HAL open science

Real Time Simulation of Biological Soft Tissues: A PGD Approach

Siamak Niroomandi, David Gonzalez, Icíar Alfaro, Felipe Bordeu Weldt, Adrien Leygue, Elías Cueto, Francisco Chinesta

To cite this version:

Siamak Niroomandi, David Gonzalez, Icíar Alfaro, Felipe Bordeu Weldt, Adrien Leygue, et al.. Real Time Simulation of Biological Soft Tissues: A PGD Approach. International Journal for Numerical Methods in Biomedical Engineering, 2013, 29 (5), pp.586-600. 10.1002/cnm.2544 . hal-01007231

HAL Id: hal-01007231

https://hal.science/hal-01007231v1

Submitted on 17 Jan 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.

Distributed under a Creative Commons Attribution 4.0 International License

Real-time simulation of biological soft tissues: a PGD approach

S. Niroomandi [1], D. González [1], I . A l f a r o [1], F. Bordeu [2],
A. Leygue [2], E. Cueto [1] and F. Chinesta [2]

[1] *Aragon Institute of Engineering Research, Universidad de Zaragoza, Zaragoza, Spain*
[2] *EADS Corporate International Chair, Ecole Centrale de Nantes, Nantes, France*

We introduce here a novel approach for the numerical simulation of nonlinear, hyperelastic soft tissues at kilohertz feedback rates necessary for haptic rendering. This approach is based upon the use of proper generalized decomposition techniques, a generalization of PODs. Proper generalized decomposition techniques can be considered as a means of a priori model order reduction and provides a physics-based meta-model without the need for prior computer experiments. The suggested strategy is thus composed of an offline phase, in which a general meta-model is computed, and an online evaluation phase in which the results are obtained at real time. Results are provided that show the potential of the proposed technique, together with some benchmark test that shows the accuracy of the method.

KEY WORDS: real time; surgery; model order reduction; proper generalized decomposition

1. INTRODUCTION

Real-time simulation is one of the most challenging scenarios for simulation-based engineering sciences. The term *real time* strongly depends on the particular pursued application, but surgery simulation is among the most restrictive ones. Haptic surgery simulators compute the response of biological soft tissues and give it back to the peripherals at, at least, 25 Hz of feedback rate if visual realism is needed and, notably, 500 Hz–1 kHz if haptic (force) response is desired [1, 2]. But biological soft tissues are known to be highly nonlinear [3–5], very frequently modeled in a hyperelastic framework. It is well-known, in addition, that at least nonlinear strain measures should be incorporated into the simulation, otherwise performing arbitrarily bad in terms of visual perception, spurious gain in volume, and others [2].

This feedback rate for nonlinear problems constitutes indeed a great challenge for nowaday simulation techniques based upon FEMs. This is perhaps the ultimate reason for the lack, up to our knowledge, of surgery simulators of the second generation [1], that is, those that incorporate state-of-the-art constitutive modeling of soft tissues to the simulation. A few references can be cited that incorporate nonlinear tissue behavior (mainly Kirchhoff–Saint Venant hyperelasticity) (see [6–8]). Many of them are indeed based upon explicit finite elements, possibly implemented in graphics processing units. This is due to the very astringent conditions dictated by the haptic feedback rates that prevent the use, in nowaday computers, of standard Newton-like methods for the solution of nonlinear systems of equations.

Recently, model order reduction has been seen as a powerful means to achieve real-time performance in the simulation of nonlinear solids. For instance, Barbič and James [9] proposed recently a method for model order reduction of Kirchhoff–Saint Venant solids at haptic feedback rates. For more sophisticated constitutive laws, recent work on POD methods showed that visual feedback performance can be easily obtained, but this is not the case very often if we need haptic feedback [10–12]. In this case, linearization schemes for the resulting nonlinear systems of equations is mandatory, although not straightforward to accomplish. Very often, Newton strategies are out of reach, due to the inherent lack of time employed during stiffness matrix updates. In [13, 14], a different strategy was developed that employed a combination of POD techniques with asymptotic numerical methods [15–18]. In that case, no tangent stiffness matrix updates are necessary, and only a set of linear problems, all with the same stiffness matrix, must be solved.

Proper generalized decomposition (PGD) methods, on the contrary, arose recently as a means to overcome the so-called *curse of dimensionality* associated to problems defined in high-dimensional spaces [19–23]. Although the origins of the technique can be traced back to the large time increment method and its associated radial loading [24], PGD is nowadays seen as a powerful method of model order reduction that generalizes POD (and hence its name). Its field of application has gone far beyond the initial objectives, and it is now being applied in a variety of fields: simulation-based control of processes [25, 26], efficient simulation of plates and shells [27], and simulation of gene regulatory networks [28], to cite a few.

The main idea for the extension of a technique initially developed to deal with high-dimensional problems to a more general setting lies precisely in its ability to treat standard models as if they were multidimensional. The key idea was initially put forth in [29]. In it, parametric equations were cast into a multidimensional setting, thus taking advantage of the PGD solution structure. Going one step forward, PGD can advantageously be employed into real-time simulation frameworks by simply considering not only all parameters but also all possible boundary conditions (including initial boundary conditions, see [30]) as new dimensions of the problem.

The resulting PGD solution to the problem is expressed as a finite sum of separable functions that provides actually a meta-model for the problem, for which no prior computer experiment (also known as *snapshots* in the model order reduction community) is necessary. This meta-model can then be successfully applied in real time to obtain the response of the system at kilohertz rates, as will be demonstrated in subsequent sections of this paper. This approach allows even to solve models on handheld devices such as tablets and smartphones, thus opening the range of possible applications of the technique.

In this paper, we present a novel technique for the simulation of biological soft tissues under hyperelasticity assumptions at haptic feedback rates. It is based on the use of the aforementioned PGD approach and an explicit linearization of the weak form generated by nonlinear strain measures. The use of PGD therefore allows not only for a complete generalization of previous works in the field (see [10, 11]) but also to a completely new formulation of the problem. For the problem at hand, POD techniques compute the solution (the so-called snapshots) of different complete models for different contact positions between a surgical tool—scalpel—and an organ. By performing a statistical analysis on top of these results, POD techniques extract the so-called *modes*, that is, those displacement fields that best represent the solution of the complete problems. These modes, taken as Ritz-like, global basis, are then used to approximate problems different to the original ones, that is, for contact positions not initially considered. Therefore, the design of an appropriate simulation, or computer experiment, campaign is the key aspect of the method. Also, how to efficiently and accurately interpolate among reduced basis is another crucial aspect not fully resolved.

