ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

TOWARDS A NON-INTRUSIVE PROPER GENERALIZED DECOMPOSITION SCHEME FOR MODEL ORDER REDUCTION

X. Zou[1,2], P. Díez[2], M. Conti[1], and F. Auricchio[1]

[1]Università degli Studi di Pavia
Via Ferrata 3, 27100 Pavia, Italy
e-mail: {xi.zou, michele.conti, auricchio}@unipv.it

[2] Universitat Politècnica de Catalunya
Jordi Girona 1-3 E-08034 Barcelona, Spain
e-mail: pedro.diez@upc.edu

Keywords: Proper Generalized Decomposition, Non-intrusive, Model Order Reduction.

Abstract. *Proper Generalized Decomposition (PGD) has become a popular technique of model order reduction to solve the so-called curse of dimensionality in multidimensional partial differential equations, especially for real-time simulations. Therefore, it is very interesting to implement the PGD algorithm non-intrusively for a better integration with off-the-shelf finite element (FE) codes. This paper provides a non-intrusive scheme for PGD computation based on commercial softwares to minimize the implementation effort. Specifically, also for the sake of simplicity, we chose linear elasticity models with two independent material parameters as the extra coordinates for the preliminary implementations. The strategy is using proper separated representation of the functional of material properties to split the problem into sequentially independent linear equations, and solve the equations of mechanical problem, which usually have a large scale, with an efficient external solver. The proposed non-intrusive PGD scheme can be easily realized using arbitrary codes for iteration control cooperating with commercial FE softwares. Numerical examples were realized with MATLAB (The MathWorks, Inc., USA) as the main procedure for iterations, and Abaqus (Dassault Systèmes, France) as the external solver. In particular, we have applied the algorithm on a biomedical problem. The proposed scheme is extensible and also considered to be a promising bridge between academic research and industrial application.*

1 INTRODUCTION

Modern simulation-based engineering sciences relies on physical models depending on many parameters described by partial differential equations (PDEs) to be solved in a multidimensional parametric space, where the depending parameters are viewed as extra coordinates. For this reason, the dimensionality of the problem is increased and the computational cost is raised exponentially, leading to the so-called *curse of dimensionality*.

Proper generalized decomposition (PGD) method [1, 2, 3, 4] is a recently developed model order reduction technique particularly powerful on tackling the curse of dimensionality. Readers may refer to [5] for a recent review on PGD method. As a model order reduction method, it can be divided into off-line phase and on-line phase. In this paper, we will focus on the off-line phase of PGD method, which produces a computational vademecum [6] consisting of functions of the parameters. Based on the vademecum, the value of the functions can be calculated as fast as real-time in the on-line phase, so that to enable real-time and multi-query simulation.

Numerical methods for solving PDEs, especially the finite element method [7], are intensely implemented in both academia and industry in the form of computer aided engineering (CAE), covering a wide range of fields. There are numerous finite element (FE) codes highlighting different features. Academical institutes tend to develop less complexed but more flexible codes for research purposes. Industrial companies invest resources on commercial CAE softwares and make rules for the application routines, because the provided reliable results with robust FE solvers, and user-friendly graphical interfaces for the pre-processing and post-processing can improve productivity. To bridge the gap in between, we propose a scheme for the application of PGD method in a non-intrusive manner. The idea is using a general-purposed code to perform input/output (I/O) and to control the whole program, in which the key subroutines are executed by calling the external solver. The non-intrusive implementation is guaranteed for PGD method, thanks to the feasibility of separated representation for functions of parameters. Recent work by Courard *et al.* [8] has implied the non-intrusive PGD idea that starts from taking geometric parameters as extra coordinates, and makes SAMCEF (Siemens, Germany) work with an in-house MATLAB code for the generation of virtual charts to optimize engineering design process.

In this work we concentrate on physical models of linear elasticity, whose governing equation is vector-valued. The problems considered include a very simple but typical one-dimensional model, and a practical three-dimensional model as an application in biomechanics. As a preliminary step towards a complete non-intrusive PGD scheme, we take the material properties as the extra coordinates. Further work is in progress for solving problems with load or boundary conditions or geometric parameters as the extra coordinates based on the proposed scheme.

2 PROBLEM STATEMENT

2.1 Governing equation in linear elasticity with two materials

Under the infinitesimal deformation assumption, the relationship between strain tensor $\boldsymbol{\varepsilon}$ and displacement $\mathbf{u}$ reads

$$\boldsymbol{\varepsilon} = \nabla_s \mathbf{u} = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right]. \tag{1}$$

The stress tensor $\boldsymbol{\sigma}$ and the strain tensor are related with elasticity tensor $\mathbb{C}$ by Hook's law

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\varepsilon}. \tag{2}$$

Consider an elastic domain $\Omega \subset \mathbb{R}^d, (d = 1, 2, 3)$, the function of interest is displacement $\mathbf{u}$, of the following boundary value problem (BVP) of the govering equations. The strong form of the BVP is stated as follows: find $\mathbf{u}$ satisfying the equilibrium equation and the boundary conditions

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = 0 \quad \text{in } \Omega, \tag{3}$$

$$\mathbf{u} = \mathbf{u}_D \quad \text{on } \Gamma_D, \tag{4}$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{t}_N \quad \text{on } \Gamma_N, \tag{5}$$

where $\mathbf{b}$ represents the known external body forces. The Dirichlet boundary $\mathbf{u}_D$ and Neumann boundary traction $\mathbf{t}_N$ are also known data sets.

