Proper Generalized Decomposition solution of the parameterized Helmholtz problem: application to inverse geophysical problems.

Marianna Signorini*[1], Sergio Zlotnik[2] and Pedro Díez[2]

[1] *MOX– Modellistica e Calcolo Scientifico, Dipartimento di Matematica "F. Brioschi", Politecnico di Milano*
[2] *Laboratori de Càlcul Numèric (LaCàN), Universitat Politècnica de Catalunya, Jordi Girona 1, E-08034 Barcelona, Spain*

SUMMARY

The identification of the geological structure from seismic data is formulated as an inverse problem. The properties and the shape of the rock formations in the subsoil are described by material and geometric parameters, which are taken as input data for a predictive model. Here, the model is based on the Helmholtz equation, describing the acoustic response of the system for a given wave length. Thus, the inverse problem consists in identifying the values of these parameters such that the output of the model agrees the best with observations. This optimization algorithm requires multiple queries to the model with different values of the parameters. Reduced Order Models are especially well suited to significantly reduce the computational overhead of the multiple evaluations of the model.
In particular, the Proper Generalized Decomposition (PGD) produces a solution explicitly stating the parametric dependence, where the parameters play the same role as the physical coordinates. A PGD solver is devised to inexpensively explore the parametric space along the iterative process. This exploration of the parametric space is in fact seen as a post-process of the generalized solution. The approach adopted demonstrates its viability when tested in two illustrative examples.

KEY WORDS: parameterized Helmholtz problem; Proper Generalised decomposition (PGD); parameter identification; inverse problems; seismic analysis

INTRODUCTION

Seismic inversion and parameter identification is a very important task in geophysics. In particular, geologists are interested in understanding subsoil structures and layers, especially their physical properties and dimensions. These properties are usually inferred from acoustic data, originated from land or marine surveys. Different inversion techniques have been developed during the years in order to deal with this problem [1, 2, 3].

The pressure field produced during the explorations is typically modelled with the transient wave equation and the inversion is performed via the minimization of a suitable functional depending on the records of the waves on the ground surface. We consider the Helmholtz equation in order to properly describe the stationary pressure field during a seismic survey performed, for example, with the vibroseis technique. The Helmholtz equation describes the phenomenon in a steady state case

*Correspondence to: Marianna Signorini, marianna.signorini@polimi.it

Contract/grant sponsor: Spanish Ministry of Economy and Competitiveness; contract/grant number: CICYT-DPI2014-51844-C2-2-R

as a sum of harmonic waves and it is used in different fields in order to describe the physics of the problem, such as in acoustics and seismology [4, 5, 6].

We deal with the identification of a number of parameters, that are necessary in order to describe specific geometric or material properties of the subsoil. We perform the parameter identification by minimizing a proper functional that takes into account the difference between the recorded solution and the numerical one, as usually done. The minimization of such a functional, nevertheless, requires multiple evaluations of the solution for slightly different values of the parameters, in order to explore the space of design parameters (the ones to be identified). Moreover, the cost functional may have many local minima and various minimizations could be required (starting from different parametric guess). Therefore, we need an efficient numerical tool that allows us to evaluate the solution of the Helmholtz equation for different choices of the design parameters in an efficient way.

In order to have an easily evaluable solution of the problem, the idea is to solve the more complex parameterized Helmholtz equation via a Reduced Order Model (ROM) method. (Examples of ROM methods are, for example, the Proper Orthogonal Decomposition (POD) method [7] or the Reduced Basis method [8]). With these kind of methods the parametric solution is produced during a costly offline phase, but the solution is easily evaluated in real-time during the online phase, namely during the minimization procedure. Moreover it is possible to evaluate the solution in all the points of the parametric space with relatively low computational cost. This means that we can have an idea on the influence of the different parameters on the solution restricteld to the observation boundary. Among the ROM methods, we use the Proper Generalized Decomposition (PGD) method, described in [9, 10]. The use of PGD with parameters controlling geometries has been proposed in [11] to control the external boundary of the domain and later extended in [12] to the parameterization of internal interfases. Moreover, parametric solutions involving frequency (with fixed domain) of transient heat problems are described in [13]. Finally, PGD has also been used recently by Germoso et al. [14] to account for non-linear soil mechanics during a seismic event.

Concerning the algorithm applied in order to perform the minimization, we use the matlab routine `fmincon`, suitable for constrained minimization problems. In particular the method implemented is the Sequential Quadratic Programming (SQP) algorithm, described in [15]. The constraints on the parameters corresponds to the boundaries of the parametric space in which the parametric solution of the Helmholtz problem is defined.

We underline that the proposed methodology is general, so it could be used also for real-time monitoring of the electromagnetic field of electronic devices or simply to obtain a parametric and real-time evaluable solution of the Helmholtz equation in other contexts. In this work we apply the method to two test cases, a toy example and a more realistic one. We compare the performance of the PGD method and of the inversion with different options, such as the solution of the problem with or without compression or the inverse problem with various choices of the reference values.

This work is organized as follows.

In the first chapter the direct and the inverse problem are described. In particular, the parameterized Helmholtz equation is introduced and the extended weak formulation of the parameterized version of the associated boundary value problem is given.

The second chapter is devoted to the application of the PGD technique to the direct problem. A separable expression of the solution is assumed and the separability of the extended weak formulation is discussed. The alternated directions fixed-point algorithm, proper of the PGD method ([12]) is presented for this specific case in both its continuous and discretized version. The definition of the required matrices can be found in the Appendix.

In the third chapter some computational aspects are formalized. One first big issue is how to separate input data and the choice of the sampling parametric points (discussed also in the numerical results). This problem is faced in the first two paragraphs of the chapter. In the third one the parallel implementation is proposed for the separation of input data. In the remainder of the chapter the focus is on the importance of compressing the PGD solution. Two alternative methods are introduced here and compared in the numerical results. Finally the choice of the solver for the inverse problem is described.

The numerical results are presented in the forth chapter. Two test cases are considered. The first test case is a synthetic example, while the second is a more realistic one, a simplified version of a real ground profile described in [16].

1. PROBLEM STATEMENT

1.1. Direct Problem: the Helmholtz Equation

The Boundary Value Problem (BVP) under consideration consists in complementing the Helmholtz equation with proper boundary conditions and reads: find a complex-valued function p taking values in Ω (see Figure 1) such that

$$\begin{cases} -\sigma\omega^2 p - \nabla\cdot(\nabla p) = 0 & \text{in } \Omega \\ \nabla p\cdot\mathbf{n} - i\sqrt{\sigma}\omega p = 0 & \text{on } \Gamma_R \\ \nabla p\cdot\mathbf{n} = g & \text{on } \Gamma_N \end{cases}, \tag{1}$$

where p is the unknown complex pressure field, ω is the given angular frequency, σ is the squared slowness of the medium (equal to the squared inverse of the propagation velocity of the waves in the medium) and g represents a sound source on the boundary. For $g = 0$, Γ_N is a reflecting boundary. The boundary condition on Γ_R represents full absorption of the sound.

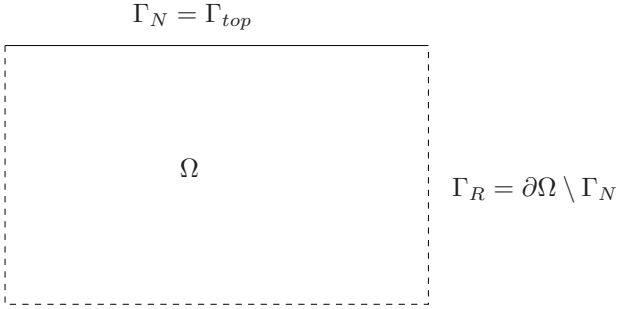


Figure 1. Domain

Accordingly, the standard weak formulation of the problem reads: find $p \in H^1(\Omega;\mathbb{C})$ such that

$$a(p,q) = l(q) \quad \forall q \in H^1(\Omega;\mathbb{C}) \tag{2}$$

where the bilinear and the linear forms $a(\cdot,\cdot)$ and $l(\cdot)$ are defined as:

$$a(p,q) = -\int_\Omega \sigma\omega^2 p\bar{q}\,d\mathbf{x} + \int_\Omega \nabla p\cdot\nabla\bar{q}\,d\mathbf{x} - i\int_{\Gamma_R}\sqrt{\sigma}\omega p\bar{q}\,ds \tag{3}$$

$$l(q) = \int_{\Gamma_N} g\bar{q}\,ds \tag{4}$$

being $\bar{q}$ the complex conjugate of q.

1.2. Parameterization of the Problem

The field of material properties, σ, is assumed to be piecewise constant. Thus, σ is characterized by 1) the shape, size and location of the different rock formations (that is, geometric parameters describing a set of subdomains where the material properties are uniform) and 2) the material properties of each type of rock (that is, the material parameters).

Accordingly, we assume that, on the one hand, there are n_θ geometric parameters

$$\theta_j \in I_{\theta_j} \qquad \text{for } j = 1,\dots,n_\theta\ .$$

describing the shape of subdomains in Ω with uniform propagation velocity. On the other hand we consider n_σ material parameters

$$\sigma_s \in I_{\sigma_s} \qquad \text{for } s = 1, \dots, n_\sigma \ .$$

representing the squared slowness, uniform in one or more subdomains. The intervals I_{θ_j} and I_{σ_s} stand for the expected ranges of variation of the parameters. We aim at providing the solution for all the values in these ranges.

