Multiplicative Jacobian Energy Decomposition Method for Fast Porous Visco-Hyperelastic Soft Tissue Model

Stéphanie Marchesseau[1], Tobias Heimann[1], Simon Chatelin[2], Rémy Willinger[2], and Hervé Delingette[1]

[1] Asclepios Research Project, INRIA Sophia Antipolis, France
[2] University of Strasbourg, IMFS-CNRS, Strasbourg, France

Abstract. Simulating soft tissues in real time is a significant challenge since a compromise between biomechanical accuracy and computational efficiency must be found. In this paper, we propose a new discretization method, the *Multiplicative Jacobian Energy Decomposition* (MJED) which is an alternative to the classical Galerkin FEM (Finite Element Method) formulation. This method for discretizing non-linear hyperelastic materials on linear tetrahedral meshes leads to faster stiffness matrix assembly for a large variety of isotropic and anisotropic materials. We show that our new approach, implemented within an implicit time integration scheme, can lead to fast and realistic liver deformations including hyperelasticity, porosity and viscosity.

1 Introduction

The simulation of soft tissue deformation has attracted a growing interest in the past 15 years both in the biomechanics and the medical image analysis communities. Modeling in silico the deformation of soft tissues is of high interest in particular for surgical gesture training[1] and therapy planning[2]. In this paper, we focus on the simulation of liver deformation in the context of surgery training. In such case, it is crucial that soft tissue deformation is simulated in real-time, i.e. at a minimum of 25 frames per second for visual feedback. Furthermore, in surgery simulation, there are additional constraints of numerical stability during the occurrence of contact between soft tissue and (virtual) surgical instruments.

To simulate soft tissues efficiently and realistically, some authors have relied on the *Total Lagrangian Explicit Dynamic* (TLED) algorithm[3, 4] to simulate deformations with explicit time integration schemes. However, the main limitation of these explicit schemes is that they require very small time steps to keep the computation stable, especially for stiff materials. Indeed, it is necessary to iterate multiple times to propagate applied forces from a node to the whole mesh. Therefore, with such approaches, it is difficult to produce realistic simulations of contact with rigid objects, such as surgical tools. Implicit integration schemes require the evaluation of a global stiffness matrix and the solution of linear system of equations at each time step.

In this paper, we first introduce the *Multiplicative Jacobian Energy Decomposition* (MJED): a general algorithm to implement hyperelastic materials based on total Lagrangian FEM with implicit time integration schemes. Then we propose a realistic biomechanical model of the liver which combines hyperelasticity, viscoelasticity as well as poroelasticity. The viscoelasticity of our liver model is based on Prony series, the parameters of which have been experimentally estimated through a dynamic strain sweep testing. Finally, we take into account the porous medium of the liver parenchyma through a poro-elastic model which computes the fluid pressure and the resulting applied pressure on the solid phase.

2 Multiplicative Jacobian Energy Decomposition

Under large deformation, linear elasticity is no longer valid and the liver behavior is better represented as an hyperelastic material. Since we are using implicit time integration schemes, it is necessary at each time step to compute hyperelastic forces and stiffness matrices with a discretization method. The Finite Element Method is a widely used approach to this end, however the constraint of real-time simulation is not always satisfied. The objective of this section is to introduce a fast discretization method suitable for all hyperelastic materials.

To discretize the liver geometry, we use tetrahedral linear finite elements. T_P is the rest tetrahedron (with vertices P_i) which is transformed under the deformation function $\phi(X)$ into the tetrahedron T_Q (with vertices Q_i). Any hyperelastic material is fully determined by its strain energy function W_{h} which describes the amount of energy necessary to deform the material. This strain energy function is defined in a way which is invariant to the application of rigid transformations: it involves the invariants of the *Cauchy-deformation tensor* defined as $\mathbf{C} = \nabla\phi^T\nabla\phi$. There are numerous invariants of $\mathbf{C}$ (see [5] for detailed explanation) but the one commonly used are the following: $I_1 = \mathrm{tr}\mathbf{C}$, $I_2 = \frac{1}{2}((\mathrm{tr}\mathbf{C})^2 - \mathrm{tr}\mathbf{C}^2)$, $I_4 = a^T\mathbf{C}a$ (where a is the main fiber direction), and the Jacobian $J = \det\nabla\phi$.

