



HAL
open science

Accounting for Large Deformations in real-Time Simulations of Soft Tissues Based on Reduced Order Models

Siamak Niroomandi, Iciar Alfaro, Elías Cueto, Francisco Chinesta

► **To cite this version:**

Siamak Niroomandi, Iciar Alfaro, Elías Cueto, Francisco Chinesta. Accounting for Large Deformations in real-Time Simulations of Soft Tissues Based on Reduced Order Models. *Computer Methods and Programs in Biomedicine*, 2011, 105 (1), pp.1-12. 10.1016/j.cmpb.2010.06.012 . hal-01004972

HAL Id: hal-01004972

<https://hal.science/hal-01004972>

Submitted on 24 Apr 2017

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.

Public Domain

Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models

S. Niroomandi^a, I. Alfaro^a, E. Cueto^a, F. Chinesta^b

^a*Group of Structural Mechanics and Materials Modeling. Aragón Institute of Engineering Research. University of Zaragoza. María de Luna, 5. E-50018. Zaragoza, Spain*

^b*GEM-UMR CNRS. École Centrale de Nantes. 1, rue de la Noë. F-44300 Nantes, France*

Abstract

Model reduction techniques have shown to constitute a valuable tool for real-time simulation in surgical environments and other fields. However, some limitations, imposed by real-time constraints, have not yet been overcome. One of such limitations is the severe limitation in time (established in 500 Hz of frequency for the resolution) that precludes the employ of Newton-like schemes for solving non-linear models as the ones usually employed for modeling biological tissues. In this work we present a technique able to deal with geometrically non-linear models, based on the employ of model reduction techniques, together with an efficient non-linear solver. Examples of the performance of the technique over some examples will be given.

Key words: Real-time simulation, model reduction, large deformations, haptic devices, Kirchhoff-Saint Venant.

1. Introduction

Minimally invasive surgery (MIS) gets more and more common nowadays. A minimally invasive procedure typically involves use of laparoscopic devices and remote-control manipulation of instruments with indirect observation of the surgical field through an endoscope or similar device, and are carried out through the skin or through a body cavity or anatomical opening. This may result in less pain, less strain of the organism, small injuries (aesthetic reasons), economic gain because of shorter hospital stays, etc. On the other

hand, there exist some important difficulties for the surgeon as a result of his restricted vision of the organs, difficult handling of the instruments, very restricted mobility, difficult hand-eye coordination and no tactile perception. Therefore, an important training phase is required before a surgeon acquires the skills necessary to adequately perform minimally invasive surgery. Currently, surgeons are trained to perform minimally invasive surgery by using mechanical simulators —just like plane pilots have been trained for many years— or living animals. There is some consensus on the limited realism of the mentioned simulators due to the complexity of the simulations to be carried [1]. The latter training method consists in practicing simple or complex surgical procedures on living animals (often pigs for abdominal surgery).

Because of the limitations of current training methods, there is a large interest in developing surgery simulation software, possibly with haptic feedback, for providing efficient and quantitative gesture training systems [2]. A simulator consists of several parts: input devices, the core of the simulator and output devices [1]. The input devices consist of a force-feedback (haptic) device with which the user moves the virtual surgery tools, a mouse and a keyboard to have a good view of the operation zone. The core of the simulator does the geometrical and the physical modeling. Geometrical models are obtained using medical imagery like CT-scan and MRI after they have been converted to standard 3D graphics formats. Physical models depend on the specific soft tissue. Many existing, commercial, surgery simulators employ linear elastic models or even mass-spring models. The core of the simulator also detects collision between virtual tools and the soft tissue and then calculates the corresponding reaction forces on the tools. The computed data is sent to the output devices which consist of a screen and the haptic device. The haptic device transforms reaction forces and moments to the user and makes him/her have the perception of virtual contact with the tissue. There are some requirements for visual and haptic feedback that make simulations difficult to perform in real time. A surgery simulator must provide a realistic visual presentation of the surgical procedure. Visual feedback is specially important in video-surgery because it helps the surgeon to acquire a tridimensional perception of the environment. If the positions, orientations and deformations of objects on the screen are updated at a rate less than about 30 times per second, users will no longer perceive the simulation as continuous motion. Haptic feedback provides the sensation of the movement to the user and therefore it significantly enhances his surgical performance. But it should have a frequency between 300 and 1000 Hz that is very difficult to

achieve for complex tissue behavior.

In order to comply with these requirements different methods have been used during last years often with trade-off between time and accuracy. Among them are mass-spring methods, surface models and finite element or boundary element methods. Spring-mass models consist of points with mass that are linked by springs and dashpots. They have been used extensively for simulating soft tissues [3, 4, 5]. The main advantage of spring models is their ease of implementation and they have been used for statics and dynamics analysis. Another advantage is their ability to model cutting and suturing simply by removing or adding connections between vertices. However, there are some disadvantages with these models. The most important is that, in general, they do not reproduce even the laws of linear Elasticity [6].

The models of the last category are based on the laws of continuum mechanics. Among the available numerical methods for solving these models, finite element and boundary element methods can be cited [7]. A good overview of different soft tissue models based on continuum mechanical is given in [8]. Very recently, a system based upon the use of neural networks has been presented [9]. In it, the system is trained with a large set of possible load states in order to achieve real-time performance in the execution loop.

Very recently, General Purpose Graphic Processing Units (GPGPUs) have been incorporated into the field of surgical simulation. This graphic hardware is able to perform very fast operations, and a complete nonlinear, explicit, finite element code has been implemented in that work with a gain on speed on the order of 20 times [10]. This approach has allowed to implement medium-sized models (16000 tetrahedra) including neo-hookean behavior in an explicit total Lagrangian approach. The main drawback of this elegant approach to the problem is the conditional stability of explicit integration procedures and also the explicit evaluation of the constitutive laws.

Other methods like mesh free methods have been recently used for real-time simulation of soft tissues. Lim and De applied the point collocation-based method of finite spheres (PCMFS) technique to simulate tissue deformations that are geometrically nonlinear [11]. The technique is based on a novel combination of a multi-resolution approach coupled with a fast analysis scheme in the nonlinear model.

Capel et al. [12] computed the equations of motion for a linearly elastic material using FEM and then by using coarse volumetric meshes to do simulations at interactive rates. Krysl et al. applied model order reduc-

tion approach for dimensional reduction of nonlinear finite element models of solid dynamics [13]. Although they proposed a general approach the computations cannot be performed in real time. Barbic and James, however, exploited dimensional model reduction techniques to build reduced-coordinate deformable models for material objects with complex geometry using St. Venant-Kirchhoff constitutive laws. In this way they have performed simulations at real-time update rates of geometrically-nonlinear materials [14]. Unfortunately the method seems not to be generalized to more complex material behaviors like neo-Hookean constitutive laws, etc.

In a previous paper the authors introduced a new technique for the real-time simulation of non-linear tissue behavior based on a model reduction technique known as Proper Orthogonal (POD) or Karhunen-Loève decompositions. The technique is based upon the construction of a complete model (using Finite Element modeling or other numerical techniques, but data could also be extracted from experimental results, if available) and the extraction and storage of the most relevant information in order to construct a model with very few degrees of freedom, but that takes into account the highly non-linear response of most living tissues. It was applied to the simulation of palpation a human cornea composed of a nonlinear, hyperelastic matrix reinforced with collagen fibers [15]. But the reduced model is actually linear although the best set of basis vectors, in statistical sense, is employed, if no updating of the tangent stiffness matrix is performed, as in a standard Newton-Raphson procedure. So the results had some error, more important the higher the strain is, in comparison with a standard FEM solution.

In this paper we apply an advanced non-linear solver, the asymptotic numerical method (ANM) to simulate the nonlinear behavior of soft tissues in a neighborhood of a known equilibrium state. In this method complex equilibrium paths are sought in the form of asymptotic expansions, and they are determined by solving several linear problems with a single tangent stiffness matrix [16]. Then we apply model reduction techniques (in this work we have used again POD) to the obtained system of equations. These reduced models are obtained for some “most probable” load states and then are properly interpolated for any other load state. In this way the system can be solved in very short times. The organization of the paper is as following: the fundamentals of model reduction techniques are explained in section 2. In section 3 first the asymptotic numerical method is introduced and then the formulation for geometrically-nonlinear problems is obtained. In section 4 it is shown how ANM formulation in reduced coordinates of POD can be

obtained and in section 5 several examples are given to show the performance of the method.

2. Model reduction techniques

The essential idea of model reduction techniques in the context of Continuum Mechanics is to extract the most relevant information of a given simulation, that is performed off-line and stored for this particular application, in order to construct a good-quality Ritz-like basis to perform reduced-model (on-line) simulations with very few degrees of freedom [17][15]. These basis functions are global and, in statistical terms, of very good quality. This is in sharp contrast with the finite element method, for instance, that employs general-purpose, locally supported, piecewise polynomial shape functions.

2.1. Fundamentals: Karhunen-Loève or Proper Orthogonal decomposition

The technique we employed is known by a wide variety of names, since it has been employed and re-discovered in many branches of science and engineering. Maybe the most common names are Karhunen-Loève decomposition [18] [19], proper orthogonal decomposition (POD) [20] or *empirical orthogonal functions* [21].

