

Real-time simulation techniques for augmented learning in science and engineering

C. Quesada · D. González · I. Alfaro · E. Cueto · A. Huerta · F. Chinesta

Received: date / Accepted: date

Abstract In this paper we present the basics of a novel methodology for the development of simulation-based and augmented learning tools in the context of applied science and engineering. It is based on the extensive use of model order reduction, and particularly, of the so-called Proper Generalized Decomposition (PGD) method. This method provides a sort of meta-modeling tool without the need for prior computer experiments that allows the user to obtain real-time response in the solution of complex engineering or physical problems. This real-time capability also allows for its implementation in deployed, touch-screen, handheld devices or even to be immersed into electronic textbooks. We explore here the basics of the proposed methodology and give examples on a few challenging applications never until now explored, up to our knowledge.

This work has been partially supported by the IUF (*Institut Universitaire de France*) and by the Spanish Ministry of Economy and Competitiveness, through grants number DPI2011-27778-C02-01/02 and DPI2014-51844-C2-1/2-R.

C. Quesada, D. González, I. Alfaro, E. Cueto
Aragón Institute of Engineering Research (I3A)
Universidad de Zaragoza. Maria de Luna, 3. E-50018
Zaragoza, Spain
E-mail: ecueto@unizar.es

A. Huerta
Laboratori de Calcul Numeric (LaCaN)
Dep. Matematica Aplicada III
Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034
Barcelona, Spain

Zienkiewicz Centre for Computational Engineering, College
of Engineering,
Swansea University, Swansea SA2 8PP, UK

F. Chinesta
GEM, UMR CNRS - Centrale Nantes
Institut Universitaire de France
1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France

Keywords Augmented learning · real-time simulation · model order reduction · Proper Generalized Decomposition

1 Introduction

Learning in a discovery environment is well-known to be different from learning in a more expository context [67]. Recently, an experiment by MIT showed that Ethiopian children were able to learn to read by themselves with the aid of tablets specially equipped [68]. Specially in higher education contexts, and particularly in engineering and applied sciences studies, it has been recently claimed that simulation-based learning enhances not only self-determined motivation, but also improves learning in general [45]. One step further, *augmented learning* has emerged as a new way to improve the learning process by adapting the environment to the learner. This is achieved, notably, with the use, among other devices, of touchscreens or haptic peripherals [42, 65, 31, 66, 44, 46]. Some authors claim that these results can be obtained even on a non-conscious manner [64].

While simulations allow science educators to reproduce phenomena otherwise impossible to reproduce experimentally at the classroom, handheld devices permit to benefit from their portability, interactivity and sociability, see [43] [60]. Since the appearance of interactive books, also the editorial world is suffering changes in the way we look at higher education bibliography and supporting material.

When talking about higher education, however, this task becomes increasingly hard. Simulation of real engineering problems, sometimes even for, in principle, simple physical equations becomes a task of hours or is merely unaffordable. This is the case, for instance,

of the Schroedinger equation. As Nobel-prize winner R. B. Laughlin once stated about this equation, “no computer existing, or that will ever exist, can break this barrier because it is a catastrophe of dimension” [50].

The task of implementing augmented learning strategies for complex engineering or physical problems is therefore a formidable challenge. These problems take days of supercomputing facilities to be solved. The simple possibility of solving them on deployed, handheld platforms seems to be out of reach. In addition, traditionally, simulation-based engineering sciences make use of static data. By *static* we mean here that inputs can not be changed dynamically during the course of the simulation. But in augmented learning environments we face a clear example of Dynamic Data Driven Application Systems (DDDAS). Nowadays, the linkage of simulation tools with external dynamic data for real-time control of simulations and applications is becoming more and more frequent. In the educational context, the external dynamic data is given by the learner’s interaction instead of measurement devices. DDDAS constitute one of the most challenging applications of simulation-based engineering sciences, due to the dynamic incorporation of additional data into an executing application, and *vice versa*, the ability of an application to dynamically steer the interaction process [11].

One of the most challenging applications in the field of DDDAS (also in the field of augmented learning) is that of *haptic* systems. By haptic we refer to those peripherals in an informatics system that provide the user with tactile sensations. Haptic devices need a feedback rate that is 20 to 40 times faster than visual ones [29], and instead of sending a predefined image may require some computations.

Haptic peripherals are particularly extended in the field of virtual surgery training, for instance, surgery planning and also augmented reality. Models for living soft tissues in virtual surgery are usually highly non-linear [32], hyperelastic, possibly incorporating the presence of collagen fibre networks. Again, the challenge is formidable and constitutes what is commonly known as the *third generation* of surgery simulators [29].

In this paper we analyze how a simulation technique recently proposed by the authors can help in the development of augmented learning strategies for applied sciences and engineering higher education. The novel technique is called Proper Generalized Decomposition (hereafter, PGD) [24] and constitutes a generalization of Proper Orthogonal Decomposition, which is a classical approach in reduced order modeling [53, 41, 52, 63, 9, 10, 20, 14, 54]. Unlike many other previous model order reduction techniques, PGD does not need for prior computer experiments (the so-called *snapshots*) that,

after a statistical treatment, are used as a basis for subsequent simulations. In classical POD, these simulations need not to differ greatly from the existing samples or snapshots. Instead, PGD provides a practical means of obtaining meta-models for complex scenarios. These meta-models or computational vademecums [26] are then at real-time feedback rates.

Although other model order reduction techniques have been successfully applied to real-time simulation (notably the work by Barbič and James, see [13] [12]), the main difference of the approach here presented is that it allows for the reduction of parametric models. These parametric models give rise to high dimensional problems, whose solution is far from being trivial and suffers the so-called *curse of dimensionality*. But, on the contrary, once solved, allow the user to play with those parameters and see in real time their effect on the result of the simulation. It will be highlighted how PGD techniques can help in developing suitable strategies for solving complex, state-of-the-art engineering and physics problems at real time rates even on a smartphone or tablet and how these simulations can interact with user inputs seamlessly.

2 Proper Generalized Decomposition: a brief summary

The origin of the PGD technique can be seen as arising from a variety of sources. On one hand, the work by Pierre Ladevèze on the so-called LARge Time INcrement (LATIN) method [47] can be seen as a first attempt to solve engineering problems in a non-incremental way. That is, problems are solved for any time at once. This technique proposed a space-time separated representation of the results. In fact, it considered time as an additional coordinate very much like the spatial ones.

On the other hand, Proper Orthogonal Decomposition techniques [41, 52, 53] proposes the obtention of the *best* functions to approximate a problem after a statistical treatment of some previously obtained results, the so-called *snapshots*. These best functions when multiplied by time functions, give a practical and fast approximation to the solution of problems which are slightly different to the original snapshots. Proper Orthogonal Decomposition techniques are also known as Principal Component Analysis or Karhunen-Loeve transform among many different names in different disciplines.

On a completely different setting, many problems in science and engineering are defined in high-dimensional spaces (a notable example is that of parametric problems, if we consider parameters as new coordinates). High-dimensional problems induce forceful difficulties in a context of mesh-based techniques such as finite

differences, finite volumes or finite elements. This is due to the so-called *curse of dimensionality* [50], an exponential increase of the number of degrees of freedom of the problem with the number of state space dimensions. Such high-dimensional problems are very common in science and engineering. Examples of such problems are: the afore-mentioned Schroedinger equation [19], the chemical master equation [38] governing gene regulatory networks or Fokker-Planck equations in the context of the kinetic theory descriptions of complex fluids [37] [16].

