MOX-Report No. 20/2021

A separated representation involving multiple time scales within the Proper Generalized Decomposition framework

Pasquale, A.; Ammar, A.; Falcó, A.; Perotto, S.; Cueto, E.;

Duval, J.-L.; Chinesta, F.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

A Separated Representation involving Multiple Time Scales within the Proper Generalized Decomposition framework

Angelo Pasquale[1,2], Amine Ammar[1], Antonio Falcó[3], Simona Perotto[4], Elías Cueto[5], Jean-Louis Duval[6], Francisco Chinesta[7]

April 10, 2021

[1]ESI Chair, LAMPA, Arts et Métiers ParisTech, 2 Boulevard du Ronceray BP 93525, 49035 Angers cedex 01, France
[2]Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
[3]ESI Chair, CEU-UCH Cardenal Herrera, San Bartolomée 55, 46115 Alfara del Patriarca (Valencia), Spain
[4]MOX, Dept. of Mathematics, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Milano, Italy
[5]Aragon Institute of Engineering Research, Universidad de Zaragoza, Calle de Mariano Esquillor, s/n, 50018 Zaragoza, Spain
[6]ESI Group, Parc Icade, Immeuble le Seville, 3 bis, Saarinen, CP 50229, 94528 Rungis Cedex, France
[7]ESI Chair, PIMM, Arts et Métiers ParisTech, 151 Boulevard de l'Hopital, F-75013 Paris, France
simona.perotto@polimi.it, angelo.pasquale@mail.polimi.it,
{angelo.pasquale,amine.ammar,francisco.chinesta}@ensam.eu,
afalco@uchceu.es, ecueto@unizar.es, jean-louis.duval@esi-group.com

Abstract

Solutions of partial differential equations can exhibit multiple time scales. Standard discretization techniques are constrained to capture the finest scale to accurately predict the response of the system. In this paper, we provide an alternative route to circumvent prohibitive meshes arising from the necessity of capturing fine-scale behaviors. The proposed methodology is based on a time-separated representation within the standard Proper Generalized Decomposition, where the time coordinate is transformed into a multi-dimensional time through new separated coordinates, each representing one scale, while continuity is ensured in the scale coupling. For instance, when considering two different time scales, the governing Partial Differential Equation is commuted into a nonlinear system that iterates between the so-called microtime and macrotime, so that the time coordinate can be viewed as a 2D time. The macroscale effects are taken into account by means of a finite element-based macro-discretization, whereas the microscale effects are handled with unidimensional parent spaces that are replicated throughout the time domain. The resulting separated representation allows us a very fine time discretization without impacting the computational efficiency. The proposed formulation is explored and numerically verified on thermal and elastodynamic problems.

1 Introduction

Many engineering problems are defined in very large time intervals (e.g., when dealing with fatigue, aging, dynamics with loadings involving multiple characteristic times) and, at the same time, the response must encompass the different time scales present in the model. Following standard time marching approaches, a suitable time step that captures the evolution of the finest scale has to be adopted to ensure a reliable modeling, deriving into a prohibitive simulation cost. Many numerical algorithms have been proposed to bypass the time marching approach up to the finest time scale.

For instance, when only macro-scale effects are envisaged, time-homogenization techniques perform well, although subject to a number of hypotheses which are not always realistic, as well as to the scale separation.

In [1], it is proposed a technique that makes use of the separation of variables. The time domain is partitioned into a number of subdomains, where times defining the subdomain partition describe the macrotime, whereas the subdomain discretization resolves fast responses (microtime). The main issue characterizing this technique is the necessity of ensuring the continuity of the resulting two-scale discretization. Lagrange multipliers are adopted for this purpose. This choice makes substantially prohibitive the computational implementation associated with this procedure.

In [4, 6] a numerical algorithm based on the LArge Time INcrement–Proper Generalized Decomposition (LATIN–PGD) is presented. Temporal and spatial multi-scale behaviors in solid mechanic problems are handled by associating a temporal macro basis with the interfacial degrees of freedom coupling different macro domains. The scalability of the methodology is restored via an appropriate correction of the temporal basis founded on the residual.

Also in [5], the time domain is partitioned into subintervals, and a common reduced basis is adopted in every subinterval. Afterwards, additional interface restrictions are imposed to ensure the continuity of the primal variable and of the associated time derivative. Hence, values at the endpoints of each macro interval are enforced to vanish, and the offset is calculated on the macro discretization.

An alternative view is proposed in [7, 3], within the so-called Parareal scheme, that iterates between macro and micro domain partitions, so that the initial conditions for each micro interval are provided by the macro resolution of the problem. The algorithm is easily parallelizable, making it very efficient for addressing general models.

Another numerical strategy based on combining different PGDs under the partition of unity (PU) rationale is proposed in [8, 2]. Here, the main idea is to use macro shape functions fulfilling the partition of unity. This ensures moving throughout different PGDs in diverse intervals while guaranteeing a perfect continuity. Actually, the overlap between the PGDs solutions related to the overlapping subdomains ensures the continuity when the PGDs solutions are multiplied by the macroscopic shape functions thanks to the standard PU features. Despite its effectiveness, the necessity of combining microscopic discretization with the macroscopic PU enrichment, hinders the computational implementation of this approach.

This paper aims at computing the time evolution of the unknown fields involved by a multiscale Partial Differential Equation (PDE), taking into account all the information that it involves, i.e., covering all the time scales. The solution procedure that we are proposing includes neither major hypotheses, nor the need of time scale separation, as for time homogenization techniques. In more detail, the new approach ensures continuity in a direct and computationally inexpensive manner without resorting to the use of Lagrange multipliers, penalty or the PU paradigm. Our proposal directly derives from the discrete tensor formulation of the separated representation involved in the PGD constructor, making use of a tensor formalism. For that purpose, in the next section, after providing the basics for a PGD tensor representation, we introduce the new approach to manage temporal multi-scale problems and we apply it to the heat equation. In Section 3 we extend such an approach to computational solid mechanics and, more concretely, to elastodynamics. Finally Section 4 shortly summarizes the main achievements of the present work.

2 Time multiscale with PGD

2.1 Tensor-based PGD methods

Here we shortly recall the PGD algorithm in a general tensor framework, as described in [12, 13]. The solution of a multidimensional problem is built as a sum of tensor products of functions defined in some sub-spaces with moderate dimension (1, 2 or 3), thus providing a general separable representation form. In particular, the authors make use of the best rank-1 approximation property of tensors of order 3 or higher [18] (in [15], it has been proved that tensors of order 3 or higher can fail to have best rank-n approximation) to propose an iterative method based on the so-called *projection-enrichment technique* [12].

Let Ω be a multidimensional domain involving several coordinates x_i (not necessarily one-dimensional), which can coincide, for instance, with spatial coordinates, time, model or geometric parameters. We consider the weak formulation of a linear problem:

$$\begin{gathered} \textit{Find } \psi \in V(\Omega) \textit{ s.t.} \\ a(\psi(x_1,\ldots,x_d),\psi^*(x_1,\ldots,x_d)) = b(\psi^*(x_1,\ldots,x_d)) \quad \forall \psi^* \in V(\Omega), \end{gathered} \tag{2.1}$$

with $V(\Omega)$ an adequate function space ensuring the well-posedness of such a formulation.

We assume that, after a discretization of problem (2.1), we are lead to solve the linear system:

$$^{\mathrm{t}}\boldsymbol{\psi}^*\mathcal{A}\,\boldsymbol{\psi} = {}^{\mathrm{t}}\boldsymbol{\psi}^*\mathcal{B}, \tag{2.2}$$

where the operator on the left-hand side and the right-hand side member are expressed in a separated form as

$$\mathcal{A} = \sum_{k=1}^{N_A} \mathbf{A}_1^k \otimes \mathbf{A}_2^k \otimes \cdots \otimes \mathbf{A}_d^k, \quad \mathcal{B} = \sum_{k=1}^{N_B} \mathbf{b}_1^k \otimes \mathbf{b}_2^k \otimes \cdots \otimes \mathbf{b}_d^k. \tag{2.3}$$

The approximated PGD solution $\boldsymbol{\psi}$ is sought in the discrete separated form

$$\boldsymbol{\psi} = \sum_{j=1}^{N_F} \alpha_j\, \mathbf{f}_1^j \otimes \mathbf{f}_2^j \otimes \ldots \otimes \mathbf{f}_d^j, \tag{2.4}$$

and an analogous representation is adopted for the test function $\boldsymbol{\psi}^*$, being

$$\boldsymbol{\psi}^* = \mathbf{f}_1^i \otimes \mathbf{f}_2^i \otimes \ldots \otimes \mathbf{f}_d^i,$$

with $i = 1,\ldots,N_F$.

