Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2013)
J. Bender, J. Dequidt, C. Duriez, and G. Zachmann (Editors)

Multi-criteria adaptation of physical simulations

F. Caillaud[1,2] and X. Faure[1,2] and F. Zara[2] and F. Jaillet[2,3] and J-M. Moreau[2]

[1]Financed by the Rhône-Alpes Research Program on Hadrontherapy for ETOILE (National French Hadrontherapy Centre)
[2]Université de Lyon, CNRS, Université Lyon 1, LIRIS, SAARA team, UMR5205, F-69622, Villeurbanne, France
[3]Université de Lyon, IUT Lyon 1, Computer Science Department, F-01000, Bourg-en-Bresse, France

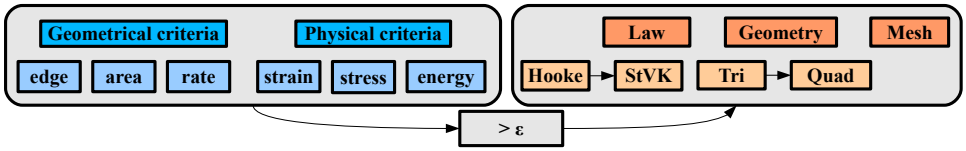


Figure 1: *Overview of our approach to dynamically adapt the simulation according to criteria.*

Abstract

This paper presents a GPU implementation of a revisited Mass-Tensor model to dynamically adapt the simulation of deformable objects. In this model, the mechanical forces are computed at the node. This facilitates the manipulation of the elements to improve the computation performance while preserving accuracy. The study of several criteria, either geometric or mechanical, determines the most appropriate response. For example, the adaptation involves different actions: to locally refine the object's mesh, to raise the degree of an object's element or to adapt the mechanical laws.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling I.6.8 [Simulation And Modeling]: Types of Simulation—Parallel

1. Introduction

The accurate and interactive simulation of deformable objects has become a challenge in computer graphics [DCA99, SDR*05]. As an alternative to the classical approach of the FEM, the Tensor-Mass solves the mechanical equations by directly computing forces applied on each node, instead of forces applied on each element. This approach offers a good compromise between accuracy, computation time and control.

A symbolic computation approach for this model [FZJM12] enables to multiply the possible combinations between geometrical elements (quads, triangles, tetrahedra, hexahedra with different integration nodes) and constitutive laws (from linear and non linear elasticity). This new possibility gives wide opportunities at low development cost, to mix several mechanical behaviors with different kinds of elements inside a same object.

In this paper, we explore the opportunity to improve the simulation in precision or time, by adaptively modifying the model according to criteria (see Fig. 1).

2. Background on the Tensor-Mass Model

Considering an object's element E with n nodes, the contribution of the forces applied on each node P_i for $i \in [0, n-1]$ of this element is defined by:

$$\mathbf{F}_E(P_i) = -\frac{\partial \mathbf{W}_E(P_i)}{\partial \mathbf{U}_i}$$

with $\mathbf{W}_E$ the total strain energy of the element in 2D defined by:

$$\begin{aligned}\mathbf{W}_E &= \int_E W_{law}(x,y)\, dx\, dy \\ &= \sum_{i=0}^{n_g-1} w_i\, \mathbf{W}_{law}(\xi_i,\eta_i)\, det(J)\end{aligned}$$

with J the transformation between spatial (x,y) and local (ξ,η) coordinates, n_g the number of sampling points (ξ_i,η_i) of weights w_i and W_{law} the strain energy depending on the law (Hooke's or Saint Venant-Kirchhoff's law).

The total force $\mathbf{F_i}$ applied on node P_i is then computed by summing the contributions from all the neighboring elements.

At time t, the Newton equation governing the object's movement involves the following formulation for the node P_i of mass m_i and displacement $\mathbf{U_i}$:

$$m_i \, \ddot{\mathbf{U}}_\mathbf{i}(t) = \mathbf{F}_i(t).$$

This equation is linked to the differential equation system by assuming the sparse matrix of the masses to be diagonal, with the distribution of the object's mass on each node (mass-lumping).

