SAM
Science Arts & Métiers
Archive ouverte Arts & Métiers ParisTech
ARTS
ET MÉTIERS
ParisTech

Space–time proper generalized decompositions for the resolution of transient elastodynamic models

L. Boucinha [a], A. Gravouil [a,c,*], A. Ammar [b]

[a] Université de Lyon, INSA-Lyon, LaMCoS, CNRS UMR5259, 18-20 rue des Sciences, F-69621 Villeurbanne, France
[b] Arts et Métiers ParisTech, ENSAM Angers, 2 boulevard du Ronceray, F-49035 Angers, France
[c] Institut Universitaire de France, France

ARTICLE INFO

Keywords:
Model reduction
Elastodynamic
Singular value decomposition (SVD)
Proper generalized decomposition (PGD)
Multi-field proper generalized decomposition (MF-PGD)
Tensorial formalism

ABSTRACT

In this paper, we investigate ability of proper generalized decomposition (PGD) to solve transient elastodynamic models in space–time domain. Classical methods use time integration schemes and an incremental resolution process. We propose here to use standard time integration methods in a non-incremental strategy. As a result, PGD gives a separated representation of the space–time solution as a sum of tensorial products of space and time vectors, that we interpret as space–time modes. Recent time integration schemes are based on multi-field formulations. In this case, separated representation can be constructed using state vectors in space and same vectors in time. However, we have experienced bad convergence order using this decomposition. Furthermore, temporal approximation must be the same for all fields. Thus, we propose an extension of classical separated representation for multi-field problems. This multi-field PGD (MF-PGD) uses space and time vectors that are different for each field. Calculation of decomposition is done using a monolithic approach in space and time, potentially allowing the use of different approximations in space and time. Finally, several simulations are performed with the transient elastodynamic problem with one dimension in space. Different approximations in time are investigated: Newmark scheme, single field time discontinuous Galerkin method and two fields time continuous and discontinuous Galerkin methods.

1. Introduction

Thanks to progresses in computer technologies, resolution of problems involving millions of unknowns has become ordinary in engineering applications [17]. And such simulations should be performed as many times as possible in optimization contexts. It seems now to be clear that traditional approaches (like finite element method) are no more compatible with phases of industrial design or times of scientific studies. Therefore innovative methodologies must be proposed in order to exploit the impressive amount of computational resources today available, in a more efficient way.

Reduced order modelling techniques appear to be good candidates to achieve this issue. In the context of evolution problems, models are traditionally projected on a reduced basis in space and resolved in time. Then, strategies differ from the definition of reduced spatial basis. In structural dynamics, the oldest strategies use structure eigen-modes [18]. More recently, methods based on Proper Orthogonal Decomposition (POD) were proposed (see [30,47,36,19] for examples in transient elastodynamics and [14,29] for physical interpretation of POD modes). These methods are well suited only when the displacement vector presents a similar behaviour over time [19]. Thus, for transient problem, enrichment strategies are needed in order to improved reduce spatial basis over simulation time [49].

In this paper, we use a different strategy: the full space–time solution is expressed as a linear combination of space–time modes, that are calculated *a priori* thanks to proper generalized decomposition (PGD). The PGD has been proven to be highly efficient in different applications [32,12,4,45]. For space–time decomposition of parabolic problems [3,46], computational requirements can be decreased of several orders of magnitude. However, to our knowledge, there is no significant work with application to second order hyperbolic equations. In this paper, we investigate PGD's ability to solve second order hyperbolic equations and concentrate on transient elastodynamic models.

Key point of PGD method is the separated representation [31,11]. Each space–time mode is decomposed as a tensorial product of one mode in space and an other in time. Separated representation drastically reduces computational storage if few modes are required to represent model solution with a good accuracy. As an example, suppose we use a space–time mesh with n_S points in

* Corresponding author at: Université de Lyon, INSA-Lyon, LaMCoS, CNRS UMR5259, 18-20 rue des Sciences, F-69621 Villeurbanne, France.
E-mail address: anthony.gravouil@insa-lyon.fr (A. Gravouil).

http://dx.doi.org/10.1016/j.cma.2012.11.003

space and n_T time instants. Also suppose that m space–time modes are necessary to accurately represent the model solution. Then, we need $n_S * n_T$ numerical values to store the full solution whereas only $m * (n_S + n_T)$ values are required for the separated representation. This drastically reduces memory requirement as long as $m \ll n_S, \; n_T$. Question is how many space–time modes do we need to accurately represent solution of transient elastodynamic problems in space–time domain?

Resolution of linear elastodynamic problems has been a challenging task for decades (and it still is), coming up with the creation of a huge amount of efficient time integration methods. We report the reader to [24,27] for overviews of existing time schemes and corresponding properties. Our purpose here is not to develop a new time integration algorithm. It is to propose a general strategy that can potentially decrease computational costs and/or memory requirements of almost all existing time integration methods, the only condition being that space and time field approximations can be uncoupled. This condition is shortly restrictive but excludes space–time finite element approximations based on unstructured meshes as used in [1].

Time integration algorithms differ in the way equations of motion are resolved and time derivatives are approximated (see [41,21,54,10,26,16] for different examples). However, all time integration methods end up in a recursive formula that allows an incremental resolution of the problem in time. An alternative could be to recast this incremental procedure over the whole space–time domain. The obtained problem could be identified with a square linear system whose size equals dimension of spatial approximation multiplied by dimension of temporal one. Its resolution would give the whole space–time solution, solving only one linear system. This strategy is not used in practice since it is computationally inefficient compared to an incremental strategy. In this paper, we propose to use it in conjunction with PGD, in order to make space–time problem's resolution efficient.

Indeed, PGD allows to break space–time problem into two non-linear problems, one in space and an other in time, that are resolved alternatively until convergence. Each resolution step consists in solving a linear system in space and an other in time. Thus, depending on the number of resolution steps, complexity of PGD algorithms can be of several orders of magnitude lower than complexity of a brutal resolution of space–time problem. For simplicity suppose $n_S = n_T = n$ and solving a linear system of size $n \times n$ requires $O(n^3)$ operations.[1] As space–time problem is of size $n^2 \times n^2$, its brutal resolution requires $O(n^6)$ operations whearas PGD algorithms complexity is $2\xi * O(n^3)$, where ξ is the resolution steps number.

Following this example, incremental resolution can be done with $O(n^3)$ operations for lumped explicit time schemes and $O(n^4)$ for implicit time schemes. Thus PGD methods can have a complexity of same order of magnitude (or bigger) as classical incremental algorithms. In fact, PGD methods exhibit their full potential when multiple space–time resolutions must be performed (as in the context of non-linear solver [32,13,42,40]) or when problems are defined in higher dimensional spaces (due to physical modelisations [2,11,45,9] or in optimisation context [35,48]). In this paper, we only illustrate potential of PGD method for one transient dynamic problem resolution.

Outline of this paper is as follows: in Section 2, we establish separated representation of elastodynamic problems over space–time domain. Space–time operators are identified and different examples are given for single field and two fields formulations. In Sections 3 and 4, we show how to use these operators within PGD methods in order to construct space–time separated representation of the solution. Particularly, in Section 3, we shortly review classical definition of PGD for single field case and in Section 4, we introduce a new formulation of PGD for multi-field problems, which is the main contribution of this paper. Finally, in Section 5, several numerical simulations are performed with the transient elastodynamic problem with one dimension in space. Importance of time integration schemes on decomposition is highlighted.

2. A strategy based on tensorial formalism to build space–time operators

In this section, we describe a generic strategy that allows to build space–time separated representation of elastodynamic problems. Aims is to identify separated representation of bilinear and linear operators involved in the linear system equivalent to the whole space–time problem. Technical details are based on tensorial formalism and thus multilinear algebra definitions are given in Appendix A. Different examples are given for single field and two fields formulations. Also, particular attentions are paid to boundary conditions. Following classical methodology employed in finite element method [24], we show how they can be introduced in the right hand side of the equivalent space–time problem.

2.1. Recasting space–time problem as a linear system

Let's consider an evolution problem whose unknown field is discretized on a structured space–time mesh with n_S points in space and n_T time instants. We denote by $\underline{\underline{\boldsymbol{Y}}}$ its space–time discret representation with $Y_{ij} = y(x_i, t_j)$. We want to identify a linear system whose direct resolution gives $\underline{\underline{\boldsymbol{Y}}}$. Using tensorial formalism, such linear system can be expressed as:

Problem 1 (*Single field*). Find $\underline{\underline{\boldsymbol{Y}}} \in \mathbb{R}^{n_S \times n_T}$ such that

$$\underline{\underline{\underline{\underline{\boldsymbol{B}}}}} : \underline{\underline{\boldsymbol{Y}}} = \underline{\underline{\boldsymbol{L}}} \quad \text{with} \quad \begin{cases} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}} \in \mathbb{R}^{n_S \times n_S \times n_T \times n_T} \\ \underline{\underline{\boldsymbol{L}}} \in \mathbb{R}^{n_S \times n_T} \end{cases} \tag{1}$$

This linear system is called the space–time problem. It is said to be separable when bilinear and linear operators involved in (1) verify the following properties [4]:

$$\underline{\underline{\underline{\underline{\boldsymbol{B}}}}} = \sum_{k=1}^{n_B} \underline{\underline{\boldsymbol{B}}}^{Sk} \otimes \underline{\underline{\boldsymbol{B}}}^{Tk} \quad \text{and} \quad \underline{\underline{\boldsymbol{L}}} = \sum_{k=1}^{n_L} \underline{\boldsymbol{L}}^{Sk} \otimes \underline{\boldsymbol{L}}^{Tk} \tag{2}$$

where superscripts S and T denote space and time operators, respectively. The technical difficulty is to identify such operators for a given evolution problem.

Let's now consider a two-fields formulation of the evolution problem, where the two unknown fields can be discretized on different meshes in space and time. We denote by $\underline{\underline{\boldsymbol{Y}}}_1$ and $\underline{\underline{\boldsymbol{Y}}}_2$ their discrete representations. Such problem involves two set of coupled equations, that can be recasted into matrix form using a monolithic approach. Thus, a generalization of the space–time problem to multi-field formulations can be written as:

Problem 2 (*Multi-field*). Find $\begin{matrix} \underline{\underline{\boldsymbol{Y}}}_1 \in \mathbb{R}^{n_1^S \times n_1^T} \\ \underline{\underline{\boldsymbol{Y}}}_2 \in \mathbb{R}^{n_2^S \times n_2^T} \end{matrix}$ *such that*

$$\begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{12} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{21} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_1 \\ \underline{\underline{\boldsymbol{Y}}}_2 \end{bmatrix} = \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}_1 \\ \underline{\underline{\boldsymbol{L}}}_2 \end{bmatrix} \quad \text{with} \quad \begin{cases} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{ij} \in \mathbb{R}^{n_i^S \times n_j^S \times n_i^T \times n_j^T} \\ \underline{\underline{\boldsymbol{L}}}_i \in \mathbb{R}^{n_i^S \times n_i^T} \end{cases} \tag{3}$$

This multi-field problem is said to be separable if each component of bilinear and linear operators involved in (3) have the separation properties (2), that is:

[1] This complexity is not representative of practical algorithms in use. However, PGD preserves matrix operators structures (such as sparsity). So important here is that same complexity of linear system solver can be considered for comparison of the different methods.

$$\underline{\underline{\underline{\boldsymbol{B}}}}_{ij} = \sum_{k=1}^{n_B(i,j)} \underline{\underline{\boldsymbol{B}}}_{ij}^{Sk} \otimes \underline{\underline{\boldsymbol{B}}}_{ij}^{Tk} \quad \text{and} \quad \underline{\underline{\boldsymbol{L}}}_i = \sum_{k=1}^{n_L(i)} \underline{\boldsymbol{L}}_i^{Sk} \otimes \underline{\boldsymbol{L}}_i^{Tk} \tag{4}$$

Remark 1. Thanks to separation properties, it is possible to recast the monolithic system (3) in a single field way using state vector in space. The only condition is that the different fields should have the same temporal approximations, that is $n_1^T = n_2^T = n_T$. This will lead to Problem 1 with $\underline{\underline{\boldsymbol{Y}}} \in \mathbb{R}^{(n_1^S+n_2^S)\times n_T}$, $\underline{\underline{\underline{\boldsymbol{B}}}} \in \mathbb{R}^{(n_1^S+n_2^S)\times(n_1^S+n_2^S)\times n_T \times n_T}$ and $\underline{\underline{\boldsymbol{L}}} \in \mathbb{R}^{(n_1^S+n_2^S)\times n_T}$.

2.2. Reference problem

We illustrate construction of previously mentioned operators for the transient elastodynamic problem with one dimension in space. It models traction-compression waves travelling in a linear elastic medium $\Omega = [0, L]$ during time interval $I = [0, T]$. The scalar displacement field is denoted by $u(x, t)$ with $x \in \Omega$ and $t \in I$. The medium is submitted to imposed displacement $u^d(x, t)$ on $\partial\Omega_u \times I$ and punctual external forces $f^d(x_i, t)$ on $\partial\Omega_f \times I$, with $\partial\Omega_f \cup \partial\Omega_u = \partial\Omega$ and $\partial\Omega_f \cap \partial\Omega_u = \varnothing$. Initial state is known and described by initial displacement field $u_0(x)$ and initial velocity field $v_0(x)$. The medium is characterized by its density ρ, its elasticity modulus E and its section A. Then, the strong formulation of the problem can be written as: *find* $u : \Omega \times I \to \mathbb{R}$ *such that*

$$\rho \frac{\partial^2 u}{\partial t^2} = E \frac{\partial^2 u}{\partial x^2} \quad \text{on } \Omega \times I \tag{5a}$$

$$u = u^d \quad \text{on } \partial\Omega_u \times I \tag{5b}$$

$$\frac{\partial u}{\partial x} = \frac{f^d}{EA} \quad \text{on } \partial\Omega_f \times I \tag{5c}$$

$$u = u_0 \quad \text{on } \Omega \times \{0\} \tag{5d}$$

$$\frac{\partial u}{\partial t} = v_0 \quad \text{on } \Omega \times \{0\} \tag{5e}$$

In the following, we use dot convention for time derivatives, that is $\dot{u}(x, t) = \frac{\partial u}{\partial t}$ and $\ddot{u}(x, t) = \frac{\partial^2 u}{\partial t^2}$.

2.3. Weak formulation in space and Newmark family of time integration schemes

We give here a first illustration of the strategy starting from a weak formulation in space and resolution in time with Newmark family of integration schemes. In this case, tensorial formalism is introduced once the problem has been fully approximated over space and time domains.

2.3.1. Space weak formulation

We define the following functional spaces

$$\mathcal{U} = \{u | u \in \mathcal{H}^1(\Omega),\ u = u^d\ \forall x \in \partial\Omega_u\} \tag{6}$$

$$\mathcal{U}_0 = \{u | u \in \mathcal{H}^1(\Omega)\ ,\ u = 0\ \forall x \in \partial\Omega_u\} \tag{7}$$

and scalar products

$$\langle u, v \rangle_\Omega = \int_\Omega A\, u\, v\, \mathrm{d}x \tag{8}$$

$$\langle u, v \rangle_\Omega^{def} = \int_\Omega EA \frac{\mathrm{d}u}{\mathrm{d}x} \frac{\mathrm{d}v}{\mathrm{d}x} \mathrm{d}x \tag{9}$$

$$\langle u, v \rangle_{\partial\Omega_f} = \sum_{i=1}^{n_F} u(x_i)\, v(x_i) \quad \forall \{x_i\}_{i=1,\dots,n_F} \in \partial\Omega_f \tag{10}$$

where n_F is the number of spatial points including in $\partial\Omega_f$. We impose Dirichlet boundary conditions u^d and initial conditions u_0 and v_0 in a strong sense. Thus, a spatial weak formulation of (5) writes: find $u : I \to \mathcal{U}$ such that

$$\mathcal{B}_S(u, u^\star) = \mathcal{L}_S(u^\star)\ ,\ \forall u^\star \in \mathcal{U}_0 \tag{11a}$$

$$u(0) = u_0\ ,\ \dot{u}(0) = v_0 \tag{11b}$$

where bilinear and linear forms in space are defined as:

$$\mathcal{B}_S(u, u^\star) = \langle \rho\ddot{u}, u^\star \rangle_\Omega + \langle u, u^\star \rangle_\Omega^{def} \tag{12}$$

$$\mathcal{L}_S(u^\star) = \langle f^d(x, t), u^\star \rangle_{\partial\Omega_f} \quad \forall t \in I \tag{13}$$

2.3.2. Spatial approximation using finite elements

We introduce spatial approximation using finite elements. This writes for unknown and virtual fields, respectively:

$$u(x, t) = \underline{\boldsymbol{\phi}}(x) \cdot \underline{\boldsymbol{u}}(t) + \underline{\boldsymbol{\phi}}^d(x) \cdot \underline{\boldsymbol{u}}^d(t) \tag{14}$$

$$u^\star(x) = \underline{\boldsymbol{\phi}}(x) \cdot \underline{\boldsymbol{u}}^\star \tag{15}$$

Initial displacement and velocity fields are identified with their values at spatial points, that is $u_0(x) \equiv \underline{\boldsymbol{u}}_0$ and $v_0(x) \equiv \underline{\boldsymbol{v}}_0$. Introduction of finite element approximations (14) and (15) in the weak formulation in space (11) leads to the classical discrete equations of motion, that is: find $\underline{\boldsymbol{u}} : I \to \mathbb{R}^{n_S}$ such that

$$\underline{\underline{\boldsymbol{M}}} \cdot \underline{\ddot{\boldsymbol{u}}}(t) + \underline{\underline{\boldsymbol{K}}} \cdot \underline{\boldsymbol{u}}(t) = \underline{\boldsymbol{F}}(t) \tag{16}$$

$$\underline{\boldsymbol{u}}(0) = \underline{\boldsymbol{u}}_0, \quad \underline{\dot{\boldsymbol{u}}}(0) = \underline{\boldsymbol{v}}_0 \tag{17}$$

where discrete bilinear operators in space are classical mass and stiffness matrices:

$$\underline{\underline{\boldsymbol{M}}} = \int_\Omega \rho A \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\phi}}(x) \mathrm{d}x, \quad \underline{\underline{\boldsymbol{K}}} = \int_\Omega EA \frac{\mathrm{d}\underline{\boldsymbol{\phi}}(x)}{\mathrm{d}x} \otimes \frac{\mathrm{d}\underline{\boldsymbol{\phi}}(x)}{\mathrm{d}x} \mathrm{d}x$$

2.3.3. Temporal approximation using Newmark family of time integration schemes

We now introduce temporal approximation with Newmark family of integration schemes [41]. Time interval is approximated with $n_T + 1$ time instants, that is $I \equiv \{t_i | i = 0, \dots, n_T\}$ where $t_0 = 0$ and $t_{n_T} = T$, and time increment is constant, that is $t_i - t_{i-1} = \Delta t,\ \forall i$. Then, Newmark approximations at time t_i write:

$$\underline{\boldsymbol{u}}(t_i) = \underline{\boldsymbol{u}}(t_{i-1}) + \Delta t \underline{\dot{\boldsymbol{u}}}(t_{i-1}) + \Delta t^2 \left(\frac{1}{2} - \beta\right) \underline{\ddot{\boldsymbol{u}}}(t_{i-1}) + \Delta t^2 \beta \underline{\ddot{\boldsymbol{u}}}(t_i) \tag{17}$$

$$\underline{\dot{\boldsymbol{u}}}(t_i) = \underline{\dot{\boldsymbol{u}}}(t_{i-1}) + \Delta t(1 - \gamma) \underline{\ddot{\boldsymbol{u}}}(t_{i-1}) + \Delta t \gamma \underline{\ddot{\boldsymbol{u}}}(t_i) \tag{18}$$

These approximations require that the acceleration vector $\underline{\ddot{\boldsymbol{u}}}(t)$ must be known at time t_0. This is classicaly done using equation of motion taken at time t_0:

$$\underline{\ddot{\boldsymbol{u}}}(t_0) = \underline{\underline{\boldsymbol{M}}}^{-1} \cdot (\underline{\boldsymbol{F}}(t_0) - \underline{\underline{\boldsymbol{K}}} \cdot \underline{\boldsymbol{u}}(t_0)) \tag{19}$$

One can then resolve problem (16) incrementally, by introducing Newmark approximations (17) and (18) in equation of motion (16) taken at time t_i. The recursive formula, obtained for $i = 1, \dots, n_T$, is initialized with the known boundary conditions (17) and (19).

2.3.4. Space–time reformulation using tensorial formalism

We now reformulate the incremental problem as a unique linear system. To this end, we identify the displacement vector at all time instants with a second order tensor, that is:

$$\underline{\underline{\boldsymbol{u}}} = [\underline{\boldsymbol{u}}(t_1) \cdots \underline{\boldsymbol{u}}(t_{n_T})] \in \mathbb{R}^{n_S \times n_T} \tag{20}$$

In a similar way, $\dot{\underline{\underline{\boldsymbol{u}}}}$ denotes all vectors $\dot{\underline{\boldsymbol{u}}}(t_i)$, $\ddot{\underline{\underline{\boldsymbol{u}}}}$ all vectors $\ddot{\underline{\boldsymbol{u}}}(t_i)$ and $\underline{\underline{\boldsymbol{F}}}$ denotes all vectors $\underline{\boldsymbol{F}}(t_i)$ for $i = 1, \dots, n_T$.