We assume that the evolution of a certain field $u(\mathbf{x}, t)$, governed by a PDE, is known. In practical applications, this field is expressed in a discrete form, that is, it is known typically at the nodes of a spatial mesh and for some time steps of existing simulations $u(\mathbf{x}_i, t^n) \equiv u_i^n$. The same can be wrote introducing a time discretization $u^n(\mathbf{x}) \equiv u(\mathbf{x}, t = n\Delta t)$; $\forall n \in [1, \dots, P]$. The main idea of the Karhunen-Loève (K-L) decomposition is how to obtain the most typical or characteristic structure $\phi(\mathbf{x})$ among these $u^n(\mathbf{x}) \forall n$. This is equivalent to obtaining a function $\phi(\mathbf{x})$ that maximizes the functional λ defined by

$$\lambda = \frac{\sum_{n=1}^{n=P} \left[\sum_{i=1}^{i=N} \phi(\mathbf{x}_i) u^n(\mathbf{x}_i) \right]^2}{\sum_{i=1}^{i=N} (\phi(\mathbf{x}_i))^2} \quad (1)$$

The maximization ($\delta\lambda = 0$) leads to:

$$\sum_{n=1}^{n=P} \left[\left(\sum_{i=1}^{i=N} \tilde{\phi}(\mathbf{x}_i) u^n(\mathbf{x}_i) \right) \left(\sum_{j=1}^{j=N} \phi(\mathbf{x}_j) u^n(\mathbf{x}_j) \right) \right] = \lambda \sum_{i=1}^{i=N} \tilde{\phi}(\mathbf{x}_i) \phi(\mathbf{x}_i); \quad \forall \tilde{\phi} \quad (2)$$

which can be rewritten in the form

$$\sum_{i=1}^{i=N} \left\{ \sum_{j=1}^{j=N} \left[\sum_{n=1}^{n=P} u^n(\mathbf{x}_i) u^n(\mathbf{x}_j) \phi(\mathbf{x}_j) \right] \tilde{\phi}(\mathbf{x}_i) \right\} = \lambda \sum_{i=1}^{i=N} \tilde{\phi}(\mathbf{x}_i) \phi(\mathbf{x}_i); \quad \forall \tilde{\phi} \quad (3)$$

Defining the vector $\boldsymbol{\phi}$ such that its i -th component is $\phi(\mathbf{x}_i)$, Eq. (3) takes the following matrix form

$$\tilde{\boldsymbol{\phi}}^T \mathbf{c} \boldsymbol{\phi} = \lambda \tilde{\boldsymbol{\phi}}^T \boldsymbol{\phi}; \quad \forall \tilde{\boldsymbol{\phi}} \Rightarrow \mathbf{c} \boldsymbol{\phi} = \lambda \boldsymbol{\phi} \quad (4)$$

where the two-point correlation matrix is given by

$$c_{ij} = \sum_{n=1}^{n=P} u^n(\mathbf{x}_i) u^n(\mathbf{x}_j) \Leftrightarrow \mathbf{c} = \sum_{n=1}^{n=P} \mathbf{u}^n (\mathbf{u}^n)^T \quad (5)$$

which is symmetric and positive definite. If we define the matrix \mathbf{Q} containing the discrete field history:

$$\mathbf{Q} = \begin{pmatrix} u_1^1 & u_1^2 & \dots & u_1^P \\ u_2^1 & u_2^2 & \dots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_N^1 & u_N^2 & \dots & u_N^P \end{pmatrix} \quad (6)$$

it is straightforward to verify that the matrix \mathbf{c} in Eq. (4) results

$$\mathbf{c} = \mathbf{Q} \mathbf{Q}^T \quad (7)$$

where the diagonal components are given by

$$c_{ii} = (\mathbf{Q} \mathbf{Q}^T)_{ii} = \sum_{j=1}^{j=P} (u_i^j)^2. \quad (8)$$

Thus, the functions defining the most characteristic structure of $u^n(\mathbf{x})$ are the eigenfunctions $\phi_k(\mathbf{x}) \equiv \boldsymbol{\phi}_k$ associated with the highest eigenvalues.

2.2. *A posteriori reduced modeling*

If some direct, off-line, simulations have been carried out and their results stored in memory, we can determine $u(\mathbf{x}_i, t^n) \equiv u_i^n, \quad \forall i \in [1, \dots, N]$,

$\forall n \in [1, \dots, P]$, and from these the r eigenvectors related to the r -highest eigenvalues $\phi_k = \phi_k(\mathbf{x}_i)$, $\forall i \in [1, \dots, N]$, $\forall k \in [1, \dots, r]$ (with $r \ll N$). Now, we can try to use these r eigenfunctions for approximating the solution of a problem slightly different to the one that has served to define $u(\mathbf{x}_i, t^n)$. For this purpose we need to define the matrix \mathbf{A}

$$\mathbf{A} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \dots & \phi_r(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \dots & \phi_r(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \dots & \phi_r(\mathbf{x}_N) \end{pmatrix}. \quad (9)$$

Now, we consider the linear system of equations resulting from the discretization of a partial differential equation (PDE) in the form

$$\mathbf{K} \mathbf{U} = \mathbf{F}. \quad (10)$$

Obviously, in the case of evolution problems \mathbf{F} contains the contribution of the solution at the previous time step.

Then, assuming that the unknown vector contains the nodal degrees of freedom, it can be expressed as

$$\mathbf{U} = \sum_{i=1}^{i=r} \zeta_i \phi_i = \mathbf{A} \boldsymbol{\zeta}, \quad (11)$$

the global system of equations now reads

$$\mathbf{K} \mathbf{U} = \mathbf{F} \Rightarrow \mathbf{K} \mathbf{A} \boldsymbol{\zeta} = \mathbf{F}, \quad (12)$$

and multiplying both terms by \mathbf{A}^T it results

$$\mathbf{A}^T \mathbf{K} \mathbf{A} \boldsymbol{\zeta} = \mathbf{A}^T \mathbf{F}, \quad (13)$$

which proves that the final system of equations is of low order, i.e. the dimensions of $\mathbf{A}^T \mathbf{K} \mathbf{A}$ are $r \times r$, with $r \ll N$, and the dimensions of both $\boldsymbol{\zeta}$ and $\mathbf{A}^T \mathbf{F}$ are $r \times 1$.

Note, on the other hand, that non-homogeneous boundary conditions (like rigid-body motions, for instance) do not impose any special difficulties to the method. A simple change of variables can deal with this, as proved in

[22].

3. Asymptotic numerical methods

The evolution of a non-linear solid mechanics problem is usually done with the help of iterative methods based upon the Newton-Raphson technique. This iterative structure of the method needs for an update of the tangent stiffness matrix at each iteration. Modified Newton methods, which employ a constant tangent matrix in order to save computation time are also very popular, but fail notably in the neighborhood of buckling points, for instance.

Nowadays different alternatives exist. Potier-Ferry and coworkers have developed in the last years a family of methods based upon asymptotic expansions that are able to accurately trace the non-linear solution of the problem with a minimum number of matrix updates, see for instance [16, 23]. In this technique, the nodal displacement vector \mathbf{U} and the load parameter λ (here it plays the role of a pseudo-time, ranging from 0 to 1) are represented by power series expansions with respect to a control parameter a . Introducing these expansions into the weak form of the problem a sequence of linear problems, that share a unique tangent stiffness matrix, is obtained. Thus, it seems natural to combine this asymptotic technique with a Ritz-like technique to obtain a reduced model that can accurately describe non-linear constitutive equations without the need for iterations.

3.1. Problem formulation

We consider here a Lagrangian description of a solid, whose points are described at the initial configuration by the vector \mathbf{X} . The solid is denoted by Ω_0 at its initial, undeformed configuration, while its boundary is denoted, as usual, by Γ . The prescribed displacements and tractions are applied to Γ_u and Γ_t . We assume that $\Gamma = \Gamma_u \cup \Gamma_t$ and $\Gamma_u \cap \Gamma_t = \emptyset$. After the body is loaded each material point is described by its position vector, \mathbf{x} , in the current deformed configuration.

Following classical approaches to the problem, the deformed configuration of the solid will be given by

$$\mathbf{x} = \mathbf{X} + \mathbf{u} \tag{14}$$

while the deformation gradient tensor can then be obtained as $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}$. By

recalling that the Green-Lagrange strain tensor, \mathbf{E} , is given by

$$\mathbf{E} = \frac{1}{2}(\mathbf{F}^T \mathbf{F} - \mathbf{1}) = \gamma_l(\mathbf{u}) + \gamma_{nl}(\mathbf{u}, \mathbf{u}) \quad (15)$$

where we have used the same notation as in [16]:

$$\begin{aligned} \gamma_{nl}(\mathbf{u}, \mathbf{v}) &= \frac{1}{2} \nabla(\mathbf{u}^T) \nabla(\mathbf{v}), \\ \gamma_l(\mathbf{u}) &= \frac{1}{2} (\nabla(\mathbf{u}^T) + \nabla(\mathbf{u})) \end{aligned} \quad (16)$$

Hyperelastic materials are based upon the definition of a strain-energy function, Ψ . The second Piola-Kirchhoff stress tensor \mathbf{S} can be obtained after this function by

$$\mathbf{S} = \frac{\partial \Psi}{\partial \mathbf{E}} \quad (17)$$

that is a symmetric tensor and is related to the first Piola-Kirchhoff stress tensor, \mathbf{P} , by $\mathbf{P} = \mathbf{F}\mathbf{S}$.

Using quantities related to the reference configuration the equilibrium equation is as following

$$\nabla \cdot \mathbf{P} + \mathbf{B} = 0 \quad \text{in } \Omega_0 \quad (18)$$

in which \mathbf{B} is the body force. Note that, for simplicity, quasi-static processes are assumed. Application of reduced-order modeling to dynamical problems has been successfully treated in [14].

