Computers and Structures 315 (2025) 107826

Contents lists available at ScienceDirect

Computers and Structures

journal homepage: www.elsevier.com/locate/cas

Proper generalized decomposition surrogate modeling with application to the identification of Rayleigh damping parameters

Clément Vella [a,*], Serge Prudhomme [b]

[a] LaMcube – Univ. Lille, CNRS, Centrale Lille, UMR 9013, F-59000 Lille, France
[b] Department of Mathematics and Industrial Engineering, Polytechnique Montréal, C.P. 6079, succ. Centre-ville, Montréal, Québec H3C 3A7, Canada

HIGHLIGHTS

- Low-rank representations of the damped parameterized equation for 3D elastodynamics.
- Enhanced performance of the spatial solver by a new prediction–correction strategy.
- Damping identification using the surrogate model and Particle Swarm Optimization.
- Scalable method for accurate simulations of 3D viscoelastic structures.

ARTICLE INFO

Keywords:
Model reduction
Proper generalized decomposition
Surrogate modeling
Parametric Rayleigh damping
Inverse problem
Ritz pairs

ABSTRACT

This paper extends the Proper Generalized Decomposition framework to develop a reduced-order model parameterized by Rayleigh damping coefficients. The developed method incorporates damping modes to construct a damped surrogate model effectively. A novel method is introduced for treating the problem in space: during the offline phase, the spatial problem is initially projected onto the subspace spanned by the Ritz vectors of the system to provide an efficient prediction of the spatial modes. The prediction is then refined using a MinRes iterative solver. This two-step, prediction–correction process reduces the computational cost of a full-order solution while improving the accuracy of the reduced model. The resulting Proper Generalized Decomposition surrogate is subsequently employed within a Particle Swarm Optimization algorithm to determine optimal damping coefficients based on a given snapshot. Numerical experiments demonstrate the effectiveness of the developed method.

1. Introduction

Although computer resources and computational methods have greatly improved over the last half-century, there is still a crucial need to devise more efficient digital tools. This results from the ever increasing complexity of problems such as those encountered in design, control, optimization, or uncertainty quantification [1]. Indeed, accounting for the variability or uncertainty in model and input parameters, including e.g., variations in boundary or initial conditions, material, and geometric parameters, has become essential, driven by the need to finely study physical behaviors and improve the predictive capability of computational simulations. However, the drastic increase in the number of calculations needed to be performed has ruled out the use of conventional methods due to the so-called curse of dimensionality. Reduced-order modeling has emerged as a way to overcome this issue by bringing innovative strategies to deal with high-dimensional problems.

Several model reduction approaches have been put forth for applications in structural dynamics. One of the most prolific methods is the Proper Orthogonal Decomposition (POD) [1]. This data-driven approach builds a reduced model generally obtained by the Singular Value Decomposition (SVD) performed on snapshots describing the evolution of the states of the system. The performance of the POD has been demonstrated for both linear and nonlinear dynamics subjected to transient loads [2–4], and most notably, within a goal-oriented framework for elastodynamics, in the cases of either space-time approximation [5] and parameterized problems [6]. One well-known shortcoming of the Proper Orthogonal Decomposition (POD) is that it requires snapshots computed in an offline stage by an expensive high-fidelity solver, e.g., the Finite

* Corresponding author.
Email addresses: clement.vella@univ-lille.fr (C. Vella), serge.prudhomme@polymtl.ca (S. Prudhomme).

https://doi.org/10.1016/j.compstruc.2025.107826
Received 10 October 2024; Accepted 15 May 2025
Available online 29 May 2025

Element Method. In [7], the snapshots are evaluated at a given time-step only if the estimated error exceeds a given threshold and reduced bases are subsequently enriched in an adaptive manner. Such an approach allows one to mitigate the cost of evaluating the full-order model. Similarly, the so-called Reduced Basis (RB) technique provides a snapshot selection strategy designed to minimize the error committed by the reduced model [8,9].

By contrast, the Proper Generalized Decomposition (PGD) [10,11] completely removes this burden as the solution of the governing PDE is searched for along with the construction of the reduced basis. The PGD reduced model is usually obtained in an iterative manner by constructing one mode, or enrichment, at a time. In other words, the computation of a new enrichment is based on a residual that assesses and corrects the error committed by the previous iterate of the decomposition. In that regard, an enrichment can be viewed as an error correction.

An extensive study on the minimal residual formulation for the space-time separated PGD is presented in [12–14], with the goal of identifying an optimal norm to minimize the discrete residual in elastodynamics. In [15], the authors developed a space-time representation using a PGD approach based on a non-incremental Newmark integration scheme. They considered the discretized wave equation, both in space and time, within the Newmark framework to derive an algebraic system. A fully discrete space-time separated representation of the displacement was then introduced into this system, which was solved iteratively for the spatial and temporal modes using a fixed-point procedure. This implementation was extended to efficiently solve geometrically nonlinear structural dynamics problems, where residuals are linearized over the space-time domain around the current solution [16]. In [17], a strategy was proposed to address the challenges associated with space-time separation, where the number of spatial modes is adaptively estimated at each time step, assuming strong spatial separation. A PGD approach with separated material and geometric parameters was presented in [18]. The PGD is remarkably versatile, so much so that the parameter-separated strategy was extended to solve parameterized linear systems [19] and parameterized eigenvalue problems [20]. Alternatively, space-frequency separated representations have been developed for a broad range of applications: 2D acoustics [21,22], non-linear soil mechanics [23,24], linear and non-linear structural dynamics [25,26], and transient electronics [27]. This space-frequency separation has proven particularly efficient when additional parameters, such as material or geometric properties, are taken into account.

A recent variant of the PGD, called sparse PGD (sPGD), has been introduced as a core component of a multi-parametric framework for digital twins [28]. Used in an a posteriori setting with limited data, sPGD was combined with a Harmonic-Modal Hybrid (HMH) solver and the Discrete Empirical Interpolation Method (DEIM). Building on this, the authors developed a surrogate model for a Vertical Axis Rotating Machine subjected to nonlinear bearing forces, unbalances and imperfections, and varying loads and speeds [29]. The model, parameterized by unknown system properties, employs the HMH approach to construct a sparse solution library. Parameters are identified via Levenberg-Marquardt optimization using measured shaft displacements. The HMH formulation was also extended to degenerated domains, leveraging space-separated representations to enable high-resolution 3D modeling at reduced computational cost [30]. PGD was also implemented for transient Thermo-Hydro-Mechanical (THM) problems in porous media, with applications to deep geological repositories [31]. The study demonstrated the efficiency of PGD in delivering real-time solutions for complex, multiparameter THM scenarios.

On another note, with the growing interest in Machine Learning applied to ROM, it is worth mentioning Separable Physics-Informed Neural Networks (SPINN) [32] and Separable DeepONet (SepDeepONet) [33] frameworks. Like the PGD, the SPINN and SepDeepONet break the problem down by handling one-dimensional coordinates individually, with each sub-network dedicated to computing the separated modes for its respective coordinate. As a result, it reduces both the number of forward passes and the computational burden of Jacobian matrix calculations. These studies have demonstrated that this framework also achieves linear scaling of computational cost with increasing discretization density. In a similar manner, a hybrid approach combining PGD and PINNs was introduced in [34].

In the case of high-order tensors, i.e., whenever the order is higher than one, attempts to compute low-rank decomposition representations inevitably lead to ill-posed formulations [35]. This is mainly due to the non-uniqueness of such separated representations [36], which may result in convergence issues, as pointed out in [12,13,37], especially for the Galerkin-based PGD. The issue has been overcome with the implementation of a greedy rank-one strategy, in which one mode per parameter is computed, followed by an update of the formerly computed modes [38]. This updating procedure was observed to be necessary in the case of problems in elastodynamics but could also make the Galerkin-based PGD [37] diverge on some occasions. This is due to the tendency of the algorithm to compute modes that are not necessarily linearly independent. It was shown in [37] that a displacement–momentum formulation, where two bases are simultaneously built, one for the displacement and the other for the conjugate momentum, led to a robust PGD solver. One can then take advantage of the separated modes to orthogonalize them with respect to well chosen metrics, hence preventing ill-conditioning and divergence. The methodology was subsequently extended to a space-discrete, time-continuous Hamiltonian formalism. The updating procedure was further interpreted in terms of the preservation of the symplectic structure by the PGD [39]. Moreover, an additional approximation, inspired by Modal Decomposition, was introduced to mitigate the cost associated with solving the spatial problem. Aitken's delta-squared process and mode-orthogonalization were incorporated as well within the fixed-point iteration procedure to ensure convergence and stability of the algorithm.

In this paper, we further extend this implementation to parameterize the PGD reduced model with respect to the Rayleigh damping coefficients α, $\beta \in \mathbb{R}_+$. This hypothesis on the damping matrix C, written as $C = \alpha K + \beta M$ [40–42], is particularly convenient in Modal Decomposition since the damping operator is also diagonal in the eigenspace. Thus, the solution of the generalized eigenvalue problem $\boldsymbol{K}\boldsymbol{u} = \lambda \boldsymbol{M}\boldsymbol{u}$ allows one for computing the dynamical response of a structure in which damping is modeled with Rayleigh damping. Yet, these coefficients do not have a clear physical meaning regarding the materials [40,41]. Their estimation remains nevertheless an active research topic in both the experimental and numerical communities [43,44], as control of damping phenomena is a critical issue in the construction, automotive, or aerospace industries. Similarly to the Modal Decomposition, several methods based on Krylov subspaces have been developed to deal with proportionally damped reduced models [45–47]. However, both Modal Decomposition and Krylov-based methods operate in two steps: 1) build a ROM for the undamped model; 2) project the damped model onto the reduced basis for given values of α and β. The reduced model is actually not parameterized with respect to the damping parameters.

In contrast to these methods, the developed reduced model directly incorporates the dependence on Rayleigh damping parameters in the modes. Each enrichment term includes four modes: one spatial mode, one temporal mode, and one mode for each of the damping parameters α and β. The treatment of the problem in space draws upon our previous work. In the offline phase, computations are initially performed in the subspace spanned by the Ritz vectors of the system. This approach offers a computationally efficient estimation of the spatial modes but limits the accuracy of the PGD approximation to the information contained in the selected Ritz vectors [39]. To enhance accuracy, the estimation is subsequently refined using a Minimal Residual iterative solver. This two-step, prediction–correction process reduces the computational cost of a full-order solution while improving the accuracy of the reduced model. Subsequently, the PGD can be used as a surrogate to perform the optimization of the damping parameters with respect to a

given snapshot. A Particle Swarm Optimization (PSO) algorithm [48,49] drives the optimization process and queries the PGD surrogate to estimate the damped response of the structure. The numerical experiment is carried out with a snapshot generated by FEM, whose damping is modeled according to the Rayleigh hypothesis. It also assumes the knowledge of the time-dependent external load. The proof of concept consists in testing the capability of the parametric PGD to: 1) build a ROM that includes damping modes; 2) employ the surrogate to identify the coefficients that were used to generate the snapshot (inverse problem). The authors in [1] pointed out that the offline phase is somewhat suboptimal when the goal is to use the ROM for optimization. Indeed, the offline phase involves an exploration of the parametric space, but it is not known a priori whether the explored regions will subsequently be exploited throughout the online phase (optimization); or worse, the optimization process could lead to assess the response of the system in some regions where the offline phase did not gather enough information about the system's behavior. This remark especially applies to methods like the POD, where the exploration is contingent to costly, full-order solutions [50]. It will be shown that the developed method does not suffer from this issue. The first reason is that the need for full-order solutions is irrelevant here as the PGD is used as an *a priori* ROM. The second reason, of paramount importance, is that the complexity of the offline phase weakly depends on the discretization of the damping parameter spaces.

The paper is organized as follows: in Section 2, we describe the model problem in its parametric form and highlight the computational burden incurred by the use of conventional FEM solvers. The PGD approach is described in Section 3, with a particular focus on both the complexity of the algorithm and the solution of the problem in space. In Section 4, we describe the combination of the PSO method with the PGD as a surrogate. Some numerical experiments are presented in Section 5 to illustrate the performance of the developed approach. We finally provide some concluding remarks in Section 6.

2. Model problem

2.1. Strong formulation

The model problem we shall consider is that of elastodynamics in three dimensions under the assumption of infinitesimal deformation and linear viscoelastic damping. Let Ω be an open bounded domain of $\mathbb{R}^3$, with Lipschitz boundary $\partial\Omega$, and let $\mathcal{I} = (0,T)$ denote the time interval. The boundary $\partial\Omega$ is supposed to be decomposed into two portions, $\partial\Omega_D$ and $\partial\Omega_N$, such that $\partial\Omega = \overline{\partial\Omega_D \cup \partial\Omega_N}$. The displacement field, denoted by $u : \bar{\Omega}\times\bar{\mathcal{I}} \to \mathbb{R}^3$, satisfies the following partial differential equation:

$$\rho\frac{\partial^2 u}{\partial t^2} - \nabla\cdot\sigma(u) = f, \qquad \forall (x,t)\in\Omega\times\mathcal{I}, \tag{1}$$

where, in the case of infinitesimal deformation, the stress tensor $\sigma(u)$ and strain tensor $\varepsilon(u)$ are given by:

$$\sigma(u) = \mathbb{E}:\varepsilon(u) + \mathbb{D}:\dot{\varepsilon}(u), \qquad \forall (x,t)\in\Omega\times\mathcal{I}, \tag{2}$$

$$\varepsilon(u) = \frac{1}{2}\left(\nabla u + (\nabla u)^T\right), \qquad \forall (x,t)\in\Omega\times\mathcal{I}, \tag{3}$$

and is subjected to the initial conditions:

$$u(x,0) = u_0(x), \qquad \forall x\in\Omega, \tag{4}$$

$$\frac{\partial u}{\partial t}(x,0) = v_0(x), \qquad \forall x\in\Omega, \tag{5}$$

as well as to the boundary conditions:

$$u(x,t) = 0, \qquad \forall (x,t)\in\partial\Omega_D\times\mathcal{I}, \tag{6}$$

$$\sigma(u)\cdot n = g_N(x,t), \qquad \forall (x,t)\in\partial\Omega_N\times\mathcal{I}. \tag{7}$$

The functions $f : \Omega\times\mathcal{I}\to\mathbb{R}^3$, $u_0 : \Omega\to\mathbb{R}^3$, $v_0 : \Omega\to\mathbb{R}^3$, and $g_N : \partial\Omega_N\times\mathcal{I}\to\mathbb{R}^3$ are supposed to be sufficiently regular to yield a well-posed problem. The tensors $\mathbb{E}$ and $\mathbb{D}$ describe the elastic and viscous local properties, respectively, of the medium occupied in Ω. The latter is assumed to be isotropic, with density ρ and Lamé coefficients λ and μ (we note that the material parameters could possibly vary in space). The constitutive Eq. (2) thus reduces to:

$$\sigma(u) = 2\mu\varepsilon(u) + \lambda\mathrm{tr}\,(\varepsilon(u))\, I_3 + \mathbb{D}:\dot{\varepsilon}(u),$$

where $I_3\in\mathbb{R}^{3\times3}$ is the identity matrix. In the following, we will denote the first and second time derivatives by $\dot{u} = \partial u/\partial t$ and $\ddot{u} = \partial^2 u/\partial t^2$.