To simulate an elastic body composed of two, for example, parts with different material properties, the domain is partitioned into two subdomains Ω_1 and Ω_2. For the sake of simplicity, we consider only the homogeneous, isotropic materials with the same Poisson's ratio, so that the elasticity tensor $\mathbb{C}$ depends only on Young's Modulus E within each subdomain. So that the Young's Modulus E can be written as

$$E(\mathbf{x}) = \begin{cases} E_1 & \text{for } \mathbf{x} \in \Omega_1, \\ E_2 & \text{for } \mathbf{x} \in \Omega_2. \end{cases} \tag{6}$$

Geometrically, we suppose that $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$ and $\partial\Omega_1 \cap \partial\Omega_2$ is the interface between the two open subdomains. In principle, there is no difficulty to increase the number of subdomains with the same strategy.

The standard weak form of the problem reads as follows: find $\mathbf{u} \in V$ such that

$$a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}), \quad \forall \mathbf{v} \in V_0, \tag{7}$$

where $V := \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = \mathbf{u}_D \text{ on } \Gamma_D\}$, and its corresponding test function space is $V_0 := \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = 0 \text{ on } \Gamma_D\}$. The bilinear and linear forms $a(\cdot, \cdot)$ and $l(\cdot)$ are given by

$$a(\mathbf{u}, \mathbf{v}) := \int_\Omega \nabla_s \mathbf{u} : \mathbb{C} : \nabla_s \mathbf{v} \, d\Omega \quad \text{and} \quad l(\mathbf{v}) := \int_\Omega \mathbf{b} \cdot \mathbf{v} \, d\Omega + \int_{\Gamma_N} \mathbf{t}_N \cdot \mathbf{v} \, d\Gamma. \tag{8}$$

2.2 PGD separated representation with material properties as extra coordinates

The general procedure of the parametrization problem is discussed in [9]. Here we introduce the PGD weak form simply by assuming the material properties as extra coordinates. The displacement is then generalized as $\mathbf{u} = \mathbf{u}(\mathbf{x}, E_1, E_2)$, in which $\mathbf{x} \in \Omega$, $E_1 \in I_{E_1}$ and $E_2 \in I_{E_2}$, and thus the weak form reads: find $\mathbf{u} \in V \otimes L^2(I_{E_1}) \otimes L^2(I_{E_2})$, such that

$$\mathcal{A}(\mathbf{u}, \mathbf{v}) = \mathcal{L}(\mathbf{v}), \quad \forall \mathbf{v} \in V_0 \otimes L^2(I_{E_1}) \otimes L^2(I_{E_2}), \tag{9}$$

where the bilinear and linear forms are defined by

$$\mathcal{A}(\mathbf{u}, \mathbf{v}) := \int_{I_{E_1}} \int_{I_{E_2}} a(\mathbf{u}, \mathbf{v}) \, dE_2 dE_1 \quad \text{and} \quad \mathcal{L}(\mathbf{v}) := \int_{I_{E_1}} \int_{I_{E_2}} l(\mathbf{v}) \, dE_2 dE_1. \tag{10}$$

It can be seen that the dimensionality of the problem is increased by the introduced extra coordinates, which may cause the so-called *curse of dimensionality*. To overcome this with PGD methodology, the separated representation is used to approximate $\mathbf{u}(\mathbf{x}, E_1, E_2)$ as

$$\mathbf{u}(\mathbf{x}, E_1, E_2) \approx \mathbf{u}^n_{\text{PGD}}(\mathbf{x}, E_1, E_2) = \sum_{m=1}^{n} \boldsymbol{\chi}^m(\mathbf{x}) \varepsilon_1^m(E_1) \varepsilon_2^m(E_2). \tag{11}$$

This representation implies that we can approximate the solution by using only n terms of modes, which are products of the separated functions $\boldsymbol{\chi}(\mathbf{x})$, $\varepsilon_1(E_1)$ and $\varepsilon_2(E_2)$. Hereafter, the explicit dependence of these functions is frequently omitted in the notations since readers can easily identify from the names and the subscripts. The normalization of the products is useful by defining the magnitude of the modes

$$M^m := \|\boldsymbol{\chi}^m\| \cdot \|\varepsilon_1^m\| \cdot \|\varepsilon_2^m\|. \tag{12}$$

Derivation of the PGD algorithm requires assuming that the bilinear and linear forms $a(\mathbf{u}, \mathbf{v})$ and $l(\mathbf{v})$ are also separable. That is to say, the functions involved should be separable. For our purposes, as generally seen in real structural problems, we assume that the load is independent of the material properties; while the elasticity tensor $\mathbb{C}$ is dependent on the Young's modulus of the subdomains in a generalized form of Equation (6)

$$\mathbb{C}(\mathbf{x}, E_1, E_2) = \sum_{k=1}^{n_{\mathbb{C}}} \mathbb{C}_k(\mathbf{x})\mu_1^k(E_1)\mu_2^k(E_2), \tag{13}$$

where $\mathbb{C}_k(\mathbf{x})$, $\mu_1^k(E_1)$ and $\mu_2^k(E_2)$ describe the material properties in each of the $n_{\mathbb{C}}$ subdomains. In the simplest two-material case, without losing generality, the Heaviside-like function $H_k(\mathbf{x}) : \Omega \to \{0, 1\}$ is used as

$$\mathbb{C}(\mathbf{x}, E_1, E_2) = \mathbb{C}_1 H_1(\mathbf{x})E_1 + \mathbb{C}_2 H_2(\mathbf{x})E_2, \tag{14}$$

in which $\mathbb{C}_i$ is a constant tensor and

$$H_i(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \in \Omega_i, \\ 0, & \text{otherwise.} \end{cases} \tag{15}$$

The function related to Young's modulus reads

$$\mu_i^j(E_i) = \begin{cases} E_i, & i = j, \\ 1, & i \neq j. \end{cases} \tag{16}$$

In practical FE analysis, following the standard approach, $\mathbb{C}$ is represented in a discretized form by the stiffness matrix $\mathbf{K}$, which is computed by FE softwares as long as the user inputs given values for E_1 and E_2 and specifies the mapping of material properties to the subdomains (or element sets) in the pre-processing.