Solution $\boldsymbol{\psi}$ is built-up by using a projection-enrichment iterative scheme [12]. In particular, it will be assumed that the global convergence is attained when the error estimator $\varepsilon = \|\mathcal{A}\,\boldsymbol{\psi} - \mathcal{B}\|_2$ is small enough, where $\|\cdot\|_2$ denotes the standard Frobenius norm of a tensor of order d.

The projection stage consists of finding the set of coefficients α_j in (2.4) verifying the relations

$$\sum_{j=1}^{N_F} H_{ij}\alpha_j = J_i \quad \forall i = 1,\ldots,N_F$$

where

$$H_{ij} = \sum_{k=1}^{N_A} \left({}^{\mathrm{t}}\mathbf{f}_1^i\, \mathbf{A}_1^k\, \mathbf{f}_1^j\right) \cdot \left({}^{\mathrm{t}}\mathbf{f}_2^i\, \mathbf{A}_2^k\, \mathbf{f}_2^j\right) \cdot \ldots \cdot \left({}^{\mathrm{t}}\mathbf{f}_d^i\, \mathbf{A}_d^k\, \mathbf{f}_d^j\right),$$

$$J_i = \sum_{k=1}^{N_B} \left({}^{\mathrm{t}}\mathbf{f}_1^i\, \mathbf{b}_1^k\right) \cdot \left({}^{\mathrm{t}}\mathbf{f}_2^i\, \mathbf{b}_2^k\right) \cdot \ldots \cdot \left({}^{\mathrm{t}}\mathbf{f}_d^i\, \mathbf{b}_d^k\right).$$

The enrichment stage includes new candidates for enriching the reduced separated approximation basis, so that $\boldsymbol{\psi}$ can be updated as

$$\boldsymbol{\psi} = \underbrace{\sum_{j=1}^{N_F} \alpha_j \, \mathbf{f}_1^j \otimes \mathbf{f}_2^j \otimes \cdots \otimes \mathbf{f}_d^j}_{=:\boldsymbol{\psi}_{\mathbf{f}}} + \underbrace{\mathbf{r}_1 \otimes \mathbf{r}_2 \otimes \cdots \otimes \mathbf{r}_d}_{=:\boldsymbol{\psi}_{\mathbf{r}}}, \tag{2.5}$$

and (2.2) is replaced by the system

$$^{\mathrm{t}}\boldsymbol{\psi}^* \mathcal{A} \, \boldsymbol{\psi}_{\mathbf{r}} + {}^{\mathrm{t}}\boldsymbol{\psi}^* \mathcal{A} \, \boldsymbol{\psi}_{\mathbf{f}} = {}^{\mathrm{t}}\boldsymbol{\psi}^* \mathcal{B}. \tag{2.6}$$

In particular, within a fixed point alternating direction algorithm, at each iteration we look for the computation of a single discrete function, $\mathbf{r}_j$, all the other components of $\boldsymbol{\psi}_{\mathbf{r}}$ being assumed known, and after setting $\boldsymbol{\psi}^*$ to

$$\mathbf{r}_1 \otimes \cdots \mathbf{r}_{j-1} \otimes \mathbf{r}_j^* \otimes \mathbf{r}_{j+1} \cdots \otimes \mathbf{r}_d \, .$$

This strategy leads to solve the linear system

$$\mathbf{K} \, \mathbf{r}_j + \mathbf{v} = \tilde{\mathbf{v}}, \tag{2.7}$$

where

$$\mathbf{K} = \sum_{k=1}^{N_A} \Bigg(\mathbf{A}_j^k \prod_{\substack{h=1 \\ h \neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h \, \mathbf{A}_h^k \, \mathbf{r}_h \Bigg),$$

$$\mathbf{v} = \sum_{i=1}^{N_F} \sum_{k=1}^{N_A} \Bigg(\alpha_i \, \mathbf{A}_j^k \, \mathbf{f}_j^i \prod_{\substack{h=1 \\ h \neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h \, \mathbf{A}_h^k \, \mathbf{f}_h^i \Bigg), \quad \tilde{\mathbf{v}} = \sum_{k=1}^{N_B} \Bigg(\mathbf{b}_j^k \prod_{\substack{h=1 \\ h \neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h \, \mathbf{b}_h^k \Bigg).$$

For the explicit computations leading to system (2.7), we refer the interested reader to [12], whereas a convergence analysis for this greedy rank-1 update algorithm can be found in [18].

Remark 2.1. As discussed in [12], the convergence of the fixed point strategy characterizing the enrichment step is guaranteed for symmetric discrete operators. Actually, numerical tests performed with non symmetric discrete operators (e.g., with hyperbolic operators) exhibit some difficulties in the convergence of the fixed point procedure as well as a loss of optimality in terms of number of function products to be computed. For this reason, all the problems analyzed in this work are suitably rewritten in order to deal with a symmetric operator. In general, the new problem to be solved reads:

$$^{\mathrm{t}}\boldsymbol{\psi}^* \, {}^{\mathrm{t}}\!\mathcal{A}\mathcal{A} \, \boldsymbol{\psi} = {}^{\mathrm{t}}\boldsymbol{\psi}^* \, {}^{\mathrm{t}}\!\mathcal{A}\mathcal{B},$$

and the associated projection stage becomes

$$\sum_{j=1}^{N_F} H^S_{ij}\alpha_j = J^S_i \quad \forall i = 1,\dots,N_F$$

where

$$H^S_{ij} = \sum_{k'=1}^{N_A}\sum_{k=1}^{N_A} \left({}^{\mathrm{t}}\mathbf{f}^i_1\, {}^{\mathrm{t}}\mathbf{A}^{k'}_1\, \mathbf{A}^k_1\, \mathbf{f}^j_1\right) \cdot \left({}^{\mathrm{t}}\mathbf{f}^i_2\, {}^{\mathrm{t}}\mathbf{A}^{k'}_2\, \mathbf{A}^k_2\, \mathbf{f}^j_2\right) \cdot \ldots \cdot \left({}^{\mathrm{t}}\mathbf{f}^i_d\, {}^{\mathrm{t}}\mathbf{A}^{k'}_d\, \mathbf{A}^k_d\, \mathbf{f}^j_d\right),$$

$$J^S_i = \sum_{k'=1}^{N_A}\sum_{k=1}^{N_B} \left({}^{\mathrm{t}}\mathbf{f}^i_1\, {}^{\mathrm{t}}\mathbf{A}^{k'}_1\, \mathbf{b}^k_1\right) \cdot \left({}^{\mathrm{t}}\mathbf{f}^i_2\, {}^{\mathrm{t}}\mathbf{A}^{k'}_2\, \mathbf{b}^k_2\right) \cdot \ldots \cdot \left({}^{\mathrm{t}}\mathbf{f}^i_d\, {}^{\mathrm{t}}\mathbf{A}^{k'}_d\, \mathbf{b}^k_d\right).$$

Analogously, system (2.6) will be replaced by the new one

$${}^{\mathrm{t}}\boldsymbol{\psi}^*\, {}^{\mathrm{t}}\mathcal{A}\,\mathcal{A}\,\boldsymbol{\psi}_{\mathbf{r}} + {}^{\mathrm{t}}\boldsymbol{\psi}^*\, {}^{\mathrm{t}}\mathcal{A}\,\mathcal{A}\,\boldsymbol{\psi}_{\mathbf{f}} = {}^{\mathrm{t}}\boldsymbol{\psi}^*\, {}^{\mathrm{t}}\mathcal{A}\,\mathcal{B},$$

so that system (2.7) takes the form

$$\mathbf{K}^S\,\mathbf{r}_j + \mathbf{v}^S = \tilde{\mathbf{v}}^S,$$

with

$$\mathbf{K}^S = \sum_{k'=1}^{N_A}\sum_{k=1}^{N_A} \left({}^{\mathrm{t}}\mathbf{A}^{k'}_j\, \mathbf{A}^k_j \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h\, {}^{\mathrm{t}}\mathbf{A}^{k'}_h\, \mathbf{A}^k_h\, \mathbf{r}_h \right),$$

$$\mathbf{v}^S = \sum_{i=1}^{N_F}\sum_{k'=1}^{N_A}\sum_{k=1}^{N_A} \left(\alpha_i\, {}^{\mathrm{t}}\mathbf{A}^{k'}_j\, \mathbf{A}^k_j\, \mathbf{f}^i_j \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h\, {}^{\mathrm{t}}\mathbf{A}^{k'}_h\, \mathbf{A}^k_h\, \mathbf{f}^i_h \right),$$

$$\tilde{\mathbf{v}}^S = \sum_{k'=1}^{N_A}\sum_{k=1}^{N_B} \left({}^{\mathrm{t}}\mathbf{A}^{k'}_j\, \mathbf{b}^k_j \prod_{\substack{h=1\\h\neq j}}^{d} {}^{\mathrm{t}}\mathbf{r}_h\, {}^{\mathrm{t}}\mathbf{A}^{k'}_h\, \mathbf{b}^k_h \right).$$

