HAL
open science

Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links

François Goulette, Zhuo-Wei Chen

To cite this version:

François Goulette, Zhuo-Wei Chen. Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links. Computer Methods in Applied Mechanics and Engineering, 2015, 10.1016/j.cma.2015.06.015 . hal-01256777

HAL Id: hal-01256777

https://minesparis-psl.hal.science/hal-01256777v1

Submitted on 15 Jan 2016

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

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

Fast computation of soft tissue deformations in real-time simulation with Hyper-Elastic Mass Links

François Goulette[a,*], Zhuo-Wei Chen[a]

[a] *Mines ParisTech, PSL Research University, Centre de Robotique, 60 Bd St Michel 75006 Paris, France*

Abstract

Virtual surgery simulators show a lot of advantages in the world of surgery training, where they allow to improve the quality of surgeons' gesture. One of the current major technical difficulties for the development of surgery simulation is the possibility to perform a real-time computation of soft tissue deformation by considering the accurate modeling of their mechanical properties. However today, few models are available, they are still time consuming and limited in number of elements by algorithm complexity. We present in this paper a new method and framework that we call 'HEML' (*Hyper-Elastic Mass Links*), which is particularly fast. It is derived from the finite element method, can handle visco-hyperelastic and large deformation modeling. Although developed initially for medical applications, the HEML method can be used for any numerical computation of hyperelastic material deformations based on a tetrahedral mesh. A comparison with existing methods shows a much faster speed. A comparison with Mass-Spring methods, that are particularly fast but not realistic, shows that they can be considered as a degenerate case of the HEML framework.

Keywords: Real-time simulation, Hyperelasticity, Soft tissue, HEML

*Corresponding author. Tel.: +33 1 40 51 92 35; Fax: +33 1 43 26 10 51.
Email address: `francois.goulette@mines-paristech.fr` (François Goulette)

© 2015, Goulette and Chen. This manuscript version is made available under1 the CC-BY-NC-ND 4.0 license
http://creativecommons.org/licenses/by-nc-nd/4.0/
DOI: http://dx.doi.org/10.1016/j.cma.2015.06.015

1. Introduction

The simulation of surgical systems which provides visual and haptic feedback using fast computational algorithms has developed rapidly in the last two decades to facilitate the surgical training process [1]. To have a real-time performance, these systems must compute the force of deformed soft tissue at rates of more than 25 Frames Per Second (FPS) for visual display (25 Hz) and provide it to the surgeon through haptic feedback at frequencies of at least 500 Hz [2].

In this work, we focus on the simulation of soft tissue deformation in the context of surgical simulators, which show many potential interests in educational, practical, ethical and economical issues. The real-time computation of soft tissues may have many applications in several medical domain, such as visceral surgery, brain surgery, gynaecology and childbirth, urology. The real-time constraint on such simulators depends on the minimal computation time for discrete differential equations that accurately model the mechanical properties of soft tissues. However, the complexity of soft biological tissues' mechanical behavior makes the simulation in real-time a very challenging task. In gynecology, the vaginal tissue, for instance, shows a nonlinear relationship between stress and strain levels and a visco-hyperelastic behavior [3, 4, 5].

Several computational methods and models have been developed to simulate soft tissue deformations in real-time. Early model used linear stress-strain models and constrained themselves to small deformations to calculate a fast solution. In surgery simulation, scientists have mainly focused on the mass-spring models [6, 7, 8], due to their simplicity of implementation and their low computation complexity properties. However, they suffer from a lack of realism, which led to further research on extensions of the model. Cotin et al. [9] proposed a so-called mass-tensor model which is based on continuum mechanics and linear elasticity theory; this model has been developed further to handle large deformations and large displacements with the Saint Venant-Kirchhoff constitutive law [10, 11]. However, this model is limited to a specific material. The requirement of a nonlinear geometric and material algorithm for soft tissue sim-

ulation led to the work of Miller et al. [12], in which they presented a Total Lagrangian Explicit Dynamic (TLED) algorithm based on the Finite Element Method (FEM). This algorithm shows better mathematical performance in each time step. An algorithm allowing real time computation of geometric nonlinearity for virtuel surgery simulation by using the point collocation-based method of finite spheres (PCMFS) is proposed by Lim et al. [13], however, a linear stress-strain law is used in this context. Other possible approaches as the Proper Orthogonal Decomposition (POD) and the Proper Generalized Decomposition (PGD) methods can be found in [14, 15]. More recently, Marchesseau et al. [16] proposed a Multiplicative Jacobian Energy Decomposition (MJED) method for discretizing hyperelastic materials on linear tetrahedral meshes which leads to faster matrix assembly than the standard FEM. Though this approach is not limited to one specific hyperelastic material but can not reach the ideal 25 frames per second needed for the real-time simulation. Other studies of deriving discrete computational algorithms from the equations of continuum mechanics based on the FEM which tend to obtain real-time computations can be found in ([17, 18, 19, 20, 21]). However, the heavy complexity of these methods makes computation time a real challenge.

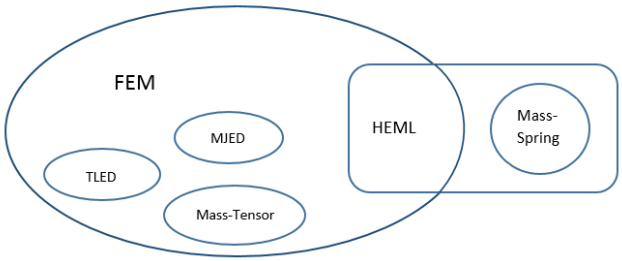
For these reasons, we tried to find a compromise between biomechanical accuracy and computational efficiency to realize a real-time simulator. We propose to design a fast algorithm to compute the elastic force field for any hyperelastic model, handling large deformations and large displacements. The algorithm is designed under the P1-finite element approximation in homogeneous isotropic cases. Hyperelastic models include the Saint Venant-Kirchhoff constitutive law (used in mass-tensor), and other important hyperelastic constitutive laws such as Neo-Hookean and Mooney-Rivlin. We chose to call this approach *Hyper-Elastic Mass Link* (HEML), for the following reasons: '*Link*', because forces at a given node are given as a sum of forces function of the links (vectors) to all connected neighbors. '*Mass*': as in mass-spring or mass-tensor, masses are affected to the mesh nodes, used in the discrete differential equations. '*Hyper-Elastic*', because the framework presented may be used to design algorithms for

computation of any hyperelastic material. We propose a schema to illustrate the position of HEML method compared to other methods (Figure 1). HEML is an extension of the mass-spring method. The mass-spring method may be seen as a degenerate case of HEML (demonstrated later). In the meantime, HEML is derived from the finite element method which includes many other methods like mass-tensor, TLED, MJED, etc. The novel contribution in the HEML method is that the computations are based on local displacements, as in mass-spring method. This feature presents advantages in terms of numerical robustness for large displacements of objects.

The HEML method is generated with the idea of constructing a physically realistic mechanical model of soft tissue for an educational simulator equipped with a haptic device. In such a case, the model should be as realistic as possible, and efficient enough to allow the real-time performance. This method is extremely fast to calculate the force field, and is applicable for the tetrahedral mesh due to the specific characteristic of constant deformation gradient tensor over a given tetrahedron for P1 element which will be used in our method. It should be noticed that this kind of element does not provide the best accuracy for stress or strain, but this is not considered as the first priority for real-time applications. On the other hand, the property of a constant deformation gradient tensor over a given tetrahedron in the P1 approximation leads to a closed-form expression of the forces with respect to the links (edges) of the mesh, for any hyperelastic material. As it is based on the finite element method by using the strain energy density function w, which is a scalar function of one of the strain tensors, it can handle the non-linear and large deformation problems. It can also handle easily parallelization, and element removal. Unlike the mass-tensor method, HEML is not limited to only one hyperelastic material. A preliminary version of the principle was initially published in [22].

This paper is structured as follows. Some existing numerical methods and their comparisons with HEML will be presented in the next section. The framework of HEML method will be presented in Section 3. In Section 4, we present the specific hyperelastic materials by showing their equations in form of HEML

method and their implementation. Some computational examples including their validation and comparison are presented in Section 5. Section 6 contains conclusions and discussion.



9 Figure 1: The HEML method compared to other frameworks

2. Existing numerical methods

Table 1 and 2 summarize the comparisons of forces computation and dynamic computation of these different methods. The equations will be explained shortly.

In table 1,

- Mass-spring: $\mathbf{f}_i$ represent forces at each node i, which are proportional to link elongation. $N(\mathbf{P}_i)$ is the set of vertices $\mathbf{P}_j$ adjacent to $\mathbf{P}_i$, k_{ij} is the stiffness coefficient between vertices $\mathbf{P}_i$ and $\mathbf{P}_j$, l_{ij}^0 is the rest length between $\mathbf{P}_i$ and $\mathbf{P}_j$.

- Mass-tensor: the elastic force $\mathbf{f}_i$ is the sum of all contributions from all adjacent tetrahedra T_i for a given vertex $\mathbf{P}_i$. $[\mathbf{K}_{ii}]$ is the sum of stiffness tensors associated to the tetrahedra adjacent to $\mathbf{P}_i$, $[\mathbf{K}_{ij}]$ is the sum of stiffness tensors associated to the tetrahedra adjacent to edge (i, j) and $N(\mathbf{P}_i)$ is the list of $\mathbf{P}_i$ neighbors.

- TLED: $\mathbf{f}$ is the nodal force contribution from each element. Here, v^0 is the initial volume of the element, $\widetilde{\mathbf{S}}$ is the vector form of the second Piola-Kirchoff stress tensor $\mathbf{S}$.

- MJED: $\mathbf{f}$ is the force at each node Q_i, W_h is the strain energy which is decomposed into two terms $f(J)$ and $g(\widetilde{I})$, where $\widetilde{I} = (I_1, I_2, I_4...)$ are the invariant of $\mathbf{C}$ (see section 3.2), therefore g is independent of Jacobian J, its derivative will not involve any matrix inversions.

- HEML: $\mathbf{f}_{i,k}$ is the contribution of force of tetrahedron T_k at node m_i, the strain energy W_k is a function of squared edge length vector $\mathbf{l}_k$, more details can be found in the next section.

In Table 2, for the mass-spring, mass-tensor and HEML methods, the global system of discretized equations of motion are based on the nodal positions (x_i or $\mathbf{P}_i$); for the TLED method, it is based on the nodal displacements $\mathbf{u}$.