3. Adaptive Tensor-Mass simulation

As stated previously, the possibilities offered by [FZJM12] allows us to modify an object or a scene during the simulation. The necessary actions are part of an adaptive strategy, and may concern the constitutive law (Hooke, Saint Venant-Kirchhoff, etc.), the geometry (quad, triangle, etc.), the refinement (addition or suppression of nodes), or the shape functions used for each element of the object (Fig. 1). Algorithm 1 summarizes our approach.

Algorithm 1 Adaptation at each step of the simulation according to the evaluation of the criteria.

```
{lc ← list criteria}
for each element e do
    p ← position of element's nodes
    u ← displacement of element's nodes
    f ← force of element's nodes
    for each criterion c do
      c.eval(e,p,u,f)
    endfor
    for each strategy s do
      if s.actionNeeded(e,lc) then
        s.action(e)
      endif
    endfor
endfor
```

Geometrical criteria The Hookean model is known to reach its limit for large displacements (detected by the edge length ratio R_L) or for rotations (where unwanted area/volume increase is detected by R_A). When this happens, a good solution is to switch to a non-linear law, leading to a more accurate simulation while controlling the computation time (see Fig. 2)[†].

Other criteria like *Aspect Ratio* or *Harmonic Mean of the length of the edges* can be useful to consider the quality of the elements. This time, a geometric action like refinement is advocated (as a change in the constitutive law should have limited action).

[†] The Open Source Framework SOFA was used for the implementation [ACF*07].

Physical criteria. Besides classical geometric estimation errors, we can also consider a physical criterion noted R_S linked to the stress applied on each element E of area $A(t)$ at time t, with:

$$R_S = \frac{1}{A(t)} \int_E \mathbf{F}_E(x,y)dx \, dy.$$

This physical criterion has to be discretized. But as the forces at the Gaussian point g_i are not known, we also need to interpolate it according to the forces applied on the vertices. We obtain:

$$R_S = \frac{1}{A(t)} \sum_{i=0}^{n_g-1} w_i \sum_{j=0}^{n-1} \|\Lambda_j(g_i) \, \mathbf{F}_E(P_j)\|$$

with $\mathbf{F}_E(P_j)$ the force applied on the vertex P_j and w_i the weighting function applied on the Gaussian point g_i. Note that most of the terms have already be computed during simulation. Fig. 3 shows our results.

4. GPU implementation

A GPU implementation of the Tensor-Mass model has been proposed in [FZJM12]. But in adaptive simulation on the GPU, elements are handled by groups with the same association geometry/constitutive law. When an element is promoted to another group, this implies to keep up-to-date the list of elements' indices; and to load the new list for each group as well as all information linked to the considered elements. The GPU implementation involves:

1. low memory (50 times less than Abaqus^{TM}). Thus, for each group, we can set an array of the number of elements, meaning that there is no need to manage resizing at each adaptation step;
2. reduced reading and writing times to transfer data from RAM to GPU memory (14 % of the total time, in the worst case).

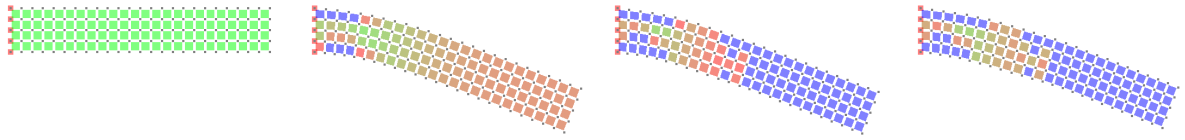
References

[ACF*07] Allard J., Cotin S., Faure F., Bensoussan P.-J., Poyer F., Duriez C., Delingette H., Grisoni L.: Sofa an open source framework for medical simulation. In *MMVR'15* (Feb. 2007). 2

[DCA99] Delingette H., Cotin S., Ayache N.: A Hybrid Elastic Model Allowing Real-Time Cutting Deformations and Force Feedback for Surgery Training and Simulation. In *Computer Animation'99* (1999), IEEE Computer Society, pp. 70–81. 1