2.1 Decomposition of Strain Energy

We decouple in the strain energy, the invariants of $\mathbf{C}$ from J so as to avoid complex derivative expressions and matrix inversion of $\mathbf{C}$. Instead of computing the force and stiffness matrix using the classical Galerkin FEM[6], we compute them directly using the Rayleigh-Ritz method by deriving the energy with respect to the nodal position:

$$F_i = -\left(\frac{\partial W_{\mathrm{h}}}{\partial Q_i}\right)^T \text{ and } \mathbf{K}_{ij} = \left(\frac{\partial^2 W_{\mathrm{h}}}{\partial Q_j \partial Q_i}\right) \tag{1}$$

It is important to note that the approach developed in this section is completely equivalent to the classical FEM one but leads to more efficient computation. A comparison with the open source software FEBio proved this equivalence. We

propose to write the strain energy functions as a sum of terms $W_h^k = f^k(J)g^k(\tilde{I})$ or exponentials of W_h^k, where $\tilde{I} = (I_1, I_2, I_4...)$. This decomposition applies to every material models we studied so far (Costa , Veronda Westmann, Boyce Arruda, StVenant Kirchhoff, NeoHookean, Ogden, Mooney Rivlin). This gives the following expression of the tetrahedron strain energy:

$$W_h = V_0 \sum_{k=1}^{n} f^k(J)g^k(\tilde{I}) + V_0 \, \exp \left(\sum_{k=n+1}^{n'} f^k(J)g^k(\tilde{I}) \right) \tag{2}$$

Using this exact decomposition of strain energy enables complex material formulation to be computed more efficiently with only a sum of reasonably simple terms. With this decomposition, getting $f^{k'}(J)$ requires a 1D derivation, and getting $\mathbf{S}_h^k = 2\frac{\partial g^k(\tilde{I})}{\partial \mathbf{C}}$ requires to combine well-known derivatives of the invariants (such as $\frac{\partial I_1}{\partial \mathbf{C}} = \mathbf{Id}$ or $\frac{\partial I_2}{\partial \mathbf{C}} = \mathbf{Id}I_1 - \mathbf{C}$ where $\mathbf{Id}$ is the 3×3 identity matrix). For instance, the nodal force expression becomes:

$$F_{h,i} = -V_0 \sum_{k=1}^{n} \left(f^{k'}(J)g^k(\tilde{I}) \left(\frac{\partial J}{\partial Q_i} \right)^T + f^k(J)\nabla\phi \, \mathbf{S}_h^k \, D_i \right) \tag{3}$$

where D_i are gradients of shape functions, called *shape vectors*.

2.2 Formulation of the Stiffness Matrix

Implicit time integration schemes require the computation of the tangent stiffness matrix at each time step. This naturally involves elasticity tensors computed as the derivative of $\mathbf{S}_h^k$ with respect to $\mathbf{C}$ for each tetrahedron and at each time step. The MJED leads to far simpler expressions of those tensors because $\mathbf{S}_h^k$ is independent of J. Furthermore, in many common material models, the term containing those elasticity tensors can be precomputed. The full expression of the stiffness matrix includes 6 terms. Due to space constraints, we only focus below on the term involving the fourth order elasticity tensor: $\mathbf{R}_h^k = f^k(J) \left(\frac{\partial \mathbf{S}_h^k}{\partial Q_j} D_i \right)^T \nabla\phi^T$

which requires the computation of the tensor $\frac{\partial \mathbf{S}_h^k}{\partial \mathbf{C}} : \mathbf{H}$ where $\mathbf{H}$ is a symmetric matrix. In all cases, this tensor can be written as a sum of two kinds of terms, $\beta_1^k \mathbf{A}_1^k \mathbf{H} \mathbf{A}_1^k$ or $\beta_2^k(\mathbf{H} : \mathbf{A}_2^k)\mathbf{A}_2^k$ where β_u^k are scalars, $\mathbf{A}_u^k$ are symmetric matrices, and $\mathbf{A} : \mathbf{B} = \text{tr}(\mathbf{B}^T\mathbf{A})$ for any two matrices $\mathbf{A}, \mathbf{B}$. Therefore, the term $\mathbf{R}_h^k$ is a combination of two terms:

$$f^k(J)\nabla\phi\mathbf{L}_1^k(i,j)\nabla\phi^T \text{ and } f^k(J)\nabla\phi\mathbf{L}_2^k(i,j)\nabla\phi^T \tag{4}$$

where $\mathbf{L}_1^k(i,j)$ and $\mathbf{L}_2^k(i,j)$ are linear matrices depending on the shape vectors D_i, D_j, the matrices $\mathbf{A}_u^k$ and the scalars β_u^k. This formulation leads to an optimization for the assembly of the stiffness matrix for two reasons. First, no fourth order tensors are required, only scalars and symmetric matrices are involved in the computation. Second, except for the Ogden model, the matrices $\mathbf{A}_u^k$ are constant and therefore matrices $\mathbf{L}_1^k(i,j)$ and $\mathbf{L}_2^k(i,j)$ can be precomputed for each tetrahedron before the simulation.

2.3 Coping with highly compressed elements

In case of high compression, the volumetric terms $f^k(J)$ in the strain energy become dominant. This makes the stiffness matrix singular and thus leads to numerically unstable computations. We propose to *regularize* a second term of the stiffness matrix $\mathbf{G}_h^k = f^{k''}(J)\, g^k(\tilde{I})\, \frac{\partial J}{\partial Q_j} \otimes \frac{\partial J}{\partial Q_i}$ by replacing it with the following expression : $\mathbf{G}_h^k = f^{k''}(J)g^k(\tilde{I})\left((1-b)\frac{\partial J}{\partial Q_j} \otimes \frac{\partial J}{\partial Q_i} + b\frac{\partial J}{\partial Q_j} \cdot \frac{\partial J}{\partial Q_i}\mathbf{Id}\right)$. The closer b is to 1, the more $\mathbf{G}_h^k$ resembles to a diagonal matrix. In practice, we set $b = (1 - J)$ if $0 \leq J \leq 1$, $b = 0$ if $J \geq 1$ and $b = 1$ if $J \leq 0$. With this technique, it is even possible to handle inverted elements when the strain energy remains finite at $J = 0$.

3 Modeling Visco - Poro - Hyperelasticity

3.1 Visco-Hyperelasticity based on Prony Series

To accurately model the viscoelasticity of the liver,we propose to rely on Prony series [4]. In this method, time-dependent stresses are added to the hyperelastic stress tensor $\mathbf{S}_v$. Time-dependence is given by $\alpha(t) = \alpha_\infty + \sum_i \alpha_i \exp(-t/\tau_i)$ with the condition $(\alpha_\infty + \sum_i \alpha_i) = 1$. The visco-hyperelastic SPK tensor $\mathbf{S}_v$ can be written as:

$$\mathbf{S}_v = \mathbf{S}_h - \sum_i \gamma_i \quad \text{where} \quad \gamma_i = \int_0^t \alpha_i \left(1 - \exp\left(\frac{t' - t}{\tau_i}\right)\right) \frac{\partial \mathbf{S}_h}{\partial t'} dt' \quad (5)$$

After a discretization over time this results in the recursive formula: $\gamma_i^n = a_i \mathbf{S}_h^n + b_i \gamma_i^{n-1}$ where $a_i = \Delta t \alpha_i / (\Delta t + \tau_i)$ and $b_i = \tau_i / (\Delta t + \tau_i)$. Δt is the time step used for discretization and has to be the same as the time step for the solvers of the simulation.