9 Table 1: Comparison of internal forces computation

Methods	Internal forces	Computation time	Generality of constitutive laws	Calculation with respect to the global/local displacement
Mass-spring	$\mathbf{f}_i = \sum_{j \in N(\mathbf{P}_i)} k_{ij}(\lVert \mathbf{P}_i\mathbf{P}_j \rVert - l_{ij}^0)\frac{\mathbf{P}_i\mathbf{P}_j}{\lVert \mathbf{P}_i\mathbf{P}_j \rVert}$	Real-time	Not based on physical constitutive laws	Local
Mass-tensor	$\mathbf{f}_i = [\mathbf{K}_{ii}]\mathbf{P}_i^0\mathbf{P}_i + \sum_{j \in N(\mathbf{P}_i)} [\mathbf{K}_{ij}]\mathbf{P}_j^0\mathbf{P}_j$	Real-time only in linear form	- Linear elasticity - Saint-Venant Kirchhoff for non-linear	Global
TLED	$\mathbf{f} = \int_{v0} \mathbf{B}^T \widetilde{\mathbf{S}} dV$	Real-time with GPU implementation	Available for all constitutive laws	Global
MJED	$\mathbf{f} = \frac{\partial W_h}{\partial Q_i}$, $W_h = V_0 \sum_{k=1}^{n} f^k(J) g^k(\widetilde{I})$	Not real-time	Available for all constitutive laws	Global
HEML	$\mathbf{f}_{i,k} = \frac{\partial W_k(\mathbf{I}_k)}{\partial m_i}$	Real-time	Available for all constitutive laws	Local

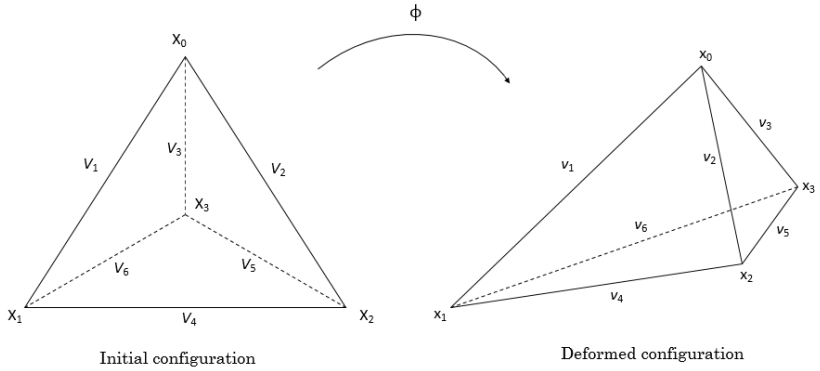
9 Table 2: Comparison of dynamic computation

Methods	Dynamic formulation	Computation time	viscosity	Inverted element
Mass-spring	$m_i\ddot{x}_i + \gamma_i\dot{x}_i + f_{int_i} = f_{ext_i}$	Real-time	Rayleigh damping	No
Mass-tensor	$m_i \frac{d^2\mathbf{P}_i}{dt^2} = \gamma_i \frac{d\mathbf{P}_i}{dt} + \mathbf{F}_i$	Real-time only in linear form		No
TLED	$\mathbf{M}\ddot{\mathbf{u}}_n + \mathbf{K}(\mathbf{u}_n) \cdot \mathbf{u}_n = \mathbf{R}_n$	Faster than FEM	Prony series	No
MJED	not explained	Faster than FEM	Prony series	Yes
HEML	$m_i\ddot{x}_i + \gamma_i\dot{x}_i + f_{int_i} = f_{ext_i}$	Real-time	Prony series	Yes

3. The *Hyper-Elastic Mass Link* framework

3.1. Notations for tetrahedral mesh

In the HEML method, the modeled object is discretized into a tetrahedral mesh as defined by finite element theory. Consider a given tetrahedron, the four vertices can be denoted as X_i for initial configuration and x_i $(0 \leq i \leq 3)$ for deformed configuration (Figure 2). The six edge vectors are denoted as $\boldsymbol{V}_i$ and $\boldsymbol{v}_i$ $(1 \leq i \leq 6)$. Any three edge vectors out of six are enough to express the others, so we denote $\mathbf{V}$ and $\mathbf{v}$ the matrices: $\mathbf{V} = (\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3)$, $\mathbf{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$. Considering non-degenerate tetrahedra in initial state, $\mathbf{V}$ is invertible. We denote L_i the lengths of edges $\boldsymbol{V}_i$, l_i the lengths of edges $\boldsymbol{v}_i$, $\mathbf{L}$ and $\mathbf{l}$ the vectors of the squared six lengths: $\mathbf{L} = (L_1^2, L_2^2 ... L_6^2)^T$, $\mathbf{l} = (l_1^2, l_2^2 ... l_6^2)^T$, and $\Delta\mathbf{l} = \mathbf{l} - \mathbf{L}$ the vector of differences.



9 Figure 2: Initial and deformed configurations

3.2. Potential energy

Rubber and some biological soft tissue materials are said to be hyperelastic [23, 24, 25]. Usually, these kind of materials undergo large deformations. In order to describe the geometrical transformation problems, the deformation gradient tensor is introduced by :

$$\mathbf{F} = \mathbf{I} + \nabla\mathbf{u} \tag{1}$$

where $\mathbf{I}$ is the unity tensor. $\mathbf{u}$ is the displacement vector.

Because of large displacements and rotations, Green-Lagrangian strain is adopted for the non-linear relationships between strains and displacements. We note $\mathbf{C}$ the stretch tensor or the right Cauchy-Green deformation tensor ($\mathbf{C} = \mathbf{F}^T\mathbf{F}$).

In the case of hyperelastic law, there exists an elastic potential function w (or strain energy density function) which is a scale function of one of the strain tensors. In this work, we consider a homogeneous, isotropic and hyperelastic material. The assumption of isotropy of the material allows to write the potential w as a function of only the three invariants of $\mathbf{C}$:

$$w = w(I_1, I_2, I_3) \qquad \text{where} \qquad \begin{aligned} I_1 &= tr(\mathbf{C}), \\ I_2 &= \frac{1}{2}\big({I_1}^2 - tr(\mathbf{C}^{\,2})\big), \\ I_3 &= det(\mathbf{C}) \end{aligned} \tag{2}$$

In the "P1 approximation" of finite elements, mesh elements are tetrahedral; the approximation states that the deformation gradient tensor $\mathbf{F}$ is constant over a given tetrahedron T_k, it is also the same case for tensor $\mathbf{C}$. Defining W the total energy of the material, being equal to the integral of w over the mesh volume, and defining W_k the energy of the material over the tetrahedron T_k, being equal to the integral of w over the tetrahedron volume, we can decompose W over each tetrahedron T_k, being the sum of W_k over all the tetrahedra. It can be demonstrated that (see Appendix A), under the P1 approximation, the value of $\mathbf{C}$ depends linearly on the vector $\mathbf{l}$ (squared edge length). Hence, the energy W_k for each tetrahedron T_k depends only on $\mathbf{l}$, which itself depends only on the vertices positions:

$$W_k = W_k(\mathbf{l}) = W_k(x_0, x_1, x_2, x_3) \tag{3}$$

3.3. Hamilton's principle for hyperelastic materials in P1 approximation

In the HEML method, the resulting system is solved dynamically. The motion of a system is defined by the position vector of each particle in a function

of time t. For a system of N particles, as time unfolds, the motion of the entire system of particles traces a curve in $3N$ dimensional space known as configuration space, called the true path. A different path defined by virtual displacement δx_i is known as the varied path. We consider only the paths that coincide with the true path at time t_1 and t_2, and it follows that $\delta x(t_1) = \delta x(t_2) = 0$. Under these conditions, the Hamilton's principle is given as:

$$\int_{t_1}^{t_2} (\delta L + \delta W) dt = 0 \tag{4}$$

where L is the Lagrangian, given by the kinetic energy of the particle T minus the potential energy of the particle V. The notation δ implies a variation in the system - an imaginary change of configuration that complies with the system constraints. For a system of n particles, δT is defined as

$$\delta T = \sum_{i=1}^{n} m_i \ddot{x_i} \delta x_i \tag{5}$$

where m_i is the particle mass and x_i is the position vector of the particle. For a visco-hyperelastic material in P1 approximation, the potential energy includes hyperelastic potential energy and dissipation potential energy, its variation is given by

$$\delta V = -(\sum_{i=1}^{n} \frac{\partial W^h}{\partial x_i} \delta x_i + \frac{\partial D}{\partial \dot{x_i}} \delta x_i) \tag{6}$$

and the variation of the work done by external forces undergoing a virtual displacement is

$$\delta W = -\sum_{i=1}^{n} f_{ext_i} \cdot \delta x_i \tag{7}$$

With Equations (5),(6) and (7), the Hamilton's principle can be written as

$$\sum_{i=1}^{n} \int_{t_1}^{t_2} (m_i \ddot{x_i} + \frac{\partial W^h}{\partial x_i} + \frac{\partial D}{\partial \dot{x_i}} - f_{ext_i}) \delta x_i dt = 0 \tag{8}$$

then, the motion of each particle i to be solved at each time step is

$$m_i\ddot{x}_i + \frac{\partial W^h}{\partial x_i} + \frac{\partial D}{\partial \dot{x}_i} - f_{ext_i} = 0 \tag{9}$$

The homogeneity of the masses is considered in our calculation. The idea is to take into account the density of the body and calculate the mass of a tetrahedron k which depends on its volume Vol_k, and then distribute the mass on the four vertices of the tetrahedron. So the mass of each vertex depends on all the tetrahedra containing this point: $m_i = \rho\Sigma_k \frac{1}{4}\mathrm{Vol}_k$. This formulation corresponds to the lumped mass matrix approximation used in Finite Element Methods [26]. This approximation allows to perform easily an update of meshes in case of cutting or modification, without need of matrix inversion. It has an impact only on the dynamic deformation but not on the solution at rest.

3.4. Internal hyperelastic forces

From the Equation (9), we can see that the internal force caused by hyperelastic energy can be computed directly by deriving the energy with respect to the nodal position, instead of computing it by using the first derivative of the energy with respect to $\mathbf{C}$ (leading to the second Piola-Kirchoff stress tensor $\mathbf{S}$):

$$f_{int}(x_i) = -\frac{\partial W}{\partial x_i} \tag{10}$$

For a given tetrahedron T_k, the force at a node x_i can be expressed by:

$$f_{i,k} = -\frac{\partial W_k}{\partial x_i} \tag{11}$$

$f_{i,k}$ is the contribution of the force of tetrahedron T_k at node x_i. It is a vector of dimension 3.