On the contrary, PGD methods consider a parametric problem. In this case, the parameter is the contact position. By formulating the parametric problem as a high-dimensional one, PGD methods allow for the efficient solution and ulterior storage of the model in the form of a sum of separable functions. Therefore, rather than creating reduced models for particular positions of the tool, PGD computes a general solution for *any* position of the tools, so that no subsequent interpolation of reduced models is necessary, nor the computation of snapshots. Thus, the proposed method is based upon an offline phase in which this general solution is computed and an online one in which the solution is only evaluated at impressive feedback rates, here on the order of kilohertz.

The paper is organized as follows. In Section 2, we introduce the basics of PGD applied to the problem of a hyperelastic solid under moving punctual loads, which is the most frequent case in surgery simulation. In Section 3, a very simple linearization of the nonlinear problem is introduced that allows for a simple yet effective computation of the PGD approach to the problem. Although this simple linearization is by no means the only possible one, its performance is analyzed in Section 4, through a series of benchmark problems. It is shown how the PGD approach to the problem of real-time simulation of soft tissue deformation opens new insights on how the problem can be attacked.

2. A PROPER GENERALIZED DECOMPOSITION APPROACH TO VIRTUAL SURGERY

As already mentioned in Section 1, the key issue in the use of PGD approaches for real-time simulation, and the one that makes it completely different in spirit from POD, lies in the formulation of the original problem as a parametric one. This parametric problem is then reformulated as a high-dimensional problem by considering each parameter as a new coordinate in the state space. The PGD method then looks for an effective solution in the form of a finite sum of separable functions, so as to be able to avoid the curse of dimensionality associated to high-dimensional problems and mesh-based discretization techniques.

In this framework, the problem of determining the response of an organ to the load transmitted by the contact with a surgical tool could be formulated to determine the displacement at any point of the model, $\boldsymbol{u}(x, y, z)$, for any load position $\boldsymbol{s}$ and for any force vector orientation and module, $\boldsymbol{t}$, thus rendering a problem defined in the physical space ($\mathbb{R}^3$), plus a six-dimensional state space ($\mathbb{R}^6$).

For the sake of simplicity in the following development, and without loss of generality, we assume a load vector $\boldsymbol{t}$ with unit module and oriented in the vertical direction. This renders a problem defined in $\mathbb{R}^6(\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s}))$, with all the characteristics of the aforementioned one.

Let us consider the weak form of the equilibrium equations (balance of linear momentum). Again, for the sake of simplicity, we omit inertia terms. The interested reader could consult PGD-ODE [30] for the treatment of ODEs in the framework of PGD. Under these assumptions, the weak form of the problem, extended to the whole geometry of the organ, Ω and the portion of its boundary that is accessible to the surgeon, $\bar{\Gamma} \subset \Gamma_t$,[‡] consists in finding the displacement $\boldsymbol{u} \in \mathcal{H}^1$ such that for all $\boldsymbol{u}^* \in \mathcal{H}^1_0$:

$$\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s \boldsymbol{u}^* : \boldsymbol{\sigma} \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} \boldsymbol{u}^* \cdot \boldsymbol{t} \, \mathrm{d}\Gamma \mathrm{d}\bar{\Gamma} \tag{1}$$

where $\Gamma = \Gamma_u \cup \Gamma_t$ represents the boundary of the organ, divided into essential and natural regions, and where $\Gamma_t = \Gamma_{t1} \cup \Gamma_{t2}$, that is, regions of homogeneous and nonhomogeneous, respectively, natural boundary conditions. Here, $\boldsymbol{t} = \boldsymbol{e}_k \cdot \delta(\boldsymbol{x} - \boldsymbol{s})$, where δ represents the Dirac delta function and $\boldsymbol{e}_k$ the unit vector along the z-coordinate axis (we consider here, for the ease of exposition, a unit load directed toward the negative z axis of reference).

Once regularized, the Dirac delta term is approximated by a truncated series of separable functions in the spirit of the PGD method, that is,

$$t_j \approx \sum_{i=1}^{m} f_j^i(\boldsymbol{x}) g_j^i(\boldsymbol{s}) \tag{2}$$

where m represents the order of truncation and f_j^i, g_j^i represent the jth component of vectorial functions in space and boundary positions, respectively.

[‡]Typically, of all of the natural regions of the boundary, $\Gamma_t \approx \Gamma$—which in this case coincides with virtually all the boundary of organs, $\Gamma = \partial\Omega$, because they are not fixed or clamped but in contact to other organs— only a portion is accessible to the surgeon. In minimally invasive surgery, surgeons operate through a small incision on the skin of the patient, having access only to a limited portion of the boundary, here termed $\bar{\Gamma}$.

The PGD approach to the problem is characterized by the construction, in an iterative way, of an approximation to the solution in the form of a finite sum of separable functions. Assume that we have converged to a solution, at iteration n of this procedure,

$$u_j^n(\boldsymbol{x},\boldsymbol{s}) = \sum_{k=1}^{n} X_j^k(\boldsymbol{x}) \cdot Y_j^k(\boldsymbol{s}), \tag{3}$$

where the term u_j refers to the jth component of the displacement vector, $j = 1, 2, 3$, and functions $\boldsymbol{X}^k$ and $\boldsymbol{Y}^k$ represent the separated functions used to approximate the unknown field, obtained in previous iterations of the PGD algorithm.

If we look for an improvement of this approximation, the $(n+1)$th term will look like

$$u_j^{n+1}(\boldsymbol{x},\boldsymbol{s}) = u_j^n(\boldsymbol{x},\boldsymbol{s}) + R_j(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}), \tag{4}$$

where $\boldsymbol{R}(\boldsymbol{x})$ and $\boldsymbol{S}(\boldsymbol{s})$ are the sought functions that improve the approximation.

In this framework, the admissible variation of the displacement will be given by

$$u_j^*(\boldsymbol{x},\boldsymbol{s}) = R_j^*(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}) + R_j(\boldsymbol{x}) \cdot S_j^*(\boldsymbol{s}). \tag{5}$$

At this point, several options are at hand so as to determine the new pair of functions $\boldsymbol{R}$ and $\boldsymbol{S}$. The most frequently used, due to both its ease of implementation and good convergence properties, in general, is a fixed-point algorithm in which functions $\boldsymbol{R}$ and $\boldsymbol{S}$ are sought iteratively. We describe briefly the implementation of this algorithm.