Notice that Newmark approximations (17) and (18) can be rewriten as:

$$-\underline{\boldsymbol{u}}(t_{i-1}) + \underline{\boldsymbol{u}}(t_i) - \Delta t \dot{\underline{\boldsymbol{u}}}(t_{i-1}) - \Delta t^2 \left(\frac{1}{2} - \beta\right) \ddot{\underline{\boldsymbol{u}}}(t_{i-1}) - \Delta t^2 \beta \ddot{\underline{\boldsymbol{u}}}(t_i) = 0 \tag{21}$$

$$-\dot{\underline{\boldsymbol{u}}}(t_{i-1}) + \dot{\underline{\boldsymbol{u}}}(t_i) - \Delta t(1-\gamma)\ddot{\underline{\boldsymbol{u}}}(t_{i-1}) - \Delta t \gamma \ddot{\underline{\boldsymbol{u}}}(t_i) = 0 \tag{22}$$

Then, we first recast Eqs. (21), (22) and (16) taken at all time instants t_i for $i = 1, \dots, n_T$, as a unique system of equations. Thanks to tensorial implementations (A.8) and (A.13), this can be written as:

$$\begin{bmatrix} \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{A}}}_1 & \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{A}}}_2 & \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{A}}}_3 \\ \underline{\underline{\boldsymbol{0}}}_S \otimes \underline{\underline{\boldsymbol{0}}}_T & \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{A}}}_1 & \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{A}}}_4 \\ \underline{\underline{\boldsymbol{K}}} \otimes \underline{\underline{\boldsymbol{I}}}_T & \underline{\underline{\boldsymbol{0}}}_S \otimes \underline{\underline{\boldsymbol{0}}}_T & \underline{\underline{\boldsymbol{M}}} \otimes \underline{\underline{\boldsymbol{I}}}_T \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{u}}} \\ \dot{\underline{\underline{\boldsymbol{u}}}} \\ \ddot{\underline{\underline{\boldsymbol{u}}}} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{u}}(t_0) \otimes \underline{\boldsymbol{A}}_5 + \dot{\underline{\boldsymbol{u}}}(t_0) \otimes \underline{\boldsymbol{A}}_6 + \ddot{\underline{\boldsymbol{u}}}(t_0) \otimes \underline{\boldsymbol{A}}_7 \\ \dot{\underline{\boldsymbol{u}}}(t_0) \otimes \underline{\boldsymbol{A}}_5 + \ddot{\underline{\boldsymbol{u}}}(t_0) \otimes \underline{\boldsymbol{A}}_8 \\ \underline{\underline{\boldsymbol{F}}} \end{bmatrix} \tag{23}$$

where matrices $\underline{\underline{\boldsymbol{A}}}_i$ and vectors $\underline{\boldsymbol{A}}_i$ depend on Newmark parameters β and γ as well as time increment Δt; matrices $\underline{\underline{\boldsymbol{I}}}_S$ and $\underline{\underline{\boldsymbol{I}}}_T$ are identity matrix in space and time; $\underline{\underline{\boldsymbol{0}}}_S$ and $\underline{\underline{\boldsymbol{0}}}_T$ are null matrix in space and time.

Second, we express $\dot{\underline{\underline{\boldsymbol{u}}}}$ and $\ddot{\underline{\underline{\boldsymbol{u}}}}$ in function of $\underline{\underline{\boldsymbol{u}}}$ thanks to the first two equations of (23). We insert the obtained expression of $\ddot{\underline{\underline{\boldsymbol{u}}}}$ in the third equation of (23), and replace $\underline{\boldsymbol{u}}(t_0)$, $\dot{\underline{\boldsymbol{u}}}(t_0)$ and $\ddot{\underline{\boldsymbol{u}}}(t_0)$ with their correspondings values given by (17) and (19).

Finally, after some simplications, we obtain the formulation in displacement of the whole space–time problem, as a unique linear system. This linear system can be identified with Problem 1 and reads:

Example 1 (*Newmark*). Find $\underline{\underline{\boldsymbol{u}}} \in \mathbb{R}^{n_S \times n_T}$ such that

$$\begin{aligned} \left(\underline{\underline{\boldsymbol{K}}} \otimes \underline{\underline{\boldsymbol{N}}}_1 + \underline{\underline{\boldsymbol{M}}} \otimes \underline{\underline{\boldsymbol{N}}}_2\right) : \underline{\underline{\boldsymbol{u}}} = {} & \left(\underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{N}}}_1\right) \\ & : \underline{\underline{\boldsymbol{F}}} + \left(\underline{\underline{\boldsymbol{M}}} \cdot \underline{\boldsymbol{u}}_0\right) \otimes \underline{\boldsymbol{N}}_3 + \left(\underline{\underline{\boldsymbol{M}}} \cdot \underline{\boldsymbol{v}}_0\right) \otimes \underline{\boldsymbol{N}}_4 \\ & + \left(\underline{\boldsymbol{F}}(t_0) - \underline{\underline{\boldsymbol{K}}} \cdot \underline{\boldsymbol{u}}_0\right) \otimes \underline{\boldsymbol{N}}_5 \end{aligned} \tag{24}$$

Newmark bilinear operators in time are given by

$$\underline{\underline{\boldsymbol{N}}}_1 = \beta \begin{bmatrix} \beta & 0 & \cdots & 0 \\ a & \ddots & 0 & \vdots \\ b & \ddots & \ddots & 0 \\ 0 & b & a & \beta \end{bmatrix} \quad \text{with} \quad \begin{cases} a = \frac{1}{2} - 2\beta + \gamma \\ b = \frac{1}{2} + \beta - \gamma \end{cases}$$

$$\underline{\underline{\boldsymbol{N}}}_2 = \frac{\beta}{\Delta t^2} \begin{bmatrix} 1 & 0 & \cdots & 0 \\ -2 & \ddots & 0 & \vdots \\ 1 & \ddots & \ddots & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}, \quad \underline{\boldsymbol{N}}_3 = \frac{\beta}{\Delta t^2} \begin{bmatrix} 1 \\ -1 \\ 0 \\ \vdots \end{bmatrix}$$

$$\underline{\boldsymbol{N}}_4 = \frac{\beta}{\Delta t} \begin{bmatrix} 1 \\ 0 \\ \vdots \end{bmatrix}, \quad \underline{\boldsymbol{N}}_5 = \beta \begin{bmatrix} \frac{1}{2} - \beta \\ \frac{1}{2} + \beta - \gamma \\ 0 \\ \vdots \end{bmatrix}$$

Remark 2. The right hand side of Eq. (16) contains the contribution of boundary conditions due to exterior forces and imposed displacement:

$$\underline{\boldsymbol{F}}(t) = \underline{\boldsymbol{F}}_f^d(t) - \underline{\boldsymbol{F}}_u^d(t) - \underline{\boldsymbol{F}}_{\ddot{u}}^d(t)$$

One can easily writes these contributions under a sum of scalar products of spatial vectors and temporal functions, that is:

$$\underline{\boldsymbol{F}}(t) = \sum_i \underline{\boldsymbol{a}}_i \, b_i(t) \quad \text{with } \underline{\boldsymbol{a}}_i \in \mathbb{R}^{n_S}$$

This is straightforward for exterior forces $f^d(x_i, t)$ since they are imposed at spatial points, we get:

$$\underline{\boldsymbol{F}}_f^d(t) = \sum_{i=1}^{n_F} \underline{\boldsymbol{\phi}}(x_i) f^d(x_i, t) \quad \forall \{x_i\}_{i=1,\dots,n_F} \in \partial\Omega_f$$

Imposed displacement $u^d(x, t)$ is continuously approximated in space throw the finite element basis $\left\{\phi_i^d\right\}_{i=1,\dots,n_U^d}$, where n_U^d denotes the number of spatial points including in $\partial\Omega_u$. After introduction in the space weak form, this leads to two contributions $\underline{\boldsymbol{F}}_u^d(t)$ and $\underline{\boldsymbol{F}}_{\ddot{u}}^d(t)$ due to bilinear forms, see Eq. (12). These contributions thus write as a matrix–vector product. We decompose this product row by row and obtain the following decomposition:

$$\underline{\boldsymbol{F}}_u^d(t) = \int_\Omega EA \frac{\mathrm{d}\underline{\boldsymbol{\phi}}(x)}{\mathrm{d}x} \otimes \frac{\mathrm{d}\underline{\boldsymbol{\phi}}^d(x)}{\mathrm{d}x} \mathrm{d}x \cdot \underline{\boldsymbol{u}}^d(t) = \sum_{i=1}^{n_U^d} \underline{\boldsymbol{K}}_i^d \, u_i^d(t)$$

$$\underline{\boldsymbol{F}}_{\ddot{u}}^d(t) = \int_\Omega \rho A \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\phi}}^d(x) \mathrm{d}x \cdot \ddot{\underline{\boldsymbol{u}}}^d(t) = \sum_{i=1}^{n_U^d} \underline{\boldsymbol{M}}_i^d \ddot{u}_i^d(t)$$

Remark 3. The right hand side of Eq. (24) contains all contributions due to boundary conditions. In particular, initial conditions in displacement and velocity are taken into account as equivalent forces. Also, contribution of exterior forces and imposed displacement are given in a non-separated way as:

$$\underline{\underline{\boldsymbol{F}}} = \underline{\underline{\boldsymbol{F}}}_f^d - \underline{\underline{\boldsymbol{F}}}_u^d - \underline{\underline{\boldsymbol{F}}}_{\ddot{u}}^d$$

One can express these contributions directly as a sum of products of space and time vectors thanks to the decomposition introduced in Remark 2. Hence, we have:

$$\underline{\underline{\boldsymbol{F}}}_f^d = \sum_{i=1}^{n_F} \underline{\boldsymbol{\phi}}_i \otimes \underline{\boldsymbol{f}}_i^d \quad \text{with } \underline{\boldsymbol{f}}_i^d = [f^d(x_i, \Delta t) \cdots f^d(x_i, T)]'$$

$$\underline{\underline{\boldsymbol{F}}}_u^d = \sum_{i=1}^{n_U^d} \underline{\boldsymbol{K}}_i^d \otimes \underline{\boldsymbol{u}}_i^d \quad \text{with } \underline{\boldsymbol{u}}_i^d = [u_i^d(\Delta t) \cdots u_i^d(T)]'$$

$$\underline{\underline{\boldsymbol{F}}}_{\ddot{u}}^d = \sum_{i=1}^{n_U^d} \underline{\boldsymbol{M}}_i^d \otimes \ddot{\underline{\boldsymbol{u}}}_i^d \quad \text{with } \ddot{\underline{\boldsymbol{u}}}_i^d = [\ddot{u}^d(\Delta t) \cdots \ddot{u}^d(T)]'$$

2.4. Space–time weak formulation – One field case

We give here a second illustration of the strategy starting from a space–time weak formulation. In this case, identification of the linear system equivalent to the space–time problem is straightforward. Indeed, tensorial formalism can be introduced at the approximation step, using a structured space–time finite elements mesh. Then, after introduction of approximations in the weak forms, identification of the space–time separated representation of operators is direct.

2.4.1. *Space–time discontinuous Galerkin method*

We consider the following functional spaces:

$$\mathcal{U} = \{u | u \in \mathcal{H}^1(\Omega \times I),\ u = u^d\ \forall x \in \partial\Omega_u \times I\} \tag{25}$$

$$\mathcal{U}_0 = \{u | u \in \mathcal{H}^1(\Omega \times I),\ u = 0\ \forall x \in \partial\Omega_u \times I\} \tag{26}$$

and we use scalar products defined by Eqs. (8) to (10).

As an example, we use the space–time weak formulation of the elastodynamics problem proposed in [25]. In this formulation, time interval is decomposed in N_T subintervals as $I \equiv \bigcup_{i=1,\dots,N_T} I_i$ with $I_i =]t^+_{i-1}, t^-_i[$ where $t^+_0 = 0$ and $t^-_{N_T} = T$. Then, space–time domain is decomposed in N_T subintervals called space–time slabs as $\Omega \times I = \bigcup_{i=1,\dots,N_T} \Omega \times I_i$.

The problem is weakly formulated over each slab $\Omega \times I_i$ and continuity between two slabs is weakly inforced. Writing this formulation over the whole space–time domain, we obtain: find $u \in \mathcal{U}$ such that

$$\mathcal{B}_{\text{STDG-U}}(u, u^\star) = \mathcal{L}_{\text{STDG-U}}(u^\star) \quad \forall u^\star \in \mathcal{U}_0 \tag{27}$$

where space–time bilinear and linear forms are given by:

$$\begin{aligned}\mathcal{B}_{\text{STDG-U}}(u, u^\star) = {} & \sum_{i=1}^{N_T} \int_{I_i} \langle \rho\ddot{u}, \dot{u}^\star \rangle_\Omega \mathrm{dt} + \sum_{i=1}^{N_T} \int_{I_i} \langle u, \dot{u}^\star \rangle^{def}_\Omega \mathrm{dt} \\ & + \sum_{i=2}^{N_T} \langle \rho\dot{u}(x, t^+_{i-1}) - \rho\dot{u}(x, t^-_{i-1}), \dot{u}^\star(x, t^+_{i-1}) \rangle_\Omega \\ & + \sum_{i=2}^{N_T} \langle u(x, t^+_{i-1}) - u(x, t^-_{i-1}), u^\star(x, t^+_{i-1}) \rangle^{def}_\Omega \\ & + \langle \rho\dot{u}(x, 0), \dot{u}^\star(x, 0) \rangle_\Omega \\ & + \langle u(x, 0), u^\star(x, 0) \rangle^{def}_\Omega \end{aligned} \tag{28}$$

$$\begin{aligned}\mathcal{L}_{\text{STDG-U}}(u^\star) = {} & \sum_{i=1}^{N_T} \int_{I_i} \langle f^d(x, t), \dot{u}^\star \rangle_{\partial\Omega_f} \mathrm{dt} + \langle \rho\, v_0(x), \dot{u}^\star(x, 0) \rangle_\Omega \\ & + \langle u_0(x), u^\star(x, 0) \rangle^{def}_\Omega \end{aligned} \tag{29}$$

The first two terms in (28) and the first term in (29) act to weakly inforce the equation of motion over all space–time slabs, while the remaining terms weakly enforce continuity of displacement and velocity between space–time slabs. In particular, initial displacement and velocity are weakly inforced throw the two last terms in (28) and (29).

2.4.2. *Space and time approximations*

We introduce approximations using continuous finite elements in space and piecewise continuous in time. Displacement $u^d(x, t)$ is imposed in a strong way. Initial conditions $u_0(x)$ and $v_0(x)$ are weakly imposed. This writes for unknown and virtual fields as

$$u(x, t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}(t) : \underline{\underline{\boldsymbol{u}}} + \underline{\boldsymbol{\phi}}^d(x) \otimes \underline{\boldsymbol{\psi}}(t) : \underline{\underline{\boldsymbol{u}}}^d \tag{30}$$

$$u^\star(x, t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}(t) : \underline{\underline{\boldsymbol{u}}}^\star \tag{31}$$

External load $f^d(x_i, t)$ is approximated in time with the same temporal basis as $u(x, t)$.

2.4.3. *Equivalent space–time problem*

Introducing finite element approximations (30) and (31) in (27) and using tensorial notations directly lead to the space–time representation of the problem as a unique linear system. This system has the general form of Problem 1, and can be written has:

Example 2 (*TDG-u*). Find $\underline{\underline{\boldsymbol{u}}} \in \mathbb{R}^{n_S \times n_T}$ such that

$$\begin{aligned} & \left(\underline{\underline{\boldsymbol{K}}} \otimes (\underline{\underline{\boldsymbol{Q}}}^{10} + \underline{\underline{\boldsymbol{\Psi}}}^{00}) + \underline{\underline{\boldsymbol{M}}} \otimes (\underline{\underline{\boldsymbol{Q}}}^{12} + \underline{\underline{\boldsymbol{\Psi}}}^{11})\right) : \underline{\underline{\boldsymbol{u}}} \\ & \quad = (\underline{\underline{\boldsymbol{K}}} \cdot \underline{\boldsymbol{u}}_0) \otimes \underline{\boldsymbol{\Psi}}^0_0 + (\underline{\underline{\boldsymbol{M}}} \cdot \underline{\boldsymbol{v}}_0) \otimes \underline{\boldsymbol{\Psi}}^1_0 + (\underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{10}) \\ & \quad : \underline{\underline{\boldsymbol{F}}}^d_f - (\underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{10}) : \underline{\underline{\boldsymbol{F}}}^d_u - (\underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{12}) : \underline{\underline{\boldsymbol{F}}}^d_{\ddot{u}} \end{aligned} \tag{32}$$

Bilinear operators in time associated with the time discontinuous Galerkin method are given by:

$$\begin{aligned}\underline{\underline{\boldsymbol{\Psi}}}^{ij} = {} & \frac{d^i\underline{\boldsymbol{\psi}}(0)}{dt^i} \otimes \frac{d^j\underline{\boldsymbol{\psi}}(0)}{dt^j} \\ & + \sum_{i=2}^{N_T} \left(\frac{d^i\underline{\boldsymbol{\psi}}(t^+_{i-1})}{dt^i} \otimes \frac{d^j\underline{\boldsymbol{\psi}}(t^+_{i-1})}{dt^j} - \frac{d^i\underline{\boldsymbol{\psi}}(t^+_{i-1})}{dt^i} \otimes \frac{d^j\underline{\boldsymbol{\psi}}(t^-_{i-1})}{dt^j} \right)\end{aligned}$$

$$\underline{\boldsymbol{\Psi}}^i_0 = \frac{d^i\underline{\boldsymbol{\psi}}(0)}{dt^i} \quad \text{and} \quad \underline{\underline{\boldsymbol{Q}}}^{ij} = \int_I \frac{d^i\underline{\boldsymbol{\psi}}(t)}{dt^i} \otimes \frac{d^j\underline{\boldsymbol{\psi}}(t)}{dt^j} \mathrm{dt}$$

Remark 4. The right hand side of Eq. (32) contains all contribution due to boundary conditions. In particular, initial displacement and velocity fields are weakly imposed. Indeed, the matrix of unknowns $\underline{\underline{\boldsymbol{u}}}$ includes projection coefficients associated with $u(x, t = 0)$. Contributions due to exterior forces $f^d(x, t)$ and imposed displacement $u^d(x, t)$ need to be decomposed, in order to be identified as a sum of products of space and time vectors. Following the same method as used in Remark 3, we can directly write these decompositions as:

$$\underline{\underline{\boldsymbol{F}}}^d_f = \sum_{i=1}^{n_F} \underline{\phi_i} \otimes \underline{\boldsymbol{f}}^d_i \quad \text{with} \underline{\boldsymbol{f}}^d_i = [f^d(x_i, 0) \cdots f^d(x_i, T)]' \tag{33}$$

$$\underline{\underline{\boldsymbol{F}}}^d_u = \sum_{i=1}^{n^d_U} \underline{\boldsymbol{K}}^d_i \otimes \underline{\boldsymbol{u}}^d_i \quad \text{with} \underline{\boldsymbol{u}}^d_i = [u^d_i(0) \cdots u^d_i(T)]' \tag{34}$$

$$\underline{\underline{\boldsymbol{F}}}^d_{\ddot{u}} = \sum_{i=1}^{n^d_U} \underline{\boldsymbol{M}}^d_i \otimes \underline{\boldsymbol{u}}^d_i \tag{35}$$

2.5. *Space–time weak formulation – Two fields case*

As a last illustration, we use a multi-field formulation of the problem. Here, unknowns are displacement and velocity fields. Using different approximations in time for both fields, the linear system equivalent to the space–time problem can be identified with the general multi-field Problem 2. Also, it can be recasted in a single field form (as Problem 1) using a state vector in space and same approximations in time.

2.5.1. *Two-field space–time discontinuous Galerkin method*

We use functional spaces and scalar products respectively defined by (25), (26), (8) and (10). In addition, we introduce the following functional spaces:

$$\mathcal{V} = \{v | v \in \mathcal{H}^1(\Omega \times I),\ v = v^d\, \forall x \in \partial\Omega_u \times I\} \tag{36}$$

$$\mathcal{V}_0 = \{v | v \in \mathcal{H}^1(\Omega \times I),\ v = 0\, \forall x \in \partial\Omega_u \times I\} \tag{37}$$

where v^d is the velocity associated to imposed displacement u^d, that is $v^d = \dot{u}^d$.

We use the two-fields version proposed in [26] of previously mentioned space–time weak formulation of the elastodynamic problem. As for the one field case, space–time domain $\Omega \times I$ is decomposed in N_T space–time slabs. But displacement $u(x, t)$ and velocity $v(x,t)$ are two distinct fields. Then, continuity between the first time derivative of the displacement field and the velocity field is weakly imposed. This space–time weak formulation is written over the space–time domain as: it find $\begin{cases} u \in \mathcal{U} \\ v \in \mathcal{V} \end{cases}$ such that $\begin{cases} \forall u^\star \in \mathcal{U}_0 \\ \forall v^\star \in \mathcal{V}_0 \end{cases}$

$$\mathcal{B}_{\text{STDG-UV}}\left(\{u, v\}, \{u^\star, v^\star\}\right) = \mathcal{L}_{\text{STDG-UV}}\left(\{u^\star, v^\star\}\right) \tag{38}$$

where space–time bilinear and linear forms are given by:

$$\begin{aligned}
&\mathcal{B}_{\text{STDG-UV}}\left(\{u, v\}, \{u^\star, v^\star\}\right) \\
&= \sum_{i=1}^{N_T} \int_{I_i} \langle \rho \dot{v}, v^\star \rangle_\Omega \mathrm{d}t + \sum_{i=1}^{N_T} \int_{I_i} \langle u, v^\star \rangle_\Omega^{def} \mathrm{d}t + \sum_{i=1}^{N_T} \int_{I_i} \langle \dot{u} - v, u^\star \rangle_\Omega^{def} \mathrm{d}t \\
&\quad + \sum_{i=2}^{N_T} \left\langle \rho v(x, t_{i-1}^+) - \rho v(x, t_{i-1}^-), v^\star(x, t_{i-1}^+) \right\rangle_\Omega \\
&\quad + \sum_{i=2}^{N_T} \left\langle u(x, t_{i-1}^+) - u(x, t_{i-1}^-), u^\star(x, t_{i-1}^+) \right\rangle_\Omega^{def} + \langle \rho v(x,0), v^\star(x,0) \rangle_\Omega \\
&\quad + \langle u(x,0), u^\star(x,0) \rangle_\Omega^{def}
\end{aligned} \tag{39}$$

$$\begin{aligned}
\mathcal{L}_{\text{STDG-UV}}\left(\{u^\star, v^\star\}\right) &= \sum_{i=1}^{N_T} \int_{I_i} \langle f^d(x,t), v^\star \rangle_{\partial\Omega_f} \mathrm{d}t + \langle \rho v_0(x), v^\star(x,0) \rangle_\Omega \\
&\quad + \langle u_0(x), u^\star(x,0) \rangle_\Omega^{def}
\end{aligned} \tag{40}$$

The third term in (39) weakly inforced the continuity between the displacement and velocity fields over time. Remaining terms are similar to those of the one field case and have been previously described.

