Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 389 (2022) 114336
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Nonintrusive parametric solutions in structural dynamics

F. Cavaliere[a,b,c,*], S. Zlotnik[a,b], R. Sevilla[c], X. Larrayoz[d], P. Díez[a,b]

[a] Laboratori de Càcul Numèric (LaCàN), E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Spain
[b] International Centre for Numerical Methods in Engineering, CIMNE, Barcelona, Spain
[c] Zienkiewicz Centre for Computational Engineering, Faculty of Science and Engineering, Swansea University, Swansea, SA1 8EN, Wales, UK
[d] Centro Técnico de SEAT S.A, Autovía A-2, km 85, Martorell 08760, Spain

Received 13 July 2021; received in revised form 19 October 2021; accepted 7 November 2021
Available online 29 November 2021

Abstract

A nonintrusive reduced order method able to solve a parametric modal analysis is proposed in this work. The main objective is being able to efficiently identify how a variation of user-defined parameters affects the dynamic response of the structure in terms of fundamental natural frequencies and corresponding mode shapes. A parametric version of the inverse power method (IPM) is presented by using the proper generalised decomposition (PGD) rationale. The proposed approach utilises the so-called encapsulated PGD toolbox and includes a new algorithm for computing the square root of a parametric object. With only one offline computation, the proposed PGD-IPM approach provides an analytical parametric expression of the smallest (in magnitude) eigenvalue (or natural frequency) and corresponding eigenvector (mode shape), which contains all the possible solutions for every combination of the parameters within pre-defined ranges. A Lagrange multiplier deflation technique is introduced in order to compute subsequent eigenpairs, which is also valid to overcome the stiffness matrix singularity in the case of a free-free structure. The proposed approach is nonintrusive and it is therefore possible to be integrated with commercial finite element (FE) packages. Two numerical examples are shown to underline the properties of the technique. The first example includes one material and one geometric parameter. The second example shows a more realistic industrial example, where the nonintrusivity of the approach is demonstrated by employing a commercial FE package for assembling the FE matrices. Finally, a multi-objective optimisation study is performed proving that the developed method could significantly assist the decision-making during the preliminary phase of a new design process.

Keywords: Algebraic PGD; Parametric modal analysis; Reduced order model; Shape optimisation

1. Introduction

The dynamic response of an engineering structure depends on its material and geometric properties. Changes in the design parameters can have considerable effects on the structural dynamic behaviour. For this reason, during the development process of a new system, designers have to run sensitivity analysis, such that potential problems can be identified and corrected before time, money and resources are wasted on prototyping and manufacturing a non-optimal design. The modal analysis is widely used in industry to predict the dynamic properties of a structure under

* Corresponding author at: Laboratori de Càcul Numèric (LaCàN), E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya, Spain.
E-mail address: fabiola.cavaliere1@upc.edu (F. Cavaliere).

https://doi.org/10.1016/j.cma.2021.114336

free-vibration conditions. This consists of solving a generalised eigenvalue problem where the eigenvectors represent the natural deformation of the structure when vibrating (mode shapes), and the eigenvalues are the corresponding natural frequencies. Trying to efficiently identify how the mode shapes and natural frequencies depend upon the design parameters is still today a challenging computational task. In fact, standard mesh-based discretisation methods would approach this multidimensional problem by solving it for several combinations of the parameters. Due to the exponential increase of the computational complexity with the dimension of the problem, this task becomes unaffordable in real industrial applications. In order to overcome this issue, known as the *curse of dimensionality*, reduced order methods (ROMs) represent an attractive alternative. ROMs are based on the idea that the essential behaviour of complex systems can be accurately described by simplified low-order models. Several ROMs have been developed in the last decades and employed in the most diverse applications. Usually, they are classified into *a-posteriori* and *a-priori* methods. Some of the most popular a-posteriori methods are the Krylov-based methods [1], the reduced basis method [2], and the proper orthogonal decomposition (POD) technique [3], which is one of the most applied in the context of structural dynamics [4–7]. These approaches first need to solve the full-order problem for a suitably chosen set of parameters, such that a set of basis functions describing the most relevant characteristics of the solution can be computed. The obtained reduced model is then used to solve similar problems at a much cheaper computational cost. Clearly, the accuracy of the solution for any new value of the parameters highly depends on the right choice of the representative set of problems used to extract the reduced basis. An appealing alternative is represented by the proper generalised decomposition (PGD) method [8–10], that is able to provide explicit parametric solutions to parametric boundary value problems. In its standard from, PGD is an *a-priori* reduced order technique, in the sense that it computes the reduced basis without relying on previously computed full-order solutions associated with arbitrary samples of the parametric space. This is possible thanks to the main assumption of the method, that is to treat the parameters as extra coordinates and approximate the solution of the resulting high-dimensional problem as a sum of functional products, each one involving the basis functions that depend separately on the parameters by assumption. The PGD algorithm is constructed such that the basis functions are computed simultaneously and on-the-fly during an offline stage, usually performed by employing high-performance computing resources. In order to provide the solution in the described explicit parametric representation, the method requires the input data of the problem to be described in the same separated format. Unfortunately, it is not always possible to find an analytical expression of the separated input data (such as stiffness matrices), especially when general geometric parameters are involved in the problem. For this reason, the authors recently developed [11] a *hybrid* version of the PGD method. On one hand, it can still be considered an a-priori approach, as it does not require any previous full-order computation of the solution. On the other hand, a pre-process step is added, where the input data are only sampled (without solving the problem) in the parametric domain of interest and then expressed in the required separated format. This then allows to proceed as in the standard PGD method and explore any user-defined parametric domain with only one offline computation. Afterwards, in an online stage, the obtained solution can be particularised in real time for any choice of the parameters in the domain. The PGD method has been successfully tested in the most diverse fields, such as flow problems [12–17], thermal problems [18–20], solid mechanics [21–23], fracture mechanics [24,25], geophysical problems [26,27], elastic metamaterials and coupled magneto-mechanical problems [28,29]. Several applications are available also in the dynamic framework. For example, [30] and [31] proposed a POD-PGD approach to implement a real-time integration scheme able to solve the equations of solid dynamics depending on parametric initial conditions. In [32,33] and [34], PGD is coupled with a hybrid integration scheme which combines modal and harmonic analysis to efficiently solve frequency-dependent parametric problems. In this work, the PGD approach has been coupled with an eigensolver technique, the inverse power (IPM) method, to efficiently identify how a variation of user-defined material and geometric parameters affects the dynamic response of the structure in terms of dominant eigenmodes and related natural frequencies. Similar works have been presented in the field of nuclear reactor analysis, which confirms the potentiality of the proposed method in general areas of applications. In particular, the PGD method has been coupled with standard eigenvalue solvers, such as the Arnoldi technique [35], the IPM [36] and the direct power method [37], in order to solve single- and multi-group neutron diffusion eigenvalue problems. The PGD-IPM computational tool presented in this work uses a PGD-based library known as Encapsulated-PGD toolbox [38]. The toolbox not only allows to solve parametric boundary value problems as in the standard PGD approach, but it can perform basic algebraic operations with parametric objects expressed in the standard separated PGD format. Here, in particular, it is able to perform all the algebraic parametric operations needed by the inverse power iteration in a transparent manner. In other words, the user only needs to prepare the

input data in the PGD separated format and perform one offline computation. As a result, the smallest (*fundamental*) eigenvalue and corresponding eigenmode are obtained in the parametric format. If the structure is unconstrained, the structure undergoes rigid body motions and the IPM cannot be applied because of the stiffness matrix singularity. To overcome this issue, first the parametric rigid body modes are computed as detailed in [11], where the PGD method was coupled with the Inertia Relief (IR) method to perform the static analysis of an unconstrained structure. Here, in addition, a Lagrange multipliers technique is presented which imposes orthogonality to the set of rigid body modes and computes the first non-zero eigenvalue. The same technique can be used to compute the subsequent non-zero eigenvalues and the corresponding eigenmodes. Another important challenge in PGD applications is represented by complex geometric parametrisations. In fact, if geometric parameters are introduced in the problem, it is not trivial to find separable representation of the input quantities. Several works have been presented in order to solve this issue [39–44]. In the above-mentioned work [11], the authors proposed also an algebraic approach which consists of a sampling of the input data such that they can be expressed in a separated format even when geometric parameters are considered. By using this algebraic approach together with the Encapsulated-PGD toolbox, the method preserves a nonintrusive implementation. This means that, unlike the standard PGD method, it does not require to access and modify the FE source code. As a consequence, the method can easily interact with commercial FE packages allowing its wider application in industry.

The paper is organised as follows. Section 2 reviews the formulation of the modal analysis in the standard non-parametric framework. In particular, the IPM is introduced as the eigensolver to be coupled with the PGD method. The description of the parametric modal analysis is given in Section 3. In 3.2, the implementation of a new algorithm able to calculate the square root of a parametric quantity is introduced and used in the parametric eigensolver. Two numerical examples are proposed in Section 4 in order to show the potential of the proposed method. It is worth noting that the same FE models were tested under unconstrained static conditions in [11]. In the first example, the modal analysis of a simple linear elastic three-dimensional (3D) structure with one material and one geometric parameter is considered to underline the main properties of the developed ROM. In the second example, an industrial application is considered, that is the modal analysis of a dummy car. Here, the nonintrusive interaction with the commercial FE software MSC-Nastran is demonstrated. A multi-objective optimisation study is also performed to prove the potential of the method as a fast and reliable tool to support designers during the decision-making procedure. Finally, conclusions and outlooks are summarised in Section 5.

2. Problem statement: finite element discretisation and modal analysis

In the absence of volumetric forces, the unknown displacement field $\boldsymbol{u}(\boldsymbol{x}, t)$ is to be computed for $\boldsymbol{x} \in \Omega$ and $t \in (0, T]$ such that

$$\rho\, \ddot{\boldsymbol{u}} = \nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}), \tag{1}$$

where ρ stands for the density, $\ddot{\boldsymbol{u}} := \frac{\partial^2 u}{\partial t^2}$ is the acceleration, and $\boldsymbol{\sigma}$ denotes the Cauchy stress tensor. The differential Eq. (1) is to be complemented with boundary conditions and initial conditions. Typically, the boundary of the domain Ω, $\partial\Omega$, is partitioned into the disjoint parts Γ_D and Γ_N, where Dirichlet and Neumann boundary conditions are prescribed. In the free-free case analysed in the following, Γ_N coincides with $\partial\Omega$ and the Neumann boundary conditions are homogeneous everywhere.

The Cauchy stress tensor dependence on $\boldsymbol{u}$ is given by the constitutive law (here, the generalised Hooke's law associated with the fourth order stiffness tensor $\mathbf{C}$) and the definition of the linear strain tensor, namely

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \mathbf{C} : \boldsymbol{\varepsilon}(\boldsymbol{u}), \text{ and } \boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^{\top}\right). \tag{2}$$

Using weighted residuals in space (integrating in Ω), problem (1) is re-written in the weak form

$$\underbrace{\int_{\Omega} \rho\, \boldsymbol{v}(\boldsymbol{x}) \cdot \ddot{\boldsymbol{u}}(\boldsymbol{x}, t)\, d\Omega}_{m(\ddot{\boldsymbol{u}}, \boldsymbol{v})} + \underbrace{\int_{\Omega} \boldsymbol{\varepsilon}(\boldsymbol{v}(\boldsymbol{x})) : \boldsymbol{\sigma}(\boldsymbol{u}(\boldsymbol{x}, t))\, d\Omega}_{a(\boldsymbol{u}, \boldsymbol{v})} = 0 \tag{3}$$

for every $\boldsymbol{v}(\boldsymbol{x})$ taking values in Ω, and at very time t. The discretised versions of the bilinear forms $m(\cdot, \cdot)$ and $a(\cdot, \cdot)$ in a finite element space are the mass matrix $\mathbf{M}$ and the stiffness matrix $\mathbf{K}$, respectively. Similarly, the discrete

version of the unknown $\boldsymbol{u}(\boldsymbol{x}, t)$ is the vector of time-dependent nodal values $\mathbf{U}(t) \in \mathbb{R}^{n_{\text{dof}}}$. Thus, the semi-discrete counterpart of Eq. (3) is the following linear system of ordinary differential equations

$$\mathbf{M}\ddot{\mathbf{U}}(t) + \mathbf{K}\mathbf{U}(t) = \mathbf{0}. \tag{4}$$

2.1. Modal analysis

The modal analysis solution of Eq. (4) assumes that the time dependence of $\mathbf{U}(t)$ is harmonic, that is

$$\mathbf{U}(t) = \cos(\omega t - \alpha)\boldsymbol{\phi}, \tag{5}$$

where ω is the *angular velocity* and α the phase angle. Note that the frequency of the mode is $\frac{\omega}{2\pi}$. Hence

$$\ddot{\mathbf{U}}(t) = -\omega^2 \cos(\omega t - \alpha)\boldsymbol{\phi} = -\omega^2 \mathbf{U}(t) \tag{6}$$

and therefore, replacing Eq. (6) in Eq. (4), it turns out that

$$\mathbf{K}\boldsymbol{\phi} = \omega^2 \mathbf{M}\boldsymbol{\phi}. \tag{7}$$

Expression (7) is an eigenvalue problem. Since $\mathbf{M}$ is symmetric positive-definite and $\mathbf{K}$ is symmetric semi-positive, the solution of Eq. (7) provides eigenvalues ω_i^2 and their corresponding eigenvectors $\boldsymbol{\phi}_i$, for $i = 1, 2, \ldots, n_{\text{dof}}$ (it is assumed they are sorted such that $\omega_1 \leq \omega_2 \leq \cdots \leq \omega_{n_{\text{dof}}}$).