2.1. Computation of $S(\boldsymbol{s})$ assuming $R(\boldsymbol{x})$ is known

In this case, following standard assumptions of variational calculus, we have

$$u_j^*(\boldsymbol{x},\boldsymbol{s}) = R_j(\boldsymbol{x}) \cdot S_j^*(\boldsymbol{s}), \tag{6}$$

or, equivalently, $\boldsymbol{u}^*(\boldsymbol{x},\boldsymbol{s}) = \boldsymbol{R} \circ \boldsymbol{S}^*$, where the symbol '$\circ$' denotes the so-called entry-wise Hadamard or Schur multiplication for vectors. Once substituted into Equation (1), it gives

$$\begin{aligned}\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\boldsymbol{R} \circ \boldsymbol{S}^*) : \mathbf{C} : \nabla_s \left(\sum_{k=1}^{n} \boldsymbol{X}^k \circ \boldsymbol{Y}^k + \boldsymbol{R} \circ \boldsymbol{S} \right) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} \\ = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\boldsymbol{R} \circ \boldsymbol{S}^*) \cdot \left. \sum_{k=1}^{m} \boldsymbol{f}^k \circ \boldsymbol{g}^k \right) \mathrm{d}\Gamma \mathrm{d}\bar{\Gamma},\end{aligned} \tag{7}$$

or equivalently (we omit obvious functional dependencies)

$$\begin{aligned}\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\boldsymbol{R} \circ \boldsymbol{S}^*) : \mathbf{C} : \nabla_s(\boldsymbol{R} \circ \boldsymbol{S}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} \\ = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\boldsymbol{R} \circ \boldsymbol{S}^*) \cdot \left(\sum_{k=1}^{m} \boldsymbol{f}^k \circ \boldsymbol{g}^k \right) \mathrm{d}\Gamma \mathrm{d}\bar{\Gamma} - \int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\boldsymbol{R} \circ \boldsymbol{S}^*) \cdot \mathcal{R}^n \mathrm{d}\Omega \mathrm{d}\bar{\Gamma},\end{aligned} \tag{8}$$

where $\mathcal{R}^n$ represents

$$\mathcal{R}^n = \mathbf{C} : \nabla_s \boldsymbol{u}^n. \tag{9}$$

Because the symmetric gradient operates on spatial variables only, we have

$$\begin{aligned}
&\int_{\bar{\Gamma}} \int_{\Omega} (\nabla_s \boldsymbol{R} \circ \boldsymbol{S}^*) : \mathbf{C} : (\nabla_s \boldsymbol{R} \circ \boldsymbol{S}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} \\
&\quad = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\boldsymbol{R} \circ \boldsymbol{S}^*) \cdot \left(\sum_{k=1}^{m} \boldsymbol{f}^k \circ \boldsymbol{g}^k \right) \mathrm{d}\Gamma \mathrm{d}\bar{\Gamma} - \int_{\bar{\Gamma}} \int_{\Omega} (\nabla_s \boldsymbol{R} \circ \boldsymbol{S}^*) \cdot \mathcal{R}^n \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}
\end{aligned} \tag{10}$$

where all the terms depending on $\boldsymbol{x}$ are known, and hence, we can compute all integrals over Ω and Γ_{t2} (support of the regularization of the initially punctual load) to derive an equation to compute $\boldsymbol{S}(\boldsymbol{s})$.

2.2. *Computation of $R(\boldsymbol{x})$ assuming $S(\boldsymbol{s})$ is known*

Equivalently, in this case, we have

$$u_j^*(\boldsymbol{x}, \boldsymbol{s}) = R_j^*(\boldsymbol{x}) \cdot S_j(\boldsymbol{s}), \tag{11}$$

which, once substituted into Equation (1), gives

$$\begin{aligned}
&\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s (\boldsymbol{R}^* \circ \boldsymbol{S}) : \mathbf{C} : \nabla_s \quad \sum_{k=1}^{n} \boldsymbol{X}^k \circ \boldsymbol{Y}^k + \boldsymbol{R} \circ \boldsymbol{S} \Bigg) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} \\
&\quad = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\boldsymbol{R}^* \circ \boldsymbol{S}) \cdot \quad \sum_{k=1}^{m} \boldsymbol{f}^k \circ \boldsymbol{g}^k \Bigg) \mathrm{d}\Gamma \mathrm{d}\bar{\Gamma}.
\end{aligned} \tag{12}$$

In this case, all the terms depending on $\boldsymbol{s}$ (load position) can be integrated over $\bar{\Gamma}$, leading to a generalized elastic problem to compute function $\boldsymbol{R}(\boldsymbol{x})$.

This simple algorithm renders, in general, excellent convergence properties (see [22] and references therein).

3. ONE POSSIBLE EXPLICIT LINEARIZATION OF THE FORMULATION

The formulation introduced in Section 2 assumes implicitly small strains. But this assumption has been found to be very insufficient for virtual surgery. Strains are large very often, and when solved in a small strain setting, organs appear to suffer an unphysical gain in volume that renders the simulations clearly nonrealistic [1]. Although real-time simulation of surgery does not look nowadays for accuracy levels similar to those common in usual engineering practice (quoting Cotin and Bro-Nielsen, [31], '... the model may be physically correct if it looks right'), the inclusion of nonlinear strain measures seems to be crucial.

Soft tissues are frequently formulated under hyperelasticity assumptions [5]. Again, for the sake of simplicity, we refer ourselves to a Kirchhoff–Saint Venant constitutive framework. Despite being very limited (and even unstable under compression due to the lack of polyconvexity of the strain energy functional), Kirchhoff–Saint Venant constitutive equations are widely used at this moment for real-time simulation of soft tissues (see [7, 13, 32] among others).

The Kirchhoff–Saint Venant model is characterized by the energy density functional given by

$$\Psi = \frac{\lambda}{2}(\mathrm{tr}(\boldsymbol{E}))^2 + \mu \boldsymbol{E} : \boldsymbol{E} \tag{13}$$

where λ and μ are Lame's constants. The Green–Lagrange strain tensor, $\boldsymbol{E}$, has the form

$$\boldsymbol{E} = \frac{1}{2}\left(\boldsymbol{F}^T\boldsymbol{F} - 1\right) = \nabla_s \boldsymbol{u} + \frac{1}{2}\left(\nabla \boldsymbol{u} \cdot \nabla \boldsymbol{u}^T\right) \tag{14}$$

where $\boldsymbol{F} = \nabla \boldsymbol{u} + \boldsymbol{I}$ is the gradient of deformation tensor. The second Piola–Kirchhoff stress tensor can be obtained by

$$\boldsymbol{S} = \frac{\partial \Psi(\boldsymbol{E})}{\partial \boldsymbol{E}} = \mathbf{C} : \boldsymbol{E} \tag{15}$$

in which $\mathbf{C}$ is the fourth-order constitutive (elastic) tensor.

Very few works have been written about PGD approximations for nonlinear solid mechanics problems, other than the Ladeveze's works in complex thermomechanical nonlinear models [24]. Here, we focus on the nonlinear and parametric case within a fully separated representation. In this case, we restrict ourselves to quasi-static problems, for the sake of simplicity, and therefore introduce a pseudo-time $t \in [0, 1]$ to perform the linearization.