The procedure for parameter identification requires multiple queries to the model, each with a different set of parametric values. We aim at producing an offline computed generalized solution, explicitly stating the parametric dependence of p (both for geometric and material parameters). Once this solution is available, exploring the parametric space does not require successive computations of the model but simple and fast evaluations as a post-process.

The compact notation $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$ is used to denote the geometric and material parameters, respectively. Consequently, the spaces where these parameters lie read

$$I_{\boldsymbol{\theta}} = I_{\theta_1} \times \cdots \times I_{\theta_{n_\theta}} \quad \text{and} \quad I_{\boldsymbol{\sigma}} = I_{\sigma_1} \times \cdots \times I_{\sigma_{n_\sigma}} \tag{5}$$

The input data of the problem, σ, depends on $(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})$ and therefore the parametric solution $p(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})$ takes values on $\mathcal{D} = \Omega \times I_{\boldsymbol{\theta}} \times I_{\boldsymbol{\sigma}}$.

As it is standard in the PGD setup, the extended weak formulation of the parametrized version of (1) reads: find $p \in H^1(\Omega) \otimes L^2(I_{\boldsymbol{\theta}}) \otimes L^2(I_{\boldsymbol{\sigma}})$ such that

$$A(p, q) = L(q) \quad \forall q \in H^1(\Omega) \otimes L^2(I_{\boldsymbol{\theta}}) \otimes L^2(I_{\boldsymbol{\sigma}}) \ , \tag{6}$$

where the bilinear and linear forms are:

$$A(p, q) = \int_{I_{\boldsymbol{\sigma}}} \int_{I_{\boldsymbol{\theta}}} a(p, q) \, d\boldsymbol{\theta} \, d\boldsymbol{\sigma} \quad \text{and} \quad L(q) = \int_{I_{\boldsymbol{\sigma}}} \int_{I_{\boldsymbol{\theta}}} l(q) \, d\boldsymbol{\theta} \, d\boldsymbol{\sigma} \tag{7}$$

It is worth noting that the tensor product $H^1(\Omega) \otimes L^2(I_{\boldsymbol{\theta}}) \otimes L^2(I_{\boldsymbol{\sigma}})$ does not assume the separability of the functions and therefore the complexity of the described problem increases with the number of parametric dimensions.

1.3. Inverse Problem

Seismic inversion is a well-known problem in geophysics [3, 2, 17, 18], consisting in recovering the ground velocity profile (distribution of material properties), from a set of measures on the surface during seismic explorations (experimental observations). Different techniques have been developed in order to deal with this inverse problem. Despite the problem may be ill-posed because different velocity profiles can explain the same observations, often these techniques allow determining the main features of the geologic profile. As indicated above, the inverse problem involves an optimization procedure requiring multiple queries and therefore the evaluation of the solution of the problem for a large number of slightly different subsoil configurations.

In the previous section, a simplified geological model is proposed taking into account the main features of the geologic profile and describing the principal unknown characteristics of the subsoil with a set of geometric and material parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$. Thus, the inverse problem consists in identifying $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$ by minimizing a suitable cost functional, e.g. the same used in the full waveform inversion method [3]. This cost functional accounts for the misfit between the observations and the computed outcome of the parametric model, measured with a properly defined norm (typically a L^2 norm restricted to the boundary where the observations are taken).

Let p_{obs} denote the observed data in a portion of the surface Γ_{obs}. The cost functional $J(\boldsymbol{\theta}, \boldsymbol{\sigma})$ is introduced as

$$J(\boldsymbol{\theta}, \boldsymbol{\sigma}) = \int_{\Gamma_{obs}} (p(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})|_{\Gamma_{obs}} - p_{obs})^2 \, ds. \tag{8}$$

The minimization problem reads: find $(\boldsymbol{\theta}, \boldsymbol{\sigma})$ such that

$$(\boldsymbol{\theta}, \boldsymbol{\sigma}) = \underset{\boldsymbol{\theta} \in I_{\boldsymbol{\theta}}; \boldsymbol{\sigma} \in I_{\boldsymbol{\sigma}}}{\operatorname{argmin}} \; J(\boldsymbol{\theta}, \boldsymbol{\sigma}) \; ,$$

Solving the minimization problem requires multiple evaluation of the solution p of the problem (1). As described in the next section, the PGD technique provides a solution $p_{\text{PGD}}(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma}) \simeq p(\boldsymbol{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})$ with explicitly parametric dependence.

2. NUMERICAL METHODOLOGY

2.1. *PGD formulation*

The Proper Generalized Decomposition (PGD) method [12, 10] is particularized to the Helmholtz equation with geometric and material parameters. The standard PGD formulation in briefly recalled before concentrating in this specific case.

The ideas behind the PGD technique are readily summarized as follows: 1) the solution is assumed to be separable (expressed as a sum of terms consisting in a product of functions depending of one parameter each), 2) a greedy algorithm is used to compute the terms sequentially and 3) each term is computed iterating in each parametric dimension using an alternated directions fixed-point algorithm. The two first items are discussed here while the third is illustrated in section 2.2.

The separability assumption states that the solution $p(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})$ of the problem (6) is approximated by a sum of n separated terms (also denoted by modes), namely

$$\begin{aligned} p(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\sigma}) \approx p^n_{\text{PGD}}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\sigma}) &= \sum_{m=1}^{n} \left(F^m_x(\mathbf{x}) \prod_{j=1}^{n_\theta} F^m_{\theta_j}(\theta_j) \prod_{s=1}^{n_\sigma} F^m_{\sigma_s}(\sigma_s) \right) \\ &= p^{n-1}_{\text{PGD}}(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\sigma}) + F^n_x(\mathbf{x}) \prod_{j=1}^{n_\theta} F^n_{\theta_j}(\theta_j) \prod_{s=1}^{n_\sigma} F^n_{\sigma_s}(\sigma_s). \end{aligned} \tag{9}$$

where, in the case under consideration, functions F^m_x, F_{θ_j} and F_{σ_s} take complex values. Note that an alternative option is accounting for the complex character only with one function (e.g. F^m_x) and assuming $F^m_{\theta_j}$ and $F^m_{\sigma_s}$ to be real. This would reduce the richness of the proposed approximation and would require a larger number of modes to achieve the same accuracy. We notice that the implementation with Matlab® does not have any additional complexity with respect to the case in which the unknowns are real numbers.

The fundamental concepts and the notation to derive PGD for both material and geometric parameters are briefly recalled following [12].

The essential idea is making the geometric parameters $\boldsymbol{\theta}$ to appear explicitly in the weak formulation (6). This is a natural feature for the material parameters $\boldsymbol{\sigma}$, but the effect of $\boldsymbol{\theta}$ in (6) is hidden, affecting the description of the shape of Ω and the different material subdomains (internal boundaries).

For the sake of a simpler presentation, the boundary source term g is taken independent of $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$ and such that the geometric parameters $\boldsymbol{\theta}$ do not affect the boundary of Ω, only the internal boundaries (the identification of shape and location of the rock formations). On the contrary, the material property $\sigma(\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\sigma})$ has a an explicit parametric dependence described in Appendix A.

The following strategy is adopted to make the geometric parameters $\boldsymbol{\theta}$ appear explicitly in the formulation. It is a generalization to the geometrical parameterization proposed in [11]. The domain Ω is partitioned into a set of simple (triangular) macro-elements $T_1, \cdots, T_{n_T}$, having shapes depending on $\boldsymbol{\theta}$. The macro-elements are inside the material subdomains (no macro-element is overlapping two material subdomains) and are located in order to properly describe the internal boundaries. A reference element $\hat{T}$ is introduced such that each macro-element T_e, $e = 1, \ldots, n_T$, is assumed to be the image of $\hat{T}$ by an *isogeometric* mapping depending on $\boldsymbol{\theta}$,

$$\begin{aligned} \Psi_e : \hat{T} &\to T_e \\ \hat{\mathbf{x}} &\mapsto \mathbf{x} = \Psi_e(\hat{\mathbf{x}}). \end{aligned}$$

The mappings Ψ_e are explicitly dependent of $\boldsymbol{\theta}$ and therefore also their Jacobian matrices, denoted by $\mathbf{J}_e(\boldsymbol{\theta})$, do.