In this spirit, F. Chinesta and coworkers proposed recently a technique to overcome this curse of dimensionality in the polymer modeling problem [7,8]. This technique, which is at the inception of the PGD method, assumes a separated form for the approximation of the unknown field of the problem, say u , thus generalizing the method of separation of variables:

$$u(x_1, \dots, x_N) \approx \sum_{i=1}^Q F_i^1(x_1) \cdot \dots \cdot F_i^N(x_N) \quad (1)$$

where x_i denotes a scalar or vector coordinate defined in a domain Ω_i of moderate dimension, namely $\Omega_i \subset \mathbb{R}^d$, with $d \leq 3$, in general. This approximation is then substituted into the set of equations characterizing the problem at hand. Usually, in an engineering or applied science context, this is a weak form of the problem, or directly the strong form of the equation, as in the gene regulatory network problem, see [6].

This simple approach overcomes the mentioned curse of dimensionality, opening the application of the PGD technique to a wide variety of problems. This technique has received a warm welcome in the scientific community, as proved by the large number of papers devoted to it in the very last years. Just to name a few, it is possible to find PGD applications to thermal problems [33,34], theories of plates and shells [17], gene regulatory networks [6], and structural mechanics [49] among many others [35,22,2,5,3,21,36]. The technique is beginning also to be understood from a mathematical point of view and its properties, convergence and error estimates have also been studied recently [51,4,48,59]. The interested reader can consult some recent surveys on the method [25,23].

But the main interest from the perspective pursued in this paper is that PGD methods opened the door to an innovative treatment of classical problems, not necessarily defined in high dimensions. When dealing with parametric problems, for instance, it has been seen that these parameters can advantageously be considered as new coordinates of the problem. Since the PGD is able to cope with an arbitrary number of dimensions with-

out the typical exponential increase in terms of degrees of freedom, this procedure is now at hand.

By considering parameters as new spatial dimensions, a new class of solutions is obtained for this type of problems. The PGD method is therefore able to obtain solutions for *any* value of the parameters at *any* point of the space, at *any* time instant [61,39]. So to speak, PGD provides the analyst with a sort of meta-model, but with the strong novelty of no need for previous computer experiments. Many problems in science and engineering can thus be analyzed once and for all, since the method provides the solution for any value (in a given interval) of the parameters. In Appendix A the interested reader can find the details of the PGD procedure for a solid mechanics problem in which the position of the applied load is taken as a parameter.

This simple approach has been implemented in a two-stage approach. The first stage is concerned with the *off-line* computations in which the general form of the solution is computed, and as mentioned before once and for all. This phase may take intensive computing resources, and may need for supercomputing resources, parallel computing, GPUs, ... Note, however, that in general the computer cost at this stage is not critical since it will be done only once. Then, with the results obtained in the first off-line stage, an *on-line* phase can be accomplished. At this on-line stage the user, namely the learner in an educational context, can make use of the results in real-time, even on deployed, handheld devices, thanks to the simplified structure of the approximation.

It is important to mention that the structure of the results file in which the multidimensional solution is stored (represented in a finite element context) is a collection of nodal values for all the separated functions composing the approximation. It is therefore very efficient. When a parameter is particularized, what we are doing in practice is a hyperplane cut of the multidimensional solution. This is done just by performing interpolation at the functions related to the parameter (dimension) that is being considered.

Should the reader be interested in the development of his or her own PGD applications, it may be interested in consulting the sample Matlab code that can be downloaded from the author's web page at http://amb.unizar.es/PGD_Basic_web.rar.

3 Towards simulation-based and augmented learning for complex physics

Based on the two-stage strategy presented before, one off-line intensive computing phase followed by a fast

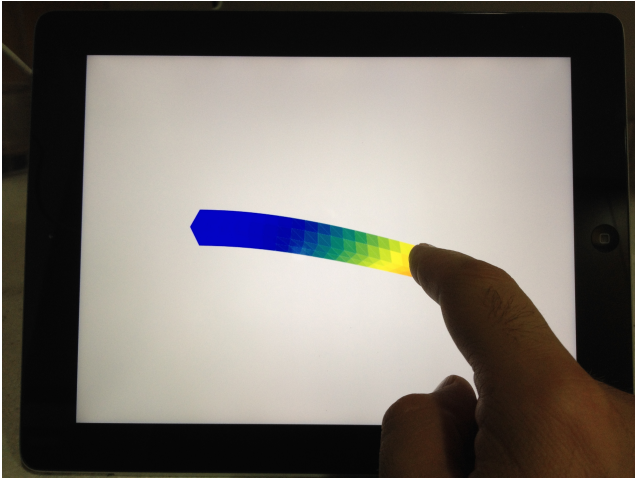


Fig. 1 Implementation on a `html` file with javascript. It represents the response of a cantilevered hyperelastic beam. The user applies load by touching the screen.

real-time on-line simulation phase, the authors have developed and implemented a PGD platform able to run on smart-phones and tablets running Android or iOS operative systems, but also standard `epub` or `html` formats.

In our implementation, the different F_i^j are usually stored as one-dimensional arrays of values, thus minimizing the needed memory usage. Note that the Android 3.0 operative system, for instance, precludes the use of more than 65536 points (nodes) in the model. This is avoided by using meshes of the boundary of three-dimensional solids only, instead of a fully 3D-mesh. In Fig. 1 a screenshot is shown on the appearance of the application developed by using `html` and javascript. Moreover, lighter versions have been implemented on Nokia or iPhone smartphones with similar outputs, see Fig. 2. When running on a handheld device such as a smartphone or tablet, the application is able to provide the user with feedback rates enough for visual realism (on the order of 30 Hz). If haptic feedback rates are needed (i.e., on the order of 500 Hz), the application must be run on a laptop (tested on a MacBook pro running OS X Lion).

Features of the open `epub` format also allow to implement real-time simulation apps on iPad and iPhone devices by noting that the `epub` format is no more than plane `html` text with, possibly, javascript embedded on it. This opens a plethora of possibilities to include PGD simulations on handheld devices, see Fig. 3. All these examples have been uploaded to our YouTube channel for visualization and permanent storage, see <https://www.youtube.com/channel/UCbh8NjqDDiTOKmN1sGR0ipg>.

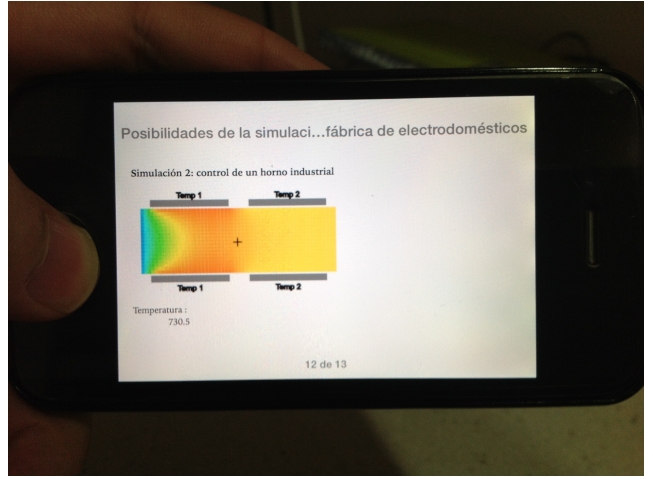


Fig. 2 Implementation of thermal interactive problems on an iPhone. In this case an industrial furnace thermal simulation is running. The learner can manually adjust the working temperature on the heaters to observe the effect on the resulting temperature field. The source code for this example is included in Appendix B.

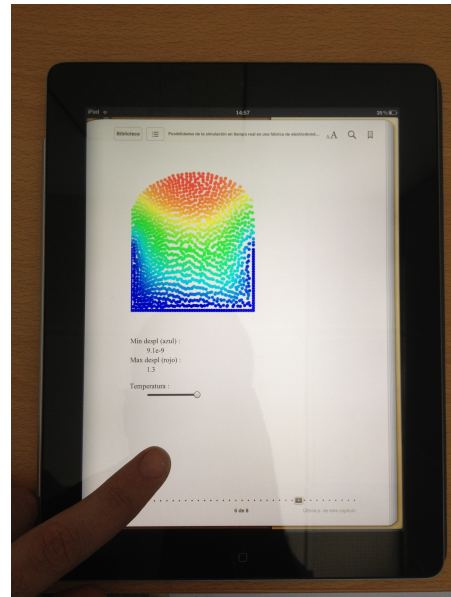


Fig. 3 The iBook app in iOS devices (iPhones and iPads) also allows to easily implement real-time simulations on `epub` documents.