Eigenmodes $\boldsymbol{\phi}_i$, for $i = 1, 2, \ldots, n_{\text{dof}}$, associated with eigenfrequencies ω_i, are the natural mode shapes, representing the deformation of the structure as vibrating in its ith mode. Modes associated with different frequencies are *orthogonal* with respect to both $\mathbf{M}$ and $\mathbf{K}$. That is $\boldsymbol{\phi}_i^\top \mathbf{M}\boldsymbol{\phi}_j = 0$ and $\boldsymbol{\phi}_i^\top \mathbf{K}\boldsymbol{\phi}_j = 0$ for i and j such that $\omega_i \neq \omega_j$.

Note that once the eigenmode $\boldsymbol{\phi}_i$ is available, the corresponding eigenfrequency is readily computed invoking the Rayleigh quotient:

$$\omega_i^2 = \frac{\boldsymbol{\phi}_i^\top \mathbf{K}\boldsymbol{\phi}_i}{\boldsymbol{\phi}_i^\top \mathbf{M}\boldsymbol{\phi}_i}. \tag{8}$$

In order to enforce unicity (up to their sign) of the eigenmodes, they are normalised with respect to the metric provided by $\mathbf{M}$. That is, $\boldsymbol{\phi}_i$ are selected such that $\boldsymbol{\phi}_i^\top \mathbf{M}\boldsymbol{\phi}_i = 1$, and consequently $\boldsymbol{\phi}_i^\top \mathbf{K}\boldsymbol{\phi}_i = \omega_i^2$, for $i = 1, 2, \ldots, n_{\text{dof}}$.

The solution $\mathbf{U}(t)$ of Eq. (4) is recovered as a linear combination of the modes $\boldsymbol{\phi}_i$ multiplied by their time harmonic dependence, $\cos(\omega_i t - \alpha_i)$, and their corresponding amplitude. The amplitude and the phase α_i of each mode are to be computed using the initial conditions.

Often, structural engineering analysis does not require obtaining the full time dependence. The fundamental modes corresponding to the lowest eigenfrequencies provide the essential information to assess the structural dynamics response. Thus, obtaining the lowest eigenfrequencies ω_i and their corresponding eigenmodes is a pertinent outcome for engineering analysis.

2.2. Numerical eigenvalue solver: the inverse power method

Several numerical methods are available in the literature [45–47] in order to solve the eigenvalue problem (7). Depending on the mathematical structure, the number of eigenpairs of interest and the computational cost associated to the algebraic operations, a different numerical eigensolver might be recommended. Roughly speaking, two main categories can be identified: *global* approaches, such as the QR method [48], that approximates all the eigenvalues, or *partial* methods which compute a smaller set of eigenvalues, such as Lanczos [49], Arnoldi [50], Davidson [51], and Jacobi–Davidson [52] methods. The simplest eigensolvers are the well-known power methods, that aim at computing the largest eigenvalue and its corresponding eigenvector (direct power method) or the lowest (inverse power method). After computing the largest (resp. lowest) eigenvalue, a deflation technique is used to remove it from the problem and the same method provides the second largest (resp. lowest). In this work we are interested only in a few lowest eigenvalues, so the IPM represents a suitable choice. Moreover, the generalisation to the parametric problem introduced in the next section suggests adopting the algorithmically simplest methodology. Assuming $\omega_1^2 > 0$, the IPM iterates approximations to eigenvector $\boldsymbol{\phi}_1$. An initial guess $\boldsymbol{\phi}_1^0$ is selected. Then, the ν-th iteration $\boldsymbol{\phi}_1^\nu$ is obtained solving the following linear system

$$\mathbf{K}\boldsymbol{\phi}_1^\nu = \mathbf{M}\boldsymbol{\phi}_1^{\nu-1}, \quad \text{for } \nu = 1, 2, 3 \ldots \tag{9}$$

The iterative sequence is expected to converge to $\boldsymbol{\phi}_1$ and the corresponding eigenvalue ω_1^2 is obtaining computing the Rayleigh quotient, as indicated in Eq. (8). Note also that the obtained value of $\boldsymbol{\phi}_1$ is to be normalised by dividing by its $\mathbf{M}$-norm, that is dividing by $\sqrt{\mathbf{M}^\top \boldsymbol{\phi}_1 \mathbf{M}}$.

The subsequent eigenvectors and eigenvalues are computed using a deflation technique, which consists of removing the already computed eigenvectors from the original matrix. In practice, in a power method this can also be done at the level of the iteration (9), enforcing orthogonality to the already computed eigenvectors. Thus, the second smallest eigenvalue ω_2^2 and its corresponding eigenvector $\boldsymbol{\phi}_2$ are computed similarly, but enforcing $\mathbf{M}$-orthogonality to $\boldsymbol{\phi}_1$ (which is already available) at every iteration, that is $\boldsymbol{\phi}_1^\top \mathbf{M} \boldsymbol{\phi}_2^\nu = 0$. In the standard algorithms, this is readily done subtracting the projection of the iterated approximation, for example by means of the Gram–Schmidt-type orthogonalisation processes (see [47]), that is replacing $\boldsymbol{\phi}_2^\nu$ by

$$\boldsymbol{\phi}_2^\nu - (\boldsymbol{\phi}_1^\top \mathbf{M} \boldsymbol{\phi}_2^\nu)\boldsymbol{\phi}_1. \tag{10}$$

Here, for the sake of easing the generalisation to the parametric case analysed in the next section, an equivalent strategy is adopted, enforcing orthogonality already in the linear system to be solved at each iteration using Lagrange multipliers. Accordingly, Eq. (9) is to be replaced by a new $(n_{\text{dof}} + 1) \times (n_{\text{dof}} + 1)$ system of equations

$$\begin{bmatrix} \mathbf{K} & \mathbf{M}\boldsymbol{\phi}_1 \\ (\mathbf{M}\boldsymbol{\phi}_1)^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_2^\nu \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{M}\boldsymbol{\phi}_2^{\nu-1} \\ 0 \end{bmatrix}, \tag{11}$$

where λ is the Lagrange multiplier, which is an instrumental unknown to be discarded as part of the result.

When the first n eigenvalues $\omega_1^2, \omega_2^2, \ldots, \omega_n^2$ are already computed and the corresponding eigenvectors are collected in the $n_{\text{dof}} \times n$ matrix $\boldsymbol{\Phi}_n = \left[\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_n\right]$, the iterative scheme to compute $\boldsymbol{\phi}_{n+1}$ boils down to solve the following $(n_{\text{dof}} + n) \times (n_{\text{dof}} + n)$ system of equations at each iteration

$$\begin{bmatrix} \mathbf{K} & \mathbf{M}\boldsymbol{\Phi}_n \\ (\mathbf{M}\boldsymbol{\Phi}_n)^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{n+1}^\nu \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\boldsymbol{\phi}_{n+1}^{\nu-1} \\ \mathbf{0} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{\lambda}$ is the $n \times 1$ vector of Lagrange multipliers. An important case to take into account, which is typical in the automotive and aerospace applications, is the case of a free-free structure (with no loads and no constrains). The stiffness matrix $\mathbf{K}$ associated to this kind of systems is singular, with a six-dimensional kernel (in 3D) containing the rigid-body modes. This is to say $\omega_i^2 = 0$ for $i = 1, 2, \ldots, 6$ and the corresponding rigid-body modes are precisely $\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_6$, collected in the $n_{\text{dof}} \times 6$ matrix $\boldsymbol{\Phi}_6$.

The computation of the 6 rigid-body modes is to be performed using a different technique concerning only matrix $\mathbf{K}$ (therefore, independent of $\mathbf{M}$) that is described in detail in [11], both for the standard implementation and the parametric one.

Once the rigid-body modes $\boldsymbol{\Phi}_6$ are obtained, the fundamental eigenfrequency and eigenvector, ω_7^2 and $\boldsymbol{\phi}_7$ are computed using the iterative scheme described in Eq. (12). This is summarised in Algorithm 1 for a general eigenvalue ω_{n+1}^2 and eigenvector $\boldsymbol{\phi}_{n+1}$, assuming that the previous ones are available in $\boldsymbol{\Phi}_n$.

3. Parametric modal analysis

As mentioned in the introduction, the goal of this work is to solve the parametric version of the problem stated in Section 2, and more precisely of the eigenvalue problem (7) arising from the modal analysis. The input data of the problem is assumed to depend on a set of n_p parameters $\boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_{n_p}]^T \in \boldsymbol{\mathcal{M}} \subset \mathbb{R}^{n_p}$ describing the material properties (e.g. elastic modulus, density, etc.) and the geometric characterisation of the shape of the structure. Typically, the multidimensional parametric domain $\boldsymbol{\mathcal{M}}$ is defined as the Cartesian product of sectional intervals, for each one of the parameters, namely $\boldsymbol{\mathcal{M}} := \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_{n_p}$, with $\mu_j \in \mathcal{M}_j \subset \mathbb{R}$ for $j = 1, \ldots, n_p$. Based on this assumption, the parametric version of the modal analysis results in solving the eigenvalue problem (7) for parameter-dependent input matrices, that is $\mathbf{K}(\boldsymbol{\mu})$ and $\mathbf{M}(\boldsymbol{\mu})$. The solution also depends on the design parameters and it is given in terms of eigenvalues $\omega_i^2(\boldsymbol{\mu})$ and eigenvectors $\boldsymbol{\phi}_i(\boldsymbol{\mu})$ for $i = 1, 2, \ldots, n_{\text{dof}}$.

3.1. Inverse power iteration and deflation with the encapsulated PGD toolbox

A coupling of the proper generalised decomposition (PGD) method with the inverse power method (IPM) is proposed to solve the parametric eigenvalue problem. The final goal is being able to compute a set of n_{eig} smallest (in

Algorithm 1 Inverse power method to compute $\boldsymbol{\phi}_{n+1}$ and ω_{n+1}

Input: $\mathbf{K}$, $\mathbf{M}$, $\boldsymbol{\Phi}_n$, tolerance ε and initial guess $\boldsymbol{\phi}^{\texttt{old}}$ (for $\nu = 0$)

1: Normalise $\boldsymbol{\phi}^{\texttt{old}} \leftarrow \boldsymbol{\phi}^{\texttt{old}} / \sqrt{\boldsymbol{\phi}^{\texttt{old}\top} \mathbf{M} \boldsymbol{\phi}^{\texttt{old}}}$

2: **while** $E_\phi > \varepsilon \, \|\boldsymbol{\phi}^{\texttt{new}}\|$ **do**

3: Solve $\begin{bmatrix} \mathbf{K} & \mathbf{M}\boldsymbol{\Phi}_n \\ (\mathbf{M}\boldsymbol{\Phi}_n)^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}^{\texttt{new}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{M}\boldsymbol{\phi}^{\texttt{old}} \\ \mathbf{0} \end{bmatrix}$

4: Normalise $\boldsymbol{\phi}^{\texttt{new}} \leftarrow \boldsymbol{\phi}^{\texttt{new}} / \sqrt{\boldsymbol{\phi}^{\texttt{new}\top} \mathbf{M} \boldsymbol{\phi}^{\texttt{new}}}$

5: Compute errors $E_\phi = \|\boldsymbol{\phi}^{\texttt{new}} - \boldsymbol{\phi}^{\texttt{old}}\|$

6: Update $\boldsymbol{\phi}^{\texttt{old}} \leftarrow \boldsymbol{\phi}^{\texttt{new}}$

7: Store solution $\boldsymbol{\phi}_{n+1} = \boldsymbol{\phi}^{\texttt{new}}$

8: Compute $\omega_{n+1} = \sqrt{\boldsymbol{\phi}_{n+1}^\top \mathbf{K} \boldsymbol{\phi}_{n+1}}$

Output: $\boldsymbol{\phi}_{n+1}$ & ω_{n+1}

magnitude) non-zero eigenvalues and the corresponding eigenmodes, both in a parametric format. From a conceptual point of view, the extension of the IPM algorithm from the non-parametric to the parametric framework is as simple as rewriting all quantities in Algorithm 1 with their parametric dependency. However, in a numerical sense, the algorithm requires several parametric algebraic operations to be performed, which is certainly a challenging task. The recently developed Encapsulated PGD Toolbox, see [38], plays a key role in this work. The toolbox contains a collection of PGD-based algorithms able to perform algebraic operations between parametric objects, such as scalars, vectors and matrices depending on the parameters $\boldsymbol{\mu}$. This allows the introduction of a general methodology to solve all the parametric operations involved in the IPM algorithm, that is: (1) solving a linear system of equations (line 3 in Algorithm 1); (2) divide a vector by a scalar (lines 1 and 4), vector–matrix–vector product (lines 1, 4 and 8); and (3) compute the square root of a scalar (parameter-dependent) quantity (lines 1, 4 and 8). All operations but the square root were already available in the toolbox. The square root routine was developed in the context of this work and added to the toolbox, following the algorithmic structure described in Section 3.2.