2.3 Non-intrusive PGD mode search with alternated directions

In general, the number of modes n which needed to approximate $\mathbf{u}_{\text{PGD}}^n$ is not known a priori. The greedy approach is used to calculate the modes on-the-fly with proper stopping criteria. Note that

$$\mathbf{u}_{\text{PGD}}^n(\mathbf{x}, E_1, E_2) = \mathbf{u}_{\text{PGD}}^{n-1}(\mathbf{x}, E_1, E_2) + \boldsymbol{\chi}^n \varepsilon_1^n \varepsilon_2^n, \tag{17}$$

we can compute the modes term by term. From now on the superscript n for the functions is omitted since we are more focused on the terms. Due to the bilinearity, we have turned the problem into: find $\boldsymbol{\chi} \in V$, $\varepsilon_1 \in L^2(I_{E_1})$ and $\varepsilon_2 \in L^2(I_{E_2})$ such that

$$\mathcal{A}(\boldsymbol{\chi}\varepsilon_1\varepsilon_2, \mathbf{v}) = \mathcal{L}(\mathbf{v}) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \mathbf{v}), \quad \forall \mathbf{v} \in V_0 \otimes L^2(I_{E_1}) \otimes L^2(I_{E_2}). \tag{18}$$

To apply FE approximation properly, the fixed-point iteration method is used by treating the terms of test function $\mathbf{v}$ separately from normal variation $\mathbf{v} = \mathbf{u}^* = \boldsymbol{\chi}^*\varepsilon_1\varepsilon_2 + \boldsymbol{\chi}\varepsilon_1^*\varepsilon_2 + \boldsymbol{\chi}\varepsilon_1\varepsilon_2^*$. With this idea, the alternating directions approach is adopted to split (18) into three sequential linear problems:

1. Assuming ε_1 and ε_2 known, update $\boldsymbol{\chi}$ such that

$$\mathcal{A}(\boldsymbol{\chi}\varepsilon_1\varepsilon_2, \boldsymbol{\chi}^*\varepsilon_1\varepsilon_2) = \mathcal{L}(\boldsymbol{\chi}^*\varepsilon_1\varepsilon_2) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \boldsymbol{\chi}^*\varepsilon_1\varepsilon_2), \quad \forall \boldsymbol{\chi}^* \in V_0. \tag{19}$$

2. Assuming $\boldsymbol{\chi}$ and ε_2 known, update ε_1 such that

$$\mathcal{A}(\boldsymbol{\chi}\varepsilon_1\varepsilon_2, \boldsymbol{\chi}\varepsilon_1^*\varepsilon_2) = \mathcal{L}(\boldsymbol{\chi}\varepsilon_1^*\varepsilon_2) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \boldsymbol{\chi}\varepsilon_1^*\varepsilon_2), \quad \forall \varepsilon_1^* \in L^2(I_{E_1}). \tag{20}$$

3. Assuming $\boldsymbol{\chi}$ and ε_1 known, update ε_2 such that

$$\mathcal{A}(\boldsymbol{\chi}\varepsilon_1\varepsilon_2, \boldsymbol{\chi}\varepsilon_1\varepsilon_2^*) = \mathcal{L}(\boldsymbol{\chi}\varepsilon_1\varepsilon_2^*) - \mathcal{A}(\mathbf{u}_{\text{PGD}}^{n-1}, \boldsymbol{\chi}\varepsilon_1\varepsilon_2^*), \quad \forall \varepsilon_2^* \in L^2(I_{E_2}). \tag{21}$$

We would like to address that (19) is a mechanical problem, since $\boldsymbol{\chi}$ is a vector-valued function, that usually takes much more cost to compute, while (20) and (21) are one-dimensional problems, since ε_1 and ε_2 are scalar functions, which can be solved with little computational cost. The key idea of the non-intrusive scheme we propose is making use of off-the-shelf FE softwares to solve the mechanical problem numerically with the expectation of efficiency improvement for both coding and computing.

2.4 FE approximation and resulting linear algebra formulation

Choosing proper subspaces $V^h \subset V$, $V_0^h \subset V_0$, $V_1^h \subset L^2(I_{E_1})$ and $V_2^h \subset L^2(I_{E_2})$, we perform the Galerkin approximation by choosing proper discrete approximations following or imitating the standard FE formulations. That is

$$\boldsymbol{\chi} \approx \mathbf{N}^T\mathbf{U}, \quad \varepsilon_1 \approx \mathbf{N}_1^T\boldsymbol{\varepsilon}_1, \quad \varepsilon_2 \approx \mathbf{N}_2^T\boldsymbol{\varepsilon}_2, \tag{22}$$

and thus

$$\mathbf{u}_{\text{PGD}}^n \approx \sum_{m=1}^{n} \mathbf{N}^T\mathbf{U}^m \cdot \mathbf{N}_1^T\boldsymbol{\varepsilon}_1^m \cdot \mathbf{N}_2^T\boldsymbol{\varepsilon}_2^m, \tag{23}$$

where $\mathbf{N}$ is the matrix of shape functions of the standard FE approach for the mechanical problem, $\mathbf{N}_1$ and $\mathbf{N}_2$ are vectors of shape functions for one-dimensional problems. Note that the shape functions are fixed, only the degree of freedom (DOF) vectors $\mathbf{U}$, $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_1$ will be updated throughout the computation. Empirically, to initialize the computation, we assume $\mathbf{U}$ as a zero vector while $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_2$ are vectors in which each component equals to one. The dimensions of the DOF vectors can be chosen according to the desired precision.

In standard FE theory, the global stiffness matrix $\mathbf{K}$ is an assembly of elemental stiffness matrices, so it has a natural separable formulation. According to (13), the global stiffness matrix $\mathbf{K}$ can be rewritten as

$$\mathbf{K} = E_1\mathbf{K}_1 + E_2\mathbf{K}_2, \tag{24}$$

where $\mathbf{K}_1$ and $\mathbf{K}_2$ are stiffness-like matrices. It is convenient that we still refer $\mathbf{K}_1$ and $\mathbf{K}_2$ to stiffness matrices without any ambiguity.