As we mentioned before, W_k is a function of the vector $\mathbf{l}$, then Equation (11) can be expressed by the following equation:

$$\frac{\partial W_k(\mathbf{l}_k)}{\partial x_i} = \frac{\partial W_k}{\partial \mathbf{l}_k}\frac{\partial \mathbf{l}_k}{\partial x_i} \tag{12}$$

From the relation between the squared edge lengths l_i^2 and the edge vectors $\boldsymbol{v}_i$, one can express the squared edge lengths by the four nodes. This leads to a

formulation of the derivative of $\mathbf{l}_k$ over each vertex x_i of the tetrahedron, linear to the matrix $\mathbf{v}$, with 4 (6×3) constant matrices $\mathbf{DLM}_i$ that depend on the initial state (see Appendix B):

$$0 \leq i \leq 3, \frac{\partial \mathbf{l}_k}{\partial x_i} = \mathbf{DLM}_i \cdot \mathbf{v}_k \tag{13}$$

Then, after calculating the derivative of w with respect to $\mathbf{l}$, which depends on the material constitutive law, the force of each vertex x_i of tetrahedron T_k can be expressed by a linear function of the vectors of the six squared lengths, and of the links (see Appendix C).

3.5. Visco-hyperelasticity

In general, biological soft tissues are characterized by Fung as "bio-viscoelastic solids" which manifest a time-dependent mechanical behavior [27]. The nonlinear viscoelastic response has been found in many biological tissues by means of experimental tests (see for example [28] for rabbit ligaments, [29] for blood vessels, [30] for aortic valves). To have a full description of the mechanical response of biological soft tissues, it requires including the nonlinear viscoelastic behavior.

Without loss of generality regarding the computation of elastic forces, we chose to build a visco-hyperelastic constitutive model relying on Prony series [31, 32], in which a time dependent term is added to the underlying hyperelastic formulation. The time-dependent strain energy function $\hat{W}$ can be expressed in the form of a convolution time integral:

$$\hat{W}(W^h, t) = \int_0^t \alpha(t-s) \frac{\partial W^h}{\partial s} \mathrm{d}s \tag{14}$$

where t is the time. The relaxation functions $\alpha(t)$ is given by a Prony series:

$$\alpha(t) = \alpha_\infty + \sum_{i=1}^{n} \alpha_i e^{-t/\tau_i} \tag{15}$$

where α_∞, α_i and τ_i are positive real constants. By imposing the condition $(\alpha_\infty + \sum_{i=1}^{n} \alpha_i) = 1$, we may rewrite Equation (15) as:

$$\alpha(t) = 1 - \sum_{i=1}^{n} \alpha_i (1 - e^{-t/\tau_i}) \tag{16}$$

The visco-hyperelastic force $\widehat{f}$ can be written as:

$$\widehat{f} = -\frac{\partial \hat{W}}{\partial x} = -\int_0^t \alpha(t-s) \frac{\partial f}{\partial s} \mathrm{d}s \tag{17}$$

By using Equation (16), Equation (17) can be restated as:

$$\widehat{f} = -\int_0^t [1 - \sum_{i=1}^{n} \alpha_i (1 - e^{(s-t)/\tau_i})] \frac{\partial f}{\partial s} \mathrm{d}s = -(f - \sum_{i=1}^{n} \gamma_i) \tag{18}$$

where

$$\gamma_i = \int_0^t \alpha_i (1 - e^{(s-t)/\tau_i})] \frac{\partial f}{\partial s} \mathrm{d}s \tag{19}$$

This integral equation can be converted into an incremental update formula after a discretization over time:

$$\gamma_i^n = A_i f^n + B_i \gamma_i^{n-1} \tag{20}$$

with $A_i = \Delta t \alpha_i / (\Delta t + \tau_i)$ and $B_i = \tau_i / (\Delta t + \tau_i)$. Δt is the time step. Superscripts are added to indicate time increment.

4. Specific materials and implementation

4.1. Isotropic hyperelastic models

HEML may be seen as a method, but also as a framework for deriving specific equations and implementations for specific materials. We have used this approach to derive elastic forces for various specific isotropic hyperelastic materials: Saint Venant-Kirchhoff, Neo-Hookean and Mooney-Rivlin. Saint Venant-Kirchhoff model is a hyperelastic material model which is an extension of the linear elastic material model to the nonlinear regime. The volumetric energy function is usually formulated with the Green-Lagrange tensor $\mathbf{E}$, and the Lamé coefficients λ and μ. Neo-Hookean [33] and Mooney-Rivlin [34, 35] are

popular hyperelastic material models that can be used for predicting the non-linear stress-strain behavior of rubber or biomechanical materials undergoing large deformations [36, 37, 38]. For each type of hyperelastic model, the energy density is a function of the three invariants I_1, I_2, I_3, and we express it by the squared edge lengths $\mathbf{l}$ or $\Delta\mathbf{l}$. Then we derive a closed-form expression of forces with respect to $\mathbf{l}$ or $\Delta\mathbf{l}$. We present in Table 3 the energy density and forces per unit volume derived at the nodes of a given tetrahedron, for each material models. It should be noticed that the equations presented in this table do not take into account the material's compressibility which will be discussed later.

9Table 3: Energy and forces per unit volume for different hyperelastic material models

Material model	Energy density w	Force per unit volume at vertex i
General type	$w(\mathbf{l}) = w(I_1, I_2, I_3)$	$\frac{\partial w}{\partial \mathbf{l}} \frac{\partial \mathbf{l}}{\partial x_i}$
Saint Venant -Kirchhoff	$\frac{\lambda}{2}(tr(\mathbf{E}))^2 + \mu tr(\mathbf{E}^2)$ $= \Delta\mathbf{l}^T \times \mathbf{MW}_{STVK} \times \Delta\mathbf{l}$	$\Delta\mathbf{l}^T \times \mathbf{MW}_{STVK}(\mathbf{DLM}_i \otimes \mathbf{v})$
Neo-Hookean (Incompressible)	$C_1(I_1 - 3) = C_1(\mathbf{vtr} \cdot \mathbf{l} - 3)$	$C_1 \mathbf{vtr}^T(\mathbf{DLM}_i \otimes \mathbf{v})$
Mooney-Rivlin (Incompressible)	$C_1(I_1 - 3) + C_2(I_2 - 3)$ $= C_1(\mathbf{vtr} \cdot \mathbf{l} - 3)+$ $C_2[\frac{1}{2}\mathbf{l}^T(\mathbf{vtr} \otimes \mathbf{vtr} - \mathbf{Mtr})\mathbf{l} - 3]$	$C_1 \mathbf{Vtr}^T(\mathbf{DLM}_i \otimes \mathbf{v})+$ $C_2 \mathbf{l}^T \times \mathbf{MW}_{M-R}(\mathbf{DLM}_i \otimes \mathbf{v})$

For a tetrahedral element, we can express the right Cauchy-Green strain tensors in a formulation of the squared six lengths: $\mathbf{C} = \sum_{i=1}^{6} l_i \cdot \mathbf{C}_i$. The three invariants of tensor $\mathbf{C}$ can be expressed by the vector of squared edge lengths $\mathbf{l}$. In the Saint Venant-Kirchhoff model, $\mathbf{MW}_{STV-K}$ is a 6×6 matrix: $\mathbf{MW}_{STVK} = \frac{\lambda}{8}(\mathbf{vtr} \otimes \mathbf{vtr}) + \frac{\mu}{4}\mathbf{Mtr}$. For the Neo-Hookean and Mooney-

Rivlin models, C_1 and C_2 are material parameters, $\mathbf{MW}_{M-R}$ is a 6×6 matrix: $\mathbf{MW}_{M-R} = \frac{1}{2}(\mathbf{vtr} \otimes \mathbf{vtr} - \mathbf{Mtr})$, In which **vtr** and **Mtr** are the 6-vector of traces of $\mathbf{C}_i$ and the 6×6-matrix of traces of $\mathbf{C}_i\mathbf{C}_j$ respectively (see Appendix C).

4.2. Compressibility

When a certain compressibility is involved in the kinematics of deformation, the deformation is separated in a volumetric part $\mathbf{F}^{vol}$ and an isochoric part $\mathbf{F}^{iso}$ by a multiplicative split [39], for numerical purpose:

$$\mathbf{F} = \mathbf{F}^{iso}\mathbf{F}^{vol} \tag{21}$$

where

$$\mathbf{F}^{vol} = J^{1/3}\mathbf{I}, \quad \mathbf{F}^{iso} = J^{-1/3}\mathbf{F} \tag{22}$$

in which J is the determinant of $\mathbf{F}$.

This decomposition is such that $\det(\mathbf{F}^{iso}) = 1$. It is easy to see that $\mathbf{F}$ and $\mathbf{F}^{iso}$ have the same eigenvectors. The isochoric part of the right Cauchy-Green strain tensor $\mathbf{C}$ can be then defined as

$$\mathbf{C}^{iso} = J^{-2/3}\mathbf{C} \tag{23}$$

The first strain invariant of $\mathbf{C}^{iso}$ is defined by $\hat{I}_1 = J^{-2/3}I_1$ which replaces I_1 in $w(I_1)$. The modified invariants related to $\hat{\mathbf{C}} = \hat{\mathbf{F}}^T\hat{\mathbf{F}} = J^{-2/3}\mathbf{C}$ are expressed by

$$\hat{I}_1 = I_1 J^{-2/3}, \;\; \hat{I}_2 = I_2 J^{-4/3} \tag{24}$$

The strain energy density function then can be decomposed into a volumetric part and an isochoric part [40, 41, 42]:

$$w = w_{iso}(\hat{I}_1, \hat{I}_2) + w_{vol}(J) \tag{25}$$

Generally, the soft biological tissues are assumed to be near incompressible. The term $w_{vol}(J)$ represents usually a penalty added to the finite element model

to account for the incompressible behavior of the material [43, 44, 45]. In the Saint Venant-Kirchhoff case, the material is compressible: incompressibility is a special, limit case, when the Poisson coefficient tends towards 0.5, which is in practice not possible for Lamé coefficients to become infinite. According to Bonet and Wood [46], the Saint Venant-Kirchhoff material is useless for modeling large deformations due to the fact that during a compression, it gets weaker and weaker as the compression increases, and the stress drops to zero as the object becomes flat [47].