According to the standard PGD philosophy, the encapsulated toolbox assumes a separated representation of the multidimensional solution, which is obtained using a greedy-type algorithm combined with an alternated directions scheme, as deeply described in [38] and [53]. As a consequence, the PGD approximation of the n-th eigenmode $\phi_n(\boldsymbol{\mu})$ can be expressed as the linear combination of an a-priori unknown number of terms $n_{\boldsymbol{\phi}_n}$, namely

$$\boldsymbol{\phi}_n(\boldsymbol{\mu}) \approx \boldsymbol{\phi}_n^{\texttt{PGD}}(\boldsymbol{\mu}) = \sum_{i=1}^{n_{\boldsymbol{\phi}_n}} \beta_{\boldsymbol{\phi}_n}^i \, \boldsymbol{\phi}_n^i \prod_{j=1}^{n_p} F_{\boldsymbol{\phi}_n}^{j,i}(\mu_j). \tag{13}$$

Every i-th term of this sum is given by the product of a series of functions, each one depending separately on one of the problem parameters. More precisely, $\boldsymbol{\phi}_n^i$ refers to the spatial dimension and $F_{\boldsymbol{\phi}_n}^{j,i}(\mu_j)$ corresponds to the set of parametric functions depending separately on each parameter μ_j, for $j = 1, 2, \ldots, n_p$. If spatial and parametric terms are normalised, a factor or amplitude $\beta_{\boldsymbol{\phi}_n}^i$ appears, that indicates the relevance of the i-th term of the sum to the final separated solution. Analogously, the PGD approximation of the angular frequency $\omega_n(\boldsymbol{\mu})$ reads

$$\omega_n(\boldsymbol{\mu}) \approx \omega_n^{\texttt{PGD}}(\boldsymbol{\mu}) = \sum_{i=1}^{n_{\omega_n}} \beta_{\omega_n}^i \prod_{j=1}^{n_p} F_{\omega_n}^{j,i}(\mu_j). \tag{14}$$

Note that, being ω_n a scalar, its spatial part becomes β_{ω_n}. In order to perform parametric operations by means of the encapsulated toolbox, the input data of the routines must also be provided in the separated form. In this case,

a pre-process step is necessary to define the parametric stiffness and mass matrices as

$$\mathbf{K}(\boldsymbol{\mu}) = \sum_{i=1}^{n_{\mathbf{K}}} \mathbf{K}^i \prod_{j=1}^{n_p} F_{\mathbf{K}}^{j,i}(\mu_j) \quad \text{and} \quad \mathbf{M}(\boldsymbol{\mu}) = \sum_{i=1}^{n_{\mathbf{M}}} \mathbf{M}^i \prod_{j=1}^{n_p} F_{\mathbf{M}}^{j,i}(\mu_j). \tag{15}$$

If an exact analytical separated expression exists for the two matrices, the amplitudes do not appear in (15) because no normalisation of the PGD terms is needed. It is important to underline that it is not always trivial to find a separated analytical representation of the input data, especially when geometric parameters are considered in the problem. Furthermore, assuming that a separated expression can be found, standard a-priori PGD methods require a modification of the weak form of the problem. Such an "intrusive" approach is practically not employable in the industrial context, where commercial FE software is used, i.e. source codes are typically not accessible. For this reason, a *hybrid* and *nonintrusive* algebraic approach was proposed by the authors in a previous work [11], which is able to deal with general shape parametrisations and allows an easy interaction with commercial software. The main idea is to add a pre-process step to the method. From the computational point of view, this process consists in assembling (without solving the problem) the input matrices for every possible combination of the parametric values in the discretised domain and then expressing them in the separated format. If each introduced parametric dimension $\mu_j \in \mathcal{M}_j$ for $j = 1, 2, \ldots, n_p$ is discretised using m_j nodal values, then the full-order sampling of the parametric matrices consists of evaluating $\mathbf{K}(\boldsymbol{\mu})$ and $\mathbf{M}(\boldsymbol{\mu})$ in the set of m_{tot} points used to discretise the parametric domain $\boldsymbol{\mathcal{M}} := \mathcal{M}_1 \times \mathcal{M}_2 \times \cdots \times \mathcal{M}_{n_p}$, where $m_{tot} = \prod_{j=1}^{n_p} m_j$. It is worth noting that this technique preserves efficiency, since the computational cost for the matrices assembly is small compared to the cost of a full-order dynamic simulation. For the example tested in this work, which is discretised with a coarse FE mesh relatively to real industrial models, the time needed to assemble the input matrices for one configuration is four times cheaper than solving the corresponding dynamic problem. If finer meshes are considered, the cost of the assembling is expected to be negligible compared to the full-order computation. In addition, the sampling of the matrices can be done in parallel. Once the matrices are sampled, they can be expressed as in Eq. (15). The number of PGD terms needed to describe the two matrices can be reduced to a much smaller number than m_{tot} by performing a PGD-based data compression [53]. In fact, every time a PGD operation (such as product, sum, difference) leads to a large number of terms, data compression is advisable to reduce the number of PGD terms whilst maintaining a desired level of accuracy.

If the structure is unconstrained, the parametric matrix $\boldsymbol{\Phi}_6(\boldsymbol{\mu}) = \left[\boldsymbol{\phi}_1(\boldsymbol{\mu}), \boldsymbol{\phi}_2(\boldsymbol{\mu}), \ldots, \boldsymbol{\phi}_6(\boldsymbol{\mu})\right]$ containing the first six (in 3D) rigid-body modes must be also given as separated input data to the IPM. As explained in detail in [11], it can be computed in a separated format as the kernel of the stiffness matrix. These rigid-body modes depend on the design variables only if geometric parameters are involved in the problem because, by definition, the rigid body modes of a structure do not depend on the material properties.

Once the initial input of the problem is available, a "cascade" application of the PGD method can be performed, in the sense that the output of each parametric algebraic operation, obtained by calling an encapsulated PGD routine, can be directly used as the input of the next operation, until the global solution is obtained. The parametric input data are computed as separated multidimensional tensors, which we indicate with the superscript $\square^{\mathtt{PGD}}$. An overloading of the arithmetic operators allows to use the standard Matlab symbols to call the algebraic operations contained in the encapsulated library, which makes the method highly user-friendly. Table 1 summarises some of the algebraic operations available in the Encapsulated PGD library and the corresponding Matlab symbols.

Fig. 1 shows a pseudo-code describing the algorithmic aspects of the method when the first non-zero n_{eig} smallest eigenpairs of an unconstrained structures are sought.

Analogously to Eq. (12), the parametric IPM consists of iteratively solving the following system of equations:

$$\begin{bmatrix} \mathbf{K}(\boldsymbol{\mu}) & \mathbf{M}(\boldsymbol{\mu})\boldsymbol{\Phi}_n(\boldsymbol{\mu}) \\ (\mathbf{M}(\boldsymbol{\mu})\boldsymbol{\Phi}_n(\boldsymbol{\mu}))^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\phi}_{n+1}(\boldsymbol{\mu}) \\ \boldsymbol{\lambda}(\boldsymbol{\mu}) \end{bmatrix} = \begin{bmatrix} \mathbf{M}(\boldsymbol{\mu})\boldsymbol{\phi}_{n+1}(\boldsymbol{\mu}) \\ \mathbf{0} \end{bmatrix}. \tag{16}$$

If the first non-zero eigenmode $\boldsymbol{\phi}_7(\boldsymbol{\mu})$ is sought, then the matrix of already known mode shapes coincides with the matrix of rigid body modes, i.e. $\boldsymbol{\Phi}_n(\boldsymbol{\mu}) = \boldsymbol{\Phi}_6(\boldsymbol{\mu})$. For every new computed eigenmode $\boldsymbol{\phi}_{n+1}$, the matrix is updated as $\boldsymbol{\Phi}_n(\boldsymbol{\mu}) = [\boldsymbol{\Phi}_n(\boldsymbol{\mu}), \boldsymbol{\phi}_{n+1}]$. The Lagrange multipliers $\boldsymbol{\lambda}(\boldsymbol{\mu})$ are introduced in order to get unicity by enforcing mass-orthogonality to the set of already known eigenvectors. It is important to point out that, according to the author's knowledge, this deflation technique has not been proposed before in the context of the standard IPM. This is due to the fact that in a non-parametric context other more efficient methods can be used, as discussed

Table 1
Some of the algebraic operations available in the encapsulated PGD toolbox. Matlab symbols can be used in a standard way.

Matlab symbol	Encapsulate PGD routine
$\mathbf{K}^{\texttt{PGD}} \backslash \boldsymbol{f}^{\texttt{PGD}}$	Solve linear system
$\boldsymbol{u}^{\texttt{PGD}} + \boldsymbol{v}^{\texttt{PGD}}$	Sum (or difference)
$\mathbf{K}^{\texttt{PGD}} * \boldsymbol{u}^{\texttt{PGD}}$	Product
$\boldsymbol{u}^{\texttt{PGD}} ./ \boldsymbol{v}^{\texttt{PGD}}$	Division
$(\boldsymbol{u}^{\texttt{PGD}})'$	Transpose
sqrt($\boldsymbol{u}^{\texttt{PGD}}$)	Square root
$\sim \boldsymbol{u}^{\texttt{PGD}}$	Compression
$[\boldsymbol{u}^{\texttt{PGD}}, \boldsymbol{v}^{\texttt{PGD}}]$ and $[\boldsymbol{u}^{\texttt{PGD}}; \boldsymbol{v}^{\texttt{PGD}}]$	Arrays concatenation

in Section 2.1. Nevertheless, in this novel extension of the eigenvalue problem to the parametric framework, the Lagrange multipliers technique proved to be the most efficient. Alternative strategies were also tested. For example, Felippa et al. [54] proposed a modification of the stiffness matrix as $\mathbf{K} + \boldsymbol{\Phi}_6 \boldsymbol{\Phi}_6^\top$, whose eigenvalues are identical to those of $\mathbf{K}$ but the eigenvalues associated to the rigid body modes are raised to unity. This overcomes the stiffness singularity but leads to full matrices, which is not advisable, especially in the parametric format. Furthermore, extra operations should be performed in order to normalise the obtained eigenvectors at every iteration, e.g. by means of the Gram–Schmidt orthogonalisation. The Lagrange multipliers technique proposed here solves the singularity issue and compute an orthonormal set of eigenvectors at the same time, reducing to the minimum the number of algebraic operations needed in the parametric case.

As it is shown in the pseudo-code, a PGD guess vector $\boldsymbol{\phi}^{\texttt{old}}$ needs to be prepared every time a new eigenmode is sought. Then the extended system in Eq. (16) is iteratively solved by calling the corresponding encapsulated routine until convergence is reached, that is when a quantity $E_{\boldsymbol{\phi}}$ is smaller than a user-defined tolerance. Here $E_{\boldsymbol{\phi}}$ is defined as:

$$E_{\boldsymbol{\phi}} = \frac{\left| \sum_{i=1}^{n_{\boldsymbol{\phi}^{\texttt{new}}}} \beta^i_{\boldsymbol{\phi}^{\texttt{new}}} - \sum_{j=1}^{n_{\boldsymbol{\phi}^{\texttt{old}}}} \beta^j_{\boldsymbol{\phi}^{\texttt{old}}} \right|}{\sum_{j=1}^{n_{\boldsymbol{\phi}^{\texttt{old}}}} \beta^j_{\boldsymbol{\phi}^{\texttt{old}}}}, \tag{17}$$

where $\beta^i_{\boldsymbol{\phi}^{\texttt{new}}}$ and $\beta^j_{\boldsymbol{\phi}^{\texttt{old}}}$ represent, respectively, the amplitudes of the PGD terms describing $\boldsymbol{\phi}^{\texttt{new}}$ and $\boldsymbol{\phi}^{\texttt{old}}$ after being normalised. The normalisation is performed every time a new eigenvector $\boldsymbol{\phi}^{\texttt{new}}$ is computed, by dividing it by its $\mathbf{M}$-norm, namely $\sqrt{(\boldsymbol{\phi}^{\texttt{new,PGD}})^\top \mathbf{M}^{\texttt{PGD}} \boldsymbol{\phi}^{\texttt{new,PGD}}}$, where $sqrt$ and $*$ are the Matlab symbols used to perform the encapsulated square root and product between PGD objects. Note that, in a parametric format, the normalisation requires four algebraic operations: a product, a compression, a square root and a division.

Finally, once convergence is reached, the sought eigenvector $\boldsymbol{\phi}^{\texttt{PGD}}_{n+1}$ can be stored and the corresponding eigenvalue can be calculated according to the Rayleigh quotient $(\omega^{\texttt{PGD}}_{n+1})^2 = (\boldsymbol{\phi}^{\texttt{PGD}}_{n+1})^\top \mathbf{K}^{\texttt{PGD}} \boldsymbol{\phi}^{\texttt{PGD}}_{n+1}$. The same procedure is repeated until the desired number n_{eig} of eigenpairs is obtained.

It is worthy to emphasise that the described algorithm is solved by means of just one offline computation. The resulting eigenpairs represent the compact version of all the possible solutions for every value of the parameters. Once this offline stage is completed, the obtained *computational vademecum* can be used for optimisation studies, or it could be uploaded on light computing devices such as a tablet, where the designers could visualise in real time how the global response of the structure would change with a variation of the parameters.

3.2. PGD square root

The PGD operations necessary to code the parametric version of the algorithm are collected in Table 1. All of them but the square root were already available in the Encapsulated PGD toolbox, and the corresponding algorithms presented in [38]. This section presents the algorithm devised to implement the square root operation in the toolbox.