2.2 A two-time scale separated representation within the PGD method

In the proposal of numerical methods to manage multi-scale phenomena, a crucial point is represented by the capability to ensure the continuity of the approximation. This turns out to be an issue also when dealing with a PGD approach. For instance, in [8], PGD is combined with the Partition of Unity (PU) paradigm to circumvent the continuity problem. Thus, a generic multiscale function u is expressed as

$$u(x) = \sum_{i=1}^{N} N_i(x) u_i \sum_{j=1}^{J} G_j(\tau(x - x_i)) g_j, \tag{2.8}$$

where the collection, $\{x_i\}_{i=1}^N$, of the so-called centroids x_i introduces a macro-partition of the time window, $N_i(x)$ is the macroscale shape function associated with the i-th centroid, x_i, $\tau(x - x_i)$ is a dependent variable characterized by an offset based on x_i, and $G_j(\tau)$ denotes the j-th microscale shape function, with g_j the corresponding microscale degree of freedom. Here, the microscale effects occurring in the compact support of the macroscale shape function $N_i(x)$ are mapped into the corresponding microscale space, via τ. Formulation (2.8) allows the authors to automatically ensure the continuity to the approximation u as proved in [8]. Other formulations enforce the continuity in different way, for instance by resorting to Lagrange multipliers as in [1].
The present work takes a step forward with respect to the available literature by proposing a new approach which ensures the continuity of the PGD solution in a computationally inexpensive manner, without resorting to Lagrange multipliers or penalization strategies.

We exemplify the new method on a simple problem, i.e., on the heat equation solved in a one-dimensional spatial domain $\Omega_x = (0, L)$ over the time window $\Omega_t = (0, T_f)$, and completed with non homogeneous Dirichlet data,

$$\begin{cases} \dfrac{\partial u}{\partial t} - k\dfrac{\partial^2 u}{\partial x^2} = f & \text{in } \Omega_x \times \Omega_t \\ u(x,0) = u_0(x) & \text{in } \Omega_x \\ u(0,t) = u(L,t) = u_g(t) & \text{in } \Omega_t, \end{cases} \tag{2.9}$$

with k the diffusivity coefficient and $f = f(x,t)$ the source term.

A standard time-marching approach requires a discretization of Ω_t by means of N_t times. On the contrary, the technique here presented is based on splitting the whole time interval $(0, T_f)$ into M *macro* subintervals $\Omega_{i+1} = [T_i, T_{i+1}]$, for $i = 0, \ldots, M-1$, of constant amplitude ΔT, by choosing a sequence $\{T_i\}_{i=0}^M$ of equispaced *macro-dofs* and by setting $T_0 = 0$ and $T_M = T_f$. Afterwards, in the first macro subinterval, a new continuous time-*microscale*, $\tau(\Omega_1)$, is introduced so that the subinterval Ω_1 is discretized by involving $m+1$ *micro-dofs*, $\{\tau_j^1\}_{j=0}^m$ and setting $\tau_0^1 = T_0$. Also the discretization at the microscale is uniform, with a constant time step set to $\Delta\tau$ throughout Ω_1. Moreover, the last micro-dof is internal to Ω_1, being $\tau_m^1 = T_1 - \Delta\tau$. By means of a tensor product, the micro discretization in Ω_1 is replicated throughout the other subintervals Ω_{i+1}, covering the whole time window (see Figure 1 for a sketch). In such a way, a fine time grid consisting of $N_t = M(m+1)$ dofs is obtained by mapping $m+1$ micro-dofs onto a coarser grid of M macro intervals, without *a priori* needing a fine discretization of the whole time window. In particular, the value of the solution at the right endpoint, T_{i+1}, of each subinterval Ω_{i+1} is recovered by means of this tensor product. As a consequence, such a discretization is intentionally not meant to predict values at $T_M = T_f$. Figure 2 shows the Cartesian product of the macro and micro dofs together, with the associated nodal evaluations of the generic unknown function u in (2.9).

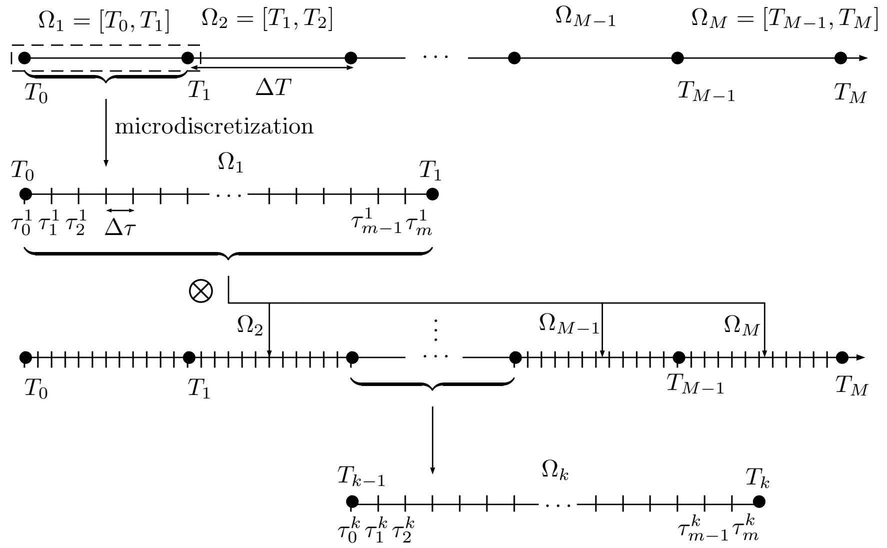

Figure 1: Construction of macro and micro scales.

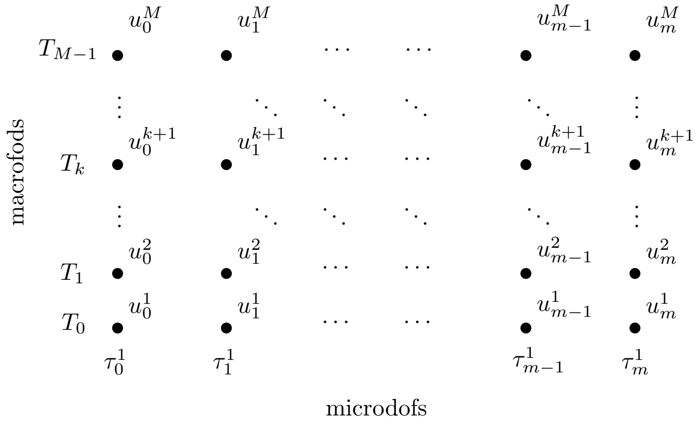
u_0^M u_1^M u_{m-1}^M u_m^M
T_{M-1}
u_0^{k+1} u_1^{k+1} u_{m-1}^{k+1} u_m^{k+1}
T_k
u_0^2 u_1^2 u_{m-1}^2 u_m^2
T_1
u_0^1 u_1^1 u_{m-1}^1 u_m^1
T_0
τ_0^1 τ_1^1 τ_{m-1}^1 τ_m^1
macrofods
microdofs

Figure 2: Tensor product of the macro and micro dofs.

If we discretize the time derivative in (2.9) by means of a standard finite difference formula, we have:

$$\forall n \in \{0, 1, \ldots, N_t - 1\}, \quad \frac{\partial u}{\partial t} \approx \frac{1}{\Delta t}(u_{n+1} - u_n), \tag{2.10}$$

that we can write also as

$$\frac{\partial u}{\partial t} \approx \mathbf{E}_{N_t}\,\mathbf{u}, \tag{2.11}$$

with

$$\mathbf{E}_{N_t} = \frac{1}{\Delta t}\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{pmatrix}_{N_t} \quad \text{and } \mathbf{u} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_t} \end{pmatrix}, \tag{2.12}$$

where the first equation understands that the initial value u_0 has to be included in the right-hand side of the equation associated with $n = 0$, while the subscript N_t for the operator $\mathbf{E}$ denotes the size of the corresponding matrix. This notation will be particularly useful to simplify the discussion which follows.

Using a two-scale time variable, the time derivative in (2.10) is replaced by the new one

$$\frac{\partial}{\partial \tau} u(T, \tau),$$

where $T \in \{T_i\}_{i=0}^{M-1}$ is adopted to identify the subinterval Ω_{i+1} of interest, while $\tau \in [T_0, T_1)$ represents the actual temporal independent variable, spanning Ω_1, i.e, via tensor product, the generic subinterval Ω_{i+1}. This leads to perform the time derivation with respect to variable τ.