2.2. Semi-weak formulation

We consider here the semi-weak formulation with respect to the spatial variable in order to construct the discrete problem in space using the Finite Element Method (FEM). Multiplying (1) by an arbitrary smooth function $u^* = u^*(x)$ and integrating over the whole domain Ω, one obtains:

$$\int_\Omega \rho\ddot{u}\cdot u^* - (\nabla\cdot\sigma(u))\cdot u^*\, dx = \int_\Omega f\cdot u^*\, dx, \qquad \forall t\in\mathcal{I}. \tag{8}$$

Following the same development as in [39], the semi-discrete formulation of the problem then reads:

Find u such that for all $t\in\mathcal{I}, u(\cdot,t)\in V$, and

$$\int_\Omega \rho\ddot{u}\cdot u^* + \varepsilon(u):\mathbb{E}:\varepsilon(u^*) + \dot{\varepsilon}(u):\mathbb{D}:\varepsilon(u^*)\, dx = \int_\Omega f\cdot u^*\, dx + \int_{\partial\Omega_N} g_N\cdot u^*\, dx, \qquad \forall u^*\in V, \quad \forall t\in\mathcal{I}, \tag{9}$$

and

$$u(x,0) = u_0(x), \qquad \forall x\in\Omega,$$

$$\dot{u}(x,0) = v_0(x), \qquad \forall x\in\Omega,$$

where V is the vector space of vector-valued functions defined on Ω:

$$V = \left\{ v\in\left[H^1(\Omega)\right]^3 \,:\, v = 0 \text{ on } \partial\Omega_D \right\}.$$

2.3. Spatial discretization

We partition the domain into N_e tetrahedral elements K_e such that $\overline{\Omega} = \cup_{e=1}^{N_e} K_e$ and $\mathrm{Int}(K_i)\cap\mathrm{Int}(K_j) = \emptyset$, $\forall i,j = 1,\dots,N_e$, $i\neq j$. We then associate with the mesh the finite-dimensional Finite Element space W^h, $\dim W^h = s$, of scalar-valued continuous and piecewise polynomial functions defined on Ω, that is:

$$W^h = \left\{ v_h : \Omega\to\mathbb{R} \,:\, v_h|_{K_e}\in\mathbb{P}_k(K_e),\ e = 1,\dots,N_e \right\},$$

where $\mathbb{P}_k(K_e)$ denotes the space of polynomial functions of degree k on K_e. Let ϕ_i, $i = 1,\dots,s$, denote the basis functions of $W^h = \mathrm{span}\{\phi_i\}$. We then introduce the finite element subspace V^h of V such as:

$$V^h = \left[W^h\right]^3\cap V,$$

and search for finite element solutions u_h satisfying $u_h(\cdot,t)\in V^h$, $\forall t\in\bar{\mathcal{I}}$, in the form:

$$u_h(x,t) = \sum_{j=1}^{s}\phi_j(x)q_j(t),$$

where the vectors of degrees of freedom, $q_j\in\mathbb{R}^3$, depend on time. We introduce the set of $n = 3\ s$ vector-valued basis functions as:

$$\chi_{3i-2}(x) = \begin{bmatrix}\phi_i(x)\\0\\0\end{bmatrix}, \quad \chi_{3i-1}(x) = \begin{bmatrix}0\\\phi_i(x)\\0\end{bmatrix}, \quad \chi_{3i}(x) = \begin{bmatrix}0\\0\\\phi_i(x)\end{bmatrix}, \quad i = 1,\dots,s.$$

Using the Galerkin method, the Finite Element problem thus reads:

Find u_h such that $u_h(\cdot,t) \in V^h, \forall t \in \bar{\mathcal{I}}$, and

$$\int_\Omega \rho\chi_i(x)\cdot\ddot{u}_h(x,t) + \varepsilon(\chi_i)(x):\mathbb{E}:\varepsilon(u_h)(x,t) + \varepsilon(\chi_i)(x):\mathbb{D}:\dot{\varepsilon}(u_h)(x,t)\,dx$$
$$= \int_\Omega \chi_i(x)\cdot f(x,t)\,dx + \int_{\partial\Omega_N} \chi_i(x)\cdot g_N(x,t)\,ds, \quad \forall i = 1,\dots,n, \quad \forall t \in \mathcal{I},$$

satisfying the initial conditions

$$u_h(x,0) = u_{0,h}(x), \quad \forall x \in \Omega,$$
$$\dot{u}_h(x,0) = v_{0,h}(x), \quad \forall x \in \Omega,$$

where $u_{0,h}$ and $v_{0,h}$ are interpolants or projections of u_0 and v_0 in the space V^h. The above problem can be conveniently recast in compact form as:

$$M\ddot{\boldsymbol{q}}(t) + C\dot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{10}$$
$$\boldsymbol{q}(0) = \boldsymbol{u}_0, \tag{11}$$
$$\dot{\boldsymbol{q}}(0) = \boldsymbol{v}_0, \tag{12}$$

where M, C, and K are the global mass, damping, and stiffness matrices, respectively:

$$M_{ij} = \int_\Omega \rho\chi_i\cdot\chi_j\,dx,$$
$$K_{ij} = \int_\Omega \varepsilon(\chi_i):\mathbb{E}:\varepsilon(\chi_j)\,dx, \qquad \forall i,j = 1,\dots,n,$$
$$C_{ij} = \int_\Omega \varepsilon(\chi_i):\mathbb{D}:\varepsilon(\chi_j)\,dx,$$

$\boldsymbol{f}(t)$ is the loading vector at time t whose components are given by:

$$f_i(t) = \int_\Omega \chi_i(x)\cdot f(x,t)\,dx + \int_{\partial\Omega_N} \chi_i(x)\cdot g_N(x,t)\,ds, \qquad \forall i = 1,\dots,n,$$

$\boldsymbol{q}(t)$ is the global vector of degrees of freedom:

$$\boldsymbol{q}(t) = \begin{bmatrix} q_1(t) & \dots & q_s(t) \end{bmatrix}^T,$$

where $q_i \in \mathbb{R}^3$, $i = 1,\dots,s$, and $\boldsymbol{u}_0$ and $\boldsymbol{v}_0$ are the initial vectors:

$$\boldsymbol{u}_0 = \begin{bmatrix} u_{0,1} & \dots & u_{0,s} \end{bmatrix}^T,$$
$$\boldsymbol{v}_0 = \begin{bmatrix} v_{0,1} & \dots & v_{0,s} \end{bmatrix}^T.$$

Note that $u_{0,i} \in \mathbb{R}^3$ and $v_{0,i} \in \mathbb{R}^3$, $i = 1,\dots,s$, are vectors whose components are the initial displacements and velocities in the three spatial directions. Moreover, the above definition of C is not practical and will be replaced by the model described in the next section.

2.4. Damping modeling and parameter discretization

We assume here that damping is "proportional" to K and M, that is, $C = \alpha K + \beta M$, where α, $\beta \in \mathbb{R}_+$ are the so-called Rayleigh damping coefficients. Furthermore, we shall rewrite the damping matrix as follows:

$$C(\alpha,\beta) = \alpha\bar{\alpha}K + \beta\bar{\beta}M, \tag{13}$$

where the constant parameters $\bar{\alpha}$, $\bar{\beta} \in \mathbb{R}_+$ are interpreted as reference parameters while α and β encapsulate the variability in the parameters. Let $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ be spaces for parameters α, $\beta \in \mathbb{R}_+$, respectively, such that:

$$\mathcal{D}_\alpha = \left[\alpha_{\min}, \alpha_{\max}\right],$$
$$\mathcal{D}_\beta = \left[\beta_{\min}, \beta_{\max}\right].$$

Let $\mathcal{D}$ be the global parameter space defined as the tensor product of $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$, i.e., $\mathcal{D} = \mathcal{D}_\alpha \times \mathcal{D}_\beta$. Our goal is to compute the mechanical response of a system over $\Omega \times \mathcal{I} \times \mathcal{D}$. Finite element solutions are sought for in the form:

$$u_h(x,t,\alpha,\beta) = \sum_{j=1}^{s} \phi_j(x) q_j(t,\alpha,\beta).$$

In other words, we aim at modeling the mechanical behavior for multiple pairs of values $(\alpha,\beta) \in \mathcal{D}$, (α,β) being chosen so that the computed approximations are in the following solution manifold:

$$\mathcal{M}^h = \left\{ v_h : \mathcal{D} \to W^h \otimes L^2(\mathcal{I});\ \int_{\mathcal{D}} |||v_h|||^2 d\mu < \infty \right\},$$

where:

$$|||u||| = \sqrt{\int_{\mathcal{I}}\int_\Omega \frac{1}{2}\rho\dot{u}\cdot\dot{u} + \frac{1}{2}\varepsilon(u):\mathbb{E}:\varepsilon(u)\,dxdt}.$$

Later on, the parameter spaces $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ will be partitioned into $(n_\alpha - 1)$ and $(n_\beta - 1)$ subintervals, respectively, that is, $\mathcal{D}_\alpha = \cup_{i=2}^{n_\alpha}[\alpha_{i-1},\alpha_i]$ and $\mathcal{D}_\beta = \cup_{i=2}^{n_\beta}[\beta_{i-1},\beta_i]$. Here, n_α and n_β denote the numbers of grid points α_i, $i = 1,\dots,n_\alpha$, and β_i, $i = 1,\dots,n_\beta$, chosen in $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$, respectively. We also define $\|\cdot\|_{\mathcal{M}^h}$ such that:

$$\forall u \in \mathcal{M}^h, \quad \|u\|_{\mathcal{M}^h} = \int_{\mathcal{D}} |||u|||^2 d\mu.$$

In many applications, it is useful to know the mechanical response for many pairs of parameters (α,β) in the global parameter space $\mathcal{D}$. However, this entails a substantial computational burden when using conventional methods, as shown in the next section.

2.5. Discretization in time

The time domain $\mathcal{I}$ is divided into n_t subintervals $\mathcal{I}^i = [t^{i-1}, t^i]$, with $i = 1,\dots,n_t$, of size $h_i = t^i - t^{i-1}$. For the sake of simplicity, we assume here that all intervals are of same length h_t, i.e., $h_i = h_t$, $i = 1,\dots,n_t$. Introducing the vector $\boldsymbol{p}(t) = M\dot{\boldsymbol{q}}(t)$, Eq. (10) can be recast into the system of coupled equations:

$$\dot{\boldsymbol{p}}(t) + C\dot{\boldsymbol{q}}(t) + K\boldsymbol{q}(t) = \boldsymbol{f}(t), \qquad \forall t \in \mathcal{I}, \tag{14}$$
$$\dot{\boldsymbol{q}}(t) - M^{-1}\boldsymbol{p}(t) = 0, \qquad \forall t \in \mathcal{I}. \tag{15}$$

The two Eqs. (14) and (15) are discretized using the Crank–Nicolson scheme (also referred to as the implicit trapezoidal rule) to obtain, for $i = 1,\dots,n_t$:

$$\begin{bmatrix} h_tK + 2C & 2I_n \\ 2I_n & -h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^i \\ \boldsymbol{p}^i \end{bmatrix} = \begin{bmatrix} -h_tK + 2C & 2I_n \\ 2I_n & h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^{i-1} \\ \boldsymbol{p}^{i-1} \end{bmatrix} + h_t \begin{bmatrix} \boldsymbol{f}^i + \boldsymbol{f}^{i-1} \\ 0 \end{bmatrix}, \tag{16}$$

with:

$$\boldsymbol{q}^0 = \boldsymbol{u}_0,$$
$$\boldsymbol{p}^0 = M\boldsymbol{v}_0.$$

Eq. (16) can be rewritten as:

$$\begin{bmatrix} h_tK + 2C & 2I_n \\ 2I_n & -h_tM^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{q}^i \\ \boldsymbol{p}^i \end{bmatrix} = \begin{bmatrix} \boldsymbol{b}_q^i \\ \boldsymbol{b}_p^i \end{bmatrix},$$

with

$$\boldsymbol{b}_q^i = \left[-h_tK + 2C\right]\boldsymbol{q}^{i-1} + 2\boldsymbol{p}^{i-1} + h_t\left(\boldsymbol{f}^i + \boldsymbol{f}^{i-1}\right),$$
$$\boldsymbol{b}_p^i = 2\boldsymbol{q}^{i-1} + h_tM^{-1}\boldsymbol{p}^{i-1}.$$

Developing above system yields the two equations:

$$\left[h_t K + 2C\right] \boldsymbol{q}^i + 2\boldsymbol{p}^i = \boldsymbol{b}_q^i,$$
$$2\boldsymbol{q}^i - h_t M^{-1} \boldsymbol{p}^i = \boldsymbol{b}_p^i.$$

Rewriting the second equation as:

$$\boldsymbol{p}^i = \frac{2}{h_t} M \boldsymbol{q}^i - \frac{1}{h_t} M \boldsymbol{b}_p^i,$$

and substituting $\boldsymbol{p}^i$ for this new expression in the first equation lead to:

$$\left[h_t K + 2C + \frac{4}{h_t} M\right] \boldsymbol{q}^i = \boldsymbol{b}_q^i + \frac{2}{h_t} M \boldsymbol{b}_p^i.$$

Finally, applying the Rayleigh hypothesis for C (13), one has to solve at each time-step the coupled equations:

$$\left[(h_t + 2\alpha\bar{\alpha})K + \left(\frac{4}{h_t} + 2\beta\bar{\beta}\right) M\right] \boldsymbol{q}^i = \boldsymbol{b}_q^i + \frac{2}{h_t} M \boldsymbol{b}_p^i, \tag{17}$$
$$\boldsymbol{p}^i = \frac{2}{h_t} M \boldsymbol{q}^i - \frac{1}{h_t} M \boldsymbol{b}_p^i. \tag{18}$$

Since K and M are both symmetric and positive-definite, so is the left-hand side matrix in (17). Therefore, the equation can be solved at the cost of one Cholesky factorization of the matrix $A = \left[(h_t + 2\alpha\bar{\alpha})K + \left(4h_t^{-1} + 2\beta\bar{\beta}\right) M\right]$ (as long as the time-step h_t remains constant), while all subsequent operations inside one time-step loop will have costs similar to those of matrix-vector multiplications.