2.5.2. Space and time approximations

We introduce approximations using continuous finite elements in space and piecewise continuous in time for both fields. We use the same approximation in space but allow different approximations in time. This writes for unknown and virtual fields:

$$u(x,t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}_u(t) : \underline{\underline{\boldsymbol{u}}} + \underline{\boldsymbol{\phi}}^d(x) \otimes \underline{\boldsymbol{\psi}}_u(t) : \underline{\underline{\boldsymbol{u}}}^d \tag{41}$$

$$v(x,t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}_v(t) : \underline{\underline{\boldsymbol{v}}} + \underline{\boldsymbol{\phi}}^d(x) \otimes \underline{\boldsymbol{\psi}}_v(t) : \underline{\underline{\boldsymbol{v}}}^d \tag{42}$$

$$u^\star(x,t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}_u(t) : \underline{\underline{\boldsymbol{u}}}^\star \tag{43}$$

$$v^\star(x,t) = \underline{\boldsymbol{\phi}}(x) \otimes \underline{\boldsymbol{\psi}}_v(t) : \underline{\underline{\boldsymbol{v}}}^\star \tag{44}$$

2.5.3. Equivalent space–time problem: multi-field approach

Introducing finite element approximations (41) to (44) in (38) and recasting it with tensorial implementation (A.13), one directly obtains the linear system equivalent to the space–time problem. This linear system can be identified with the generic multi-field Problem 2. This writes:

Example 3 (*TDG-uv with multi-field approach*). Find $\begin{cases} \underline{\underline{\boldsymbol{u}}} \in \mathbb{R}^{n_S \times n_u^T} \\ \underline{\underline{\boldsymbol{v}}} \in \mathbb{R}^{n_S \times n_v^T} \end{cases}$ such that

$$\begin{aligned}
&\begin{bmatrix} \underline{\underline{\boldsymbol{K}}} \otimes \left(\underline{\underline{\boldsymbol{Q}}}_{uu}^{01} + \underline{\underline{\boldsymbol{\Psi}}}_{uu} \right) & -\underline{\underline{\boldsymbol{K}}} \otimes \underline{\underline{\boldsymbol{Q}}}_{uv}^{00} \\ \underline{\underline{\boldsymbol{K}}} \otimes \underline{\underline{\boldsymbol{Q}}}_{vu}^{00} & \underline{\underline{\boldsymbol{M}}} \otimes \left(\underline{\underline{\boldsymbol{Q}}}_{vv}^{01} + \underline{\underline{\boldsymbol{\Psi}}}_{vv} \right) \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{u}}} \\ \underline{\underline{\boldsymbol{v}}} \end{bmatrix} \\
&= \begin{bmatrix} \left(\underline{\underline{\boldsymbol{K}}} \cdot \underline{\boldsymbol{u}}_0 \right) \otimes \underline{\boldsymbol{\Psi}}_{u0} \\ \left(\underline{\underline{\boldsymbol{M}}} \cdot \underline{\boldsymbol{v}}_0 \right) \otimes \underline{\boldsymbol{\Psi}}_{v0} \end{bmatrix} + \begin{bmatrix} \underline{\boldsymbol{0}} \\ \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}_{vv}^{00} : \underline{\underline{\boldsymbol{F}}}_f^d \end{bmatrix} \\
&\quad - \begin{bmatrix} \underline{\boldsymbol{0}} \\ \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}_{vu}^{00} : \underline{\underline{\boldsymbol{F}}}_u^d \end{bmatrix} - \begin{bmatrix} \underline{\boldsymbol{0}} \\ \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}_{vv}^{01} : \underline{\underline{\boldsymbol{F}}}_v^d \end{bmatrix}
\end{aligned} \tag{45}$$

Bilinear operators in time associated with the two fields time discontinuous Galerkin method are given by:

$$\underline{\underline{\boldsymbol{\Psi}}}_{uu} = \underline{\boldsymbol{\psi}}_u(0) \otimes \underline{\boldsymbol{\psi}}_u(0) + \sum_{i=2}^{N_T} \left(\underline{\boldsymbol{\psi}}_u(t_{i-1}^+) \otimes \underline{\boldsymbol{\psi}}_u(t_{i-1}^+) - \underline{\boldsymbol{\psi}}_u(t_{i-1}^+) \otimes \underline{\boldsymbol{\psi}}_u(t_{i-1}^-) \right)$$

$$\underline{\boldsymbol{\Psi}}_{u0} = \underline{\boldsymbol{\psi}}_u(0) \quad \text{and} \quad \underline{\underline{\boldsymbol{Q}}}_{uv}^{ij} = \int_I \frac{d^i \underline{\boldsymbol{\psi}}_u(t)}{dt^i} \otimes \frac{d^j \underline{\boldsymbol{\psi}}_v(t)}{dt^j} \mathrm{d}t$$

Operators $\underline{\underline{\boldsymbol{Q}}}_{uu}^{ij}$, $\underline{\underline{\boldsymbol{Q}}}_{vv}^{ij}$ are defined similarly to $\underline{\underline{\boldsymbol{Q}}}_{uv}^{ij}$; operator $\underline{\underline{\boldsymbol{\Psi}}}_{vv}$ similarly to $\underline{\underline{\boldsymbol{\Psi}}}_{uu}$ and operator $\underline{\boldsymbol{\Psi}}_{v0}$ similarly to $\underline{\boldsymbol{\Psi}}_{u0}$.

Remark 5. The right hand side of Eq. (45) contains a contribution due to the imposed velocity v^d (other contributions are defined in Remark 4). This contribution can be decomposed as:

$$\underline{\underline{\boldsymbol{F}}}_v^d = \sum_{i=1}^{n_v^d} \underline{\underline{\boldsymbol{M}}}_i^d \otimes \underline{\boldsymbol{v}}_i^d \quad \text{with } \underline{\boldsymbol{v}}_i^d = [v_i^d(0) \cdots v_i^d(T)]' \tag{46}$$

2.5.4. Equivalent space–time problem: state vector approach

The multi-field problem (45) can be recasted into a single field problem using a state vector approach. To do this, we defined the following state vector $\underline{\boldsymbol{y}}$:

$$\underline{\boldsymbol{y}}(x,t) = \begin{bmatrix} u(x,t) \\ v(x,t) \end{bmatrix} \tag{47}$$

Then, we assume the same approximations in time for the displacement and velocity fields, that is $\underline{\boldsymbol{\psi}}_u(t) = \underline{\boldsymbol{\psi}}_v(t) = \underline{\boldsymbol{\psi}}(t)$. By this way, the space–time approximation of the state vector reads:

$$\underline{\boldsymbol{y}}(x,t) = \underline{\underline{\boldsymbol{\phi}}}(x) \otimes \underline{\boldsymbol{\psi}}(t) : \underline{\underline{\boldsymbol{y}}} + \underline{\underline{\boldsymbol{\phi}}}^d(x) \otimes \underline{\boldsymbol{\psi}}(t) : \underline{\underline{\boldsymbol{y}}}^d \tag{48}$$

with the following implementations:

$$\underline{\underline{\boldsymbol{\phi}}} = \begin{bmatrix} \phi_1 & \cdots & \phi_{n_S} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \phi_1 & \cdots & \phi_{n_S} \end{bmatrix} \quad \text{and} \quad \underline{\underline{\boldsymbol{y}}} = \begin{bmatrix} u_1(0) & \cdots & u_1(T) \\ \vdots & \cdots & \vdots \\ u_{n_S}(0) & \cdots & u_{n_S}(T) \\ v_1(0) & \cdots & v_1(T) \\ \vdots & \cdots & \vdots \\ v_{n_S}(0) & \cdots & v_{n_S}(T) \end{bmatrix}$$

Finally, introduction of approximation (48) in the weak form (38) gives the following equivalent linear system, that can be identified with the generic one field Problem 1. That is:

Example 4 (*TDG-uv with state vector approach*). Find $\underline{\underline{\boldsymbol{y}}} \in \mathbb{R}^{2n_S \times n_T}$ such that

$$\begin{aligned}
&\left(\underline{\underline{\boldsymbol{B}}}^{S1} \otimes \left(\underline{\underline{\boldsymbol{Q}}}^{01} + \underline{\underline{\boldsymbol{\Psi}}} \right) + \underline{\underline{\boldsymbol{B}}}^{S2} \otimes \underline{\underline{\boldsymbol{Q}}}^{00} \right) : \underline{\underline{\boldsymbol{y}}} = \underline{\boldsymbol{y}}_0 \otimes \underline{\boldsymbol{\Psi}}_0 + \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{00} \\
&\quad : \widetilde{\underline{\underline{\boldsymbol{F}}}}_f^d - \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{00} : \widetilde{\underline{\underline{\boldsymbol{F}}}}_u^d - \underline{\boldsymbol{I}}_S \otimes \underline{\underline{\boldsymbol{Q}}}^{01} : \widetilde{\underline{\underline{\boldsymbol{F}}}}_v^d
\end{aligned} \tag{49}$$

where bilinear operators in time are implemented similarly to the multi-field approach whereas bilinear operators in space are implemented as follow:

$$\underline{\underline{\boldsymbol{B}}}^{S1} \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{K}}} & \underline{\underline{\boldsymbol{0}}}_S \\ \underline{\underline{\boldsymbol{0}}}_S & \underline{\underline{\boldsymbol{M}}} \end{bmatrix} \quad \text{and} \quad \underline{\underline{\boldsymbol{B}}}^{S2} \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{0}}}_S & -\underline{\underline{\boldsymbol{K}}} \\ \underline{\underline{\boldsymbol{K}}} & \underline{\underline{\boldsymbol{0}}}_S \end{bmatrix}$$

and linear forms are expressed thanks to:

$$\underline{\boldsymbol{y}}_0 \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{K}}}.\underline{\boldsymbol{u}}_0 \\ \underline{\underline{\boldsymbol{M}}}.\underline{\boldsymbol{v}}_0 \end{bmatrix} \quad \underline{\underline{\widetilde{\boldsymbol{F}}}}{}_f^d \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{0}}} \\ \underline{\underline{\boldsymbol{F}}}{}_f^d \end{bmatrix} \quad \underline{\underline{\widetilde{\boldsymbol{F}}}}{}_u^d \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{0}}} \\ \underline{\underline{\boldsymbol{F}}}{}_u^d \end{bmatrix} \quad \underline{\underline{\widetilde{\boldsymbol{F}}}}{}_v^d \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{0}}} \\ \underline{\underline{\boldsymbol{F}}}{}_v^d \end{bmatrix}$$

3. The proper generalized decomposition for non-symmetric operators

In this section, we shortly review classical PGD methods that allow to build a space–time separated representation of the solution of an evolution problem. We only consider one field cases and problems formulated in the form of the generic Problem 1.

As in [5,48], we adopt an algebraic point of view. This allows to design generic PGD solvers that can be used with various time integration methods. Then, PGD can be viewed as an alternative to [7] for efficient iterative resolution of linear system given in tensor like format.

As shown in Section 2, contributions of boundary conditions (such that Neumann and Dirichlet conditions over time, and also initial conditions) can be expressed as equivalent external loads. Therefore, there is no need for special procedures (as proposed in [20,9]) in order to imposed boundary conditions, since they directly appear in the right hand side of the space–time problem.

3.1. Separated representation

The PGD aims at finding *a priori* a separated representation of $\underline{\underline{\boldsymbol{Y}}}$, the space–time problem solution. We mean by *a priori*, that $\underline{\underline{\boldsymbol{Y}}}$ needs not to be known (that is nor calculated or stored) before calculation of its decomposition. We only need the separated representation of operators involved in the space–time problem, see Eqs. (2).

Adopting a discrete point of view, the separated representation is defined as a sum of tensorial products of space and time vectors. It was first introduced in the framework of the LATIN method under the name "radial approximation" (see [31,42]). In this paper, we adopt a slightly different representation, that has been used in the context of multidimensional problems (see [8,2,11]). We are seeking $\underline{\underline{\boldsymbol{Y}}}$ as a linear combination of normalized tensorial products of space and time vectors, that is we are seeking $\underline{\underline{\boldsymbol{Y}}}$ under the form:

$$\underline{\underline{\boldsymbol{Y}}}_m = \sum_{i=1}^{m} \alpha_i \underline{\boldsymbol{W}}_i \otimes \underline{\boldsymbol{\Lambda}}_i \quad \text{with} \quad \begin{cases} \underline{\boldsymbol{W}}_i \in \mathbb{R}^{n_S} \\ \underline{\boldsymbol{\Lambda}}_i \in \mathbb{R}^{n_T} \\ \alpha_i \in \mathbb{R} \end{cases} \tag{50}$$

where integer m is the decomposition rank, scalar coefficients α_i are the separation values and products $\underline{\boldsymbol{W}}_i \otimes \underline{\boldsymbol{\Lambda}}_i$ are the space–time modes. These are normalized with respect to a chosen metric. To this end, we denote by

$$\|\underline{\underline{\square}}\|_{\underline{\underline{\underline{\boldsymbol{N}}}}} = \sqrt{\underline{\underline{\square}} : \underline{\underline{\underline{\boldsymbol{N}}}} : \underline{\underline{\square}}}$$

the norm of a second order tensor $\underline{\underline{\square}}$ associated with bilinear operator $\underline{\underline{\underline{\boldsymbol{N}}}}$. Also, we impose the following normality property to the decomposition:

$$\|\underline{\boldsymbol{W}}_i \otimes \underline{\boldsymbol{\Lambda}}_i\|_{\underline{\underline{\underline{\boldsymbol{N}}}}} = 1, \quad \forall i = 1, \ldots, m \tag{51}$$

The main advantage of this normalization is that separation values can be used to directly quantify (in a chosen metric) the contribution of a given space–time mode to the whole decomposition. This will be usefull in order to reordonnate the decomposition in cases where non-optimal algorithms where used to build the separated representation. We discuss this aspect in Section 3.4.

3.2. Classical definitions

The PGD is classically defined as the solution of a minimization problem. The space and time vectors are seeking as the ones which minimize an associated functional $J : \mathbb{R}^{n_S \times n_T} \to \mathbb{R}^+$. This functional can be defined using Galerkin orthogonality criteria [3,43] or a minimal residual formulation [8,42]. These definitions are summarized in Table 1.

The PGD converges if it tends to the space–time solution as the decomposition rank tends to infinity. For symmetric problems, convergence can be proven for both PGD definitions; but for non-symmetric problems, convergence is guaranteed only with the minimal residual definition (see [43,34,5] for convergence analysis). However, convergence of Galerkin based PGD has been numerically observed for non-symmetric problems (see for example results obtain in [3,46] with parabolic problems). In the elastodynamics case, bilinear operators associated to the space–time problem are non-symmetric (due to unsymmetrical nature of temporal operators). Then, minimal residual PGD should be used to insure decomposition convergence.

A rank m PGD is said to be optimal if it minimizes the distance (in a given metric) between the reference solution $\underline{\underline{\boldsymbol{Y}}}$ and all possible separated representations of rank m. Optimality of PGDs can be evaluated by expressing functional J in an *a posteriori* way (that is in function of the reference solution $\underline{\underline{\boldsymbol{Y}}}$). For both *a priori* definitions of the functional J, it is easy to find an equivalent *a posteriori* functional whose minimization leads to the same problem (see Table 2).

Then, *a posteriori* definitions of PGD can be compared with the Singular Value Decomposition (SVD) of matrix $\underline{\underline{\boldsymbol{Y}}}$. Indeed, SVD can be defined as the one which minimizes the distance, in the Frobenius norm (denoted by $\|\underline{\underline{\square}}\|_2 = \sqrt{\underline{\underline{\square}} : \underline{\underline{\square}}}$), between the matrix $\underline{\underline{\boldsymbol{Y}}}$ and all separated representation of rank m (see [50]). In this case, m is the matrix rank and the coefficients α_i are its singular values. Then, SVD gives the optimal decomposition in the $\|..\|_2$ metric, while Galerkin PGD gives the optimal decomposition in the $\|..\|_{\underline{\underline{\underline{\boldsymbol{B}}}}}$ metric (if it exists) and residual PGD in the $\|..\|_{\underline{\underline{\underline{\boldsymbol{B}}}}' : \underline{\underline{\underline{\boldsymbol{B}}}}}$ metric.

Remark 6. A new definition of PGD, called minimax PGD, has been recently introduced by Nouy in [46]. It is based on Petrov–Galerkin criteria and allows to improve convergence properties of Galerkin based PGD with respect to a desired metric. We have tested this formulation for the elastodynamic problems with respect to the Frobienus metric but experienced divergence of the decomposition after some PGD iterations. Also, when applied to the symmetrized problem, we do not obtain better results than with minimal residual PGD. Thus, we do not consider the minimax definition in this paper.

Table 1
a priori definitions of functional $J : \mathbb{R}^{n_S \times n_T} \to \mathbb{R}^+$.

$J(\underline{\underline{\boldsymbol{Y}}}^\star)$	Galerkin PGD	Residual PGD
a priori	$\lvert\underline{\underline{\boldsymbol{Y}}}^\star : \underline{\underline{\boldsymbol{L}}} - \frac{1}{2}(\underline{\underline{\boldsymbol{Y}}}^\star : \underline{\underline{\underline{\boldsymbol{B}}}} : \underline{\underline{\boldsymbol{Y}}}^\star)\rvert$	$\|\underline{\underline{\boldsymbol{L}}} - \underline{\underline{\underline{\boldsymbol{B}}}} : \underline{\underline{\boldsymbol{Y}}}^\star\|_2^2$

Table 2
Equivalent *a posteriori* definitions of functional J and comparison with SVD of the space–time solution $\underline{\underline{\boldsymbol{Y}}}$.

$J(\underline{\underline{\boldsymbol{Y}}}^\star)$	Galerkin PGD	Residual PGD	SVD
a posteriori	$\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}^\star\|_{\underline{\underline{\underline{\boldsymbol{B}}}}}^2$	$\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}^\star\|_{\underline{\underline{\underline{\boldsymbol{B}}}}' : \underline{\underline{\underline{\boldsymbol{B}}}}}^2$	$\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}^\star\|_2^2$

3.3. Algorithms

The simplest way to compute PGD is based on a progressive construction [8,2,46]. In order to compute decomposition of rank m, we suppose decomposition of rank $m-1$ has been previously calculated and search for the best rank one enrichment for which functional J reaches its minimum value. The minimization process is alternatively performed with respect to space and time vectors, leading to two stationnary conditions. In this paper, we additionaly imposed normality of the enrichment with respect to a chosen metric. This allows to calculate separation values. Then, progressive definition of PGD reads:

Definition 1 (*Progressive PGD*).
Knowing $\underline{\underline{\boldsymbol{Y}}}_{(m-1)}$, find $\alpha_m, \underline{\boldsymbol{W}}_m$ and $\underline{\boldsymbol{\Lambda}}_m$ such that

$$(\underline{\boldsymbol{W}}_m, \underline{\boldsymbol{\Lambda}}_m) = \arg \min_{\substack{\underline{\boldsymbol{W}}^\star \in \mathbb{R}^n \\ \underline{\boldsymbol{\Lambda}}^\star \in \mathbb{R}^{n_T}}} J\left(\underline{\underline{\boldsymbol{Y}}}_{(m-1)} + \alpha_m \underline{\boldsymbol{W}}^\star \otimes \underline{\boldsymbol{\Lambda}}^\star\right) \tag{52a}$$

$$\|\underline{\boldsymbol{W}}_m \otimes \underline{\boldsymbol{\Lambda}}_m\|^2_{\underline{\underline{\underline{\underline{\boldsymbol{N}}}}}} = 1 \tag{52b}$$

This contruction of PGD is optimal only if bilinear operator involved in the space–time problem admits a rank one decomposition [46], that is if:

$$\underline{\underline{\underline{\underline{\boldsymbol{B}}}}} = \underline{\underline{\boldsymbol{B}}}^S \otimes \underline{\underline{\boldsymbol{B}}}^T \tag{53}$$

However, this is not the case for elastodynamic problems and then progressive construction of PGD is not optimal. An optimal definition can be introduced by minimizing functional J with respect to all space and time vectors [44], that is:

Definition 2 (*Optimal PGD*). Find $\{\alpha_i, \underline{\boldsymbol{W}}_i, \underline{\boldsymbol{\Lambda}}_i\}_{i=1,\dots,m}$ such that

$$\{\underline{\boldsymbol{W}}_i, \underline{\boldsymbol{\Lambda}}_i\}_{i=1,\dots,m} = \arg \min_{\substack{\underline{\boldsymbol{W}}^\star_1,\dots,\underline{\boldsymbol{W}}^\star_m \in \mathbb{R}^{n_S \times m} \\ \underline{\boldsymbol{\Lambda}}^\star_1,\dots,\underline{\boldsymbol{\Lambda}}^\star_m \in \mathbb{R}^{n_T \times m}}} J\left(\sum_{i=1}^{m} \alpha_i \underline{\boldsymbol{W}}^\star_i \otimes \underline{\boldsymbol{\Lambda}}^\star_i\right) \tag{54a}$$

$$\|\underline{\boldsymbol{W}}_i \otimes \underline{\boldsymbol{\Lambda}}_i\|_{\underline{\underline{\underline{\underline{\boldsymbol{N}}}}}} = 1, \quad \forall i = 1,\dots,m \tag{54b}$$

One way to resolve this problem is based on subspace iteration algorithm [44]. It consists in solving one system in space of size $m * n_S \times m * n_S$ and an other in time of size $m * n_T \times m * n_T$ alternatively. So, this algorithm requires the resolution of spatial and temporal problems whose size increases with the decomposition rank. Thus, it becomes computationaly untractable as decomposition rank increases. An alternative was proposed in [46]. It consists in constructing the decomposition in a progressive way, and updating all temporal vectors with the optimal problem, at the end of each progressive step. This procedure still requires resolution of the optimal problem in time whose size increases with the decomposition rank. It is then tractable only if temporal discretization remains coarse. But in the context of elastodynamics, describtion of waves propagation often requires a fine discretization in time. Thus, in this paper, we propose an other updating strategy which only requires resolution of spatial and temporal problems of size $n_S \times n_S$ and $n_T \times n_T$. Our strategy can be viewed as a Gauss–Seidel type substitution method for resolution of the optimal problem. Implementation details are given in Appendix B.