The body integrals appearing in (6) are rewritten accounting for the mappings and showing the explicit dependence on the parameters

$$\begin{aligned} A(p,q) =& \int_{I_{\boldsymbol{\theta}}}\int_{I_{\boldsymbol{\sigma}}}\int_{\Omega}\left(-\sigma\omega^2 p\bar{q} + \nabla p\cdot\nabla\bar{q}\right)\, d\mathbf{x}\, d\boldsymbol{\theta}\, d\boldsymbol{\sigma} \\ &- i\int_{I_{\boldsymbol{\theta}}}\int_{I_{\boldsymbol{\sigma}}}\int_{\Gamma_R}\sqrt{\sigma}\omega p\bar{q}\, ds\, d\boldsymbol{\theta}\, d\boldsymbol{\sigma} \\ =& \sum_{e=1}^{n_T}\int_{I_{\boldsymbol{\theta}}}\int_{I_{\boldsymbol{\sigma}}}\int_{\hat{T}}\left(-\sigma\omega^2 p_e\bar{q}_e|\mathbf{J}_e(\boldsymbol{\theta})| + \nabla_{\hat{\mathbf{x}}}p_e\cdot(\boldsymbol{D}_e(\boldsymbol{\theta})\nabla_{\hat{\mathbf{x}}}\bar{q}_e)\right)\, d\hat{\mathbf{x}}\, d\boldsymbol{\theta}\, d\boldsymbol{\sigma} \\ &- i\int_{I_{\boldsymbol{\theta}}}\int_{I_{\boldsymbol{\sigma}}}\int_{\Gamma_R}\sqrt{\sigma}\omega p\bar{q}\, ds\, d\boldsymbol{\theta}\, d\boldsymbol{\sigma}, \end{aligned} \tag{10}$$

where $\boldsymbol{D}_e := |\boldsymbol{J}_e|\boldsymbol{J}_e^{-\mathsf{T}}\boldsymbol{J}_e^{-1}$, p_e is the restriction of p to T_e such that

$$p = \sum_{e=1}^{n_T} p_e\circ\Psi_e^{-1},$$

and analogously for q_e.

The input data of the problem describing the bilinear form $A(\cdot,\cdot)$ has to be expressed in a separable form in terms of the selected parameters, $\boldsymbol{\theta}$ and $\boldsymbol{\sigma}$. This is natural for the material parameters (σ is straightforwardly expressed as a separated function of $\boldsymbol{\sigma}$). For the geometric parameters, separable representations of the determinant of the Jacobian matrix, $|\boldsymbol{J}_e|$, and matrix $\boldsymbol{D}_e$ are required. In practice, this means obtaining the expressions given in the Appendix A in (15). The procedure is detailed in Section 3.1.

2.2. *Alternated directions fixed-point iterations*

The PGD greedy algorithm consists in computing sequentially p^n_{PGD} for $n = 1, 2, \ldots$ to reach a satisfactory approximation as indicated in (9). In practice, p^n_{PGD} is computed assuming that p^{n-1}_{PGD} is available, taking as unknown only the last term, namely F^n_x, $F^n_{\theta_j}$ and $F^n_{\sigma_s}$, for $j = 1,\ldots,n_\theta$ and $s = 1,\ldots,n_\sigma$. In the following, the dependence on n of the unknown functions is omitted in the notation, that is the unknown modes are denoted by F_x, F_{θ_j} and F_{σ_s}.

Thus, for each PGD term, the unknown is a single separated function $\hat{p}$, the product of the unknown modes:

$$\hat{p} = p^n_{\text{PGD}} - p^{n-1}_{\text{PGD}} = F_x\prod_{j=1}^{n_\theta}F_{\theta_j}\prod_{s=1}^{n_\sigma}F_{\sigma_s}. \tag{11}$$

Consequently, the test function q in (6) is selected as a variation of $\hat{p}$, formally

$$\begin{aligned} q = \delta\hat{p} =& \delta F_x\prod_{j=1}^{n_\theta}F_{\theta_j}\prod_{s=1}^{n_\sigma}F_{\sigma_s} + F_x\sum_{\tilde{\jmath}=1}^{n_\theta}\delta F_{\theta_{\tilde{\jmath}}}\prod_{j\neq\tilde{\jmath}}F_{\theta_j}\prod_{s=1}^{n_\sigma}F_{\sigma_s} \\ &+ F_x\prod_{j=1}^{n_\theta}F_{\theta_j}\sum_{\tilde{s}=1}^{n_\sigma}\delta F_{\sigma_{\tilde{s}}}\prod_{s\neq\tilde{s}}F_{\sigma_s}\ , \end{aligned} \tag{12}$$

where δF_x, $\delta F_{\theta_{\tilde{\jmath}}}$ and $\delta F_{\sigma_{\tilde{s}}}$ are the test functions corresponding to each parametric dimension.

The alternated directions strategy consists in taking as unknown only one of the modes (searching direction) and considering all the other modes known. Let us denote by $F_\star$ the unknown mode, where now $\star$ stands for the searching direction and takes any of the values x, θ_j and σ_s, for $j = 1,\ldots,n_\theta$ and $s = 1,\ldots,n_\sigma$. Note that, with this notation, (12) is readily rewritten as

$$q = \sum_{\star}\delta F_\star\prod_{\tilde{\star}\neq\star}F_{\tilde{\star}}. \tag{13}$$

Let $V_\star$ denote the natural functional space for the variable $\star$, that is $H^1(\Omega)$ for $\star = x$ and $L^2(I_\star)$ for other values.

Thus, one iteration requires solving for some value of $\star$ the following problem: given all the modes but $F_\star$, that is given $F_{\tilde{\star}}$ for all $\tilde{\star} \neq \star$, find $F_\star \in V_\star$ such that

$$A(F_\star \prod_{\tilde{\star}\neq\star} F_{\tilde{\star}}, q_\star) = L(q_\star) - A(p_{\text{PGD}}^{n-1}, q_\star) \text{ for all } \delta F_\star \in V_\star, \text{ being } q_\star = \delta F_\star \prod_{\tilde{\star}\neq\star} F_{\tilde{\star}}. \quad (14)$$

Note that solving (14) for some direction $\star$ is a linear problem in a 1D setup for $\star = \theta_j$ and for $\star = \sigma_s$ (the geometric and material parameters). For $\star = x$, it has the dimension of the original problem.

Remark 1
The option of selecting $\delta F_\star \in V_\star$ is natural for $\star = x$ (Galerkin formulation). In the case of the parametric directions, for $\star = \theta_j$ and for $\star = \sigma_s$, taking $\delta F_\star \in V_\star$ results in a least-squares criterion to approximate $F_\star$. This is the option adopted here. However, other authors implement the PGD iteration for the parametric dimensions using other variational setups. For instance, in [19], a point collocation method is employed such that the pointwise values of $F_\star$ are obtained in a 1D grid of points and the function is interpolated. In this case, the test functions $\delta F_\star$ are taken as Dirac delta functions centered in the grid nodes.

3. COMPUTATIONAL ASPECTS

3.1. Separation of input data

The determinant of the Jacobians $|\mathbf{J}_e|$ and the matrices $\mathbf{D}_e$ appearing in the bilinear form $A(\cdot,\cdot)$ (see eq. (10)) depend on $\theta_1, \cdots \theta_{n_\theta}$ in an a priori non-separable way. Nevertheless, in order to obtain the separability of the bilinear form with respect to all the parametric dimensions, these functions are required to be separable. The idea, therefore, is to build a separable approximation of $|\mathbf{J}_e|$ and $\mathbf{D}_e$.

Let us focus on a generic non-separable continuous funcion

$$f : I_{\boldsymbol{\theta}} \to \mathbb{R}\,, \quad f = f(\theta_1, \cdots \theta_{n_\theta}).$$

The goal is to find a separable function $f^{sep} \simeq f$.

Let us introduce the following discretization of the parametric space

$$\mathcal{T}_{\boldsymbol{\theta}}^h = \mathcal{T}_{\theta_1}^h \times \cdots \times \mathcal{T}_{\theta_{n_\theta}}^h \,, \text{ with } \mathcal{T}_{\theta_j}^h = [\theta_j^1, \cdots \theta_j^{n_j}]\,,$$

where n_j is the number of sampling points in the j^{th} direction. $\mathcal{T}_{\boldsymbol{\theta}}^h$ is a tensor of rank n_θ, with $\prod_{j=1}^{n_\theta} n_j$ components.

In order to obtain f^{sep}, first of all f is evaluated in the multidimensional grid $\mathcal{T}_{\boldsymbol{\theta}}^h$, obtaining the n_θ rank tensor with $\prod_{j=1}^{n_\theta} n_j$ components

$$f_h = f(\mathcal{T}_{\boldsymbol{\theta}}^h)\,.$$

Then f_h is separated in a compact representation by means of an algebraic method, such as the High Order Singular Value Decomposition (HOSVD) [20], the Candecomp/Parafac (CP)[21, 22] or a PGD-projection [23] to obtain,

$$f_h^{sep} = \sum_{m=1}^{n_f} \mathbf{a}_1^m \otimes \cdots \otimes \mathbf{a}_{n_\theta}^m \,, \text{ with } \mathbf{a}_j^m \in \mathbb{R}^{n_j}.$$

This approximation is a discrete separable function and it is described by a lower number of coefficients $n_f \sum_{j=1}^{n_\theta} n_j \ll \prod_{j=1}^{n_\theta} n_j$.

The continuous separable approximation of f, f^{sep}, is defined in each point of the continuous parametric space as the interpolation of f_h^{sep}. Next section presents some details on f^{sep} and the error introduced by this separation. A detailed study of how separation of given data affects PGD can be found in [24].

3.2. Selection of sampling points

The choice of the sampling points $\mathcal{T}_{\boldsymbol{\theta}}^h$ has an impact on the convergence of the PGD method as shown in [24]. We compare two different choices. The first one considers as sampling points $\mathcal{T}_{\boldsymbol{\theta}}^h$ the grid nodes defining the finite element space $V_{\theta_j}^h$, while the second choice consists in sampling at the Gauss points of each parametric dimension.