4.3. HEML formulation for compressible Neo-Hookean material

HEML is a framework to derive numerical algorithms for hyperelastic materials. In this section, we detail and summarize how to derive a numerical algorithm for a compressible Neo-Hookean material.

For the potential energy of the Neo-Hookean material, several formulations can be used. We can use the formulation defined in [39, 48] as:

$$w(I_1, I_2, J) = C_1(I_1 - 3) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2 \tag{26}$$

With C_1 a coefficient regarding the Neo-Hookean elasticity, μ and λ coefficients regarding compressibility. In this formulation, I_3 is not directly used but appears as J for the compressibility term.

We now describe how to derive the computation of forces from this formulation. From Equation 26, we first need to derive in a closed form the formulation of w with respect to I_1, I_2 and J :

$$\begin{aligned} \frac{\partial w}{\partial I_1} &= C_1 \\ \frac{\partial w}{\partial I_2} &= 0 \\ \frac{\partial w}{\partial J} &= -\tfrac{\mu}{J} + \lambda \tfrac{\ln J}{J} \end{aligned} \tag{27}$$

For each tetrahedron T_k, the energy W_k can be expressed as a function of the links (vector l), as seen above (see Section 3.2). As C is constant over the tetrahedron, the energy W_k si equal to the energy density multiplied by the initial volume of the tetrahedron Vol_k :

$$W_k(I_1, I_2, J) = Vol_k w(I_1, I_2, J) \tag{28}$$

The internal forces $f_{i,k}$ at nodes i regarding the tetrahedron T_k are given by the partial derivatives of W_k with respect to x_i (see Equation 11). Using the combined derivations with respect to (I_1, I_2, J), to l and to x, these partial derivatives are equal to:

$$0 \leq i \leq 3, \frac{\partial W_k}{\partial x_i} = Vol_k \left(\frac{\partial w}{\partial I_1} \frac{\partial I_1}{\partial l} \frac{\partial l}{\partial x} + \frac{\partial w}{\partial I_2} \frac{\partial I_2}{\partial l} \frac{\partial l}{\partial x} + \frac{\partial w}{\partial J} \frac{\partial J}{\partial x} \right) \tag{29}$$

The partial derivatives of the energy w with respect to I_1, I_2 and J have been computed in closed form and their equations are given above (Equation 27). The partial derivatives of I_1 and I_2 with respect to l are given in Appendix C (Equations C.2 and C.3). The partial derivatives of l with respect to x are given in Equation 13. The partial derivatives of J with respect to x are given inAppendix C (Equation C.4).

Hence, the partial derivatives of W_k with respect to x_i can be computed in closed form:

$$0 \leq i \leq 3, \frac{\partial W_k}{\partial x_i} = Vol_k C_1 \mathbf{vtr}^T (\mathbf{DLM}_i \otimes \mathbf{v}) + 0 + \frac{\lambda \ln J - \mu}{J} \frac{\partial vol}{\partial x_i} \tag{30}$$

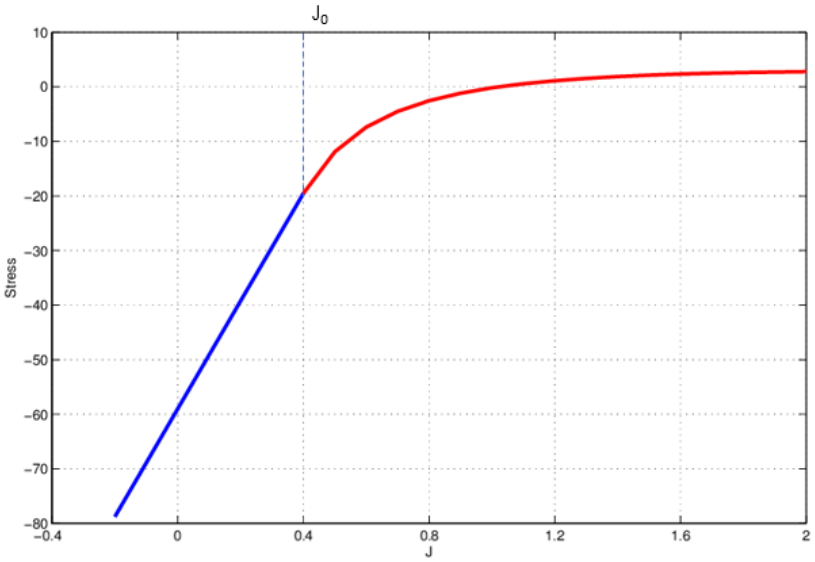
This equation provides the expression of the (opposite of the) force $f_{i,k}$, to be implemented further in the algorithm as will be seen below.

This process can be followed for any constitutive law expressed with an energy potential, the distinctive part for other materials being the equations of partial derivatives (Equation 27). See above in Section 4.1 and Table 3 the expressions of forces for some common constitutive laws.

4.4. Inverted element

From the physical point of view, J is always positive, otherwise it means that the body could be flattened ($J = 0$) or interpenetrated ($J < 0$) which are not physical at all. The standard finite element method cannot handle flat or inverted elements. This problem has been studied in the framework of hyperelasticity theory, see e.g. [49]. Using off-line computation, some solutions have been proposed [50]. However, extreme compression and even inversion of tetrahedra can occur in some interactive simulations. To have a robust simulation, it is important to propose an approach that can handle nearly flat and even inverted elements in such situations.

Let us consider the expression of compressible Neo-Hookean material already used above, and its compressibility term. Under uniaxial unconstrained tension, the relationship between the first Piola-Kirchhoff stress P_{11} and J is $\mathrm{P}_{11} = \frac{1}{6J}(\lambda \log J - \mu)$, which can be presented generally by the red curve in Figure 3. The singularity at the origin means that infinite stress is needed to flatten an element. To avoid instabilities due to this infinite stress, we modify the constitutive model to remove the singularity by linearizing at a given limit. Thus a threshold J_0 is defined for the J: when $J \leq J_0$, we substitute the stress by its tangent curve at J_0 (blue curve in Figure 3), the relationship becomes to linear: $\mathrm{P}_{11} = (\frac{\lambda - \lambda \log J_0 - \mu}{6J_0^2})J + \frac{2\lambda \log J_0 - \lambda}{6J_0}$. The new model (red curve and blue curve in Figure 3) can be even extended to pass the origin into the inverted regime. We set $J_0 = 0.4$ in our code.



9Figure 3: Representation of the first Piola-Kirchhoff stress in the tension direction P_{11} before and after substitution. Initial stress: red. Tangent curve: blue.

4.5. Semi-explicit integration scheme

Different integration schemes can be used to solve the dynamic system. Among the explicit methods, a scheme based on the Euler method is the most simple one because it uses limited first order development. Vertex velocities and displacements at time $t + dt$ are derived from those at time t:

$$\begin{aligned} v_{t+dt} &= v_t + \frac{f_t}{m} dt \\ u_{t+dt} &= u_t + v_t dt \end{aligned} \tag{31}$$

By depending on the updated velocity for the new position and not that of the previous step, we obtained the modified Euler scheme which we describe as a semi-explicit method:

$$v_{t+dt} = v_t + \frac{f_t}{m} dt$$
$$u_{t+dt} = u_t + v_{t+dt} dt \tag{32}$$

In practice, this modified Euler method is more stable than the previous method. It has been demonstrated by Provot in his thesis [51] that this method gives results better than that of Runge-Kutta 2. We implemented this integration method for the HEML model. Note that in this method, the force at time t contributes to the velocity at time $t + dt$.

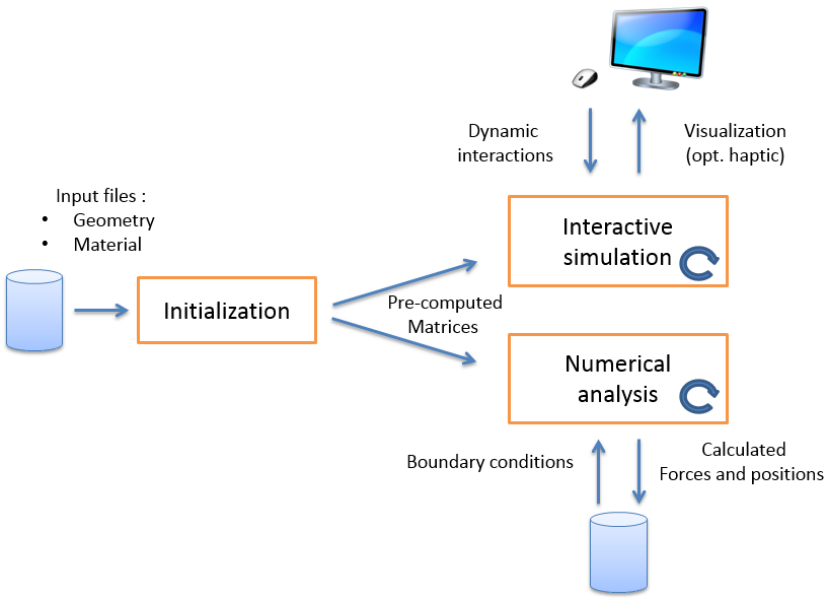
4.6. Implementation of HEML

Using the HEML framework, the formulations of forces for compressible Neo-Hookean materials, compressible Mooney-Rivlin materials, and Saint Venant-Kirchhoff materials (see Section 4.3 and Table 3) have been implemented.

The programming code, written in C++, is structured into three main modules: initialization; interactive simulation; numerical analysis (Figure 4). The initialization module sets the nodal displacement and nodal force, loads the input data, such as the element information and the material type. Another function for this module is to pre-compute some matrices for the subsequent computations. Once the boundary conditions are defined, the "numerical analysis" module calculates the forces and updates the nodal positions. With the "interactive simulation" module, we can visualize the deformation in a real-time environment; this module can be coupled with a haptic device. The "interactive simulation" and "numerical analysis" modules have parallel structures but manage interactions with users differently.

In order to compute an elastic force field on a material from the equations presented above, one has to perform three kinds of operations: an initialization regarding the initial state of the material (geometry and material parameters); a computation of forces in deformed state, at each step of the numerical integration process; an integration process to get the current points velocity and

position. We present these three modules by pseudo-code. Concerning initialization (Algorithm 1), one has to load the mesh, initialize the variables and precompute some quantities. Concerning the force field in deformed state (Algorithm 2), for each tetrahedron one has to compute the six edge vectors and squared lengths, and the four forces at each vertex using the corresponding equations given in Table 3. In the end one has to sum up, for each node of the element, the forces coming from incident tetrahedra. Concerning integration (Algorithm 3), the velocities and positions at time $t + dt$ are obtained by using the force at time t. This algorithm has been implemented into a laptop, with the Visual C++ developing environment under Windows XP. The laptop was an Intel Core 2 Duo at 2.40 GHz, 3.45 Go RAM.