3.2 Poro-Elasticity

Following Kerdok's model [7], we propose to model the liver as a fluid-filled sponge as it filters the blood through its parenchyma. The proportion of free-fluid (blood, water...) in the liver parenchyma in the reference configuration is set to a constant f_w, $1 - f_w$ represents the initial ratio of the solid phase (e.g. hepatic cells...). We introduce the effective volumetric Jacobian $J^* = (f_w + J - 1)/f_w$, and define the volumetric Cauchy stress following Hencky's elasticity : $\sigma_{\text{Heq}} = K_0\, f_w\, \ln(J^*)$ where K_0 is the bulk modulus of the material. With this model, when J gets close to $1 - f_w$, the solid phase of the liver is completely compressed and the resulting stress is infinite. To avoid instabilities due to this infinite stress, we substitute σ_{Heq} when $J \leq J_0$ by its tangent curve at $J_0 = 1 - f_w + K_0/K_{\text{lim}}$ where K_{lim} is a bulk modulus and represents the slope of the tangent (see Fig.1). The fluid phase of the liver also applies some volumetric stresses due to the transient response of the fluid through the porous liver parenchyma.

A straightforward way of modeling the porous behavior is through the linear Darcy's law. In this setting, variation of fluid pressure P_{fluid} is governed by the variation of volume change and a diffusive process:

$$\frac{1}{K_{\text{lim}}}\dot{P}_{\text{fluid}} = \kappa\nabla^2 P_{\text{fluid}} - \frac{\dot{J}}{J} \tag{6}$$

where κ is the permeability parameter, kept constant to decrease the computational cost. Finally, the total Cauchy stress response in the volumetric part is defined by summing the solid and the fluid terms: $\sigma_{\text{p}} = \sigma_{\text{heq}}\mathbf{Id} - P_{\text{fluid}}\mathbf{Id}$. The simulated fluid pressure field during a deformation under gravity force is shown in Fig.1. Highest pressure in the fluid occurs when the liver is compressed either by the gravity pressure (diffusion starts at the top) or by elastic reaction (diffusion starts at the bottom).
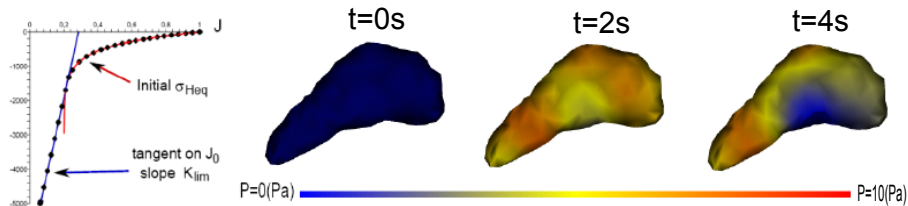

Fig. 1. (Left) Representation of the static Cauchy stress before and after substitution. The dot curve represents the new stress. Here $f_{\text{w}} = 0.8$. (Right) Pressure field of the porous component on a liver under gravity. Using $f_{\text{w}} = 0.5$, $K_0 = 400$ Pa, $K_{\text{lim}} = 2200$Pa and $\kappa = 20$ m^4/Ns.

4   Results and Validation

4.1   Computation time of the hyperelastic implementation

Decreasing computation time of the hyperelastic term is essential to reach real-time simulation as this term represents around 60% of the total time needed in one step (see Fig.2). In order to validate the MJED method, we compared our implementation with the classical FEM method explained in [6], referred to as "Standard FEM", implemented in SOFA[3]. We measured the time elapsed for the computation of the nodal forces and the stiffness matrices averaged over 100 iterations. We simulated the deformation of a cube with 20 700 tetrahedra and 4300 nodes. The results are given in Fig.2. For all implemented models the proposed strategy is more efficient than the standard FEM, up to five times as fast for St Venant Kirchhoff material.