The corresponding algorithm is shown in Algorithm 1. Its complexity depends on the number of non-zero elements in the matrices that are dealt with and on their sparsity patterns (note that K and M feature the same number of non-zero elements and sparsity patterns, properties that are directly inherited by A). As an approximation, if we consider that all matrices are dense, of size n, the cost of Algorithm 1 is thus proportional to that of a Cholesky factorization, i.e., $\mathcal{O}(n^3)$, and n_t matrix-vector multiplications, each having a cost of $\mathcal{O}(n^2)$. The resulting complexity of the algorithm is therefore of the order of $\mathcal{O}(n^3 + n_t n^2)$. We also note that the Cholesky factorization is the most expensive operation, but that the integrating scheme could also be costly if n_t becomes large.

Accounting for parameters, the evaluation of the mechanical response over the discrete global parameter space $\mathcal{D}$ by means of Algorithm 1 has complexity of order $\mathcal{O}(n_\alpha n_\beta (n^3 + n_t n^2))$. It shows that the complexity of such a task rapidly grows with respect to the number of parameters. We describe below a reduced-order technique that efficiently tackles the evaluation of u_h when dealing with many parameters. Furthermore, we will denote the complexity of solving a linear system of n algebraic equations in n unknowns as $\mathrm{lin}(n)$. This notation is used to generalize the complexity, whether a direct or iterative solver is employed, and regardless of the sparsity pattern of the operators.

Algorithm 1 Classic elastodynamics FEM–Crank–Nicolson solver

Input: assembled operators K, M and $\boldsymbol{f}$
  initial conditions $\boldsymbol{q}^0$ and $\boldsymbol{p}^0$
Define $A = \left[(h_t + 2\alpha\bar{\alpha})K + \left(4h_t^{-1} + 2\beta\bar{\beta}\right) M\right]$
Factorize A: `L = cholesky(A)` *see* `sksparse.cholmod` [51] *sparse Cholesky decomposition*
for $i = 1, \ldots, n_t$ **do**
  Update $\boldsymbol{b}_q^i$ and $\boldsymbol{b}_p^i$
  Solve $\boldsymbol{q}^i$ = `L.solve_A`$(\boldsymbol{b}_q^i + 2h_t^{-1} M \boldsymbol{b}_p^i)$ *forward / backward substitution for Eq. (17)*
  Compute $\boldsymbol{p}^i$ *Eq. (18)*
end for
Output: $Q = \left[\boldsymbol{q}^0 \ \ldots \ \boldsymbol{q}^{n_t}\right]$ and $P = \left[\boldsymbol{p}^0 \ \ldots \ \boldsymbol{p}^{n_t}\right]$

3. Parametric PGD reduced-order modeling

The Proper Generalized Decomposition method aims at approximating both the generalized coordinates $\boldsymbol{q}$ and their generalized momenta $\boldsymbol{p}$ in separated form. We are thus searching for a space-time separated representation of $\boldsymbol{q}$ and $\boldsymbol{p}$ as:

$$\boldsymbol{q}(t,\alpha,\beta) \approx \boldsymbol{q}_m(t,\alpha,\beta) = \sum_{i=1}^{m} \boldsymbol{\varphi}_i^q \psi_i^q(t) \xi_i^q(\alpha) \zeta_i^q(\beta),$$
$$\boldsymbol{p}(t,\alpha,\beta) \approx \boldsymbol{p}_m(t,\alpha,\beta) = \sum_{i=1}^{m} \boldsymbol{\varphi}_i^p \psi_i^p(t) \xi_i^p(\alpha) \zeta_i^p(\beta).$$

For the sake of clarity in the presentation, we shall, from now on, drop the subscript i and write the decompositions of rank m as:

$$\boldsymbol{q}(t,\alpha,\beta) \approx \boldsymbol{q}_m(t,\alpha,\beta) = \boldsymbol{q}_{m-1}(t,\alpha,\beta) + \boldsymbol{\varphi}_q \psi_q(t) \xi_q(\alpha) \zeta_q(\beta),$$
$$\boldsymbol{p}(t,\alpha,\beta) \approx \boldsymbol{p}_m(t,\alpha,\beta) = \boldsymbol{p}_{m-1}(t,\alpha,\beta) + \boldsymbol{\varphi}_p \psi_p(t) \xi_p(\alpha) \zeta_p(\beta).$$

The bold notations $\boldsymbol{z}$, $\boldsymbol{\varphi}$, $\boldsymbol{\psi}$, $\boldsymbol{\xi}$, and $\boldsymbol{\zeta}$ denote vertical concatenations such that:

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{q} \\ \boldsymbol{p} \end{bmatrix}, \quad \boldsymbol{\varphi} = \begin{bmatrix} \boldsymbol{\varphi}_q \\ \boldsymbol{\varphi}_p \end{bmatrix}, \quad \boldsymbol{\psi} = \begin{bmatrix} \psi_q \\ \psi_p \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} \xi_q \\ \xi_p \end{bmatrix}, \quad \boldsymbol{\zeta} = \begin{bmatrix} \zeta_q \\ \zeta_p \end{bmatrix}.$$

The following weighted residual form of (14)–(15) is considered:

$$\begin{aligned} &\int_D \int_I \boldsymbol{q}^{*T} \left[\dot{\boldsymbol{p}} + \left(\alpha\bar{\alpha} K + \beta\bar{\beta} M\right) \dot{\boldsymbol{q}} + K\boldsymbol{q}\right] dt d\mu \\ &\quad = \int_D \int_I \boldsymbol{q}^{*T} \boldsymbol{f} \, dt d\mu, \quad \forall \boldsymbol{q}^* \in \left[L^2(D) \otimes L^2(I)\right]^n, \\ &\int_D \int_I \boldsymbol{p}^{*T} \left[\dot{\boldsymbol{q}} - M^{-1}\boldsymbol{p}\right] dt d\mu = 0, \quad \forall \boldsymbol{p}^* \in \left[L^2(D) \otimes L^2(I)\right]^n. \end{aligned} \tag{19}$$

The separated representation is computed in a progressive manner by adding one quadruplet of modes $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ at each enrichment, following the so-called greedy rank-one update algorithm. In other words, we substitute $\boldsymbol{q}_m$ and $\boldsymbol{p}_m$ for $\boldsymbol{q}$ and $\boldsymbol{p}$, respectively, so that the modes $(\boldsymbol{\varphi}_q, \psi_q, \xi_q, \zeta_q)$ and $(\boldsymbol{\varphi}_p, \psi_p, \xi_p, \zeta_p)$ become the new unknowns of the problem. Subsequently, the modes are computed one by one at each enrichment, with $\boldsymbol{z}_{m-1}$ being known. This computing paradigm has the virtue of drastically reducing the complexity by splitting the computational costs into lower dimensional sub-problems.

The goal in this section is to show how the PGD sub-problems are modified in comparison to our previous work [39] with the introduction of additional modes for parameters α and β. This development is rather straightforward and is drawn from an example with the transient diffusion equation in [11], where the diffusivity is considered as an extra-coordinate. The authors highlight that the analytical solution of the transient diffusion equation cannot be written in a parameter-separated format. Yet, the PGD strategy is somehow efficient to tackle the diffusion problem. Likewise, modal superposition does not express the solution of the problem that is therein dealt with in a separated form [52]. We will show in Section 5 that the PGD was found to be efficient when applied to the problem at hand. Special attention will be devoted to the way we address the spatial problem and a new resolution strategy will be presented. The effect of the PGD technique on the complexity will be detailed as well.

3.1. Fixed-point algorithm

If the complexity reduction offered by the PGD strategy is appealing, it however leads to a non-linear formulation for the modes. At each enrichment m, the problem to be solved for $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ reads:

$$\begin{aligned} &\int_D \int_I \boldsymbol{q}^{*T} \left[\boldsymbol{\varphi}_p \dot{\psi}_p \xi_p \zeta_p + \left(\alpha\bar{\alpha} K + \beta\bar{\beta} M\right) \boldsymbol{\varphi}_q \psi_q \xi_q \zeta_q + K \boldsymbol{\varphi}_q \psi_q \xi_q \zeta_q\right] dt d\mu \\ &\quad = \int_D \int_I \boldsymbol{q}^{*T} \boldsymbol{r}_{m-1}^q \, dt d\mu, \ \forall \boldsymbol{q}^*, \end{aligned}$$

$$\int_D \int_I \boldsymbol{p}^{*T} \left[\boldsymbol{\varphi}_q \dot{\psi}_q \xi_q \zeta_q - M^{-1} \boldsymbol{\varphi}_p \psi_p \xi_p \zeta_p \right] dt d\mu = \int_D \int_I \boldsymbol{p}^{*T} \boldsymbol{r}^p_{m-1} \, dt d\mu, \ \forall \boldsymbol{p}^*,$$

where $\boldsymbol{r}^q_{m-1}$ and $\boldsymbol{r}^p_{m-1}$ are the mth residuals:

$$\boldsymbol{r}^q_{m-1} = \begin{cases} \boldsymbol{f}, & \text{if } m = 1, \\ \boldsymbol{f} - \dot{\boldsymbol{p}}_{m-1} - \left(\alpha\bar{\alpha} K + \beta\bar{\beta} M \right) \dot{\boldsymbol{q}}_{m-1} - K \boldsymbol{q}_{m-1}, & \text{if } m > 1, \end{cases}$$

$$\boldsymbol{r}^p_{m-1} = \begin{cases} \boldsymbol{0}, & \text{if } m = 1, \\ M^{-1} \boldsymbol{p}_{m-1} - \dot{\boldsymbol{q}}_{m-1}, & \text{if } m > 1. \end{cases}$$

The above problem is solved in an iterative manner using the following fixed point algorithm:

1. Solve (19) for $\boldsymbol{\varphi}$ with $(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ known. This step will be referred to as the spatial problem and is written in a generic form as:

$$A_S(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta}) \boldsymbol{\varphi} = \boldsymbol{b}_S(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{20}$$

   where the $2n \times 2n$ matrix A_S and vector $\boldsymbol{b}_S$ of size $2n$ will be specified in Section 3.2.
2. Solve (19) for $\boldsymbol{\xi}$ with $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta})$ known. This problem consists in solving the $2n_\alpha$ algebraic equations:

$$A_\xi(\alpha, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta}) \boldsymbol{\xi} = \boldsymbol{b}_\xi(\alpha, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{21}$$

   where the 2×2 matrix A_ξ and vector $\boldsymbol{b}_\xi$ of size 2 are explicitly provided in Appendix A. In other words, there is one 2×2 linear system to solve for each value of the parameter α.
3. Solve (19) for $\boldsymbol{\zeta}$ with $(\boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi})$ known. This problem consists in solving the $2n_\beta$ algebraic equations:

$$A_\zeta(\beta, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}) \boldsymbol{\zeta} = \boldsymbol{b}_\zeta(\beta, \boldsymbol{\varphi}, \boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{z}_{m-1}), \tag{22}$$

   where the 2×2 matrix A_ζ and vector $\boldsymbol{b}_\zeta$ of size 2 are explicitly provided in Appendix A. As for α, there is one 2×2 linear system to solve for each value of the parameter β.
4. Solve (19) for $\boldsymbol{\psi}$ with $(\boldsymbol{\varphi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ known. The temporal problem corresponds to the system of first-order differential equations:

$$\dot{\boldsymbol{\psi}} = f_{\mathcal{T}}(t, \boldsymbol{\psi}, \boldsymbol{\varphi}, \boldsymbol{\xi}, \boldsymbol{\zeta}, \boldsymbol{z}_{m-1}), \tag{23}$$

   where the vector-valued function $f_{\mathcal{T}}$ is explicitly provided in Appendix A.

The fixed-point procedure is illustrated in Fig. 1. It appears that the resulting complexity of the PGD algorithm is of order $\mathcal{O}\left(m k_{\max} \left(\text{lin}(n) + n_t + n_\alpha + n_\beta\right)\right)$, with $k_{\max}$ the maximum number of fixed-point iterations to reach convergence. We will show in the next section that this complexity can further be decreased by projecting the spatial problem onto a subspace spanned by Ritz vectors.

3.2. Problem in space

3.2.1. Projection in ritz subspace

We assume that $(\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\zeta})$ is known and we search for the new spatial mode $\boldsymbol{\varphi}$. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}^*_q \psi_q \xi_q \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}^*_p \psi_p \xi_p \zeta_p$. Equation (19) leads to the linear system:

$$A_S \boldsymbol{\varphi} = \boldsymbol{b}_S, \tag{24}$$

where:

$$A_S = \begin{bmatrix} k_{qq} K + m_{qq} M & c_{qp} I_n \\ c_{pq} I_n & m_{pp} M^{-1} \end{bmatrix},$$

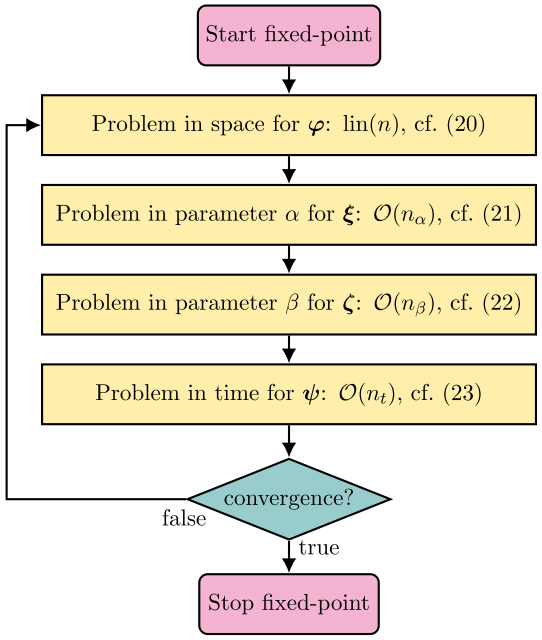

Fig. 1. Flowchart of the conventional fixed-point algorithm with associated computational complexities.

$$\boldsymbol{b}_S = \begin{bmatrix} \int_D \int_I \psi_q \xi_q \zeta_q \boldsymbol{r}^q_{m-1} \, dt d\mu \\ \int_D \int_I \psi_p \xi_p \zeta_p \boldsymbol{r}^p_{m-1} \, dt d\mu \end{bmatrix},$$

and the real coefficients are defined in Appendix A. The introduction of viscous damping, assumed to be proportional to K and M, allows us to write (24) in a form similar to that in our previous work [39]. Therefore, following [39], we can project Eq. (24) onto the subspace of approximated eigenvectors, namely the Ritz vectors, which verify the properties (with $m \leqslant r \ll n$):

$$\left(\hat{\Lambda}, \hat{V}\right) \in \mathbb{R}^{r \times r} \times \mathbb{R}^{n \times r}, \quad \text{such that} \quad \hat{V}^T K \hat{V} = \hat{\Lambda}, \quad \text{and} \quad \hat{V}^T M \hat{V} = I_r,$$

where the Ritz values and associated Ritz vectors are:

$$\hat{\Lambda} = \text{diag}\left(\hat{\lambda}_1, \ \ldots \ , \hat{\lambda}_r\right),$$
$$\hat{V} = \left[\hat{\boldsymbol{v}}_1 \ \ldots \ \hat{\boldsymbol{v}}_r\right].$$