Recall that PGD combines a greedy strategy (computing sequentially the rank-one terms) and, for each rank-one problem, an iterative alternated directions scheme looping in all the parametric dimensions (for $\gamma = 1, 2, \ldots, n_p$) assuming that all the sectional information is known for any other sectional dimension $j \neq \gamma$. The core of the

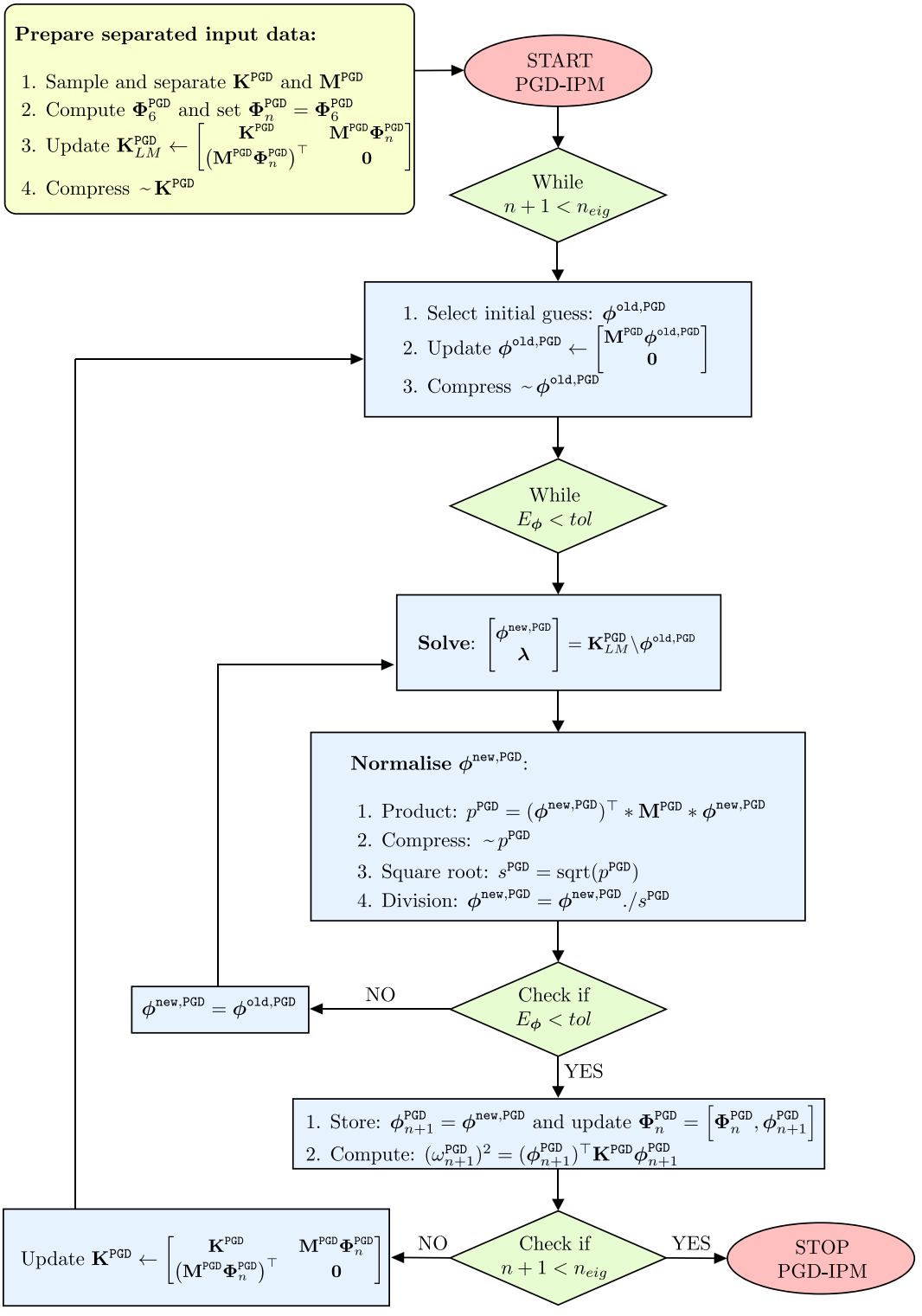

Fig. 1. Pseudo-code of the PGD-IPM method.

algorithm is summarised in the expression used to iterate (how to compute the sectional dimension γ, assuming that the rest of the sectional dimensions are known). The goal of this section is to present this expression for the square root operation. In this particular case, the expression for the first term ($i = 1$) is different than for the subsequent terms ($i = 2, 3, \dots$).

The input is a scalar parametric value $X(\boldsymbol{\mu})$ expressed in a separated format

$$X(\boldsymbol{\mu}) = \sum_{i=1}^{n_X} \beta_X^i \prod_{j=1}^{n_p} F_X^{j,i}(\mu_j), \tag{18}$$

which is assumed to be nonnegative for every value of $\boldsymbol{\mu}$. Note that, in accordance with the previous developments, the amplitude β_X^i of each term arises from a normalisation of the *parametric sectional* modes $F_X^{j,i}(\mu_j)$.

The aim is to compute $Y(\boldsymbol{\mu}) = \sqrt{X(\boldsymbol{\mu})}$, that is $Y(\boldsymbol{\mu})$, nonnegative and such that $Y(\boldsymbol{\mu})^2 = X(\boldsymbol{\mu})$. The solution is written in the form of a sum of n_Y rank-one terms, namely

$$Y(\boldsymbol{\mu}) = \sum_{i=1}^{n_Y} \beta_Y^i \prod_{j=1}^{n_p} F_Y^{j,i}(\mu_j). \tag{19}$$

First rank-one term ($i = 1$).

The standard greedy approach in the PGD consists of computing the rank-one terms sequentially. Let us denote the first rank-one term, prior to the normalisation of the sectional modes that brings out amplitudes, by

$$\tilde{Y}(\boldsymbol{\mu}) = \prod_{j=1}^{n_p} F_{\tilde{Y}}^{j}(\mu_j). \tag{20}$$

The idea is to find $\tilde{Y}$ such that

$$(\tilde{Y})^2 = X. \tag{21}$$

An alternated directions strategy is adopted that consists of computing the sectional mode $F_{\tilde{Y}}^{\gamma}(\mu_\gamma)$ assuming that the rest of the modes $F_{\tilde{Y}}^{j}(\mu_j)$ are known for $j \neq \gamma$, and looping for $\gamma = 1, 2, \dots, n_p$. Given γ, the problem is solved multiplying by a weighting function

$$\delta\tilde{Y}(\boldsymbol{\mu}) = \delta F^{\gamma}(\mu_\gamma) \prod_{j \neq \gamma} F_{\tilde{Y}}^{j}(\mu_j) \tag{22}$$

and integrating in all parametric dimensions but γ, that is

$$\int \cdots \int_{\mu_j \neq \mu_\gamma} \left(\tilde{Y}\right)^2 \delta\tilde{Y}(\boldsymbol{\mu}) \prod_{j \neq \gamma} d\mu_j = \int \cdots \int_{\mu_j \neq \mu_\gamma} X \delta\tilde{Y}(\boldsymbol{\mu}) \prod_{j \neq \gamma} d\mu_j \tag{23}$$

Using expressions (20) and (22) in (23) results in

$$\underbrace{\left[\prod_{j \neq \gamma} \int_{\mu_j} (F_{\tilde{Y}}^{j}(\mu_j))^3 d\mu_j\right]}_{=:\beta_\star} (F_{\tilde{Y}}^{\gamma}(\mu_\gamma))^2 \delta F^{\gamma}(\mu_\gamma) = \underbrace{\sum_{i=1}^{n_X} \beta_X^i \left[\prod_{j \neq \gamma} \int_{\mu_j} F_X^{j,i}(\mu_j) F_{\tilde{Y}}^{j}(\mu_j) d\mu_j\right] F_X^{i,\gamma}(\mu_\gamma)}_{=:R(\mu_\gamma)} \delta F^{\gamma}(\mu_\gamma) \tag{24}$$

for all $\delta F^{\gamma}(\mu_\gamma)$. Note that scalar $\beta_\star$ and function $R(\mu_\gamma)$ introduced in (24) are computable at this stage of the alternated directions algorithm. Thus, the resulting sectional mode $F_{\tilde{Y}}^{\gamma}(\mu_\gamma)$ is updated in this iteration using the following expression:

$$F_{\tilde{Y}}^{\gamma}(\mu_\gamma) = \sqrt{R(\mu_\gamma)/\beta_\star}, \tag{25}$$

which defines the core of the PGD square root algorithm (for the first term, $i = 1$).

Subsequent rank-one terms ($i = 2, 3, \dots$).

Now, assume that expression (19) for Y is obtained for some n_Y and it has to be enhanced adding the $n_Y + 1$ term. This rank-one term is denoted, prior to normalisation of the sectional contributions, by

$$\Delta Y(\boldsymbol{\mu}) = \prod_{j=1}^{n_p} F^j_{\Delta Y}(\mu_j). \tag{26}$$

The algorithm proposed is based on the fact that in the PGD greedy algorithm, the first modes produce a fair approximation of $\sqrt{X(\boldsymbol{\mu})}$ and therefore, in order to compute the next term, a linearisation of the equation is sufficient to improve the approximation. The following approximated equation for ΔY results from neglecting the quadratic term $(\Delta Y)^2$ in front of $Y\Delta Y$, namely

$$(Y + \Delta Y)^2 = X \quad \text{results in} \quad Y\Delta Y \approx \frac{1}{2}(X - Y^2). \tag{27}$$

Once again, the unknowns $F^j_{\Delta Y}$, for $j = 1, 2, \ldots, n_p$ are computed with the standard alternate directions strategy, that is assuming that $F^j_{\Delta Y}$ are known for $j \neq \gamma$, multiplying by a variation δY, and integrating in all sectional dimensions but γ, analogously to (23). The variation δY reads

$$\delta Y(\boldsymbol{\mu}) = \delta F^\gamma(\mu_\gamma) \prod_{j\neq\gamma} F^j_{\Delta Y}(\mu_j). \tag{28}$$

Thus, in $F^\gamma_{\Delta Y}$ is computed such that, for all $\delta F^\gamma(\mu_\gamma)$

$$\begin{aligned}
&\sum_{i=1}^{n_Y} \beta^i_Y \overbrace{\left[\prod_{j\neq\gamma}\left[\int_{\mu_j} F^{j,i}_Y(\mu_j)\left(F^j_{\Delta Y}(\mu_j)\right)^2 d\mu_j\right]\right]}^{=:\beta^i_\star} F^\gamma_{\Delta Y}(\mu_\gamma) F^{i,\gamma}_Y(\mu_\gamma)\delta F^\gamma(\mu_\gamma) = \\
&\underbrace{\left[\int\cdots\int_{\mu_j\neq\mu_\gamma} \frac{1}{2}(X - Y^2)\left(\prod_{j\neq\gamma} F^j_{\Delta Y}(\mu_j)\, d\mu_j\right)\right]}_{=:R(\mu_\gamma)} \delta F^\gamma(\mu_\gamma)
\end{aligned} \tag{29}$$

Thus, the expression defining the core of the alternated directions algorithm for the subsequent terms $i = 2, 3, \ldots$ (and analogous to (25) for $i = 1$) reads

$$F^\gamma_{\Delta Y}(\mu_\gamma) = R(\mu_\gamma) / \left(\sum_{i=1}^{n_Y} \beta^i_Y \beta^i_\star F^{i,\gamma}_Y(\mu_\gamma)\right). \tag{30}$$

4. Numerical examples

Two numerical examples are presented in this section to show the potential of the proposed PGD-IPM method. In the first example, the method is used to analyse the dynamic properties of a 3D solid structure characterised by both material and geometric parameters. In the second example, a more realistic industrial case is investigated, which concerns the dynamic response of a dummy car. Also, a multi-objective optimisation study is performed, which makes the PGD-IPM method a very interesting tool to be employed in the context of design optimisation problems.

4.1. Parametric inverse power method with material and geometric parameters

The modal analysis of an unconstrained linear elastic 3D structure characterised by one material and one geometric parameter is considered. The two variables, which are treated as additional coordinates of the problem, are denoted with $\mu \in \mathcal{M}_\mu$ and $\theta \in \mathcal{M}_\theta$ for the material and geometric parameters, respectively. As depicted in Fig. 2, the reference domain $\hat{\Omega}$ consists of a block with dimensions $[-L_x/2, L_x/2]\times[-L_y/2, L_y/2]\times[-L_z/2, L_z/2]$ with an inclusion given by $[-L_x/6, L_x/6]\times[-L_y/4, L_y/4]\times[-L_z/2, L_z/2]$, where $L_x = 6$, $L_y = 12$ and $L_z = 1$. The spatial discretisation, also shown in Fig. 2, consists of a regular mesh with 236 nodes and 742 linear tetrahedral elements.

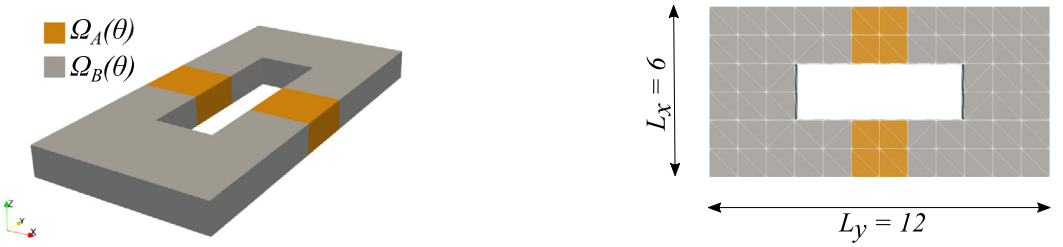

Fig. 2. Computational domain, showing the partition into two non-overlapping subdomains $\Omega_A(\theta)$ and $\Omega_B(\theta)$ (left) and top view of the discretised computational domain (right).