The boundary conditions of the body are defined by

$$\begin{aligned} \mathbf{u}(\mathbf{X}) &= \bar{\mathbf{u}} \quad \text{on } \Gamma_u, \\ \mathbf{P}\mathbf{N} &= \bar{\lambda} \bar{\mathbf{t}} \quad \text{on } \Gamma_t \end{aligned} \quad (19)$$

where \mathbf{N} is the unit vector normal to Γ , $\bar{\mathbf{t}}$ is an applied traction and $\bar{\lambda}$ is a loading parameter. The work associated to the equilibrium equation is given by

$$\int_{\Omega_0} \mathbf{S} : \delta \mathbf{E} \, d\Omega = \bar{\lambda} \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta \mathbf{u} \, d\Gamma \quad \forall \delta \mathbf{u} \in H^1(\Omega) \quad (20)$$

where in the above equation $\delta \mathbf{E}$ is expressed by

$$\delta \mathbf{E} = \frac{1}{2}[\mathbf{F}^T \nabla(\delta \mathbf{u}) + \nabla(\delta \mathbf{u})^T \mathbf{F}] = \gamma_l(\delta \mathbf{u}) + \gamma_{nl_S}(\mathbf{u}, \delta \mathbf{u}) \quad (21)$$

where $\gamma_{nl_S}(\mathbf{u}, \delta \mathbf{u})$ is defined by

$$\gamma_{nl_S}(\mathbf{u}, \delta \mathbf{u}) = \gamma_{nl}(\mathbf{u}, \delta \mathbf{u}) + \gamma_{nl}(\delta \mathbf{u}, \mathbf{u}) \quad (22)$$

The Saint-Venant Kirchhoff model (linear elastic behaviour under large strains) is usually considered as a suitable approximation to most living tissues for real-time purposes, although it seems to be quite limited if a more detailed description is needed. Nevertheless, we can say that the state-of-the-art in the field assumes this model as suitable, see for instance, [14]. It is characterized by the energy function

$$\Psi = \frac{\lambda}{2}(\text{tr}(\mathbf{E}))^2 + \mu \mathbf{E} : \mathbf{E} \quad (23)$$

where λ and μ are Lamé's constants. The second Piola-Kirchhoff stress tensor can be obtained by

$$\mathbf{S} = \frac{\partial \Psi(\mathbf{E})}{\partial \mathbf{E}} = \mathbb{C} : \mathbf{E} \quad (24)$$

in which \mathbb{C} is the classical fourth-order elastic tensor.

3.2. An asymptotic numerical method for geometrically nonlinear problems

Asymptotic numerical methods, as created by Potier-Ferry and coworkers [16][24][25] (whose notation is also being used here) seek to obtain an analytic expression of the load-displacement curve of a solid or structure in terms of an expansion parameter “ a ”. This expansion of the terms of interest is developed in the neighborhood of a known solution $(\mathbf{u}^n; \mathbf{S}^n; \lambda^n)$ at step n . The series is truncated at order N , being $\bar{\lambda}$ the loading parameter introduced in the weak form. For Kirchhoff-Saint Venant models it is sufficient to take

$$\left\{ \begin{array}{c} \mathbf{u}^{n+1}(a) \\ \mathbf{S}^{n+1}(a) \\ \bar{\lambda}^{n+1}(a) \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{u}^n(a) \\ \mathbf{S}^n(a) \\ \bar{\lambda}^n(a) \end{array} \right\} + \sum_{p=1}^N a^p \left\{ \begin{array}{c} \mathbf{u}_p \\ \mathbf{S}_p \\ \bar{\lambda}_p \end{array} \right\}, \quad (25)$$

where $(\mathbf{u}_p, \mathbf{S}_p, \bar{\lambda}_p)$ are unknowns. Above, $(\mathbf{u}^{n+1}(a), \mathbf{S}^{n+1}(a), \bar{\lambda}^{n+1}(a))$ represents the solution along a portion of the loading curve described continuously

with respect to “ a ”. The introduction of Eq. (25) into Eq. (20) and Eq. (24) leads to a series of linear problems with the same tangent operator, a circumstance that seems to be optimal for the purposes of this work.

The series expansion of $\delta \mathbf{E}(\mathbf{u})$ gives

$$\delta \mathbf{E}^{n+1}(a) = \gamma_l(\delta \mathbf{u}) + \gamma_{nl_S}(\delta \mathbf{u}, \mathbf{u}^n) + \sum_{p=1}^n a^p \gamma_{nl_S}(\delta \mathbf{u}, \mathbf{u}_p) \quad (26)$$

The series expansions of \mathbf{S} gives in turn

$$\mathbf{S}^{n+1}(a) = \mathbb{C} : \mathbf{E}^{n+1}(a) = \mathbb{C} : [\gamma_{nl}(\mathbf{u}^n, \mathbf{u}^n) + \gamma_l(\mathbf{u}^n) + \quad (27)$$

$$\sum_{p=1}^n a^p \left(\gamma_l(\mathbf{u}_p) + \gamma_{nl_S}(\mathbf{u}^n, \mathbf{u}_p) + \sum_{i=1}^{p-1} \gamma_{nl_S}(\mathbf{u}_i, \mathbf{u}_{p-i}) \right)] \quad (28)$$

at order p we obtain

$$\mathbf{S}_p = \mathbb{C} : \left\{ \gamma_l(\mathbf{u}_p) + \gamma_{nl_S}(\mathbf{u}^n, \mathbf{u}_p) + \sum_{i=1}^{p-1} \gamma_{nl_S}(\mathbf{u}_i, \mathbf{u}_{p-i}) \right\} \quad (29)$$

Introducing this asymptotic expansion into Eq. (24) gives

$$\int_{\Omega_0} \left\{ \left(\mathbf{S}^n + \sum_{p=1}^N a^p \mathbf{S}_p \right) : \left(\gamma_l(\delta \mathbf{u}) + \gamma_{nl_S}(\mathbf{u}^n, \delta \mathbf{u}) + \sum_{p=1}^N a^p \gamma_{nl_S}(\mathbf{u}_p, \delta \mathbf{u}) \right) \right\} d\Omega = \quad (30)$$

$$\left(\bar{\lambda}^n + \sum_{p=1}^N a^p \bar{\lambda}_p \right) \Psi_{ext}(\delta \mathbf{u}) \quad (31)$$

with $\Psi_{ext}(\delta \mathbf{u}) = \int_{\Gamma_t} \mathbf{t} \cdot \delta \mathbf{u} d\Gamma$. Introducing Eq. (29) into Eq. (30) gives a series of problems, one for each level of p ($p = 1, \dots, N$), that take the form

$$\mathcal{L}(\delta \mathbf{u}, \mathbf{u}^n) = \bar{\lambda}_p \Psi_{ext}(\delta \mathbf{u}) + F_p^{nl}(\delta \mathbf{u}) \quad (32)$$

with

$$\begin{aligned} \mathcal{L}(\delta \mathbf{u}, \mathbf{u}^n) = & \\ \int_{\Omega} \{ \mathbf{S}^n : \gamma_{nl_S}(\mathbf{u}^n, \mathbf{u}_p) + [\gamma_l(\delta \mathbf{u}) + \gamma_{nl_S}(\mathbf{u}_p, \delta \mathbf{u})] : \mathbb{C} : [\gamma_l(\mathbf{u}_p) + \gamma_{nl_S}(\mathbf{u}^n, \mathbf{u}_p)] \} d\Omega & \end{aligned} \quad (33)$$

and where $\mathbf{F}_p^{nl}(\delta \mathbf{u})$ is equal to zero at order one and at order p it reads

$$\begin{aligned} \mathbf{F}_p^{nl}(\delta \mathbf{u}) = & \\ - \int_{\Omega} \left\{ \sum_{i=1}^{p-1} \mathbf{S}_i : \gamma_{nl_S}(\mathbf{u}_{p-i}, \delta \mathbf{u}) + \sum_{i=1}^{p-1} [\gamma_{nl_S}(\mathbf{u}_i, \mathbf{u}_{p-i})] : \mathbb{C} : [\gamma_l(\delta \mathbf{u}) + \gamma_{nl_S}(\mathbf{u}^n, \delta \mathbf{u})] \right\} d\Omega & \end{aligned} \quad (34)$$

Replacing Eq. (29) into Eq. (30), and discretization of Eq. (32) by standard finite element procedure, it leads to a sequence of linear problems in the form

$$\text{order } 1 \left\{ \begin{array}{l} \mathbf{K}_t \mathbf{u}_1 = \bar{\lambda}_1 \mathbf{f} \\ \mathbf{u}_1^T \mathbf{u}_1 + \bar{\lambda}_1^2 = 1 \end{array} \right. \quad (35)$$

$$\text{order } p \left\{ \begin{array}{l} \mathbf{K}_t \mathbf{u}_p = \bar{\lambda}_p \mathbf{f} + \mathbf{f}_p^{nl}(\mathbf{u}_i) \quad i < p \\ \mathbf{u}_p^T \mathbf{u}_1 + \bar{\lambda}_p \bar{\lambda}_1 = 0 \end{array} \right. \quad (36)$$

where \mathbf{K}_t denotes the tangent stiffness matrix associated with Eq. (33), common to the problems at different orders p . It is the same as the one applied in a classical iterative algorithm like Newton-Raphson, but without updating. Above, \mathbf{u}_p is the discretized form of the displacement field at order p , \mathbf{f} is the loading vector and \mathbf{f}_p^{nl} is the discretized form associated with $\mathbf{F}_p^{nl}(\delta \mathbf{u})$ in Eq. (34), which at order p only depends on the values of \mathbf{u}_i , $i < p$. The solution of the problem can be obtained as a sequence of p linear problems:

$$\text{order } 1 \left\{ \begin{array}{l} \hat{\mathbf{u}} = \{\mathbf{K}_t\}^{-1} \mathbf{f} \\ \lambda_1 = \frac{1}{\sqrt{\hat{\mathbf{u}}^T \hat{\mathbf{u}}_1}} \\ \mathbf{u}_1 = \bar{\lambda}_1 \hat{\mathbf{u}} \end{array} \right. \quad (37)$$

$$\text{order } p \begin{cases} \mathbf{u}_p^{nl} = \{\mathbf{K}_t\}^{-1} \mathbf{f}_p^{nl} \\ \lambda_p = -\bar{\lambda}_1 \{\mathbf{u}_p^{nl}\}^T \mathbf{u}_1 \\ \mathbf{u}_p = \frac{\bar{\lambda}_p}{\lambda_1} \mathbf{u}_1 + \mathbf{u}_p^{nl} \end{cases} \quad (38)$$

4. A geometrically nonlinear reduced-order model

As explained before the solution of reduced-order models leads to a very fast calculation of the system of equations but the price to pay is the impossibility of non-linear simulations. On the other hand, by using ANM we can obtain the solution of highly nonlinear problems in the neighbourhood of an equilibrium point very accurately. In this section we combine these two methods to solve geometrically nonlinear problems. This approach was previously established in a very different context by Yvonnet and coworkers [25] and later generalized in [26].