We now introduce the mapping:

$$R = \begin{bmatrix} \hat{V} & 0 \\ 0 & M\hat{V} \end{bmatrix}.$$

The problem in space (24), using $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$, can thus be rewritten as:

$$\hat{A}_S \hat{\boldsymbol{\varphi}} = \hat{\boldsymbol{b}}_S, \tag{25}$$

with:

$$\hat{A}_S = R^T A_S R = \begin{bmatrix} k_{qq} \hat{\Lambda} + m_{qq} I_r & c_{qp} I_r \\ c_{pq} I_r & m_{pp} I_r \end{bmatrix},$$

$$\hat{\boldsymbol{b}}_S = R^T \boldsymbol{b}_S = \begin{bmatrix} \hat{\boldsymbol{b}}_q \\ \hat{\boldsymbol{b}}_p \end{bmatrix}.$$

The structure of $\hat{A}_S$ exhibits a coupling of the components of $\hat{\boldsymbol{\varphi}}_q$ and $\hat{\boldsymbol{\varphi}}_p$. The solution can be explicitly expressed, component-wise, as:

$$\hat{\boldsymbol{\varphi}}_{q,i} = \frac{1}{\hat{m}_{qq} m_{pp} - c_{qp} c_{pq}} \left(m_{pp} \hat{\boldsymbol{b}}_{q,i} - c_{qp} \hat{\boldsymbol{b}}_{p,i} \right), \tag{26}$$

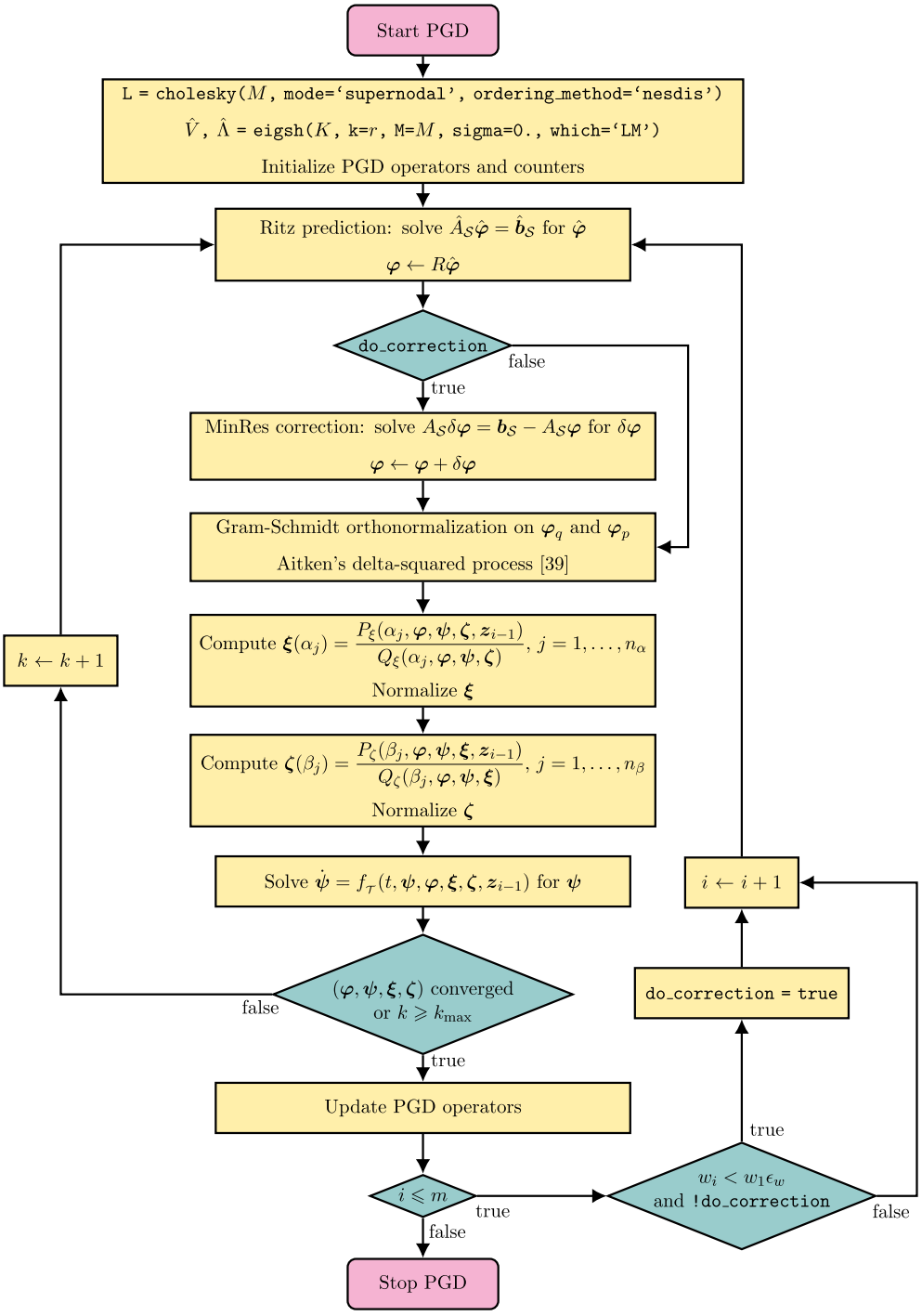

Fig. 2. Flowchart of the PGD offline phase with the hybrid space solver.

$$\hat{\boldsymbol{\varphi}}_{p,i} = \frac{1}{\hat{m}_{qq} m_{pp} - c_{qp} c_{pq}} \left(\hat{m}_{qq} \hat{\boldsymbol{b}}_{p,i} - c_{pq} \hat{\boldsymbol{b}}_{q,i} \right), \tag{27}$$

with:

$$\hat{m}_{qq} = k_{qq} \hat{\lambda}_i + m_{qq}.$$

The complexity of the spatial problem (24) is linear in terms of the dimension r of the Ritz subspace, which results in the overall complexity of order $\mathcal{O}\left(m k_{\max} \left(r + n_t + n_\alpha + n_\beta\right)\right)$. It is noteworthy that part of the complexity of the computation is transferred to the solution of the generalized eigenproblem, i.e., $\boldsymbol{K}\boldsymbol{u} = \lambda \boldsymbol{M}\boldsymbol{u}$. However, the solution of the eigenproblem, followed by the PGD offline phase, allows one to obtain the solution to the parameterized problem for all parameter values, in contrast to one FEM solution per pair (α, β) for all possible values of α and β. Strictly speaking, this method consists in searching for the spatial modes as a linear combination of Ritz vectors and can be interpreted as an extra reduction step *per se*. It will be shown to be extremely efficient computation-wise.

3.2.2. Hybrid space solver

The solution to the projected spatial problem (25) involves a trade-off between computational efficiency and solution accuracy. On one

hand, the projection of the system onto a subspace spanned by Ritz vectors is extremely efficient but introduces additional numerical errors. Moreover, it was observed that the accuracy of the PGD approximation is bounded by the information contained in the Ritz vectors. On the other hand, solving the full system (24) does not introduce any extra approximation and thus enables one to keep finding modes that increase the ROM accuracy. Unfortunately, as it relies on assembling and factorizing the matrix A_S in (24) at each fixed-point iteration, this turns out to be prohibitive as soon as the number of spatial degrees of freedom becomes too large. We refer the reader to [39] for further details on this issue.

In this section, we develop a compromise, which consists of adaptively searching for spatial modes either as a linear combination of Ritz vectors, performing a projection (25), or as a linear combination of the full-order FE basis functions, solving for the full problem (24). In fact, the relevant information contained in the Ritz vectors is usually extracted after around 20 modes (see e.g., Fig. 13). However, when dealing with a problem involving several parameters, the number of enrichments should be increased to obtain a satisfactory accuracy, in which case one should consider solving the full-order problem (24). As in [39], spatial modes $\boldsymbol{\varphi}_q$ and $\boldsymbol{\varphi}_p$ are orthonormalized with respect to K and M^{-1}, respectively. For the damping parameter modes, they are normalized such that:

$$\begin{aligned} &\int_{D_\alpha} \left(\xi_i^q\right)^2 d\alpha = 1, \quad \int_{D_\alpha} \left(\xi_i^p\right)^2 d\alpha = 1, \\ &\int_{D_\beta} \left(\zeta_i^q\right)^2 d\beta = 1, \quad \int_{D_\beta} \left(\zeta_i^p\right)^2 d\beta = 1, \end{aligned} \qquad i = 1, \dots, m.$$

Thus, for each PGD enrichment, a factor w_i can be defined in the space-discrete formalism as:

$$w_i = \int_D \int_I \left(\frac{1}{2} \left(\psi_i^q \xi_i^q \zeta_i^q\right)^2 \left(\boldsymbol{\varphi}_i^q\right)^T K \boldsymbol{\varphi}_i^q + \frac{1}{2} \left(\psi_i^p \xi_i^p \zeta_i^p\right)^2 \left(\boldsymbol{\varphi}_i^p\right)^T M^{-1} \boldsymbol{\varphi}_i^p \right) dt d\mu.$$

With separability and normalization, the integral reduces to:

$$w_i = \frac{1}{2} \int_I \left(\psi_i^q\right)^2 + \left(\psi_i^p\right)^2 dt.$$

Alike mode participation factors in modal analysis, w_i measures the contribution of the ith mode in the description of the dynamic response. We will hereafter refer to these factors as "contribution factors". Their magnitude is expected to decrease as the rank of the enrichment increases. We thus develop the following adaptive approach (see Fig. 2):

1. Compute enrichment terms using the Ritz approximation of the spatial problem: solve (25) for $\hat{\boldsymbol{\varphi}}$ and set $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$. Repeat until $w_i/w_1 < \epsilon_w$, a user-defined tolerance, at which point a MinRes correction will be activated, as described in Step 2;
2. Use the mode $\boldsymbol{\varphi}$ computed in Step 1 as an initial guess $\boldsymbol{\varphi}^0 := \boldsymbol{\varphi}$ and solve (24) for $\boldsymbol{\varphi}$ by means of the Minimal Residual (MinRes) iterative solver. This approach is similar to a prediction–correction method.

Such an algorithm circumvents the computational burden incurred by the resolution of (24). Furthermore, the prediction of the spatial mode given by $\boldsymbol{\varphi} = R\hat{\boldsymbol{\varphi}}$ has proved effective when used as an initial guess, since it significantly decreases the number of MinRes iterations (as shown in Fig. 14).

4. Parameter identification using particle swarm optimization

The problem we address here deals with the identification of the parameters α and β with respect to given snapshots u_S. The goal is to find values $(\alpha^*, \beta^*) \in D$ such that a displacement field, denoted by $u_h = u_h(\alpha, \beta) \in \mathcal{M}^h$ and evaluated by means of a chosen computational

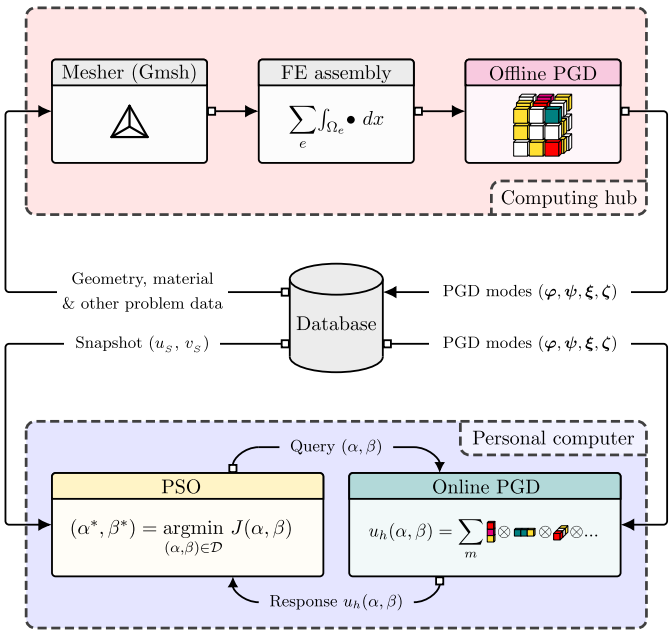

Fig. 3. Pseudo flowchart of the deployed solution.

method, minimizes a given cost function $J : D \to \mathbb{R}$. The problem thus reads:

$$\text{Find } (\alpha^*, \beta^*) \in D \text{ such that } (\alpha^*, \beta^*) \in \underset{(\alpha,\beta)\in D}{\operatorname{argmin}} J(\alpha, \beta).$$

The cost function measures the deviation between the snapshot u_S and the computed approximation u_h, usually defined with respect to a Quantity of Interest (QoI). In the test case that will be presented thereafter, we focus on the mechanical response over the boundary $\partial\Omega_N$, where Neumann boundary conditions are enforced. The cost function can be formulated as:

$$J(\alpha, \beta) = \frac{\sqrt{\int_I \int_{\partial\Omega_N} \frac{1}{2} \rho \dot{e}(\alpha, \beta) \cdot \dot{e}(\alpha, \beta) + \frac{1}{2} \varepsilon\left(e(\alpha, \beta)\right) : \mathbb{E} : \varepsilon\left(e(\alpha, \beta)\right) \, dx dt}}{\sqrt{\int_I \int_{\partial\Omega_N} \frac{1}{2} \rho v_S \cdot v_S + \frac{1}{2} \varepsilon\left(u_S\right) : \mathbb{E} : \varepsilon\left(u_S\right) \, dx dt}},$$

with:

$$e(\alpha, \beta) = u_S - u_h(\alpha, \beta),$$
$$\dot{e}(\alpha, \beta) = v_S - \dot{u}_h(\alpha, \beta).$$

The optimization problem is computationally demanding, particularly if $u_h(\alpha, \beta)$ is evaluated by a conventional FEM solver. Thus, the use of the PGD as a surrogate model should improve the efficiency of the process as it provides fast evaluations of $u_h(\alpha, \beta)$ (see Fig. 3).