With a view to the application of PGD for approximating problem (2.9) with the two-scale time discretization, matrix $\mathbf{E}_{N_t}$ in (2.12) associated with the standard time coordinate deserves to be modified in order to replicate the PGD tensor structure. The idea is to introduce a suitable macro-micro decomposition of the time derivative operator in (2.11). For this purpose, let us identify the partial derivative of order zero with respect to T, $\frac{\partial^0}{\partial T^0}$, with the identity operator, so that, $\frac{\partial^0}{\partial T^0} f(T) = f(T)$, and, as a consequence, identity

$$\frac{\partial}{\partial \tau} u(T, \tau) = \left(\frac{\partial^0}{\partial T^0} \otimes \frac{\partial}{\partial \tau} \right) u(T, \tau)$$

holds true. This equality suggested the authors in [1] to propose the approximation

$$\frac{\partial}{\partial t} \approx \frac{\partial^0}{\partial T^0} \otimes \frac{\partial}{\partial \tau}, \tag{2.13}$$

for the standard time derivative, which will be investigated and further developed throughout this work. Approximation (2.13) can thus be written in matrix form as

$$\mathbf{E}_{N_t} \approx \mathbf{E}_{T^0,\tau} = \mathbf{E}_{m+1} \otimes \mathbf{I}_M = \begin{pmatrix} \mathbf{E}_{m+1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{E}_{m+1} \end{pmatrix}_M, \tag{2.14}$$

where now the subscript M refers to the number of blocks constituting matrix $\mathbf{E}_{T^0,\tau}$, while matrix $\mathbf{E}_{m+1}$, sized according to discretization at the micro-scale, is defined as in (2.12), after replacing Δt with $\Delta\tau$, and replicated throughout the subintervals Ω_{i+1} by means of the Kronecker product with the discrete identity operator $\mathbf{I}_M$.[1]
However, definition (2.14) leads to have some entries along the first sub-diagonal which are identically null. This happens essentially when switching from a block to the subsequent one, namely, when moving from the last internal dof associated with the micro-scale in a certain subinterval, Ω_{i+1}, to the first dof associated with the macro-scale of the subinterval Ω_{i+2}, resulting in a loss of continuity between the fine and the coarse time grids. An effective, yet computationally demanding, remedy to this issue is based on enforcing the continuity in all the macrodofs, either by using a penalized formulation or by employing Lagrange multipliers, as detailed in [1].

In this work, we propose to exploit the properties of the tensor product to recover the equality in (2.14). To this aim, we introduce the decomposition

$$\begin{aligned}\mathbf{E}_{N_t} = {} & \frac{1}{\Delta\tau}\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0\\ -1 & 1 & 0 & 0 & \cdots & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & -1 & 1 & 0\\ 0 & \cdots & \cdots & 0 & -1 & 1 \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0\\ 0 & 1 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & 1 & 0\\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}_{M} + \\ & \frac{1}{\Delta\tau}\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -1\\ 0 & 0 & 0 & 0 & \cdots & 0\\ 0 & \ddots & \ddots & \ddots & \ddots & 0\\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & 0 & 0 & 0\\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0\\ 1 & 0 & \ddots & \ddots & \vdots\\ 0 & 1 & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & 0\\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}_{M}\end{aligned} \tag{2.15}$$

which, in a compact form, reads as

$$\mathbf{E}_{N_t} = \mathbf{E}_{m+1} \otimes \mathbf{I}_M + \mathbf{C}_{m+1} \otimes \mathbf{L}_M \tag{2.16}$$

[1] We observe that symbol $\otimes$ refers to a revision of the standard Kronecker product, being meant

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} B_{1,1}\,\mathbf{A} & B_{1,2}\,\mathbf{A} & \cdots & B_{1,n}\,\mathbf{A}\\ B_{2,1}\,\mathbf{A} & B_{2,2}\,\mathbf{A} & \cdots & B_{2,n}\,\mathbf{A}\\ \vdots & \vdots & \ddots & \vdots\\ B_{m,1}\,\mathbf{A} & B_{m,2}\,\mathbf{A} & \cdots & B_{m,n}\,\mathbf{A} \end{pmatrix} \quad \text{and} \quad \mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} b_1\,\mathbf{a}\\ \vdots\\ b_m\,\mathbf{a} \end{pmatrix},$$

respectively. This is a common choice in the PGD community to preserve the same order of the variables in the separated form [14].

where matrices $\mathbf{C}_{m+1}$ and $\mathbf{L}_M$ are defined as

$$(\mathbf{C}_{m+1})_{ij} = -\frac{1}{\Delta\tau}\begin{cases} 1 & \text{if } i=1, j=m+1 \\ 0 & \text{otherwise,} \end{cases} \qquad (\mathbf{L}_M)_{ij} = \delta_{i,j+1}, \tag{2.17}$$

respectively with δ_{ij} the Kronecker symbol. The decomposition in (2.16) allows us to reconstruct the correct time derivative operator while preserving the macro-micro decomposition characterizing the two-scale time derivative operator. This method embraces the PGD formalism while reducing significantly the computational cost, since no extra computational effort is required to enforce continuity at each dof associated with the macroscale.

Remark 2.2. The multiscale approach proposed in this paper exhibits two main benefits:

1. the possibility of describing in an efficient way physical phenomena with the fine scale resolution;
2. a description richness equal to $N_t = M(m+1)$ accomplished with a computational complexity of the order of $M+m+1$, thus yielding considerably reduced memory allocation requirements to store time operators.

Coming back to the approximation of the heat equation in (2.9), a classical discrete PGD approximation, based on a space-time separated form, reads:

$$\textit{Find } \mathbf{U}_H = \sum_{k=1}^{N_U} \mathbf{u}_1^k \otimes \mathbf{u}_2^k \textit{ such that } \mathcal{A}_H\,\mathbf{U}_H = \mathcal{B}_H, \tag{2.18}$$

where N_U denotes the number of the PGD modes, index 1 refers to the space dependence, index 2 is associated with the time, $\mathcal{B}_H$ is a suitable discrete separated approximation of the source term $f(t,x)$ evaluated at some discrete points, $\mathcal{A}_H$ is a discrete separated representation of the differential operators in (2.9), built by exploiting the tensorial identity

$$\frac{\partial}{\partial t} - k\frac{\partial^2}{\partial x^2} = \left(\frac{\partial}{\partial t} \otimes \frac{\partial^0}{\partial x^0}\right) - k\left(\frac{\partial^0}{\partial t^0} \otimes \frac{\partial^2}{\partial x^2}\right), \tag{2.19}$$

and assembled according to the discretization adopted for the space (e.g., finite differences, finite elements) and for the time. With the same notation as in (2.13), we identify the partial derivatives of order zero with the corresponding identity operator. For instance, let us assume to discretize (2.9) by employing the implicit Euler scheme for the time and a centered finite difference approximation for the space derivative, with N_x the number of spatial dofs and N_t the number of discrete times. Then, the operator $\mathcal{A}_H$ can be separated as

$$\mathcal{A}_H = \sum_{k=1}^{2} \mathbf{A}_1^k \otimes \mathbf{A}_2^k \text{ with } \mathbf{A}_1^1 = \mathbf{I}_{N_x},\ \mathbf{A}_2^1 = \mathbf{E}_{N_t}, \mathbf{A}_1^2 = -k\,\mathbf{D}_{N_x},\ \mathbf{A}_2^2 = \mathbf{I}_{N_t}, \tag{2.20}$$

where $\mathbf{E}_{N_t}$ denotes the matrix associated with the implicit Euler scheme, $\mathbf{I}_{N_x}$ and $\mathbf{I}_{N_t}$ are the identity matrices in space and time, respectively and $\mathbf{D}_{N_x}$ is the matrix of the discrete Laplacian. Figure 3 shows the sparsity pattern associated with four matrices $\mathbf{A}_i^k$.

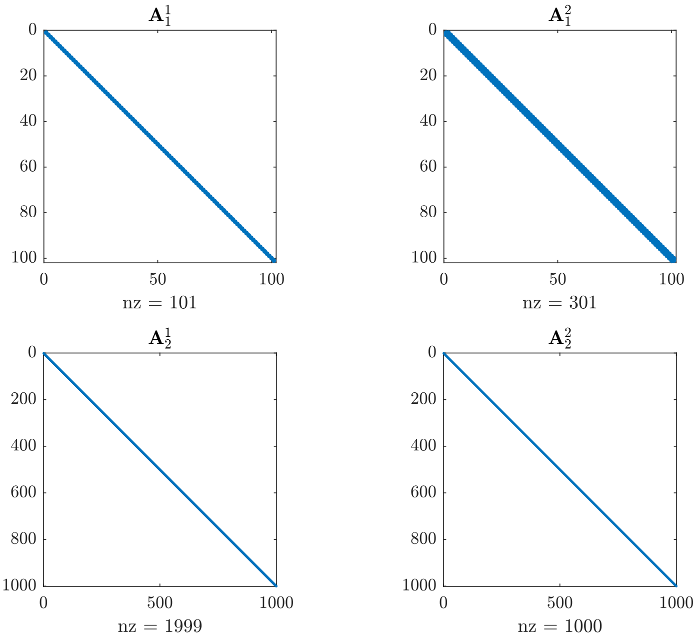

Figure 3: Sparsity pattern of discrete PGD operators associated with the separated representation (2.20), for $N_x = 101$ and $N_t = 1000$.