We recall that we perform the numerical integration with respect to the parameters at the Gauss points. Therefore the separable approximations of $|\mathbf{J}_e|$ and $\mathbf{D}_e$ have to be evaluated at these points. This means that in the case of the first choice (grid nodes), an additional error is committed, due to the need of transferring the information between the grid nodes and the Gauss points, through the interpolation.

We underline that the choice of the sampling points is necessary only when a Galerkin approximation is adopted for the parametric dimensions. In the case in which a collocation method is used (see Remark 1), this problem is easily solved by taking the collocation points.

3.3. Separable approximation and parallel implementation

The separation of $|\mathbf{J}_e|$ and $\mathbf{D}_e$ is required for each element of the of the coarse mesh $e = 1, \cdots, n_T$. The functions to be separated, for each e, are 4: one scalar function $|\mathbf{J}_e|$ and one 2×2 symmetric tensor $\mathbf{D}_e$ (equivalent to 3 scalar functions). Each of these functions have to be evaluated in the grid $\mathcal{T}_{\boldsymbol{\theta}}^h$ and to be separated by means of HOSVD.

This means that the total of the scalar functions to be separated are $4n_T$. This could require a lot of computational time if done sequentially, especially for an increasing number of coarse elements. Actually, the computational effort increases linearly with n_T. Nevertheless, all the functions to be sepatated are independent to each other and therefore the separable approximations can be obtained in parallel, before starting the PGD computation.

3.4. Compressed PGD and PGD+compression

The online phase (evaluation of the PGD solution) is faster if the number of modes of the PGD solution is lower. This is the reason why we try to reduce the terms of the solution, by applying the method called PGD compression [10, 23] in two different ways.

The PGD compression aims at reducing the number of terms in the PGD expansion while keeping the accuracy of the representation. In fact, this is possible because the standard PGD algorithm is not enforcing any orthogonality between successive terms and often the modes are strongly correlated.

Let us consider a general function computed with PGD, say

$$f_{\text{PGD}}(z_1, \cdots, z_{n_p}) = \sum_{m=1}^{n_{\text{PGD}}} \prod_{i=1}^{n_p} F_i^m(z_i),$$

depending on n_p parameters $z_1, \cdots, z_{n_p}$. We aim at approximating f_{PGD} with its compressed counterpart

$$f_{comp}(z_1, \cdots, z_{n_p}) = \sum_{m=1}^{n_{comp}} \prod_{i=1}^{n_p} \tilde{F}_i^m(z_i),$$

with $n_{comp} < n_{\text{PGD}}$.

This approximation is obtained via the least-squares criterion, namely by minimizing

$$\|f_{comp} - f_{\text{PGD}}\|_{L^2(I_{\mathbf{z}})} = \int_{I_{z_1}} \cdots \int_{I_{z_{n_p}}} (f_{comp} - f_{\text{PGD}})^2 \, dz_{n_p} \cdots dz_1.$$

This is equivalent to solve the following variational problem: find $f_{comp} \in V$ (V proper variational space) s.t.

$$\underbrace{(f_{comp}, \delta f)_{L^2(I_{\mathbf{z}})}}_{A(f_{comp}, \delta f)} = \underbrace{(f_{\text{PGD}}, \delta f)_{L^2(I_{\mathbf{z}})}}_{L(\delta f)} \quad \forall \delta f \in V.$$

The compressed function f_{comp} is built by using the PGD greedy algorithm: given f_{comp}^{n-1}, compute f_{comp}^{n} by projecting on each parametric space the difference between f_{PGD} and f_{comp}^{n-1} (see [23] for further details.

We adopt two different strategies in order to obtain a final compressed PGD solution. In the first case we compute the whole PGD solution and then we compress it, obtaining a solution with a lower number of modes. With the second method, the compression is alternated with the PGD computation. This means that it is performed each N PGD modes computed. Then the PGD computation continues, using the new compressed intermediate solution. This second technique allows us to obtain the same accuracy (in terms of error w.r.t. the classical finite element solution), but saving in the computational effort during the offline phase. We will refer to the first method with "compressed PGD" and to the second one with "PGD+compression". Similar techniques has been successfully applied in [10] and [23] to reduce the number of modes of the PGD solution. The order of convergence of the PGD solution with the number of modes, although, does not seem to be affected by adding the projection to the standard PGD algorithm. A systematic study is required to understand the influence of the projection on PGD.

3.5. Solver for the inverse problem

Consider the reference finite element solution of the problem, e.g. the one obtained with the reference velocity profile, and the PGD solution of the parameterized problem (6). In order to identify the parameters that minimize the functional J defined in (8), we use the Matlab ® function `fmincon`, which performs the minimization with constraints. In particular, the constraints in our case are given by the extreme values of the intervals in which the parameters are defined. The `fmincon` function uses the sequential quadratic programming algorithm [15].

4. NUMERICAL RESULTS

In this section a parameterized Helmholtz problem is solved using the proposed PGD methodology. Two test cases are presented: first, an academic example is used to evaluate the correctness of the methodology by comparing PGD results with standard FE solutions. The example is also used to test the different samplings proposed in Section 3.2 to obtain the separable versions of $|\mathbf{J}_e|$ and $\mathbf{D}_e$. Moreover, the behaviour of the Compressed PGD and PGD+compression options described in Section 3.4 are compared.

A second test case is based on a simplified version of a real geological cross section. This example shows the behaviour of the method in a more realistic scenario. The geological cross section is located in NW Germany in a basin where salt-tectonics is dominant [16]. Due the presence of salt, seismic studies have difficulties to resolve the underground structure and it is usual to obtain areas with high uncertainty (see for example Figures 4, 6 and 8 in [16]).

4.1. First test case

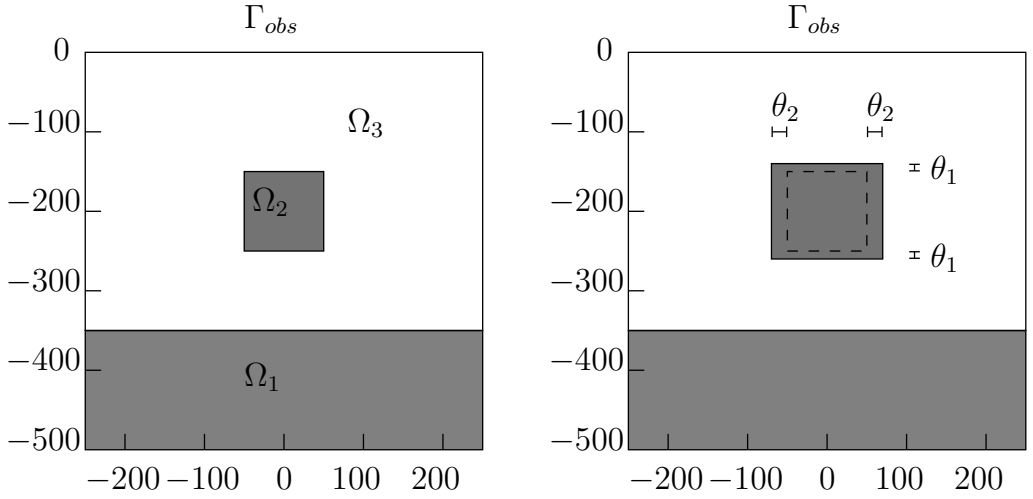
The spatial domain Ω of the first test case is composed be three different materials (as shown in Figure 2), such that $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2 \cup \bar{\Omega}_3$. The material property present in the Helmholtz problem (1) is the squared slowness $\sigma = 1/c^2$, where c is the wave propagation velocity. When defining the parameter, one has to decide if it represents the squares slowness σ, or if it represent the propagation velocity c. The two options are equivalent, but, due to numerical reasons here σ is defined as parameter. The range of values taken by σ is smaller than that of c.

Each subdomain Ω_i has associated a (constant within the domain) squared slowness σ_i, for $i = 1, 2, 3$. The values for σ_1 and σ_2 are considered parameters of the problem and span as follows,

$$\sigma_1 \text{ s.t. } v_1 = \sigma_1^{-1/2} \in (3500, 5500)$$

and

$$\sigma_2 \text{ s.t. } v_2 = \sigma_2^{-1/2} \in (2000, 4000)$$



(a) reference domain : $\theta_1 = \theta_2 = 0$ m (b) $\theta_1 = 10$ m , $\theta_2 = 20$ m

Figure 2. First test case: spatial domain (a) and geometric parameterization of the inner interface (b).

where the velocities v_i are expressed in m/s. The squared slowness of σ_3 is constant and is such that $v_3 = \sigma_3^{-1/2} = 2000$ m/s.

Moreover, two geometric parameters, θ_1 and θ_2, determine the height and the width of Ω_2 as follows (Figure 2),

$$\Omega_2 = (-50 - \theta_2, 50 + \theta_2) \times (-250 - \theta_1, -150 + \theta_1) \ ,$$

where distances are in meters.

A homogeneous Robin condition is applied on the bottom and on the two laterals of the domain, while a Neumann condition with value g is applied on the top boundary Γ_{top} with

$$g = \begin{cases} 1 & \text{for } x \in [100, 150] \\ 0 & \text{elsewhere} \end{cases} \ .$$

The choice is made with no loss of generality, due to the linear character of the problem.