The optimization problem is solved using the Particle Swarm Optimization (PSO) technique [48,49]. PSO was chosen for its weakly-intrusive nature, making it well-suited for problems where additional parameters may be introduced. Unlike gradient-based methods, PSO does not require significant modifications to the code to accommodate new parameters, ensuring flexibility in optimization while maintaining computational efficiency. A recent study implemented a PSO optimizer within the framework of Parametric Model Order Reduction to address the optimal design of fiber composites [53]. The optimization focuses on four design variables representing tow path angles. Additionally, theoretical results establish its global convergence under well-prepared initialization and suitable hyperparameter choices [54]. These findings support its applicability in high-dimensional, nonconvex settings, reinforcing its use in complex optimization tasks.

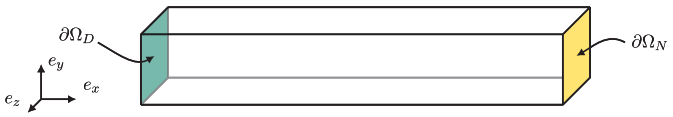

Fig. 4. Schematic of the 3D beam $\bar{\Omega} = [0, 6] \times [0, 1] \times [0, 1]$.

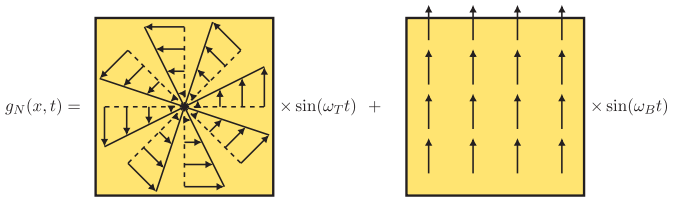

Fig. 5. Schematic of the boundary multiaxial load $\sigma \cdot n$ prescribed on $\partial\Omega_N$.

In the presented framework, one considers n_{PSO} particles individually associated with a triplet $(\alpha_p, \beta_p, J_p = J(\alpha_p, \beta_p))$, with $p = 1, \dots, n_{PSO}$, that are used to explore in an iterative manner the search space $\mathcal{D}$. The state of each particle is denoted by $x_p^k = \left(\alpha_p^k, \beta_p^k, J_p^k\right)$, with k being the index of the current PSO iteration and p the particle label. The PSO technique proceeds as follows:

1. Initialization: a population of n_{PSO} particles is randomly initialized within $\mathcal{D}$. Each particle is assigned a triplet $x_p^0 = \left(\alpha_p^0, \beta_p^0, J_p^0\right)$;
2. Individual update: each particle updates its own best state $x_p^{k_p}$, where $k_p = \underset{1 \leqslant i \leqslant k}{\operatorname{argmin}}\, J_p^i$, for $p = 1, \dots, n_{PSO}$;
3. Global update: update the global best state $x_{p_b}^k$, which represents the best solution found by any particle in the entire swarm at the current iteration, where $p_b = \underset{1 \leqslant p \leqslant n_{PSO}}{\operatorname{argmin}}\, J_p^k$;
4. State update: the state of each particle is updated based on its current state, its own best state, and global best state, according to the following equations:

$$\text{Velocity update} \quad \begin{cases} \dot{\alpha}_p^{k+1} = w\dot{\alpha}_p^{k+1} + c_1\epsilon_p^k\left(\alpha_p^{k_p} - \alpha_p^k\right) + c_2\eta_p^k\left(\alpha_{p_b}^k - \alpha_p^k\right), \\ \dot{\beta}_p^{k+1} = w\dot{\beta}_p^{k+1} + c_1\epsilon_p^k\left(\beta_p^{k_p} - \beta_p^k\right) + c_2\eta_p^k\left(\beta_{p_b}^k - \beta_p^k\right), \end{cases}$$

$$\text{Position update} \quad \begin{cases} \alpha_p^{k+1} = \alpha_p^k + \dot{\alpha}_p^{k+1}, \\ \beta_p^{k+1} = \beta_p^k + \dot{\beta}_p^{k+1}, \end{cases}$$

where w, c_1, and c_2 are three control parameters, namely the inertia weight, the cognitive, and the social acceleration coefficients, respectively, and ϵ_p^k and η_p^k are random perturbations. These parameters influence the search strategy between exploitation and exploration. The term exploitation means that one carries out the search around known promising states (own and global best states, promoted by c_1 and c_2). A drawback of assigning too much weight to exploitation is the premature attraction to local optima. Conversely, exploration promotes the evaluation of states as widely as possible within the search space (promoted by w, ϵ_p^k, and η_p^k). It may in return lead to a higher number of iterations for the algorithm to converge towards an optimal solution. Consequently, the parameters are to be tuned in order to find a good trade-off between exploitation and exploration. Steps 2 to 4 are repeated until a convergence criterion is fulfilled. It is worth noting that evaluating the state of each particle can be seamlessly performed in parallel.

5. Numerical examples

5.1. Test case: parameter optimization with respect to a QoI

The test case consists of a 3D beam clamped on one end $\partial\Omega_D$ and subjected to a multiaxial load on the other end $\partial\Omega_N$, see Fig. 4. The computational domain $\Omega = (0, 6)\times(0, 1)\times(0, 1)$ (in meters) is a rectangular parallelepiped with a squared cross-section. Its response to an external load on its right end is computed over the time interval $\mathcal{I} = (0, 1)$ (in seconds). The governing equations are those introduced in Section 2.1 with $f = 0$. Moreover, the beam is subjected to homogeneous initial conditions:

$$u(x, 0) = 0, \qquad \forall x \in \Omega,$$
$$\frac{\partial u}{\partial t}(x, 0) = 0, \qquad \forall x \in \Omega,$$

and to the boundary conditions:

$$u(x, t) = 0, \qquad \forall (x, t) \in \partial\Omega_D \times \mathcal{I},$$
$$\sigma(u) \cdot n = g_N(x, t), \qquad \forall (x, t) \in \partial\Omega_N \times \mathcal{I},$$
$$\sigma(u) \cdot n = 0, \qquad \forall (x, t) \in \partial\Omega_0 \times \mathcal{I}.$$

The external load g_N, applied to the boundary $\partial\Omega_N = \{6\} \times (0, 1) \times (0, 1)$ of the structure, is described in Fig. 5 and results in both vertical bending and torsion. The beam is free on the remainder of the boundary $\partial\Omega_0 = \partial\Omega\backslash(\partial\Omega_D \cup \partial\Omega_N)$. The values of the parameters are chosen as follows:

$E = 220$ GPa,
$\nu = 0.3$,
$\rho = 7{,}000$ kg/m^3,
$\omega_B = 160$ rad/s,
$\omega_T = 875$ rad/s,
$\bar{\alpha} = 10^{-5}$ s,
$\bar{\beta} = 1$ s^{-1},
$\alpha_{\min} = \beta_{\min} = 0.2$,
$\alpha_{\max} = \beta_{\max} = 2$,

and the Lamé coefficients are evaluated as:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}.$$

The time domain $\mathcal{I}$ is divided into $n_t = 4800$ sub-intervals of equal size. The space domain Ω is partitioned into linear tetrahedral elements and five meshes will be considered such that the number $2n$ of spatial degrees of freedom (DOF) takes values in $\{33{,}990, 147{,}174, 554{,}412, 1{,}408{,}002, 3{,}002{,}406\}$. Regardless of the spatial discretization, the number of Ritz vectors is set to $r = 300$, corresponding to the smallest Ritz values [52]. The *offline phase* of the PGD is performed for $m = 100$ modes. In the case of the hybrid space solver, the tolerance ϵ_w is set to 10^{-6}. The evaluation of the cost function J is carried out using the PGD reduced-order model, referred to as the *online phase*. Unless stated otherwise, $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$ are both discretized using $n_\alpha = n_\beta = 190$ values for α and β. Thus, the dimension of the discrete global parameter space $\mathcal{D}$ is 36,100. Note that the evaluation of J is not limited to these discrete parameter values as it can be evaluated for any pair $(\alpha, \beta) \in \mathcal{D}$ by interpolation of the PGD damping modes $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$. A linear interpolation will be considered for this purpose. The global best PSO variant is used with the inertia weight, the cognitive and the social acceleration coefficients set to $w = 0.4$, $c_1 = 0.5$, and $c_2 = 0.3$, respectively. The PSO algorithm will run in parallel with $n_{PSO} = 32$ particles. Recall that the overall solution process is illustrated in Fig. 3.

5.2. Comparison method and performance criteria

We shall report and compare the results based on the following criteria:

1. The relative error ϵ_{ROM} of the PGD approximations with respect to the full-order solutions, namely the FEM solutions described in Section 2.3, is defined as:

$$\epsilon_{ROM} = \sqrt{\frac{\|u_{FEM} - u_{ROM}\|_{\mathcal{M}^h}}{\|u_{FEM}\|_{\mathcal{M}^h}}}.$$

   For feasibility reasons, ϵ_{ROM} will be assessed only for a small number of spatial DOF, i.e., $2n = 36{,}774$, and coarse parameter discretizations $n_\alpha = n_\beta = 10$ for $\mathcal{D}_\alpha$ and $\mathcal{D}_\beta$. Three approaches will be compared for the problem in space: (1) only the Ritz prediction is performed; (2) an LU factorization is performed at each fixed-point iteration; (3) a prediction–correction with a MinRes solver is performed, as described in Section 3.2.2. The three approaches are respectively labeled "Ritz", "Full", and "Hybrid" on Fig. 13. The evolution of the contribution factors w_i will be measured to verify the efficiency of the hybrid solver. Past these results, only the hybrid solver will be considered;
2. The time speedup factor achieved by the reduced-order model when compared to the conventional FEM solver, see Algorithm 1. It will include scaling performance with respect to the size of the spatial discretization. The scaling performance refers to the offline phase that was performed on a computing hub offering more resources than most personal computers;
3. The decrease in the cost function, evaluated by the swarm particles. The relative errors in the optimized parameters $(\alpha^k_{p_b}, \beta^k_{p_b})$, with respect to the pair (α_S, β_S) employed to generate the snapshots, are defined as:

$$\epsilon^k_\alpha = \frac{\alpha^k_{p_b} - \alpha_S}{\alpha_S}, \quad \epsilon^k_\beta = \frac{\beta^k_{p_b} - \beta_S}{\beta_S}.$$

   The errors are assessed at every PSO iteration, denoted here by k, for the global best particle. The snapshots and the offline PGD are computed on the same mesh featuring $2n = 3{,}002{,}406$ spatial DOFs. The optimization was performed on a personal computer, using the PGD as a surrogate model (online phase) and the PSO to drive the parameter search.

As far as computer times are concerned, the configurations of both the computing hub and the personal computer are detailed below:

- Computing hub (AWS' `c5.12xlarge` instance):
  - Intel(R) Xeon(R) Platinum 8275CL CPU @ 3.00 GHz (24 cores, 48 threads);
  - RAM: 96 GB;
  - OS: Ubuntu 22.04.4 LTS (Jammy Jellyfish).
- Personal computer:
  - CPU: AMD Ryzen 7 PRO 4750U @ 1.7 GHz per core (8 cores, 16 threads);
  - RAM: 38 GB;
  - OS: Arch Linux kernel version 6.10.6.

The code, written using Python 3.9.19, leverages Intel's MKL with the Intel Distribution for Python and SuiteSparse's CHOLMOD Supernodal Sparse Cholesky Factorization [51]. The MinRes algorithm is SciPy `scipy.sparse.linalg.minres` [55] with a tolerance for the relative residual set to $\texttt{tol} = 5 \times 10^{-8}$. The offline PGD computation was performed using the computing hub on 16 cores. The PSO, fed with the PGD surrogate model, was carried out by the research toolkit PySwarms [56] using the personal computer on 8 cores.

Table 1
Time efficiency of the PGD offline phase with respect to different spatial discretizations with $n_\alpha = n_\beta = 190$.

# DOF in space	33,990	147,174	554,412	1,408,002	3,002,406
Offline PGD (s)	24	90	360	999	2317

Table 2
Time efficiency of the FEM solver described in Algorithm 1 for one computation.

# DOF in space	33,990	147,174	554,412	1,408,002	3,002,406
FEM (s)	29	188	847	2431	5757

Table 3
Summary of the time and memory efficiency of the developed PGD solver compared to those of a conventional FEM for a computation with $2n = 3{,}002{,}406$ spatial DOFs, $n_t = 4800$ time-steps and $n_\alpha = n_\beta = 190$ values for the damping parameters (the PGD results are provided by the code while the FEM results are estimated based on the result for a single computation).

		PGD	FEM
Offline phase	Time (s)	2317	n.a.
	Memory (GB)	59	n.a.
Optimization	Time (s)	322	10^7
	Memory (GB)	2.7	52
Storage		4.8 GB	4200 TB

5.3. Numerical results

We show in Fig. 13 the evolution of the relative error ϵ_{ROM} with respect to the number of modes, when using the Ritz projection, the full, and the hybrid approaches. We observe that ϵ_{ROM} decreases rather quickly for about the first 30 enrichments. However, the limitation of the Ritz projection is clearly highlighted in this test case as we can see that the useful information contained in the $r = 300$ Ritz vectors has been thoroughly exploited after 30 enrichments. The adaptive hybrid solver activates the MinRes correction around the 30$^{\text{th}}$ enrichment, after which an abrupt decrease in ϵ_{ROM} along with an increase of the contribution factors w_i are observed. One aspect that has not been discussed so far is the loss of information due to the projection in Eq. (25) on the representation of the original right-hand side of Eq. (24). In fact, it is rather difficult to predict in advance up to which rank the representation of the right-hand side residual in the base formed by the Ritz vectors will be accurate. In particular, it is likely that high-frequency variations in the residual will not be retained after projection. This information is however recovered as soon as the hybrid solver switches to the solution of the full-order spatial problem (24). Fig. 13 also shows the error reduction of the PGD solver with LU factorization, labeled "Full", and it confirms the relevance of the hybrid solver. Indeed, the hybrid solver mitigates computational costs with an accuracy similar to that of the "Full" solver. As a point of comparison, for 33,990 spatial DOFs, the offline phase takes 970 s when an LU factorization is used to solve the problem in space, compared to 24 s with the hybrid strategy. Regarding the latter, Fig. 14 illustrates the impact of the prediction–correction scheme, reducing the average number of MinRes iterations per linear solve (24) from 91 to 20. It is also worth mentioning that the Cholesky factorization of K was tested as a preconditioner for the MinRes solver, but this choice was not retained. While it reduces the number of iterations to an average of 10, the additional computations required for the Cholesky forward and backward solves ultimately make this approach slower.

The results regarding execution time are reported in Tables 1 and 2. Note that the execution time for the FEM solver corresponds to the computation for a single pair (α, β). As multiple evaluations of the response are needed to proceed with parameter optimization, one may conclude that the use of a classic FEM solver is prohibitive. Conversely, once the

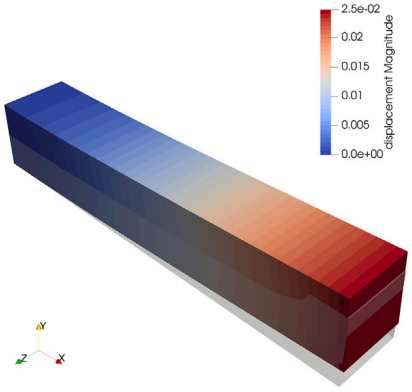

Fig. 6. 1st spatial mode.