Consistent linearizations of the resulting set of equations in the framework of PGD approximations are far from being trivial, so here we keep the formulation as simple as possible by performing a simple explicit linearization of the weak form of the problem.

Thus, load is applied along a series of time increments Δt, provoking increments in the displacement $\Delta \boldsymbol{u}(\boldsymbol{x}, \boldsymbol{s})$. At each time increment, a PGD fixed-point alternating directions algorithm similar to those introduced in Section 2 is employed. So, if we introduce the nonlinear strain measure given by Equation (14), into this incremental framework, we have (we omit obvious functional dependencies for clarity)

$$\boldsymbol{E}^{t+\Delta t} = \nabla_s(\boldsymbol{u}^t + \Delta \boldsymbol{u}) + \frac{1}{2}\left(\nabla(\boldsymbol{u}^t + \Delta \boldsymbol{u}) \cdot \nabla^T(\boldsymbol{u}^t + \Delta \boldsymbol{u})\right). \tag{16}$$

Similarly, admissible variation of strain reads

$$\begin{aligned} \boldsymbol{E}^* &= \nabla_s(\Delta \boldsymbol{u}^*) + \frac{1}{2}(\nabla(\Delta \boldsymbol{u}^*)) \cdot \nabla^T(\boldsymbol{u}^t + \Delta \boldsymbol{u}) + \frac{1}{2}\nabla(\boldsymbol{u}^t + \Delta \boldsymbol{u}) \cdot \nabla^T(\Delta \boldsymbol{u}^*) \\ &= \nabla_s(\Delta \boldsymbol{u}^*) + \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T(\boldsymbol{u}^t + \Delta \boldsymbol{u}). \end{aligned} \tag{17}$$

Once substituted into the weak form of the equilibrium equation, Equations (16) and (17) provide, for the left hand side term of Equation (1)—strain energy term—,

$$\begin{aligned} \int_{\bar{\Gamma}} \int_{\Omega(t)} \boldsymbol{E}^* : \mathbf{C} : \boldsymbol{E} \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} = \int_{\bar{\Gamma}} \int_{\Omega(t)} &\left(\nabla_s(\Delta \boldsymbol{u}^*) + \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T(\boldsymbol{u}^t + \Delta \boldsymbol{u})\right) : \mathbf{C} \\ &: \left(\nabla_s\left(\boldsymbol{u}^t + \Delta \boldsymbol{u}\right) + \frac{1}{2}\left(\nabla(\boldsymbol{u}^t + \Delta \boldsymbol{u}) \cdot \nabla^T(\boldsymbol{u}^t + \Delta \boldsymbol{u})\right)\right) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}. \end{aligned} \tag{18}$$

The simplest linearization of Equation (18) consists of keeping in the formulation only constant terms and those linear in $\Delta \boldsymbol{u}$. We thus arrive at a weak form composed of 10 terms:

$$
\begin{aligned}
&\int_{\bar{\Gamma}} \int_{\Omega(t)} \boldsymbol{E}^* : \mathbf{C} : \boldsymbol{E} \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma} \\
&\quad = \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla_s(\Delta \boldsymbol{u}^*) : \mathbf{C} : \nabla_s \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T1} + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla_s(\Delta \boldsymbol{u}^*) : \mathbf{C} : \nabla_s(\Delta \boldsymbol{u}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T2} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla_s(\Delta \boldsymbol{u}^*) : \mathbf{C} : \frac{1}{2} \nabla \boldsymbol{u}^t \cdot \nabla^T \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T3} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla_s(\Delta \boldsymbol{u}^*) : \mathbf{C} : \nabla \boldsymbol{u}^t \cdot \nabla^T (\Delta \boldsymbol{u}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T4} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T \boldsymbol{u}^t : \mathbf{C} : \nabla_s \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T5} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T \boldsymbol{u}^t : \mathbf{C} : \nabla_s(\Delta \boldsymbol{u}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T6} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T \boldsymbol{u}^t : \mathbf{C} : \frac{1}{2} \nabla \boldsymbol{u}^t \cdot \nabla^T \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T7} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T \boldsymbol{u}^t : \mathbf{C} : \nabla \boldsymbol{u}^t \cdot \nabla^T (\Delta \boldsymbol{u}) \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T8} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T (\Delta \boldsymbol{u}) : \mathbf{C} : \nabla_s \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T9} \\
&\qquad + \underbrace{\int_{\bar{\Gamma}} \int_{\Omega(t)} \nabla(\Delta \boldsymbol{u}^*) \cdot \nabla^T (\Delta \boldsymbol{u}) : \mathbf{C} : \frac{1}{2} \nabla \boldsymbol{u}^t \cdot \nabla^T \boldsymbol{u}^t \, \mathrm{d}\Omega \mathrm{d}\bar{\Gamma}}_{T10} .
\end{aligned}
\tag{19}
$$

This renders a very simple scheme that has revealed, however, for judicious choice of the time step Δt, reasonable convergence properties, as will be demonstrated in Section 4.

Remark 1

The original work of P. Ladeveze on the large time increment method [24] combined a space–time separated representation and thus produces a nonincremental solution of the problem. Generalized to this case, the displacement would be sought in the form $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, s, t)$. We have preferred, for simplicity of exposition, to keep the formulation as simple as possible, but the explicit linearization proposed in Equation (19) is by no means the only possible one.

Remark 2

Another possible choice for the aforementioned linearization is the standard forward Euler scheme. It has been noted, however, that instabilities in the results appear at zones subjected to compression, a typical characteristic of Kirchhoff–Saint Venant models [33]. These are analyzed in Section 4. However, no spurious deformation modes have been observed by employing the simple explicit algorithm stated in Equation (19).

4. NUMERICAL RESULTS

4.1. Validation: nonlinear rod

To validate the explicit PGD approach introduced in Section 2 earlier, we considered the simple case of a Kirchhoff–Saint Venant beam subjected to a pure traction force F. In this simple case, the model can be simplified to a one-dimensional one, and an analytical solution for the axial displacement u at the bar tip can be obtained as

$$u = \left(\beta - 1 + \frac{1}{3\beta}\right), \tag{20}$$

where

$$\beta = \quad K + \sqrt{K^2 - \frac{1}{27}}\Bigg)^{\frac{1}{3}}, \tag{21}$$

and, in turn, $K = F/EA$, with E as the Young's modulus of the material and A the area of the cross section at the undeformed configuration.