Now, in order to apply the new PGD formulation based on a two-scale time discretization to problem (2.9), we have to recast the differential operators in (2.19) into a multiscale framework, by adding a further temporal independent variable. This leads us to deal with a framework characterized by two dimensions in time and one dimension in space. For this purpose, we adopt the approximation

$$\begin{aligned}\frac{\partial}{\partial t} - k\frac{\partial^2}{\partial x^2} &= \left(\frac{\partial}{\partial t} \otimes \frac{\partial^0}{\partial x^0}\right) - k\left(\frac{\partial^0}{\partial t^0} \otimes \frac{\partial^2}{\partial x^2}\right) \\ &\approx \left(\frac{\partial^0}{\partial T^0} \otimes \frac{\partial}{\partial \tau} \otimes \frac{\partial^0}{\partial x^0}\right) - k\left(\frac{\partial^0}{\partial T^0} \otimes \frac{\partial^0}{\partial \tau^0} \otimes \frac{\partial^2}{\partial x^2}\right),\end{aligned}$$

and we enforce the continuity between the macro and the micro time partitions by resorting to the two-scale decomposition in (2.16), so that the operator $\mathcal{A}_H$

in (2.18) can be separated as

$$\mathcal{A}_H = \sum_{k=1}^{3} \mathbf{A}_1^k \otimes \mathbf{A}_2^k \otimes \mathbf{A}_3^k, \tag{2.21}$$

where, analogously as in (2.18), subscript 1 identifies the space dependence, while subscripts 2 and 3 refer to the time microscale and to the time macroscale, respectively. The spatial operators are defined in a straightforward way as

$$\mathbf{A}_1^1 = \mathbf{A}_1^2 = \mathbf{I}_{N_x}, \quad \mathbf{A}_1^3 = -k\,\mathbf{D}_{N_x},$$

whereas the two-scale time discretization leads us to adopt the modified tensorial decomposition in (2.15),

$$\mathbf{E}_{N_t} = \mathbf{A}_2^1 \otimes \mathbf{A}_3^1 + \mathbf{A}_2^2 \otimes \mathbf{A}_3^2,$$

with $\mathbf{A}_2^1 = \mathbf{E}_{m+1}$ the matrix characterizing the implicit Euler discretization at the microscale; $\mathbf{A}_3^1 = \mathbf{A}_3^3 = \mathbf{I}_M$ the discrete identity matrix associated with the subintervals Ω_{i+1}; $\mathbf{A}_2^2 = \mathbf{C}_{m+1}$ the correction matrix, sized according to the microscale, with a unique non-null element in the upper-right corner; $\mathbf{A}_3^2 = \mathbf{L}_M$ the lower shift matrix, sized according to the macroscale, with the first subdiagonal of elements all equal to one; $\mathbf{A}_2^3 = \mathbf{I}_{m+1}$ the discrete identity matrix associated with the microscale. Figure 4 gathers the sparsity pattern of the nine matrices involved in the separated representation (2.21).

Concerning the right-hand side in (2.18), operator $\mathcal{B}_H$ coincides with a separated representation, with respect to the spatial scale and the two temporal ones, of the source term $f(t,x)$ in (2.9), evaluated at some discrete points, and computed through a HOSVD [16].

Remark 2.3. A cross comparison between Figures 3 and 4 highlights that the two-scale PGD formulation of problem (2.9) does not affect the spatial discretization. On the contrary, the matrices associated with the two-scale time discretization are much smaller with respect to the ones characterizing the separated representation in (2.20), with a consequent reduction in terms of memory allocation (in particular, the storage requirement for the time matrices reduces from $3(m+1)M-1$ to $3(m+1+M)-1$, which means of a factor about equal to M^{-1}, for $M \ll m+1$). Notice that the time decomposition in Figure 4 suits the time grid in Figure 3, being $M(m+1) = N_t = 1000$.

3 Multiscale resolution of transient elastodynamic problems

This section generalizes the multiscale PGD formulation detailed on the heat equation in the previous section to a more challenging setting represented by an

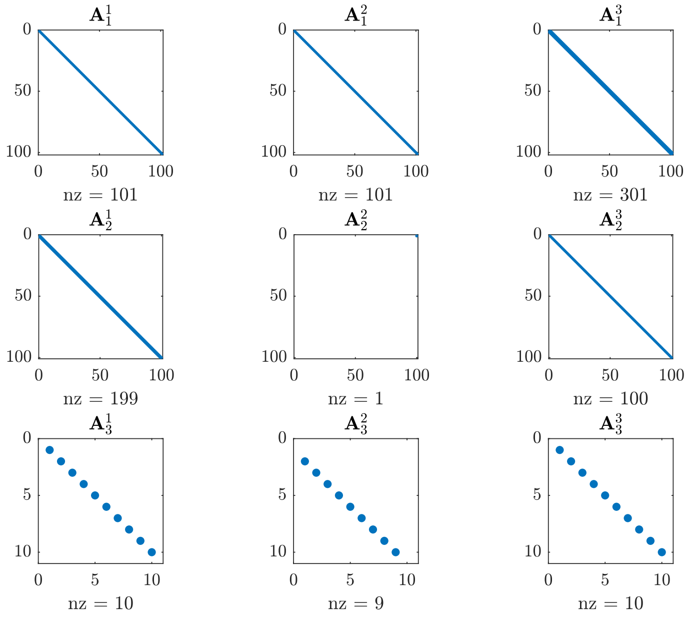

Figure 4: Sparsity pattern of the discrete PGD operators associated with the separated representation (2.21), for $N_x = 101$, $m + 1 = 100$ and $M = 10$.

elastodynamic problem in a multiscale transient regime. In this setting, different strategies based on the separation of the space and time are investigated in [14]. Here, the authors propose a multi-field Proper Generalized Decomposition which adopts standard finite elements to discretize the space, to be combined with several discretization schemes in time, such as the Newmark scheme, a single-field time discontinuous Galerkin method, a two-fields time continuous and discontinuous Galerkin approach.

To simplify the discussion, we focus on a transient elastodynamic problem in a one-dimensional spatial domain which models, for instance, the traction-compression waves travelling along a linear elastic medium, $\Omega = (0, L)$, during time interval $I = [0, T)$. The scalar displacement field is denoted by $u = u(x, t)$, for $x \in \Omega$ and $t \in I$. The medium is subject to a displacement, $u^d = u^d(x, t)$, and to an external force, $f^d = f^d(x, t)$, applied to portions, $\partial\Omega_u \times I$ and $\partial\Omega_f \times I$, of the boundary of the space-time cylinder, respectively, with $\partial\Omega_f \cup \partial\Omega_u = \partial\Omega$ and $\partial\overset{\circ}{\Omega}_f \cap \partial\overset{\circ}{\Omega}_u = \emptyset$ (notice that, in the specific context at hand, $\partial\Omega_u$ and $\partial\Omega_f$ reduce to the endpoints of the domain Ω, dealing with a one-dimensional setting). The initial state is described by the displacement field, $u_0 = u_0(x)$, and by the velocity field, $v_0 = v_0(x)$, at time $t = 0$. Finally, the medium is characterized by a density, ρ, an elasticity modulus, E, and a section, A. Thus, the strong formulation of the considered transient elastodynamic problem reads: find $u : \Omega \times I \to \mathbb{R}$, such that

$$\begin{cases} \rho \dfrac{\partial^2 u}{\partial t^2} = E \dfrac{\partial^2 u}{\partial x^2} & \text{in } \Omega \times I, \\ u = u^d & \text{on } \partial\Omega_u \times I, \\ \dfrac{\partial u}{\partial x} = \dfrac{f^d}{EA} & \text{on } \partial\Omega_f \times I, \\ u(x,0) = u_0(x) & \text{on } \Omega \times \{0\}, \\ \dfrac{\partial u}{\partial t}(x,0) = v_0(x) & \text{on } \Omega \times \{0\}. \end{cases} \tag{3.1}$$

To simplify the notation, in the following we adopt the convention $\dot{u}$ and $\ddot{u}$ to denote the first- and the second-order time derivative, $\dfrac{\partial u}{\partial t}$ and $\dfrac{\partial^2 u}{\partial t^2}$, of u respectively.