3.4. Convergence and reordering

Separation values α_i are of great interest since they contain informations related to convergence of decomposition. In particular, minimal residual definition of PGD verify the following convergence property [46]:

Property 1 (*Convergence property – Minimal residual PGD*). *Let $\underline{\underline{\boldsymbol{Y}}}_m$ be the rank m minimal residual PGD of $\underline{\underline{\boldsymbol{Y}}}$, constructed in a progressive way. Let $\underline{\underline{\underline{\underline{\boldsymbol{N}}}}} = \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}' : \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}$ be the operator norm used to normalized space–time modes. Then the decomposition verifies the following property:*

$$\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}_m\|^2_{\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}':\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}} = \|\underline{\underline{\boldsymbol{Y}}}\|^2_{\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}':\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}} - \sum_{i=1}^{m} \alpha_i^2 \tag{55}$$

Thus, separation values α_i exhibit contribution of each space–time mode to the whole decomposition, without any postprocessing. Then, if the computed decomposition is not sorted in a decreasing order (that is if coefficients α_i are not decreasing for $i = 1,\dots,m$), it is possible to rearrange the decomposition in decreasing order using a permutation of coefficients α_i and corresponding space–time modes. Such reordering is negligible in term of computational costs. More over, using equivalence between norms defined in finite dimensional spaces, we can guaranty decomposition convergence when space–time modes are normalized with a different norm than the norm used to defined J (see Table 2). In particular, decomposition will converge if we impose normality of space–time modes with respect to Frobenius norm $\|\underline{\underline{\square}}\|_2$ and define PGD with minimal residual formulation.

4. A new multi-field proper generalized decomposition

In this section, we introduce a new PGD method that allows to efficiently compute space–time separated representation of solution of multi-field evolution problems. For the sake of clarity, we only consider generic two-field problems of the form of Problem 2 but the following developments can be directly generalized to n-field problems.

We have shown in Section 2 that multi-field problems can be recasted into one field problems using state vector in space and same temporal approximations. Then, it is possible to use one-field PGD methods as describe in Section 3 to get a separated representation of both fields. However, we have experienced bad convergence order using this strategy. Furthermore, temporal approximation must be the same for all fields. These limitations have motivated the proposition of a new approach.

Traditionally, one distinguishes two classes of methods for the resolution of multi-field problems: partitioned procedures and monolithic approaches. Partitioned procedures rely on separation of physics and avoid simultaneous treatment of the different fields [15]. One of the main advantages of partitioning approaches is their ability to use different discretizations, in space and time, for each physics. In the context of reduced order modeling, partitioning approaches were used in conjunction with PGD for calculation of space–time separated representation of multi-field problems [13,40]. An example using different space and time discretizations for each physic can be found in [39]. Monolithic approaches rely on a single system of equations for the entire problem formulation [23]. As an example, Problem 2 is formulated using a space–time monolithic approach. In cases where the different physics exhibit strong interactions, monolithic approaches may be preferable to partitioned procedures in order to ensure unconditional stability and better accuracy [38]. Recently, a reduced order model, based on POD and using monolithic space–time modes, has been successfully applied to resolution of fluid–structure interaction problems [53]. One limitation of monolithic approaches is that if the problem is monolithically formulated in space, then temporal discretization must be the same for all fields. This is highly restrictive when the different fields exhibit different physical kinematics.

In this paper, we present a new multi-field PGD (MF-PGD) that overcomes this limitation. We propose to use the monolithic space–time problem to construct separated representation of both fields, and thus take advantage of the better convergence properties of monolithic approaches. Moreover, due to space–time nature of problem formulation, we potentially allow the use of different approximations in space and time for each field, which overcomes the above-mentioned limitation of monolithic approaches.

4.1. Separated representation

In the proposed multi-field PGD, each field is represented in a separated way as a sum of tensorial products of space and time vectors. That is, space and time vectors differ for each fields.[2] The coupling is automatically enforced through the space–time weak formulation of the evolution problem and thus only depends on it. As in the single field case, we use scalar coefficients (that we also called separation values) to define the decomposition. But in the multi-field case, a separation value is related to the contribution of a multi-field space–time mode. Then, the separated representation of rank m is given by:

$$\begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_{1m} \\ \underline{\underline{\boldsymbol{Y}}}_{2m} \end{bmatrix} = \sum_{i=1}^{m} \alpha_i \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \otimes \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \otimes \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} \tag{56}$$

with $\quad \underline{\boldsymbol{W}}_{1i} \in \mathbb{R}^{n_1^S},\ \underline{\boldsymbol{W}}_{2i} \in \mathbb{R}^{n_2^S},\ \underline{\boldsymbol{\Lambda}}_{1i} \in \mathbb{R}^{n_1^T},\ \underline{\boldsymbol{\Lambda}}_{2i} \in \mathbb{R}^{n_2^T},\ \alpha_i \in \mathbb{R}$

We impose normality of each multi-field space–time mode as:

$$\| \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \otimes \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \otimes \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} \|^2_{[\underline{\underline{\underline{\boldsymbol{N}}}}]} = 1, \quad \forall i = 1, \ldots, m \tag{57}$$

where operator $\|\underline{\underline{\boldsymbol{X}}}\|_{[\underline{\underline{\underline{\boldsymbol{N}}}}]}$ denotes a norm in the vectorial space of third order tensors. It can be written for a two-fields formulation and using tensorial implementation (A.13) as:

$$\|\underline{\underline{\boldsymbol{X}}}\|_{[\underline{\underline{\underline{\boldsymbol{N}}}}]} = \sqrt{\begin{bmatrix} \underline{\underline{\boldsymbol{X}}}_1 \\ \underline{\underline{\boldsymbol{X}}}_2 \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\underline{\boldsymbol{N}}}}_{11} & \underline{\underline{\underline{\boldsymbol{N}}}}_{12} \\ \underline{\underline{\underline{\boldsymbol{N}}}}_{21} & \underline{\underline{\underline{\boldsymbol{N}}}}_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{X}}}_1 \\ \underline{\underline{\boldsymbol{X}}}_2 \end{bmatrix}} \tag{58}$$

4.2. Space–time monolithic definition

The MF-PGD is defined as the solution of a minimization problem, associated with the functional $\mathcal{J} : \begin{bmatrix} \mathbb{R}^{n_1^S \times n_1^T} \\ \mathbb{R}^{n_2^S \times n_2^T} \end{bmatrix} \rightarrow \mathbb{R}^+$.

By analogy with the single field case, we can introduce Galerkin based and minimal residual PGD formulations. However, since Galerkin based PGD do not converge for the elastodynamic problems we have tested, we only focus on minimal residual formulation (see Table 3).

Key point of the proposed approach is to minimize residual of the space–time monolithic problem. Indeed, this is different than breaking the monolothic problem into two separated problems and then minimizing the residual of each separated problem.

4.3. Algorithms

We now introduce multi-field version of PGD algorithms. As a first algorithm, we introduce a progressive construction of MF-PGD. At a given rank m, we suppose decomposition of rank $m-1$ is known. Then we search for the best rank one enrichment that minimizes the functional $\mathcal{J}$. Minimization is alternatively performed, in a monolithical way, with respect to the multi-field space and time vectors. And we additionaly normalized the rank one enrichment. Then, progressive definition of multi-field PGD reads:

Table 3
a priori and equivalent *a posteriori* definitions of functional $\mathcal{J}$ for the minimal residual formulation of the multi-field PGD.

$\mathcal{J}\left(\begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_1^\star \\ \underline{\underline{\boldsymbol{Y}}}_2^\star \end{bmatrix}\right)$	Residual MF-PGD
a priori	$\left\| \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}_1 \\ \underline{\underline{\boldsymbol{L}}}_2 \end{bmatrix} - \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{12} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{21} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_1^\star \\ \underline{\underline{\boldsymbol{Y}}}_2^\star \end{bmatrix} \right\|_2$
a posteriori	$\left\| \begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_1 \\ \underline{\underline{\boldsymbol{Y}}}_2 \end{bmatrix} - \begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_1^\star \\ \underline{\underline{\boldsymbol{Y}}}_2^\star \end{bmatrix} \right\|_{[\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}]' : \cdot [\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}]}$

Definition 3 (*Progressive MF-PGD*). Knowing $\begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_{1(m-1)} \\ \underline{\underline{\boldsymbol{Y}}}_{2(m-1)} \end{bmatrix}$, find α_m, $\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}$ and $\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix}$ such that

$$\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}, \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix}\right) = arg \min_{\substack{\begin{bmatrix} \underline{\boldsymbol{W}}_1^\star \\ \underline{\boldsymbol{W}}_2^\star \end{bmatrix} \in \mathbb{R}^{n_1^S + n_2^S} \\ \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_1^\star \\ \underline{\boldsymbol{\Lambda}}_2^\star \end{bmatrix} \in \mathbb{R}^{n_1^T + n_2^T}}} \mathcal{J}\left(\begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}_{1(m-1)} \\ \underline{\underline{\boldsymbol{Y}}}_{2(m-1)} \end{bmatrix} + \alpha_m \begin{bmatrix} \underline{\boldsymbol{W}}_1^\star \otimes \underline{\boldsymbol{\Lambda}}_1^\star \\ \underline{\boldsymbol{W}}_2^\star \otimes \underline{\boldsymbol{\Lambda}}_2^\star \end{bmatrix}\right) \tag{59a}$$

$$\left\| \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \otimes \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \otimes \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} \right\|^2_{[\underline{\underline{\underline{\boldsymbol{N}}}}]} = 1 \tag{59b}$$

As for the single field case, we can also introduce an optimal algorithm for MF-PGD. This is done by minimizing the functional $\mathcal{J}$ with respect to all space and time vectors, that is:

Definition 4 (*Optimal MF-PGD*). Find $\left\{\alpha_i, \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix}, \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix}\right\}_{i=1,\ldots,m}$ such that

$$\left\{\begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix}, \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix}\right\}_{i=1,\ldots,m}$$

$$= arg \min_{\substack{\begin{bmatrix} \underline{\boldsymbol{W}}_{11}^\star \\ \underline{\boldsymbol{W}}_{21}^\star \end{bmatrix}, \ldots, \begin{bmatrix} \underline{\boldsymbol{W}}_{1m}^\star \\ \underline{\boldsymbol{W}}_{2m}^\star \end{bmatrix} \in \mathbb{R}^{(n_1^S + n_2^S) \times m} \\ \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{11}^\star \\ \underline{\boldsymbol{\Lambda}}_{21}^\star \end{bmatrix}, \ldots, \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m}^\star \\ \underline{\boldsymbol{\Lambda}}_{2m}^\star \end{bmatrix} \in \mathbb{R}^{(n_1^T + n_2^T) \times m}}} \mathcal{J}\left(\sum_{j=1}^{m} \alpha_j \begin{bmatrix} \underline{\boldsymbol{W}}_{1j}^\star \otimes \underline{\boldsymbol{\Lambda}}_{1j}^\star \\ \underline{\boldsymbol{W}}_{2j}^\star \otimes \underline{\boldsymbol{\Lambda}}_{2j}^\star \end{bmatrix}\right) \tag{60a}$$

$$\| \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \otimes \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \otimes \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} \|^2_{[\underline{\underline{\underline{\boldsymbol{N}}}}]} = 1 \quad \forall i = 1, \ldots, m \tag{60b}$$

Implementation details are given in B for both algorithms.

4.4. Convergence and reordering

Thanks to the particular choice we have made for the separated representation (56), we can preserve convergence property of minimal residual PGD in the multi-field case. Indeed, we can easily shown the following convergence property:

[2] This kind of separated representation was also used for space-space decompositions of 2D or 3D structural problems [9,52]. In these cases, the unknown field was the displacement vector $\underline{\boldsymbol{u}}(x, y, z) = [u_x(x, y, z), u_y(x, y, z), u_z(x, y, z)]'$, and each component u_x, u_y, u_z was decomposed in space using tensorial products of independant functions of each coordinate x, y, z.

Property 2 (*Convergence property – Min. residual MF-PGD*). *Let SmallVector$\underline{\boldsymbol{Y}}_{1m}\underline{\boldsymbol{Y}}_{2m}$ be the rank m minimal residual MF-PGD of $\begin{bmatrix}\underline{\boldsymbol{Y}}_1\\ \underline{\boldsymbol{Y}}_2\end{bmatrix}$, constructed in a progressive way. Let*

$$[\underline{\underline{\underline{\boldsymbol{N}}}}] \equiv \begin{bmatrix}\underline{\underline{\underline{\boldsymbol{B}}}}'_{11} & \underline{\underline{\underline{\boldsymbol{B}}}}'_{21}\\ \underline{\underline{\underline{\boldsymbol{B}}}}'_{12} & \underline{\underline{\underline{\boldsymbol{B}}}}'_{22}\end{bmatrix} : \cdot \begin{bmatrix}\underline{\underline{\underline{\boldsymbol{B}}}}_{11} & \underline{\underline{\underline{\boldsymbol{B}}}}_{12}\\ \underline{\underline{\underline{\boldsymbol{B}}}}_{21} & \underline{\underline{\underline{\boldsymbol{B}}}}_{22}\end{bmatrix} \tag{61}$$

be the operator norm used to normalized space–time multi-field modes. Then the multi-field decomposition verifies the following property:

$$\left\| \begin{bmatrix}\underline{\boldsymbol{Y}}_1\\ \underline{\boldsymbol{Y}}_2\end{bmatrix} - \begin{bmatrix}\underline{\boldsymbol{Y}}_{1m}\\ \underline{\boldsymbol{Y}}_{2m}\end{bmatrix} \right\|^2_{[\underline{\underline{\underline{\boldsymbol{B}}}}]' : \cdot [\underline{\underline{\underline{\boldsymbol{B}}}}]} = \left\| \begin{bmatrix}\underline{\boldsymbol{Y}}_1\\ \underline{\boldsymbol{Y}}_2\end{bmatrix} \right\|^2_{[\underline{\underline{\underline{\boldsymbol{B}}}}]' : \cdot [\underline{\underline{\underline{\boldsymbol{B}}}}]} - \sum_{i=1}^{m} \alpha_i^2 \tag{62}$$

Then, separation values still characterise contribution of a given multi-field space–time mode to the whole decomposition. Thus, a calculated decomposition can be reorder at negligible computational cost, by performing a permutation on separation values and corresponding multi-field space–time modes. This is shown in Algorithm 3 of Appendix B.

5. Numerical results

In this section, we illustrate and comment the ability of proper generalized decomposition methods to solve the linear elastodynamic problem with one dimension in space.

5.1. Description of the reference problem

5.1.1. Space and time approximations

In all the study, spatial approximation is done with linear finite elements. The comparison focuses more on the temporal approximations. We compare:

- Two temporal approximations associated with single field formulations, that are the Newmark scheme and the single field time discontinuous Galerkin method in displacement. The space–time problem is given by Exemple 1 for the Newmark scheme and by Exemple 2 for the time discontinuous method. Newmark parameters are set to $\beta = 0.25$ and $\gamma = 0.5$, which gives unconditionnal stability to the scheme. We choose these parameters since they are widely used in industrial codes. However, we have obtained similar results with conditionally stable Newmark schemes. For the time discontinuous method, we use piecewise continuous quadratic finite elements for the approximation in time, and we refer to this scheme as TDG P2.
- Two temporal approximations associated with multi-field formulations, that are the two-fields time continuous and time discontinuous Galerkin methods, formulated in displacement and velocity. Space–time problem is given by Exemple 3 for the time discontinuous method. Its construction is similar for the time continuous method.[3] We use linear finite elements (continuous or piecewise continuous) for the approximation in time. Identical approximations are used for the displacement and velocity fields. We refer to the time continuous Galerkin method as TG P1–P1 and to the time discontinuous Galerkin method as TDG P1–P1.

Classicaly, time integration schemes are evaluated in terms of stability, accuracy, dissipation of spurious high-frequency, starting procedure or computational work [22]. We do not have experience any problem with PGD as regard to stability or starting procedure: we obtain PGD of the unstable solution when the chosen time scheme is unstable and initial conditions are inserted in the right hand side of the space–time problem. Then, in this study, we focus on the influence on PGD of high-frequency perturbations and order of accuracy.

5.1.2. Geometry, material and boundary conditions

The calculations are performed on a beam of size $L = 0.1$ m and section $A = 5.10^{-4}$ m^2. The elasticity constant is taken to $E = 2.10^{11}$ N/m^2 and the density to $\rho = 8000$ kg/m^3. The calculation are done in the transient response until $T = 10^{-4}$ s. We imposed four different boundary conditions, that are:

- a ponctual external load $f^d(t)$ applied at the spatial point $x = 0$, and represented by a unique sinus of period $T/10$ and maximal value $F_{max} = 1.10^5$ N as

$$f^d(t) = \begin{cases} \frac{F_{max}}{2}\left(1 + \sin\left(\frac{2\pi}{T/10}t - \frac{\pi}{2}\right)\right) & \forall t < T/10 \\ 0 & \forall t \geqslant T/10 \end{cases}$$

- an imposed displacement $u^d(t)$ applied at the spatial point $x = 0$, and also represented by a unique sinus of period $T/10$ and maximal value $U_{max} = F_{max}L/EA = 1.10^{-4}$ m as

$$u^d(t) = \begin{cases} \frac{U_{max}}{2}\left(1 + \sin\left(\frac{2\pi}{T/10}t - \frac{\pi}{2}\right)\right) & \forall t < T/10 \\ 0 & \forall t \geqslant T/10 \end{cases}$$

- an initial displacement $u_0(x)$ varying linearly from $-U_{max}$ to 0 and given by:

$$u_0(x) = U_{max}\left(\frac{x}{L} - 1\right)$$

- an initial velocity $v_0(x)$, constant and equals to 10 m/s.

As represented in Fig. 1, all conditions are studied separately and are associated with four test cases. Each test case will be identify by the activated boundary condition. Moreover, for all test cases, displacement and velocity are fixed to zero at the spatial point $x = L$.

5.1.3. Physical homogeneity and multi-field norms

The multi-field formulations used in this study involve the displacement and velocity fields, that are different physical quantities. Then, multi-field norms required in MF-PGDs definitions are not homogeneous in terms of physical units (see for example, definition of the multi-field norm in Eq. (58) applied with the bilinear operators of Exemple 3).

Thus, in order to preserve physical homogeneity of multi-field norms, we propose to preconditionate the multi-field space–time problem and to use a change of variables. The preconditioner is simply chosen to:

$$[\underline{\underline{\underline{\boldsymbol{P}}}}] = \begin{bmatrix} \underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{I}}}_T & \underline{\underline{\boldsymbol{0}}}_S \otimes \underline{\underline{\boldsymbol{0}}}_T \\ \underline{\underline{\boldsymbol{0}}}_S \otimes \underline{\underline{\boldsymbol{0}}}_T & \frac{1}{\theta}\underline{\underline{\boldsymbol{I}}}_S \otimes \underline{\underline{\boldsymbol{I}}}_T \end{bmatrix} \tag{63}$$

and the change of variable is such that

$$\begin{bmatrix}\underline{\underline{\boldsymbol{u}}}\\ \underline{\underline{\boldsymbol{v}}}\end{bmatrix} = [\underline{\underline{\underline{\boldsymbol{P}}}}] : \cdot \begin{bmatrix}\underline{\underline{\boldsymbol{u}}}\\ \underline{\underline{\widetilde{\boldsymbol{v}}}}\end{bmatrix} \tag{64}$$

where the parameter θ is chosen to the time increment Δt.

[3] The two-fields time continuous Galerkin method can be constructed using continuous approximation in time, starting from the space–time weak formulation given by Eq. (38). Then, the fourth and fifth terms in (39) vanish, due to continuity of the temporal approximation. After discretization, the space–time problem is thus given by Exemple 3 with $\underline{\underline{\boldsymbol{\Psi}}}_{uu} = \underline{\boldsymbol{\Psi}}_u(0) \otimes \underline{\boldsymbol{\Psi}}_u(0)$ and $\underline{\underline{\boldsymbol{\Psi}}}_{vv} = \underline{\boldsymbol{\Psi}}_v(0) \otimes \underline{\boldsymbol{\Psi}}_v(0)$.