Here we assume that the POD basis has been calculated as explained in section 2. The terms of the asymptotic expansion associated with the displacements are expressed as functions of POD basis as

$$\mathbf{u}_p = \sum_{m=1}^M \phi^m \zeta_p^m = \mathbf{A} \zeta_p, \quad (39)$$

where ζ_p are the degrees of freedom. So the new asymptotic expansion of \mathbf{u} is expressed by

$$\mathbf{u}^{n+1}(a) = \mathbf{A} \left(\zeta^n + \sum_{p=1}^N a^p \zeta_p \right) \quad (40)$$

where (ζ^n, λ^n) represents the previous known solution. Introducing Eq. (40) into Eq. (32) results in

$$\mathcal{L}(\mathbf{A} \zeta_p, \mathbf{A} \delta \zeta) = \bar{\lambda}_p \delta \Psi_{ext}(\mathbf{A} \delta \zeta) + F_p^{nl}(\mathbf{A} \delta \zeta) \quad (41)$$

after finite element discretization and taking into account that the POD basis is orthonormal, we arrive at the following sequence of linear systems of equations:

$$\text{at order } 1 \begin{cases} \mathbf{A}^T \mathbf{K}_t \mathbf{A}^T \zeta_1 = \bar{\lambda}_1 \mathbf{A}^T \mathbf{f} \\ \zeta_1^T \zeta_1 + \bar{\lambda}_1^2 = 1 \end{cases} \quad (42)$$

$$\text{at order } p \begin{cases} \mathbf{A}^T \mathbf{K}_t \mathbf{A} \hat{\boldsymbol{\zeta}}_p = \bar{\lambda}_p \mathbf{A}^T \mathbf{f} + \mathbf{f}_p^{nl} \\ \hat{\boldsymbol{\zeta}}_p^T \hat{\boldsymbol{\zeta}}_1 + \bar{\lambda}_p \bar{\lambda}_1 = 0 \end{cases} \quad (43)$$

The size of the above equations depends on the number of the POD basis and the chosen level of expansion. In general, very low orders are necessary, as will be demonstrated below. It is noteworthy that these problems have a common tangent stiffness matrix that can be computed off-line with important time savings.

The solution of these equations can be obtained as

$$\text{order } 1 \begin{cases} \hat{\boldsymbol{\zeta}} = \{\mathbf{A}^T \mathbf{K}_t \mathbf{A}\}^{-1} \mathbf{A}^T \mathbf{f} \\ \bar{\lambda}_1 = \frac{1}{\sqrt{\hat{\boldsymbol{\zeta}}^T \hat{\boldsymbol{\zeta}}_{+1}}} \\ \hat{\boldsymbol{\zeta}}_1 = \bar{\lambda}_1 \hat{\boldsymbol{\zeta}} \\ \hat{\mathbf{u}} = \mathbf{A} \hat{\boldsymbol{\zeta}}, \mathbf{u}_1 = \bar{\lambda}_1 \hat{\boldsymbol{\zeta}}_1 \end{cases} \quad (44)$$

$$\text{order } p \begin{cases} \hat{\boldsymbol{\zeta}}_p^{nl} = \{\mathbf{A}^T \mathbf{K}_t \mathbf{A}\}^{-1} \mathbf{A}^T \mathbf{f}_p^{nl} \\ \bar{\lambda}_p = -\bar{\lambda}_1 \{\hat{\boldsymbol{\zeta}}_p^{nl}\}^T \hat{\boldsymbol{\zeta}}_1 \\ \hat{\boldsymbol{\zeta}}_p = \frac{\bar{\lambda}_p}{\bar{\lambda}_1} \hat{\boldsymbol{\zeta}}_1 + \hat{\boldsymbol{\zeta}}_p^{nl} \\ \mathbf{u}_p = \mathbf{A} \hat{\boldsymbol{\zeta}}_p \end{cases} \quad (45)$$

5. Interpolation of reduced models

From Eq. (13) we can notice that the reduced model employs the tangent stiffness matrix, \mathbf{K} , linearized from the non-linear problem formulation at a given time instant. Instead of inverting the full stiffness matrix of size $N \times N$, we employ model reduction techniques to invert the matrix $\mathbf{A}^T \mathbf{K} \mathbf{A}$, of size $r \times r$, that is much lower than the original size, as mentioned before.

However, this tangent stiffness matrix \mathbf{K} corresponds to a given state of the model (i.e., a given load position and load value, for instance, in mechanical problems). Different load values would lead to different matrices \mathbf{K} along the loading path (this is due to the non-linear character of the problem). This problem is now translated into the sequence of problems (44)-(45). This means that the non-linearity of the problem has somehow been translated to the \mathbf{f}_p^{nl} term of Eqs. (44)-(45), whose evaluation constitutes now the most time-consuming part of the algorithm.

The philosophy of the method proposed in [15] is to perform some direct simulations for a set of different positions of the surgical tool. For each tool position a reduced-order basis set can thus be computed. But the position of

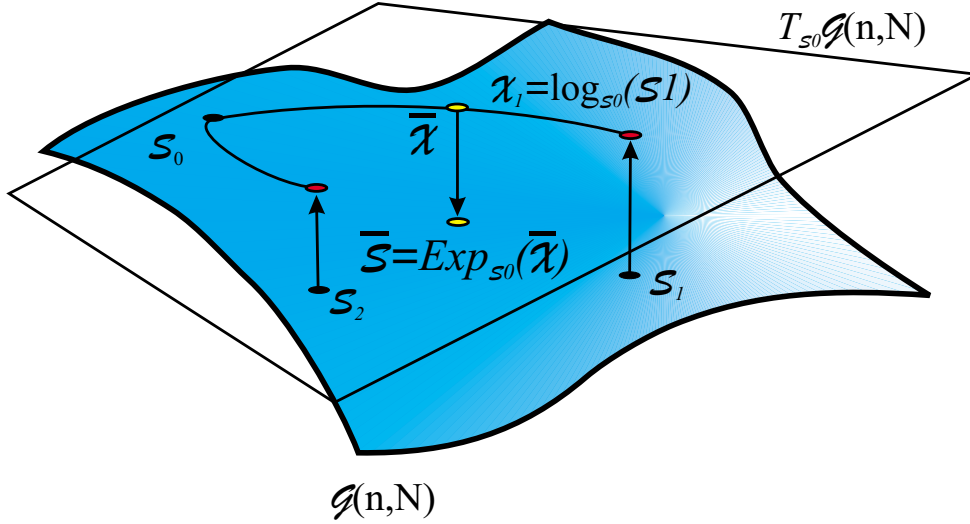


Figure 1: Schematic description of the Grassman manifold formed by the set of basis functions of all the reduced modes of a given model.

the tool at the on-line simulation does not need to be coincident to the pre-computed ones, and therefore some interpolation scheme is needed if the load is placed at positions far from the originally computed ones. This complex interpolation procedure has now two different parts: on one side we need to interpolate the reduced basis sets, from pre-computed complete models, to an arbitrary position of the load, for instance. On the other side, we need to also interpolate the terms \mathbf{f}_p^{nl} , again from the pre-computed complete models to the arbitrary, reduced ones.

5.1. Interpolation of the reduced basis: a geometrical approach

Farhat and coworkers have pointed out that the set of empirical eigenfunctions given by Eq. (4) for a given model forms the so-called Grassman manifold $\mathcal{G}(n, N)$ [27]. Therefore, in order to interpolate the set of basis functions, that consequently do not form a vectorial space, we must move to the tangent plane at a point of the manifold, which is a “flat” space, interpolate there, and project back to the manifold, as schematically explained in Fig. 1.

In this way, the columns of \mathbf{A} constitute a basis of the subspace \mathcal{S}_0 of dimension n of the space \mathbb{R}^N . At each point \mathcal{S} of the manifold $\mathcal{G}(n, N)$ one can define a tangent plane of the same dimension, $\mathcal{T}_{\mathcal{S}}$, with its points defined by a matrix $\mathbf{\Gamma} \in \mathbb{R}^{N \times n}$. The exponential mapping $Exp_{\mathcal{S}}$ transforms χ in an n -dimensional subspace \mathcal{S}' given by a matrix $\mathbf{A}' \in \mathbb{R}^{N \times n}$, such that

$$\mathbf{\Gamma} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (\text{Singular value decomposition})$$

$$\mathbf{A}' = \mathbf{A}\mathbf{V} \cos \Sigma + \mathbf{U} \sin \Sigma$$

Conversely, the logarithmic mapping $Log_{\mathcal{S}}$, defines a map between a point in the neighborhood of $\mathcal{S} \in \mathcal{G}(n, N)$ and the tangent plane at the origin. Thus, the image of \mathcal{S}' , in a neighborhood of \mathcal{S} , given by the logarithmic mapping, $\chi = Log_{\mathcal{S}}\mathcal{S}' \in \mathcal{T}_{\mathcal{S}}$ will be

$$(\mathbf{I} - \mathbf{A}\mathbf{A}^T)\mathbf{A}'(\mathbf{A}^T\mathbf{A}')^{-1} = \mathbf{U}\Sigma\mathbf{V}^T \quad (\text{Singular value decomposition})$$