4 Examples of the performance of the proposed technique

In this section several examples on how the PGD application is being used advantageously are given. In all cases we deal with complex physics, difficult to understand for undergraduate students of several disciplines. The PGD application provides them with different skills ranging from a physical interpretation of complex-valued equations, as is the case for the Helmholtz equation governing the wave agitation in a harbor, to

anatomy and (possibly) soft tissue stiffness for biomedical applications in the field of endoscopic surgery.

4.1 Wave height on a harbor

The knowledge of sea-wave propagation is crucial for many stages of harbor design. For instance, optimal shape is required to minimize the wave agitation in some zones of interest, usually those areas where more shipping activities are concentrated. Understanding how the incident wave can amplify inside the harbor is a major priority, for example when resonance conditions are of concern [15]. Furthermore, the wave behavior while propagating under complex harbor models is not an easy task to be interpreted by engineers, see Fig 4. The standard model which governs the wave propagation in harbors is the two-dimensional Helmholtz equation in unbounded domains. Its solution provides the wave height, or wave amplification, when dealing with a single incident wave in terms of period (or wavelength) and incident direction of propagation.

But, the physical wave response on harbors extremely depends on these two model parameters. A complete wave agitation analysis requires the solution of the previous problem for any possible combination of incident wave periods and directions, determined by experimental measurements in the coastal area. Typical applications are control, optimization or inverse problems, where usually a large number of solutions are required, and, consequently, it is necessary to solve many direct problems. The solution of real harbor models demands important computational resources and usually implies extremely large computing times. Thus, usual optimization procedures are inapplicable, either because they need numerous solutions or because real-time constraints are required. To take charge of this issue, standard engineering procedure lies in solving the model with a certain set of parameters trying to cover the maximum number of combinations. Clearly, this “brute force” strategy impose a bound in the final number of cases to be analyzed and, therefore, important loss of information become evident.

The presented PGD approach for this kind of problem involves, in separated form, the incident wave period and the incident direction as new coordinates in the model. With this strategy, the full 4D problem is reduced to the iterative solution of one 2D problem and two 1D problems. Hence, the solution of this model breaks the barrier that has prevented the full incident data evaluation up to nowadays. By means of readily evaluating, offline, any incident wave, engineers can obtain the wave propagation inside a harbor in a real-time

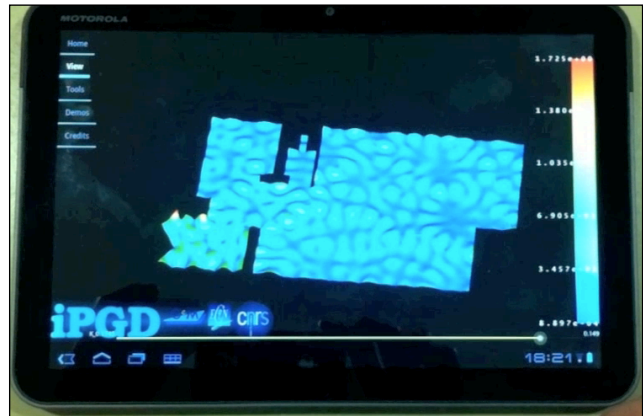


Fig. 5 Example of the use of PGD for the wave propagation in the Mataró harbor with any incident period between 5 and 14 seconds.

manner, giving them a formidable support to better analyze where the wave amplifies for all the possible real situations.

At the same time, the solution given by the PGD approach in the Mataró harbor is implemented on the PGD application for Android platforms, see Fig. 5. In this case, students can dynamically change the incident period and obtain the wave amplification, in real-time, inside the area of interest, with no need of high computational resources.

4.2 The case of anatomy and endoscopic training

Dissection is recognized as the best practice to learn gross anatomy. However, the price of this class of experimental training in medicine universities becomes prohibitive in most cases, due to problems arising from logistic and financial issues. Teaching anatomy with the help of multimedia and computers is nowadays an extended practice, see [40], for instance. Some recent specialities, such as endoscopic (laparoscopic) and minimally invasive surgery, for instance, need for an even more sophisticated training.

In endoscopic training, surgeons have been trained for years with the help of animal models, i.e., by experimenting on pigs, for instance [30]. Apart from being very expensive, this type of training does not fit with the modern ethical treatises about experimentation on animals. Instead, simulations provide the learners with an immersive environment in which it is possible a versatile change of scenario where even extremely rare cases can be studied.

The development of surgery simulators with haptic feedback is an active area of research due to the complexity of the problem. One of the sources of complexity is due to the highly non-linear behaviour of soft living

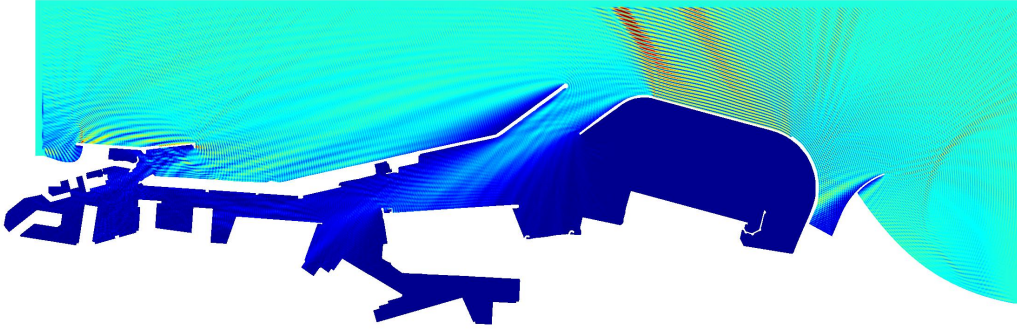


Fig. 4 Example of simulation of the wave propagation in the Barcelona harbor.

tissues, that are frequently modelled under the fiber-reinforced hyperelasticity framework [1]. Other source of complexity comes from the highly restrictive feedback rates imposed by the simulators (25 Hz for visual feedback and some 500 Hz if we want to add haptic feedback to the system). The third source of complexity comes from the multi-physic nature of the phenomena occurring in the actual surgery procedure: non-linear elasticity, contact, cutting, temperature, etc.

Such simulators should provide a physically more or less accurate response such that, with the use of haptic devices, a realistic feedback is transmitted to the surgeon in terms of both visual feedback and force feedback. By “accurate response” we mean that an advanced user should not encounter “unphysical” sensations when handling the simulator. We definitely do not pursue an accurate solution in engineering terms. Following [18], “... the model may be physically correct if it looks right”.

Ayache and co-workers [30] defined the three generations of surgical simulators as those able to, respectively, reproduce accurately the anatomy (geometry), the physics (constitutive equations of soft tissue, temperature) and, finally, physiology (blood flow, breathing, ...). Undoubtedly, no third-generation simulation has ever been developed, and only some rigorous attempts have been made at the second-generation level, see [69] [28], among others. Most of the existing simulators can be classified into the first-generation category, even if they provide haptic feedback, because they only consider linear elastic response, for instance [70] [18] [27]. Other are based upon spring and mass systems, that do not even reproduce the equations of linear elasticity (see [29] and references therein). These approaches are judged clearly insufficient and clearly non realistic by most surgeons [29].