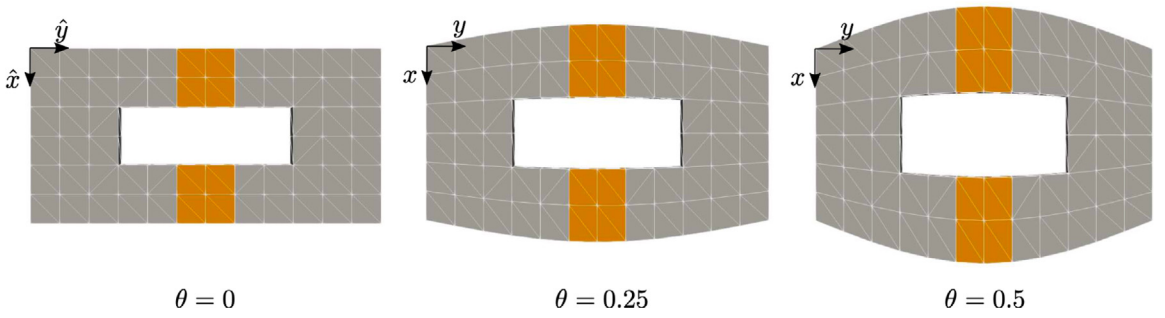

Fig. 3. Physical domain for three different values of the geometric parameter θ.

The physical domain $\Omega(\theta)$ depends upon the geometric parameter and it is split into two non-overlapping subdomains $\Omega_A(\theta)$ and $\Omega_B(\theta)$, such that the parametric Young's modulus E is defined as

$$E(\boldsymbol{x}, \mu) = \begin{cases} E_A(\mu) = \mu & \text{for} \quad \boldsymbol{x} \in \Omega_A(\theta), \\ E_B = 200 & \text{for} \quad \boldsymbol{x} \in \Omega_B(\theta), \end{cases} \tag{31}$$

where the Young modulus $E_A(\mu)$ is considered varying in the range $\mathcal{M}_\mu = [10, 410]$, and $\mathcal{M}_\mu$ is discretised with a uniform distribution of $n_\mu = 41$ points. The Poisson's ratio and the density are assumed constant in the whole domain and take values $\nu = 0.3$ and $\rho = 1$ respectively.

The geometrically parametrised domain $\Omega(\theta)$ is described with the Cartesian coordinates $\boldsymbol{x}$, and it is defined as the image of the reference domain $\hat{\Omega}$, with reference coordinates $\hat{\boldsymbol{x}}$, via a geometric mapping $\boldsymbol{\Psi}(\hat{\boldsymbol{x}}, \theta)$, namely

$$\begin{cases} x &= \psi_1(\hat{\boldsymbol{x}}, \theta) = \hat{x} + \theta \sin\left(\dfrac{\pi \hat{y}}{L_y}\right)\left(\hat{x} - \dfrac{L_x}{2}\right), \\ y &= \psi_2(\hat{\boldsymbol{x}}, \theta) = \hat{y}, \\ z &= \psi_3(\hat{\boldsymbol{x}}, \theta) = \hat{z}. \end{cases} \tag{32}$$

The parameter θ is taken to be in the interval $\mathcal{M}_\theta = [0, 0.5]$, and $\mathcal{M}_\theta$ is discretised with a uniform distribution of $n_\theta = 21$ points. The effect of the geometric parameter is depicted in Fig. 3, where the deformed computational domain for three different values of the parameter θ is presented. The particular value $\theta = 0$ leads to a deformed configuration that coincides with the reference configuration, i.e. the mapping of Eq. (32) becomes the identity.

The objective of this numerical test is to explore how changing the introduced design parameters affect the dynamic response of the structure. This can be done by performing a parametric modal analysis of the structure by means of the proposed PGD-IPM eigensolver. As explained in the previous section, the first essential step consists of defining the input data (i.e. stiffness and mass matrices) in a separated format. Thanks to the linear dependence of the stiffness matrix on the Young's modulus, an analytical separable representation of the stiffness matrix with respect to μ can be easily constructed. For the geometric parameter θ, it is not trivial to find a separated representation. For this reason, a novel algebraic technique is employed, as mentioned in Section 3 and discussed in detail in [11]. The main idea behind this algebraic technique is to generate a new geometrically deformed mesh for every nodal value

of the geometric parameter $\theta^p = [\theta^1, \theta^2, \ldots, \theta^{n_\theta}]^T$, according to the mapping of Eq. (32). For each new mesh, two stiffness-like matrices $\mathbf{K}_A(\theta^p)$ and $\mathbf{K}_B(\theta^p)$ are assembled. The quantity $\mathbf{K}_A(\theta^p)$ is calculated by imposing the Young's modulus $(E_A, E_B) = (1, 0)$, thus accounting for the contribution of the finite elements belonging to the subdomain $\Omega_A(\theta^p)$ to the global stiffness matrix. Analogously, $\mathbf{K}_B(\theta^p)$ corresponds to the choice $(E_A, E_B) = (0, 1)$ and accounts for the contribution of the finite elements belonging to the subdomain $\Omega_B(\theta^p)$. Once these matrices are sampled in the parametric nodes n_θ, a separated form of the parametric global stiffness matrix is readily available, namely

$$\mathbf{K}(\mu, \theta) = E_A(\mu) \sum_{i=1}^{n_\theta} \mathbf{K}_A^i \, k^i(\theta) + E_B \sum_{i=1}^{n_\theta} \mathbf{K}_B^i \, k^i(\theta), \tag{33}$$

with $k^i(\theta^p) = \delta_{p,i}$, for every nodal value $p = 1, 2, \ldots, n_\theta$ of the geometric parameter. If the number of PGD terms is large, it is advisable to perform data compression. This is of particular importance in the framework of the PGD-IPM as compressing the PGD terms will alleviate the cost of the subsequent algebraic operations. In this example, after performing compression with a tolerance of 10^{-5}, an accurate approximation of the stiffness matrix was obtained with only $n_\mathbf{K} = 10$ terms, instead of the initial 861 terms:

$$\mathbf{K}^{\text{PGD}}(\mu, \theta) = \sum_{i=1}^{n_\mathbf{K}} \beta_\mathbf{K}^i \, \mathbf{K}^i \; M_\mathbf{K}^i(\mu) \, G_\mathbf{K}^i(\theta). \tag{34}$$

The PGD approximation of the parametric mass matrix is obtained by following the same procedure:

$$\mathbf{M}^{\text{PGD}}(\mu, \theta) = \sum_{i=1}^{n_\mathbf{M}} \beta_\mathbf{M}^i \, \mathbf{M}^i \; M_\mathbf{M}^i(\mu) \, G_\mathbf{M}^i(\theta). \tag{35}$$

It should be mentioned that the mass matrix is actually independent on the Young modulus, that is $M_\mathbf{M}^i(\mu) = 1$. However, the general expression of Eq. (35) is used to maintain a consistent notation for all the inputs of the PGD-IPM approach. Note that, from now on, the material parametric functions will be denoted by $M(\mu)$, while $G(\theta)$ will be used for the geometric functions, omitting the subindex for a better readability.

Once the separated representation of the input data is available, the proposed method can be finally employed following the steps shown in Fig. 1. In this example, the goal is to compute the first three mode shapes $\boldsymbol{\phi}_n(\mu, \theta)$, with $n = 7, 8, 9$, corresponding to the smallest non-zero eigenvalues. For each n-th mode, the generalised PGD solution reads:

$$\boldsymbol{\phi}_n(\mu, \theta)^{\text{PGD}} = \sum_{i=1}^{N_{\boldsymbol{\phi}}} \beta^i \boldsymbol{\phi}_n^i M^i(\mu) G^i(\theta). \tag{36}$$

Fig. 4 shows the generalised results in terms of amplitude, parametric and spatial terms for the three modes $\boldsymbol{\phi}_7^{\text{PGD}}, \boldsymbol{\phi}_8^{\text{PGD}}, \boldsymbol{\phi}_9^{\text{PGD}}$. As expected, the amplitudes β^i rapidly decrease as the number of PGD terms increase. Using a tolerance of 10^{-3} to stop the enrichment of the PGD solution, a maximum number of 10 terms is needed to get an accurate representation of the three parametric mode shapes. In addition, the results show that the first four PGD terms capture the most relevant information of the generalised solution, as the fifth and subsequent terms have an amplitude at least two orders of magnitude lower than the amplitude of the first PGD term. For this reason, the first four parametric and spatial terms are also shown in Fig. 4. For each mode shape, same colours are used to depict the parametric functions and amplitudes related to the same i-term. The parametric functions seem to show a higher influence of the material parameter then the geometric one on the final response. The spatial modes provide an illustration of the deformation induced by the four most relevant terms of the generalised solution. Those illustrations already allow to identify the type of mode shapes. In fact, the first mode $\boldsymbol{\phi}_7^{\text{PGD}}$ can be identified as a flexional mode, the second $\boldsymbol{\phi}_8^{\text{PGD}}$ as torsional, while the third $\boldsymbol{\phi}_9^{\text{PGD}}$ mode shows a shear-type deformation along the longest edge of the structure.

In order to get a particularised solution for a chosen set of the parameters $(\bar{\mu}, \bar{\theta})$, the corresponding function values $M^i(\bar{\mu})$ and $G^i(\bar{\theta})$ are evaluated for each PGD-term i and then multiplied by the corresponding spatial term and amplitude of the desired mode shape. This procedure can be easily performed in a post-process step, providing real-time results for any combination of the parameters. As already known, once the parametric mode shapes are available, the corresponding three eigenvalues can be simply computed by means of the Rayleigh quotient. In order

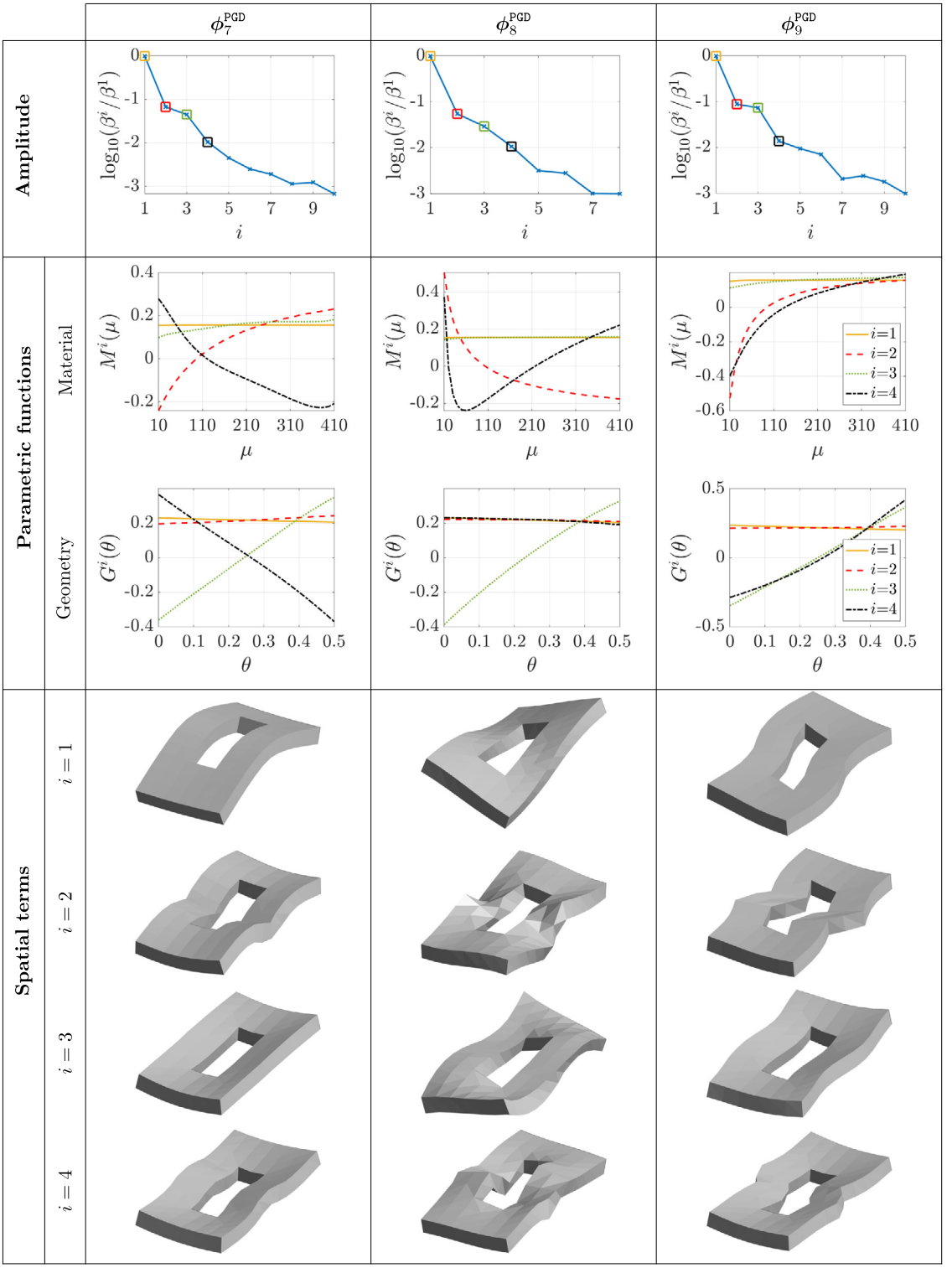

Fig. 4. PGD generalised solution for the first three shape modes $\boldsymbol{\phi}_7^{\text{PGD}}, \boldsymbol{\phi}_8^{\text{PGD}}$ and $\boldsymbol{\phi}_9^{\text{PGD}}$. For each mode, the evolution of the amplitude β^i of the PGD terms, the first four parametric functions and first four spatial terms are shown. Same colour is assigned to amplitudes values and corresponding parametric functions.