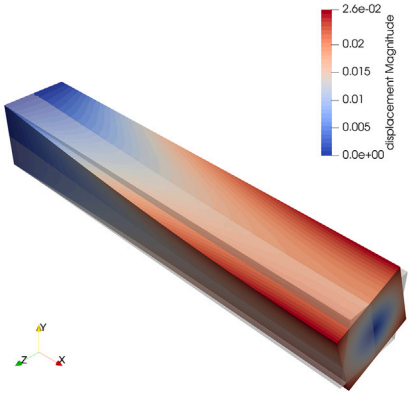

Fig. 7. 2nd spatial mode.

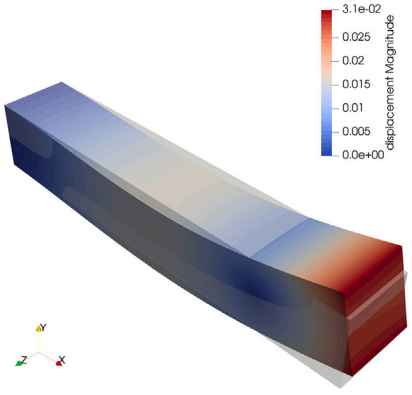

Fig. 8. 3rd spatial mode.

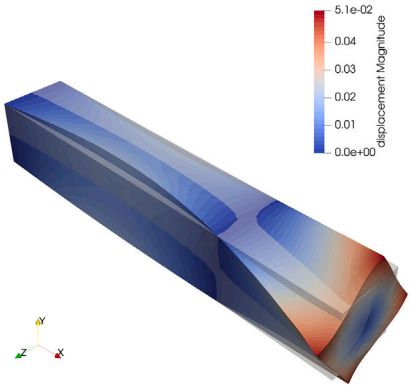

Fig. 9. 4th spatial mode.

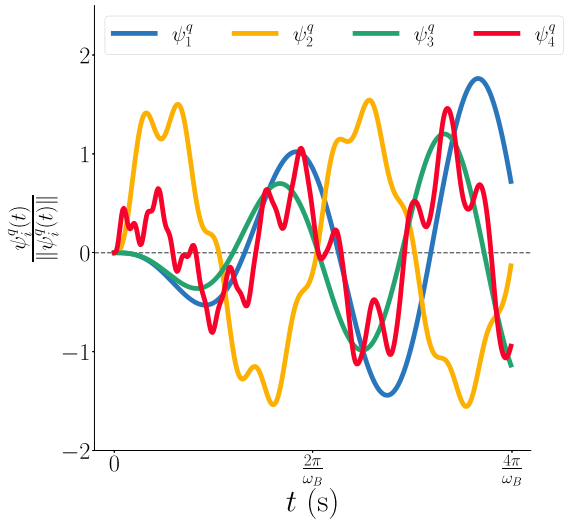

Fig. 10. First four temporal modes.

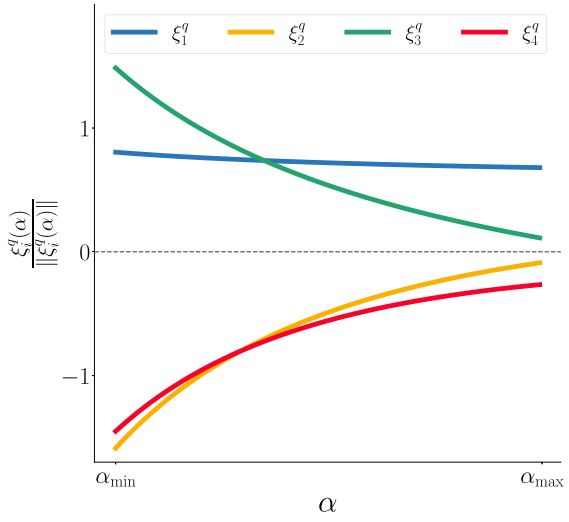

Fig. 11. First four α-modes.

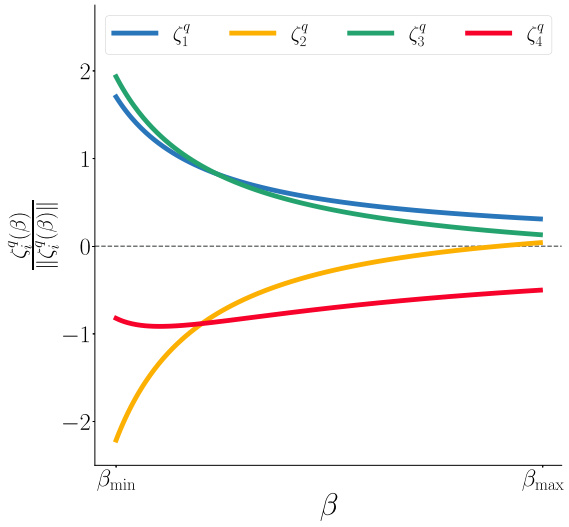

Fig. 12. First four β-modes.

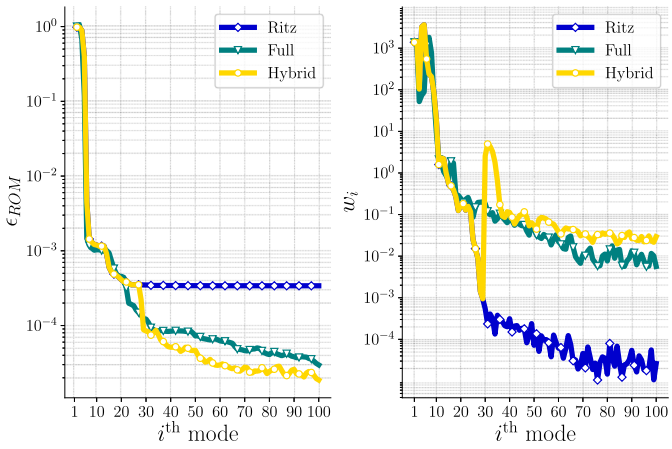

Fig. 13. (Left) Relative error ϵ_{ROM} between the reference fields and the PGD fields for the three strategies for the problem in space (y-axis in log scale). (Right) Factor w_i of the ith enrichment for the three strategies for the problem in space (y-axis in log scale).

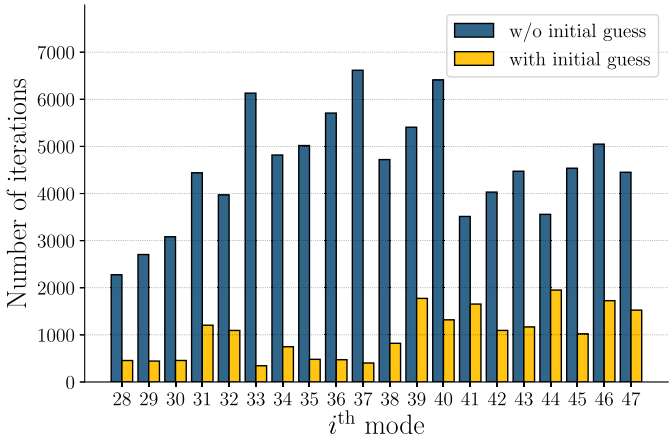

Fig. 14. Impact of an initial guess on the number of MinRes iterations (cumulative sum of MinRes iterations per fixed-point iteration).

PGD offline phase is completed, the modes can be used as a surrogate model to quickly assess the mechanical behavior for any pair $(\alpha, \beta) \in \mathcal{D}$. Fig. 15 shows the time distribution across the different phases of the computations. It clearly illustrates that the time marching scheme is the computationally most expensive phase of the FEM solver. In the case of the PGD solver, the space problem accounts for the majority of the computational time. The diagram clearly demonstrates the excellent efficiency of the Ritz estimates, even when including the time spent to compute the Ritz vectors during the preprocessing stage. Notably, the damping modes contribute minimally to the overall computational time. Note that the time required to assemble the operators is included in the diagram slices for each problem, respectively. Therefore, a possible improvement would be to work on a more efficient method for assembling the operators for the right-hand side residuals and the left-hand side (when applicable). This remark especially applies to the problem in space. Indeed, the problems related to time and damping parameters are purely one-dimensional with respect to the time variable t and damping parameters α and β. In contrast, the spatial problem does not assume separability with respect to the three spatial directions, which results in an increased workload.

Fig. 16 shows the evolution of the cost function evaluated for both the global best and global worst particles of the swarm as well as the relative errors ϵ_α^k and ϵ_β^k for the global best particle. We illustrate in Fig. 17 the PSO iterative search and in Fig. 18 the surface of the cost function near the global minimum. Note that the surface was post-processed for visualization purposes only as PSO does not require any knowledge other than the one collected by the swarm particles through their search. One observes in Fig. 16 that the best particle reaches an optimal state around the 40th iteration. The convergence of the swarm towards this optimum is reached around the 55th iteration. The final optimized damping coefficients achieve errors that are within acceptable bounds, and the value of the optimized cost function also indicates the effectiveness of the approach. Thus, we consider that $55 \times n_{PSO} = 1760$ evaluations of the cost function, and a fortiori of $u_h(\alpha, \beta)$, were required to solve the optimization problem. Using the PGD surrogate model (online phase), the optimization was performed in 322 s. If a conventional FEM solver were to be used instead, this time would have been several orders of magnitude longer. Accounting for the offline phase, the overall speedup factor is estimated to be 3800. Beyond time efficiency, the memory usage needed to store the results is also an aspect to be accounted for. After the offline phase, we are provided with separated representations for both the generalized coordinates and momenta, i.e., $2m$ modes for the space, time, and damping parameters, respectively, which amount to a total of $2m(n + n_t + n_\alpha + n_\beta)$ floating numbers to store. By contrast, the storage of all FEM solutions amounts to $2n \times n_t \times n_\alpha \times n_\beta$ floating numbers. Therefore, the PGD approximation also offers a gain in memory by several orders of magnitude. Table 3 summarizes these results, considering that numbers are stored in double precision. From a computational point of view, the PGD offline phase may be seen as a substantial overhead with respect to the optimization problem. However, the computational cost of such an approach is subsequently justified by the gains it enables.

A last noteworthy comment is about the physical interpretation of the results. The graphical representation of the parameter-separated modes provides a picture of the influence of each parameter used to simulate the physical phenomena. As a reminder, our simulation consists of a viscoelastic 3D beam that is clamped on one end and subjected to bending and torsional, periodic, mechanical efforts on the other end. The bending and torsional loading functions have different pulsations, denoted by ω_B and ω_T, respectively. We observe in Figs. 6–12 that the computed PGD modes are rather consistent with the physics being modeled. Indeed,

1. The spatial modes $\{\boldsymbol{\varphi}_1^q, \boldsymbol{\varphi}_3^q\}$, shown in Figs. 6 and 8, and $\{\boldsymbol{\varphi}_2^q, \boldsymbol{\varphi}_4^q\}$, shown in Figs. 7 and 9, contribute to the bending and torsion motions, respectively;
2. The temporal modes $\{\psi_1^q, \psi_3^q\}$, see Fig. 10, are clearly related to the bending frequency ω_B, while the modes $\{\psi_2^q, \psi_4^q\}$ are similar to signals featuring two frequencies. The slow and fast variations seem to be related to ω_B and ω_T, respectively. For mode ψ_4^q, some higher frequencies are visible in the vicinity of $t = 0$. We interpret them as small corrections as they do not appear to have a particular physical meaning regarding the model;
3. Damping modes: their magnitudes tend to decrease as the values of the damping parameters increase, which is physically sensible.

5.4. Further discussion

In this section, we make several comments in relation to the developed PGD strategy.

We first recall that the estimation of time and memory for the optimization phase with a conventional FEM solver, as reported in Table 3, was based on the fact that one solution was computed at a time, the solution fields were only stored over $\partial\Omega_N \times \bar{I}$, and the data was erased after the cost function was evaluated. Tackling an optimization problem this way is an energy drain. Conversely, the PGD allows to store once and for all the solution fields over $\bar{\Omega} \times \bar{I} \times D_\alpha \times D_\beta$. The data compression enabled by the PGD allows one to reuse past computations to carry out new analyses. For instance, if one wanted to study a specific

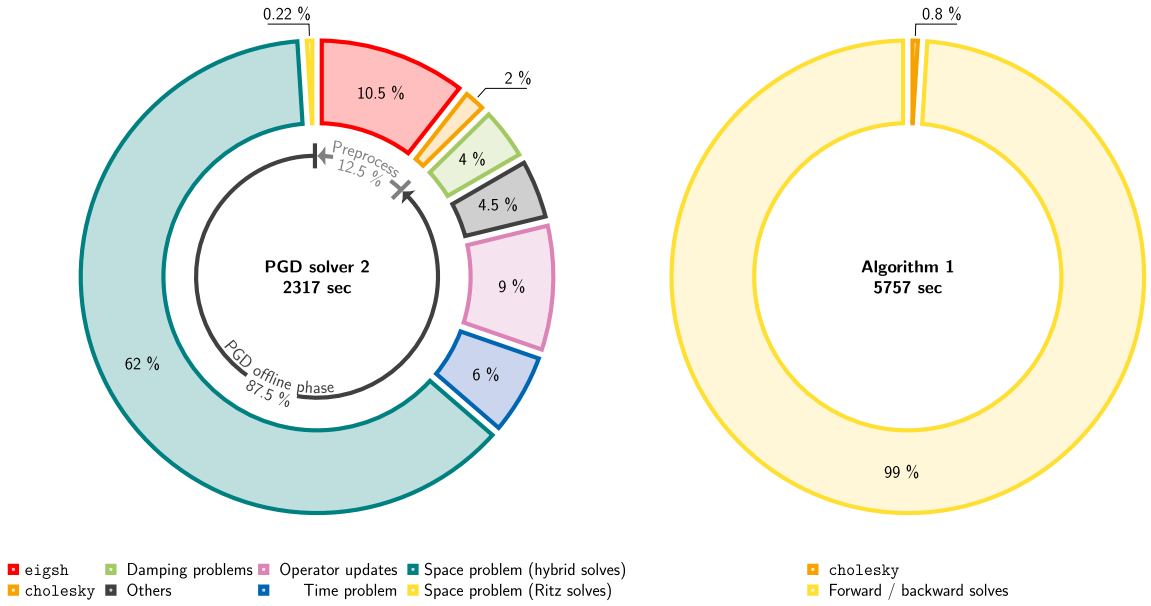

Fig. 15. (Left) Time distribution of the developed PGD solver described in Fig. 2 ($2n$ = 3,002,406 spatial DOFs, n_t = 4800 time-steps and $n_\alpha = n_\beta$ = 190 values for the damping parameters). (Right) Time distribution of the FEM solver described in Algorithm 1 ($2\,n$ = 3,002,406 spatial DOFs, n_t = 4800 time-steps for one computation).