A model was thus constructed by considering a bar of length $L = 400$ mm, $A = 40 \times 40$ mm^2, $E = 1.0$ MPa, and $F = 320$ N. Load F, however, is considered to be applied at any point along the bar axis. We therefore compute a two-dimensional solution $u = u(x, s)$, where x represents the position along the bar axis and s the point of application of the load.

This simple example served to know the importance of the chosen time step in the overall convergence properties of the proposed method. As can be noticed from Figure 1, the error for a time step of 10^{-3} is $\mathcal{O}(10^{-4})$. Note that one single PGD approach is used, which is enriched at each time step, not a different PGD approximation within each time step.

It is important to remark that the real-time strategy here introduced is based upon the computation (only once and off-line) of a general, high-dimensional solution of the problem once for life. This general solution is then evaluated at real-time feedback rates but not recomputed. That is why the time taken in the offline computation of this general solution is not so important, because it will be performed only once.

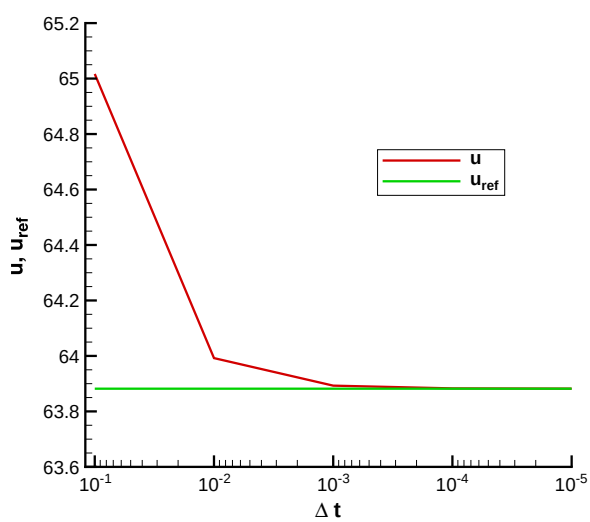

Figure 1. Convergence of the tip displacement toward the reference solution as a function of the chosen time step.

4.2. *Kirchhoff–Saint Venant beam bending*

Further validation of the proposed strategy is obtained if we consider the problem of a beam bending under a transverse load (assumed vertical, for simplicity) applied at any point of its boundary. The problem has no analytical solution, up to our knowledge, and therefore, the general, multi-dimensional solution has been compared with a reference one obtained by standard finite element models (one for each considered load position) with consistent linearization and a Newton–Raphson iterative scheme.

The mesh is composed of tetrahedral elements, with only 9×9 nodes in the 40×40 mm^2 cross section and 21 nodes in the longitudinal direction, 400 mm long. Material parameters were Young's module $E = 1.0$ MPa and Poisson's coefficient $\nu = 0.25$. With such a poor discretization, and employing tetrahedral elements, it is expected that a high error with respect to the exact solution will be obtained. However, it is not the purpose of this paper to obtain an accurate enough solution of the problem (which of course could be obtained by employing a more refined mesh and a finer time stepping).

The finite element model is shown in Figure 2, and the load is assumed to be applied at any of the points of the upper surface of the beam.

The obtained displacement for a particular location of the load (beam tip in this case, for comparison purposes) is depicted in Figure 3. Noteworthy, the simple explicit linearization algorithm here proposed does not imply a nonphysical gain of volume in the deformed model, which is the case for purely linear elastic models, nevertheless frequently employed in real-time simulation of surgery [2].

Standard forward Euler algorithms for the linearization of the weak form of the problem rendered, in our experiments, spurious deformation modes. Although they are intrinsic of the Kirchhoff–Saint Venant constitutive model, it continues to be popular among the virtual surgery community because it constitutes the fastest way to avoid spurious gain in the volume typical of linear elastic models, when large deformations are being considered [2]. An example of the result given by a forward Euler scheme for this same problem is shown in Figure 4. Although no unphysical gain in volume—typical of linear elastic approaches to the problem—is seen, the obtained displacement at beam tip is much higher than the reference one, obtained by FEMs. This is due to some well-known instabilities of the Kirchhoff–Saint Venant model under compression.

In general, the number n of functions employed in the approximation depends on the desired level of accuracy. Increasing the number of separated functions in the approximation leads, of course, to higher computational costs in the offline part of the method. But we highlight that this computation is carried out only once for life and stored in memory. The online part of the simulation is virtually

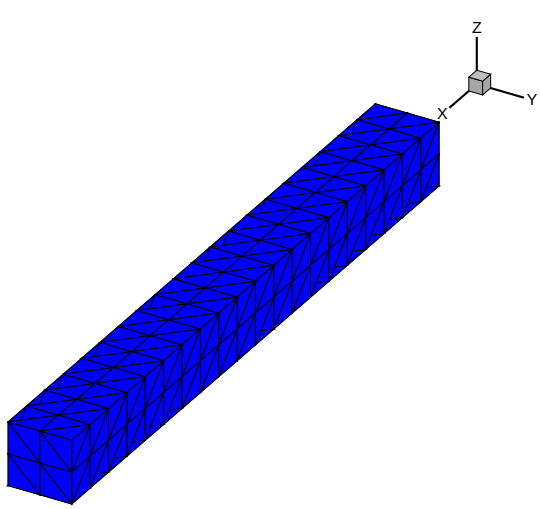

Figure 2. Model for the beam bending problem.

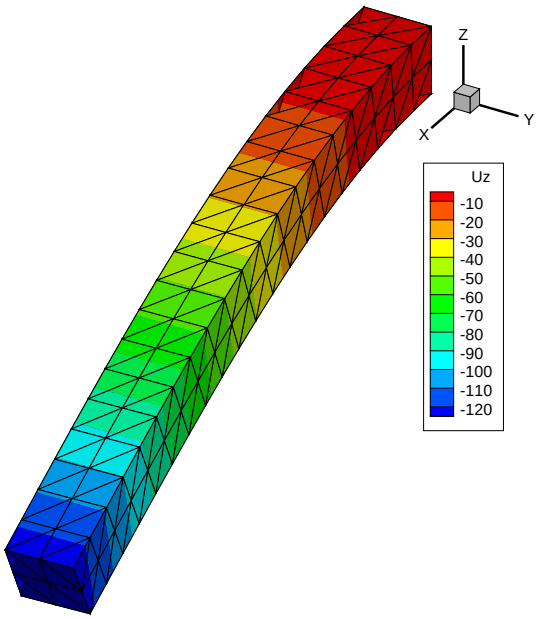

Figure 3. Deformed beam for a particular location of the point load. Note that no unphysical gain in volume is observed.