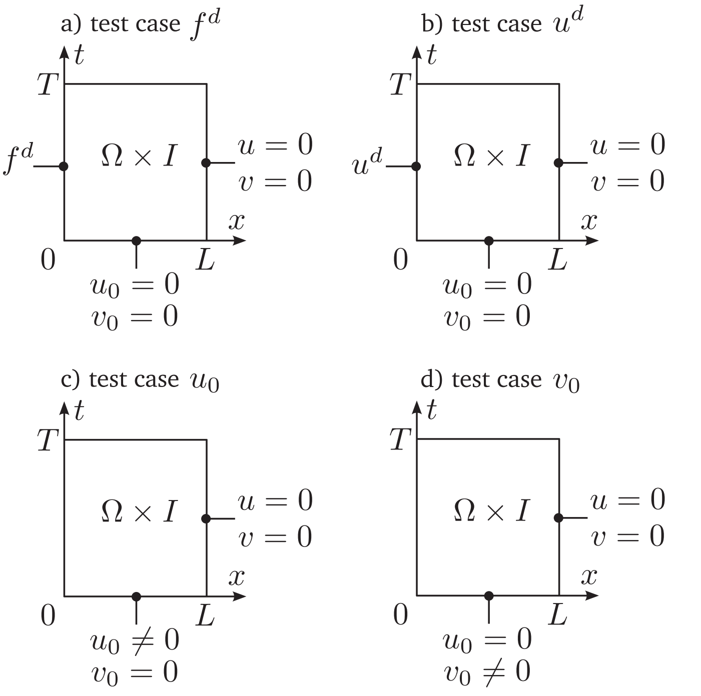

Fig. 1. Space–time geometry and description of boundary conditions for the four test cases: (a) external load, (b) imposed displacement, (c) initial displacement and (d) initial velocity.

5.1.4. Physical units and separated representations

The physical unity of each term of the separated representation depends on PGD definitions and norms used to normalize the space–time modes. Physical unity of each term is summarized in Table 4 when the decomposition is defined with the minimal residual formulation and the space–time modes normalized with different norms. One can notice that when the space–time modes are normalized with the Frobenius norm, the separation values α_i have the same units as the singular values of the displacement matrix $\underline{\underline{\boldsymbol{u}}}$. These constatations hold true in the multi-field cases after physical homogenization of the space–time problem (see the previous subsection).

Table 4
Physical units of each term of the space–time decomposition when it is build with the minimal residual formulation of the PGD and space–time modes are normalized with different operators norm $\underline{\underline{\underline{\boldsymbol{N}}}}$.

$\underline{\underline{\underline{\boldsymbol{N}}}}$	$\underline{\boldsymbol{W}}$	$\underline{\boldsymbol{\Lambda}}$	α
$\underline{\boldsymbol{I}}_S \otimes \underline{\boldsymbol{I}}_T$	[-]	[-]	[m]
$\underline{\underline{\underline{\boldsymbol{B}}}}' : \underline{\underline{\underline{\boldsymbol{B}}}}$	[-]	[m/N]	[N]

5.2. Results and analyses

We perform several numerical analyses with the four time schemes and the four test cases. If it is not specified, the spatial and temporal approximations are fixed to $\Delta x = L/40$ and $\Delta t = T/80$.

5.2.1. Direct space–time solutions

To begin, we directly resolve the linear systems associated to the space–time problems. The obtained solutions are represented in Fig. 2. Spatial discretization has been chosen to $\Delta x = L/40$ and temporal increment to $T/80$ in order to exhibit the different behaviours of temporal approximations. In particular, solution calculated with the Newmark scheme exhibits the well known high frequency perturbations that are not physical, whereas, TDG P2 and TDG P1–P1 schemes allow to attenuate these perturbations. We can also notice that TG P1–P1 scheme behaves better than Newmark one.

5.2.2. Comparison of PGDs definitions and algorithms

We now resolve space–time problems with PGDs, that is we manufacture the space–time solution in a separated way. We calculate it *a priori*, which means that we give a separated representation of the space–time solution without knowing it.

In this subsection, we compare different definitions of single and multi-field PGD, that are the Galerkin based PGDs and minimal residual PGDs. For each definitions, we calculate the decomposition with progressive and optimal algorithms (see B).

Here, our comments are based convergence of residual as decomposition rank increases. For single field cases, the relative residual is post-traited as

$$r^{h,m}_{PGD} = \frac{\left\| \underline{\underline{\boldsymbol{L}}} - \underline{\underline{\underline{\boldsymbol{B}}}} : \underline{\underline{\boldsymbol{Y}}}^h_m \right\|_2}{\left\| \underline{\underline{\boldsymbol{L}}} \right\|_2} \tag{65}$$

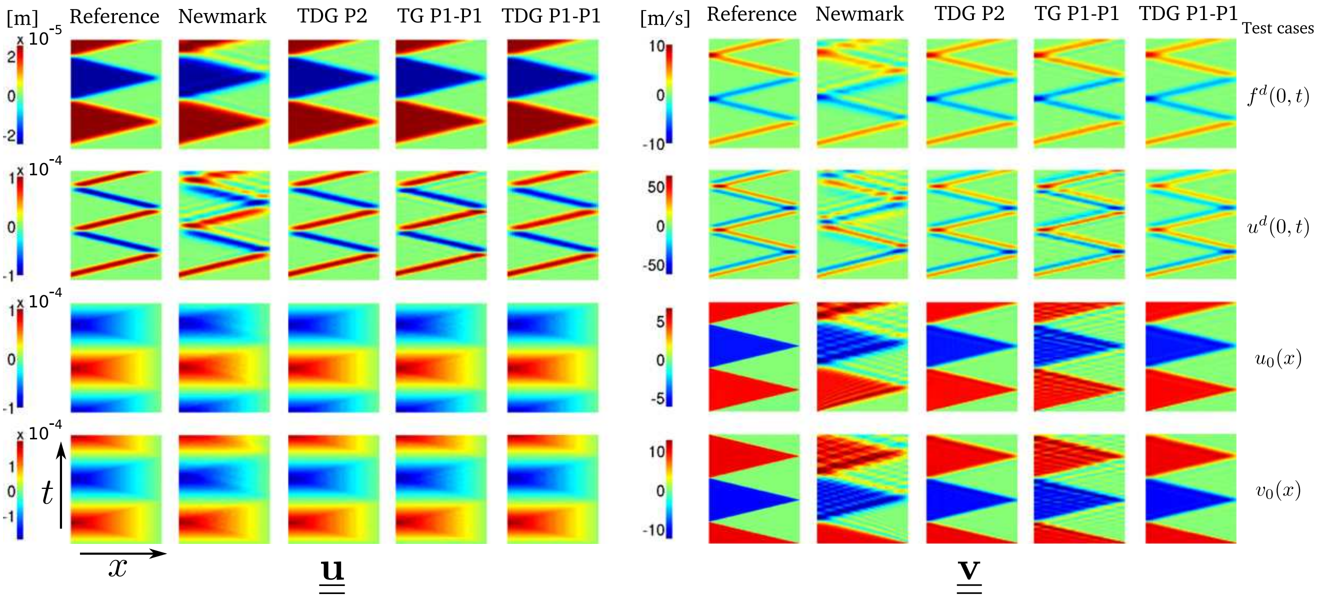

Fig. 2. Solutions obtained by direct resolution of the space–time problems, with $\Delta x = L/40$ and $\Delta t = T/80$. The velocity fields are post-traited thanks to temporal approximations for the Newmark and TDG P2 schemes. The reference solutions are calculated with TDG P2 and a finer space–time mesh.

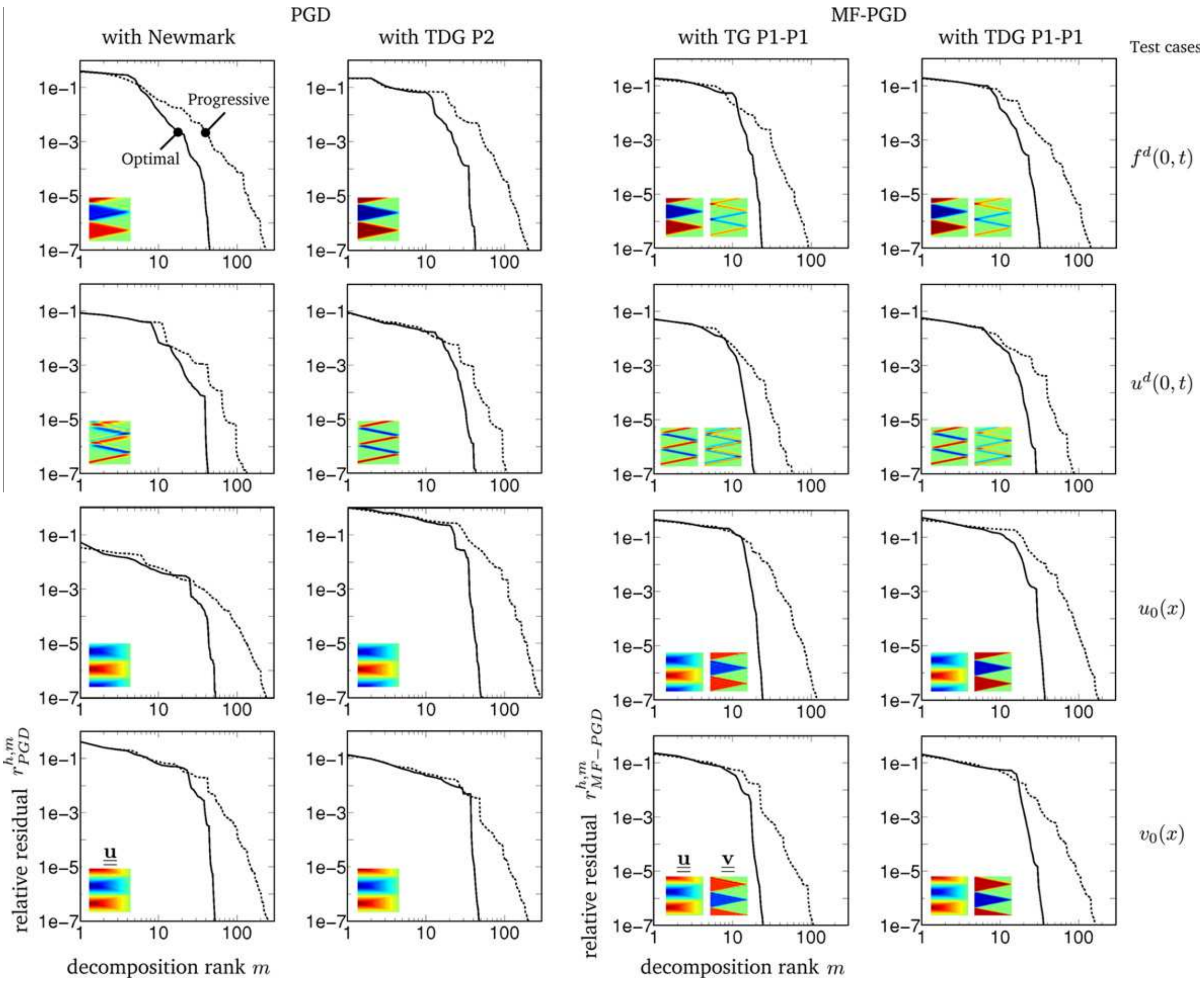

Fig. 3. Convergence in residual of progressive and optimal algorithms for minimal residual definition of PGD and MF-PGD. Recomposed solutions are displayed at the bottom left corner for each cases. Convergence criterium for fixed point loop are chosen to $k_{max} = \infty$ and $\gamma_{stop} = 10^{-6}$.

and for multi-field cases, it is:

$$r^{h,m}_{MF-PGD} = \frac{\left\| \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}_1 \\ \underline{\underline{\boldsymbol{L}}}_2 \end{bmatrix} - \begin{bmatrix} \underline{\underline{\boldsymbol{B}}}_{11}\underline{\underline{\boldsymbol{B}}}_{12} \\ \underline{\underline{\boldsymbol{B}}}_{21}\underline{\underline{\boldsymbol{B}}}_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{Y}}}^h_{1m} \\ \underline{\underline{\boldsymbol{Y}}}^h_{2m} \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}_1 \\ \underline{\underline{\boldsymbol{L}}}_2 \end{bmatrix} \right\|_2} \tag{66}$$

To better understanding, we recall that for single field cases, $\underline{\underline{\boldsymbol{Y}}}^h_m$ corresponds to the rank m decomposition of the displacement field $\underline{\underline{\boldsymbol{u}}}^h$ and for multi-field cases, $\underline{\underline{\boldsymbol{Y}}}^h_{1m}$ and $\underline{\underline{\boldsymbol{Y}}}^h_{2m}$ are respectively associated with the rank m decomposition of the displacement and velocity fields $\underline{\underline{\boldsymbol{u}}}^h$ and $\underline{\underline{\boldsymbol{v}}}^h$. The superscript h denotes solution of the discrete space–time problem.

For all time schemes and for all test cases, we experience a divergence of the residual for Galerkin based definition of the PGD and MF-PGD, calculated with progressive or optimal algorithms. Thus, for the present second order hyperbolic problem, Galerkin based definition of PGD and MF-PGD does not work.

On the contrary, minimal residual definition of PGD and MF-PGD behaves quite better: the relative residual reaches 10^{-7} for all test cases and times schemes (see Fig. 3). Then, we only use minimal residual definition in the following and we implicitely refer to it if not mentioned.

Fig. 3 also validates our proposed strategy for imposition of boundary conditions: we see that there is no need for special procedures in order to take into account Dirichlet kind of conditions, such that imposed displacement over time or initial displacement and velocity.

The residual convergence is similar for all test cases and all time schemes: for low decomposition ranks, the residual decreases linearly, in logarithmic representation, with a slope that depends on the test cases, and thus on the physics of the problem. After a given rank, the residual brutaly falls. The rank associated with the fall depends on the time schemes. Its slope also depends on the time schemes but it is principally related to the decomposition algorithms. Indeed, rel-

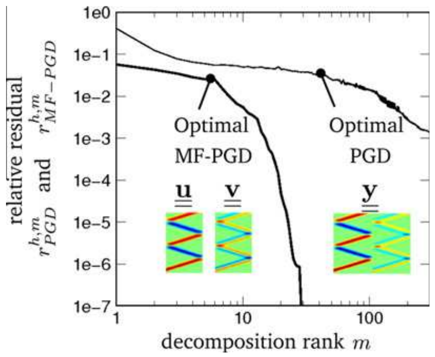

Fig. 4. Comparison of multi-field approach (computed with MF-PGD) and state vector approach (computed with PGD) for TGD P1–P1 scheme. Decompositions are computed with optimal algorithms. Solutions obtained by direct resolution are displayed at the bottom left and right corners.

Table 5
Influence of maximum number of fixed point iterations k_{max}, in progressive algorithms. Decomposition rank is given in function of relative residual r_{PGD} or r_{MF-PGD}. The convergence criterium is set to $\gamma_{stop} = 10^{-6}$.

Relative residual	Progressive PGD								Progressive MF-PGD								Test case
	Newmark				TDG P2				TG P1–P1				TDG P1–P1				
	k_{max}				k_{max}				k_{max}				k_{max}				
	1	10	100	∞	1	10	100	∞	1	10	100	∞	1	10	100	∞	
	Decomposition rank m																
10^{-3}	294	52	45	45	316	67	57	57	164	34	32	32	177	42	36	36	f^d
10^{-5}	>500	152	139	139	>500	135	130	122	314	65	60	60	366	99	84	83	
10^{-7}	>500	270	227	224	>500	249	211	204	471	113	101	102	>500	156	144	144	
10^{-3}	175	46	53	56	181	51	30	30	101	19	17	17	95	31	26	26	u^d
10^{-5}	414	99	83	79	339	75	73	73	211	34	34	34	249	61	60	61	
10^{-7}	>500	128	132	126	>500	136	119	117	301	59	56	57	382	97	92	91	
10^{-3}	163	67	39	38	482	120	104	103	214	43	40	40	247	66	59	59	u_0
10^{-5}	>500	171	133	139	>500	210	179	177	358	84	80	79	454	128	113	114	
10^{-7}	>500	282	236	245	>500	325	272	272	>500	131	111	111	>500	199	174	182	
10^{-3}	301	77	68	68	181	55	50	44	166	31	27	27	153	44	38	38	v_0
10^{-5}	>500	169	149	150	>500	145	116	122	296	95	66	66	357	114	93	93	
10^{-7}	>500	281	242	250	>500	244	205	202	478	131	109	107	>500	198	165	166	

Table 6
Influence of maximum number of fixed point iterations k_{max}, in optimal algorithms. Decomposition rank is given in function of relative residual r_{PGD} or r_{MF-PGD}. The convergence criterium is set to $\gamma_{stop} = 10^{-6}$.

Relative residual	Optimal PGD								Optimal MF-PGD								Test case
	Newmark				TDG P2				TG P1–P1				TDG P1–P1				
	k_{max}				k_{max}				k_{max}				k_{max}				
	1	10	100	∞	1	10	100	∞	1	10	100	∞	1	10	100	∞	
	Decomposition rank m																
10^{-3}	49	23	23	23	61	24	23	23	42	17	16	16	42	20	18	18	f^d
10^{-5}	78	41	40	40	94	39	36	36	54	22	21	21	56	28	28	28	
10^{-7}	89	48	46	46	106	48	43	43	63	27	24	24	63	34	33	33	
10^{-3}	50	23	20	20	60	36	23	23	35	13	13	13	42	15	15	15	u^d
10^{-5}	77	40	40	40	93	36	33	33	41	16	16	16	53	23	23	23	
10^{-7}	100	46	43	43	109	43	44	44	50	20	20	20	63	29	29	29	
10^{-3}	72	47	39	26	87	48	35	36	44	19	18	18	50	25	30	30	u_0
10^{-5}	80	51	45	44	109	51	41	44	56	25	22	22	59	32	33	33	
10^{-7}	105	54	55	54	115	58	47	54	63	27	25	25	69	38	38	38	
10^{-3}	82	42	40	40	77	35	38	38	46	25	18	18	42	24	21	21	v_0
10^{-5}	92	50	47	47	102	41	42	42	54	27	21	21	64	32	31	31	
10^{-7}	100	55	53	53	117	54	52	50	60	31	25	25	80	39	36	36	

ative residual reaches 10^{-7} for a rank which is around one order bigger for progressive algorithm than for optimal one.

5.2.3. Comparison of multi-field and state vector approaches

It is possible to recast a two-field problem into a single field one, using state vector in space and same approximations in time. To this end, the two-field time discontinuous Galerkin method has been recasted as a one-field problem, using state vector in space (see Exemple 4). Then, for this space–time problem, we can compare convergence of decomposition calculated with PGD or MF-PGD algorithms. This is done in Fig. 4 with optimal algorithms and for the imposed displacement test case. Similar results are obtained with progressive implementations and other test cases.

As shown on Fig. 4, we observe that MF-PGD behaves far better than PGD, for the same space–time problem. This can be explain looking at decomposition for both cases. With the multi-field approach (MF-PGD), decompositions of discrete displacement and velocity fields are seeking as

$$\begin{bmatrix} \underline{\underline{\boldsymbol{u}}}_m \\ \underline{\underline{\boldsymbol{v}}}_m \end{bmatrix} = \sum_{i=1}^{m} \alpha_i \begin{bmatrix} \underline{\boldsymbol{W}}_{ui} \otimes \underline{\boldsymbol{\Lambda}}_{ui} \\ \underline{\boldsymbol{W}}_{vi} \otimes \underline{\boldsymbol{\Lambda}}_{vi} \end{bmatrix} \tag{67}$$

and with the state vector approach (PGD), we are manufacturing the decomposition as

$$\underline{\underline{\boldsymbol{y}}}_m = \sum_{i=1}^{m} \alpha_i \begin{bmatrix} \underline{\boldsymbol{W}}_{ui} \\ \underline{\boldsymbol{W}}_{vi} \end{bmatrix} \otimes \underline{\boldsymbol{\Lambda}}_i \quad \text{with} \quad \underline{\underline{\boldsymbol{y}}}_m \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{u}}}_m \\ \underline{\underline{\boldsymbol{v}}}_m \end{bmatrix} \tag{68}$$

Notice that using decomposition (68), we impose to temporal modes to be the same for displacement and velocity fields. Or, $\underline{\underline{\boldsymbol{u}}}_m$ and $\underline{\underline{\boldsymbol{v}}}_m$ are related to continuous displacement and velocity fields by:

$$u(x,t) = \underline{\boldsymbol{\Phi}}(x) \otimes \underline{\boldsymbol{\Psi}}(t) : \underline{\underline{\boldsymbol{u}}}_m \quad \text{and} \quad v(x,t) = \underline{\boldsymbol{\Phi}}(x) \otimes \underline{\boldsymbol{\Psi}}(t) : \underline{\underline{\boldsymbol{v}}}_m$$

and we weakly imposed continuity between $\dot{u}$ and v in the space–time weak formulation. Then, imposing temporal modes to be the same for displacement and velocity fields (as done in (68)) appears

Table 7
Influence of discretization and maximum number of fixed point iterations k_{max}, in MF-PGD optimal algorithm. Decomposition rank is given in function of relative residual r_{MF-PGD}. The convergence criterium is set to $\gamma_{stop} = 10^{-6}$.