The proposed method can be used at two different levels. Firstly, if one considers a medicine student, freshman or sophomore, the PGD application introduced

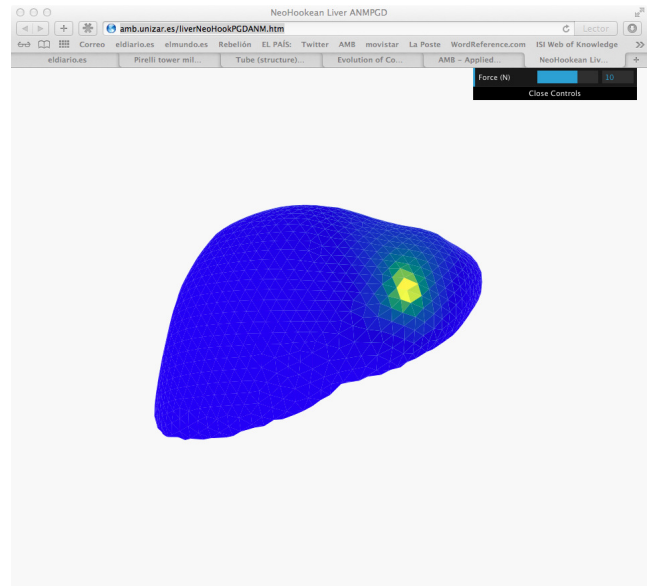


Fig. 6 An example of the use of the PGD approach in anatomy. In this case a human liver is being studied under palpation forces. The simulation runs online via a web-page (<http://amb.unizar.es/PGD.html>) stored in a university repository.

before offers the possibility of an augmented learning strategy in which not only the geometry (anatomy) of the human body can efficiently be displayed, but also the learner can dynamically interact with the tissues and their relative stiffness by tapping on the screen of the handheld device (even if no haptic experience can be obtained at this level). In this way an improved experience over traditional methods [40] can be obtained. The lightness of this approach allows even to implement interactive simulations over the web, thus making possible to store these modules in university repositories, for instance. See Fig. 6 for a practical implementation of a liver palpation simulation on the web [62].

On the other hand, if connected to appropriate haptic devices, see Fig. 7, a standard computer and moni-

tors, an operating theatre can efficiently be simulated. No special computational resources are necessary in this case for the on-line phase of the simulation, and even complex soft tissue models can be efficiently reproduced with haptic feedback and possibly augmented reality scenarios. In any case, the on-line phase of the simulation can incorporate complex, state-of-the-art constitutive laws without affecting the performance of the simulation [55,57,56], since the most part of the computer cost is paid at the off-line phase. The simulator is thus fed with the results of this simulations, possibly very time consuming, but made only once for life.

5 Conclusions

In this paper we have introduced and studied the possibilities offered by proper generalized decomposition methods in the field of simulation-based and augmented learning. The PGD method constitutes a new paradigm in simulation-based engineering sciences, but also possesses salient features in the field of augmented reality and learning. Notably, we have explored its ability to provide for effective solutions (in terms of computing time and also storage needs) for complex physics and engineering problems that have traditionally fall out of reach for traditional simulation techniques at real time feedback rates.

Several examples have been provided that show the potential of the proposed techniques to offer real-time simulation scenarios enabling augmented learning to go beyond its present limits. These examples range from problems of the theory of solid mechanics to anatomy and virtual surgery training of surgeons.

Although it is envisaged that this technique could have a strong impact on the type of problem for which augmented learning strategies are developed, it is still necessary to further study how the approaches here presented could effectively improve the learning process of students at higher education institutions. Since the field of application is very broad, this type of study will probably need a detailed campaign of surveys, different for each speciality. For the time being, this approach is being employed in three different master-level courses at Ecole Centrale Nantes, France (twenty students in the computational materials course, 10 more at the modeling of composite manufacturing processes course and 12 students at the Erasmus Mundus Master program run jointly by EC Nantes, Swansea University, U.K., Stuttgart, Germany, and UPC BarcelonaTech, Spain). This constitutes part of our current effort of research and the results will be published elsewhere.

A Details of the PGD procedure within a parametric problem

To briefly overview the details of the PGD procedure, we consider the case of a solid in whose surface a load is assumed to act at an arbitrary point. Consider, for simplicity, the static equilibrium equations of a general solid under small strain assumptions:

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0} \text{ in } \Omega, \quad (2)$$

where \mathbf{b} represents the volumetric forces applied to the body, subjected to the following boundary conditions

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \Gamma_u \quad (3)$$

$$\boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} \text{ on } \Gamma_t \quad (4)$$

The standard weak form of the problem is obtained after multiplying both sides of Eq. (2) by an admissible variation of the displacement, \mathbf{u}^* , and integrating over the domain Ω . The basic ingredient of PGD is to consider the load $\bar{\mathbf{t}}$ as a parameter of the formulation, thus enabling to obtain a parametric response surface to the problem. If the load is assumed to be applied at point \mathbf{s} of the surface and, for the sake of simplicity in the exposition, unitary and vertical, the problem is now defined in \mathbb{R}^6 , since $\mathbf{u} = \mathbf{u}(\mathbf{x}, \mathbf{s}) \in \Omega \times \bar{\Gamma}$, where $\bar{\Gamma} \subseteq \Gamma_t$ represents the portion of the boundary where the load can be applied.

Since the problem domain is now multi-dimensional, an alternative (doubly) weak form of problem (2)-(4) consists in finding the displacement $\mathbf{u} \in \mathcal{H}^1(\Omega) \times L_2(\bar{\Gamma})$ such that for all $\mathbf{u}^* \in \mathcal{H}_0^1(\Omega) \times L_2(\bar{\Gamma})$ (see [58]):

$$\int_{\bar{\Gamma}} \int_{\Omega} (\nabla_s \mathbf{u}^*)^T \boldsymbol{\sigma} d\Omega d\bar{\Gamma} = \int_{\bar{\Gamma}} \int_{\Gamma_t} (\mathbf{u}^*)^T \mathbf{t} d\Gamma d\bar{\Gamma} \quad (5)$$

where $(\nabla_s \mathbf{u})$ represents the symmetric part of the gradient of displacements, $\Gamma = \Gamma_u \cup \Gamma_t$ represents the boundary of the solid, divided into essential and natural regions, and where $\Gamma_t = \Gamma_{t1} \cup \Gamma_{t2}$, i.e., regions of homogeneous and non-homogeneous, respectively, natural boundary conditions. In turn, we assume $\mathbf{t} = \mathbf{e}_z \delta(\mathbf{x} - \mathbf{s})$, where δ represents the Dirac-delta function and \mathbf{e}_z represents the unit vector along the z direction, in this case. This Dirac-delta term should be regularized for computation purposes and approximated by:

$$t_j \approx \sum_{i=1}^m f_j^i(\mathbf{x}) g_j^i(\mathbf{s}) \quad (6)$$

by simply performing a singular value decomposition of the load, for instance.

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 [24]. Assume that we have converged to a solution, at iteration n , of this procedure,

$$u_j^m(\mathbf{x}, \mathbf{s}) = \sum_{k=1}^m X_j^k(\mathbf{x}) \cdot Y_j^k(\mathbf{s}), \quad (7)$$

where the term u_j refers to the j -th component of the displacement vector, $j = 1, 2, 3$.