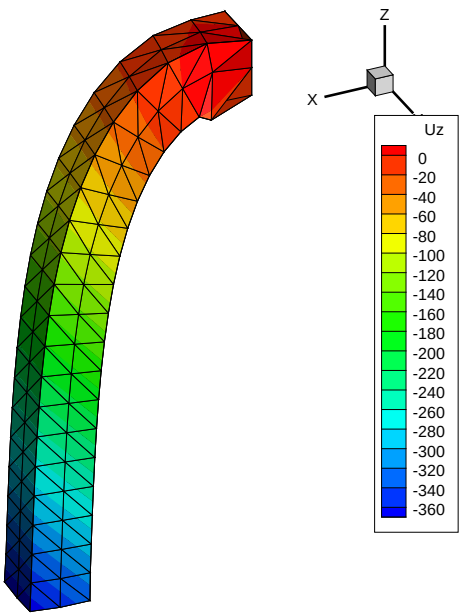

Figure 4. Spurious deformation obtained by employing standard forward Euler schemes. Note the instabilities near the beam clamping due to compressive stresses.

not affected, because only some vector multiplications should be performed in addition, which do not alter the overall efficiency of the proposed method.

4.3. Palpation of the liver

The liver is the biggest gland in the human body, after the skin. Liver geometry has been obtained from the SOFA project [34] and post-processed to obtain a mesh composed of 8559 nodes and 10,519 tetrahedra (Figure 5). The liver is connected to the diaphragm by the coronary ligament so

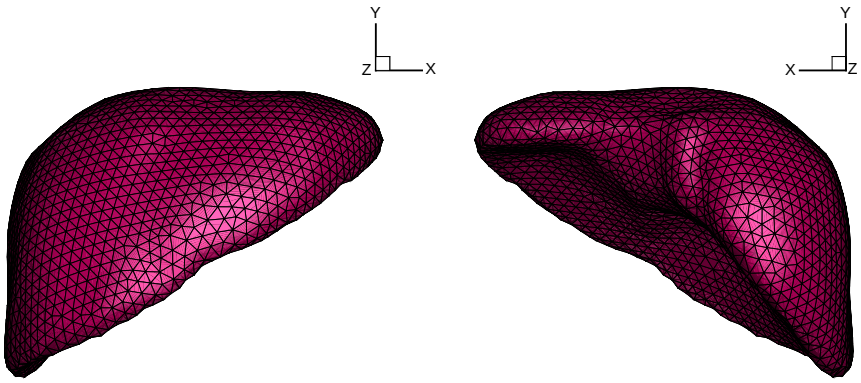

Figure 5. Finite element model for the human liver.

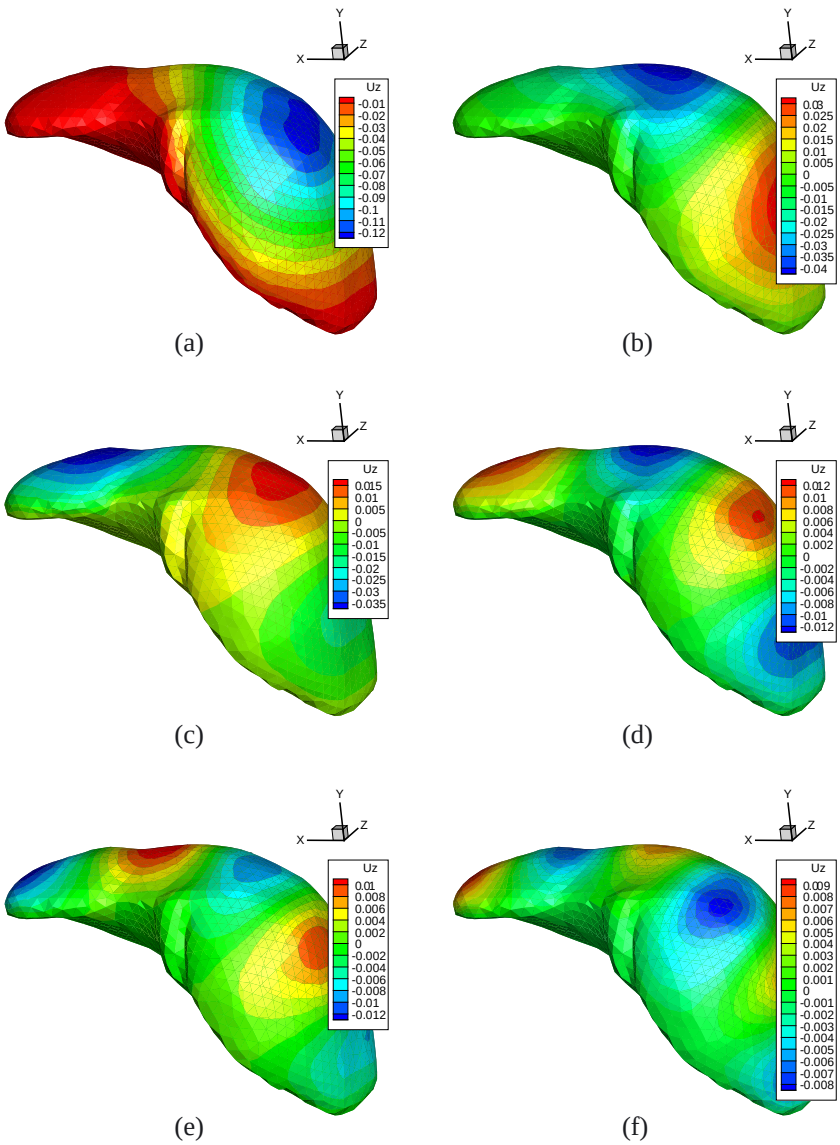

Figure 6. Six first functions $X_3^k(\boldsymbol{x})$, $k = 1, \ldots 6$, for the simulation of the liver.

it seems reasonable to assume it to be constrained at the posterior face by the rest of the organs, whereas the anterior face is accessible to the surgeon. 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 [2].

The $\bar{\Gamma}$ surface, where the load can be located, has been defined as the whole boundary of the domain, even if in this case, only the frontal part of the organ is usually accessible to the surgeon. This region includes 2009 of the 8559 nodes of the model.

Model's solution was composed by a total of $n = 167$ functional pairs $X_j^k(\boldsymbol{x}) \cdot Y_j^k(\boldsymbol{s})$ (see Equation (3)). The third component (thus $j = 3$) of the first six modes $X_3^k(\boldsymbol{x})$ is depicted in Figure 6.