9 Figure 4: Software structure, data exchange and interactions

Algorithm 1 Initialization

Load the mesh and get the element information (tetrahedra, vertices, edges...)
Initialize the matrices $\mathbf{D}_k$ and $\mathbf{DLM}_i$ (see Appendix A, B)
for every tetrahedron j **do**
  Get the nodes position in the tetrahedron, X_i
  Calculate the matrix of three vectors $\mathbf{V}$ and its inversion
  Calculate the initial volume Vol
  for every edge k **do**
    Compute the matrices $\mathbf{C}_k$ according to the $\mathbf{C}_k = \mathbf{V}^{-T}\mathbf{D}_k\mathbf{V}^{-1}$ (see Appendix A)
    Compute the vector $\mathbf{Vtr}$ and the matrix $\mathbf{Mtr}$
  end for
end for

Algorithm 2 Calculate force for different constitutive laws

for every tetrahedron j **do**
  Get the current position in the tetrahedron x_i
  Get the current matrix of three edge vectors $\mathbf{v}$
  Compute the six squared edge lengths l_j^2
  for every vertex i **do**
    Compute the force f_i summing the contributions according to the specific equations of material in Table 3
  end for
end for

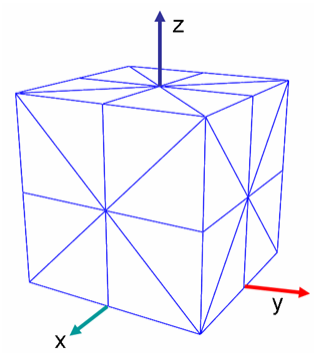
Algorithm 3 Integration

for every vertex i **do**
  Retrieve the nodal force f_t
  Get the current velocity v_{t+dt} according to Equation (32)
  Get the current position x_{t+dt} according to Equation (32)
end for

5. Analysis and comparison

5.1. Comparison with analytical solutions

To test the accuracy of the HEML method, a uniaxial unconstrained compression test is considered to conduct the comparison with analytical solutions. A cube model under uniaxially compression in the z direction, with a constant speed of 1 mm/s, is used in this test, as shown in Figure 5. As the test leads to homogeneous deformations, the equilibrium equations are automatically satisfied and the boundary conditions related to the free compression surfaces are: $\sigma_{xx} = \sigma_{yy} = 0$. For the compressible materials, under uniaxial compression, the principal stretches can be expressed by: $\lambda_z = \varepsilon, \lambda_x = \lambda_y = \sqrt{J/\varepsilon}$.



9 Figure 5: The cube model

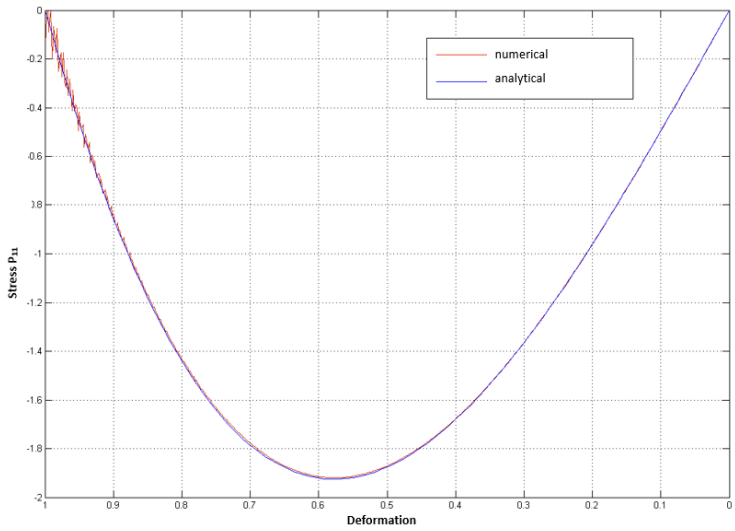
According to the definition of the first Piola-Kirchhoff stress tensor $\mathbf{P}$, which represents the current stress relative to the initial configuration surface, we can express the force at the compression surface due to the displacement like this

$$f_s = \mathrm{P}_{zz}\mathrm{S}_0 \tag{33}$$

in which P_{zz} is the first Piola-Kirchhoff stress in the compression direction, S_0 is the initial surface area.

For the Saint Venant-Kirchhoff material, the first Piola-Kirchhoff stress in the compression direction can be given as: $\mathrm{P}_{zz} = \frac{\mu(3\lambda+2\mu)}{2(\lambda+\mu)}(\varepsilon^3 - \varepsilon)$. With the

Equation (33), we can have the comparison of the force-deformation curves between the analytical result and the numerical result calculated by HEML method, as shown in Figure 6. The small oscillations seen at the left side of the figure are due to the viscosity term. It can be seen that the numerical curve matches well the analytical curve.

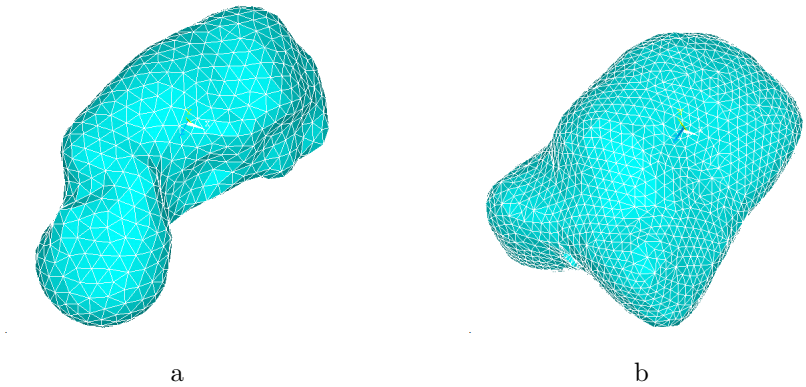


9 Figure 6: Comparison between the analytical and numerical results

5.2. Comparison with Finite Element Method

We considered two three-dimensional models (fetus and abdomen of the parturient woman) to conduct the comparison of computation time between HEML and Finite Element Method. The three-dimensional solid models were obtained from static images of MRI using free software segmentation ITK-SNAP based on the evolution of an active contour. The models, including geometry and material parameters, are provided by the SAARA Team of LIRIS of University of Lyon 1 as described in [52]. The fetus model includes 4430 tetrahedral elements and 1128 nodes (Figure 7a). For simplification sake, the fetus was considered as a homogenous material. For the mechanical behavior, it was modeled as Neo-Hookean material with a density of 950 kg/m^3 and $C_1 = 70kPa$. The abdomen model includes 21436 tetrahedral elements and 5591 nodes (Figure

7b). The Neo-Hookean constitutive law is chosen with a density of 2500 kg/m^3 and $C_1 = 5kPa$.



9 Figure 7: Mesh of the fetus and the abdomen of the parturient woman

For a complete computation of material deformations, we fixed some nodes on the objects and we added a displacement on a chosen node. The same models with same material parameters and same boundary conditions have been simulated by using a finite element research code FER [53]. The average computation time per iteration for the elastic force field are given in Table 4. It is noticeable that, for the fetus model, the computation can reach an average time of 4.02 ms. For the abdomen model, containing as many as 21436 tetrahedrons, the computation is feasible and can reach an average time of 21.24 ms. This corresponds to a rate of 47 Frame Per Second (FPS), which is above the minimum rate of 25 FPS required for real time simulation and visual display. Compared to the conventional finite element method, our algorithm shows a gain of more than 100 times speed up and can reach real-time simulation. This gain of time is due to the way we calculate the force as explained in Algorithm 2.

To have a quantitative comparison, three nodes have been randomly chosen for the fetus model to compare their displacements after deformation in the two methods. The absolute differences $\|u_i^{\texttt{HEML}} - u_i^{\texttt{FEM}}\|$ corresponding to each time steps are shown in Figure 8. It can be seen that the nodal displacements of the two methods are quite close to each other, the mean difference of nodal

9Table 4: Computation time per iteration for the fetus model and the abdomen model

	FER (ms)	HEML (ms)
Fetus (4430 elements)	761	4.02
Abdomen (21436 elements)	4070	21.24

displacements between the two methods is of order 10^{-3} mm. The change ratios $\frac{\|u_i^{\texttt{HEML}} - u_i^{\texttt{FEM}}\|}{u_i^{\texttt{FEM}}}$, at the last time step are $0.16\%, 0.18\%$ and 0.22% respectively for nodes 324, 481 and 521. This result validates the accuracy of the HEML method.

5.3. Comparison with Mass-Spring

The HEML formulation presents an interesting relationship with the Mass-Spring model, as it can be considered that a Mass-Spring model for a tetrahedral mesh is a simple, degenerated case of the HEML model.

The usual formulation of Mass Spring states that the elastic force related to the other nodes is parallel to their support and proportional to the elongation. The elastic force applied to a node x_0 by an elastic link between nodes x_0 and x_1 is given by:

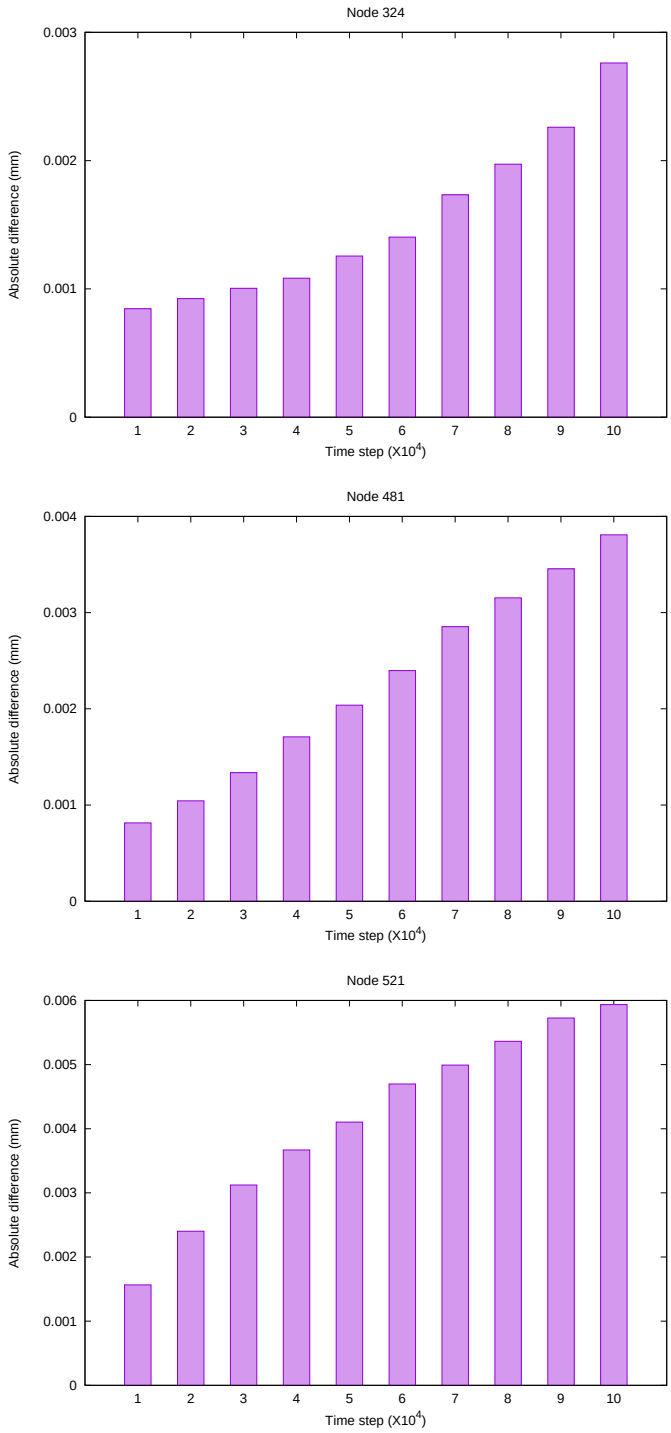
$$f_{x_0(1)} = k_1(l_1 - l_1^0)\frac{x_1 - x_0}{\|x_1 - x_0\|} \tag{34}$$

where l_1 is the distance between x_0 and x_1, l_1^0 is the distance at rest, k_1 is the proportional elastic constant of the spring.

Using the notation $\boldsymbol{v}_1$ for the link vector between x_0 and x_1, $\boldsymbol{v}_1 = x_1 - x_0$, Equation (34) reads:

$$f_{x_0(1)} = k_1(\frac{l_1 - l_1^0}{l_1})\boldsymbol{v}_1 \tag{35}$$

Let us consider a Mass-Spring model on a given node x_0 of a tetrahedron, for which the other nodes are denoted x_1, x_2, x_3 and the link vectors $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$,



9Figure 8: The absolute differences of nodal displacements $\|u_i^{\texttt{HEML}} - u_i^{\texttt{FEM}}\|$ with time step

and k_1, k_2 and k_3 are the elastic coefficients of the springs. The force applied at node x_0 reads:

$$\begin{aligned} f_{x_0} &= f_{x_0(1)} + f_{x_0(2)} + f_{x_0(3)} \\ &= k_1(\tfrac{l_1 - l_1^0}{l_1})\boldsymbol{v}_1 + k_2(\tfrac{l_2 - l_2^0}{l_2})\boldsymbol{v}_2 + k_3(\tfrac{l_3 - l_3^0}{l_3})\boldsymbol{v}_3 \end{aligned} \tag{36}$$

The dynamic formulation of a Mass-Spring model is a second order differential equation. At node x_0, it can be written as:

$$m_0\ddot{x_0} = f^{x_0} + f_D^{x_0} + f_{ext}^{x_0} \tag{37}$$

where m_0 is the mass associated to node x_0, and f^{x_0}, $f_D^{x_0}$, $f_{ext}^{x_0}$ are respectively the elastic force, the viscous (damping) force, and the external force applied to node x_0.

Let us now consider that our tetrahedral model is the discrete, P1 approximation of a continuous material model, for which an energy W_{MS}^h (MS for Mass-Spring) is expressed as:

$$W_{MS}^h = \frac{1}{2}k_1(l_1 - l_1^0)^2 + \frac{1}{2}k_2(l_2 - l_2^0)^2 + \frac{1}{2}k_3(l_3 - l_3^0)^2 \tag{38}$$

The HEML framework leads in the case where the formulation of internal forces as derivatives of W^h with respect to node positions (Equation 10):

$$f^{x_0} = -\frac{\partial W^h}{\partial x_0} \tag{39}$$

It can be easily demonstrated that:

$$\frac{\partial(l_1 - l_1^0)}{\partial x_0} = -\frac{1}{l_1}\boldsymbol{v}_1 \tag{40}$$

Hence, the formulation using the HEML framework and the energy W_{MS}^h leads to the following internal energy:

$$f_{x_0}^{HEML-MS} = k_1(\frac{l_1 - l_1^0}{l_1})\boldsymbol{v}_1 + k_2(\frac{l_2 - l_2^0}{l_2})\boldsymbol{v}_2 + k_3(\frac{l_3 - l_3^0}{l_3})\boldsymbol{v}_3 \tag{41}$$

This energy is identical to the classical formulation of spring forces (Equation 36). This demonstration for a given node of a given tetrahedron can be

easily extended to any node of any tetrahedral mesh, where the forces are linear combinations of incident forces.

It can be noted that the energy W^h_{MS} is not a regular expression as in the HEML case, as it uses the square of elongations where in the HEML case, the square of link distances are used. This is why this analogy can be seen as a degenerate case of the HEML method.

5.4. Comparison with Mass-Tensor

The HEML algorithm for Saint Venant-Kirchhoff derives from the same formula as the Mass-Tensors algorithm adapted to large deformations. However the formulation is not the same; a noticeable difference is that HEML uses relative displacements of nodes within the mesh, while Mass-Tensor uses displacements relative to original positions, $U_j = x_j - X_j$. This may lead to numerical drawbacks with large displacements, when the numerical values of displacements are orders of magnitude (x100, x1000) higher than positions. In order to reduce the number of operations in the expression of the Mass-Tensor algorithm, an arrangement is made, introducing matrices $\mathbf{B}_{pj}$, vectors $\mathbf{C}_{jkl}$ and scalars D_{jklp}, and summing up at each node from all contributors, that is the node p itself and its connected neighbors $N(p)$, leading to the following expression [10]:

$$\begin{aligned} F^p = \textstyle\sum_{j\in\{p,N(p)\}} \mathbf{B}_{pj} U_j + 4 \sum_{j,k,l\in\{p,N(p)\}^3} D_{jklp} U_l U_k^T U_j + \\ \textstyle\sum_{j,k\in\{p,N(p)\}^2} 2(U_k \times U_j^T)\mathbf{C}_{jkp} + (U_j^T \times U_k)\mathbf{C}_{pjk} \end{aligned} \tag{42}$$

The complexity of Mass-Tensors is related to the number of connected edges, up to a third power as can be seen on the Equation (42); for HEML, the complexity is linear in the number of tetrahedra. Analyzing further the equations, we have determined the exact number of operations (multiplications and additions) needed by both algorithms, to compute the same force field over a mesh from the node positions. To perform a comparison we have determined the number of operations in the case of a cube model with 125 nodes and 320 tetrahedral elements, and the result is presented in Table 5. We observe a significant improvement with a gain of more than 40 times by using HEML. It is

also important to notice that the number of operations for other kinds of hyperelastic materials, like Neo-Hookean and Mooney-Rivlin, is of the same order of magnitude as it is for Saint Venant-Kirchhoff, as one can see from the equations presented in Table 3. Additional comparison results can be found in our preliminary publication on HEML [22], which shows gain ratios of up to 200 times.

9Table 5: Number of operations (in thousands) and gain ratio for Mass-Tensor and HEML

	Mass-Tensor	HEML	Gain ratio
Cube	2999	70	43.0

6. Conclusion

We have presented the *Hyper-Elastic Mass Link* framework, a methodology for fast computation of deformable bodies that handles hyperelastic materials, is a compromise between biomechanical accuracy and computational efficiency. Equations of energy density and forces derived at each node of a given tetrahedron have been presented for three different hyperelastic material models (Saint Venant-Kirchhoff, Neo-Hookean and Mooney-Rivlin), visco-hyperelasticity and compressibility have also been discussed for these models. The complexity of the algorithms is linear in the number of tetrahedra. This algorithm has been validated by comparing the computed elastic force with an analytic solution. By comparing with the classical finite element method, mass-spring and mass-tensor method, the HEML method presents many advantages to respond to the need for surgery simulation. A much faster computation speed has been demonstrated on several experiments, this speed is fast enough to allow real time simulation.

Further works could be focused on several issues. The method could be extended to other hyperelastic materials, like Ogden material [54] or others [55] [56] [57], and to add plastic deformations; it could also be extended to non homogeneity, and to anisotropy like transversally isotropic materials [58]; to improve

computation time with specific coding for fast computation, or parallelization; to have a more abundant performance of the results (visualize and manipulate mesh data structures via the mouse, interactive animations, cutting planes or texture transparency), develop our algorithm on an open software platform; an extension of other mesh types, such as hexahedral meshes, could also be studied in the future.