$$\mathbf{\Gamma} = \mathbf{U} \tan^{-1}(\Sigma)\mathbf{V}^T$$

So, consider, for instance, $\mathbf{A}_0 \in \mathbb{R}^{N \times n}$ and $\mathbf{A}_1 \in \mathbb{R}^{N \times n}$, two matrices representing two subspaces, obtained for different parameters of the model (for instance, load positions, but the theory is completely general for other parameters of the model), s_0 and s_1 . Let $\mathcal{S}_0, \mathcal{S}_1$ be the two subspaces originated by considering parameters s_0 and s_1 . Let, in turn, $\mathcal{Y}(t)$ be the geodesic line that joins both subspaces (points in the Grassmann manifold), having \mathcal{S}_0 as origin. In that case, the initial derivative of the geodesic line, that belongs to the tangent plane at \mathcal{S}_0 , will be

$$\dot{\mathcal{Y}}_0 = Log_{\mathcal{S}_0}\mathcal{S}_1$$

such that the matrix representing this initial derivative of the geodesic will be (see [27] for a complete proof of this)

$$(\mathbf{I} - \mathbf{A}_0\mathbf{A}_0^T)\mathbf{A}_1(\mathbf{A}_0^T\mathbf{A}_1)^{-1} = \mathbf{U}\Sigma\mathbf{V}^T$$

$$\mathbf{\Gamma} = \mathbf{U} \tan^{-1}(\Sigma)\mathbf{V}^T$$

Let $\tilde{\mathcal{S}}$ denote the point of the Grassmann manifold representing the reduced-order basis for the new value of the parameter. \tilde{s} . The initial derivative of the new geodesic line, joining \mathcal{S}_0 and the sought interpolated subspace $\tilde{\mathcal{S}}$, will be

$$\tilde{\dot{\mathcal{Y}}}_0 = \tilde{r}\dot{\mathcal{Y}}_0$$

with

$$\tilde{r} = \frac{\tilde{s} - s_0}{s_1 - s_0}$$

The computation of the singular value decomposition is not a very time-consuming task. For instance, on a PC at 2GHz, the computation of the Matlab instruction `svd(A)`, with \mathbf{A} a random matrix of 5×5000 elements

takes on average much less than $1ms$, still compatible with real-time constraints.

5.2. POD with interpolation

A less rigorous, but much simpler, method to interpolate among previously computed reduced models was established in [28]. This procedure is known in the literature as Proper Orthogonal Decomposition with Interpolation (PODI) [28]. Let $[\mathbf{U}^m(\mathbf{X}_i)]_{i=1}^k$ be the response of the system for k different parameter values (in this case, loads at positions \mathbf{X}_i), at time step m . The basic algorithm is described as

1. Perform the complete model simulation for each parameter value.
2. Although in the standard PODI technique the POD procedure is applied to the complete set of snapshots of the system to obtain an orthonormal basis $\mathbf{A} = [\phi_1 \cdots \phi_n]$, we proceed by just applying the POD to each complete model (i.e., to each load position). Thus, we obtain an orthonormal basis for each system's parameter value.
3. Interpolate from neighboring basis sets to obtain a reduced basis for the new state of the system. To this end we employ finite element piece-wise linear interpolation.

Results obtained with this technique showed to be much more efficient and accurate than those obtained with the more rigorous technique base upon interpolation on the Grassman manifold. This surprising result is still not well understood by the authors and is currently one of our research topics. In any case, results presented in section 6 were obtained with the PODI technique.

Finally, in order to establish a good set of basis $\mathbf{A} = [\phi_1 \cdots \phi_n]$, whatever the chosen interpolation technique, a good sampling strategy should be chosen for the position of the loads in the complete models. In this case, a lattice was established over the domain. Loads were applied at every lattice position. Obviously, other strategies can also be implemented. If the lattice is not dense enough, it is clear that the mentioned interpolation described in this section would lead to less accurate results.

5.3. Interpolation of the non-linear force terms

The other big ingredient of the resulting method, namely the non-linear force term \mathbf{f}_p^{nl} of Eqs. (44)-(45) needs also to be interpolated between pre-computed models, since it is too computationally-demanding to be evaluated

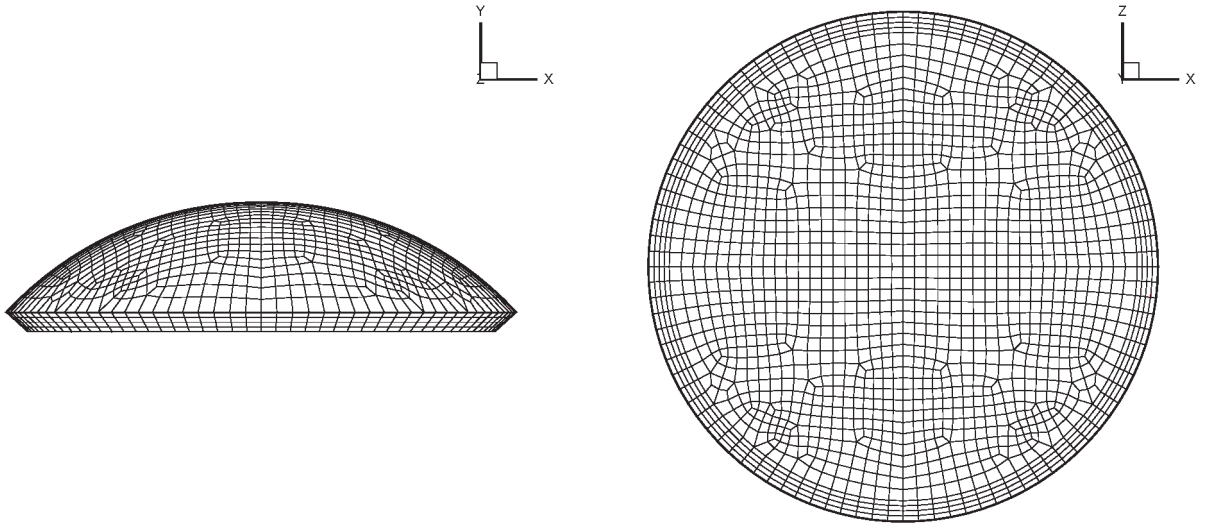


Figure 2: Geometry of the finite element model for the human cornea.

in real-time. To this end, we establish a piece-wise polynomial (FE-like) interpolation among complete models. This simple procedure has rendered excellent results, as will be shown in the subsequent section.

6. Numerical examples

In order to show the performance of this method we have applied it to some academic examples devoted to the simulation of soft tissues. The first one is the pinching of a model of the human cornea, while the second is devoted to liver palpation.

6.1. Pinching the human cornea

In this example we applied $0.014N$ to nine neighboring nodes located at the central region of the cornea, that was meshed using trilinear hexahedral elements. It consisted of 8514 nodes and 7182 elements. The mesh is shown in Figure 2 in two views. The material properties of the cornea were assumed to be $E = 2MPa$ and $\nu = 0.48$. A more detailed constitutive model can be found at [29].

In this example nine nodes were applied that provide decent approximation. The solution has been obtained using ANM-POD for $p = 1, \dots, 6$. The tangent stiffness matrix at the origin, that is independent of the load state of the organ, is employed for all the simulations. In order to verify the results we have computed the solution by full FE Newton-Raphson method. The

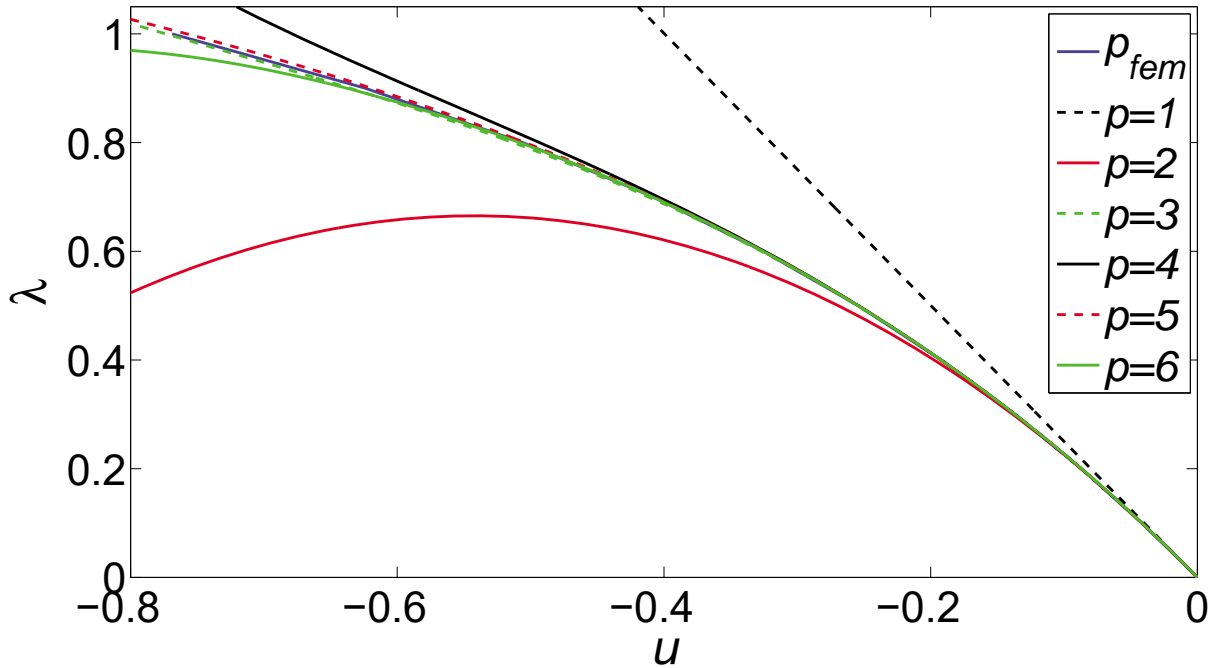


Figure 3: The loading factor vs maximum displacement [mm] for the Pinched cornea.

loading factor (λ) has been plotted versus the maximum displacement (in modulus) in Figure 3.

As it can be seen the results have good accuracy with Newton-Raphson solution. The deformed cornea obtained using ANM-POD for $\lambda = 1$ is shown in Figure 4 and the one obtained using full FE Newton-Raphson is depicted in Figure 5.