The following term of this approximation, the $(n+1)$ -th one, will be given by

$$u_j^{m+1}(\mathbf{x}, \mathbf{s}) = u_j^m(\mathbf{x}, \mathbf{s}) + R_j(\mathbf{x}) \cdot S_j(\mathbf{s}), \quad (8)$$

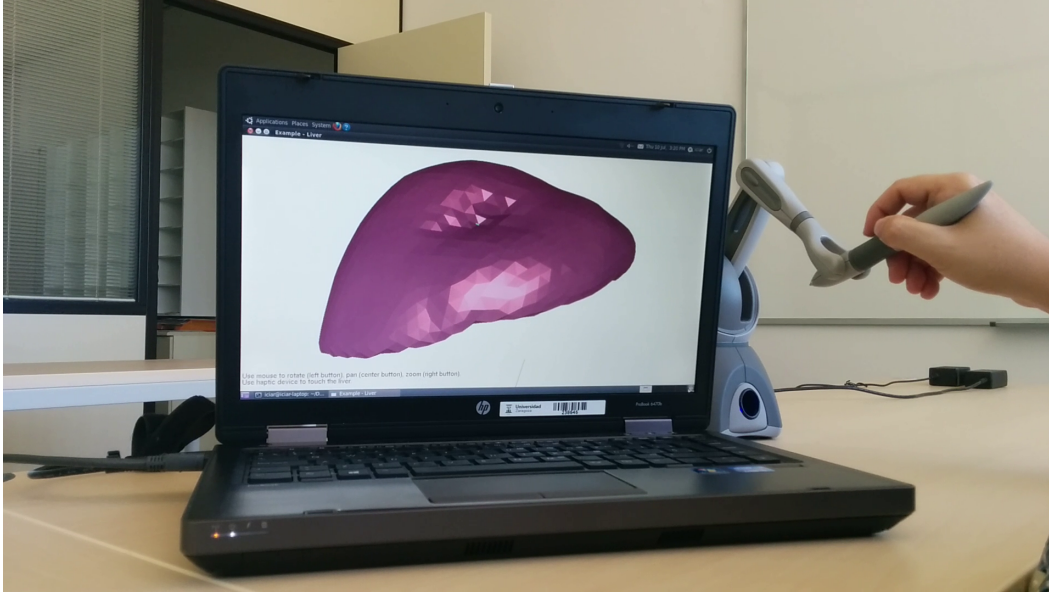


Fig. 7 An second example of the use of the PGD approach in anatomy, this time with haptic response. Palpation during cholecystectomy is studied.

where $\mathbf{R}(\mathbf{x})$ and $\mathbf{S}(\mathbf{s})$ are the sought functions that improve the approximation. Standard variational calculus gives the admissible variation of the displacement as

$$u_j^*(\mathbf{x}, \mathbf{s}) = R_j^*(\mathbf{x}) \cdot S_j(\mathbf{s}) + R_j(\mathbf{x}) \cdot S_j^*(\mathbf{s}). \quad (9)$$

In order to determine the new pair of functions R_j and S_j , in general, a fixed-point alternating directions algorithm, in which functions R_j and S_j are sought iteratively, is chosen.

A.1 Computation of $\mathbf{S}(\mathbf{s})$ assuming $\mathbf{R}(\mathbf{x})$ is known

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

$$u_j^*(\mathbf{x}, \mathbf{s}) = R_j(\mathbf{x}) \cdot S_j^*(\mathbf{s}), \quad (10)$$

or, equivalently, $\mathbf{u}^*(\mathbf{x}, \mathbf{s}) = \mathbf{R} \circ \mathbf{S}^*$, where the symbol “ \circ ” denotes the so-called entry-wise, Hadamard or Schur multiplication for vectors. Once substituted into Eq. (5), gives

$$\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\mathbf{R} \circ \mathbf{S}^*) : \mathbf{C} : \nabla_s \left(\sum_{k=1}^m \mathbf{X}^k \circ \mathbf{Y}^k + \mathbf{R} \circ \mathbf{S} \right) d\Omega d\bar{\Gamma} = \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\mathbf{R} \circ \mathbf{S}^*) \cdot \left(\sum_{k=1}^m \mathbf{f}^k \circ \mathbf{g}^k \right) d\Gamma d\bar{\Gamma}, \quad (11)$$

or, equivalently (we omit obvious functional dependencies)

$$\begin{aligned} & \int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\mathbf{R} \circ \mathbf{S}^*) : \mathbf{C} : \nabla_s(\mathbf{R} \circ \mathbf{S}) d\Omega d\bar{\Gamma} \\ &= \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\mathbf{R} \circ \mathbf{S}^*) \cdot \left(\sum_{k=1}^m \mathbf{f}^k \circ \mathbf{g}^k \right) d\Gamma d\bar{\Gamma} \\ &- \int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\mathbf{R} \circ \mathbf{S}^*) \cdot \mathcal{R}^n d\Omega d\bar{\Gamma}, \end{aligned} \quad (12)$$

where \mathcal{R}^n represents:

$$\mathcal{R}^n = \mathbf{C} : \nabla_s \mathbf{u}^m. \quad (13)$$

All the terms depending on \mathbf{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 $\mathbf{S}(\mathbf{s})$.

A.2 Computation of $\mathbf{R}(\mathbf{x})$ assuming $\mathbf{S}(\mathbf{s})$ is known

Equivalently, in this case, we have

$$u_j^*(\mathbf{x}, \mathbf{s}) = R_j^*(\mathbf{x}) \cdot S_j(\mathbf{s}), \quad (14)$$

which, once substituted into Eq. (5), gives

$$\begin{aligned} & \int_{\bar{\Gamma}} \int_{\Omega} \nabla_s(\mathbf{R}^* \circ \mathbf{S}) : \mathbf{C} : \nabla_s \left(\sum_{k=1}^m \mathbf{X}^k \circ \mathbf{Y}^k + \mathbf{R} \circ \mathbf{S} \right) d\Omega d\bar{\Gamma} = \\ & \int_{\bar{\Gamma}} \int_{\Gamma_{t2}} (\mathbf{R}^* \circ \mathbf{S}) \cdot \left(\sum_{k=1}^m \mathbf{f}^k \circ \mathbf{g}^k \right) d\Gamma d\bar{\Gamma}. \end{aligned} \quad (15)$$

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

B Source code for an epub implementation of the real-time simulator

In this appendix we include, to show the simplicity of the proposed technique, the source code of the html/javascript program to perform a simple thermal simulation. In this case, it corresponds to that in Fig. 2, a simple oven heated by two set of resistances whose temperatures can be adjusted by the user. These temperatures determine the final temperature distribution within the oven.

```

<?xml version="1.0" encoding="utf-8" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Example 2: industrial oven</title>

  <script src="javafiles/data2.js" type="text/javascript">
</script>

  <script src="javafiles/plot2.js" type="text/javascript">
</script>

  <script src="javafiles/plaque2.js" type="text/javascript">
</script>

  <script src="javafiles/ui2.js" type="text/javascript">
</script>

  <link charset="utf-8" href="Styles/main.css" media="screen" rel="stylesheet" type="text/css" ></link>
  <link charset="utf-8" href="Styles/plot.css" media="screen" rel="stylesheet" type="text/css" ></link>
</head>

<body>

<h3>On the possibilities of real-time simulation</h3>

<h1 id="heading_id_2">Simulation 2: control of an industrial oven</h1>

<canvas id="canvas" width="360" height="160"></canvas>

  <dl>

    <dt>Temperature :</dt>

    <dd id="Temp"></dd>

  </dl>

<div style="clear: both;"></div>

<form action="#" id="inputdata" method="">

  <dl>

    <dt><label for="xdim">X dim :</label></dt>

    <dd><input id="xdim" max="1" min="0" name="xdim" step="0.01" type="range" value="0.5" ></input>
    <span class="error" id="resx"></span></dd>

    <dt><label for="ydim">Y dim :</label></dt>

    <dd><input id="ydim" max="1" min="0" name="ydim" step="0.01" type="range" value="0.5" ></input>
    <span class="error" id="resy"></span></dd>

    <dt><label for="Temp1">Temp 1 (100-900):</label></dt>

    <dd><input id="Temp1" max="1" min="0" name="Temp1" step="0.01" type="range" value="0.5" ></input>
    <span class="error" id="resT1"></span></dd>

    <dt><label for="Temp2">Temp 2 (100-900):</label></dt>

    <dd><input id="Temp2" max="1" min="0" name="Temp2" step="0.01" type="range" value="0.5" ></input>
    <span class="error" id="resT2"></span></dd>

  </dl>

</form>

<div style="clear: both;"></div>