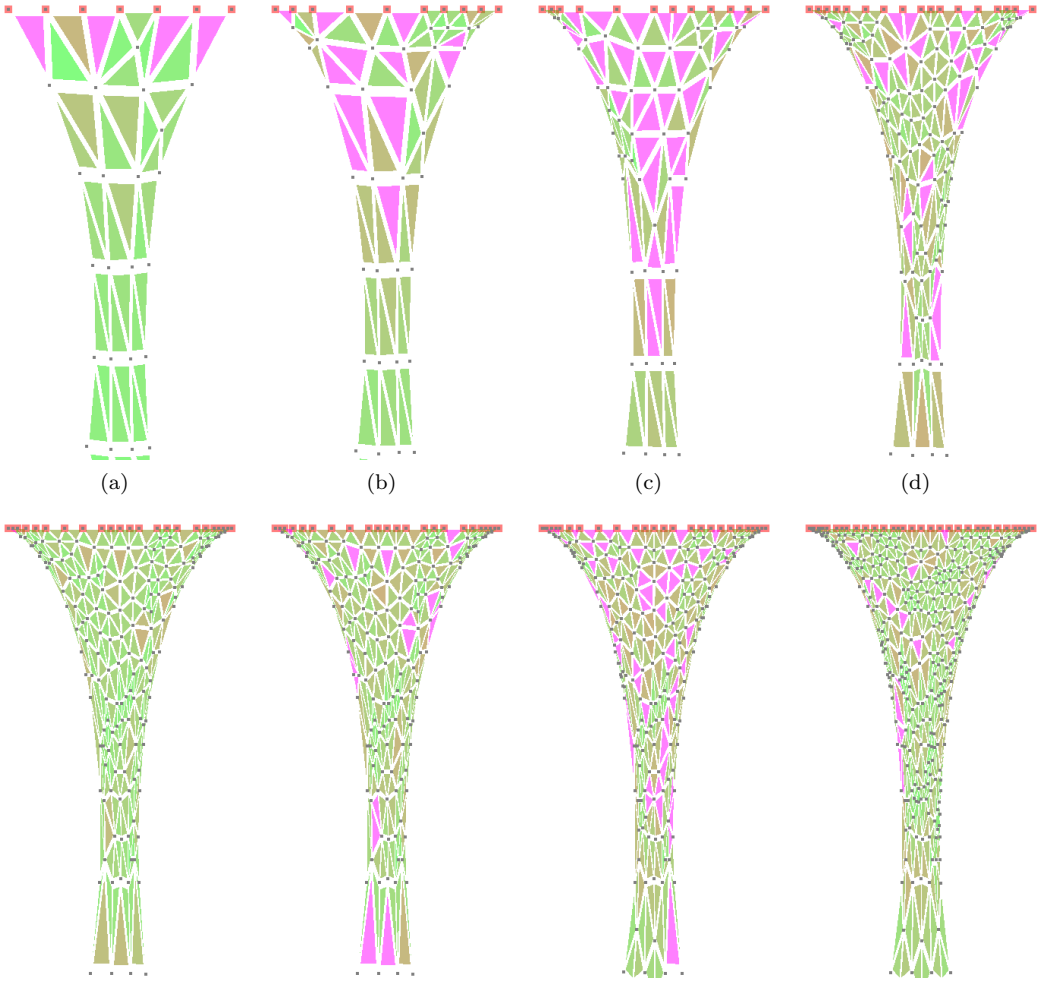
[FZJM12] Faure X., Zara F., Jaillet F., Moreau J.-M.: An Implicit Tensor-Mass solver on the GPU for soft bodies simulation. In *Eurographics Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS)* (Dec. 2012), pp. 1–10. 1, 2

[SDR*05] Schwartz J., Denninger M., Rancourt D., Moisan C., Laurendeau D.: Modelling liver tissue properties using a non-linear visco-elastic model for surgery simulation. *Medical Image Analysis 9*, 2 (2005), 103–112. 1



(a) (b) (c) (d)

Figure 2: *Simulation of a cantilever beam. The green-red range stands for max* ($\mathcal{R}_L$,$\mathcal{R}_A$) $\in [0,1]$*; and blue when the quad element has shifted from Hooke to Saint Venant-Kirchhoff constitutive law.*



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3: *Gravitational force applied to a beam fixed on its top. Successive refinements (a-h) of the mesh according to the* R_S *criterion. At each step, triangles overpassing* 90 % *of* $max(R_S)$ *are subdivided.*