We employ a standard centered finite difference scheme to approximate the space, after discretizing Ω with N_x dofs, so that the associated semi-discrete formulation is: find $\mathbf{u} : I \to \mathbb{R}^{N_x}$, such that

$$\begin{cases} \mathbf{M}\,\ddot{\mathbf{u}}(t) + \mathbf{K}\,\mathbf{u}(t) = \mathbf{F}(t), \\ \mathbf{u}(0) = \mathbf{u}_0, \\ \dot{\mathbf{u}}(0) = \mathbf{v}_0, \end{cases} \tag{3.2}$$

where $\mathbf{M} = \dfrac{\rho}{E}\mathbf{I}_{N_x}$ and $\mathbf{K} = -\mathbf{D}_{N_x}$ denote the mass and the stiffness matrix, respectively, with $\mathbf{I}_{N_x}$ and $\mathbf{D}_{N_x}$ the identity and discrete Laplacian operators, $\mathbf{u}_0$ and $\mathbf{v}_0$ are the vectors sampling the initial data, u_0 and v_0, at the spatial dofs, and where the boundary data have been properly included in the right-hand side $\mathbf{F}$.

To discretize the time, we subdivide the whole time window, I, by means of (N_t+1) uniformly distributed times, t_i, such that $t_0 = 0$, $t_{N_t} = T$, and $t_i - t_{i-1} = \Delta t$, for $i = 1, \ldots, N_t$. Following [14], we apply the Newmark scheme to (3.2), so that we have to solve the fully discretized problem: find $\mathbf{u} \in \mathbb{R}^{N_x \times N_t}$, such that

$$\begin{aligned} (\mathbf{K} \otimes \mathbf{N}_1 + \mathbf{M} \otimes \mathbf{N}_2) : \mathbf{u} = (\mathbf{I}_{N_x} \otimes \mathbf{N}_1) : \mathbf{F} + (\mathbf{M}\,\mathbf{u}_0) \otimes \mathbf{N}_3 + (\mathbf{M}\,\mathbf{v}_0) \otimes \mathbf{N}_4 \\ + (\mathbf{F}(t_0) - \mathbf{K}\,\mathbf{u}_0) \otimes \mathbf{N}_5, \end{aligned} \tag{3.3}$$

where : denotes the two times contracted product between tensors[2],

$$\mathbf{N}_1 = \begin{pmatrix} \beta & 0 & \cdots & \cdots & 0 \\ a & \ddots & \ddots & \ddots & \vdots \\ b & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \cdots & 0 & b & a & \beta \end{pmatrix}_{N_t}, \qquad \mathbf{N}_2 = \frac{1}{\Delta t^2} \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -2 & \ddots & \ddots & \ddots & \vdots \\ 1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \cdots & 0 & 1 & -2 & 1 \end{pmatrix}_{N_t}, \tag{3.4}$$

$$\mathbf{N}_3 = \frac{1}{\Delta t^2} \begin{pmatrix} 1 \\ -1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{N_t}, \qquad \mathbf{N}_4 = \frac{1}{\Delta t} \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}_{N_t}, \qquad \mathbf{N}_5 = \begin{pmatrix} \frac{1}{2} - \beta \\ \frac{1}{2} + \beta - \gamma \\ 0 \\ \vdots \\ 0 \end{pmatrix}_{N_t},$$

with $0 \leq \gamma \leq 1$ and $0 \leq \beta \leq \frac{1}{2}$ the parameters involved by the Newmark algorithm, $a = 1/2 - 2\beta + \gamma$, $b = 1/2 + \beta - \gamma$, $\mathbf{F}(t_0)$ is the right-hand side in (3.2) evaluated at the initial time t_0, and $\mathbf{F}$ collects all the vectors $\mathbf{F}(t_i)$ for $i = 1, \ldots, N_t$, accounting for the exterior force and the imposed displacement (we refer the interested reader to the reference paper [14] where the explicit computations leading to linear system (3.3) can be found).

Formulation (3.3) is instrumental to provide the discrete PGD approximation for problem (3.1) based on a standard space-time separation, which reads

$$\textit{Find } \ \mathbf{U}_E = \sum_{k=1}^{N_U} \mathbf{u}_1^k \otimes \mathbf{u}_2^k \ \textit{ such that } \ \mathcal{A}_E \, \mathbf{U}_E = \mathcal{B}_E, \tag{3.5}$$

where

$$\mathcal{A}_E = \sum_{k=1}^{2} \mathbf{A}_1^k \otimes \mathbf{A}_2^k, \quad \text{with} \quad \mathbf{A}_1^1 = \mathbf{M}, \quad \mathbf{A}_2^1 = \mathbf{N}_2, \quad \mathbf{A}_1^2 = \mathbf{K}, \quad \mathbf{A}_2^2 = \mathbf{N}_1, \tag{3.6}$$

the same notation as in (2.18) being adopted for the subscripts identifying the space and the time, respectively, while $\mathcal{B}_E$ simplifies to

$$\mathcal{B}_E = \mathbf{b}_1 \otimes \mathbf{b}_2, \quad \text{with} \quad \mathbf{b}_1 = \frac{\rho}{E} \mathbf{v}_0, \quad \mathbf{b}_2 = \mathbf{N}_4$$

after assuming $\mathbf{F} = \mathbf{0}$ (i.e., no external forces) and $\mathbf{u}_0 = \mathbf{0}$ (i.e., a vanishing initial displacement).

The discrete operators involved in the definition of $\mathcal{A}_E$ have the sparsity pattern shown in Fig. 5, when choosing $\Omega = (0, 1)$ and $I = [0, 100)$.

[2] The two times contracted product : between second or fourth order tensors is defined as $\mathbf{A} : \mathbf{B} = \sum_{i,j} A_{ij} B_{ij}$.

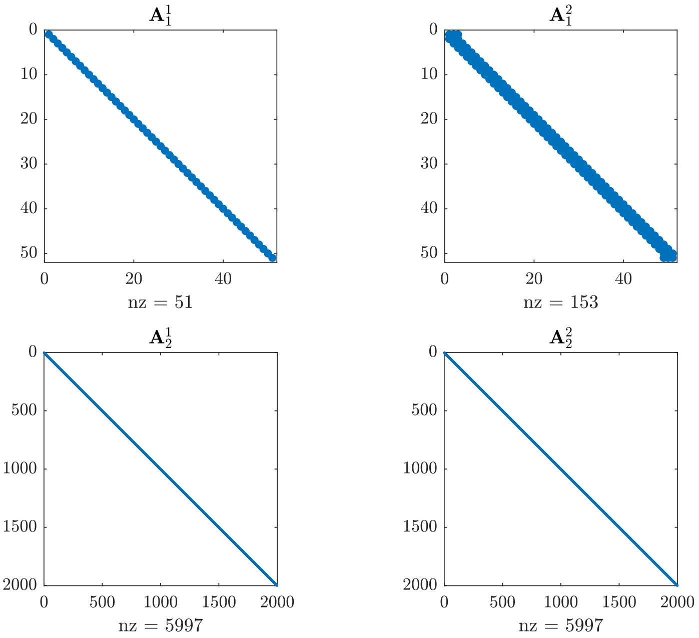

Figure 5: Sparsity pattern of discrete PGD operators associated with the separated representation (3.6), for $N_x = 51$ and $N_t = 2000$.

We now generalize the two-scale temporal discretization for a PGD approximation introduced in the previous section to the elastodynamic problem (3.1). For this purpose, the tensor decomposition correction proposed in (2.15) for the implicit Euler discrete operator, $\mathbf{E}_{N_t}$, is here extended to the Newmark discrete operators, $\mathbf{N}_1$ and $\mathbf{N}_2$, in (3.4), so that it holds

$$\begin{aligned}\mathbf{N}_1 = &\begin{pmatrix} \beta & 0 & 0 & \cdots & \cdots & 0 \\ a & \beta & 0 & 0 & \cdots & 0 \\ b & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & b & a & \beta & 0 \\ 0 & \cdots & \cdots & b & a & \beta \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}_{M} \\ &+ \begin{pmatrix} 0 & 0 & \cdots & 0 & b & a \\ 0 & 0 & \ddots & 0 & \ddots & b \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}_{M} ,\end{aligned} \tag{3.7}$$

$$
\begin{aligned}
\mathbf{N}_2 = \frac{1}{\Delta\tau^2} & \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & \cdots & \cdots & 1 & -2 & 1 \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}_{M} \\
+ \quad \frac{1}{\Delta\tau^2} & \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 & -2 \\ 0 & 0 & \ddots & 0 & \ddots & 1 \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}_{m+1} \otimes \begin{pmatrix} 0 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}_{M} ,
\end{aligned}
\tag{3.8}
$$