A fixed frequency of 15 Hz (corresponding to a minimum wavelength $\lambda_{min} \simeq 133$ m) is considered. Due to the simple geometry of the spatial domain, the coarse mesh used to introduce the geometry parameters into the equation is composed by only $n_T = 18$ macro-elements. The FE mesh, on the other hand, is much finer and it is obtained by subdividing each coarse element into 1024 elements.

Example 1: influence of the sampling points

This first example shows the influence of the different sampling points used to separate $|\mathbf{J}_e|$ and $\mathbf{D}_e$ by the CP algorithm. The two options, as described in Section 3.2, are: i) sampling at FE grid nodes of the parametric meshes, and ii) sampling at the integration points of all FE element of the parametric meshes. Note that ii) prevents one interpolation (from nodes to integration points) when solving the space subproblem of PGD scheme.

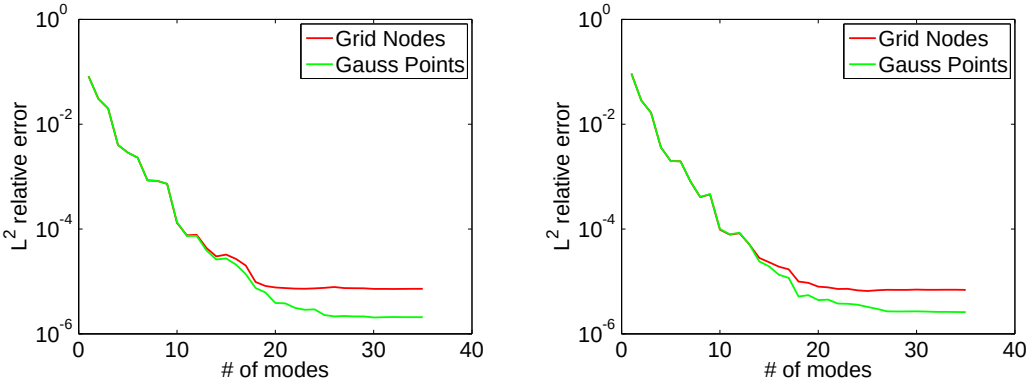
In order to test this effect in a simpler setup, only one geometric parameter, θ_1, is considered and the other three parameters are kept fixed with values $\theta_2 = 0$, σ_1 s.t. $v_1 = 4000$ m/s and σ_2 s.t. $v_2 = 2500$ m/s.

Convergence curves of the PGD solution with the number of terms are shown in Figure 3. The relative error plotted are computed as the $L^2(\Omega)$ norm of the difference between the PGD solution and the corresponding FE solution for some particular value of the parameter θ_1.

DOI: 10.1002/nme

Curves of Figure 3 show that, as expected, the lower bound of the errors for solutions computed with sampling at integration points is approximately one order of magnitude smaller than those computed with nodal sampling.

The final flattening of the convergence curves (meaning that the convergence towards the FE solution has stopped even when new terms are added) is due to a relatively coarse discretization of the parametric dimension. If parameter mesh is refined, the curves continue their convergence towards the FE solution.



(a) $\theta_1 = 0$ m (b) $\theta_1 = 10$ m

Figure 3. Example 1: Convergence of the PGD solution depending on θ_1, using different sampling nodes, for two values of the parameter.

Example 2: PGD+compression vs. compressed PGD

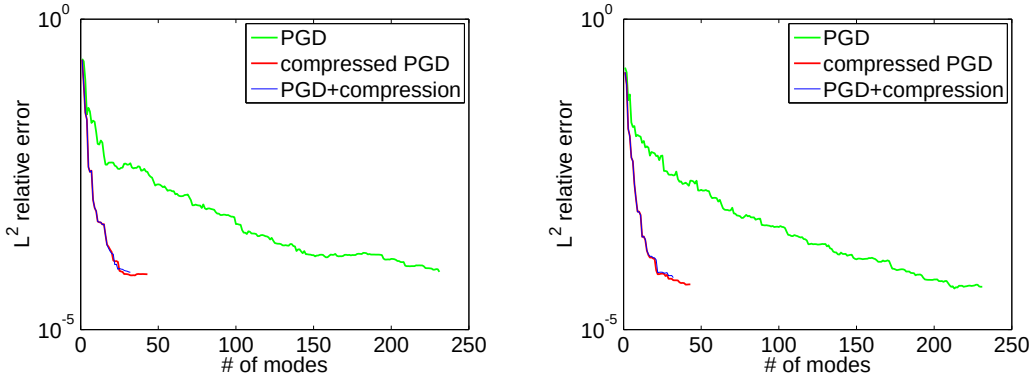
The behaviour of the PGD+compression and the compressed PGD algorithms described in Section 3.4 is studied next. Two geometric parameters, θ_1 and θ_2, are considered while keeping the values of the two material parameters σ_1 and σ_2 fixed with the same values as in the previous example.

Figure 4 shows the different convergence curves of the relative error with increasing number of terms for two different values of the parameters. The PGD solution can be largely compressed, reducing the number of terms from ~ 240 to ~ 45, by doing a projection after the complete offline phase (compressed PGD). On the other hand, the projection can be done many times along the greedy offline phase. Every a fixed number of computed terms (10 in this example), one projection is performed. The computer time in this option is reduced to approximately one third (compared with the PGD process with no compression) because the second term of the right hand side of (14) includes a sum on all already computed terms. This sum is present on all the sub problems of the greedy algorithm (space and parametric dimensions). Moreover, the projection process is extremely fast and therefore the benefits of the reduced number of terms exceeds its overhead. As shown in Figure 4, the PGD+compression algorithm produces almost the same convergence curve as the compressed PGD.

Example 3: solution of the 4D inverse problem

In the next example a PGD solution depending on the four parameters (two geometric and two material properties) is used to solve an synthetic inverse problem. The goal is to find the value of the parameters that produces the solution that reproduce best some observations. The inverse problem is stated as a constrained minimization of the functional defined in (8). The observed data is assumed to be in Γ_{obs}, that corresponds to the top side of the computational domain (Figure 2). This situation is similar to that of a seismic survey, were the measurements are done with surface seismometers and the complete wave trajectories in the whole domain are inferred.

The synthetic version of this problem is build as follows: i) a reference solution p_{FE} is created by solving a FE problem with some reference values of the parameters $(\boldsymbol{\theta}_{ref}, \boldsymbol{\sigma}_{ref})$, ii) the solution



(a) $\theta_1 = 0$ m , $\theta_2 = 0$ m (b) $\theta_1 = 30$ m , $\theta_2 = 10$ m

Figure 4. Example 2: Convergence curves of the error of PGD solution with different algorithms. Errors are computed by comparing the PGD solution with a FE solution for a given value of the parameters. The PGD, compressed PGD and PGD+compression algorithms are used until getting a relative error of $\approx 10^{-4}$.

p_{obs} on Γ_{obs} is taken as observations, that is

$$p_{obs} = p_{FE}(\boldsymbol{\theta}_{ref}, \boldsymbol{\sigma}_{ref})|_{\Gamma_{obs}} \ ,$$

and finally, iii) the constrained minimization problem is solved using the parametric PGD solution to try to recover the reference parameters from the observed data. The minimization method used here is the Sequential Quadratic Programming (SQP) [15], and its implementation is the `fmincon` function of Matlab®.

Results of the parameter recovery are shown in Table I for six different sets of reference values. The maximum error in the recovered parameters is below 6% and the mean error in the tested cases is $\sim 1\%$. The number of iterations used by the SQP method, which is the same that evaluation of the functional J (Eq. (8)) or the number of "forward" problems solved to recover the parameters, is indicated in the last column of Table I. In all cases more than 100 evaluations where required. The fast evaluation of the PGD solution allows a very efficient solve of the inverse problem. This is particularly important if a global minimum is being seek, because on that case many inverse problems need to be solved starting from different initial guesses (due to the presence of local minima).

The dimensionality of the problem considered in this test case is four (two space dimensions and two parametric dimensions). The PGD technique, although, is able to deal with a larger number of parameters (e.g. in [12] 11 parametric dimensions are successfully handled). Nevertheless, the geometrical parameterization used here requires the separation of the Jacobian matrices involved in the geometrical mapping. Because this separation is done with an *a posteriori* method, the number of required evaluations of these functions grows exponentially with the number of geometrical dimensions (the so-called "curse of dimensionality", here restricted to the geometrical dimensions). Therefore, it is recommended to keep to number of geometrical dimensions restricted to two or three. Note that the addition of other kinds of dimensions to the parametric problem, for example material parameters, does not produce an exponential grow and it can be done without a large penalization in the performance.

4.2. Second test case

A more realistic setup, based on a geological cross section from [16], is considered as a second test case. The spatial domain is 10km $\times$ 5km and the boundary conditions are the same as in previous example. Neumann boundary conditions are now applied to $x \in (6500, 6550)$ and frequency considered is fixed to 5 Hz.