Relative residual	TDG P1–P1 and optimal MF-PGD												Test case
	$\Delta x = L/40$ $\Delta t = T/80$				$\Delta x/2$ $\Delta t/2$				$\Delta x/4$ $\Delta t/4$				
	k_{max}				k_{max}				k_{max}				
	1	10	100	∞	1	10	100	∞	1	10	100	∞	
	Decomposition rank m												
10^{-3}	42	15	15	15	42	21	17	17	25	11	10	10	
10^{-5}	53	23	23	23	69	36	42	42	96	51	49	49	u^d
10^{-7}	63	29	29	29	91	46	47	47	140	78	65	64	

Table 8
Influence of convergence criterium γ_{stop}, in optimal algorithms. Decomposition rank is given in function of relative residual r_{PGD} or r_{MF-PGD}. Fixed point process iterates until reaching convergence, that is $k_{max} = \infty$.

Relative residual	Optimal PGD								Optimal MF-PGD								Test case
	Newmark				TDG P2				TG P1–P1				TDG P1–P1				
	$\log_{10}(\gamma_{stop})$				$\log_{10}(\gamma_{stop})$				$\log_{10}(\gamma_{stop})$				$\log_{10}(\gamma_{stop})$				
	−1	−2	−3	−6	−1	−2	−3	−6	−1	−2	−3	−6	−1	−2	−3	−6	
	Decomposition rank m																
10^{-3}	26	21	23	23	28	24	23	23	18	16	16	16	24	19	18	18	
10^{-5}	45	44	40	40	40	34	36	36	23	21	21	21	33	29	28	28	f^d
10^{-7}	50	50	46	46	51	42	44	43	27	25	24	24	41	34	33	33	
10^{-3}	25	20	20	20	40	24	32	23	13	13	13	13	15	15	15	15	
10^{-5}	44	40	40	40	40	34	33	33	18	16	16	16	24	24	23	23	u^d
10^{-7}	53	45	43	43	52	43	40	44	24	20	20	20	31	30	29	29	
10^{-3}	23	30	22	26	50	43	35	36	23	20	21	18	26	29	30	30	
10^{-5}	45	44	47	44	63	49	41	44	26	25	25	22	35	33	33	33	u_0
10^{-7}	53	56	51	54	69	54	49	54	30	28	28	25	39	38	38	38	
10^{-3}	41	47	40	40	44	37	40	38	24	21	18	18	24	21	21	21	
10^{-5}	51	49	47	47	52	44	42	42	26	25	21	21	32	30	31	31	v_0
10^{-7}	59	55	53	53	63	51	50	50	31	28	25	25	41	37	36	36	

to be constraining for the construction of their decompositions. Indeed, we do not impose this condition using the multi-field approach (since space and time modes are different for each field) and observe better convergence of the decomposition.

5.2.4. Influence of algorithms parameters

The PGD and MF-PGD are defined as solutions of minimization problems. Minimization is alternatively perform as regard to spatial and temporal modes. Stationnary conditions associated with these minimization problems leads to the definition of two problems, one in space and an other in time, that are non-linear and coupled. Then, a fixed point strategy is classicaly used to solve this problem. It consists of alternatively solving the problem in space and in time, until some convergence criterium is reached (see B).

Thus progressive and optimal PGDs algorithms depend on some numerical parameters that pilot convergence of this fixed point loop, performed each times a new space–time mode is calculated. These parameters are:

- k_{max} the maximum number of fixed point iterations,
- and γ_{stop} the convergence criterium.

For PGD, convergence of the fixed point problem is reached as soon as $\|\underline{\boldsymbol{W}}_m^{(k)} - \underline{\boldsymbol{W}}_m^{(k-1)}\|_2$ is smaller than γ_{stop}, where k and $k-1$ denote two successive fixed point iterates and m is the current decomposition rank. For MF-PGD, the criterium is applied to $\underline{\boldsymbol{W}}_{1m}$ and $\underline{\boldsymbol{W}}_{2m}$. We recall that space vectors are normalized in Frobenius norm, before verification of fixed point convergence.

In Tables 5 and 6, we can observe the number of space–time modes necessary to reach a given relative residual gets smaller when k_{max} (the maximum number of fixed point iterations) increases. For all test cases, for all time schemes and for both PGD and MF-PGD algorithms, decomposition calculated with a miximum of ten fixed point iterations ($k_{max} = 10$) is closed to the fully converged decomposition (with $k_{max} = \infty$ and $\gamma_{stop} = 10^{-6}$). This is still true when we refine discretization (see Table 7). The same orders of fixed point iterations ($k_{max} = O(10)$) is necessary to get the fully converged decomposition. Moreover, convergence criterium γ_{stop} has a small influence on decomposition. Indeed, as shown in Table 8, with a small value of γ_{stop}, say 10^{-1}, the fully converged decomposition is closed to the one obtained with $\gamma_{stop} = 10^{-6}$.

Thus, one can conclude that PGD and MF-PGD algorithms exhibit small dependency as regards to fixed point parameters. A relatively small number of fixed point iterations is needed and very small convergence criterium can be used for computation of each new space–time mode.

In the following, all decompositions are calculated with parameters $k_{max} = 50$ and $\gamma_{stop} = 10^{-2}$.

5.2.5. Influence of normalization of space–time modes and reordering

In all previous manufactured decompositions, space–time modes were normalized in norms defined with bilinear operators of the physical problems. For minimal residual formulations, these operators are the symmetrized ones and write

$$\underline{\underline{\underline{\boldsymbol{N}}}} = \underline{\underline{\underline{\boldsymbol{B}}}}' : \underline{\underline{\underline{\boldsymbol{B}}}} \quad \text{for PGD}$$

$$\text{and} \quad [\underline{\underline{\boldsymbol{N}}}] \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{B}}}'_{11} & \underline{\underline{\boldsymbol{B}}}'_{21} \\ \underline{\underline{\boldsymbol{B}}}'_{12} & \underline{\underline{\boldsymbol{B}}}'_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{B}}}_{11} & \underline{\underline{\boldsymbol{B}}}_{12} \\ \underline{\underline{\boldsymbol{B}}}_{21} & \underline{\underline{\boldsymbol{B}}}_{22} \end{bmatrix} \quad \text{for MF-PGD.}$$

In this subsection, we normalize space–time modes in Frobenius norms, that are defined with

$$\underline{\underline{\boldsymbol{N}}} = \underline{\boldsymbol{I}}_S \otimes \underline{\boldsymbol{I}}_T \quad \text{for PGD}$$

$$\text{and} \quad [\underline{\underline{\boldsymbol{N}}}] \equiv \begin{bmatrix} \underline{\boldsymbol{I}}_S \otimes \underline{\boldsymbol{I}}_T & \underline{\underline{\boldsymbol{0}}} \\ \underline{\underline{\boldsymbol{0}}} & \underline{\boldsymbol{I}}_S \otimes \underline{\boldsymbol{I}}_T \end{bmatrix} \quad \text{for MF-PGD.}$$

We observe no significant influence of space–time modes normalization on decomposition convergence in relative residual.

Also, we compare convergence of decompositions obtained with and without reordering. Reordering is performed in Frobenius norms (see Algorithm 3 of Appendix B).

Here, our comments are based on convergence of relative error (calculated in Frobenius norm), between separated and non-separated solutions, as decomposition rank increases. Relative error indicators read:

$$\varepsilon^{h,m}_{ROM}(\underline{\underline{\boldsymbol{Y}}}) = \frac{\|\underline{\underline{\boldsymbol{Y}}}^h - \underline{\underline{\boldsymbol{Y}}}^h_m\|_2}{\|\underline{\underline{\boldsymbol{Y}}}^h\|_2} \tag{69}$$

with ROM $=$ PGD, MF-PGD or SVD

where $\underline{\underline{\boldsymbol{Y}}}^h$ denotes the discret space–time problem solution and $\underline{\underline{\boldsymbol{Y}}}^h_m$ its decomposition.

Without reordering, convergence of decomposition in relative errors is characterized by a succession of stagnation steps and abrupt falls (see Figs. 5 and 6). It exhibits the fact that contribuant space–time modes (in Frobenius norm) are calculated after less contribuant modes. Indeed, looking at separation values associated with each space–time mode, we observe they are not sorted in decreasing order (see Fig. 7).

Then, we can reordonnate decompositions in function of separation values. By this way, we can gather space–time modes associated with convergence falls, together at the beginning of the decomposition, and shift modes associated with stagnation steps at the end of the decomposition (see Figs. 5 and 6).

We also compare convergence of PGD and MF-PGD with SVD of the discrete space–time solutions (see Figs. 5 and Fig. 6). SVD gives the optimal decomposition if we evaluate convergence of error in Frobenius norm. We observe that PGD calculated with optimal algorithms give a good approximation of SVD: the first separation values are of the same order of magnitude as singular values and similar space–time modes are found in the decomposition (see Fig. 7). However, PGD and MF-PGD do not give space–time solutions to machine precision when decomposition rank equals the smallest problem dimension size, that is when $m = n_S$, as do SVD (see relative error obtained for the last decomposition rank in Figs. 5 and 6).

Finally, decompositions calculated with TG P1–P1 and TDG P1–P1 approximations converge more quickly (in error norm) than decomposition calculated with Newmark and TDG P2 approximations, for all test cases.

5.2.6. Influence of discretization

We recall our principal objective is to compute the exact solution of an elastodynamic problem. Previously shown solutions contained different kinds of errors. In this subsection, we study influence of discretization on various error types.

Suppose we know the exact analytical solution of the considered elastodynamic problem. We can express it on a space–time mesh, and thus, denote it by $\underline{\underline{\boldsymbol{Y}}}$. This is the solution we track. For

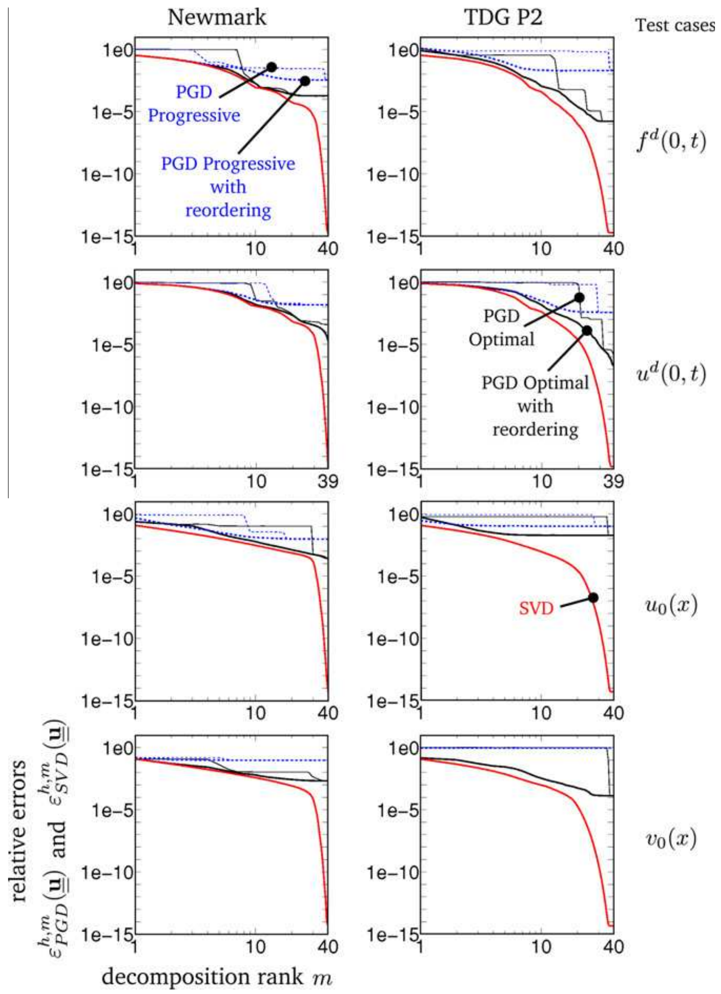

Fig. 5. Influence of reordering in Frobenius norm for progressive and optimal PGDs, and comparison with SVD of the discrete displacement field.

the displacement test case, this solution can be written as finite sum of analytical functions. Starting from the strong problem formulation (see Eqs. (5)) and using Laplace transform, we easily show that the exact analytical solution is given by:

$$u_{exact}(x,t) = \sum_{k=0}^{n} \left(u^d\left(t - \frac{x + 2Lk}{c}\right) - u^d\left(t + \frac{x - 2L(k+1)}{c}\right)\right) \tag{70}$$

where $c = \sqrt{E/\rho}$ is the waves velocity and n is the lower integer part of $(cT - x)/(2L)$.

Considerate now a weak formulation (in space or space–time). In order to compute a solution, we use approximations in space and time. These approximations are a first source of error that we call discretization error. We use the following error indicator to characterized it:

$$\varepsilon^{h}_{DISC}(\underline{\underline{\boldsymbol{Y}}}) = \frac{\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}^h\|_2}{\|\underline{\underline{\boldsymbol{Y}}}\|_2} \tag{71}$$

where $\underline{\underline{\boldsymbol{Y}}}^h$ denotes the discrete solution of the space–time problem and h is related to discretization parameters in space and time, that is $h = (\Delta x, \Delta t)$. Convergence order of discretization error is related to the chosen approximation spaces. Since we use piecewise linear finite elements in space, we have a quadratic rate of convergence of the spatial discretization. Concerning temporal discretization, Newmark (with $\beta = 0.25,\ \gamma = 0.5$) and TG P1–P1 schemes give quadratic rate of convergence whereas TDG P2 and TDG P1–P1 schemes are third order accurate (see [24,41,26]). However, in this study we evaluate space–time rate of convergence, that is $(\Delta x, \Delta t)$

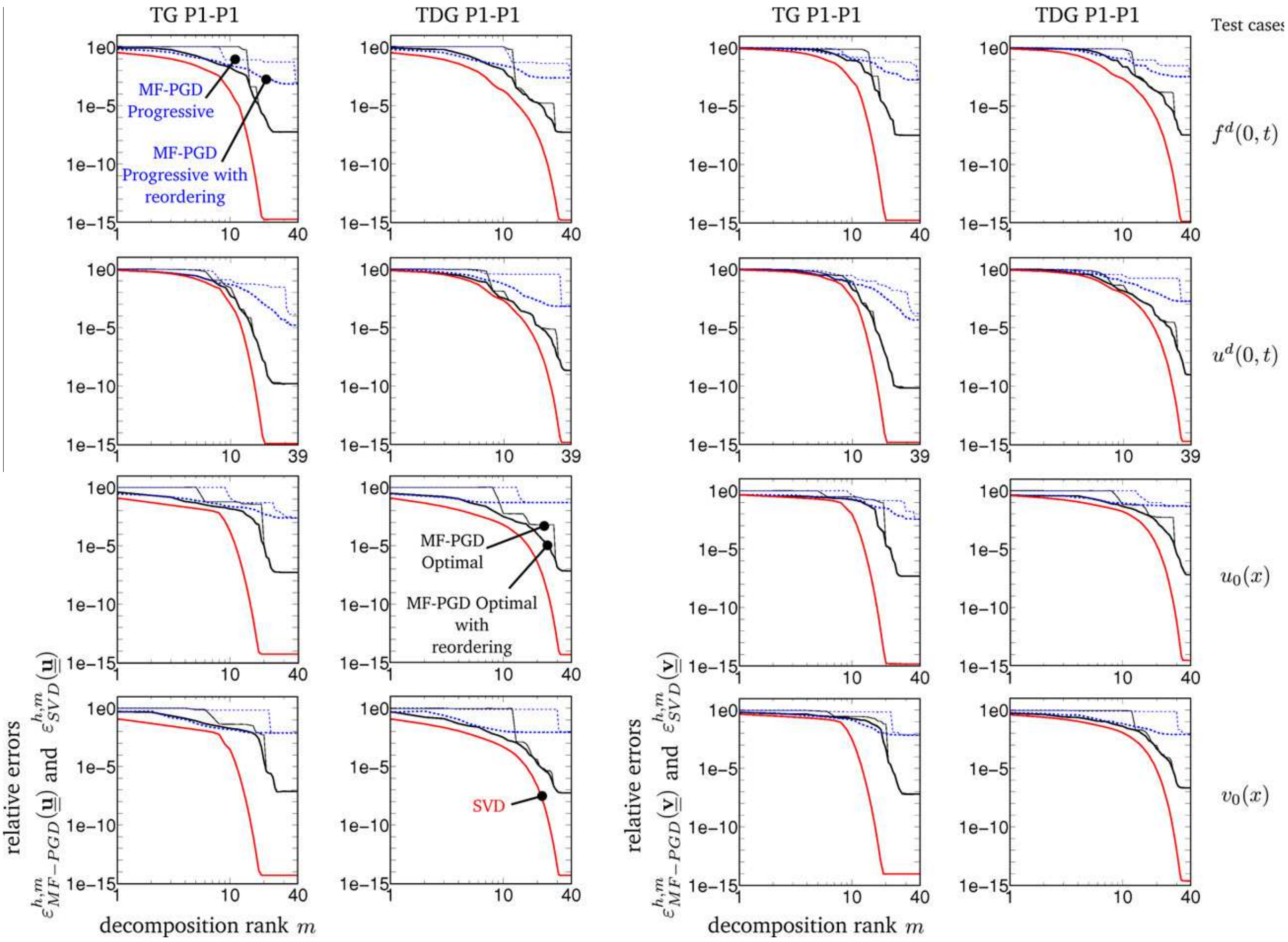

Fig. 6. Influence of reordering in Frobenius norm for progressive and optimal MF-PGDs, and comparison with SVD of the discrete displacement and velocity fields.

are decrease at the same time. Then, space–time accuracy order is a combinaison of accuracy orders in space and time.

In this paper, we use PGD methods in order to solve the space–time problem at low computational costs. The obtained solution is given in a separated representation. It is expressed, after spatial and temporal approximations, as a linear combinaison of m tensorial products of space and time vectors. We denote this solution by $\underline{\underline{\boldsymbol{Y}}}^h_m$. It would be exactly the discrete space–time solution if $m=\infty$. Here, due to the fact we use a finite number of tensorial products, an other error is committed. We call it decomposition error and we evaluate it as:

$$\varepsilon^{h,m}_{ROM}(\underline{\underline{\boldsymbol{Y}}}) = \frac{\|\underline{\underline{\boldsymbol{Y}}}^h - \underline{\underline{\boldsymbol{Y}}}^h_m\|_2}{\|\underline{\underline{\boldsymbol{Y}}}^h\|_2} \tag{72}$$

with ROM = PGD, MF-PGD

Then, if we look at the total error committed with the PGD methods, we should consider both discretization and decomposition errors. We characterize the total error with the following indicator:

$$\varepsilon^{h,m}_{TOT}(\underline{\underline{\boldsymbol{Y}}}) = \frac{\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}^h_m\|_2}{\|\underline{\underline{\boldsymbol{Y}}}\|_2} \tag{73}$$

Finally, it is interesting to compute SVD of the exact analytical solution. This decomposition is denoted by $\underline{\underline{\boldsymbol{Y}}}_m$ and is, to our sens, the best space–time decomposition we could obtained. Then, difference between $\underline{\underline{\boldsymbol{Y}}}$ and $\underline{\underline{\boldsymbol{Y}}}_m$ exhibits error due to space–time decomposition. We call this error the exact error and express it thanks to the following error indicator:

$$\varepsilon^{m}_{EXACT}(\underline{\underline{\boldsymbol{Y}}}) = \frac{\|\underline{\underline{\boldsymbol{Y}}} - \underline{\underline{\boldsymbol{Y}}}_m\|_2}{\|\underline{\underline{\boldsymbol{Y}}}\|_2} \tag{74}$$

Links between all mentioned error indicators are illustrated on Fig. 8.

In this subsection, we compute all error indicators with displacement fields $\underline{\underline{\boldsymbol{u}}}, \underline{\underline{\boldsymbol{u}}}^h, \underline{\underline{\boldsymbol{u}}}^h_m$ and $\underline{\underline{\boldsymbol{u}}}_m$. The obtained convergence curves are displayed in function of decomposition rank m on Fig. 9, for different discretizations.

Here, we clearly exhibit importance of the temporal approximations for the space–time proper generalized decomposition of the elastodynamic problem. Indeed, the total error committed with PGD methods will never be lower than the discretization error[4] (compare blue curves and discontinuous magenta curves in Fig. 9). Then, it is unnecessary to try to compute a PGD with a bad time integration scheme. And a PGD should be performed with the best available temporal approximations. As an example, one can compare discretization errors obtained with Newmark scheme, TDG P2, TG P1–P1 and TDG P1–P1 methods on Fig. 9: looking at blue curves, we see that Newmark scheme gives a discretization error one order bigger than other methods, at fixed discretization.

We can also notice that, whatever the temporal approximation is, all decompositions tend to the SVD of the analytical solution as

[4] In fact, we observe on Fig. 9 that the total error can be a little smaller than discretization error. This is due to the fact that high frequency perturbations (that are non physical and due to temporal approximations) are shifted at the end of the decomposition.