---

[3] SOFA is an Open Source medical simulation software available at www.sofa-framework.org
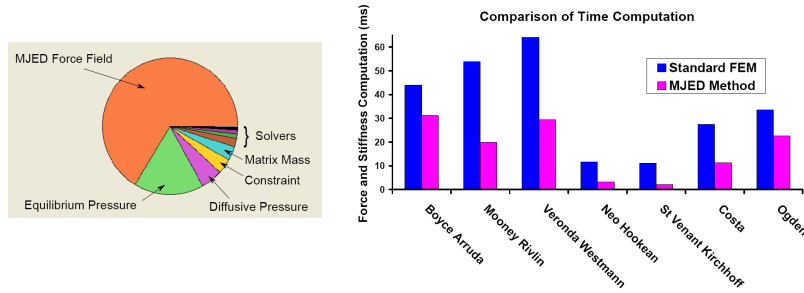
Fig. 2. (Left) Break-down of computational time between the components during one time step. (Right) Comparison of the computation times of nodal forces and stiffness matrices between two different discretization methods averaged over 100 iterations.

4.2   Visco-elasticity validation

To calibrate the visco-elastic parameters of our liver model, tests on at least 60 samples from 5 animals were performed on porcine livers. Dynamic viscoelastic behavior of hepatic tissue was investigated using *in vitro* Dynamic Mechanical Analysis (DMA) without perfusion in order to capture only the viscoelasticity. Dynamic Frequency Sweep tests were performed on a dedicated stress-controlled AR2000 rheometer (TA-Instruments, New Castle, DE, USA) in the linear viscoelastic strain range of the samples (see Fig.3). From the results, the Dynamic Modulus G can be obtained as a function of the frequency or function of the time, and the viscoelastic behavior can be modeled after fitting a generalized Maxwell model with two modes of relaxation to those measurements: $G(t) = G_0(g_\infty + g_1 \exp(-t/\tau_1) + g_2 \exp(-t/\tau_2)$) where $G_0\ g_\infty = G_\infty$ is called the equilibrium modulus, g_1, g_2, τ_1, τ_2 are parameters such as $g_\infty + g_1 + g_2 = 1$.
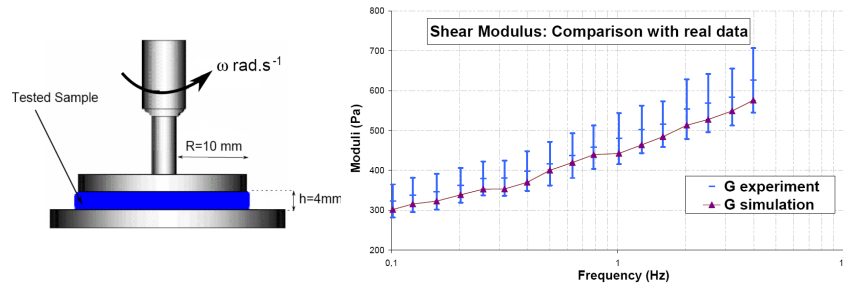

Fig. 3. (Left) Rheometer: Lower plate is fixed whereas upper is sinusoidally rotating. (Right) Comparison of the simulated values with the data obtained by DMA testing. The moduli are given on a X-log scale. The material is St Venant Kirchhoff with $G_0 = 770$Pa, $(\alpha_1, \alpha_2) = (0.235, 0.333)$ and $(\tau_1, \tau_2) = (0.27\text{s}, 0.03\text{s})$.

From the rheological experiments we derive the shear modulus G_0 required in the hyperelastic term and the Prony series parameters for the viscous term. To check the validity of those parameters, we compared *in silico* simulations with the performed *in vitro* rheological tests. Dynamic frequency sweep tests are simulated using similar geometries and boundary conditions as in the real DMA tests. We estimated the values of the Dynamic Modulus and compared them with experimental data. Figure 3 shows the results: the simulation manages to capture the viscous behavior of the liver for small deformations with a mean relative error of 5%. Given the fitting errors and the standard deviation of the values obtained with the DMA tests, this mean error is reasonably good.