The spatial domain is divided into six material subdomains each one with a different propagation velocity. Same as in the previous test case, the squared slowness σ_i are expressed in term of the

Reference values				Recovered values				Relative errors [%]				# it
[m]		[m/s]		[m]		[m/s]						
L_1	L_2	v_1	v_2	L_1	L_2	v_1	v_2	L_1	L_2	v_1	v_2	
50	50	2500	4000	51.07	50.84	2492	4002	1.7	2.1	0.30	0.057	161
50	50	3000	4000	50.07	49.91	3020	4009	0.18	0.14	0.68	0.23	155
50	50	3500	4000	50.34	49.39	3564	3999	1.2	0.68	1.8	0.013	189
50	50	2500	5000	47.08	50.18	2513	5029	0.36	5.8	0.51	0.57	163
50	80	3000	4000	50.19	80	3024	4020	0.38	0	0.82	0.50	127
60	80	3000	4000	60.15	80	2988	4044	0.25	0	0.41	1.1	101

Table I. Example 3: Results of the inverse problem for six different reference values. L_1 and L_2 are the lengths of the x and y sides of subdomain Ω_2 (controlled by parameters θ_1 and θ_2). v_1 and v_2 are the propagation velocities on subdomains Ω_1 and Ω_2, respectively (controlled by parameters σ_1 and σ_2). The last column shows the number of evaluations of the objective function required by the SQP algorithm.

corresponding wave propagation velocities. The first subdomain, corresponding to the salt body, has a parameterized σ_1 taking values in the range

$$\sigma_1 \text{ s.t. } v_1 = \sigma_1^{-1/2} \in (4450, 4700),$$

corresponding to standard velocities for salt [25]. The other five subdomains representing other rock types have constant velocities with values (in m/s) as follows:

v_2	v_3	v_4	v_5	v_6
2000	4000	3000	3500	2500

Moreover, two geometric parameters control the location of internal interfaces related to the salt subdomain. Their effect on the geometry is shown at Figure 5. The first parameter controls the smoothness of the upper part of the subdomain (called the *diapir*) and the second parameter controls the thickness of the salt layer crossing the domain. The coarse geometry mesh is composed by $n_T = 463$ elements (shown At Figure 5), each one subdivided into 256 triangular FE elements.

Solutions presented next are obtained by using the PGD+compression method, with a compression step done every 20 terms. This choice is not mandatory. The number of modes between two successive compressions has been fixed to 20 after a trial and error procedure, in order to guarantee that an effective compression is performed each time.

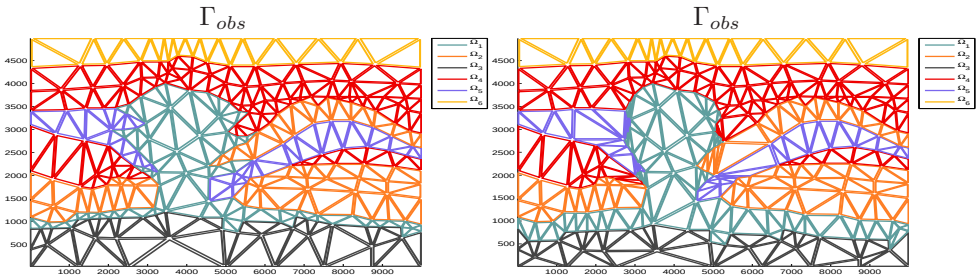


Figure 5. Second test case: Coarse geometry mesh for two set of values of the geometric parameters. Parameter 1 controls the smoothness of the upper part of the diapir and parameter 2 controls the thickness of the salt layer crossing the domain. Element colours indicate the different subdomains.

Example 4: Parameter sensitivity

This example shows how the sensitivity of the parameters can be easily identified from the parametric PGD solution. For this example the squared slowness of the fist subdomain (salt) is considered fixed s.t. $v_1 = \sigma_1^{-1/2} = 4500$ m/s and, therefore, the PGD solution depends on two

Reference values[-]		Computed values[-]		Relative errors [%]		# it
θ_1	θ_2	θ_1	θ_2	θ_1	θ_2	
0.5	0.5	0.50149	0.48318	0.2983	3.3646	52
0.8	0.3	0.79893	0.29862	0.13381	0.45842	56

Table II. Example 4. Results of the parameter identification problem. See caption of Table I for references.

geometric parameters only. Figure 6 show the real part of the solution obtained via PGD and FE for two given sets of parameters values.

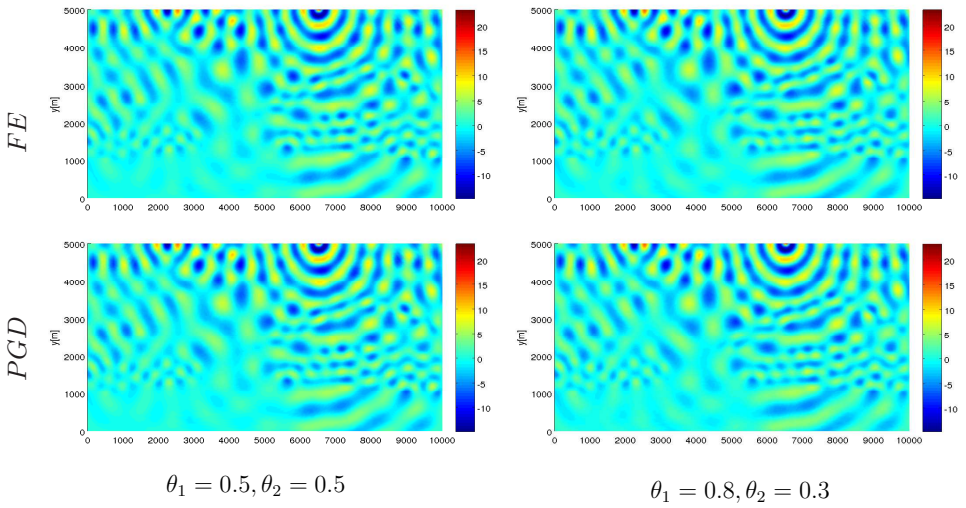


Figure 6. Example 4: Real part of the solution for two sets of geometric parameters. FE solutions (top) and PGD solutions (bottom).

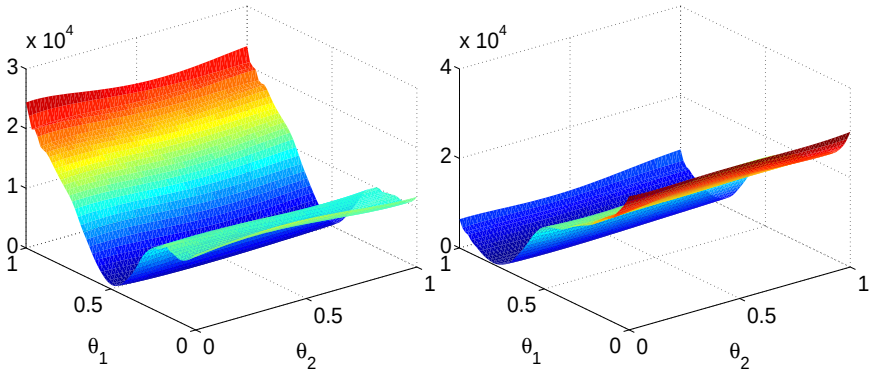
An inverse problem similar to that of the previous section is solved to recover the two parameters. The observed data is again restricted to the surface. Results of recovered parameters are shown in Table II. The relative error of the recovered parameter θ_1 are smaller than those of θ_2. Due to the extremely fast evaluation of the PGD solution, it is possible to plot the functional J in the parameter space, as shown in Figure 7. In that Figure, it can be seen that the functional J is much more sensitive to the first parameter. That explains why the SQP technique can recover θ_1 with smaller errors than θ_2. This ability of the PGD solution to easily estimate the sensitivity on the parameters is extremely helpful to setup efficiently the solver for the inverse problem.

Example 5: inverse problem with geophysical application
The last example consists in the recovery of the geometric and material parameters for the geologic cross section. In addition to the geometric parameters of the previous example, here the squared slowness of the salt (subdomain 1), σ_1, is taken as parameter with values such that the velocity range is $v_1 = \sigma_1^{-1/2} \in (4450, 4700)$.

Results of the inverse problem are shown in Table III. Same as in last example, the sensitivity of θ_2 is lower that that of θ_1. Errors in the recovered parameters are $< 2\%$ for θ_1 and σ_1, while up to 35% for θ_2.

CONCLUSIONS

In this work we apply the PGD technique to a parametric formulation of the Helmholtz problem that is relevant in the context of geological seismic studies. Parameters are of two kinds:



$\theta_1 = 0.5, \theta_2 = 0.5$ $\theta_1 = 0.8, \theta_2 = 0.3$

Figure 7. Example 4: Plot of the functional J in two cases

Reference values			Computed values			Relative errors [%]			# it
[-]		[m/s]	[-]		[m/s]				
θ_1	θ_2	v_1	θ_1	θ_2	v_1	θ_1	θ_2	v_1	
0.8	0.3	4650	0.7967	0.1958	4652.3	0.41	34.7	0.050	114
0.5	0.5	4500	0.5029	0.5637	4509.3	0.59	12.7	0.207	88
0.2	0.6	4550	0.1968	0.5949	4541.9	1.61	0.84	0.179	81

Table III. Example 5. Results of the parameter identification problem. See caption of Table I for references.

material parameters, namely squared slowness of the different materials and geometric parameters, determining the location of internal interfaces between materials.

After the offline phase, PGD allows to obtain in real time the spatial solution for any given set of parameters. This extremely fast response is ideal for the solution of inverse problems, in which the values of the parameters need to be recovered to fit some observed data. The multi-query character (many evaluations of the objective function) of the inverse techniques, makes PGD perfectly suited for that. Moreover, when a global minimum is required, the importance of having a fast forward solver is even bigger, as many inverse problems starting from different initial guesses will be needed, due to the presence of local minima.