With the help of external FE software, we can obtain $\mathbf{K}_1$ and $\mathbf{K}_2$ in the non-intrusive manner. In particular, using Equation (24), $\mathbf{K}_1$ can be obtained by letting $E_1 = 1$ and $E_2 = 0$, while $\mathbf{K}_2$ be obtained by $E_1 = 0$ and $E_2 = 1$.

By putting (22) to (24), along with (8), (10), (11) and (13) into (19) to (21), we can obtain the algebraic formulation as

1. Assuming $\boldsymbol{\varepsilon}_1$ and $\boldsymbol{\varepsilon}_2$ known, update $\mathbf{U}$ by solving

$$(M_1 C_2 \mathbf{K}_1 + M_2 C_1 \mathbf{K}_2)\mathbf{U} = Q_1 Q_2 \mathbf{F} - \sum_{k=1}^{n-1} (M_1^k C_2^k \mathbf{K}_1 + M_2^k C_1^k \mathbf{K}_2)\mathbf{U}^k. \tag{25}$$

   Letting $\tilde{\mathbf{K}} = M_1 C_2 \mathbf{K}_1 + M_2 C_1 \mathbf{K}_2$ and $\tilde{\mathbf{F}} = Q_1 Q_2 \mathbf{F} - \sum_{k=1}^{n-1} (M_1^k C_2^k \mathbf{K}_1 + M_2^k C_1^k \mathbf{K}_2)\mathbf{U}^k$, Equation (25) is simplified to $\tilde{\mathbf{K}}\mathbf{U} = \tilde{\mathbf{F}}$, which has a standard FE form, thus can be seen as a fictitious mechanical problem and be sent to the external FE software for a non-intrusive solution, taking advantage of its robust solvers.

2. Assuming $\mathbf{U}$ and $\boldsymbol{\varepsilon}_2$ known, update $\boldsymbol{\varepsilon}_1$ by solving

$$(K_1 C_2 \mathbf{M}_1 + M_2 K_2 \mathbf{C}_1)\boldsymbol{\varepsilon}_1 = F Q_2 \mathbf{Q}_1 - \sum_{k=1}^{n-1} (K_1^k C_2^k \mathbf{M}_1 + K_2^k M_2^k \mathbf{C}_1)\boldsymbol{\varepsilon}_1^k. \tag{26}$$

3. Assuming $\mathbf{U}$ and $\boldsymbol{\varepsilon}_1$ known, update $\boldsymbol{\varepsilon}_2$ by solving

$$(M_1 K_1 \mathbf{C}_2 + K_2 C_1 \mathbf{M}_2)\boldsymbol{\varepsilon}_2 = F Q_1 \mathbf{Q}_2 - \sum_{k=1}^{n-1} (K_1^k M_1^k \mathbf{C}_2 + K_2^k C_1^k \mathbf{M}_2)\boldsymbol{\varepsilon}_2^k. \tag{27}$$

The coefficients and matrices are defined as follows

$$\mathbf{K}_i := \int_{\Omega_i} \nabla_s \mathbf{N} : \mathbb{C}_i : \nabla_s \mathbf{N}^T \, d\Omega, \quad K_i := \mathbf{U}^T \mathbf{K}_i \mathbf{U}, \quad K_i^k := \mathbf{U}^T \mathbf{K}_i \mathbf{U}^k, \quad i = 1, 2. \tag{28}$$

$$\mathbf{M}_i := \int_{I_{E_i}} E_i \mathbf{N}_i \mathbf{N}_i^T \, dE_i, \quad M_i := \boldsymbol{\varepsilon}_i^T \mathbf{M}_i \boldsymbol{\varepsilon}_i, \quad M_i^k := \boldsymbol{\varepsilon}_i^T \mathbf{M}_i \boldsymbol{\varepsilon}_i^k, \quad i = 1, 2. \tag{29}$$

$$\mathbf{C}_i := \int_{I_{E_i}} \mathbf{N}_i \mathbf{N}_i^T \, dE_i, \quad C_i := \boldsymbol{\varepsilon}_i^T \mathbf{C}_i \boldsymbol{\varepsilon}_i, \quad C_i^k := \boldsymbol{\varepsilon}_i^T \mathbf{C}_i \boldsymbol{\varepsilon}_i^k, \quad i = 1, 2. \tag{30}$$

$$\mathbf{Q}_i := \int_{I_{E_i}} \mathbf{N}_i^T \, dE_i, \quad Q_i := \mathbf{Q}_i^T \boldsymbol{\varepsilon}_i, \quad i = 1, 2. \tag{31}$$

$$\mathbf{F} := \int_{\Omega} \mathbf{N}^T \mathbf{b} \, d\Omega + \int_{\Gamma_N} \mathbf{N}^T \mathbf{t}_N \, d\Gamma, \quad F := \mathbf{U}^T \mathbf{F}. \tag{32}$$

In the definitions above, the load vector $\mathbf{F}$ is exactly the same as that in standard FE theory, since we have the assumption that it is independent of the material properties. Therefore, $\mathbf{F}$ can be obtained directly from the external FE software. As previously stated, $\mathbf{K}_i$ can be obtained non-intrusively by giving proper values of E_1 and E_2. By definition, $\mathbf{M}_i$ and $\mathbf{C}_i$ are quasi-diagonal sparse matrices, and $\mathbf{Q}_i$ is a vector, all of them can be calculated in conventional ways.

2.5 Non-intrusive PGD algorithm

We take advantage of MATLAB as the main code, and Abaqus as the external FE solver. See the flowchart of non-intrusive PGD algorithm in Figure 1.