$m+1$ and M denoting the number of the temporal micro-dofs and of the time macro subintervals as in Section 2.2, with $N_t = M(m+1)$. Decompositions (3.7) and (3.8) can be rewritten in a compact form as

$$
\mathbf{N}_l = \mathbf{N}_{l,m+1} \otimes \mathbf{I}_M + \mathbf{C}_{l,m+1} \otimes \mathbf{L}_M \quad \text{with} \quad l = 1, 2, \tag{3.9}
$$

where matrices $\mathbf{N}_{1,m+1}$ and $\mathbf{N}_{2,m+1}$ are defined exactly as $\mathbf{N}_1$ and $\mathbf{N}_2$, respectively except for the size (equal to $m+1$ instead of N_t); $\mathbf{I}_M$ is the discrete identity matrix associated with the macroscale; $\mathbf{L}_M$ is the lower shift matrix defined as in (2.17); the correction matrices $\mathbf{C}_{1,m+1}$ and $\mathbf{C}_{2,m+1}$ are given by

$$
(\mathbf{C}_{1,m+1})_{ij} = \begin{cases} a & \text{if } i = 1, j = m+1 \\ b & \text{if } i = 1, j = m \text{ or } i = 2, j = m+1, \\ 0 & \text{otherwise,} \end{cases}
$$

$$
(\mathbf{C}_{2,m+1})_{ij} = \frac{1}{\Delta\tau^2} \begin{cases} -2 & \text{if } i = 1, j = m+1 \\ 1 & \text{if } i = 1, j = m \text{ or } i = 2, j = m+1, \\ 0 & \text{otherwise.} \end{cases}
$$

The tensorial decompositions in (3.9) ensure the continuity between the macro and the micro time scales, analogously to decomposition (2.16). This allows us to extend the temporal continuity between different scales to $\mathcal{A}_E$ in (3.5), after separating this operator in terms of the coordinates x, T and τ, and by exploiting decompositions (3.9), so that we have

$$
\mathcal{A}_E = \sum_{k=1}^{4} \mathbf{A}_1^k \otimes \mathbf{A}_2^k \otimes \mathbf{A}_3^k, \tag{3.10}
$$

subscript 1 referring to the space dependence, while 2 and 3 to the temporal micro- and macroscale, respectively as in the previous section. In particular, the spatial contributions coincide with matrices $\mathbf{M}$ and $\mathbf{K}$ in (3.2), being

$$
\mathbf{A}_1^1 = \mathbf{A}_1^2 = \mathbf{M} = \frac{\rho}{E} \mathbf{I}_{N_x}, \quad \mathbf{A}_1^3 = \mathbf{A}_1^4 = \mathbf{K} = -\,\mathbf{D}_{N_x} \,.
$$

The operators associated with the time microscale are

$$\mathbf{A}_2^1 = \mathbf{N}_{2,m+1}, \quad \mathbf{A}_2^2 = \mathbf{C}_{2,m+1}, \quad \mathbf{A}_2^3 = \mathbf{N}_{1,m+1}, \quad \mathbf{A}_2^4 = \mathbf{C}_{1,m+1},$$

while the ones related to the temporal macroscale are

$$\mathbf{A}_3^1 = \mathbf{A}_3^3 = \mathbf{I}_M, \quad \mathbf{A}_3^2 = \mathbf{A}_3^4 = \mathbf{L}_M\,.$$

In particular, the time microscale is characterized by the Toeplitz matrices, $\mathbf{A}_2^1$ and $\mathbf{A}_2^3$, and by the correction matrices, $\mathbf{A}_2^2$ and $\mathbf{A}_2^4$, with only three not null elements; the diagonal matrices $\mathbf{A}_3^1$ and $\mathbf{A}_3^3$ and the lower shift matrices $\mathbf{A}_3^2$ and $\mathbf{A}_3^4$ identify the time macroscale. The sparsity pattern associated with the twelve operators in (3.10) is shown in Fig. 6, for $\Omega = (0, 1)$ and $I = [0, 100)$.

The right-hand side in (3.5) has to be separated, as operator $\mathcal{A}_E$, in terms of the spatial scale and of the two temporal ones, so that we have

$$\mathcal{B}_E = \mathbf{b}_1 \otimes \mathbf{b}_2 \otimes \mathbf{b}_3, \quad \text{with} \quad \mathbf{b}_1 = \frac{\rho}{E}\,\mathbf{v}_0, \quad \mathbf{b}_2 = \frac{1}{\Delta\tau}\begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}_{m+1}, \quad \mathbf{b}_3 = \begin{pmatrix}1\\0\\\vdots\\0\end{pmatrix}_{M}.$$

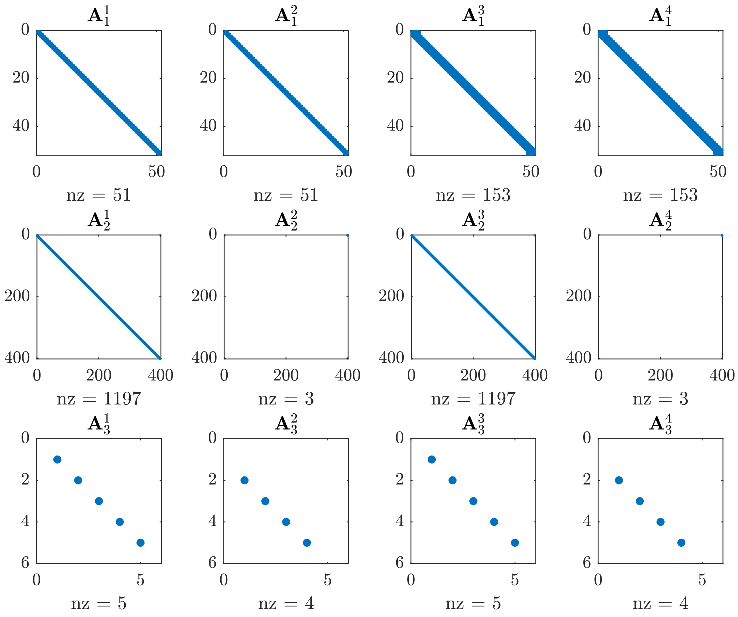

Figure 6: Sparsity pattern of the discrete PGD operators associated with the separated representation (3.10), for $N_x = 51$, $m{+}1 = 400$ and $M = 5$.

Remark 3.1. As expected, it turns out that the temporal multiscale PGD discrete formulation does not affect the spatial representation, the sparsity pattern of matrices $\mathbf{A}_1^1$ and $\mathbf{A}_1^2$ in Fig. 5 being preserved by matrices $\mathbf{A}_1^1$-$\mathbf{A}_1^2$ and $\mathbf{A}_1^3$-$\mathbf{A}_1^4$ of Fig. 6, respectively. On the contrary, the multiscale time discretization yields matrices with a considerably lower size, albeit in a larger number, with a remarkable saving in terms of memory allocation (notice that the decomposition adopted in the two-scale discretization suits the time grid in Fig. 5, being $N_t = M(m+1) = 2000$).

3.1 Numerical simulations

In this section we assess in more detail the PGD approximation combining the two-scale Newmark scheme with the centered finite difference method. To this aim, we consider a beam with constant section $A = 1$, in a traction and compression regime, where an initial velocity, v_0, is prescribed.

Simulations are performed by considering $\Omega = (0,1)$ over the time domain $I = [0,100)$, under the following assumptions:

- average constant acceleration (middle point rule) for Newmark scheme, i.e.,
$$\beta = \frac{1}{4}, \quad \gamma = \frac{1}{2} \implies a = \frac{1}{2}, \quad b = \frac{1}{4};$$
- material parameters: $\kappa = \dfrac{\rho}{E} \in K = \{25, 100, 400, 500\}$;
- sinusoidal initial velocity:
$$v_0(x) = \begin{cases} \dfrac{1}{100} \sin\Big(\dfrac{\pi x}{0.3}\Big)^2 & \text{for } x < 0.3 \\ 0 & \text{otherwise;} \end{cases}$$
- discretization parameters: $N_x = 51$, $m+1 = 400$, $M = 5$.

Figure 7 collects the space-time contour plot for the propagation of the elastic wave, for each $\kappa \in K$. The values in K have been selected in order to obtain solutions with a different level of detail in terms of behaviour at the microscale. Indeed, quantity κ can be easily related to the speed of the stress wave along the beam, after rewriting the differential equation in (3.1) as

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2},$$

with $c = \sqrt{E/\rho} = \sqrt{1/\kappa} \in C = \left\{\frac{1}{5}, \frac{1}{10}, \frac{1}{20}, \frac{1}{10\sqrt{5}}\right\}$. In particular, the time taken by the wave to travel along the whole beam (i.e., to propagate from the left- to the right-endpoint) is equal to $\Delta t = 1/c \in \{5, 10, 20, 10\sqrt{5}\}$. Now, since we have set the number of the macro subintervals to $M = 5$, it turns out that the microscale time variable τ varies in the interval $[0, 20)$. As a consequence, we register two complete wave paths (i.e., two round trips where three reflections take place) for $\kappa = 25$, a single complete wave path (with a unique reflection) when $\kappa = 100$, and a single one-way path for $\kappa = 400$. On the contrary, when $\kappa = 500$, the wave propagates too fast with respect to the microscale adopted, causing a delay between macro-dofs and reflection. This discrepancy is evident, for instance, by the low accuracy characterizing the contour plot in the bottom-right panel of Fig. 7.