Several synthetic examples of the inverse problem presented here show that both kind of parameters can be recovered in most cases with accuracy smaller than $\sim 5\%$. Importantly, the PGD solution allows to easily study the sensitivity of the parameters on the objective function. In this way, it becomes clear when a parameter can or cannot be identified with a given set of observations.

The offline phase of the PDG method requires some algorithmic decisions. Some of them were studied in this work: first, the sampling of the parameter dimensions used to compute the separated Jacobians (required by the geometric parameterization) needs to be done at the integration points in order to avoid interpolation errors that affect the final convergence of the PGD solution. Second, "compressing" PGD solution to reduce its number of terms via a L^2 projection allows to reduce the time of both the online and offline phases. The added computational overhead of the compression step is much smaller than the benefits of keeping the number of terms low.

REFERENCES

1. Symes WW. The seismic reflection inverse problem. *IOPscience* 2009; **25**.
2. Tarantola A. Inversion of seismic reflection data in the acoustic approximation. *Geophysics* 1984; **49**(8):1259–1266.
3. Virieux J, Operto S. An overview of full-waveform inversion in exploration geophysics. *Geophysics* 2009; **74**(6):WCC1–WCC26.
4. Alford RM, Kelly KR, Boore DM. Accuracy of finite-differences modeling of the acoustic wave equation. *Geophysics* 1974; **39**(6):834–842.
5. Plessix RE. A helmholtz iterative solver for 3d seismic-imaging problems. *Geophysics* 2007; **72**(5):SM185–SM194.
6. Song ZM, Williamson PR. Frequency-domain acoustic-wave modeling and inversion of crosshole data: Part 1-2.5-0 modeling method. *Geophysics* 1995; **60**(3):784–795.
7. Kerschen G, Golinval JC, Vakasis AF, Bergman LA. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: An overview. *Nonlinear Dynamics , Springer* 2005; **41**:147–169.
8. Haasdok B, Ohlberger M, Rozza G. A reduced basis method for evolution schemes with parameter-dependent explicit operators. *Electronic Transactions on Numerical Analysis* 2008; **32**:145–161.
9. Chinesta F, Leygue A, Bordeu F, Aguado JV, Cueto E, Gonzalez D, Alfaro I, Ammar A, Huerta A. PGD-Based Computational Vademecum for Efficient Design, Optimization and Control. *Archives of Computational Methods in Engineering* Jan 2013; **20**(1):31–59.
10. Chinesta F, Keunings R, Leygue A. *The Proper Generalized Decomposition for Advanced Numerical Simulations. A Primer*. SpringerBriefs in Applied Sciences and Technology, Springer, 2014.
11. Ammar A, Huerta A, Chinesta F, Cueto E, Leygue A. Parametric solutions involving geometry: a step towards efficient shape optimization. *Computer Methods in Applied Mechanics and Engineering* 2014; **268**:178–193.
12. Zlotnik S, Díez P, Modesto D, Huerta A. Proper generalized decomposition of a geometrically parametrized heat problem with geophysical applications. *Int. J. Numer. Meth. Engng* 2015; **103**(10):737–758.
13. Aguado JV, Huerta A, Chinesta F. R. *International Journal for Numerical Methods in Engineering* 2015; **102**:991–1017, doi:10.1002/nme.4784.
14. Germoso C, Aguado JV, Fraile A, Alarcon E. Efficient PGD-based dynamic calculation of non-linear soil behavior. *Comptes Rendus Mecanique* 2016; **344**:24–41.
15. Nocedal J, Wright S. *Numerical Optimization*. Springer Series, 2006.
16. Mohr M, Kukla PA, Urai JL, Bresser G. Multiphase salt tectonic evolution in nw germany: seismic interpretation and retro-deformation. *Int. J. Earth Sci. (Geol Rundsch)* 2005; (94):917–940.
17. Attanayake J. Seismic migration (sm) 13 November 2006. Presentation.
18. Albertin U, Kapoor J, Randall R, Smith M, Brown G, Soufleris C, Whitfield P, Dewey F, Farnsworth J, Grubitz G, *et al.*. The time for depth imaging. *Oilfield Review* 2002; .
19. Neron D, Ladevèze P. Proper generalized decomposition for multiscale and multiphysics problems. *Archives of Computational Methods in Engineering* 2010; **17**(4):351–372.
20. De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* 2000; **21**(4):1253–1278.
21. Carroll JD, Chang JJ. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika* 1970; **35**(3):283–319.
22. Harshman RA. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA working papers in phonetics* 1970; **16**:1–84.
23. Modesto D, Zlotnik S, Huerta A. Proper Generalized Decomposition for parameterized helmholtz problems in heterogeneous and unbounded domains: application to harbor agitation. *Comput. Methods Appl. Mech. Eng.* 2015; **295**:127–149, doi:10.1016/j.cma.2015.03.026.
24. Zlotnik S, Díez P, González D, Cueto E, Huerta A. Effect of the separated approximation of input data in the accuracy of the resulting pgd solution. *Advanced Modeling and Simulation in Engineering Sciences* 2015; **2**(28), doi:10.1186/s40323-015-0052-6.
25. Yan F, Han DH, Yao Q, Li H. Seismic velocities of halite salt: anisotropy, dispersion, temperature and stress effects. *SEG* 2014; .

A. SEPARABLE APPROXIMATIONS AND MATRICES

The following separable approximations are introduced for each element $e = 1, \cdots, n_T$ belonging to the coarse mesh :

$$|\boldsymbol{J}_e| \simeq \sum_{l=1}^{n_{Je}} \prod_{j=1}^{n_\theta} T_{\theta_j}^{e,l} \text{ and } \boldsymbol{D}_e \simeq \sum_{a=1}^{3} \sum_{m=1}^{n_{De}^a} \prod_{j=1}^{n_\theta} G_{\theta_j}^{e,a,m} I_a \tag{15}$$

where

$$I_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad I_2 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad I_3 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

The scalar $\tilde{\sigma}_e$, associated with the macro-element T_e, is defined to simplify notation in the matrix formulation (Appendix B),

$$\tilde{\sigma}_e = \begin{cases} 1, & \text{if } \sigma \text{ in } T_e \text{ is described by one of the parameters} \\ \sigma|_{T_e}, & \text{else (if the value of} \sigma \text{in } T_e \text{ is a fixed value)} \end{cases}$$

The following definitions are used in the matrix formulation of the PGD algorithm (Appendix B),

$$M_{\hat{x}} = \int_{\hat{T}} N_{\hat{x}} N_{\hat{x}}^T d\hat{x} \qquad K_{\hat{x}}^a = \int_{\hat{T}} \nabla N_{\hat{x}}^T \cdot (I_a \nabla N_{\hat{x}})\, d\hat{x} \qquad M_{x,e}^R = \mathbb{A}_{\partial T_e} \int_{\Gamma_R \cap \partial T_e} N_x^{1D} (N_x^{1D})^T dx$$

$$M_{\theta_j}^{e,l} = \int_{I_{\theta_j}} T_{\theta_j}^{e,l} N_{\theta_j} N_{\theta_j}^T d\theta_j \qquad M_{\theta_j}^{e,a,m} = \int_{I_{\theta_j}} G_{\theta_j}^{e,a,m} N_{\theta_j} N_{\theta_j}^T d\theta_j \qquad M_{\theta_j} = \int_{I_{\theta_j}} N_{\theta_j} N_{\theta_j}^T d\theta_j$$

$$M_{\sigma_s}^e = \int_{I_{\sigma_s}} \left[\sigma_s \tilde{\delta}_{se} + (1 - \tilde{\delta}_{se})\right] N_{\sigma_s} N_{\sigma_s}^T d\sigma_s \qquad M_{\sqrt{\sigma_s}}^e = \int_{I_{\sigma_s}} \left[\sqrt{\sigma_s} \tilde{\delta}_{se} + (1 - \tilde{\delta}_{se})\right] N_{\sigma_s} N_{\sigma_s}^T d\sigma_s$$

$$M_{\sigma_s} = \int_{I_{\sigma_s}} N_{\sigma_s} N_{\sigma_s}^T d\sigma_s$$

$$\mathbf{q}_{x,e} = \mathbb{A}_{\partial T_e} \int_{\Gamma_N \cap \partial T_e} g(x) N_x^{1D} dx \qquad \mathbf{q}_{\theta_j} = \int_{I_{\theta_j}} N_{\theta_j} d\theta_j \qquad \mathbf{q}_{\sigma_s} = \int_{I_{\sigma_s}} N_{\sigma_s} d\sigma_s$$

where $\tilde{\delta}_{se}$ is defined as

$$\tilde{\delta}_{se} = \begin{cases} 1 & \text{the parameter } \sigma_s \text{ describes the material properties of } T_e \\ 0 & \text{otherwise} \end{cases}.$$

B. MATRIX FORMULATION OF THE DISCRETE PROBLEM

The matrix formulation of the alternated directions scheme is described in the Algorithm 1. Unknowns are denoted by $\mathbf{f}_\star$ instead of by $\mathbf{f}_\star^n$. All the matrices and vectors used in the algorithm are defined next. Appendix A presents the definition of all the local matrices and vectors. Let us recall that the symbol $\mathbb{A}$ stands for the matrix assembling operator.