```

```

<script charset="utf-8" type="text/javascript">

    window.addEventListener("load", ui.init, false);

</script>

</body>

</html>

```

This piece of code establishes the environment for the html code configuring the **epub** document shown in Fig. 2. It makes a call to several javascript routines, such as **data2.js**, for instance, whose content is reproduced below:

```

var Tnode = {
dim: 1199,
modes: 55,
data: [
[0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.000000000, 0.000000000,
...
0.01760231, 0.01750595, 0.01740960, 0.01731324, 0.01721689]
]
}

```

data2.js contains, in this case, the values, at the nodal points of the mesh represented in Fig. 2, of the separated representation of the parametric solution. In this case, the solution is a function of the position (coordinates x and y , although three-dimensional solutions could be equally represented) and the temperatures at the heaters (thus, a four-dimensional solution). The other essential ingredient is the function **plot2.js**, whose code is reproduced below:

```

var Plot = function(canvasRef, values) {
    this.context = canvasRef.getContext("2d");

    if(values) {
        this.setValues(values);
    }
}

Plot.prototype.context = null;

/* Element size */
var axisx=300/59, axisy=100/19;
/* mesh is 59x19 elements. Mesh and margins positioning*/
Plot.prototype.width = 360;
Plot.prototype.height = 160;
Plot.prototype.origin = { x: 30, y: 130 };
Plot.prototype.axis = { x: axisx, y: -axisy };

Plot.prototype.getRealCoords = function(p) {
    return {
        x: this.origin.x + this.axis.x * p.x,
        y: this.origin.y + this.axis.y * p.y
    };
};

Plot.prototype.clear = function() {
    this.context.clearRect(0, 0, Plot.prototype.width, Plot.prototype.height);
};

Plot.prototype.refresh = function() {
    var i, j, ii, jj, jjj, vals, points = [], color, T;

    this.clear();

    for(i = 0; i < this.dataSource.range.y - 1; i++) {
        for(j = 0; j < this.dataSource.range.x - 1; j++) {
            points = [];
            T = 0;
            for(ii = 0; ii <=1 ; ii++) {
                for(jj = 0; jj <=1 ; jj++) {
                    if(ii == 1) jjj = 1 - jj;
                    else jjj = jj;

```



```

        var node = (j+jjj)+(i+ii)*this.dataSource.range.x;
        T += this.dataSource.ds(node);
        points.push({
            x: (j + jjj),
            y: (i + ii)
        });
    }
}
color = this.colorMap(T / 4);
this.plotPoints(points, color);
}
}

this.context.fillStyle="rgba(0,0,0,0.5)";
this.context.fillRect(Plot.prototype.origin.x+15,Plot.prototype.origin.y-100-10-2,120,10);
this.context.fillRect(Plot.prototype.origin.x+15+120+30,Plot.prototype.origin.y-100-10-2,120,10);
this.context.fillRect(Plot.prototype.origin.x+15,Plot.prototype.origin.y+2,120,10);
this.context.fillRect(Plot.prototype.origin.x+15+120+30,Plot.prototype.origin.y+2,120,10);

this.context.font="12px Arial";
this.context.strokeText("Temp 1",Plot.prototype.origin.x+55,Plot.prototype.origin.y-100-15);
this.context.strokeText("Temp 1",Plot.prototype.origin.x+55,Plot.prototype.origin.y+10+14);
this.context.strokeText("Temp 2",Plot.prototype.origin.x+55+120+30,Plot.prototype.origin.y-100-15);
this.context.strokeText("Temp 2",Plot.prototype.origin.x+55+120+30,Plot.prototype.origin.y+10+14);

var x = parseFloat(document.getElementById("xdim").value, 10),
    y = parseFloat(document.getElementById("ydim").value, 10);
var xOffset = ui.plaque.normalize("x", x, 0, 1),
yOffset = ui.plaque.normalize("y", y, 0, 1);
this.plotMarker({x: xOffset, y: yOffset});

};

Plot.prototype.plotMarker = function(ds) {
    var point = this.getRealCoords(ds);
    this.context.fillStyle="rgb(0,0,0)";
    this.context.beginPath();
    this.context.moveTo(point.x-5, point.y);
    this.context.lineTo(point.x+5, point.y);
    this.context.moveTo(point.x, point.y-5);
    this.context.lineTo(point.x, point.y+5);
    this.context.stroke();
}

Plot.prototype.plotPoints = function(ps, c) {
    var rp = { x: 0, y: 0 };
    this.context.beginPath();
    this.setColor(c);
    for(var i = 0; i < ps.length; i++) {
        rp = this.getRealCoords(ps[i]);
        if(i == 0) {
            this.context.moveTo(rp.x, rp.y);
        } else {
            this.context.lineTo(rp.x, rp.y);
        }
    }
    this.context.fill();
};

Plot.prototype.setColor = function(c) {
    this.context.fillStyle = "rgb("+
        c.r+", "+
        c.g+", "+
        c.b+")";
};

Plot.prototype.setDataSource = function(ds) {
    this.dataSource = ds;
    this.initColorMap();
};

```

```

Plot.prototype.initColorMap = function() {
    // Compute the range
    this.colorMap = colorMap(0, 900); //always the same temperature range, from 0 to 900 deg C
};

var colorMap = function(min, max) {
    var range = max - min;
    if(range == 0) {
        return function(value) {
            return {r: 0, g: 0, b: 255};
        };
    } else {
        return function(value) {
            var normalizedValue = (value - min) / range,
                red,
                green,
                blue;

            if(normalizedValue <= 0.25) {
                red = 0;
                green = 4 * normalizedValue;
                blue = 1;
            } else if(normalizedValue > 0.25 && normalizedValue <= 0.5) {
                red = 0;
                green = 1;
                blue = 1 - 4 * (normalizedValue - 0.25);
            } else if(normalizedValue > 0.5 && normalizedValue <= 0.75) {
                red = 4 * (normalizedValue - 0.5);
                green = 1;
                blue = 0;
            } else {
                red = 1;
                green = 1 - 4 * (normalizedValue - 0.75);
                blue = 0;
            }
            return { r: Math.round(red * 255, 10), g: Math.round(green * 255, 10),
                b: Math.round(blue * 255, 10) };
        };
    }
};

var mm = {x: 0, y: 0};

```

References

1. Alastrue, V., Calvo, B., Pena, E., Doblare, M.: Biomechanical modeling of refractive corneal surgery. *Journal of Biomechanical Engineering-Transactions of the ASME* **128**, 150–160 (2006)
2. Ammar, A., Chinesta, F., Cueto, E.: Coupling finite elements and proper generalized decompositions. *International Journal for Multiscale Computational Engineering* **9**(1), 17–33 (2011)
3. Ammar, A., Chinesta, F., Cueto, E., Doblare, M.: Proper generalized decomposition of time-multiscale models. *International Journal for Numerical Methods in Engineering* **90**(5), 569–596 (2012). DOI 10.1002/nme.3331. URL <http://dx.doi.org/10.1002/nme.3331>
4. Ammar, A., Chinesta, F., Diez, P., Huerta, A.: An error estimator for separated representations of highly multidimensional models. *Computer Methods in Applied Mechanics and Engineering* **199**(25–28), 1872 – 1880 (2010). DOI 10.1016/j.cma.2010.02.012. URL <http://www.sciencedirect.com/science/article/pii/S0045782510000708>
5. Ammar, A., Cueto, E., Chinesta, F.: Nonincremental proper generalized decomposition solution of parametric uncoupled models defined in evolving domains. *International Journal for Numerical Methods in Engineering* pp. n/a–n/a (2012). DOI 10.1002/nme.4413. URL <http://dx.doi.org/10.1002/nme.4413>
6. 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 in press* (2012)
7. 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. *J. Non-Newtonian Fluid Mech.* **139**, 153–176 (2006)
8. 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. *J. Non-Newtonian Fluid Mech.* **144**, 98–121 (2007)
9. Ammar, A., Pruliere, E., Ferec, J., Chinesta, F., Cueto, E.: Coupling finite elements and reduced approximation bases. *European Journal of Computational Mechanics* **18**(5–6), 445–463 (2009)
10. Amsallem, D., Farhat, C.: An Interpolation Method for Adapting Reduced-Order Models and Application to Aeroelasticity. *AIAA Journal* **46**, 1803–1813 (2008)