Appendix A. Linear expression of tensor C w.r.t l

The "P1 approximation" states that the deformation gradient tensor $\mathbf{F}$ is constant over a given tetrahedron T_k, which leads to the relation between the edge vectors in deformed and initial states is: $1 \leq i \leq 6, \boldsymbol{v}_i = \mathbf{F}\,\boldsymbol{V}_i$. The squared edge lengths can be written as, for example for the l_1^2:

$$\begin{aligned} l_1^2 &= \boldsymbol{v}_1^T \cdot \boldsymbol{v}_1 \\ &= (\,\boldsymbol{V}_1^T \mathbf{F}^T) \cdot (\mathbf{F}\,\boldsymbol{V}_1) \\ &= \boldsymbol{V}_1^T (\mathbf{F}^T \mathbf{F})\,\boldsymbol{V}_1 \\ &= \boldsymbol{V}_1^T \mathbf{C}\,\boldsymbol{V}_1 \end{aligned} \tag{A.1}$$

So, all the squared lengths can be given like this:

$$\begin{cases} l_1^2 = \boldsymbol{V}_1^T \mathbf{C}\,\boldsymbol{V}_1 \\ l_2^2 = \boldsymbol{V}_2^T \mathbf{C}\,\boldsymbol{V}_2 \\ l_3^2 = \boldsymbol{V}_3^T \mathbf{C}\,\boldsymbol{V}_3 \\ l_4^2 = (\,\boldsymbol{V}_2 - \boldsymbol{V}_1)^T \mathbf{C}(\,\boldsymbol{V}_2 - \boldsymbol{V}_1) \\ l_5^2 = (\,\boldsymbol{V}_3 - \boldsymbol{V}_2)^T \mathbf{C}(\,\boldsymbol{V}_3 - \boldsymbol{V}_2) \\ l_6^2 = (\,\boldsymbol{V}_1 - \boldsymbol{V}_3)^T \mathbf{C}(\,\boldsymbol{V}_1 - \boldsymbol{V}_3) \end{cases} \tag{A.2}$$

Remember that we denoted the matrix $\mathbf{V} = (\,\boldsymbol{V}_1,\, \boldsymbol{V}_2,\, \boldsymbol{V}_3)$, we define a matrix $\mathbf{D} = \mathbf{V}^T\mathbf{C}\mathbf{V}$:

$$\mathbf{D} = \mathbf{V}^T\mathbf{C}\mathbf{V} = \begin{pmatrix} \boldsymbol{V}_1^T \mathbf{C}\,\boldsymbol{V}_1 & \boldsymbol{V}_1^T \mathbf{C}\,\boldsymbol{V}_2 & \boldsymbol{V}_1^T \mathbf{C}\,\boldsymbol{V}_3 \\ \boldsymbol{V}_2^T \mathbf{C}\,\boldsymbol{V}_1 & \boldsymbol{V}_2^T \mathbf{C}\,\boldsymbol{V}_2 & \boldsymbol{V}_2^T \mathbf{C}\,\boldsymbol{V}_3 \\ \boldsymbol{V}_3^T \mathbf{C}\,\boldsymbol{V}_1 & \boldsymbol{V}_3^T \mathbf{C}\,\boldsymbol{V}_2 & \boldsymbol{V}_3^T \mathbf{C}\,\boldsymbol{V}_3 \end{pmatrix} \tag{A.3}$$

Combining with the Equation (A.2), we can express the matrix $\mathbf{D}$ by the six squared edge lengths:

$$\mathbf{D} = \begin{pmatrix} l_1^2 & \frac{1}{2}(l_1^2 + l_2^2 - l_4^2) & \frac{1}{2}(l_1^2 + l_3^2 - l_6^2) \\ \frac{1}{2}(l_1^2 + l_2^2 - l_4^2) & l_2^2 & \frac{1}{2}(l_2^2 + l_3^2 - l_5^2) \\ \frac{1}{2}(l_1^2 + l_3^2 - l_6^2) & \frac{1}{2}(l_2^2 + l_3^2 - l_5^2) & l_3^2 \end{pmatrix} \tag{A.4}$$

We can see that $\mathbf{D}$ is a linear function of the squared lengths $l_1^2, l_2^2, ..., l_6^2$, it can also be written as:

$$\mathbf{D} = l_1^2 \cdot \mathbf{D}_1 + l_2^2 \cdot \mathbf{D}_2 + l_3^2 \cdot \mathbf{D}_3 + l_4^2 \cdot \mathbf{D}_4 + l_5^2 \cdot \mathbf{D}_5 + l_6^2 \cdot \mathbf{D}_6 \tag{A.5}$$

with:

$$\mathbf{D}_1 = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 \\ \frac{1}{2} & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_2 = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{bmatrix}, \quad \mathbf{D}_3 = \begin{bmatrix} 0 & 0 & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$
$$\mathbf{D}_4 = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \mathbf{D}_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{2} \\ 0 & -\frac{1}{2} & 0 \end{bmatrix}, \quad \mathbf{D}_6 = \begin{bmatrix} 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 \end{bmatrix} \tag{A.6}$$

From Equation (A.3), $\mathbf{C}$ can be expressed by: $\mathbf{C} = \mathbf{V}^{-T}\mathbf{D}\mathbf{V}^{-1}$. As $\mathbf{V}$ is the matrix of the initial edge vectors, it can be precomputed in the initialization function, and $\mathbf{D}$ is a linear function of the squared lengths $l_1^2, l_2^2, ..., l_6^2$, so under the P1 approximation, the tensor $\mathbf{C}$ depends linearly on the squared edge lengths: $\mathbf{C} = l_1^2 \cdot \mathbf{C}_1 + l_2^2 \cdot \mathbf{C}_2 + l_3^2 \cdot \mathbf{C}_3 + l_4^2 \cdot \mathbf{C}_4 + l_5^2 \cdot \mathbf{C}_5 + l_6^2 \cdot \mathbf{C}_6$, in which $\mathbf{C}_i = \mathbf{V}^{-T}\mathbf{D}_i\mathbf{V}^{-1}$.

Appendix B. Derivatives of l w.r.t nodal positions

From the relation between the squared edge lengths (l_i^2) expressed by the edge vectors, we can express their derivatives with respect to their four defining

vertices. We take the first squared edge length l_1^2 for example, it can be written as: $l_1^2 = \boldsymbol{v}_1^T \boldsymbol{v}_1 = (x_1 - x_0)^T (x_1 - x_0)$, so its derivative with respect to the first node is: $\frac{\partial l_1^2}{\partial x_0} = -2(x_1 - x_0)^T = -2\boldsymbol{v}_1^T$. Finally, we can have the derivatives of $\mathbf{l}$ with respect to four vertices of a tetrahedron:

$$\begin{aligned} \tfrac{\partial \mathbf{l}}{\partial x_0} &= \mathbf{DLM}_0 \cdot \mathbf{v}, \quad \tfrac{\partial \mathbf{l}}{\partial x_1} = \mathbf{DLM}_1 \cdot \mathbf{v}, \\ \tfrac{\partial \mathbf{l}}{\partial x_2} &= \mathbf{DLM}_2 \cdot \mathbf{v}, \quad \tfrac{\partial \mathbf{l}}{\partial x_3} = \mathbf{DLM}_3 \cdot \mathbf{v} \end{aligned} \tag{B.1}$$

with

$$\begin{aligned} \mathbf{DLM}_0 &= \begin{pmatrix} -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{DLM}_1 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2 & -2 & 0 \\ 0 & 0 & 0 \\ 2 & 0 & -2 \end{pmatrix}, \\ \mathbf{DLM}_2 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \\ -2 & 2 & 0 \\ 0 & 2 & -2 \\ 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{DLM}_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & -2 & 2 \\ -2 & 0 & 2 \end{pmatrix} \end{aligned} \tag{B.2}$$

Appendix C. Derivatives of C invariants and of w

We denote **vtr** and **Mtr** the 6-vector of traces of $\mathbf{C}_i$ and the 6×6-matrix of traces of $\mathbf{C}_i\mathbf{C}_j$ $(1 \le i \le 6, 1 \le j \le 6)$ respectively. The three invariants can be expressed as:

$$\begin{aligned} I_1 &= \mathbf{vtr} \cdot \mathbf{l}, \\ I_2 &= \tfrac{1}{2}\mathbf{l}^T(\mathbf{vtr} \otimes \mathbf{vtr} - \mathbf{Mtr})\mathbf{l}, \\ I_3 &= (\tfrac{\mathrm{vol}}{\mathrm{Vol}})^2, \end{aligned} \tag{C.1}$$

in which the vol and Vol represent the deformed and initial volume.

For the isotropic materials, the energy density function w depends only on the three invariants, so the problem of derivative of w with respect to $\mathbf{l}$ turns to a problem of derivatives of the invariants with respect to $\mathbf{l}$. For the first invariant:

$$\frac{\partial I_1}{\partial \mathbf{l}} = \mathbf{vtr}^T \tag{C.2}$$

For the second invariant:

$$\frac{\partial I_2}{\partial \mathbf{l}} = \frac{1}{2}\mathbf{l}^T(\mathbf{vtr} \otimes \mathbf{vtr} - \mathbf{Mtr}). \tag{C.3}$$

The third invariant is a function of volume. To calculate the force per unit volume, we can derive the energy density w with respect to vol: $\frac{\partial w(I_3)}{\partial x} = \frac{\partial w(I_3)}{\partial vol}\frac{\partial vol}{\partial x}$. There is only $\frac{\partial vol}{\partial x}$ which still needs to be determined. According to the space analytic geometry, the volume of a tetrahedron can be calculated from the three vectors: $vol = \frac{1}{6}\det(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)$. Its derivative with respect to the four vertices of an element can be expressed like this:

$$\begin{aligned}
\tfrac{\partial \mathrm{vol}}{\partial x_1} &= \tfrac{1}{6}(\boldsymbol{v}_2 \wedge \boldsymbol{v}_3)^T \\
\tfrac{\partial \mathrm{vol}}{\partial x_2} &= \tfrac{1}{6}(\boldsymbol{v}_6 \wedge \boldsymbol{v}_1)^T \\
\tfrac{\partial \mathrm{vol}}{\partial x_3} &= \tfrac{1}{6}(\boldsymbol{v}_2 \wedge \boldsymbol{v}_4)^T \\
\tfrac{\partial \mathrm{vol}}{\partial x_0} &= [\tfrac{\partial \mathrm{vol}}{\partial x_{ix}}, \tfrac{\partial \mathrm{vol}}{\partial x_{iy}}, \tfrac{\partial \mathrm{vol}}{\partial x_{iz}}]^T \quad (i = 1, 2, 3)
\end{aligned} \tag{C.4}$$

in which the subscripts x, y, z represent the three directions.

As a result, the expression of energy density w and the forces for different constitutive laws for the HEML method can be obtained as shown in Table 3.

Acknowledgements

This work is partly supported by the SAGA project of French Agence Nationale de la Recherche.

The authors thank for their valuable comments on preliminary versions of the paper: Samuel Forest, Stéphane Avril, François Faure and Yannick Tillier.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] C. Basdogan, C-H. Ho, and MA. Srinivasan. Virtual environments for medical training: graphical and haptic simulation of laparoscopic common bile duct exploration. *IEEE/ASME Trans. Mechatronics*, 6:269–285, 2001.

[2] S.P. DiMaio and S.E. Salcudean. Interactive simulation of needle insertion models. *IEEE Trans. Biomed. Eng.*, 52(7):1167–1179, 2005.

[3] Y. Tillier, A. Paccini, M. Durand-Reville, and J.-L. Chenot. Finite element modelling for soft tissues surgery based on linear and nonlinear elasticity behaviour. *Comput. Aided Surg.*, 11(2):63–68, 2006.

[4] E. Peña, B. Calvo, M.A Martínez, P. Martins, T. Mascarenhas, R.M.N. Jorge, A. Ferreira, and M. Doblaré. Experimental study and constitutive modeling of the viscoelastic mechanical properties of the human prolapsed vaginal tissue. *Biomech. Model Mechanobiol.*, 9:35–44, 2010.