4.3   Complete Liver Model

To analyze the influence of each component in the complete model, several simulations were performed using the same liver mesh (1240 vertices and 5000 tetrahedra) with Boyce Arruda hyperelastic material[7]. The liver mesh was segmented from a CT scan image and meshed with tetrahedra by the GHS3D software[4]. An Euler implicit time integration scheme was used, the linear equation was solved with a conjugated gradient algorithm. As boundary conditions, several nodes of the liver were fixed along the vena cava and suspensive ligament. The liver then deformed under the action of gravity.All computations were performed on a laptop PC with a Intel Core Duo processor at 2.80 GHz (simulations are available in the video clip). Adding viscosity to hyperelasticity increases the amplitude of the oscillations as the material becomes less stiff (see Fig.4). The frame rate is around 7 FPS against 7.5 FPS for hyperelasticity alone. We did not reach the 25 FPS needed for real-time interaction. However, the implicit integration scheme allows larger time step (0.3s for instance) which speeds up the simulation and makes user interactions efficient. High amount of extension and compression are possible which may be somewhat unrealistic, therefore the porous component is necessary to control the amount of viscosity. The maximum amplitude with
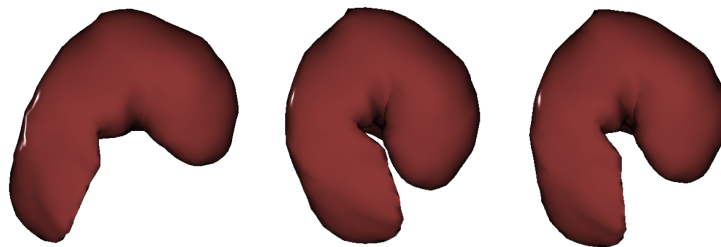

Fig. 4. Comparison of the maximum amplitudes under gravity. (Left) Hyperelastic Liver, (Middle) Visco-Hyperelastic liver, (Right) Visco-Poro-Hyperelastic.

---

[4] GHS3D is a mesh generator for tetrahedral elements, developed at INRIA, France

porosity is in between the hyperelasticity alone and the visco-hyperelasticity. The addition of this component decreases the computational efficiency (6 FPS) since a semi-implicit integration scheme is used for the porous component. Because of the fast variation of the explicit term $\dot{J}/J$, the time step has to be decreased to 0.15s. On our laptop PC, the simulation is still fluid enough to allow user interactions.

5   Conclusion

In this paper, we have introduced an efficient method to assemble stiffness matrices for complex biomechanical material models which compares favorably with the standard FEM method. We have also proposed a poro-visco-hyperelastic liver model suitable for real-time interaction which is, up to our knowledge, among the most realistic ones. Several model parameters have been identified from rheometric tests performed on porcine livers and a validation study has been successfully performed to reproduce those tests. Despite those advances, further research is needed to achieve realistic liver surgery simulations including the realistic contact with neighboring structures, the influence of breathing and cardiac motion, the simulation of hepatic resection, bleeding and suturing.

References

1. Delingette, H., Ayache, N.: Soft tissue modeling for surgery simulation. In: Computational Models for the Human Body. Elsevier (2004) 453–550
2. Hawkes, D., Barratt, D., Blackall, J., Chan, C., Edwards, P., Rhode, K., Penney, G., McClelland, J., Hill, D.: Tissue deformation and shape models in image-guided interventions: a discussion paper. Medical Image Analysis **9**(2) (2005) 163 – 175
3. Miller, K., Joldes, G., Lance, D., Wittek, A.: Total lagrangian explicit dynamics finite element algorithm for computing soft tissue deformation. Communications in Numerical Methods in Engineering **23** (2006) 121–134
4. Taylor, Z., Comas, O., Cheng, M., passenger, J., hawkes, D., Atkinson, D.., Ourselin, S.: On modelling of anisotropic viscoelasticity for soft tissue simulation: numerical solution and gpu execution. medical image Analysis **13** (2009) 234–244
5. Weiss, J., Maker, B., Govindjee, S.: Finite element implementation of incompressible, transversely isotropic hyperelasticity. Computer methods in applied mechanics and engineering **135** (1996) 107–128
6. Zienkiewicz, C., Taylor, R.: The Finite Element Method, Volume 2 : Solid Mechanics. Butterworth-Heinemann (2000)
7. Kerdok, A.E.: Characterizing the Nonlinear Mechanical Response of Liver to Surgical Manipulation. PhD thesis, Harvard University (2006)