Table 2
Accuracy of the PGD results with respect to the full-order FE computations measured as the relative error between the PGD and FE solutions in the $\mathcal{L}_2$ and $\mathcal{L}_\infty$ norm according to Eqs. (37) and (38).

	$\omega_7^{PGD}(\mu,\theta)$	$\omega_8^{PGD}(\mu,\theta)$	$\omega_9^{PGD}(\mu,\theta)$
$\|\varepsilon_{\text{PGD}}\|_2$	9.18×10^{-4}	1.00×10^{-3}	1.00×10^{-3}
$\|\varepsilon_{\text{PGD}}\|_\infty$	4.20×10^{-3}	3.56×10^{-3}	1.15×10^{-1}

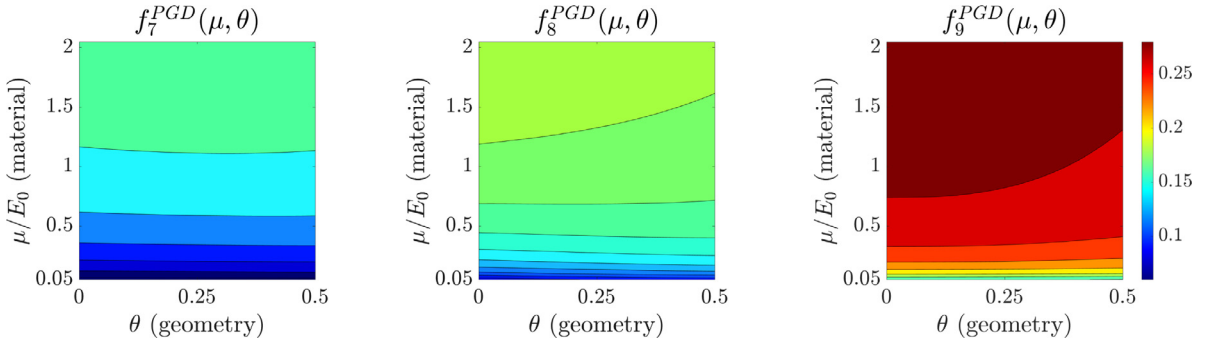

Fig. 5. Variation of the first three smallest natural frequencies f_7, f_8 and f_9 with respect to the parameters μ and θ.

to validate the PGD results, the accuracy with respect to the full-order FE computations is measured as the relative error between the PGD and FE eigenvalue solutions in the $\mathcal{L}_2(\mathcal{M}_\mu \times \mathcal{M}_\theta)$ norm, that is

$$\|\varepsilon_{\text{PGD}}\|_2 = \left(\frac{\int_{\mathcal{M}_\theta} \int_{\mathcal{M}_\mu} \left(\omega^{\text{PGD}} - \omega^{\text{FE}}\right)^2 \, d\mu \, d\theta}{\int_{\mathcal{M}_\theta} \int_{\mathcal{M}_\mu} (\omega^{\text{FE}})^2 \, d\mu \, d\theta} \right)^{1/2}. \tag{37}$$

Also the maximum error is calculated as the $\mathcal{L}_\infty(\mathcal{M}_\mu \times \mathcal{M}_\theta)$ norm:

$$\|\varepsilon_{\text{PGD}}\|_\infty = \max_{\mu \in \mathcal{M}_\mu, \theta \in \mathcal{M}_\theta} \left(\frac{\left|\omega^{\text{PGD}} - \omega^{\text{FE}}\right|}{\omega^{\text{FE}}} \right). \tag{38}$$

Table 2 reports the calculated $\mathcal{L}_2$ and $\mathcal{L}_\infty$ errors for the three computed eigenvalues, proving that a high level of accuracy can be obtained by using the proposed PGD-IPM method. It is worth noting that to compute this error measure, the problem had to be solved by means of the standard FE method for each possible combination of the parameters, that is $n_\mu \times n_\theta = 21 \times 41 = 861$ FE simulations. It is important to underline that the main goal in the PGD context is not to reduce the computational cost, but to provide a method which is able to explore an arbitrary large parametric space with only one offline computation. Another important advantage concerns the storage memory. In fact, the obtained PGD computational vademecum needs $\sim$182 KB of storage memory for the three eigenmodes versus the $\sim$14900 KB needed to store 861 full-order FE solutions for each of the three eigenmodes.

To conclude, another important property of the method is shown, which is the possibility to explore the design space and check, in real time, the effects of the design parameters on a predefined quantity of interest (QoI). In this case, the frequency associated to each mode shape is chosen as QoI, which is computed as:

$$f_n^{\text{PGD}}(\mu,\theta) = \frac{\omega_n^{\text{PGD}}(\mu,\theta)}{2\pi}. \tag{39}$$

The variation of the three frequencies f_7^{PGD}, f_8^{PGD} and f_9^{PGD} in the parametric space is depicted in Fig. 5. This results clearly show that the geometric parameters have less influence on the frequency, especially for values of μ smaller than the constant Young's modulus E_0 in the remaining domain. This conclusion could be used to make decisions during the design process.

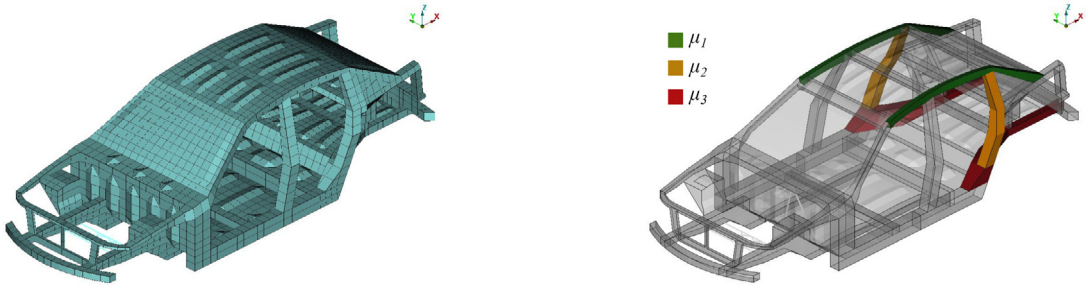

Fig. 6. Geometry and mesh properties of the BIW structure used for the dynamic global torsional stiffness analysis (left). The three car components highlighted (right) are characterised by parametric properties, that is the thickness of each one of the components.

4.2. Industrial test: parametric modal analysis of a dummy car

The PGD-IPM method is now employed to solve a more realistic problem, which is the noise, vibration and harshness analysis (NVH) of a vehicle. Finite element modal analysis is usually employed in order to predict the dynamic properties of a vehicle in terms of its natural frequencies and shape modes. At this purpose, the PGD-IPM method is proposed here to study the influence of predefined design parameters on the dynamic behaviour of the body in white (BIW) car structure, where BIW is the technical name used in the automotive industry to indicate a car body's frame when all the components have been joined together.

The geometry of the BIW and the mesh discretisation are shown in Fig. 6. Isoparametric quadrilateral shell elements are used to discretise the FE model which is formed by a total number of 3819 nodes, each one characterised by six degrees of freedom (three translations and three rotations). All the car components are characterised by a linear elastic material. In this example, the thickness of three car components highlighted in Fig. 6 are introduced as extra coordinates of the problem. The three parameters are denoted by $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3]^T$ and they vary in the intervals $\mathcal{M}_j = [0.7, 1.5]$ mm, for $j = 1, 2, 3$. Each parametric domain is discretised with $n_1 = n_2 = n_3 = 9$ equidistant nodes. Although from a physical point of view the thickness is meant as a geometric parameter, in the shell element formulation it is treated as a material property.

The proposed PGD-IPM approach is employed following the same procedure described in the previous example. An important difference with respect to the previous test is that, in this numerical example, the commercial FE package MSC-Nastran is employed, demonstrating the nonintrusive character of the proposed PGD-IPM. The software is used to assemble the parametric input matrices for each possible combination of the parameters (without solving the problem) and stored. Afterwards, the matrices can be expressed in the required separated form $\mathbf{K}^{\text{PGD}}(\boldsymbol{\mu})$ and $\mathbf{M}^{\text{PGD}}(\boldsymbol{\mu})$, as done in the previous example. Also in this example the structure is in its unconstrained configuration, so that the rigid body modes need to be computed. The goal of this numerical test is to compute the first three eigenvectors $\boldsymbol{\phi}_7^{\text{PGD}}$, $\boldsymbol{\phi}_8^{\text{PGD}}$ and $\boldsymbol{\phi}_9^{\text{PGD}}$ associated to the smallest non-zero eigenvalues by means of the PGD-IPM eigensolver.

In this example, it can be verified that the first and third eigenvectors represent two different kinds of flexional modes for every combination of the parameters, while the second eigenvector always represents a torsional mode. However, when more complex models are analysed, the order of the modes can easily change with the parameters, so their identification would represent an important task.

Since the torsional mode plays a crucial role in the characterisation of the dynamic properties of the BIW, Fig. 7 shows only the PGD generalised solution for the second shape mode $\boldsymbol{\phi}_8^{\text{PGD}}$, defined as:

$$\boldsymbol{\phi}_8^{\text{PGD}}(\mu_1, \mu_2, \mu_3) = \sum_{i=1}^{N} \beta_i \, \boldsymbol{\phi}_8^i \, M_1^i(\mu_1) \, M_2^i(\mu_2) \, M_3^i(\mu_3). \tag{40}$$

Although this example is more complex and realistic than the previous one, the amplitudes β^i decrease rapidly with the number of PGD terms and only 11 terms are needed to ensure an amplitude of 10^{-3}. Also in this case, the first four terms contain the most important information about the solution, since the fifth and subsequent terms have an amplitude of at least two order of magnitude smaller than the amplitude of the first term. The parametric

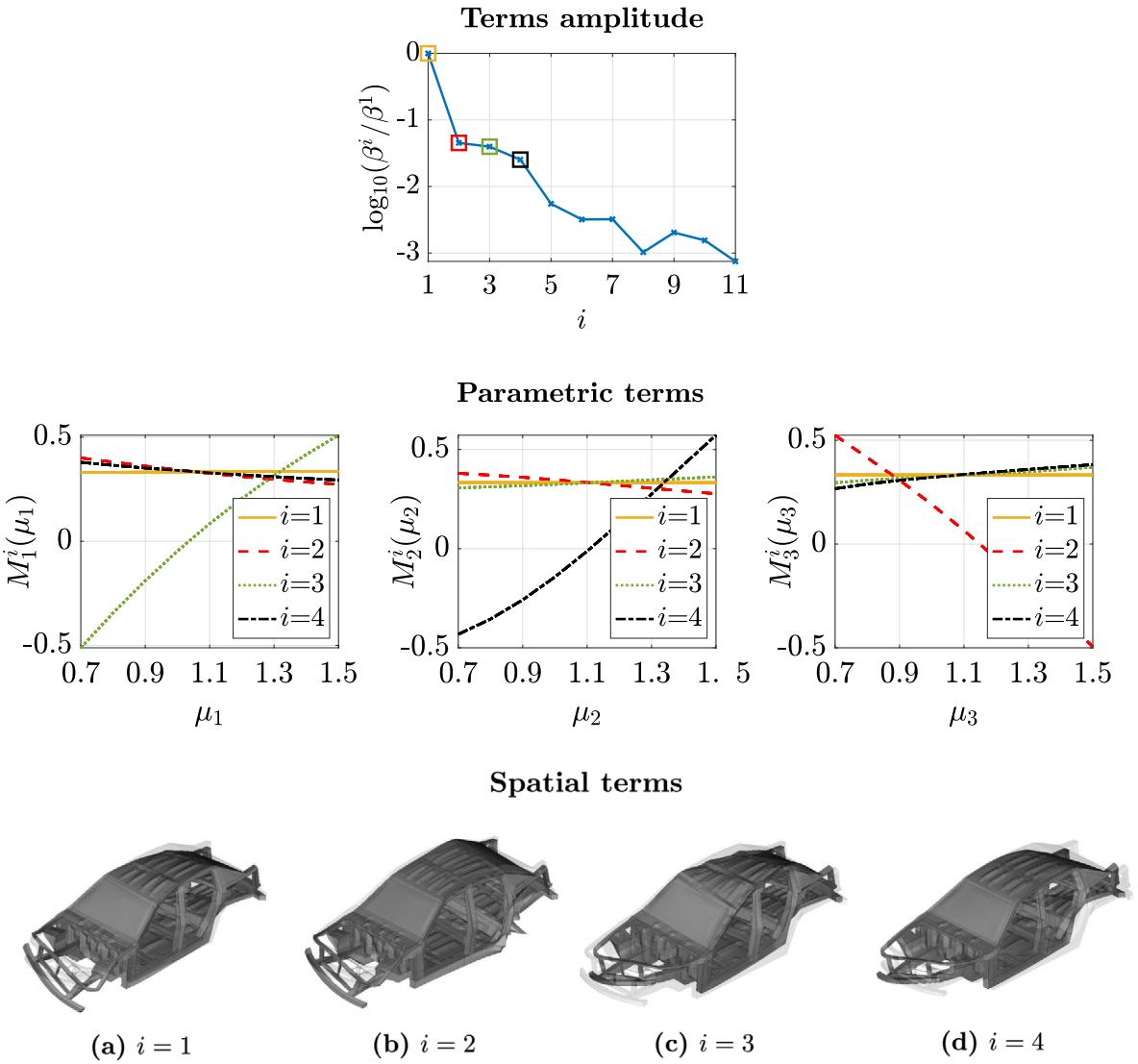

Fig. 7. PGD generalised solution for the first torsional mode $\boldsymbol{\phi}_8^{\text{PGD}}$. Evolution of the amplitude β^i of the PGD terms (top). First four parametric functions of the generalised solution (centre). First four spatial PGD terms (bottom). Same colour is assigned to amplitude values and correspondent parametric functions.

functions, shown in Fig. 7, contain the information about the influence of the parameters on the particularised eigenmode $\boldsymbol{\phi}_8^{\text{PGD}}(\mu_1, \mu_2, \mu_3)$. Finally, the spatial terms clearly show that this eigenmode represents a torsional mode shape. Fig. 8 shows the deformation and the equivalent von Mises stress field for the three eigenvectors when three different set of parameters are chosen. Also the frequencies associated to each parametric choice are indicated, which are calculated according to Eq. (39).