The pre-processing of the mechanical model can be performed in Abaqus/CAE, which will produce an input file containing data of the node-element topology, the load and boundary conditions, and the material properties with associated element sets. In our non-intrusive scheme, stiffness matrix $\mathbf{K}_i$ and load vector $\mathbf{F}$ are generated within Abaqus. In addition, the Python interface provided by Abaqus is also used for data I/O.

The matrices $\mathbf{M}_i$, $\mathbf{C}_i$ and vector $\mathbf{Q}_i$ are all calculated and stored in MATLAB once for all. While the scalar parameters K_i, M_i, C_i, Q_i are updated and stored with mode index k in each iteration.

The idea that using the external FE solver to resolve the mechanical problem (25) can also be generalized to the parametric problems (26) and (27). However, the parametric problems are one-dimensional, which can be solved very quickly within the main code, so that this generalization is usually unnecessary.

Two error limits, η_1 and η_2, are introduced within the stopping criteria of the iteration loops: η_1 controls the iteration of alternative direction, η_2 controls the search for PGD modes. In the following examples, we have empirically chosen $\eta_1 = 10^{-5}$ and $\eta_2 = 1$.

In standard FE analysis in mechanics, one is usually more interested in the global nodal result $\mathbf{U}$ other than the field result $\mathbf{u}(\mathbf{x})$. To be compatible, we can rewrite (23) as

$$\mathbf{u}^n_{\text{PGD}}(\mathbf{x}, E_1, E_2) \approx \mathbf{N}^T(\mathbf{x})\mathbf{U}_{\text{PGD}}(E_1, E_2), \tag{33}$$

by introducing

$$\mathbf{U}_{\text{PGD}} := \sum_{m=1}^{n} \mathbf{U}^m \cdot \mathbf{N}_1^T \boldsymbol{\varepsilon}_1^m \cdot \mathbf{N}_2^T \boldsymbol{\varepsilon}_2^m. \tag{34}$$

So that we can use $\mathbf{U}_{\text{PGD}}$ to make comparisons with standard FE result $\mathbf{U}_{\text{FE}}$.

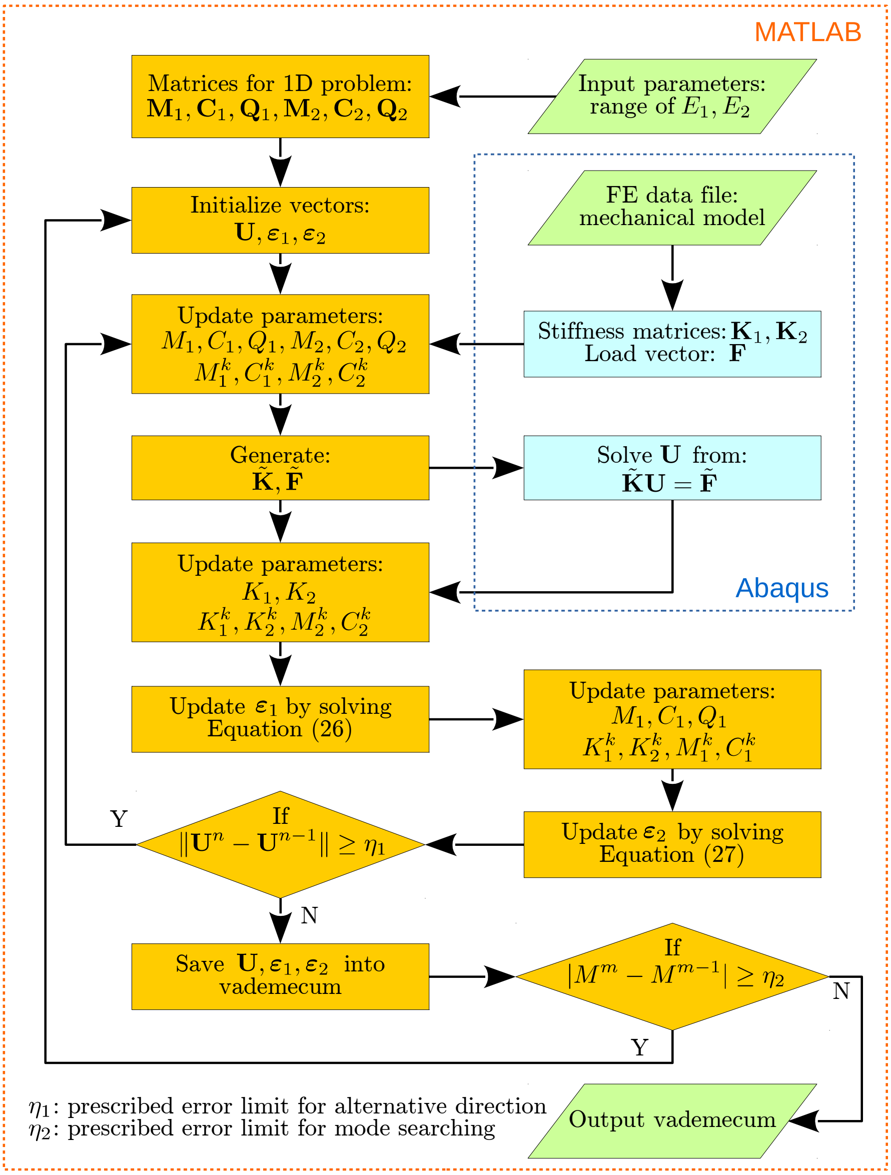
MATLAB
Matrices for 1D problem: $\mathbf{M}_1, \mathbf{C}_1, \mathbf{Q}_1, \mathbf{M}_2, \mathbf{C}_2, \mathbf{Q}_2$
Input parameters: range of E_1, E_2
Initialize vectors: $\mathbf{U}, \varepsilon_1, \varepsilon_2$
FE data file: mechanical model
Update parameters: $M_1, C_1, Q_1, M_2, C_2, Q_2$ $M_1^k, C_1^k, M_2^k, C_2^k$
Stiffness matrices: $\mathbf{K}_1, \mathbf{K}_2$ Load vector: $\mathbf{F}$
Generate: $\tilde{\mathbf{K}}, \tilde{\mathbf{F}}$
Solve $\mathbf{U}$ from: $\tilde{\mathbf{K}}\mathbf{U} = \tilde{\mathbf{F}}$
Update parameters: K_1, K_2 $K_1^k, K_2^k, M_2^k, C_2^k$
Abaqus
Update ε_1 by solving Equation (26)
Update parameters: M_1, C_1, Q_1 $K_1^k, K_2^k, M_1^k, C_1^k$
Y
If $\|\mathbf{U}^n - \mathbf{U}^{n-1}\| \geq \eta_1$
Update ε_2 by solving Equation (27)
N
Save $\mathbf{U}, \varepsilon_1, \varepsilon_2$ into vademecum
If $|M^m - M^{m-1}| \geq \eta_2$
N
Y
η_1: prescribed error limit for alternative direction
η_2: prescribed error limit for mode searching
Output vademecum