If we consider a case in which the load is placed at a position for which a reduced model has not been stored, the interpolation procedure, as described in section 5, is applied. As mentioned before, the tangent stiffness matrix at origin is identical for all load states. The procedure follows by (linearly) interpolating the reduced basis set from the four nearest neighbor pre-computed states and also by interpolating the non-linear force term \mathbf{f}_p^{nl} for each order of the expansion.

The resulting load-displacement curve is depicted in Fig. 6. The excellent agreement with the complete model results (denoted by p_{FEM} in the legend) is noticeable.

6.2. Palpation of the liver

In this example we consider the palpation of a liver with a surgical tool, assumed perfectly rigid. The liver is the biggest gland in the human body, after the skin. Liver geometry has been obtained from the SOFA project

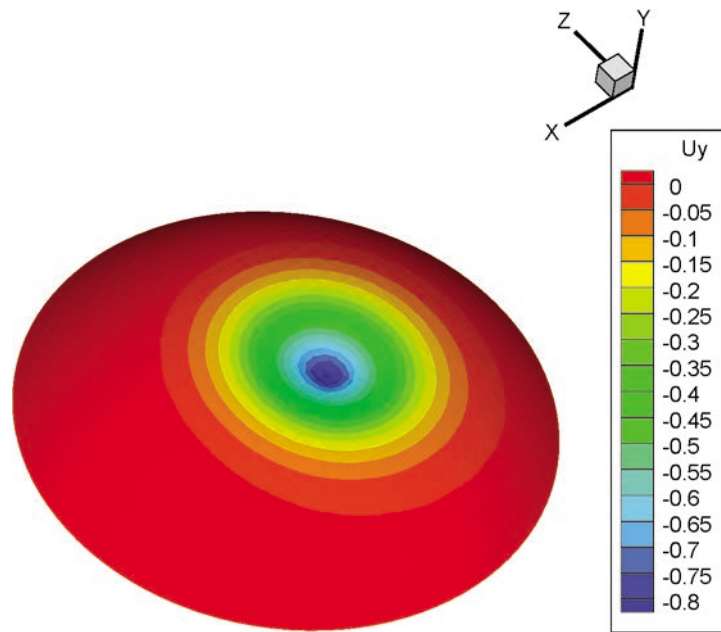


Figure 4: u_y , [mm], contour of the pinched cornea obtained by ANM-POD.

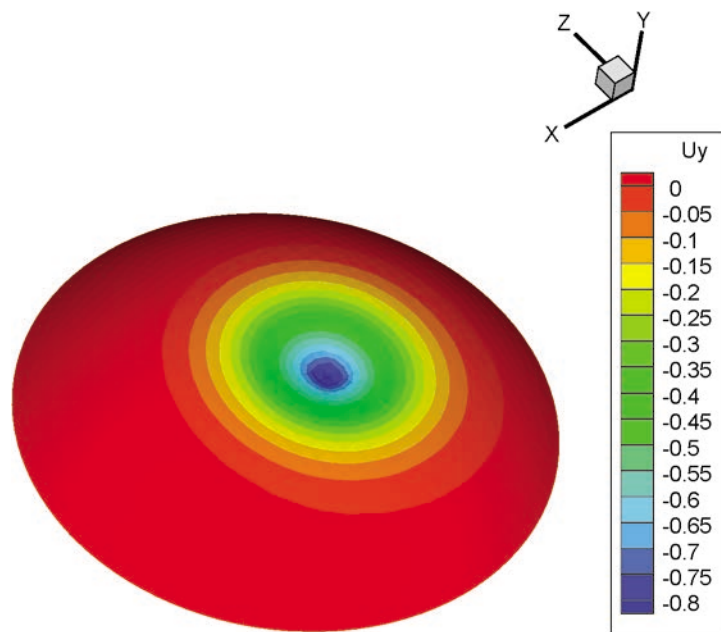


Figure 5: u_y , [mm], contour of the pinched cornea obtained by Newton-Raphson.

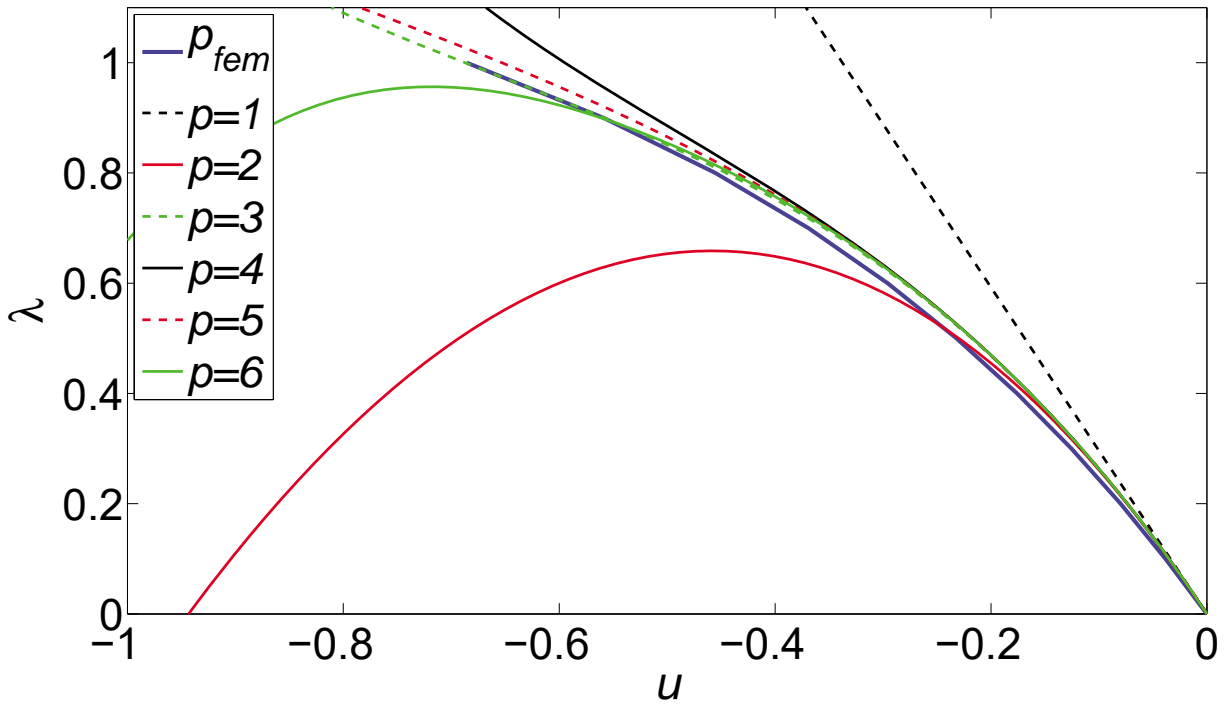


Figure 6: The loading factor vs minimum displacement, $[mm]$, for the Pinched cornea. The resulting behavior is interpolated among four previously stored reduced models.

[30] and post-processed in order to obtain a mesh composed by 2853 nodes and 10519 tetrahedra, whose geometry is shown in Fig. 7. The anterior surface of the liver is considered free, while the posterior one was assumed to be supported over different organs (it is connected to the diaphragm by the coronary ligament, for instance). The inferior vena cava travels along the posterior surface, and the liver is frequently assumed clamped at that location. Although the assumed boundary conditions are not strictly correct from a physiological point of view, our main interest is to show that the model can be solved under real-time constraints with reasonable accuracy.

Although the literature on the mechanical properties of the liver is not very detailed, we have assumed a Young's modulus of 160 kPa, and a Poisson coefficient of 0.48, thus nearly incompressible [1].

As in the previous example, a load of $1.2N$ has been applied at an arbitrary point and the reduced model has been constructed, composed in this case by the 9 modes depicted in Fig 8, that capture the 99.9% of the energy of the system.

The FE complete model was left to run until lack of convergence of the Newton-Raphson algorithm. A standard Newton-Raphson algorithm was employed, without any modification such as arc-length methods or anything

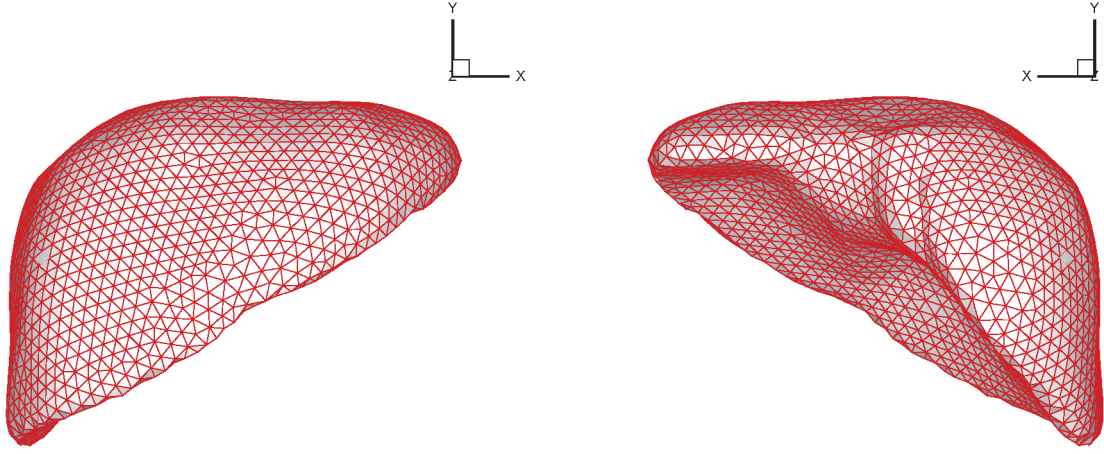


Figure 7: Geometry of the finite element model for the liver.

similar. The load (in terms of load factor λ)-displacement curve obtained by POD-ANM techniques is depicted in Fig. 9. It is worth noting the high accuracy of the results for a wide range of displacements (up to $\approx 5\text{mm}$ under the tool tip) for expansions of order 5 and 6, less than 5%. This accuracy can also be noticed from Figs. 10 and 11, representing the displacements for FE complete mode and POD-ANM (sixth-order expansion) techniques, respectively.