11. Authors, V.: Final report. DDDAS workshop at Arlington, VA. Tech. rep., National Science Foundation (2006)
12. Barbič, J., James, D.: Time-critical distributed contact for 6-DoF haptic rendering of adaptively sampled reduced deformable models. In: Metaxas, D and Popovic, J (ed.) SYMPOSIUM ON COMPUTER ANIMATION 2007: ACM SIGGRAPH/ EURO-GRAPHICS SYMPOSIUM PROCEEDINGS, pp. 171–180. ACM SIGGRAPH; Eurog Assoc, ASSOC COMPUTING MACHINERY, 1515 BROADWAY, NEW YORK, NY 10036-9998 USA (2007). Symposium on Computer Animation, San Diego, CA, AUG 03-04, 2007
13. Barbič, J., James, D.L.: Real-time subspace integration for St. Venant-Kirchhoff deformable models. *ACM Transactions on Graphics (SIGGRAPH 2005)* **24**(3), 982–990 (2005)
14. Barrault, M., Maday, Y., Nguyen, N., Patera, A.: An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique* **339**(9), 667–672 (2004). DOI {10.1016/j.crma.2004.08.006}
15. Bellotti, G.: Transient response of harbours to long waves under resonance conditions. *Coastal Engineering* **54**, 680–693 (2007)
16. Bird, R.B., Curtiss, C.F., Armstrong, R.C., Hassager, O.: Dynamics of polymeric liquids, vol. 2. John Wiley and Sons. (1987)
17. 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* **201-204**(0), 1 – 12 (2012). DOI 10.1016/j.cma.2011.08.025. URL <http://www.sciencedirect.com/science/article/pii/S0045782511002891>
18. Bro-Nielsen, M., Cotin, S.: Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum* **15**(3), 57–66 (1996)
19. Cancès, E., Defranceschi, M., Kutzelnigg, W., Bris, C.L., Maday, Y.: Computational quantum chemistry: a primer. In: *Handbook of Numerical Analysis*, vol. X, pp. 3–270 (2003)
20. Chaturantabut, S., Sorensen, D.C.: Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **32**, 2737–2764 (2010). DOI 10.1137/090766498. URL <http://dx.doi.org/10.1137/090766498>
21. Chinesta, F., Ammar, A., Cueto, E.: Proper generalized decomposition of multiscale models. *International Journal for Numerical Methods in Engineering* **83**(8-9), 1114–1132 (2010). DOI 10.1002/nme.2794. URL <http://dx.doi.org/10.1002/nme.2794>
22. Chinesta, F., Ammar, A., Cueto, E.: Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering* **17**, 327–350 (2010)
23. Chinesta, F., Ammar, A., Cueto, E.: Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering* **17**, 327–350 (2010). URL <http://dx.doi.org/10.1007/s11831-010-9049-y>
24. Chinesta, F., Cueto, E.: PGD-Based Modeling of Materials, Structures and Processes. Springer International Publishing Switzerland (2014)
25. Chinesta, F., Ladeveze, P., Cueto, E.: A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering* **18**, 395–404 (2011). URL <http://dx.doi.org/10.1007/s11831-011-9064-7>
26. Chinesta, F., Leygue, A., Bordeu, F., Aguado, J., Cueto, E., Gonzalez, D., Alfaro, I., Ammar, A., Huerta, A.: PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Archives of Computational Methods in Engineering* **20**(1), 31–59 (2013). DOI 10.1007/s11831-013-9080-x. URL <http://dx.doi.org/10.1007/s11831-013-9080-x>
27. Cotin, S., Delingette, H., Ayache, N.: Real-time elastic deformations of soft tissues for surgery simulation. In: H. Hagen (ed.) *IEEE Transactions on Visualization and Computer Graphics*, vol. 5 (1), pp. 62–73. IEEE Computer Society (1999). URL citeseer.ist.psu.edu/cotin98realtime.html
28. Courtécuisse, H., Jung, H., Allard, J., Duriez, C., Lee, D.Y., Cotin, S.: Gpu-based real-time soft tissue deformation with cutting and haptic feedback. *Progress in Biophysics and Molecular Biology* **103**(2-3), 159 – 168 (2010). DOI DOI:10.1016/j.pbiomolbio.2010.09.016. URL <http://www.sciencedirect.com/science/article/B6TBN-514BPD2-2/2/fab24de7c7b90422e0e56e3f35b30aa3>. Special Issue on Biomechanical Modelling of Soft Tissue Motion
29. Delingette, H., Ayache, N.: Soft tissue modeling for surgery simulation. In: N. Ayache (ed.) *Computational Models for the Human Body*, *Handbook of Numerical Analysis* (Ph. Ciarlet, Ed.), pp. 453–550. Elsevier (2004)
30. Delingette, H., Ayache, N.: Hepatic surgery simulation. *COMMUNICATIONS OF THE ACM* **48**(2), 31–36 (2005). DOI {10.1145/1042091.1042116}
31. Doswell, J.T.: Augmented learning: Context-aware mobile augmented reality architecture for learning. In: *Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies, ICALT '06*, pp. 1182–1183. IEEE Computer Society, Washington, DC, USA (2006). URL <http://dl.acm.org/citation.cfm?id=1156068.1156186>
32. Fung, Y.C.: *Biomechanics. Mechanical properties of living tissues*. Springer-Verlag (1993)
33. Ghnatios, C., Chinesta, F., Cueto, E., Leygue, A., Poitou, A., Breitskopf, 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* **42**(9), 1169 – 1178 (2011). DOI 10.1016/j.compositesa.2011.05.001. URL <http://www.sciencedirect.com/science/article/pii/S1359835X11001369>
34. Ghnatios, C., 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* **213-216**(0), 29 – 41 (2012). DOI 10.1016/j.cma.2011.11.018. URL <http://www.sciencedirect.com/science/article/pii/S0045782511003641>
35. Gonzalez, D., Masson, F., Poulhaon, F., Cueto, E., Chinesta, F.: Proper generalized decomposition based dynamic data driven inverse identification. *Mathematics and Computers in Simulation* **82**, 1677–1695 (2012)
36. González, D., Ammar, A., Chinesta, F., Cueto, E.: Recent advances on the use of separated representations. *International Journal for Numerical Methods in Engineering* **81**(5), 637–659 (2010). DOI 10.1002/nme.2710. URL <http://dx.doi.org/10.1002/nme.2710>
37. H. C. Öttinger: *Stochastic Processes in Polymeric Fluids*. Springer (1996)
38. Hegland, M., Burden, C., Santoso, L., MacNamara, S., Boothm, H.: A solver for the stochastic master equation applied to gene regulatory networks. *Journal of Computational and Applied Mathematics* **205**, 708–724 (2007)
39. Heyberger, C., Boucard, P.A., NÄ©ron, D.: Multiparametric analysis within the proper generalized decomposition framework. *Computational Mechanics* **49**, 277–289 (2012). DOI 10.1007/s00466-011-0646-x. URL <http://dx.doi.org/10.1007/s00466-011-0646-x>
40. Inwood, M., Ahmad, J.: Development of instructional, interactive, multimedia anatomy dissection software: A student-led initiative. *CLINICAL ANATOMY* **18**(8), 613–617 (2005). DOI {10.1002/ca.20140}