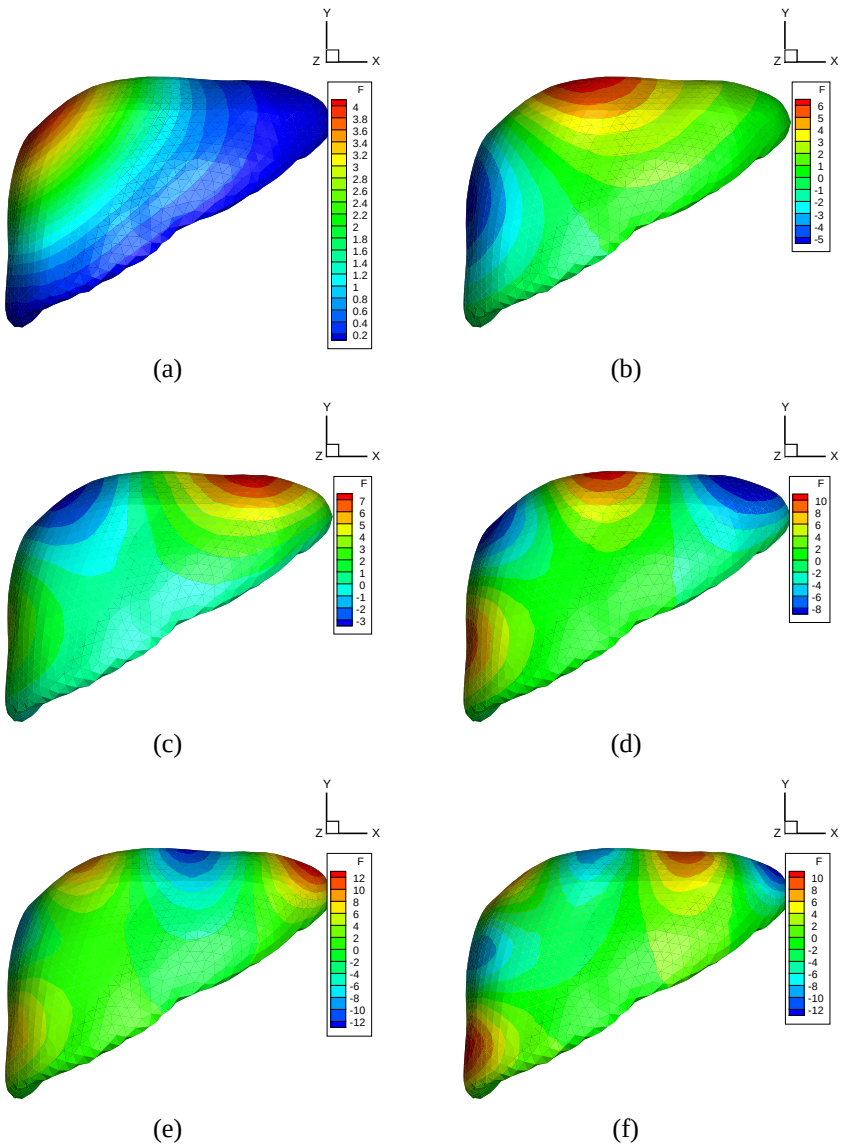

Figure 7. Six first functions $Y_3^k(\boldsymbol{s})$, $k = 1, \ldots 6$, for the simulation of the liver. Note that, in this case, functions $\boldsymbol{Y}^k(\boldsymbol{s})$ are defined on the boundary of the liver only.

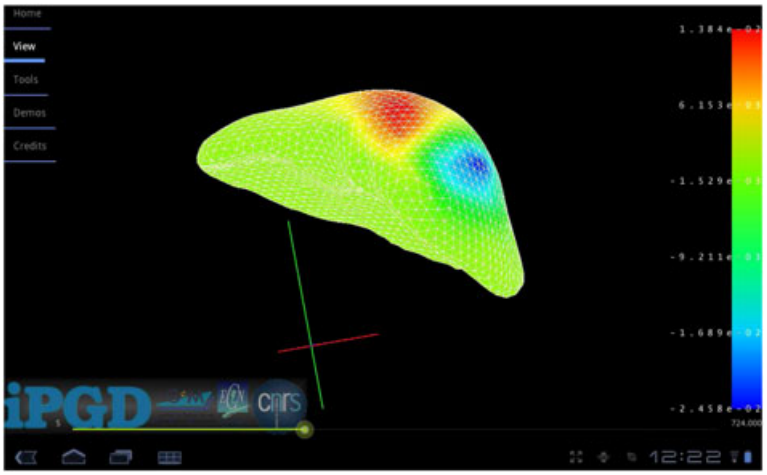

Figure 8. An example of the implementation of the iPGD application for the liver problem.

The same is carried out in Figure 7 for functions $\boldsymbol{Y}$, although in this case, they are defined only on the boundary of the domain, that is, $\bar{\Gamma} = \partial\Omega$.

It is noteworthy that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ sets of functions present a structure similar to that generated by POD methods, despite the fact that they are not, in general, optimal. Note how the frequency content of each pair of functions increases as we increase the number of the function, k.

The solution provided by the method agrees well with reference finite element solutions obtained by employing full-Newton-Raphson iterative schemes (following the same tendency than that shown for the beam bending problem). But, notably, the computed solution can be stored in a so compact form that an implementation of the method is possible on handheld devices such as smartphones and tablets. For instance, for Android-operated devices, an application has been developed (we call it iPGD and is freely downloadable from [35]) that runs the model on a Motorola Xoom tablet running Android 3.0 without problems (only the surface of the model is represented for simplicity, given the limitations of the Android OS) (Figure 8). The 25-Hz feedback rate necessary for continuous visual perception is achieved without problems.

For more sophisticated requirements, such as those dictated by haptic peripherals, a simple laptop (in our case a MacBook pro running MAC OSX 10.7.4, equipped with 4-Gb RAM and an Intel core i7 processor at 2.66 GHz) is enough to achieve this performance. Feedback rates in the order of kilohertz are obtained without problems.

5. CONCLUSIONS

Model order reduction seems to play an important role in real-time simulation of soft biological tissues. In the last times, there have been a number of publications on the use of POD techniques to this class of problems. However, POD-based approaches seem to over-simplify models, and approaches other than those used in [14] do not reproduce properly the nonlinearity of soft tissues. In this paper, a new approach to the problem has been introduced. It is based on the use of PGD methods. This implies a complete change of paradigm, because PGD (in contrast to POD) do not need prior computer experiments to generate the snapshots needed to construct the optimal basis functions. The reduced approximation bases, in a separated form, are constructed on the fly.

PGD, on the contrary, operate in a two-stage approach. Firstly, a general meta-model is computed a priori once for life. During this intensive computation phase, the solution to the high-dimensional model is computed as a finite sum of separable functions. This compact solution, although not optimal, in general, provides a very light format to store the solution in the form of a meta-model that provides the solution to the problem for any parameter value. In this particular problem,

parameters are chosen as the position of the contact force between organ and surgical tool (scalpel) and orientation of the load (thus rendering a problem defined in $\mathbb{R}^9$).