6.3. Timings

The examples below have been performed on a laptop running Matlab on a Windows-operated computer at 2 GHz., with 2 Gb RAM. The code prototype is obviously thought for the design and testing of new algorithms. Under these circumstances, the examples ran at around 20 Hz, which is enough for visual perception, but not for haptic environments. This can be improved by considering more sophisticated programming and not a general-purpose language such as Matlab. The summary of the computational cost for a typical simulation is as follows:

- Interpolation of the basis and the non-linear force term: $3.9 \cdot 10^{-3}s$.
- Construction of the reduced stiffness matrix $\mathbf{A}^T \mathbf{K}_t \mathbf{A}$ (note that the complete tangent stiffness matrix at the origin, \mathbf{K}_t , is identical for all simulations and therefore could be previously stored in memory): $4.7 \cdot 10^{-2}s$.

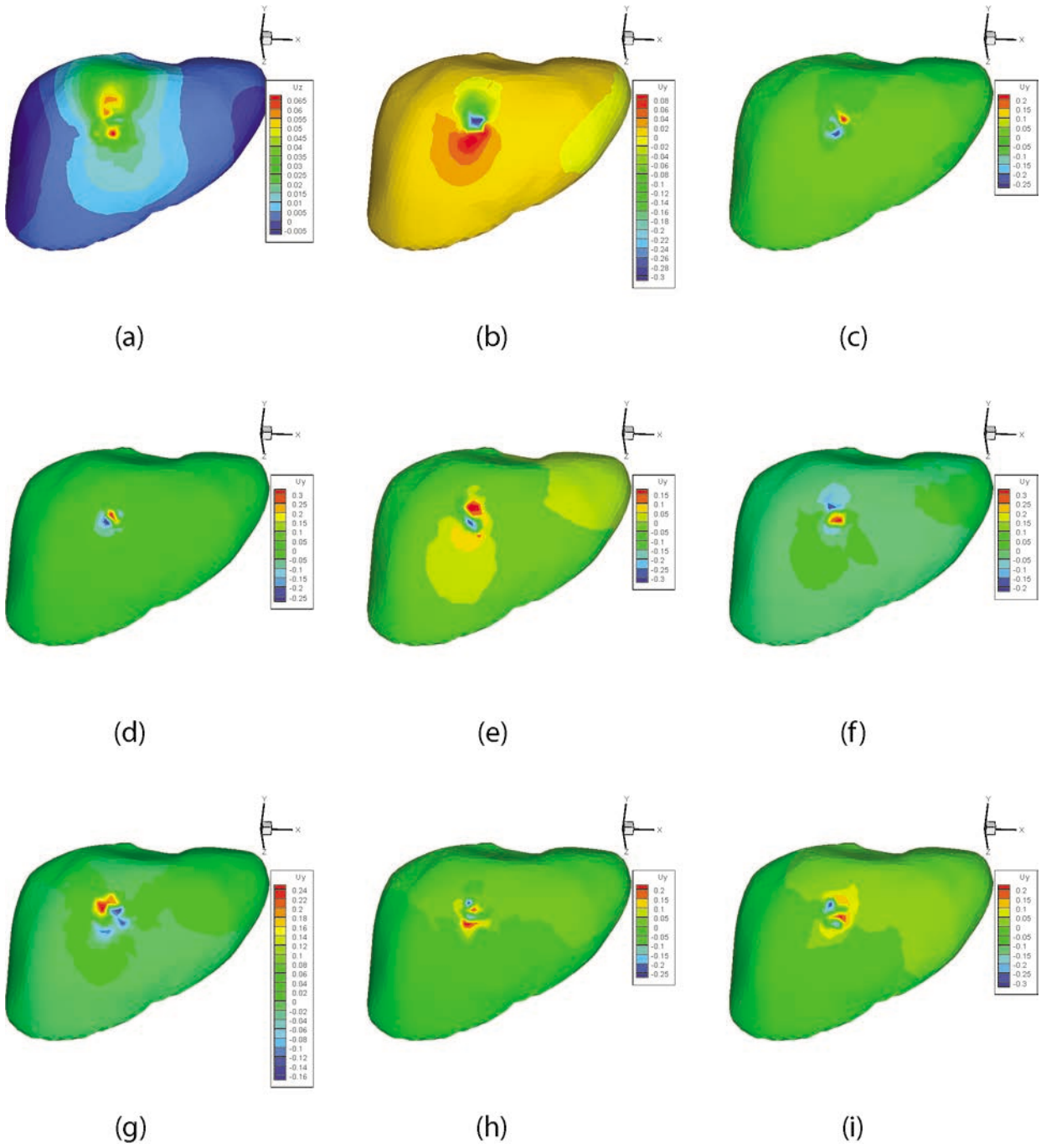


Figure 8: Nine most important eigenmodes for the simulation of the liver palpation. The corresponding eigenvalues are: $2.69E - 03$ (a), $8.28E - 05$ (b), $8.82E - 07$ (c), $1.07E - 08$ (d), $6.73E - 11$ (e), $5.21E - 13$ (f), $2.52E - 15$ (g).

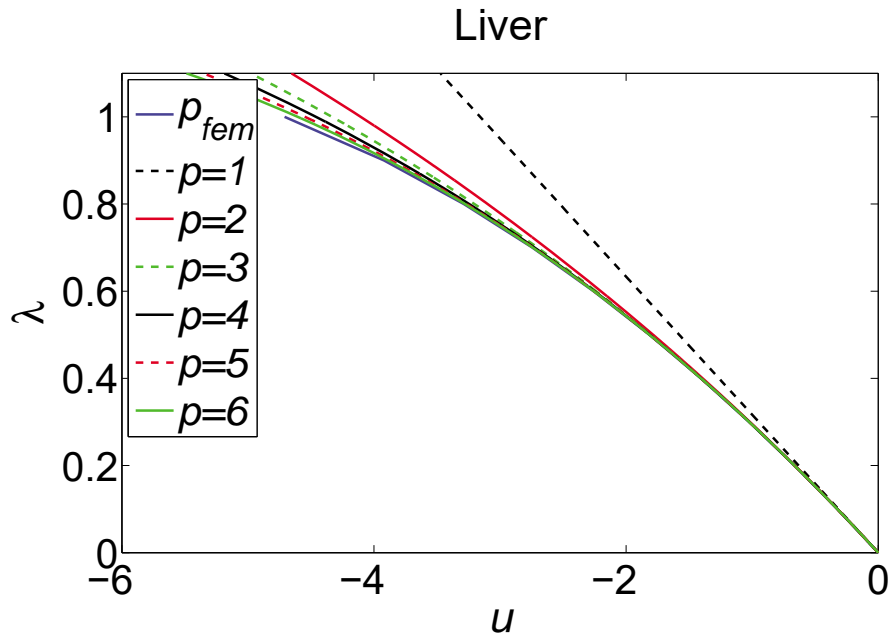


Figure 9: Load-displacement, $[mm]$, curve for the liver palpation for different orders of approximation. The continuous blue line represents the solution for the complete model employing Newton-Raphson methods.

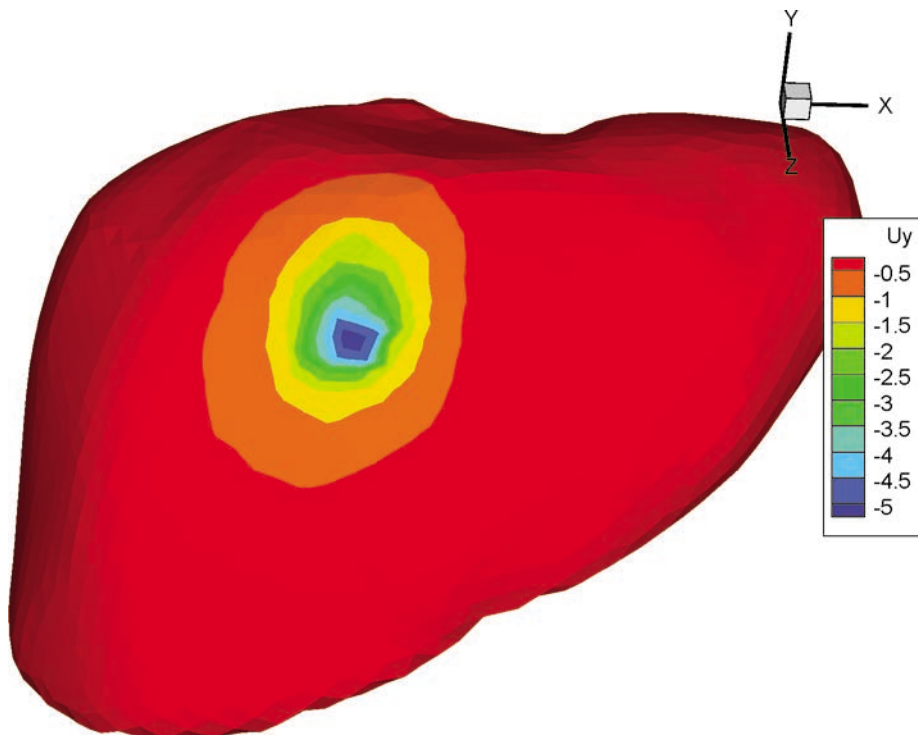


Figure 10: Result for the FEM model, displacement u_y , $[mm]$.