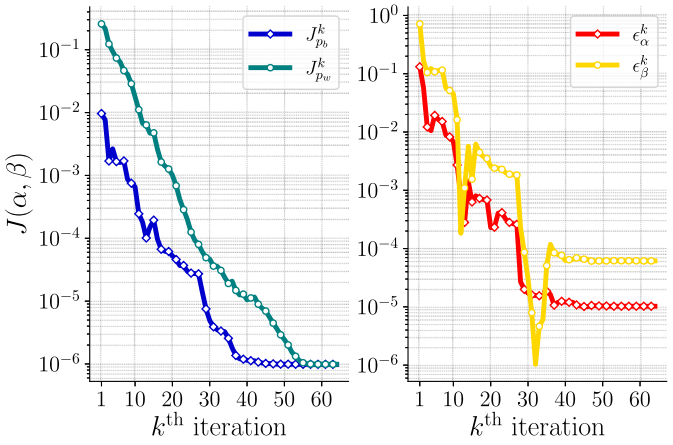

Fig. 16. (Left) Values of the global best cost $J^k_{p_b}$ and the global worst cost $J^k_{p_w}$ during the PSO iterations. (Right) Values of the errors ϵ_α and ϵ_β for the global best particle p_b during the PSO iterations.

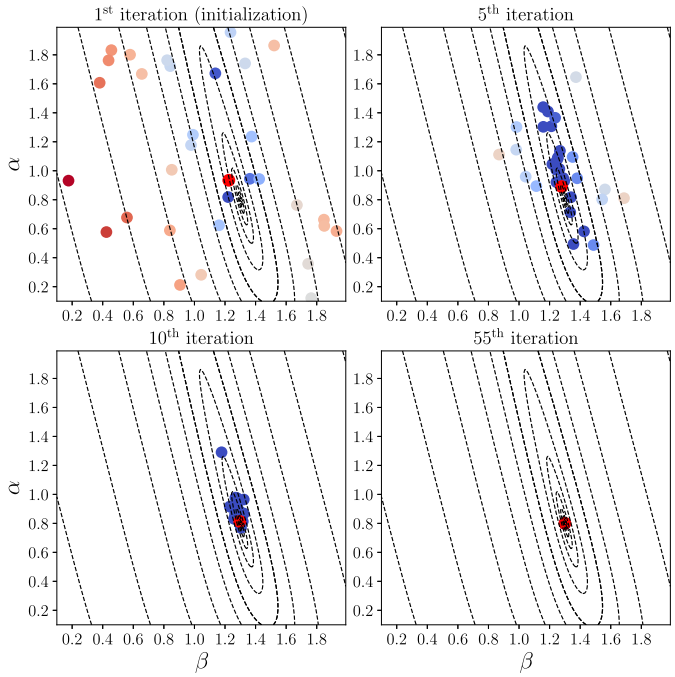

Fig. 17. The swarm particles throughout PSO iterations.

QoI, e.g., the dissipated energy, the solution is readily available and one could simply skip the offline phase.

The developed PGD solver begins with the computation of eigenpair approximations, as in Modal Decomposition. The mechanical behavior of the structure could be efficiently assessed for several damping parameters using modal superposition instead of PGD. However, as previously discussed in [39], the drawback of the method is that many of the Ritz vectors are irrelevant in describing the behavior of the structure, since they do not account for external loads. We actually computed that, for $2n = 33{,}990$ spatial DOFs, the computation of around 16,000 Ritz vectors (approximately 4 h on the computing hub) was needed to recover the same accuracy as that provided by the PGD approach using 100 modes. This shortcoming in the modal superposition method can be mitigated by considering additional modes, referred to in the literature as static corrections [52]. In that regard, it also seems interesting to enrich the subspace spanned by the Ritz vectors with such static corrections in the context of the PGD approach presented herein. It is also worth noting that for larger-scale problems, solving the generalized eigenvalue problem is computationally demanding. To address this challenge, Automated Multi-Level Substructuring (AMLS) [57] can be employed to improve scalability. Studies on vibro-acoustic analysis with FE models exceeding a dozen million degrees of freedom show that AMLS is significantly faster than Lanczos-type methods, especially for fine meshes and a high number of required eigenvectors [58].

Finally, it is important to highlight that the success of PGD depends on the separability of the input data. This is indeed the case here: the external load is explicitly expressed as a sum of products of space and time functions (recall Fig. 5), and the damping operator is written as a sum of products of the damping parameters with the stiffness and mass

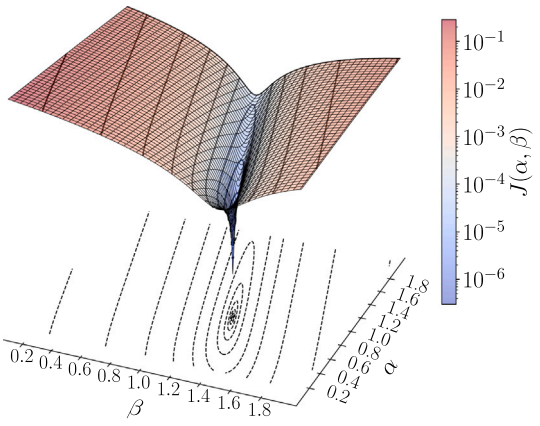

Fig. 18. Cost function J over D.

matrices. However, if the input data is not explicitly separable, one can approximate a separated representation using the SVD or, in the case of geometric parameters for example, employ morphing techniques with respect to a reference mesh [18]. In general, while these tasks are essential to the performance of PGD-based solvers, they are not trivial and can be computationally intensive.

6. Conclusion

We have presented in this paper an extended implementation of the PGD approach to efficiently parameterize reduced-order models with respect to Rayleigh damping coefficients. The results demonstrate that the hybrid solver, based on a two-step prediction–correction process, successfully mitigates computational costs while achieving accuracy comparable to that of full-order solutions. The spatial modes are initially estimated in the subspace formed by the Ritz vectors, as long as the contribution of these modes is significant. When this is no longer the case, the estimates are further refined using a Minimal Residual iterative solver, thereby capturing additional significant information. This hybrid strategy effectively balances the trade-off between computational expense and solution accuracy. Ultimately, the PGD reduced-basis was integrated as a surrogate in the Particle Swarm Optimization algorithm to optimize the damping parameters for a given snapshot, demonstrating the effectiveness of the developed PGD approach for optimization. The numerical experiments confirm that the parametric PGD can accurately build a ROM that includes damping modes and efficiently identifies the damping coefficients used in the snapshot.

CRediT authorship contribution statement

Clément Vella: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Serge Prudhomme:** Writing – review & editing, Supervision, Funding acquisition, Formal analysis, Conceptualization.

Funding

Clément Vella acknowledges the funding awarded by the French Ministry of National Education, Higher Education, Research and Innovation referred to as “Specific Doctoral Contracts for Normaliens”. Serge Prudhomme is grateful for the support from the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant [RGPIN-2019–7154].

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.

Appendix A

The PGD formulation requires the computation of numerous coefficients. Their definitions are enumerated thereafter:

$$k_x = \boldsymbol{\varphi}_q^T K \boldsymbol{\varphi}_q, \quad k_t = \int_{\mathcal{I}} \psi_q^2 \, dt, \quad k_\alpha = \int_{D_\alpha} \xi_q^2 \, d\alpha, \quad k_\beta = \int_{D_\beta} \zeta_q^2 \, d\beta,$$

$$m_x = \boldsymbol{\varphi}_p^T M^{-1} \boldsymbol{\varphi}_p, \quad m_t = \int_{\mathcal{I}} \psi_p^2 \, dt, \quad m_\alpha = \int_{D_\alpha} \xi_p^2 \, d\alpha, \quad m_\beta = \int_{D_\beta} \zeta_p^2 \, d\beta,$$

$$c_x = \boldsymbol{\varphi}_q^T \boldsymbol{\varphi}_p, \quad c_t = \int_{\mathcal{I}} \psi_q \dot{\psi}_p \, dt = \psi_q(T)\psi_p(T) - d_t, \quad c_\alpha = \int_{D_\alpha} \xi_q \xi_p \, d\alpha,$$

$$c_\beta = \int_{D_\beta} \zeta_q \zeta_p \, d\beta, \quad \xi_x = \boldsymbol{\varphi}_q^T M \boldsymbol{\varphi}_q, \quad \xi_t = \int_{\mathcal{I}} \psi_q \dot{\psi}_q \, dt = \frac{1}{2}\psi_q(T)^2,$$

$$\xi_\alpha = \bar{\alpha} \int_{D_\alpha} \alpha \xi_q^2 \, d\alpha, \quad \xi_\beta = \bar{\beta} \int_{D_\beta} \beta \zeta_q^2 \, d\beta.$$

With normalizations, the coefficients k_x, m_x, k_α, m_α, k_β and m_β are all set to unity.

Problem in space. The parameters k_{qq}, m_{qq}, c_{qp}, c_{pq}, and m_{pp} are defined as follows:

$$\begin{aligned}
k_{qq} &= \xi_t \xi_\alpha k_\beta + k_t k_\alpha k_\beta, \\
m_{qq} &= \xi_t k_\alpha \xi_\beta, \\
c_{qp} &= c_t c_\alpha c_\beta, \\
c_{pq} &= d_t c_\alpha c_\beta, \\
m_{pp} &= -m_t m_\alpha m_\beta.
\end{aligned}$$

Problem in time. We assume that $(\boldsymbol{\varphi}_q, \xi_q, \zeta_q)$ and $(\boldsymbol{\varphi}_p, \xi_p, \zeta_p)$ are known and search for ψ_q and ψ_p. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q^* \xi_q \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p^* \xi_p \zeta_p$:

$$\begin{aligned}
&\int_{\mathcal{I}} \psi_q^{*T} \left[c_\alpha c_\beta c_x \dot{\psi}_p + \xi_\alpha k_\beta k_x \dot{\psi}_q + k_\alpha \xi_\beta \xi_x \dot{\psi}_q + k_\alpha k_\beta k_x \psi_q \right] dt \\
&= \int_D \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}_{m-1}^q \, dt d\mu, \quad \forall \psi_q^* \in C^0(\bar{\mathcal{I}}), \\
&\int_{\mathcal{I}} \psi_p^{*T} \left[c_\alpha c_\beta c_x \dot{\psi}_q - m_\alpha m_\beta m_x \psi_p \right] dt \\
&= \int_D \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}_{m-1}^p \, dt d\mu, \quad \forall \psi_p^* \in C^0(\bar{\mathcal{I}}),
\end{aligned}$$

which simplifies to:

$$\begin{aligned}
&c_\alpha c_\beta c_x \dot{\psi}_p + (\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) \dot{\psi}_q + k_\alpha k_\beta k_x \psi_q = b_{\mathcal{T}}^q, \\
&c_\alpha c_\beta c_x \dot{\psi}_q - m_\alpha m_\beta m_x \psi_p = b_{\mathcal{T}}^p,
\end{aligned}$$

with:

$$\begin{aligned}
b_{\mathcal{T}}^q &= \int_D \xi_q \zeta_q \boldsymbol{\varphi}_q^T \boldsymbol{r}_{m-1}^q \, d\mu, \\
b_{\mathcal{T}}^p &= \int_D \xi_p \zeta_p \boldsymbol{\varphi}_p^T \boldsymbol{r}_{m-1}^p \, d\mu.
\end{aligned}$$

Above equations are discretized using the Crank–Nicolson time-marching scheme, such that, given $\boldsymbol{\psi}^0$, one computes the i^{th} iterate ($i > 0$) as:

$$A_{\mathcal{T}} \boldsymbol{\psi}^i = B_{\mathcal{T}} \boldsymbol{\psi}^{i-1} + \boldsymbol{b}_{\mathcal{T}}^i, \qquad i = 1, \dots, n_t,$$

where:

$$A_{\mathcal{T}} = \begin{bmatrix} h_t k_\alpha k_\beta k_x + 2(\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) & 2c_\alpha c_\beta c_x \\ 2c_\alpha c_\beta c_x & -h_t m_\alpha m_\beta m_x \end{bmatrix},$$

$$B_{\mathcal{T}} = \begin{bmatrix} -h_t k_\alpha k_\beta k_x + 2(\xi_\alpha k_\beta k_x + k_\alpha \xi_\beta \xi_x) & 2c_\alpha c_\beta c_x \\ 2c_\alpha c_\beta c_x & h_t m_\alpha m_\beta m_x \end{bmatrix},$$

$$\boldsymbol{b}^i_{\mathcal{T}} = h_t \begin{bmatrix} b^{q,i}_{\mathcal{T}} + b^{q,i-1}_{\mathcal{T}} \\ b^{p,i}_{\mathcal{T}} + b^{q,i-1}_{\mathcal{T}} \end{bmatrix}.$$

Problem in damping parameter α. We assume that $(\boldsymbol{\varphi}_q, \psi_q, \zeta_q)$ and $(\boldsymbol{\varphi}_p, \psi_p, \zeta_p)$ are known and search for ξ_q and ξ_p. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q \xi^*_q \zeta_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p \xi^*_p \zeta_p$:

$$\begin{aligned} &\int_{D_\alpha} \xi^*_q \left(c_t c_\beta c_x \xi_p + \xi_t k_\beta k_x \alpha\bar{\alpha} \xi_q + \xi_t \xi_\beta \xi_x f_q + k_t k_\beta k_x \xi_q \right) d\alpha \\ &= \int_D \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}^q_{m-1} \, dt d\mu, \quad \forall \xi^*_q \in C^0(D_\alpha), \\ &\int_{D_\alpha} \xi^*_p \left(d_t c_\beta c_x \xi_q - m_t m_\beta m_x \xi_p \right) d\alpha \\ &= \int_D \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}^p_{m-1} \, dt d\mu, \quad \forall \xi^*_q \in C^0(D_\alpha), \end{aligned}$$

which simplifies to:

$$\begin{aligned} &c_t c_\beta c_x \xi_p + (\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) \xi_q = b^q_\xi, \\ &d_t c_\beta c_x \xi_q - m_t m_\beta m_x \xi_p = b^p_\xi, \end{aligned}$$

with:

$$\begin{aligned} b^q_\xi &= \int_{D_\beta} \int_{\mathcal{I}} \psi_q \zeta_q \boldsymbol{\varphi}^T_q \boldsymbol{r}^q_{m-1} \, dt d\beta, \\ b^p_\xi &= \int_{D_\beta} \int_{\mathcal{I}} \psi_p \zeta_p \boldsymbol{\varphi}^T_p \boldsymbol{r}^p_{m-1} \, dt d\beta. \end{aligned}$$