This meta-model is then evaluated under real-time restrictions very efficiently (e.g., reaching more than 1 kHZ in a MacBook pro laptop). In this paper, some benchmark examples have been given to justify the accuracy of the proposed approach. In particular, two different explicit algorithms for the time integration of the resulting equations have been proposed. These algorithms have shown to work well, although the development of more robust strategies of linearization of the weak form of the problem, based on the use of asymptotic numerical methods, is now being sought. This is part of our current effort in this research.

ACKNOWLEDGEMENTS

This study was supported by the Spanish Ministry of Economy and Innovation (contract/grant number: CICYT DPI2011-27778-C02-01) and the European Union, Seventh Framework Program (Vision Advanced Infrastructure for Research (VISIONAIR) project, contract/grant number: EU-FP7-STREP-262044).

REFERENCES

1. Delingette H, Ayache N. Hepatic surgery simulation. *Communications of the ACM* 2005; **48**:31–36.
2. Delingette H, Ayache N. Soft tissue modeling for surgery simulation. In *Computational Models for the Human Body*, Ayache N (ed.), Handbook of Numerical Analysis (Ph. Ciarlet, Ed.) Elsevier: Amsterdam: North-Holland; New York, 2004; 453–550.
3. Fung YC. *Biomechanics: Mechanical Propeties of Living Tissues*. Springer-Verlag: New York, 1993.
4. Alastrue V, Calvo B, Pena E, Doblare M. Biomechanical modeling of refractive corneal surgery. *Journal of Biomechanical Engineering-Trasactions of the ASME* 2006; **128**:150–160.
5. Holzapfel GA, Gasser TC, Ogden RW. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity* 2000; **61**:1–48.
6. Taylor ZA, Comas O, Cheng M, Passenger J, Hawkes DJ, Atkinson D, Ourselin S. On modelling of anisotropic viscoelasticity for soft tissue simulation: numerical solution and GPU execution. *Medical Image Analysis* 2009; **13**(2):234–244. Includes Special Section on Functional Imaging and Modelling of the Heart.
7. Taylor ZA, Cheng M, Ourselin S. High-speed nonlinear finite element analysis for surgical simulation using graphics processing units. *IEEE Transactions on Medical Imaging* 2008; **27**(5):650–663.
8. Lim Y-J, De S. 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* 2007; **196**:3011–3024.
9. Barbič J, James DL. Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics (SIGGRAPH 2005)* 2005; **24**(3):982–990.
10. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer Methods and Programs in Biomedicine* 2008; **91**(3):223–231.
11. Niroomandi S, Alfaro I, Gonzalez D, Cueto E, Chinesta F. Real-time simulation of surgery by reduced-order modeling and X-FEM techniques. *International Journal for Numerical Methods in Biomedical Engineering* 2012; **28**(5):574–588.
12. Taylor ZA, Crozier S, Ourselin S. A reduced order explicit dynamic finite element algorithm for surgical simulation. *IEEE Transactions on Medical Imaging* 2011; **30**(9):1713–1721.
13. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering* 2010; **81**(9):1180–1206.
14. Niroomandi S, Alfaro I, Cueto E, Chinesta F. Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models. *Computer Methods and Programs in Biomedicine* 2012; **105**(1):1–12.
15. Cochelin B, Damil N, Potier-Ferry M. Asymptotic-numerical methods and Padé approximants for non-linear elastic structures. *International Journal for Numerical Methods in Engineering* 1994; **37**:1187–1213.
16. Abichou H, Zahrouni H, Potier-Ferry M. Asymptotic numerical method for problems coupling several nonlinearities. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**(51–52):5795–5810.
17. Cao H-L, Potier-Ferry M. An improved iterative method for large strain viscoplastic problems. *International Journal for Numerical Methods in Engineering* 1999; **44**:155–176.
18. Cochelin B, Damil N, Potier-Ferry M. The asymptotic numerical method: an efficient perturbation technique for nonlinear structural mechanics. *Revue Europeenne des Elements Finis* 1994; **3**:281–297.
19. Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics* 2006; **139**:153–176.
20. Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics* 2007; **144**:98–121.

21. Chinesta F, Ammar A, Cueto E. Recent advances in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering* 2010; **17**(4):327–350.
22. Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering* 2011; **18**:395–404.
23. Ladeveze P, Passieux J-C, Neron D. The LATIN multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering* 2010; **199**(21–22):1287–1296.
24. Ladeveze P. *Nonlinear Computational Structural Mechanics*. Springer: N.Y., 1999.
25. Ghnatios, Ch, Chinesta F, Cueto E, Leygue A, Poitou A, Breitkopf P, Villon P. Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion. *Composites Part A: Applied Science and Manufacturing* 2011; **42**(9):1169–1178.
26. Ghnatios, Ch, Masson F, Huerta A, Leygue A, Cueto E, Chinesta F. Proper generalized decomposition based dynamic data-driven control of thermal processes. *Computer Methods in Applied Mechanics and Engineering* 2012; **213–216**(0):29–41.
27. Bognet B, Bordeu F, Chinesta F, Leygue A, Poitou A. Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity. *Computer Methods in Applied Mechanics and Engineering* 2012; **201–204**(0):1–12.
28. Ammar A, Cueto E, Chinesta F. Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions. *International Journal for Numerical Methods in Biomedical Engineering* 2012; **28**(9):960–973.
29. Pruliere E, Chinesta F, Ammar A. On the deterministic solution of multidimensional parametric models using the proper generalized decomposition. *Mathematics and Computers in Simulation* 2010; **81**(4):791–810.
30. Gonzalez D, Masson F, Poulhaon F, Cueto E, Chinesta F. Proper generalized decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation* 2012; **82**:1677–1695.
31. Bro-Nielsen M, Cotin S. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum* 1996; **15**(3):57–66.
32. Taylor ZA, Ourselin S, Crozier S. A reduced order finite element algorithm for surgical simulation. *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, Buenos Aires, Argentina, August 31–September 4, 2010; 239–242.
33. Bonet J, Kulasegaram S. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering* 2000; **47**:1189–1214.
34. Allard J, Cotin S, Faure F, Bensoussan P-J, Poyer F, Duriez C, Delingette H, Grisoni L. SOFA an open source framework for medical simulation. *Medicine Meets Virtual Reality (MMVR'15)*, Long Beach, USA, February 2007; 13–18.
35. Bordeu F, Leygue A, Alfaro I, González D, Modesto D, Cueto E, Huerta A, Chinesta F. iPGD, an interactive PGD application for Android, 2012. (Available from: http://centrale-nantes-composites.com [Accessed date: February 16 2012]).