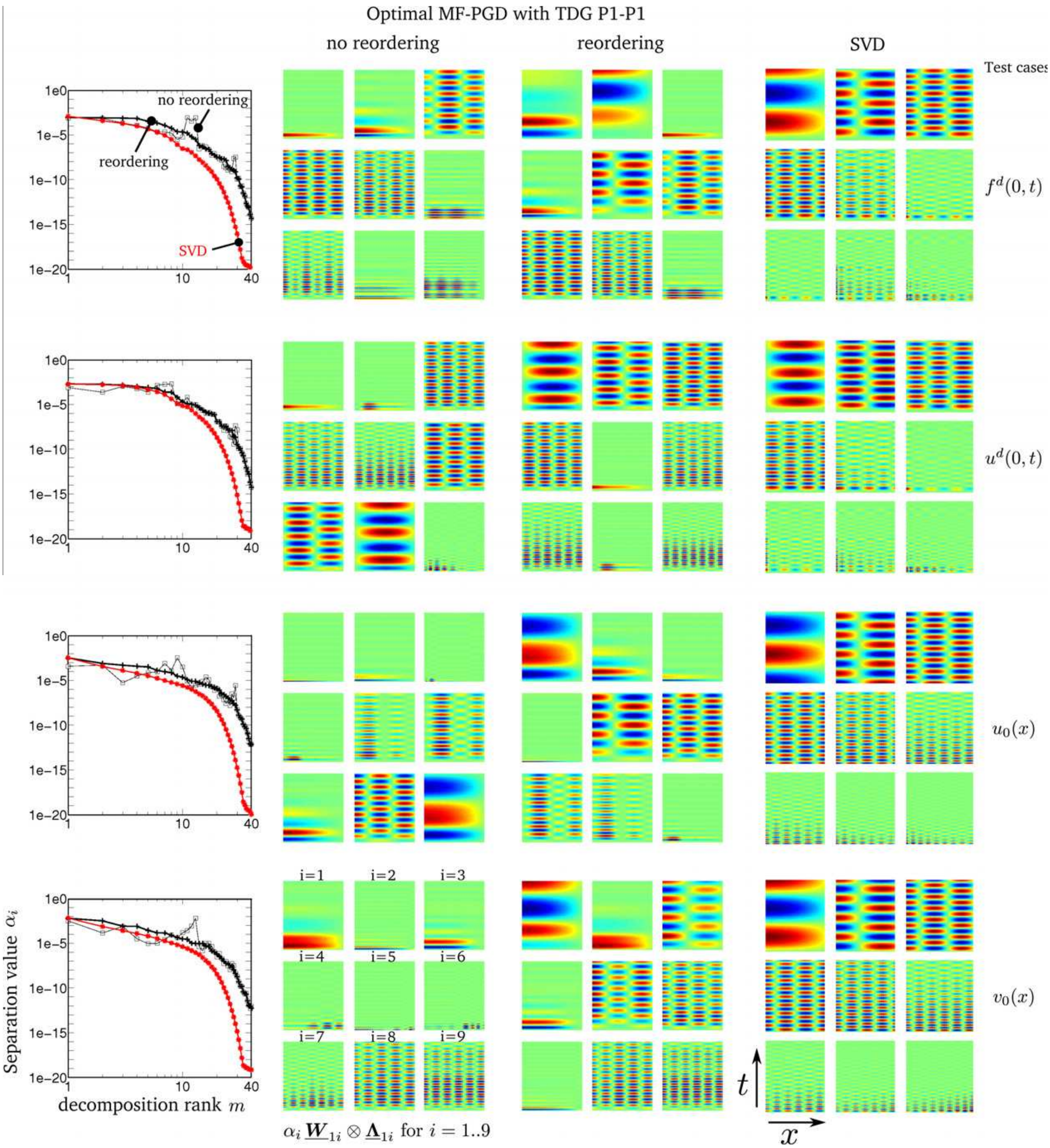

Fig. 7. First ninth space–time modes of the discrete displacement field, for TDG P1–P1 scheme, calculated with optimal MF-PGD without and with reordering in Frobenius norm, and SVD.

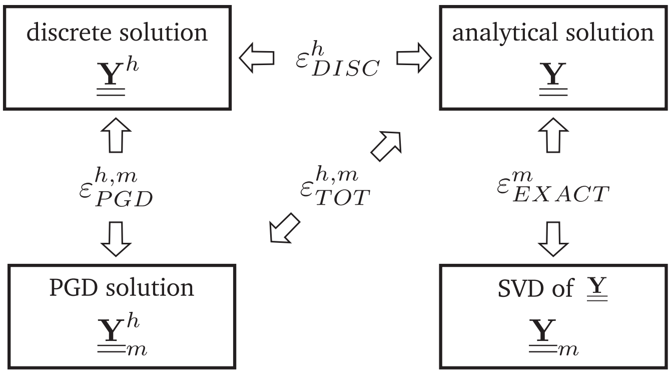
discrete solution
$\underline{\underline{\mathbf{Y}}}^h$
ε^h_{DISC}
analytical solution
$\underline{\underline{\mathbf{Y}}}$
$\varepsilon^{h,m}_{PGD}$
$\varepsilon^{h,m}_{TOT}$
ε^m_{EXACT}
PGD solution
$\underline{\underline{\mathbf{Y}}}^h_m$
SVD of $\underline{\underline{\mathbf{Y}}}$
$\underline{\underline{\mathbf{Y}}}_m$

Fig. 8. Illustration of error indicators.

discretization gets finer (compare red curve and all discontinuous magenta curves on Fig. 9).

Finally, we see an harmful effect of discretization on the convergence of PGDs (or MF-PGDs). If we look at the decomposition errors on Fig. 9 (black curves), we observe that the step at the end of the decomposition (and associated with non-contribuant space–time modes), increases as the discretization gets finer. In this study, all calculations were performed until the relative residual reaches 10^{-7} (see Fig. 10). So, this means that finding the optimal decomposition by minimizing the residual does not give the optimal decomposition that minimizes the error mesured in the Frobenius norm. And the distance between these two decompositions gets bigger as discretization gets finer.

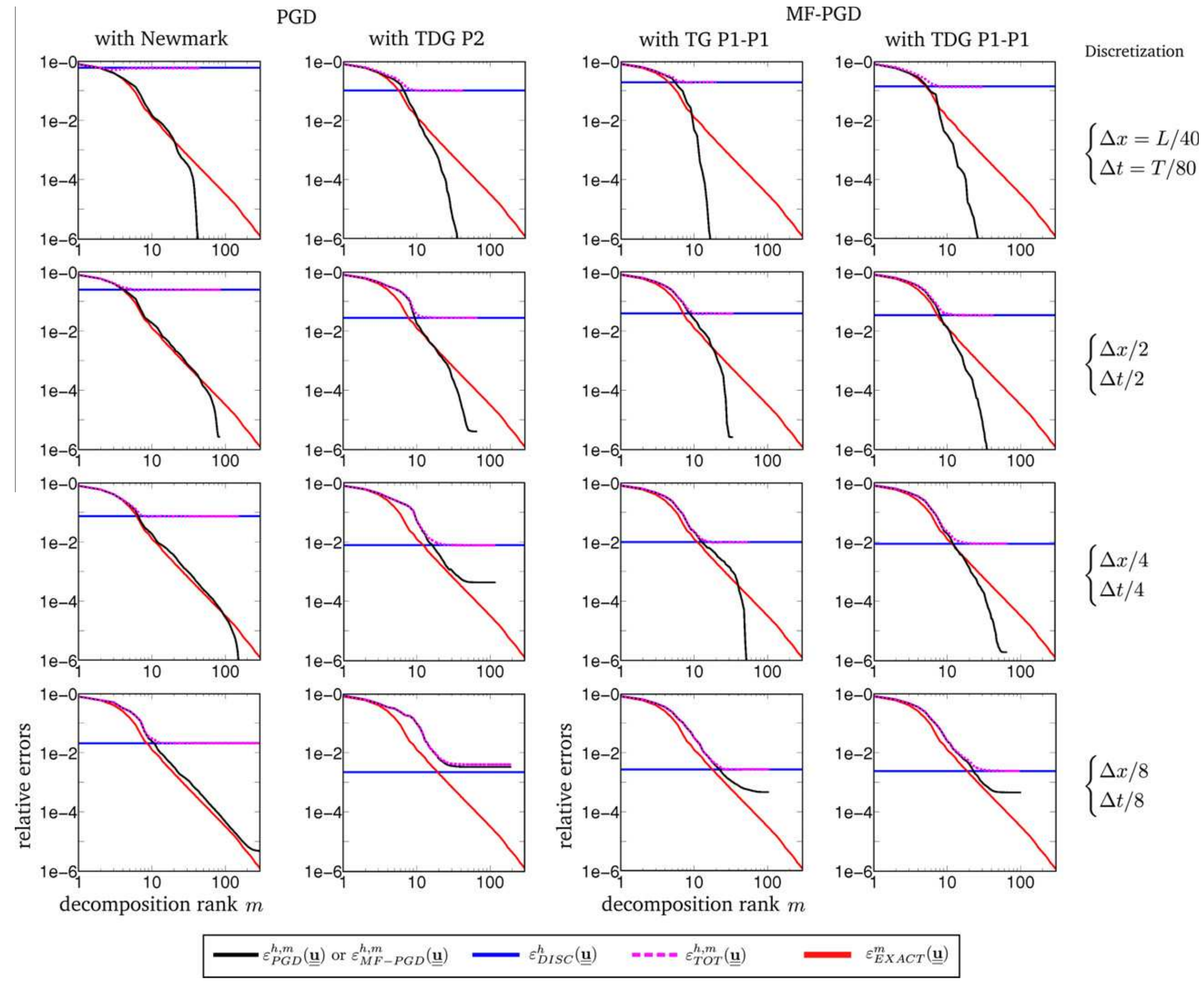

Fig. 9. Influence of discretization on the decomposition convergence, in relative errors. Decompositions are calculated with optimal algorithms. Calculations are stopped when relative residual reaches 10^{-7} (see Fig. 10). A reordering were performed in Frobenius norm. Test case is u^d.

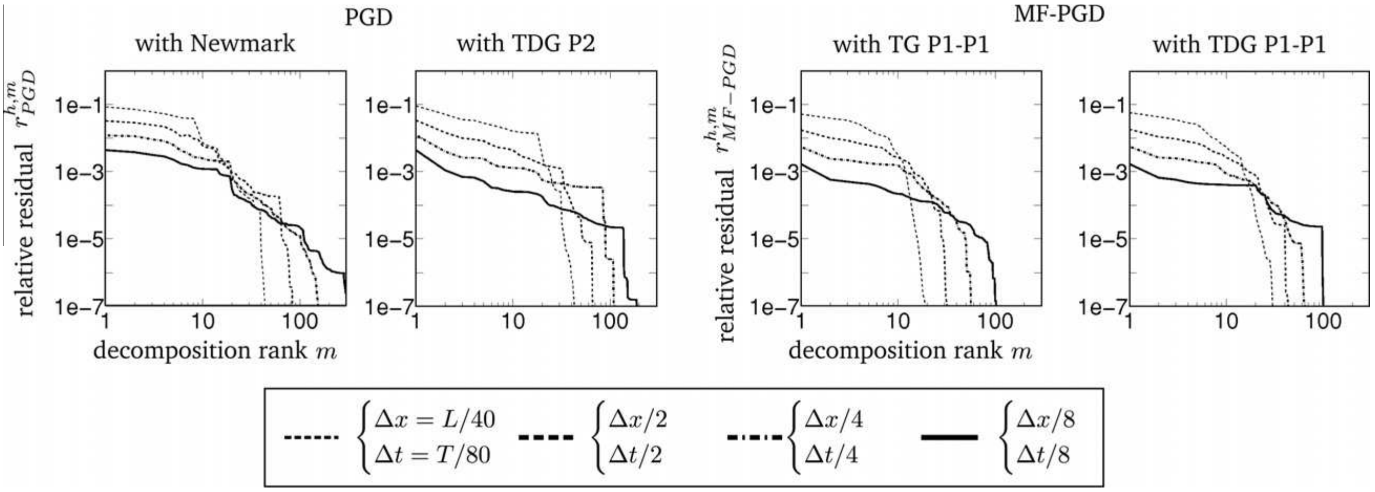

Fig. 10. Relative residuals associated with decompositions shown in Fig. 9. Decompositions are calculated with optimal algorithms and no reordering were performed. Calculations are stopped when relative residual reaches 10^{-7}. Test case is u^d.

Then, it exhibits two facts: first, PGD methods should be used in conjuction with efficient error estimators, as done in [6,33], and second we need an efficient algorithm for PGD calculation that can break dependency of the proposed algorithms to discretization.

6. Conclusions and perspectives

We propose a general strategy, based on tensorial formalism, for the resolution of second order hyperbolic equations with PGD methods. In particular, we show there is no need for special procedure in order to imposed Neumann or Dirichlet conditions (and notably, initial conditions in displacement and velocity), since they can be directly introduced in the right hand side of the space–time problem.

In order to use recent time integration schemes, we introduce a new PGD method for the resolution of multi-field problems. This multi-field PGD (MF-PGD) takes advantage of monolithic approaches while allowing the use of different approximations in space and time for each field. It converges far better than PGD applied to the multi-field problem recasted as a single field one using state vector in space.

Numerical results hightlight importance of temporal approximations for PGD of elastodynamic problems. It appears to be unnecessary to try to calculate a PGD with inaccurate time schemes. Indeed, PGD methods commit errors due to discretization and decomposition. Then, even if the decomposition error is very small, the total error cannot be lower than the discretization one. This is reinforced by the fact we observe an harmful effect of discretization on the convergence of PGDs to the optimal decomposition given by SVD.

Different definitions of PGD (and MF-PGD) are investigated. Galerkin based PGDs appears to not converge for the second order hyperbolic problem we have treated. On the contrary, minimal residual PGD converges for all temporal approximations we have compared and all test cases. Also, we evaluate convergence of progressive and optimal implementations of PGD (and MF-PGD). Our optimal implementation *a priori* gives a decomposition closed to SVD of space–time solutions. But this algorithm is still computationally not optimal.

Then, future works will focus on definition of PGD algorithms that can track optimal decomposition with a reasonable algorithmic complexity. Krylov subspace type solvers have been proposed in [44] but they are applicable only when space–time bilinear operators have a rank one decomposition. Finally, one can probably takes advantage of multi-scale methods in order to break discretization dependency of PGD algorithms for hyperbolic type equations.

Acknowledgement

This work was supported by the French National Research Agency under grant SIMDREAM-2010.

Appendix A. Multilinear algebra definitions

Since the strategy proposed in this paper for identification of space–time operators is strongly related to tensorial formalism, we give here some details about multilinear algebra notations and definitions we use.

We adopt the underline convention to denote tensors. That is the number of underlines relates to the tensor order. For example, $\underline{\boldsymbol{A}}$ denotes a first order tensor, $\underline{\underline{\boldsymbol{A}}}$ a second order tensor, *et cetera*. Tensorial products are denoted with "$\otimes$" and dot convention is used for contracted product. The following definitions are extracted from classical multilinear algebra textbooks, and we refer to [37,28] for more details.

We consider fourth-order tensor as linear mapping between second-order tensor:

$$(\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}}) : \underline{\underline{\boldsymbol{C}}} = \underline{\underline{\boldsymbol{D}}} \Longleftrightarrow \sum_{j,l} A_{ij} B_{kl} C_{jl} = D_{ik} \tag{A.1}$$

$$\underline{\underline{\boldsymbol{C}}} : (\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}}) = \underline{\underline{\boldsymbol{D}}} \Longleftrightarrow \sum_{i,k} C_{ik} A_{ij} B_{kl} = D_{jl} \tag{A.2}$$

The two times contracted product ":" between second or fourth order tensors is defined as:

$$\underline{\underline{\boldsymbol{A}}} : \underline{\underline{\boldsymbol{B}}} = \sum_{i,j} A_{ij} B_{ij} \tag{A.3}$$

$$(\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}}) : (\underline{\underline{\boldsymbol{C}}} \otimes \underline{\underline{\boldsymbol{D}}}) = (\underline{\underline{\boldsymbol{A}}} \cdot \underline{\underline{\boldsymbol{C}}}) \otimes (\underline{\underline{\boldsymbol{B}}} \cdot \underline{\underline{\boldsymbol{D}}}) \tag{A.4}$$

We also use transpose and inverse operations for fourth order tensors:

$$(\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}})' = \underline{\underline{\boldsymbol{A}}}' \otimes \underline{\underline{\boldsymbol{B}}}' \tag{A.5}$$

$$(\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}})^{-1} = \underline{\underline{\boldsymbol{A}}}^{-1} \otimes \underline{\underline{\boldsymbol{B}}}^{-1} \tag{A.6}$$

Implementation of mappings (A.1) and (A.2) can be done using matrix–vector algebra. To this end, we denote by $[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}]$ a matrix and $[\begin{smallmatrix} a \\ b \end{smallmatrix}]$ a vector; one can use the following matrix implementation for fourth order tensor, and vector implementation for second order tensor:

$$\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}} \equiv \begin{bmatrix} B_{11}\underline{\underline{\boldsymbol{A}}} & \cdots & B_{1n}\underline{\underline{\boldsymbol{A}}} \\ \vdots & \ddots & \vdots \\ B_{m1}\underline{\underline{\boldsymbol{A}}} & \cdots & B_{mn}\underline{\underline{\boldsymbol{A}}} \end{bmatrix} \quad \text{and} \quad \underline{\boldsymbol{a}} \otimes \underline{\boldsymbol{b}} \equiv \begin{bmatrix} b_1\underline{\boldsymbol{a}} \\ \vdots \\ b_m\underline{\boldsymbol{a}} \end{bmatrix} \tag{A.7}$$

This implementation slightly differs from the classical one using Kronecker product [51]. But it is more appropriate in order to construct space–time decomposition of elastodynamic problem formulated using spatial weak form and integration schemes based on finite difference formulas. For example, mapping (A.1) can be equivalently recasted as a linear system of equations as:

$$(\underline{\underline{\boldsymbol{A}}} \otimes \underline{\underline{\boldsymbol{B}}}) : (\underline{\boldsymbol{a}} \otimes \underline{\boldsymbol{b}}) = \underline{\boldsymbol{c}} \otimes \underline{\boldsymbol{d}} \Longleftrightarrow \begin{cases} (B_{11}\underline{\underline{\boldsymbol{A}}}) \cdot (b_1\underline{\boldsymbol{a}}) + \cdots + (B_{1n}\underline{\underline{\boldsymbol{A}}}) \cdot (b_n\underline{\boldsymbol{a}}) = d_1\underline{\boldsymbol{c}} \\ \vdots \\ (B_{m1}\underline{\underline{\boldsymbol{A}}}) \cdot (b_1\underline{\boldsymbol{a}}) + \cdots + (B_{mn}\underline{\underline{\boldsymbol{A}}}) \cdot (b_n\underline{\boldsymbol{a}}) = d_m\underline{\boldsymbol{c}} \end{cases} \tag{A.8}$$

Finally, we define sixth-order tensor as linear mapping between third-order tensor. This enables the use of fourth-order and second-order tensors in conjunction with matrix–vector algebra. We thus introduce the three times contracted product ": ." and define the left and right mappings as:

$$(\underline{\underline{\underline{\underline{\boldsymbol{A}}}}} \otimes \underline{\underline{\boldsymbol{B}}}) : \cdot \underline{\underline{\underline{\boldsymbol{C}}}} = \underline{\underline{\underline{\boldsymbol{D}}}} \Longleftrightarrow \sum_{j,l,n} A_{ijkl} B_{mn} C_{jln} = D_{ikm} \tag{A.9}$$

$$\underline{\underline{\underline{\boldsymbol{C}}}} : \cdot (\underline{\underline{\underline{\underline{\boldsymbol{A}}}}} \otimes \underline{\underline{\boldsymbol{B}}}) = \underline{\underline{\underline{\boldsymbol{D}}}} \Longleftrightarrow \sum_{i,k,m} C_{ikm} A_{ijkl} B_{mn} = D_{jln} \tag{A.10}$$

Definitions for the three times contracted product between third or sixth order tensors are given by:

$$\underline{\underline{\underline{\boldsymbol{A}}}} : \cdot \underline{\underline{\underline{\boldsymbol{B}}}} = \sum_{i,j,k} A_{ijk} B_{ijk} \tag{A.11}$$

$$(\underline{\underline{\underline{\underline{\boldsymbol{A}}}}} \otimes \underline{\underline{\boldsymbol{B}}}) : \cdot (\underline{\underline{\underline{\boldsymbol{C}}}} \otimes \underline{\underline{\boldsymbol{D}}}) = (\underline{\underline{\underline{\underline{\boldsymbol{A}}}}} : \underline{\underline{\underline{\boldsymbol{C}}}}) \otimes (\underline{\underline{\boldsymbol{B}}} \cdot \underline{\underline{\boldsymbol{D}}}) \tag{A.12}$$

Notice that a matrix can be identified with a second-order tensor, and a vector with a first-order tensor. Also, one can use the following implementation for sixth-order tensor and third-order tensor:

$$\underline{\underline{\underline{\underline{\boldsymbol{A}}}}} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \equiv \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{A}}}}} & \underline{\underline{\underline{\underline{\boldsymbol{0}}}}} \\ \underline{\underline{\underline{\underline{\boldsymbol{0}}}}} & \underline{\underline{\underline{\underline{\boldsymbol{0}}}}} \end{bmatrix}, \quad \text{and} \quad \underline{\underline{\boldsymbol{A}}} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{A}}} \\ \underline{\underline{\boldsymbol{0}}} \end{bmatrix} \tag{A.13}$$

Then, using (A.13) and (A.9) one can easily demonstrates the following property:

$$\begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{11} & \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{12} \\ \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{21} & \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{B}}}_1 \\ \underline{\underline{\boldsymbol{B}}}_2 \end{bmatrix} = \begin{bmatrix} \underline{\underline{\boldsymbol{C}}}_1 \\ \underline{\underline{\boldsymbol{C}}}_2 \end{bmatrix} \Longleftrightarrow \begin{cases} \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{11} : \underline{\underline{\boldsymbol{B}}}_1 + \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{12} : \underline{\underline{\boldsymbol{B}}}_2 & = \underline{\underline{\boldsymbol{C}}}_1 \\ \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{21} : \underline{\underline{\boldsymbol{B}}}_1 + \underline{\underline{\underline{\underline{\boldsymbol{A}}}}}_{22} : \underline{\underline{\boldsymbol{B}}}_2 & = \underline{\underline{\boldsymbol{C}}}_2 \end{cases} \tag{A.14}$$

This property is usefull to build the space–time separated represensation of multi-field formulations.

