINGETTE H., ALLARD J., GILLES B., MARCHESSEAU S., TALBOT H., COURTECUISSE H., BOUSQUET G., PETERLIK I., COTIN S. : SOFA : A Multi-Model Framework for Interactive Physical Simulation. In *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, Payan Y., (Ed.). Springer, juin 2012.
- [HMH07] HOSSEINI S. M., MANZARI M. T., HANNANI S. K. : A fully explicit three-step SPH algorithm for simulation of non-Newtonian fluid flow. *International Journal of Numerical Methods for Heat & Fluid Flow*. Vol. 17, Num. 7 (2007), 715–735.
- [HWM10] HORTON A., WITTEK A., JOLDES G. R., MILLER K. : A meshless Total Lagrangian explicit dynamics algorithm for surgical simulation. *Integration The Vlsi Journal*, Num. March (2010), 977–998.
- [KNP*04] KEISER R., NEALEN A., PAULY M., GROSS M., ALEXA M. : Point Based Animation of Elastic, Plastic and Melting Objects. *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation* (2004), 141–151.
- [MC11] MÜLLER M., CHENTANEZ N. : Solid simulation with oriented particles. *ACM Trans. Graph.* Vol. 30, Num. 4 (juillet 2011), 92 :1–92 :10.
- [MHTG05] MÜLLER M., HEIDELBERGER B., TESCHNER M., GROSS M. : Meshless deformations based on shape matching. *ACM Trans. Graph.* Vol. 24, Num. 3 (juillet 2005), 471–478.
- [PPLT09] PAIVA A., PETRONETTO F., LEWINER T., TAVARES G. : Particle-based viscoplastic fluid/solid simulation. *Comput. Aided Des.* Vol. 41, Num. 4 (avril 2009), 306–314.



Figure 1: Les figures en couleurs ou en noir et blanc peuvent être placées n'importe où dans le document.

Appendices

A. Calcul de la matrice B

$\mathbf{B}^{i,s} =$

$$\begin{bmatrix} \frac{\partial w_{is}}{\partial x} & 0 & 0 & \left[\frac{\partial w_{is}}{\partial x} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^x + [-\mathbf{R}_i(r_{,x}^0)^\wedge]^x, w_{is} \right] \\ \frac{\partial w_{is}}{\partial y} & 0 & 0 & \left[\frac{\partial w_{is}}{\partial y} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^y + [-\mathbf{R}_i(r_{,y}^0)^\wedge]^y, w_{is} \right] \\ \frac{\partial w_{is}}{\partial z} & 0 & 0 & \left[\frac{\partial w_{is}}{\partial z} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^z + [-\mathbf{R}_i(r_{,z}^0)^\wedge]^z, w_{is} \right] \\ 0 & \frac{\partial w_{is}}{\partial x} & 0 & \left[\frac{\partial w_{is}}{\partial x} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^y + [-\mathbf{R}_i(r_{,x}^0)^\wedge]^y, w_{is} \right] \\ 0 & \frac{\partial w_{is}}{\partial y} & 0 & \left[\frac{\partial w_{is}}{\partial y} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^x + [-\mathbf{R}_i(r_{,y}^0)^\wedge]^x, w_{is} \right] \\ 0 & \frac{\partial w_{is}}{\partial z} & 0 & \left[\frac{\partial w_{is}}{\partial z} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^x + [-\mathbf{R}_i(r_{,z}^0)^\wedge]^x, w_{is} \right] \\ 0 & 0 & \frac{\partial w_{is}}{\partial x} & \left[\frac{\partial w_{is}}{\partial x} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^z + [-\mathbf{R}_i(r_{,x}^0)^\wedge]^z, w_{is} \right] \\ 0 & 0 & \frac{\partial w_{is}}{\partial y} & \left[\frac{\partial w_{is}}{\partial y} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^z + [-\mathbf{R}_i(r_{,y}^0)^\wedge]^z, w_{is} \right] \\ 0 & 0 & \frac{\partial w_{is}}{\partial z} & \left[\frac{\partial w_{is}}{\partial z} [-\mathbf{R}_i(\tilde{\mathbf{p}}_s^{l,i})^\wedge]^z + [-\mathbf{R}_i(r_{,z}^0)^\wedge]^z, w_{is} \right] \end{bmatrix}$$

avec $w_{is} = w_s(\mathbf{x}_i)$

5. Matrice de Hooke

$$C = \frac{E}{(1+\nu)(2-\nu)} \times \begin{bmatrix} 1-\nu & \nu & \nu & 0 \\ \nu & 1-\nu & \nu & 0 \\ \nu & \nu & 1-\nu & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix}$$

où E et ν sont respectivement le *module de Young* et le *coefficient de Poisson*.