The solutions read, with dependencies on α explicitly written:

$$\begin{aligned} \xi_q(\alpha) &= \frac{c_t c_\beta c_x b^p_\xi(\alpha) + m_t m_\beta m_x b^q_\xi(\alpha)}{(\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) m_t m_\beta m_x + c_t d_t (c_\beta c_x)^2}, \\ \xi_p(\alpha) &= \frac{d_t c_\beta c_x b^q_\xi(\alpha) - (\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) b^p_\xi(\alpha)}{(\xi_t \xi_\beta \xi_x + k_t k_\beta k_x + \xi_t k_\beta k_x \alpha\bar{\alpha}) m_t m_\beta m_x + c_t d_t (c_\beta c_x)^2}. \end{aligned}$$

Problem in parameter β. We assume that $(\boldsymbol{\varphi}_q, \psi_q, \xi_q)$ and $(\boldsymbol{\varphi}_p, \psi_p, \xi_p)$ are known and search for ζ_q and ζ_p. We choose test functions in the form $\boldsymbol{q}^* = \boldsymbol{\varphi}_q \psi_q \xi_q \zeta^*_q$ and $\boldsymbol{p}^* = \boldsymbol{\varphi}_p \psi_p \xi_p \zeta^*_p$:

$$\begin{aligned} &\int_{D_\beta} \zeta^*_q \left[c_t c_\alpha c_x \zeta_p + \xi_t \xi_\alpha k_x \zeta_q + \xi_t k_\alpha \xi_x \beta\bar{\beta} \zeta_q + k_t k_\alpha k_x \zeta_q \right] d\beta \\ &\quad = \int_D \int_{\mathcal{I}} \boldsymbol{q}^{*T} \boldsymbol{r}^q_{m-1} \, dt d\mu, \quad \forall \zeta^*_q \in C^0(D_\beta), \\ &\int_{D_\beta} \zeta^*_p \left[d_t c_\alpha c_x \zeta_q - m_t m_\alpha m_x \zeta_p \right] d\beta \\ &\quad = \int_D \int_{\mathcal{I}} \boldsymbol{p}^{*T} \boldsymbol{r}^p_{m-1} \, dt d\mu, \quad \forall \zeta^*_q \in C^0(D_\beta), \end{aligned}$$

which simplifies to:

$$\begin{aligned} &c_t c_\alpha c_x \zeta_p + (\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) \zeta_q = b^q_\zeta, \\ &d_t c_\alpha c_x \zeta_q - m_t m_\alpha m_x \zeta_p = b^p_\zeta, \end{aligned}$$

with:

$$\begin{aligned} b^q_\zeta &= \int_{D_\alpha} \int_{\mathcal{I}} \psi_q \xi_q \boldsymbol{\varphi}^T_q \boldsymbol{r}^q_{m-1} \, dt d\alpha, \\ b^p_\zeta &= \int_{D_\alpha} \int_{\mathcal{I}} \psi_p \xi_p \boldsymbol{\varphi}^T_p \boldsymbol{r}^p_{m-1} \, dt d\alpha. \end{aligned}$$

The solutions read, with dependencies on β explicitly written:

$$\begin{aligned} \zeta_q(\beta) &= \frac{c_t c_\alpha c_x b^p_\zeta(\beta) + m_t m_\alpha m_x b^q_\zeta(\beta)}{(\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) m_t m_\alpha m_x + c_t d_t (c_\alpha c_x)^2}, \\ \zeta_p(\beta) &= \frac{d_t c_\alpha c_x b^q_\zeta(\beta) - (\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) b^p_\zeta(\beta)}{(\xi_t \xi_\alpha k_x + k_t k_\alpha k_x + \xi_t k_\alpha \xi_x \beta\bar{\beta}) m_t m_\alpha m_x + c_t d_t (c_\alpha c_x)^2}. \end{aligned}$$

Data availability

No data was used for the research described in the article.

References

[1] Benner P, Gugercin S, Willcox KE. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 2015;57:483–531.
[2] Bamer F, Bucher C. Application of the proper orthogonal decomposition for linear and nonlinear structures under transient excitation. Acta Mech 2012;223.
[3] Kerschen G, Golinval J-C, Vakakis A, Bergman L. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. Nonlinear Dyn 2005;41:147–69.
[4] Lu K, Jin Y, Chen Y, Yang Y, Hou L, Zhang Z, et al. Review for order reduction based on proper orthogonal decomposition and outlooks of applications in mechanical systems. Mech Syst Signal Process 2019;123:264–97.
[5] Hoang K, Kerfriden P, Bordas S. Space-time goal-oriented reduced basis approximation for linear wave equation; 2013. ArXiv: 1305.3528.
[6] Hoang KC, Kerfriden P, Khoo B, Bordas S. An efficient goal-oriented sampling strategy using reduced basis method for parametrized elastodynamic problems. Numer Methods Partial Differ Equ 2014;31.
[7] Fischer H, Roth J, Wick T, Chamoin L, Fau A. More DWR: space-time goal-oriented error control for incremental POD-based ROM for time-averaged goal functionals. J Comput Phys 2024;504:112863.
[8] Boyaval S, Le Bris C, Lelièvre T, Maday Y, Nguyen NC, Patera AT. Reduced basis techniques for stochastic problems. Arch Comput Methods Eng 2010;17(4):435–54.
[9] Rozza G, Huynh DBP, Patera AT. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Arch Comput Methods Eng 2008;15(3):229–75.
[10] Ladevèze P. PGD in linear and nonlinear computational solid mechanics. In: Courses and lectures. CISM International Centre for Mechanical Sciences; 1999. p. 91–152.
[11] Chinesta F, Keunings R, Leygue A. The proper generalized decomposition for advanced numerical simulations: a primer. Springer; 2014.
[12] Boucinha L, Gravouil A, Ammar A. Space–time proper generalized decompositions for the resolution of transient elastodynamic models. Comput Methods Appl Mech Eng 2013;255:67–88.
[13] Boucinha L, Ammar A, Gravouil A, Nouy A. Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models—application to transient elastodynamics in space-time domain. Comput Methods Appl Mech Eng 2014;273:56–76.
[14] Boucinha L. Réduction de modèle a priori par séparation de variables espace-temps : Application en dynamique transitoire. [Theses, INSA de Lyon]; 2013. p. 163–66.
[15] Bamer F, Shirafkan N, Cao X, Oueslati A, Stoffel M, de Saxcé G, et al. A newmark space-time formulation in structural dynamics. Comput Mech 2021:1–18.
[16] Arjoune T, Markert B, Bamer F. Non-incremental response evaluation in geometrically nonlinear structural dynamics using a space-time stiffness operator. Comput Mech 2022;70:04.
[17] Goutaudier D, Berthe L, Chinesta F. Proper generalized decomposition with time adaptive space separation for transient wave propagation problems in separable domains. Comput Methods Appl Mech Eng 2021;380:113755.
[18] Cavaliere F, Zlotnik S, Sevilla R, Larráyoz X, Díez P. Nonintrusive reduced order model for parametric solutions of inertia relief problems. Int J Numer Methods Eng 2021;122(16):4270–91.
[19] Díez P, Zlotnik S, García A, Huerta A. Encapsulated PGD algebraic toolbox operating with high-dimensional data. Arch Comput Methods Eng 2019;27.
[20] Cavaliere F, Zlotnik S, Sevilla R, Larrayoz X, Díez P. Nonintrusive parametric solutions in structural dynamics. Comput Methods Appl Mech Eng 2022;389:114336.
[21] Barbarulo A, Riou H, Kovalevsky L, Ladevèze P. PGD-VTCR: a reduced order model technique to solve medium frequency broad band problems on complex acoustical systems. Strojniški Vestnik J Mech Eng 2015;60:307–13.
[22] De Brabander P, Allix O, Ladèveze P, Hubert P, Thévenet P. On a wave-based reduced order model for transient effects computation including mid frequencies. Comput Methods Appl Mech Eng 2022;395:114990.
[23] Germoso C, Aguado JV, Fraile A, Alarcon E, Chinesta F. Efficient pgd-based dynamic calculation of non-linear soil behavior. Comptes Rendus Mécanique 2016;344(1):24–41.
[24] Germoso C, Duval JL, Chinesta F. Harmonic-modal hybrid reduced order model for the efficient integration of non-linear soil dynamics. Appl Sci 2020;10(19):6778.

[25] Malik MH, Borzacchiello D, Aguado JV, Chinesta F. Advanced parametric space-frequency separated representations in structural dynamics: a harmonic–modal hybrid approach. Comptes Rendus Mécanique 2018;346(7):590–602.
[26] Quaranta G, Argerich Martin C, Ibañez R, Duval JL, Cueto E, Chinesta F. From linear to nonlinear PGD-based parametric structural dynamics. Comptes Rendus Mécanique 2019;347(5):445–54.
[27] Malik MH, Borzacchiello D, Chinesta F, Diez P. Inclusion of frequency-dependent parameters in power transmission lines simulation using harmonic analysis and proper generalized decomposition. Int J Numer Modell Electron Netw Devices Fields 2018;31(5):e2331.
[28] Rishmawi S, Rodriguez S, Chinesta F, Gosselin F. Harmonic-modal hybrid frequency approach for parameterized non-linear dynamics. Nonlinear Dyn 2024;301:107461.
[29] Rishmawi S, Moyne L, Serroud S, Rodríguez S, Chinesta F, Tuysuz O, et al. Parameter identification of a nonlinear vertical axis rotating machine through reduced order modeling and data assimilation. Nonlinear Dyn 2025;1–20.
[30] Germoso C, Rodriguez S, Pasquale A, Chinesta F. Fast dynamical simulation combining harmonic-modal hybrid formulations with space separated representations. Results Eng 2025;25:103897.
[31] Moaven A, Massart TJ, Zlotnik S. Parametric solutions of coupled thermo-hydro-Mechanical problems in real time with proper generalized decomposition. Finite Elem Anal Des 2025;247:104352.
[32] Cho J, Nam S, Yang H, Yun S-B, Hong Y, Park E. Separable physics-informed neural networks. Adv Neural Inf Process Syst 2023.
[33] Mandl L, Goswami S, Lambers L, Ricken T. Separable physics-informed DeepONet: breaking the curse of dimensionality in physics-informed machine learning. Comput Methods Appl Mech Eng 2025;434:117586.
[34] Ghnatios C, Chinesta F. A parsimonious separated representation empowering PINN–PGD-based solutions for parametrized partial differential equations. Mathematics 2024 07;12:2365.
[35] de Silva V, Lim L-H. Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM J Matrix Anal Appl 2008;30(3):1084–127.
[36] Musharbash E, Nobile F, Vidlicková E. Symplectic dynamical low rank approximation of wave equations with random parameters. BIT Numer Math 2020:1–49.
[37] Vella C, Prudhomme S. PGD reduced-order modeling for structural dynamics applications. Comput Methods Appl Mech Eng 2022;402:115736.
[38] Nouy A. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Comput Methods Appl Mech Eng 2010;199(23):1603–26.
[39] Vella C, Gosselet P, Prudhomme S. An efficient PGD solver for structural dynamics applications. Adv Model Simul Eng Sci 2024;11:15.
[40] Semblat JF. Rheological interpretation of Rayleigh damping. J Sound Vib 1997;206(5):741–44.
[41] Kareem A, Sun W-J. Dynamic response of structures with uncertain damping. Eng Struct 1990;12(1):2–8.
[42] Zienkiewicz O, Taylor R, Zhu J. The finite element method: its basis and fundamentals. *7th* ed. Butterworth-Heinemann; 2013.
[43] Sánchez Iglesias F, Fernández López A. Rayleigh damping parameters estimation using hammer impact tests. Mech Syst Signal Process 2020;135:106391.
[44] Singh S, Mastrogiuseppe M. Multibody modelling of tether and capture system for dynamic simulations of in-air capturing. Acta Astronaut 2024;218:59–69.
[45] Beattie CA, Gugercin S. Krylov-based model reduction of second-order systems with proportional damping. In: Proceedings of the 44th IEEE conference on decision and control; 2005. p. 2278–83.
[46] Meerbergen K. Fast frequency response computation for Rayleigh damping. Int J Numer Methods Eng 2008;73(1):96–106.
[47] Wu B, Yang S, Li Z. An algorithm for solving frequency responses of a system with Rayleigh damping. Arch Appl Mech 2016;86.
[48] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95 - international conference on neural networks, vol. 4; 1995. p. 1942–48.
[49] Shi Y, Eberhart R. A modified particle swarm optimizer. In: 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No.98TH8360); 1998. p. 69–73.
[50] Keane AJ, Nair PB. Design in the presence of uncertainty, ch. 8. John Wiley & Sons, Ltd; 2005. p. 327–57.
[51] Chen Y, Davis TA, Hager WW, Rajamanickam S. Algorithm 887: CHOLMOD, supernodal sparse cholesky factorization and update/downdate. ACM Trans Math Softw 2008;35(3).
[52] Géradin M, Rixen DJ. Mechanical vibrations: theory and application to structural dynamics. Wiley; 1994.
[53] Sanmugadas V, Agarwal M, Borwankar P, Kapania RK. Parametric model order reduction for structural optimization of fiber composite structures. AIAA J 2025;1–16.
[54] Huang H, Qiu J, Riedl K. On the global convergence of particle swarm optimization methods. Appl Math Optim 2023;88:05.
[55] Virtanen P, Gommers R, Oliphant TE, Haberland M, Reddy T, Cournapeau D, Burovski E, Peterson P, Weckesser W, Bright J, van der Walt SJ, Brett M, Wilson J, Millman KJ, Mayorov N, Nelson ARJ, Jones E, Kern R, Larson E, Carey CJ, Polat I, Feng Y, Moore EW, VanderPlas J, Laxalde D, Perktold J, Cimrman R, Henriksen I, Quintero EA, Harris CR, Archibald AM, Ribeiro AH, Pedregosa F, van Mulbregt P, SciPy 1.0 Contributors, et al. SciPy 1.0: fundamental algorithms for scientific computing in python. Nat Methods 2020;17:261–72.
[56] Miranda LJV. PySwarms, a research-toolkit for particle swarm optimization in python. J Open Source Softw 2018;3.
[57] Hetmaniuk UL, Lehoucq RB. Multilevel methods for eigenspace computations in structural dynamics. In Widlund OB, Keyes DE, editors. Domain decomposition methods in science and engineering XVI. Berlin Heidelberg: Springer; 2007. p. 103–13.
[58] Kropp A, Heiserer D. Efficient broadband vibro-acoustic analysis of passenger car bodies using an fe-based component mode synthesis approach. J Comput Acoust 2003;11(2):139–57.