[5] C. Rubod, M. Brieu, M. Cosson, G. Rivaux, J.-C. Clay, L. de Landsheere, and B. Gabriel. Biomechanical properties of human pelvic organs. *J. Urology*, 79(4):17–22, 2012.

[6] R. Baumann and D. Glauser. Force feedback for virtual reality based minimally invasive surgery simulator. In *Medecine Meets Virtual Reality*, San Diego, USA, January 1996.

[7] P. Meseure and C. Chaillou. A deformable body model for surgical simulation. *J. Visual Comput. Animat.*, 11(4):197–208, 2000.

[8] N. Pirro, M.E. Bellemare, M. Rahim, O. Durieux, I. Sielezneff, B. Sastre, and P. Champsaur. Résultats préliminaires et perspectives de la modélisation dynamique pelvienne patient-spécifique. *Pelv. Perineol*, 4:15–21, 2009.

[9] S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *Vis. Comput.*, 16(8):437–452, 2000.

[10] G. Picinbono, H. Delingette, and N. Ayache. Non-linear anisotropic elasticity for real-time surgery simulation. *Graph. Model*, 65(5):305–321, 2003.

[11] H. Delingette and N. Ayache. Soft tissue modeling for surgery simulation. In *Computational Models for the Human Body*, pages 453–550. Elsevier, 2004.

[12] K. Miller, G. Joldes, D. Lance, and A. Wittek. Total langrangian explicit dynamics finite element algorithm for computing soft tissue deformation. *Commun. Numer. Meth. Engng*, 23(2):121–134, 2007.

[13] Y.-J. Lim and S. De. Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres. *Comput. Methods Appl. Mech. Engng.*, 196:3011–3024, 2007.

[14] S. Niroomandi, I. Alfaro, D. Gonzalez, E. Cueto, and F. Chinesta. Model order reduction in hyperelasticity: a proper generalized decomposition approach. *Int. J. Numer.Meth. Eng.*, 96(3):129–149, 2013.

[15] S. Banihani, T. Rabczuk, and T. Almomani. Pod for real-time simulation of hyperelastic soft biological tissue using the point collocation method of finite spheres. *Math. Probl. Eng.*, 2013, 2013.

[16] S. Marchesseau, T. Heimann, S. Chatelin, R. Willinger, and H. Delingette. Fast porous visco-hyperelastic soft tissue model for surgery simulation: Application to liver surgery. *Prog. Biophys. Mol. Biol.*, 103:185–196, 2010.

[17] J. Berkley, S. Weghorst, H. Gladstone, G. Raugi, D. Berg, and M. Ganter. Banded matrix approach to finite element modelling for soft tissue simulation. *Virtual Real.*, 4(3):203–212, 1999.

[18] H.W. Nienhuys and A.F. Van der Stappen. Combining finite element deformation with cutting for surgery simulations. In *Eurographics00*, Interlaken, 20-25 Aug 2000.

[19] C. Monserrat, V. Hermandez, M. Alcaniz, M.C. Juan, and V. Grau. A new approach for real time simulation of tissue deformations in surgery simulation. *Comput. Meth. Programs Biomed.*, 64(2):77–85, 2001.

[20] G. Debunne, M. Desbrun, M.P. Cani, and A.H. Barr. Dynamic real-time deformations using space & time adaptive sampling. In *Computer Graphics annual conference series*, Los Angeles, August 2001.

[21] M. Nesme, M. Marchal, E. Promayon, M. Chabanas, Y. Payan, and F. Faure. Physically realistic interactive simulation for biological soft tissues. *Recent Res. Devel. Biomech.*, 2:1–22, 2005.

[22] F. Goulette and S. Chendeb. A framework for fast computation of hyperelastic materials deformations in real-time simulation of surgery. In *Computational Biomechanics for Medicine (CBM) Workshop of the Medical Image Computing and Computer Assisted Intervention (MICCAI) Conference*, Copenhagen, Denmark, 1-6 Oct 2006.

[23] W.M. Lai, D. Rubin, and E. Krempl. *Continuum Mechanics.* Oxford: Butterworth, 1993.

[24] M. Sasso, G. Palmieri, G. Chiappini, and D. Amodio. Characterization of hyperelastic rubber-like materials by biaxial and uniaxial stretching tests based on optical methods. *Polym. Test*, 27:995–1004, 2008.

[25] F. Peyraut, Z.-Q. Feng, N. Labed, and C. Renaud. A closed form solution for the uniaxial tension test of biological soft tissues. *Int. J. Nonlinear Mech.*, 45:535–541, 2010.

[26] O.C. Zienkiewicz and R.L. Taylor. *The finite element method, Fifth edition, Volume 1: The basis.* Butterworth-Heinemann, 2000.

[27] Y.C. Fung. *Biomechanics: Mechanical Properties of Living Tissues.* Springer, 1993.

[28] R.V. Hingorani, P.P. Provenzano, R.S. Lakes, A. Escarcega, and Vanderby Jr. R. Nonlinear viscoelasticity in rabbit medial collateral ligament. *Ann. Biomed. Engng.*, 32:306–312, 2004.

[29] J.D. Humphrey. Mechanics of the arterial wall: review and directions. *Crit. Rev. Biomed. Eng.*, 23(1-2):1–162, 1995.

[30] J.S. Grashow, A.P. Yoganathan, and M. Sacks. Biaxial stress-stretch behavior of the mitral valve anterior leaflet at physiologic strain rates. *Ann. Biomed. Engng.*, 34:315–325, 2006.

[31] K. Miller and K. Chinzei. Mechanical properties of brain tissue in tension. *J. Biomech.*, 35(4):483–490, 2002.

[32] Z.A. Taylor, O. Comas, Cheng. M., Passenger. J., D.J. Hawkes, D. Atkinson, and S. Ourselin. On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and gpu execution. *Med. Image Anal.*, 13:234–244, 2009.

[33] R. W. Ogden. *Non-linear elastic deformations.* Dover, 1997.

[34] M. Mooney. A theory of large elastic deformation. *J. Appl. Phys.*, 11(9):582–592, 1940.

[35] R.S. Rivlin. Large elastic deformations of isotropic materials. iv. further developments of the general theory. *Phil. Trans. R. Soc. Lond. A*, 241(835):379–397, 1948.

[36] G. Venugopala Rao, C. Rubod, M. Brieu, N. Bhatnagar, and Cosson M. Experiments and finite element modelling for the study of prolapse in the pelvic floor system. *Comput. Methods Biomech. Biomed. Eng.*, 13:349–357, 2010.

[37] K.F. Noakes, A.J. Pullan, I.P. Bissett, and L.K. Cheng. Subject specific finite elasticity simulations of the pelvic floor. *J. Biomech.*, 41:3060–3065, 2008.

[38] S. Majumder, A. Roychowdhury, and S. Pal. Effects of trochanteric soft tissue thickness and hip impact velocity on hip fracture in sideways fall through 3d finite element simulations. *J. Biomech.*, 41:2834–2842, 2008.

[39] P.J. Flory. Thermodynamic relations for high elastic materials. *Trans. Faraday Soc.*, 57:829–838, 1961.

[40] R.W. Ogden. Volume changes associated with the deformation of rubber-like solids. *J. Mech. Phys. Solids*, 24:323–338, 1976.

[41] G.A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering.* Wiley, Chichester, 2000.

[42] J.E. Bischoff, E.M. Arruda, and K. Grosh. A new constitutive model for the compressibility of elastomers at finite deformations. *Rubber Chem. Tech.*, 74:541–559, 2001.

[43] C. O. Horgan and G. Saccomandi. Finite thermoelasticity with limiting chain extensibility. *J. Mech. Phys. Solids*, 51:1127–1146, 2003.

[44] C.O. Horgan and G. Saccomandi. Constitutive models for compressible nonlinearly elastic materials with limiting chain extensibility. *J. Elasticity*, 77:123–138, 2004.

[45] C. O. Horgan and G. Saccomandi. Phenomenological hyperelastic strain-stiffening constitutive models for rubber. *Rubber Chem. Technol.*, 79:152–169, 2006.

[46] J. Bonet and R.D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge University Press, 1997.

[47] G. Irving, J. Teran, and R. Fedkiw. Invertible finite elements for robust simulation of large deformation. In R. Boulic and D.K. Pai, editors, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pages 131–140, 2004.

[48] J.C. Simo and K.S. Pister. Remarks on rate constitutive equations for finite deformations. *Comput. Methods Appl. Mech. Engng.*, 46:201–215, 1984.

[49] F. Peyraut. Orientation preservation and newton-raphson convergence in the case of an hyperelastic sphere subjected to an hydrostatic pressure. *Comput. Methods Appl. Mech. Engng.*, 192:1107–1117, 2003.

[50] Feng Z.-Q. Peyraut, F. and Labed N. A material-independent algorithm for preserving of the orientation of the spatial basis attached to deforming medium. *Computational Mechanics*, 40:1053–1060, 2007.

[51] X. Provot. *Animation réaliste de vêtements.* PhD thesis, Synthim-INRIA, 1997.

[52] R. Buttin, F. Zara, B. Shariat, T. Redarce, and G. Grangé. Biomechanical simulation of the fetal descent without imposed theoretical trajectory. *Comput. Meth. Programs Biomed.*, 111:389–401, 2013.

[53] Z.-Q. Feng. http://lmee.univ-evry.fr/∼feng/FerSystem.html, 2008.

[54] R.W. Ogden. Large deformation isotropic elasticity: on the correlation of theory and experiment for compressible rubberlike solids. *Proc. R. Soc. Lond. A.*, 328:567–583, 1972.

[55] Ogden R.W. Holzapfel G.A., Gasser T.C. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity*, 61:1–48, 2000.

[56] Holzapfel G. A. Determination of material models for arterial walls from uniaxial extension tests and histological structure. *J. Theor. Biol.*, 238(2):290–302, 2006.

[57] Fronek K. Fung, Y. C. and Patitucci P. Pseudoelasticity of arteries and the choice of its mathematical expression. *Am J Physiol Heart Circ Physiol*, 237(5):H620–631, 1979.

[58] X.Q. Peng, Z.Y. Guo, and B. Moran. An anisotropic hyperelastic constitutive model with fiber-matrix shear interaction for the human annulus fibrosus. *J. Appl. Mech.*, 73(5):815–824, 2006.