Figure 1: Flowchart of non-intrusive PGD algorithm exemplified by MATLAB as the main program and Abaqus as the external solver.

3 NUMERICAL EXAMPLES

3.1 Example I: a one-dimensional model problem

Consider a one-dimensional mechanical model problem consisting of two vertical bars, shown in Figure 2, with unit cross-section area $A_1 = A_2 = 1$ mm^2 and with different lengths $L_1 = 10$ mm and $L_2 = 20$ mm and Young's moduli $E_1 \in [10, 100]$ MPa and $E_2 \in [20, 200]$ MPa. The lower end (Node 3) is fixed ($u_3 = 0$) and the upper end (Node 1) loaded with a concentrated force $F = -10$ N. From standard FE theory, we know that the load vector is $\mathbf{F} = F[1, 0, 0]^T$ and the stiffness matrix is naturally separable

$$\mathbf{K} = E_1\mathbf{K}_1 + E_2\mathbf{K}_2 = \frac{E_1}{L_1}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{E_2}{L_2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}. \tag{35}$$

Assuming no buckling would happen, we have the analytical solution of a stiffness approach

$$\mathbf{U}_{\text{ex}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{K}^{-1}\mathbf{F} = F\begin{bmatrix} L_1/E_1 + L_2/E_2 \\ L_2/E_2 \\ 0 \end{bmatrix}. \tag{36}$$

Note that taking E_1 and E_2 as extra coordinates, Equation (36) has implied that the solutions for u_1 and u_2 are nonlinear. We select 100 DOFs to discretize the extra coordinates for functions ε_1 and ε_2.

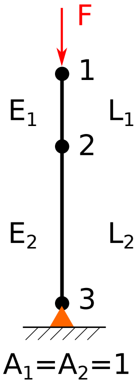

Figure 2: Sketch of the one-dimensional mechanical problem: two vertical bars.

Solving the problem with the proposed non-intrusive PGD approach, the resulting magnitudes decrease with mode sequence, which is plotted in Figure 3. From Figure 4 we can find that the relative error, which is defined as $(u_{\text{PGD}} - u_{\text{ex}})/u_{\text{ex}}$, tends to be larger when E_1 and E_2 reach their lower bounds due to the nonlinearity.

Another approach is to take the slenderness of the materials, $S_1 = 1/E_1$ and $S_2 = 1/E_2$, as extra coordinates. So that we are able to rewrite (35) and (36) as

$$\mathbf{K} = \frac{1}{S_1}\mathbf{K}_1 + \frac{1}{S_2}\mathbf{K}_2 = \frac{1}{L_1S_1}\begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \frac{1}{L_2S_2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}, \tag{37}$$

$$\mathbf{U}_{\text{ex}} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = F\begin{bmatrix} L_1S_1 + L_2S_2 \\ L_2S_2 \\ 0 \end{bmatrix}. \tag{38}$$

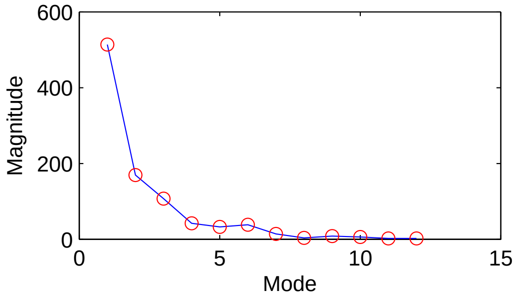

Figure 3: PGD mode-magnitudes relationship of Example I with stiffness approach.

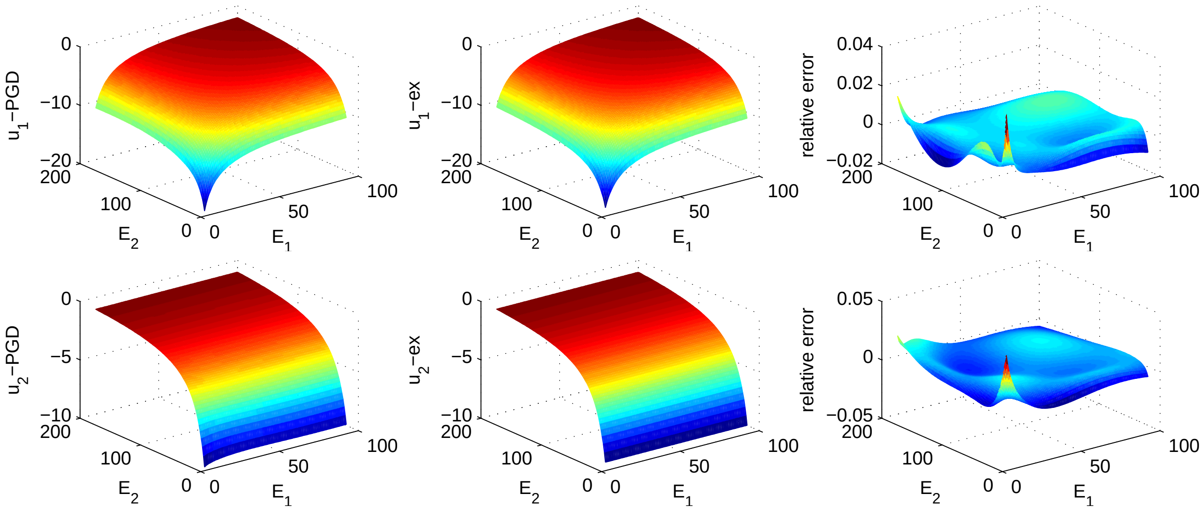

Figure 4: Verification of nodal displacements (u_1 and u_2) for Example I with stiffness approach.