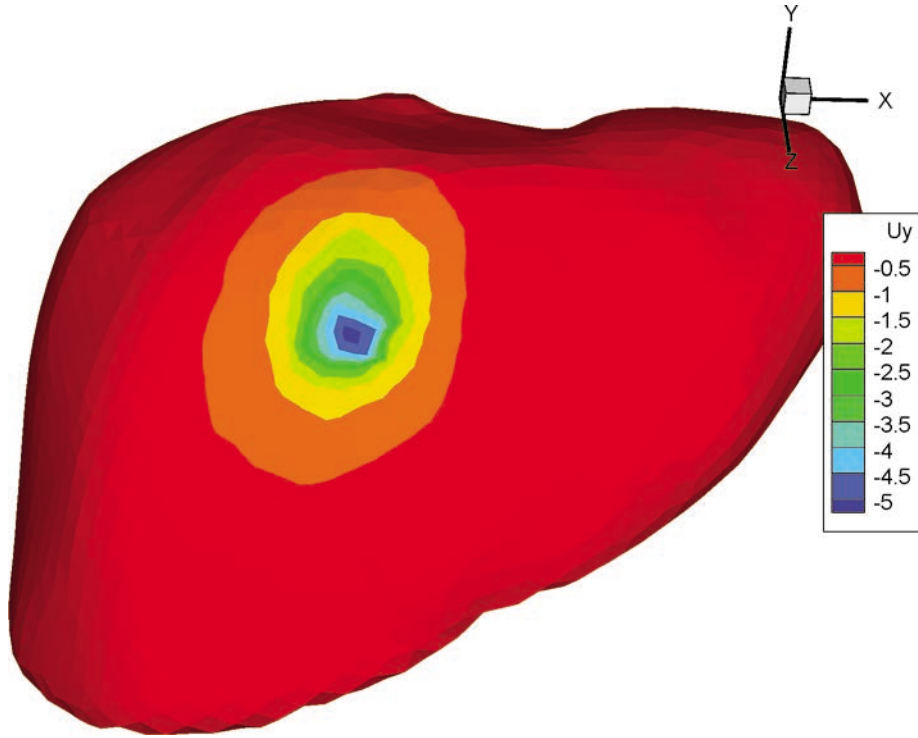


Figure 11: Result for the reduced-ANM model, displacement u_y , [mm].

- Solving the resulting system of equations (6×6 in this particular example): $3.1 \cdot 10^{-4}s$.

The use of Graphics Processing Units [10] is obviously another possibility to achieve true real-time performance, that seems to be at hand.

7. Conclusions

In a previous work by the authors [15], model reduction techniques were applied for real-time simulation of the palpation of the human cornea. It was demonstrated that even for very sophisticated constitutive laws, model reduction techniques provided reasonable results at real-time frequencies for haptic environments. However, it has been noticed that, although reduced models obtained by POD techniques are optimal in some statistical sense, no tangent stiffness matrix updating was possible under such severe constraints imposed by real-time. Thus, the models obtained by employing this technique are still linear.

In this paper we have pursued a different strategy. Through the combination of model reduction techniques and an Asymptotic Numerical Method

(ANM) we have shown that it is possible to obtain an expansion of the solution in a neighborhood of a previous equilibrium state that provides a very simple algorithm that fits reasonably well to the true solution within some radius of convergence. The main feature of the method is that it possesses only one tangent stiffness matrix that does not need to be updated within a large interval of strains (much larger than traditional Newton-Raphson schemes). It has been shown how this radius of convergence is enough for many applications, involving strains on the order of 10%.

Other approaches based upon reduced models exist, see [14], for instance. The main difference between the approach followed in this work and that of Barbic and James is that in the latter case a general-purpose reduced basis is employed. These basis functions are obtained from mass-scaled principal component analysis. These basis functions are thought to be optimal for any load state of the solid. In the present approach different sets of basis functions are obtained for different load (and possibly boundary conditions) states that can be changed according with the user experience. Basis sets for any state different from the precomputed ones are obtained after interpolation, as explained in the previous sections. In section 5.1 we have explained how the set of reduced models for different loading states of the organ forms a manifold. Therefore, it is crucial for the method to work well that the set of complete models to be solved to feed the algorithm is chosen adequately. For the moment, the brute-force approach seems to be out of reach. In our opinion, this process should be guided by surgeons, by indicating the most probable loading states for each surgical procedure, distinguishing, for instance, between experienced surgeons and medicine students.

The method has been developed for Saint Venant-Kirchhoff constitutive laws. This kind of laws is among the best state-of-the-art existing models [14], but is judged to be poor for some applications. It is well-known that it suffers from instabilities when subjected to compression (although no instabilities have been found in the examples tested by the authors), so it would be interesting to extend the proposed technique to some more sophisticated constitutive laws, such as neo-hookean laws, for instance, or other much better suited for specific applications [29].

Despite the facts commented above, the proposed technique seems to be an appealing method for the simulation of linear elastic materials undergoing large strains at real time. The authors continue to work in the development of the technique. For instance, the issue of the simulation of cutting in the context of reduced models is also a bottleneck of the method. This constitutes

our current effort of research and will be published elsewhere.

References

- [1] H. Delingette and N. Ayache. Soft tissue modeling for surgery simulation. In N. Ayache, editor, *Computational Models for the Human Body*, Handbook of Numerical Analysis (Ph. Ciarlet, Ed.), pages 453–550. Elsevier, 2004.
- [2] Herve Delingette. Towards realistic soft tissue modeling in medical simulation. Technical Report INRIA RR-3506.
- [3] K. Waters. A physical model of facial tissue and muscle articulation derived from computer tomography data. *Visualization in Biomedical Computing*, 574, 1992.
- [4] R. Kenedi, T. Gibson, J. Evans, and J. Barbenel. Tissue mechanics. *Physics in Medicine and Biology*, 20(5):163–169, 1975.
- [5] H. Delingette, G. Subsol, S. Cotin, and J. Pignon. A craniofacial surgery simulation testbed. *Visualization in Biomedical Computing*, October 1994.
- [6] U. Meier, O. Lopez, C. Monserrat, M. C. Juan, and M. Alcañiz. Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, (77):183–197, 2005.
- [7] C. Monserrat, U. Meier, M. Alcañiz, F. Chinesta, and M.C. Juan. A new approach for the real-time tissue deformations in surgery simulation. *Computer Methods and Programs in Biomedicine*, 64(2):77–85, 2001.
- [8] N. Famaey and J. Sloten. Soft tissue modeling for applications in virtual surgery and surgical robotics. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(4):351–366, August 2008.
- [9] Dhanannjay Deo and Suvranu De. Phyness: A physics-driven neural networks-based surgery simulation system with force feedback. In *Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Salt Lake City, UT, USA, pages 30–34, 2009.

- [10] Z. Taylor, M. Cheng, and S. Ourselin. High-speed nonlinear finite element analysis for surgical simulation using graphics processing units. *IEEE Transactions on Medical Imaging*, 27(5):650–663, 2008.
- [11] Yi-Je Lim and Suvranu De. Real time simulation of nonlinear tissue response in virtual surgery using the point collocation-based method of finite spheres. *Computer Methods in Applied Mechanics and Engineering*, 196:3011–3024, 2007.
- [12] S. Capell, S. Green, B. Curless, T. Duchamp, and Z. Popovic. Interactive skeleton-driven dynamic deformations. In *ACM SIGGRAPH*, 2002.
- [13] P. Krysl, S. Lall, and J.E. Marsden. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *Int. J. Numer. Meth. in Engng.*, 51:479–504, 2001.
- [14] Jernej Barbič and Doug L. James. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics (SIGGRAPH 2005)*, 24(3):982–990, August 2005.
- [15] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer Methods and Programs in Biomedicine*, 91:223–231, 2008.
- [16] B. Cochelin, N. Damil, and M. Potier-Ferry. The asymptotic-numerical method: an efficient perturbation technique for non-linear structural mechanics. *Revue Europeenne des Elements Finis*, 3(2):281–297, 1994.
- [17] D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar. On the a priori Model Reduction: Overview and recent developments. *Archives of Computational Methods in Engineering*, 12(1):91–128, 2006.
- [18] K. Karhunen. Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae, ser. Al. Math. Phys.*, 37, 1946.
- [19] M. M. Loève. *Probability theory*. The University Series in Higher Mathematics, 3rd ed. Van Nostrand, Princeton, NJ, 1963.
- [20] L. Sirovich. Turbulence and the dynamics of coherent structures part I: coherent structures. *Quarterly of applied mathematics*, XLV:561–571, 1987.

- [21] E. N. Lorenz. *Empirical Orthogonal Functions and Statistical Weather Prediction*. MIT, Departement of Meteorology, Scientific Report Ni₂¹, Statistical Forecasting Project, 1956.
- [22] D. González, A. Ammar, F. Chinesta, and E. Cueto. Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81:637–659, 2010.
- [23] H. Zahrouni, M. Potier-Ferry, H. Elasmaraoui, and N. Damil. Asymptotic numerical method for nonlinear constitutive laws. *Revue europeenne des elements finis*, 7(7):841–869, 1998.
- [24] N. Damil and M. Potier-Ferry. A new method to compute perturbed bifurcation: application to the buckling of imperfect elastic structures. *International Journal of Engineering Science*, 26(9):943–957, 1990.
- [25] J. Yvonnet, H. Zahrouni, and M. Potier-Ferry. A model reduction method for the post-buckling analysis of cellular microstructures . *Computer Methods in Applied Mechanics and Engineering*, 197:265–280, 2007.
- [26] S. Niroomandi, I. Alfaro, E. Cueto, and F. Chinesta. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 11:1180–1206, 2010.
- [27] D. Amsallem and Ch. Farhat. An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal*, 46:1803–1813, 2008.
- [28] Hung V. Ly and Hien T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33:223–236, 2005.
- [29] V. Alastrue, B. Calvo, E. Peña, and M. Doblaré. Biomechanical modeling of refractive corneal surgery. *Journal of Biomechanical Engineering-Transactions of the ASME*, 128:150–160, 2006.
- [30] Jérémie Allard, Stéphane Cotin, François Faure, Pierre-Jean Bensoussan, François Poyer, Christian Duriez, Hervé Delingette, and Laurent Grisoni. SOFA an Open Source Framework for Medical Simulation. In

Medicine Meets Virtual Reality (MMVR'15), Long Beach, USA, February 2007.