41. Karhunen, K.: Uber lineare methoden in der wahrscheinlichkeitsrechnung. *Ann. Acad. Sci. Fennicae, ser. Al. Math. Phys.* **37** (1946)
42. Ketelhut, D.J.: Augmented Learning: Research and Design of Mobile Educational Games A Review. *JOURNAL OF SCIENCE EDUCATION AND TECHNOLOGY* **19**(2), 212–214 (2010)
43. Klopfer, E., Squire, K.: Environmental Detectives - the development of an augmented reality platform for environmental simulations. *ETR&D-EDUCATIONAL TECHNOLOGY RESEARCH AND DEVELOPMENT* **56**(2), 203–228 (2008). DOI {10.1007/s11423-007-9037-6}
44. Klopfer, E., Yoon, S., Rivas, L.: Comparative analysis of Palm and wearable computers for Participatory Simulations. *JOURNAL OF COMPUTER ASSISTED LEARNING* **20**(5), 347–359 (2004). DOI {10.1111/j.1365-2729.2004.00094.x}
45. Koh, C., Tan, H.S., Tan, K.C., Fang, L., Fong, F.M., Kan, D., Lye, S.L., Wee, M.L.: Investigating the Effect of 3D Simulation-Based Learning on the Motivation and Performance of Engineering Students. *Journal of Engineering Education* **99**(3), 237–251 (2010)
46. Kopfler, E.: Augmented learning. MIT press (2008)
47. Ladeveze, P.: Nonlinear Computational Structural Mechanics. Springer, N.Y. (1999)
48. Ladeveze, P., Chamoin, L.: On the verification of model reduction methods based on the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering* **200**(23–24), 2032–2047 (2011). DOI {10.1016/j.cma.2011.02.019}
49. Ladeveze, P., Passieux, J.C., Neron, D.: The latin multiscale computational method and the proper generalized decomposition. *Computer Methods in Applied Mechanics and Engineering* **199**(21–22), 1287 – 1296 (2010). DOI 10.1016/j.cma.2009.06.023. URL <http://www.sciencedirect.com/science/article/pii/S00457825090002643>
50. Laughlin, R.B., Pines, D.: The theory of everything. *Proceedings of the National Academy of Sciences* **97**(1), 28–31 (2000)
51. Le Bris, C., Lelièvre, T., Maday, Y.: Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations. *Constructive Approximation* **30**, 621–651 (2009). URL <http://dx.doi.org/10.1007/s00365-009-9071-1>
52. Loève, M.M.: Probability theory. The University Series in Higher Mathematics, 3rd ed. Van Nostrand, Princeton, NJ (1963)
53. Lorenz, E.N.: Empirical Orthogonal Functions and Statistical Weather Prediction. MIT, Departement of Meteorology, Scientific Report Number 1, Statistical Forecasting Project (1956)
54. Nguyen, N.C., Patera, A.T., Peraire, J.: A ‘best points’ interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering* **73**(4), 521–543 (2008). DOI {10.1002/nme.2086}
55. 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* **91**(3), 223 – 231 (2008). DOI DOI:10.1016/j.cmpb.2008.04.008. URL <http://www.sciencedirect.com/science/article/B6T5J-4SNPPVY-2/2/8a417e7f1371768b4c928d1f12fc7a0f>
56. Niroomandi, S., Alfaro, I., Cueto, E., Chinesta, F.: Model order reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering* **81**(9), 1180–1206 (2010). DOI 10.1002/nme.2733. URL <http://dx.doi.org/10.1002/nme.2733>
57. 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* **105**(1), 1–12 (2012). DOI DOI:10.1016/j.cmpb.2010.06.012. URL <http://www.sciencedirect.com/science/article/B6T5J-50VGHDD-1/2/1201566766c0d280af9195bf07bfaf91>
58. Niroomandi, S., González, D., Alfaro, I., Bordeu, F., Leygue, A., Cueto, E., Chinesta, F.: Real-time simulation of biological soft tissues: a PGD approach. *International Journal for Numerical Methods in Biomedical Engineering* **29**(5), 586–600 (2013). DOI 10.1002/cnm.2544. URL <http://dx.doi.org/10.1002/cnm.2544>
59. Nouy, A.: A priori model reduction through Proper Generalized Decomposition for solving time-dependent partial differential equations. *Computer Methods in Applied Mechanics and Engineering* **199**(23–24), 1603–1626 (2010). DOI {10.1016/j.cma.2010.01.009}
60. Price, S., Rogers, Y.: Let’s get physical: The learning benefits of interacting in digitally augmented physical spaces. *Computers and Education* **43**, 137–151 (2004)
61. Pruliere, E., Chinesta, F., Ammar, A.: On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition. *MATHEMATICS AND COMPUTERS IN SIMULATION* **81**(4), 791–810 (2010). DOI {10.1016/j.matcom.2010.07.015}
62. Quesada, C., Alfaro, I., Gonzalez, D., Cueto, E.: Interactive simulation of a neohookean liver. <http://amb.unizar.es/liverNeoHookPGDANM.htm>. Accessed: 25/08/2014
63. Ryckelynck, D., Chinesta, F., Cueto, E., Ammar, A.: On the a priori Model Reduction: Overview and recent developments. *Archives of Computational Methods in Engineering* **12**(1), 91–128 (2006)
64. Seitz, A.R., Kim, D., Watanabe, T.: Rewards Evoke Learning of Unconsciously Processed Visual Stimuli in Adult Humans. *Neuron* **61**(5), 700–707 (2009)
65. Selwyn, N.: Augmented Learning: Research and Design of Mobile Educational Games. *INFORMATION COMMUNICATION & SOCIETY* **13**(5), 788–790 (2010). DOI {10.1080/13691181003792616}
66. Squire, K., Klopfer, E.: Augmented reality simulations on handheld computers. *JOURNAL OF THE LEARNING SCIENCES* **16**(3), 371–413 (2007)
67. Swaak, J., de Jong, T.: Measuring intuitive knowledge in science: The development of the what-if test. *Studies In Educational Evaluation* **22**(4), 341 – 362 (1996). DOI 10.1016/0191-491X(96)00019-3. URL <http://www.sciencedirect.com/science/article/pii/0191491X96000193>
68. Talbot, D.: Given tablets but no teachers, ethiopian children teach themselves. <http://www.technologyreview.com/news/506466/given-tablets-but-no-teachers-ethiopian-children-teach-themselves> (2012)
69. Taylor, Z., Comas, O., Cheng, M., Passenger, J., Hawkes, D., Atkinson, D., Ourselin, S.: On modelling of anisotropic viscoelasticity for soft tissue simulation: Numerical solution and gpu execution. *Medical Image Analysis* **13**(2), 234 – 244 (2009). DOI DOI:10.1016/j.media.2008.10.001. URL <http://www.sciencedirect.com/science/article/B6W6Y-4TPF4P9-1/2/d51b8636b70ee79508c7f0472dcd71a>. Includes Special Section on Functional Imaging and Modelling of the Heart
70. Wang, P., Becker, A., Jones, I., Glover, A., Benford, S., Greenhalgh, C., Vloeberghs, M.: Virtual reality simulation of surgery with haptic feedback based on the boundary element method. *Computers and Structures* **85**(7–8), 331 – 339 (2007). DOI DOI:10.1016/j.compstruc.2006.11.021. URL <http://www.sciencedirect.com/science/article/B6V28-4MVN00B-3/2/aa7590237a30247fd45e51de292a9e4e>