This post-process helps to understand which components of the BIW structure are most sensitive to changes in the parameters. In particular, for all vibrational modes, the maximum values of the von Mises stress concentrate on the windshield header. Furthermore, it is shown how small variations in the thicknesses of the three parametric components can lead to variations of the frequency in the range of about 2 Hz, which might change the perception of vibration for the occupants of the vehicle. It is important to remember that this particularised solutions are calculated in real-time, after having obtained the PGD solution by performing the offline PGD-IPM computation.

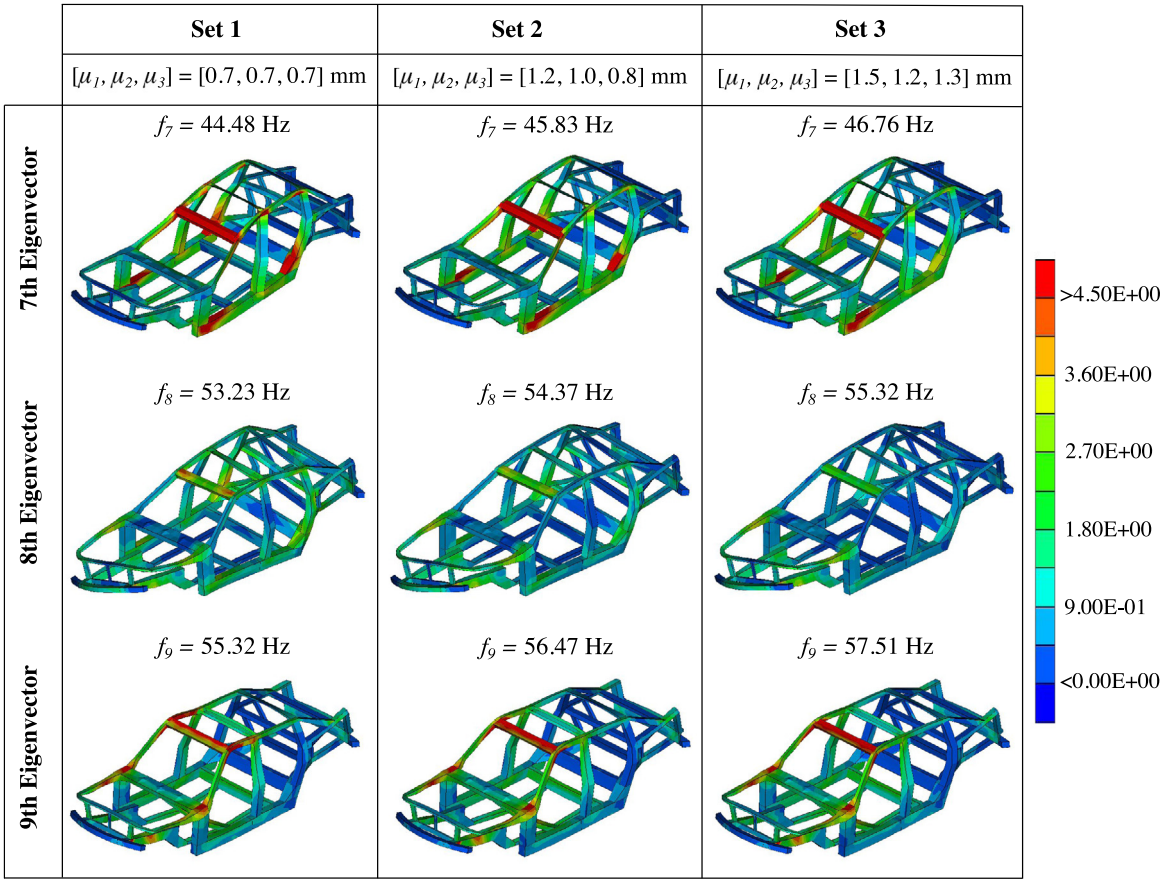

Fig. 8. Particular cases of the generalised solution, showing the von Mises stress field related to the three eigenvectors associated to the first three smallest non-zero eigenvalues, for three different sets of the parameters. The solutions are obtained in real-time after the PGD is applied to compute the spatial and parametric modes.

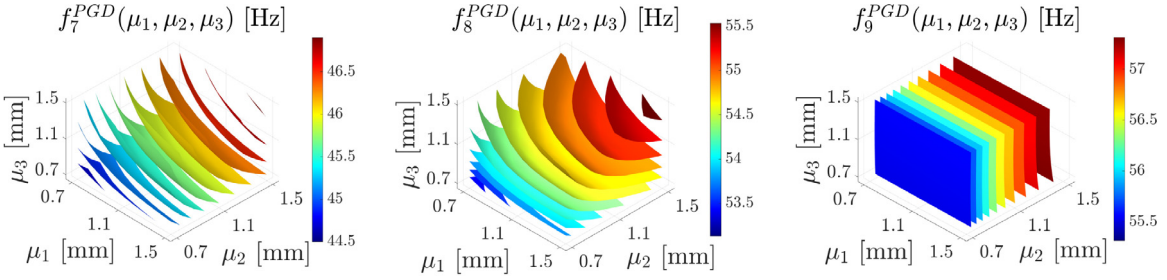

Fig. 9. Isosurfaces showing the variation of the first three smallest natural frequencies f_7, f_8 and f_9 with respect to the parameters μ_1, μ_2 and μ_3.

In order to have a better understanding of how the design parameters affect the dynamic response of the car, the variation of the three smallest natural frequencies in the parametric space is shown in Fig. 9, in terms of isosurfaces.

To conclude this numerical example, a multi-objective optimisation process is demonstrated by using the PGD-IPM results. Let us assume that the goal of the optimisation is to find the combination of parameters able to maximise the frequency associated to the torsional mode, while minimising the mass of the three car components considered

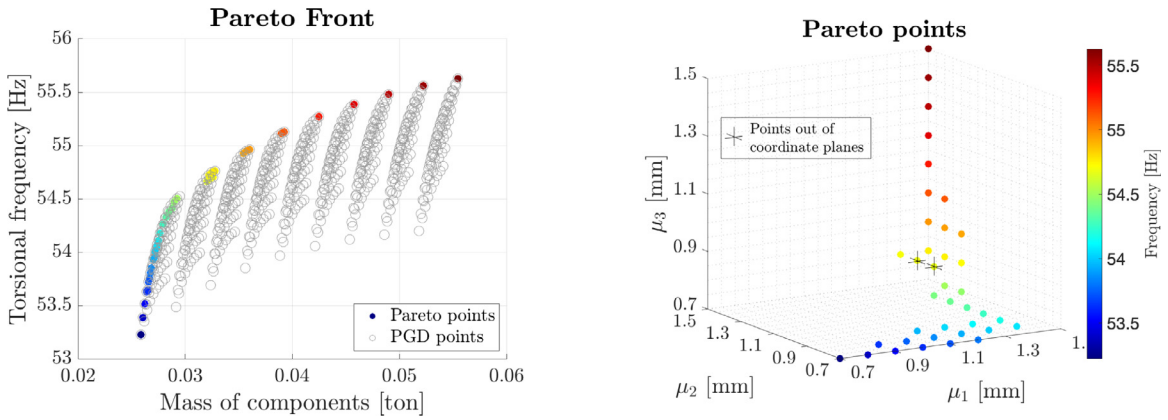

Fig. 10. Multi-objective optimisation showing the Pareto front as a function of the objectives (left) and the PGD parameters (right).

in this example. To this end, the following two objective functions are considered

$$\begin{cases} g_1(\boldsymbol{\mu}) &= \rho\,(\mu_1 A_1 + \mu_2 A_2 + \mu_3 A_3), \\ g_2(\boldsymbol{\mu}) &= f_8^{\mathtt{PGD}}(\boldsymbol{\mu}). \end{cases} \tag{41}$$

The function $g_1(\boldsymbol{\mu})$ represents the mass of the material needed to manufacture the three car components, equal to the product of the material density ρ and the parametric volume. The latter is given by the sum of the products between the car components areas (A_1, A_2, A_3) times their variable thicknesses (μ_1, μ_2, μ_3). Clearly, this quantity is strictly related to the production cost. The objective function $g_2(\boldsymbol{\mu})$ represents the parametric torsional frequency associated to $\boldsymbol{\phi}_8^{\mathtt{PGD}}$ and calculated by means of the Rayleigh quotient. Thanks to the explicit dependency of the frequency on the parameters, the optimisation problem could be directly solved by means of the `gamultiobj` function available in the Global Optimisation Toolbox released by Matlab. The function uses a genetic algorithm in order to find the Pareto front of multiple objective functions, meant as a set of optimal points in the parametric space that represent a trade-off between the objective functions. More specifically, a point is considered optimal if no objective can be improved without sacrificing at least one other objective. The results of the optimisation study are shown in Fig. 10, where the optimal Pareto front correspond to the coloured points, while the empty points represent the whole range of configurations that should be considered if all the parametric combinations are analysed. The coordinates (μ_1, μ_2, μ_3) of the optimal points are also shown in the right plot of Fig. 10. All the optimal points belong to the boundaries of the parametric domain, except two points which are indicated in the picture. In conclusion, the optimisation process allows to drastically reduce the range of solutions to be considered by a designer in the decision-making process. Note that, in this example, the Pareto front was computed by assigning the same weight to the objective functions. Nevertheless, it is straightforward to obtain other fronts if the user wants to put more emphasis on one of the objective functions.

5. Conclusion and outlook

This work proposes a nonintrusive algebraic PGD approach combined with the inverse power method (PGD-IPM) to perform the parametric modal analysis of unconstrained structures being characterised by material and geometric parameters. The developed eigensolver uses a library of PGD-based routines implemented by Díez et al. [38] to sequentially perform algebraic operations between parametric objects in a black-box format. In addition, a new algorithm was developed and added to the encapsulated library, which computes the square root of a parametric quantity. Furthermore a Lagrange multipliers deflation technique is proposed to overcome singularity issues in the case of unconstrained structures and to compute a multiple set of smallest natural frequencies and corresponding mode shapes. Two numerical examples are tested to compute the desired parametric solutions. In the first example, an academic test case with one material and one geometric parameter is proposed to show the properties of the PGD solution. The variation of the natural frequencies and mode shapes in the parametric domain is calculated with only one offline computation and the accuracy of the generalised solution is measured by comparing the PGD solution to

the whole set of standard FE full-order solutions. In the second example, a more realist industrial case is presented to show the nonintrusive interaction of the proposed method with the commercial FE package MSC-Nastran. Here, the parametric results of the modal analysis of a dummy car are presented. The solution for specific sets of parameters are computed in real-time during a post-process step at the negligible cost of a linear combination. Finally, a multi-objective optimisation study is performed, which makes the method significantly appealing for industrial applications where designers urgently need new computational tools to support the decision-making process. With a small effort, the generalised solutions generated by employing the developed technique could be uploaded on portable devices (such as tablets) such that designers could evaluate in real time the impact of certain parameters on the global response of the structure.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project is part of the Marie Skłodowska-Curie ITN-EJD ProTechTion funded by the European Union Horizon 2020 research and innovation program with Grant Number 764636. The work of Fabiola Cavaliere, Sergio Zlotnik and Pedro Díez is partially supported by the MCIN/AEI/10.13039/501100011033, Spain (Grant Number: PID2020-113463RB-C32, PID2020-113463RB-C33 and CEX2018-000797-S). Ruben Sevilla also acknowledges the support of the Engineering and Physical Sciences Research Council (Grant Number: EP/P033997/1).