The following matrices and vectors appear in the discrete version of the parametric problem.

$$\begin{aligned} \mathbf{M}_x^{\tilde{m}} = \mathbb{A}_{e=1}^{n_T} \Bigg\{ & -\omega^2 \left[\tilde{\sigma}_e M_{\hat{x}} \left(\sum_{l=1}^{n_{Je}} \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,l} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right] \\ & + \sum_{a=1}^{3} \left(K_{\hat{x}}^a \sum_{m=1}^{n_{De}^a} \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,a,m} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s} \mathbf{f}_{\sigma_s}^{\tilde{m}} \\ & - i\omega \left[\sqrt{\tilde{\sigma}_e} M_{x,e}^R \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j} \mathbf{f}_{\theta_j}^{\tilde{m}} \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sqrt{\sigma_s}}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right] \Bigg\} \end{aligned} \tag{16a}$$

Algorithm 1: Matrix formulation of the alternated directions scheme

% Computation of p^{n}_{PGD} given p^{n-1}_{PGD}
Input: $\mathbf{f}_x^m$, $\mathbf{f}_{\theta_j}^m$ and $\mathbf{f}_{\sigma_s}^m$ for $j = 1, \cdots, n_\theta$, $s = 1, \cdots, n_\sigma$ and $m = 1, \cdots, n-1$
Initialize $\mathbf{f}_x$, $\mathbf{f}_{\theta_j}$ and $\mathbf{f}_{\sigma_s}$
while *the convergence is not reached, i.e. some* $\mathbf{f}_\star$ *is not yet stationary* **do**
  $[\star = x]$: update $\mathbf{f}_x$ such that

$$\mathbf{M}_x \mathbf{f}_x = \mathbf{r}_x \ ,$$

  where $\mathbf{M}_x = \mathbf{M}_x^n$, and $\mathbf{M}_x^{\tilde{m}}$, $\mathbf{r}_x$ are defined in (16a),(17a)
  for $\tilde{\jmath} = 1, \cdots n_\theta$ **do**
    $[\star = \theta_{\tilde{\jmath}}]$: update $\mathbf{f}_{\theta_{\tilde{\jmath}}}$ such that

$$\mathbf{M}_{\theta_{\tilde{\jmath}}} \mathbf{f}_{\theta_{\tilde{\jmath}}} = \mathbf{r}_{\theta_{\tilde{\jmath}}} \ ,$$

    where $\mathbf{M}_{\theta_{\tilde{\jmath}}} = \mathbf{M}_{\theta_{\tilde{\jmath}}}^n$, and $\mathbf{M}_{\theta_{\tilde{\jmath}}}^{\tilde{m}}$, $\mathbf{r}_{\theta_{\tilde{\jmath}}}$ are defined in (16b),(17b)
  for $\tilde{s} = 1, \cdots n_\sigma$ **do**
    $[\star = \theta_{\tilde{s}}]$: update $\mathbf{f}_{\sigma_{\tilde{s}}}$ such that

$$\mathbf{M}_{\sigma_{\tilde{s}}} \mathbf{f}_{\sigma_{\tilde{s}}} = \mathbf{r}_{\sigma_{\tilde{s}}} \ ,$$

    where $\mathbf{M}_{\sigma_{\tilde{s}}} = \mathbf{M}_{\sigma_{\tilde{s}}}^n$, and $\mathbf{M}_{\sigma_{\tilde{s}}}^{\tilde{m}}$, $\mathbf{r}_{\sigma_{\tilde{s}}}$ are defined in (16c),(17c)
  Check stationarity: |updated $\mathbf{f}_\star$ − previous $\mathbf{f}_\star^{old}$| < tolerance

$$\begin{aligned}
\mathbf{M}_{\theta_{\tilde{\jmath}}}^{\tilde{m}} = \sum_{e=1}^{n_T} \Bigg\{ & -\omega^2 \left[\tilde{\sigma}_e (\mathbf{f}_{x,e}^{\tilde{m}})^H M_{\hat{x}} \mathbf{f}_{x,e}^{\tilde{m}} \sum_{l=1}^{n_{J^e}} \left(\prod_{j \neq \tilde{\jmath}} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,l} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) M_{\theta_{\tilde{\jmath}}}^{e,l} \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right] \\
& + \sum_{a=1}^{3} \left((\mathbf{f}_{x,e}^{\tilde{m}})^H K_{\hat{x}}^a \mathbf{f}_{x,e}^{\tilde{m}} \sum_{m=1}^{n_{D^e}^a} \left(\prod_{j \neq \tilde{\jmath}} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,a,m} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) M_{\theta_{\tilde{\jmath}}}^{e,a,m} \right) \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s} \mathbf{f}_{\sigma_s}^{\tilde{m}} \\
& - i\omega \left[\sqrt{\tilde{\sigma}_e} (\mathbf{f}_{x,e}^{\tilde{m}})^H M_{x,e}^R \mathbf{f}_{x,e}^{\tilde{m}} \left(\prod_{j \neq \tilde{\jmath}} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) M_{\theta_{\tilde{\jmath}}} \prod_{s=1}^{n_\sigma} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sqrt{\sigma_s}}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right] \Bigg\}
\end{aligned} \tag{16b}$$

$$\begin{aligned}
\mathbf{M}_{\sigma_{\tilde{s}}}^{\tilde{m}} = \sum_{e=1}^{n_T} \Bigg\{ & -\omega^2 \left[\tilde{\sigma}_e (\mathbf{f}_{x,e}^{\tilde{m}})^H M_{\hat{x}} \mathbf{f}_{x,e}^{\tilde{m}} \sum_{l=1}^{n_{J^e}} \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,l} \mathbf{f}_{\theta_j}^{\tilde{m}} \left(\prod_{s \neq \tilde{s}} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right) M_{\sigma_{\tilde{s}}}^e \right] \\
& + \sum_{a=1}^{3} \left((\mathbf{f}_{x,e}^{\tilde{m}})^H K_{\hat{x}}^a \mathbf{f}_{x,e}^{\tilde{m}} \sum_{m=1}^{n_{D^e}^a} \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j}^{e,a,m} \mathbf{f}_{\theta_j}^{\tilde{m}} \right) \left(\prod_{s \neq \tilde{s}} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sigma_s} \mathbf{f}_{\sigma_s}^{\tilde{m}} \right) M_{\sigma_{\tilde{s}}} \\
& - i\omega \left[\sqrt{\tilde{\sigma}_e} (\mathbf{f}_{x,e}^{\tilde{m}})^H M_{x,e}^R \mathbf{f}_{x,e}^{\tilde{m}} \prod_{j=1}^{n_\theta} (\mathbf{f}_{\theta_j}^{\tilde{m}})^H M_{\theta_j} \mathbf{f}_{\theta_j}^{\tilde{m}} \left(\prod_{s \neq \tilde{s}} (\mathbf{f}_{\sigma_s}^{\tilde{m}})^H M_{\sqrt{\sigma_s}}^e \mathbf{f}_{\sigma_s}^{\tilde{m}} \right) M_{\sqrt{\sigma_{\tilde{s}}}}^e \right] \Bigg\}
\end{aligned} \tag{16c}$$

Vectors definition

$$\mathbf{r}_x = \mathbb{A}_{e=1}^{n_T} \mathbf{q}_{x,e} \prod_{j=1}^{n_\theta} \mathbf{f}_{\theta_j}^H \mathbf{q}_{\theta_j} \prod_{s=1}^{n_\sigma} \mathbf{f}_{\sigma_s}^H \mathbf{q}_{\sigma_s} - \sum_{m=1}^{n-1} \mathbf{M}_x^m \mathbf{f}_x^m \tag{17a}$$

$$\mathbf{r}_{\theta_{\tilde{\jmath}}} = \sum_{e=1}^{n_T} \mathbf{f}_{x,e}^H \mathbf{q}_{x,e} \left(\prod_{j \neq \tilde{\jmath}} \mathbf{f}_{\theta_j}^H \mathbf{q}_{\theta_j} \right) \mathbf{q}_{\theta_{\tilde{\jmath}}} \prod_{s=1}^{n_\sigma} \mathbf{f}_{\sigma_s}^H \mathbf{q}_{\sigma_s} - \sum_{m=1}^{n-1} \mathbf{M}_{\theta_{\tilde{\jmath}}}^m \mathbf{f}_{\theta_{\tilde{\jmath}}}^m \tag{17b}$$

$$\mathbf{r}_{\sigma_{\tilde{s}}} = \sum_{e=1}^{n_T} \mathbf{f}_{x,e}^H \mathbf{q}_{x,e} \prod_{j=1}^{n_\theta} \mathbf{f}_{\theta_j}^H \mathbf{q}_{\theta_j} \left(\prod_{s \neq \tilde{s}} \mathbf{f}_{\sigma_s}^H \mathbf{q}_{\sigma_s} \right) \mathbf{q}_{\sigma_{\tilde{s}}} - \sum_{m=1}^{n-1} \mathbf{M}_{\sigma_{\tilde{s}}}^m \mathbf{f}_{\sigma_{\tilde{s}}}^m \tag{17c}$$