The slenderness approach has converted u_1 and u_2 into linear functions of S_1 and S_2, which is demonstrated in Equation (38) and illustrated in Figure 6. Thanks to the linearity, the modes required to achieve prescribed accuracy is decreased from twelve (Figure 3) to nine (Figure 5).

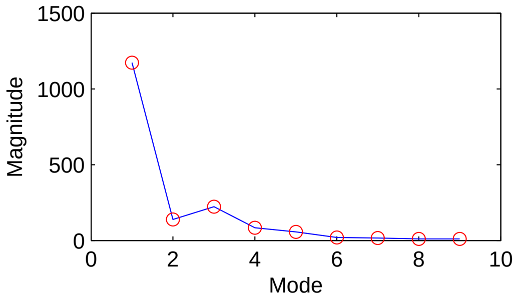

Figure 5: PGD mode-magnitudes relationship of Example I with slenderness approach.

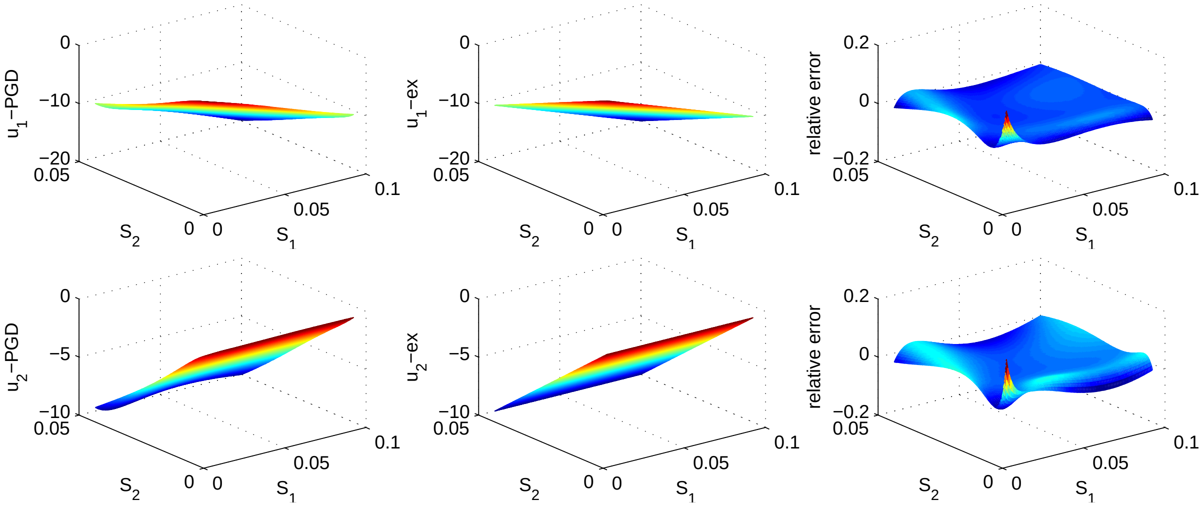

Figure 6: Verification of nodal displacements (u_1 and u_2) for Example I with slenderness approach.

3.2 Example II: a three-dimensional model of human proximal femur

As an example for biomechanical applications of the non-intrusive PGD scheme, a FE model of human proximal femur is built with tetrahedral elements in Abaqus/CAE. The femur bone is fixed at the distal end and loaded vertically at the femur head with $F = -740$ N (uniformly distributed to 74 nodes), see the sketch in Figure 7. The Young's modulus of the proximal epiphysis is $E_1 \in [8000, 10000]$ MPa and that of the cortical bone is $E_2 \in [9000, 15000]$ MPa.

Solving the problem with the proposed non-intrusive PGD scheme, the resulting mode-magnitude relationship demonstrates very well the linearity of the three-dimensional model: from Mode 2 on, the magnitudes are close to zero, which means a certain accuracy could be achieved with only two modes. The mode-magnitude plot is shown in Figure 8 and the displacement at the femur head is plotted in components in Figure 9. Since we have no analytical solution for this three-dimensional problem, there is no straightforward way to plot the error surface like Figure 4 and Figure 6. To verify the proposed non-intrusive PGD approach, we select a fixed (E_1, E_2) and compare displacements between standard FE result $\mathbf{U}_{\mathrm{FE}}$ and non-intrusive PGD result $\mathbf{U}_{\mathrm{PGD}}$ by computing nodal error $\mathbf{U}_{\mathrm{PGD}} - \mathbf{U}_{\mathrm{FE}}$. All seven modes are used to compute $\mathbf{U}_{\mathrm{PGD}}$. For the purpose of illustration, we compute the magnitude of the nodal error at each node, and the result is shown in Figure 10.

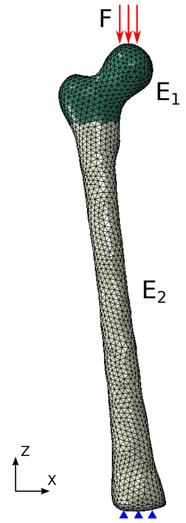

Figure 7: Sketch of the human proximal femur model.