Appendix B. PGD and MF-PGD algorithms: implementation details

In this appendix, we give implementation details related to PGDs algorithms.

Since multi-field PGD is simply an extension to classical one-field PGD, we can use multi-field algorithms applied to a single field problem in order to compte PGD of one field problem. Indeed, writing MF-PGD definitions for a single field problem leads to the classical single field definitions of PGD.

B.1. Symmetrization

In practice, minimal residual and Galerkin based PGDs are implemented with the same algorithms. Indeed, minimal residual PGD is equivalent to Galerkin based PGD defined with the symmetrized problem. Operators of the symmetrized problem are obtained by preconditionning operators of the reference problem with the adjoint operator [4]. In the multi-field case, this writes:

$$[\underline{\underline{\underline{\underline{\boldsymbol{B}}}}}^{sym}] \equiv \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}^{sym}_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}^{sym}_{12} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}^{sym}_{21} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}^{sym}_{22} \end{bmatrix} = \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{21} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{12} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{12} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{21} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}_{22} \end{bmatrix}$$

$$\text{and} \quad [\underline{\underline{\boldsymbol{L}}}^{sym}] \equiv \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}^{sym}_1 \\ \underline{\underline{\boldsymbol{L}}}^{sym}_2 \end{bmatrix} = \begin{bmatrix} \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{11} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{21} \\ \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{12} & \underline{\underline{\underline{\underline{\boldsymbol{B}}}}}'_{22} \end{bmatrix} : \cdot \begin{bmatrix} \underline{\underline{\boldsymbol{L}}}_1 \\ \underline{\underline{\boldsymbol{L}}}_2 \end{bmatrix}$$

In the following, we implicitely refer to the operators of the symmetrized problem for the construction of the minimal residual formulation of the MF-PGD.

B.2. Progressive algorithm

In order to express the two stationary conditions associated with the minimization problem (59a), we introduce the following mappings:

$$\mathcal{S}_m : \begin{cases} \mathbb{R}^{n_1^T + n_2^T} & \longrightarrow \mathbb{R}^{n_1^S + n_2^S} \\ \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} & \mapsto \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix} = \mathcal{S}_m\left(\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix}\right) \Longleftrightarrow \begin{bmatrix} \underline{\underline{\boldsymbol{S}}}^{ii}_{11} & \underline{\underline{\boldsymbol{S}}}^{ii}_{12} \\ \underline{\underline{\boldsymbol{S}}}^{ii}_{21} & \underline{\underline{\boldsymbol{S}}}^{ii}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix} \end{cases}$$
$$= \begin{bmatrix} \underline{\boldsymbol{F}}^S_{1i} \\ \underline{\boldsymbol{F}}^S_{2i} \end{bmatrix} - \sum_{j=1, j \neq i}^{m} \begin{bmatrix} \underline{\underline{\boldsymbol{S}}}^{ij}_{11} & \underline{\underline{\boldsymbol{S}}}^{ij}_{12} \\ \underline{\underline{\boldsymbol{S}}}^{ij}_{21} & \underline{\underline{\boldsymbol{S}}}^{ij}_{22} \end{bmatrix} \cdot \left(\alpha_j \begin{bmatrix} \underline{\boldsymbol{W}}_{1j} \\ \underline{\boldsymbol{W}}_{2j} \end{bmatrix}\right) \tag{B.1}$$

$$\text{and} \quad \mathcal{T}_m : \begin{cases} \mathbb{R}^{n_1^S + n_2^S} & \longrightarrow \mathbb{R}^{n_1^T + n_2^T} \\ \begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix} & \mapsto \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} = \mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1i} \\ \underline{\boldsymbol{W}}_{2i} \end{bmatrix}\right) \Longleftrightarrow \begin{bmatrix} \underline{\underline{\boldsymbol{T}}}^{ii}_{11} & \underline{\underline{\boldsymbol{T}}}^{ii}_{12} \\ \underline{\underline{\boldsymbol{T}}}^{ii}_{21} & \underline{\underline{\boldsymbol{T}}}^{ii}_{22} \end{bmatrix} \cdot \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1i} \\ \underline{\boldsymbol{\Lambda}}_{2i} \end{bmatrix} \end{cases}$$
$$= \begin{bmatrix} \underline{\boldsymbol{F}}^T_{1i} \\ \underline{\boldsymbol{F}}^T_{2i} \end{bmatrix} - \sum_{j=1, j \neq i}^{m} \begin{bmatrix} \underline{\underline{\boldsymbol{T}}}^{ij}_{11} & \underline{\underline{\boldsymbol{T}}}^{ij}_{12} \\ \underline{\underline{\boldsymbol{T}}}^{ij}_{21} & \underline{\underline{\boldsymbol{T}}}^{ij}_{22} \end{bmatrix} \cdot \left(\alpha_j \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1j} \\ \underline{\boldsymbol{\Lambda}}_{2j} \end{bmatrix}\right) \tag{B.2}$$

with the following implementations

$$\underline{\underline{\boldsymbol{S}}}^{ij}_{mn} = \sum_{k=1}^{nB(m,n)} (\underline{\boldsymbol{\Lambda}}_{mi} \cdot \underline{\underline{\boldsymbol{B}}}^{Tk}_{mn} \cdot \underline{\boldsymbol{\Lambda}}_{nj}) \underline{\underline{\boldsymbol{B}}}^{Sk}_{mn}$$

$$\underline{\underline{\boldsymbol{T}}}^{ij}_{mn} = \sum_{k=1}^{nB(m,n)} (\underline{\boldsymbol{W}}_{mi} \cdot \underline{\underline{\boldsymbol{B}}}^{Sk}_{mn} \cdot \underline{\boldsymbol{W}}_{nj}) \underline{\underline{\boldsymbol{B}}}^{Tk}_{mn}$$

$$\text{and} \quad \underline{\boldsymbol{F}}^S_{mi} = \sum_{k=1}^{nL(m)} (\underline{\boldsymbol{\Lambda}}_{mi} \cdot \underline{\boldsymbol{L}}^{Tk}_m) \underline{\boldsymbol{L}}^{Sk}_m$$

$$\underline{\boldsymbol{F}}^T_{mi} = \sum_{k=1}^{nL(m)} (\underline{\boldsymbol{W}}_{mi} \cdot \underline{\boldsymbol{L}}^{Sk}_m) \underline{\boldsymbol{L}}^{Tk}_m \tag{B.3}$$

Then we easily show that the two stationary conditions associated with minimization problem (59a) are equivalent to:

$$\alpha_m \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} = \mathcal{S}_m\left(\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix}\right) \tag{B.4a}$$

$$\alpha_m \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} = \mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right) \tag{B.4b}$$

Finally, using the following property of mapping (B.1),

$$\alpha_m \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} = \mathcal{S}_m\left(\mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right) \Big/ \alpha_m\right) \Longleftrightarrow \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} = \mathcal{S}_m\left(\mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right)\right) \tag{B.5}$$

it is easy to demonstrate that the progressive definition of the multi-field PGD (Eq. (59)) is equivalent to the following fixed point problem:

$$\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} = \mathcal{S}_m\left(\mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right)\right)$$

$$\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} = \mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right) \Big/ \alpha_m$$

$$\alpha_m = \left\| \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} \otimes \mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right) \right\|_{\underline{\underline{\underline{[\underline{\boldsymbol{N}}]}}}}$$

This naturally leads to Algorithm 1 for the computation of the progressive multi-field PGD.

Algorithm 1. Progressive MF-PGD

```
for $m = 1$ to $m_{max}$ do
    Initialize $\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix}$
    for$k = 1$ to $k_{max}$ do
        $\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} = \mathcal{S}_m\left(\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix}\right)$
        $\underline{\boldsymbol{W}}_{1m} = \underline{\boldsymbol{W}}_{1m} / \|\underline{\boldsymbol{W}}_{1m}\|_2$
        $\underline{\boldsymbol{W}}_{2m} = \underline{\boldsymbol{W}}_{2m} / \|\underline{\boldsymbol{W}}_{2m}\|_2$
        $\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} = \mathcal{T}_m\left(\begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix}\right)$
        Check convergence of $\underline{\boldsymbol{W}}_{1m}$ and $\underline{\boldsymbol{W}}_{2m}$
    end for
    $\alpha_m = \left\| \begin{bmatrix} \underline{\boldsymbol{W}}_{1m} \\ \underline{\boldsymbol{W}}_{2m} \end{bmatrix} \otimes \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} \right\|_{\underline{\underline{\underline{[\underline{\boldsymbol{N}}]}}}}$
    $\begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{\Lambda}}_{1m} \\ \underline{\boldsymbol{\Lambda}}_{2m} \end{bmatrix} \Big/ \alpha_m$
end for
```

B.3. Optimal algorithm

Using mappings $\mathcal{S}_m$ and $\mathcal{T}_m$, it is easy to show that the two stationnary conditions associated with the minimization problem (60a) are given by:

$$\alpha_i \begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} = \mathcal{S}_m \left(\begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} \right) \tag{B.6a}$$

$$\alpha_i [\, \boldsymbol{\Lambda}_{1i} \boldsymbol{\Lambda}_{2i} \,] = \mathcal{T}_m \left(\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \right) \tag{B.6b}$$

$\forall i = 1, \ldots, m$

Thanks to property (B.5), the optimal definition of the MF-PGD (Eq. (60)) is equivalent to the following fixed point problem:

$$\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} = \mathcal{S}_m \left(\mathcal{T}_m \left(\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \right) \right) \tag{B.7a}$$

$$\begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} = \frac{\mathcal{T}_m \left(\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \right)}{\alpha_i} \tag{B.7b}$$

$$\alpha_i = \left\| \begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \otimes \mathcal{T}_m \left(\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \right) \right\|_{\underline{\underline{\underline{\boldsymbol{N}}}}} \tag{B.7c}$$

$\forall i = 1, \ldots, m$

Our strategy for the resolution of the fixed point problem (B.7a) can be viewed as a Gauss–Seidel type substitution method. In order to limit algorithmic complexity, we proceed progressively. At a given rank m, we suppose that a decomposition of rank $m - 1$ has been calculated. Then, we accurately resolve the fixed point problem associated with the new space–time mode of rank m, and we finally update the remainder modes by performing only one iteration of their associated fixed point problem. This leads to Algorithm 2.

Algorithm 2. Optimal MF-PGD

```
1: for$m = 1$ to $m_{max}$do
2:    Perform steps 2 to 11 of Algorithm 1
3:    for $i = 1$ to $m$do
4:        $\begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} = \mathcal{S}_m \left( \begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} \right)$
5:        $\boldsymbol{W}_{1i} = \boldsymbol{W}_{1i} / \|\boldsymbol{W}_{1i}\|_2$
6:        $\boldsymbol{W}_{2i} = \boldsymbol{W}_{2i} / \|\boldsymbol{W}_{2i}\|_2$
7:        $\begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} = \mathcal{T}_m \left( \begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \right)$
8:        $\alpha_i = \left\| \begin{bmatrix} \boldsymbol{W}_{1i} \\ \boldsymbol{W}_{2i} \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} \right\|_{[\underline{\underline{\underline{\boldsymbol{N}}}}]}$
9:        $\begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{\Lambda}_{2i} \end{bmatrix} \Big/ \alpha_i$
10:    end for
11: end for
```

One should notice that in the proposed algorithm, the number of spatial and temporal problems, necessary to compute the decomposition of rank m, increases with $O(m^2)$. Then, this algorithm is computationally inefficient in terms of CPU time, as soon as the decomposition rank m is to big. However, the convergence of the calculated decomposition, in the frobenius norm, is closed to the optimal decomposition obtained with SVD, and then the proposed algorithm can be used to *a priori* compute an optimal decomposition, at less cost than subspace iteration type algorithm [44].

B.4. Reordering

For decomposition reordering, we use the following algorithm.

Algorithm 3. Reordering MF-PGD

```
1: Give $[\underline{\underline{\underline{\boldsymbol{N}}}}]$
2: Perform Algorithms 1 or 2
3: Rearrange $\{\alpha_i\}_{1,\ldots,m_{max}}$ in decreasing order
4: Rearrange $\left\{ \begin{bmatrix} \boldsymbol{W}_{1i} \otimes \boldsymbol{\Lambda}_{1i} \\ \boldsymbol{W}_{2i} \otimes \boldsymbol{\Lambda}_{2i} \end{bmatrix} \right\}_{1,\ldots,m_{max}}$ in function of $\{\alpha_i\}_{1,\ldots,m_{max}}$
```

References

[1] R. Abedi, B. Petracovici, H.B. Haber, A spacetime discontinuous Galerkin method for linearized elastodynamics with element-wise momentum balance, Comput. Methods Appl. Mech. Engrg. 195 (2006) 3247–3273.

[2] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory, J. Non-Newtonian Fluid Mech. 139 (2006) 153–176.

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

[4] A. Ammar, The proper generalized decomposition: a powerful tool for model reduction, Int. J. Mater. Form. 3 (2010) 89–100.

[5] A. Ammar, F. Chinesta, A. Falcó, On the convergence of a Greedy rank-one update algorithm for a class of linear systems, Arch. Comput. Methods Engrg. 17 (2010) 473–486.

[6] A. Ammar, F. Chinesta, P. Diez, A. Huerta, An error estimator for separated representations of highly multidimensional models, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1872–1880.

[7] J. Ballani, L. Grasedyck, A projection method to solve linear system in tensor format, Numer. Linear Algebra Appl. (2012), http://dx.doi.org/10.1002/nla.1818.

[8] G. Beylkin, M.J. Mohlenkamp, Algorithms for numerical analysis in high dimensions, SIAM J. Sci. Comput. 26 (2005) 2133–2159.

[9] B. Bognet, F. Bordeu, F. Chinesta, A. Leygue, A. Poitou, Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity, Comput. Methods Appl. Mech. Engrg. 201–204 (2012) 1–12.

[10] M. Cannarozzi, M. Mancuso, Formulation and analysis of variational methods for time integration of linear elastodynamics, Comput. Methods Appl. Mech. Engrg. 127 (1995) 241–257.

[11] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene, F. Boust, Alleviating mesh constraints: model reduction, parallel time integration and high resolution homogenization, Comput. Methods Appl. Mech. Engrg. 197 (2008) 400–413.

[12] F. Chinesta, A. Ammar, A. Leygue, R. Keunings, An overview of the proper generalized decomposition with applications in computational rheology, J. Non-Newtonian Fluid Mech. 166 (2011) 578–592.

[13] D. Dureisseix, P. Ladevèze, B.A. Schrefler, A computational strategy for multiphysics problems: application to poroelasticity, Int. J. Numer. Methods Engrg. 56 (10) (2003) 1489–1510.

[14] B.F. Feeny, R. Kappagantu, On the physical interpretation of proper orthogonal modes in vibrations, J. Sound Vib. 211 (1998) 607–616.

[15] C.A. Felippa, K.C. Park, C. Farhat, Partitioned analysis of coupled mechanical systems, Comput. Methods Appl. Mech. Engrg. 190 (2001) 3247–3270.

[16] D.A. French, T.E. Peterson, A continuous space–time finite element method for the wave equation, Math. Comput. 65 (1996) 491–506.

[17] F. Galland, A. Gravouil, E. Malvesin, M. Rochette, A global model reduction approach for 3D fatigue crack growth with confined plasticity, Comput. Methods Appl. Mech. Engrg. 200 (2011) 699–716.

[18] M. Géradin, D. Rixen, Mechanical Vibrations, Theory and Application to Structural Dynamics, Wiley, 1994.

[19] P. Glosmann, E. Kreuzer, On the application of Karhunen–Loeve transform to transient dynamic systems, J. Sound Vib. 328 (2009) 507–519.

[20] D. González, A. Ammar, F. Chinesta, E. Cueto, Recent advances on the use of separated representations, Int. J. Numer. Methods Engrg. 81 (2010) 637–659.

[21] H.M. Hilber, T.J.R. Hughes, R.L. Taylor, Improved numerical dissipation for time integration algorithms in structural dynamics, Earthquake Engrg. Struct. Dynam. 5 (1977) 283–292.

[22] H.M. Hilber, T.J.R. Hughes, Collocation, dissipation and 'Overshoot' for time integration schemes in structural dynamics, Earthquake Engrg. Struct. Dynam. 6 (1978) 99–118.

[23] B. Hübner, E. Walhorn, D. Dinkler, A monolithic approach to fluid structure interaction using spacetime finite elements, Comput. Methods Appl. Mech. Engrg. 193 (2004) 2087–2104.

[24] T.J.R. Hughes, The Finite Element Method, Linear Static and Dynamic Finite Element Analysis, Prentice Hall, Englewood Cliffs, 1987.

[25] G.M. Hulbert, T.J.R. Hughes, Space–time finite element methods for second-order hyperbolic equations, Comput. Methods Appl. Mech. Engrg. 84 (1990) 327–348.

[26] G.M. Hulbert, Time finite element methods for structural dynamics, Int. J. Numer. Methods Engrg. 33 (1992) 307–331.

[27] G.M. Hulbert, Computational structural dynamics, Encyclopedia of Computational Mechanics, John Wiley & Sons, 2004.

[28] M. Itskov, Tensor Algebra and Tensor Analysis for Engineers, with Applications to Continuum Mechanics, Springer, 2007.

[29] G. Kerschen, J.C. Golinval, Physical interpretation of the proper orthogonal modes using the singular value decomposition, J. Sound Vib. 249 (5) (2002) 849–865.

[30] G. Kerschen, J.C. Golinval, A.F. Vakakis, L.A. Bergman, The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview, Nonlinear Dynam. 41 (2005) 147–169.

[31] P. Ladevèze, Nonlinear Computational Structural Mechanics: New Approaches and Non-Incremental Methods of Calculation, Springer, 1999.

[32] P. Ladevèze, J.C. Passieux, D. Néron, The LATIN multiscale computational method and the proper generalized decomposition, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1287–1296.

[33] P. Ladevèze, L. Chamoin, On the verification of model reduction methods based on the proper generalized decomposition, Comput. Methods Appl. Mech. Engrg. 200 (2011) 2032–2047.

[34] C. Le Bris, T. Lelièvre, Y. Maday, Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations, Constr. Approx. 30 (2009) 621–651.

[35] A. Leygue, E. Verron, A first step towards the use of proper general decomposition method for structural optimization, Arch. Comput. Methods Engrg. 17 (2010) 465–472.

[36] E. Liberge, A. Hamdouni, Reduced order modelling method via proper orthogonal decomposition (POD) for flow around an oscillating cylinder, J. Fluids Struct. 26 (2010) 292–311.

[37] A. Lichnerowicz, Éléments de Calcul Tensoriel, Armand Colin, 1950.

[38] C. Michler, S.J. Hulshoff, E.H. van Brummelen, R. de Borst, A monolithic approach to fluid structure interaction, Comput. Fluids 33 (2004) 839–848.

[39] D. Néron, D. Dureisseix, A computational strategy for thermo-poroelastic structures with a time–space interface coupling, Int. J. Numer. Methods Engrg. 75 (9) (2008) 1053–1084.

[40] D. Néron, P. Ladevèze, Proper generalized decomposition for multiscale and multiphysics problems, Arch. Comput. Methods Engrg. 17 (2010) 351–372.

[41] N.M. Newmark, A method of computation for structural dynamics, J. Eng. Mech. Div. 9 (1959) 121–142.

[42] A. Nouy, P. Ladevèze, Multiscale computational strategy with time and space homogenization: a radial-type approximation technique for solving micro problems, Int. J. Multisc. Comput. Engrg. 170 (2) (2004) 557–574.

[43] A. Nouy, A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations, Comput. Methods Appl. Mech. Engrg. 196 (2007) 4521–4537.

[44] A. Nouy, Generalized spectral decomposition method for solving stochastic finite element equations: invariant subspace problem and dedicated algorithms, Comput. Methods Appl. Mech. Engrg. 197 (2008) 4718–4736.

[45] A. Nouy, Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems, Arch. Comput. Methods Engrg. 17 (2010) 403–434.

[46] A. Nouy, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1603–1626.

[47] A. Placzek, D.M. Tran, R. Ohayon, Hybrid proper orthogonal decomposition formulation for linear structural dynamics, J. Sound Vib. 318 (2008) 943–964.

[48] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of multidimensional parametric models using the proper generalized decomposition, Math. Comput. Simul. 81 (4) (2010) 791–810.

[49] D. Ryckelynck, A priori hyperreduction method: an adaptive approach, J. Comput. Phys. 202 (2005) 346–366.

[50] G.W. Stewart, On the early history of the singular value decomposition, SIAM Rev. 35 (4) (1993) 551–566.

[51] C.F. Van Loan, The ubiquitous Kronecker product, J. Comput. Appl. Math. 123 (2000) 85–100.

[52] P. Vidal, L. Gallimard, O. Polit, Assessment of a composite beam finite element based on the proper generalized decomposition, Compos. Struct. (2011), http://dx.doi.org/10.1016/j.compstruct.2011.12.016.

[53] A. Zilian, D. Dinkler, A. Vehre, Projection-based reduction of fluid structure interaction systems using monolithic space–time modes, Comput. Methods Appl. Mech. Engrg. 198 (2009) 3795–3805.

[54] O.C. Zienkiewicz, A new look at the Newmark, Houbolt and other time stepping formulas. A weighted residual approach, Earthquake Engrg. Struct. Dynam. 5 (1977) 413–418.