References

[1] R.W. Freund, Model reduction methods based on Krylov subspaces, Acta Numer. 12 (2003) 267–319, http://dx.doi.org/10.1017/s0962492902000120.
[2] G. Rozza, D.B.P. Huynh, A.T. Patera, Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations, Arch. Comput. Methods Eng. 15 (3) (2007) 1–47, http://dx.doi.org/10.1007/bf03024948.
[3] A. Chatterjee, An introduction to the proper orthogonal decomposition, Current Sci. (2000) 808–817.
[4] B. Feeny, R. Kappagantu, On the physical interpretation of proper orthogonal modes in vibrations, J. Sound Vib. 211 (4) (1998) 607–616, http://dx.doi.org/10.1006/jsvi.1997.1386.
[5] A. Placzek, D.-M. Tran, R. Ohayon, Hybrid proper orthogonal decomposition formulation for linear structural dynamics, J. Sound Vib. 318 (4–5) (2008) 943–964, http://dx.doi.org/10.1016/j.jsv.2008.05.015.
[6] A.G. Buchan, C.C. Pain, F. Fang, I.M. Navon, A POD reduced-order model for eigenvalue problems with application to reactor physics, Internat. J. Numer. Methods Engrg. 95 (12) (2013) 1011–1032, http://dx.doi.org/10.1002/nme.4533.
[7] K. Lu, Y. Jin, Y. Chen, Y. Yang, L. Hou, Z. Zhang, Z. Li, C. Fu, Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems, Mech. Syst. Signal Process. 123 (2019) 264–297, http://dx.doi.org/10.1016/j.ymssp.2019.01.018.
[8] 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, J. Non-Newton. Fluid Mech. 139 (3) (2006) 153–176, http://dx.doi.org/10.1016/j.jnnfm.2006.07.007.
[9] F. Chinesta, P. Ladeveze, E. Cueto, A short review on model order reduction based on proper generalized decomposition, Arch. Comput. Methods Eng. 18 (4) (2011) 395–404, http://dx.doi.org/10.1007/s11831-011-9064-7.
[10] F. Chinesta, R. Keunings, A. Leygue, The Proper Generalized Decomposition for Advanced Numerical Simulations, Springer International Publishing, 2014, http://dx.doi.org/10.1007/978-3-319-02865-1.
[11] F. Cavaliere, S. Zlotnik, R. Sevilla, X. Larráyoz, P. Díez, Nonintrusive reduced order model for parametric solutions of inertia relief problems, Internat. J. Numer. Methods Engrg. (2021) http://dx.doi.org/10.1002/nme.6702.
[12] A. Dumon, C. Allery, A. Ammar, Proper general decomposition (PGD) for the resolution of Navier–Stokes equations, J. Comput. Phys. 230 (4) (2011) 1387–1407, http://dx.doi.org/10.1016/j.jcp.2010.11.010.
[13] C. Leblond, C. Allery, A priori space–time separated representation for the reduced order modeling of low Reynolds number flows, Comput. Methods Appl. Mech. Engrg. 274 (2014) 264–288, http://dx.doi.org/10.1016/j.cma.2014.02.010.
[14] R. Ibáñez, E. Abisset-Chavanne, F. Chinesta, A. Huerta, Simulating squeeze flows in multiaxial laminates: towards fully 3D mixed formulations, Int. J. Mater. Form. 10 (5) (2016) 653–669, http://dx.doi.org/10.1007/s12289-016-1309-4.
[15] P. Díez, S. Zlotnik, A. Huerta, Generalized parametric solutions in Stokes flow, Comput. Methods Appl. Mech. Engrg. 326 (2017) 223–240, http://dx.doi.org/10.1016/j.cma.2017.07.016.
[16] R. García-Blanco, D. Borzacchiello, F. Chinesta, P. Diez, Monitoring a PGD solver for parametric power flow problems with goal-oriented error assessment, Internat. J. Numer. Methods Engrg. 111 (6) (2017) 529–552, http://dx.doi.org/10.1002/nme.5470.
[17] M. Giacomini, L. Borchini, R. Sevilla, A. Huerta, Separated response surfaces for flows in parametrised domains: comparison of a priori and a posteriori PGD algorithms, 196, 2020, 103530, http://dx.doi.org/10.1016/j.finel.2021.103530,

[18] C. Ghnatios, F. Masson, A. Huerta, A. Leygue, E. Cueto, F. Chinesta, Proper generalized decomposition based dynamic data-driven control of thermal processes, Comput. Methods Appl. Mech. Engrg. 213–216 (2012) 29–41, http://dx.doi.org/10.1016/j.cma.2011.11.018.
[19] J.V. Aguado, A. Huerta, F. Chinesta, E. Cueto, Real-time monitoring of thermal processes by reduced-order modeling, Internat. J. Numer. Methods Engrg. 102 (5) (2014) 991–1017, http://dx.doi.org/10.1002/nme.4784.
[20] A. Huerta, E. Nadal, F. Chinesta, Proper generalized decomposition solutions within a domain decomposition strategy, Internat. J. Numer. Methods Engrg. 113 (13) (2018) 1972–1994, http://dx.doi.org/10.1002/nme.5729.
[21] J.P.M. de Almeida, A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics, Internat. J. Numer. Methods Engrg. 94 (10) (2013) 961–984, http://dx.doi.org/10.1002/nme.4490.
[22] J. Reis, J.P.M. de Almeida, P. Díez, S. Zlotnik, Error estimation for proper generalized decomposition solutions: A dual approach, Internat. J. Numer. Methods Engrg. 121 (23) (2020) 5275–5294, http://dx.doi.org/10.1002/nme.6452.
[23] J. Reis, J.P.M. de Almeida, P. Díez, S. Zlotnik, Error estimation for proper generalized decomposition solutions: Dual analysis and adaptivity for quantities of interest, Internat. J. Numer. Methods Engrg. 122 (3) (2020) 752–776, http://dx.doi.org/10.1002/nme.6559.
[24] E. Giner, B. Bognet, J.J. Ródenas, A. Leygue, F.J. Fuenmayor, F. Chinesta, The proper generalized decomposition (PGD) as a numerical procedure to solve 3D cracked plates in linear elastic fracture mechanics, Int. J. Solids Struct. 50 (10) (2013) 1710–1720, http://dx.doi.org/10.1016/j.ijsolstr.2013.01.039.
[25] H. Garikapati, S. Zlotnik, P. Díez, C.V. Verhoosel, E.H. van Brummelen, A proper generalized decomposition (PGD) approach to crack propagation in brittle materials: with application to random field material properties, Comput. Mech. 65 (2) (2019) 451–473, http://dx.doi.org/10.1007/s00466-019-01778-0.
[26] S. Zlotnik, P. Díez, D. Modesto, A. Huerta, Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications, Internat. J. Numer. Methods Engrg. 103 (10) (2015) 737–758, http://dx.doi.org/10.1002/nme.4909.
[27] M. Signorini, S. Zlotnik, P. Díez, Proper generalized decomposition solution of the parameterized Helmholtz problem: application to inverse geophysical problems, Internat. J. Numer. Methods Engrg. 109 (8) (2016) 1085–1102, http://dx.doi.org/10.1002/nme.5313.
[28] A. Sibileau, A. García-González, F. Auricchio, S. Morganti, P. Díez, Explicit parametric solutions of lattice structures with proper generalized decomposition (PGD), Comput. Mech. 62 (4) (2018) 871–891, http://dx.doi.org/10.1007/s00466-017-1534-9.
[29] G. Barroso, M. Seoane, A. Gil, P. Ledger, M. Mallett, A. Huerta, A staggered high-dimensional proper generalised decomposition for coupled magneto-mechanical problems with application to MRI scanners, Comput. Methods Appl. Mech. Engrg. 370 (2020) 113271, http://dx.doi.org/10.1016/j.cma.2020.113271.
[30] C. Quesada, I. Alfaro, D. González, E. Cueto, F. Chinesta, PGD-based model reduction for surgery simulation: Solid dynamics and contact detection, in: Biomedical Simulation, Springer International Publishing, 2014, pp. 193–202, http://dx.doi.org/10.1007/978-3-319-12057-7_22.
[31] D. González, E. Cueto, F. Chinesta, Real-time direct integration of reduced solid dynamics equations, Internat. J. Numer. Methods Engrg. 99 (9) (2014) 633–653.
[32] C. Germoso, J.V. Aguado, A. Fraile, E. Alarcon, F. Chinesta, Efficient PGD-based dynamic calculation of non-linear soil behavior, Compt. Rendus Mécanique 344 (1) (2016) 24–41, http://dx.doi.org/10.1016/j.crme.2015.09.002.
[33] M.H. Malik, D. Borzacchiello, J.V. Aguado, F. Chinesta, Advanced parametric space-frequency separated representations in structural dynamics: A harmonic–modal hybrid approach, Compt. Rendus Mécanique 346 (7) (2018) 590–602, http://dx.doi.org/10.1016/j.crme.2018.04.005.
[34] G. Quaranta, C.A. Martin, R. Ibañez, J.L. Duval, E. Cueto, F. Chinesta, From linear to nonlinear PGD-based parametric structural dynamics, Compt. Rendus Mécanique 347 (5) (2019) 445–454, http://dx.doi.org/10.1016/j.crme.2019.01.005.
[35] S. González-Pintor, D. Ginestar, G. Verdú, Using proper generalized decomposition to compute the dominant mode of a nuclear reactor, Math. Comput. Modelling 57 (7–8) (2013) 1807–1815, http://dx.doi.org/10.1016/j.mcm.2011.11.066.
[36] J.P. Senecal, Efficient Coupling Algorithms and Reduced-Order Methods for High-Fidelity Multiphysics Simulations of Nuclear Reactors, Rensselaer Polytechnic Institute, 2018.
[37] Z.M. Prince, J.C. Ragusa, Application of proper generalized decomposition to multigroup neutron diffusion eigenvalue calculations, Prog. Nucl. Energy 121 (2020) 103232, http://dx.doi.org/10.1016/j.pnucene.2019.103232.
[38] P. Díez, S. Zlotnik, A. García-González, A. Huerta, Encapsulated PGD algebraic toolbox operating with high-dimensional data, Arch. Comput. Methods Eng. 27 (4) (2019) 1321–1336, http://dx.doi.org/10.1007/s11831-019-09378-0.
[39] A. Leygue, E. Verron, A first step towards the use of proper general decomposition method for structural optimization, Arch. Comput. Methods Eng. 17 (4) (2010) 465–472, http://dx.doi.org/10.1007/s11831-010-9052-3.
[40] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, A. Poitou, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, Comput. Methods Appl. Mech. Engrg. 201–204 (2012) 1–12, http://dx.doi.org/10.1016/j.cma.2011.08.025.
[41] T. Heuzé, A. Leygue, G. Racineux, Parametric modeling of an electromagnetic compression device with the proper generalized decomposition, Int. J. Mater. Form. 9 (1) (2015) 101–113, http://dx.doi.org/10.1007/s12289-014-1212-9.
[42] A. Courard, D. Néron, P. Ladevèze, L. Ballere, Integration of PGD-virtual charts into an engineering design process, Comput. Mech. 57 (4) (2015) 637–651, http://dx.doi.org/10.1007/s00466-015-1246-y.
[43] L. Chamoin, H. Thai, Certified real-time shape optimization using isogeometric analysis, PGD model reduction, and a posteriori error estimation, Internat. J. Numer. Methods Engrg. 119 (3) (2019) 151–176, http://dx.doi.org/10.1002/nme.6045.
[44] R. Sevilla, S. Zlotnik, A. Huerta, Solution of geometrically parametrised problems within a CAD environment via model order reduction, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112631, http://dx.doi.org/10.1016/j.cma.2019.112631.
[45] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, H. van der Vorst, Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide, SIAM, 2000.
[46] G.H. Golub, C.F. Van Loan, Matrix Computations, Vol. 3, JHU Press, 2013.

[47] A. Quarteroni, R. Sacco, F. Saleri, Numerical Mathematics, Springer New York, 2007, http://dx.doi.org/10.1007/b98885.
[48] J.G.F. Francis, The QR transformation a unitary analogue to the LR transformation–part 1, Comput. J. 4 (3) (1961) 265–271, http://dx.doi.org/10.1093/comjnl/4.3.265.
[49] C. Lanczos, An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators, United States Governm. Press Office Los Angeles, CA, 1950.
[50] W.E. Arnoldi, The principle of minimized iterations in the solution of the matrix eigenvalue problem, Quart. Appl. Math. 9 (1) (1951) 17–29.
[51] E. Davison, S. Wang, Properties and calculation of transmission zeros of linear multivariable systems, Automatica 10 (6) (1974) 643–658, http://dx.doi.org/10.1016/0005-1098(74)90085-5.
[52] G.L.G. Sleijpen, H.A.V. der Vorst, A Jacobi–Davidson iteration method for linear eigenvalue problems, SIAM J. Matrix Anal. Appl. 17 (2) (1996) 401–425, http://dx.doi.org/10.1137/s0895479894270427.
[53] P. Díez, S. Zlotnik, A. García-González, A. Huerta, Algebraic PGD for tensor separation and compression: An algorithmic approach, Compt. Rendus Mécanique 346 (7) (2018) 501–514, http://dx.doi.org/10.1016/j.crme.2018.04.011.
[54] C. Felippa, K. Park, M.J. Filho, The construction of free–free flexibility matrices as generalized stiffness inverses, Comput. Struct. 68 (4) (1998) 411–418, http://dx.doi.org/10.1016/s0045-7949(98)00068-6.