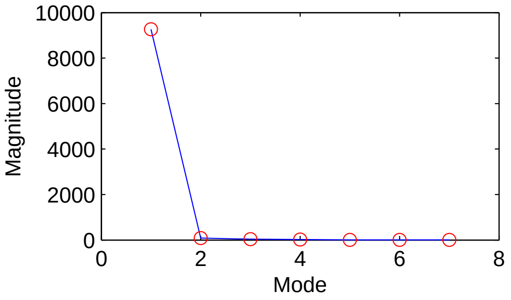

Figure 8: PGD mode-magnitude relationship for human proximal femur model.

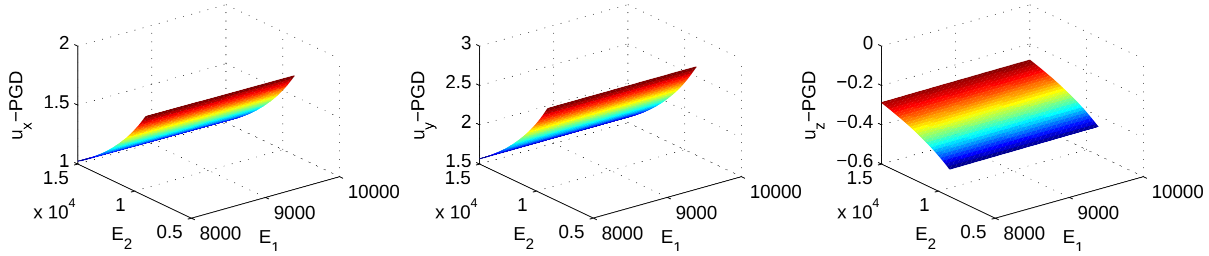

Figure 9: Non-intrusive PGD results of displacement components (u_x, u_y, u_z) at a node on the femur head. The dependence on E_1 is much less apparent than that on E_2.

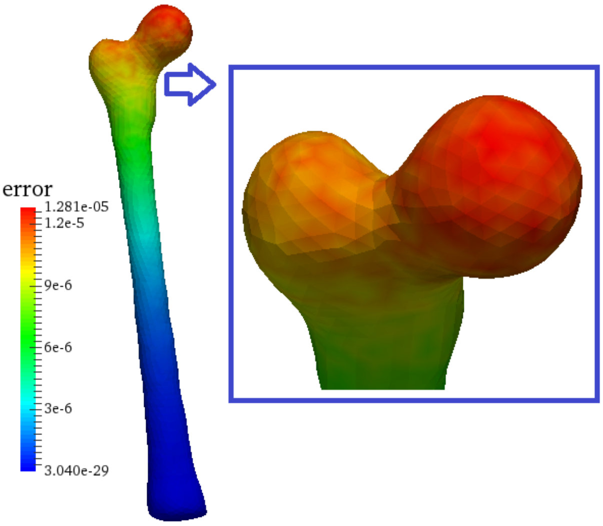

Figure 10: Magnitude of nodal error of non-intrusive PGD result with reference to standard FE result.

4 CONCLUSIONS

This work addresses a non-intrusive approach for solving multidimensional PDEs with PGD methods taking advantage of commercial FE codes with high-maturity. The material properties are considered as extra coordinates in the multidimensional governing equations of linear elasticity. One of the most significant highlights is in computation of large-scaled multidimensional problems, the effort of developing and certifying codes for solving physical problems is saved by the non-intrusive approach. In particular, the stiffness matrices and load vector are also computed by the external FE software during the pre-processing.

The proposed non-intrusive PGD scheme is highly extensible and has promisingly wide applications. For example, the presented linear elasticity problems can be easily extended to nonlinear elasticity problems [10]. In addition, geometric parameters, load and boundary conditions are also able to be taken into account as extra coordinates to be solved within the scheme.

5 ACKNOWLEDGEMENT

We acknowledge the support from the European Education, Audiovisual and Culture Executive Agency (EACEA) under the Erasmus Mundus Joint Doctorate *Simulation in Engineering and Entrepreneurship Development* (SEED), FPA 2013-0043.

Dr. M. Conti acknowledges European Reasearch Council through the ERC Starting Grant ISOBIO: Isogeometric Methods for Biomechanics (No. 259229).

REFERENCES

[1] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, 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*, **139**, 153–176, 2006.

[2] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. Part II: Transient simulation using space-time separated representations. *Journal of Non-Newtonian Fluid Mechanics*, **144**, 98–121, 2007.

[3] A. Ammar, The proper generalized decomposition: A powerful tool for model reduction. *International Journal of Material Forming*, **3**, 89–102, 2010.

[4] F. Chinesta, A. Ammar, E. Cueto, 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.

[5] F. Chinesta, P. Ladevèze, and E. Cueto, A Short Review on Model Order Reduction Based on Proper Generalized Decomposition. *Archives of Computational Methods in Engineering*, **18**, 395–404, 2011.

[6] F. Chinesta, A. Leygue, F. Bordeu, J.V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, A. Huerta, PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Archives of Computational Methods in Engineering*, **20**, 31–59, 2013.

[7] O.C. Zienkiewicz, R.C. Taylor, J.Z. Zhu, *The finite element method, Vol. I, 7th Edition*. Butterworth-Heinemann, 2013.

[8] A. Courard, D. Néron, P. Ladevèze, L. Ballere, Integration of PGD-virtual charts into an engineering design process. *Computational Mechanics*, 2015.

[9] S. Zlotnik, P. Díez, D. Modesto, A. Huerta, Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *International Journal for Numerical Methods in Engineering*, **103**, 737–758, 2015.

[10] S. Niroomandi, I. Alfaro, D. Gonzlez, E. Cueto, F. Chinesta, Model order reduction in hyperelasticity: a proper generalized decomposition approach. *International Journal for Numerical Methods in Engineering*, **96**, 129–149, 2013.