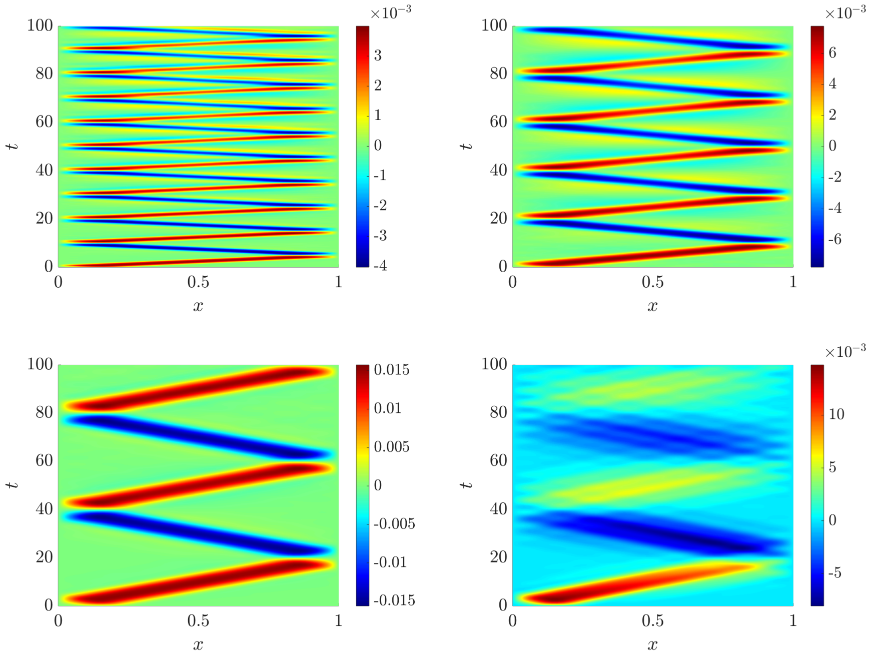

Figure 7: Space-time solutions for $\kappa = 25, 100, 400, 500$ (left-right, top-bottom)

These comments find a further confirmation in the trend characterizing the PGD modes associated with the temporal micro- and macro-scales, and with the spatial one, collected in Figure 8, 9 and 10, respectively for the first three values of κ. In particular, for $\kappa = 25, 100$, the information about the periodicity of the solution are already rich enough at the microscale, which allows us to capture the complete wave path (single and double, respectively), whereas the modes over the macroscale are almost constant (only information about wave amplitude are gathered). On the contrary, when $\kappa = 400$, the microscale follows only a one-way migration and the information about the wave reflection is recovered at the macro level through more informative macroscale modes.

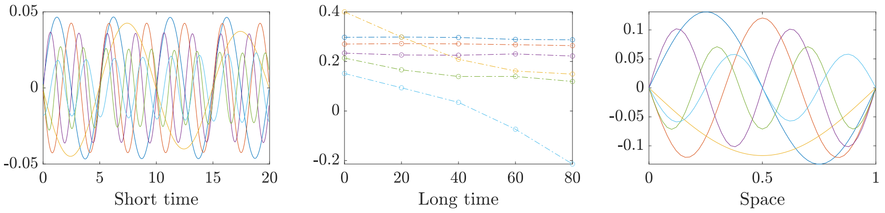

Figure 8: PGD modes - $\kappa = 25$

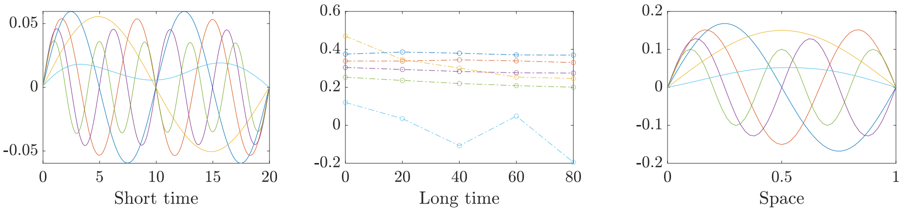

Figure 9: PGD modes - $\kappa = 100$

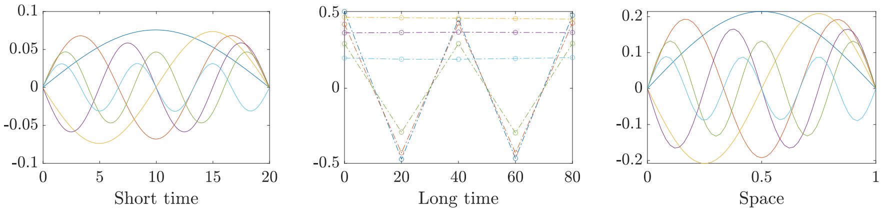

Figure 10: PGD modes - $\kappa = 400$

To sum up, we can state that the choice of the macroscale (and, consequently, of the microscale) is crucial to properly manage the periodicity of the solution. A rough selection of M and of $m+1$ may lead to an inaccurate solution, unless a sufficiently large number of temporal modes is adopted for the PGD reconstruction. Convergence of the PGD procedure can be also deteriorated by a not proper choice of parameters M and $m+1$. This issue is highlighted by Figure 11 which shows the trend of the $L^2(\Omega)$-norm of the relative error associated with the PGD approximation as a function of the number of the PGD enrichment iterations. As a reference, we adopt the solution to (3.2)-(3.3) coinciding with a centered finite difference discretization (in space) combined with the Newmark scheme (in time). In particular, the slow decay of the PGD error in the right panel, for $\kappa = 500$, can be ascribed to the decomposition adopted at the macro- and at the micro-scale, which does not match the periodicity of the wave propagation.

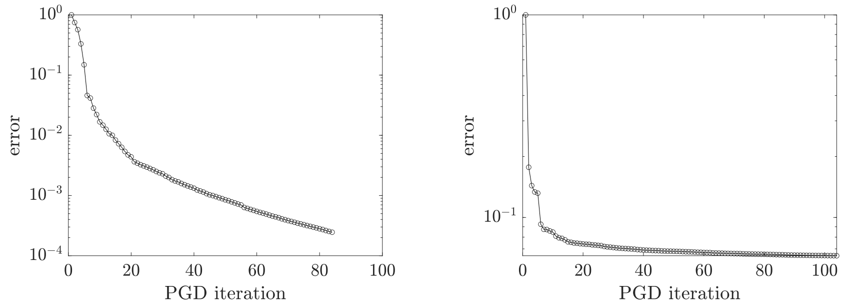

Figure 11: Error decay for successive PGD enrichments for $\kappa = 100$ (left) and $\kappa = 500$ (right)

4 Conclusions

The present work successfully accomplished addressing problems exhibiting different time scales within a multi-time separated Proper Generalized Decomposition. The classical one-dimensional time is converted into a multidimensional temporal variable, where different time coordinates correspond to different levels of resolution. This strategy leads to a considerable reduction in terms of computational complexity. Moreover, the issue related to the continuity enforcement between different scales, that motivated several works in the past, is here addressed in a very simple and efficient way, by writing the tensor decomposition in order to reproduce the tensor form of the single time scale discretization. The proposed method is numerically verified first on the one-dimensional heat problem and successively generalized to the wave propagation in elastodynamics.

References

[1] A. Ammar, F. Chinesta, E. Cueto and M. Doblare. Proper generalized decomposition of time-multiscale models. Int J Numer Methods Eng. 2012; 90: 569-596.

[2] A. Badias, D. Gonzalez, I. Alfaro, F. Chinesta and E. Cueto. Local Proper Generalized Decomposition. Int J Numer Methods Eng. 2017; 112: 1715-1732.

[3] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. Comptes Rendus Mathematique. 2002; 335/4: 387-392.

[4] D. Néron and P. Ladevèze. Proper generalized decomposition for multiscale and multiphysics problems. Arch Comput Methods Eng. 2010; 17/4: 351-372.

[5] F. Fritzen and M. R. Hassani. Space-time model order reduction for nonlinear viscoelastic systems subjected to long-term loading. Meccanica. 2018; 53: 1333–1355.

[6] JC. Passieux, P. Ladevèze, D. Néron. A scalable time-space multiscale domain decomposition method: adaptive time scale separation. Comput Mech. Springer Verlag. 2010; 46/4: 621-633.

[7] Y. Maday, M. Riahi and J. Salomon. Parareal in Time Intermediate Targets Methods for Optimal Control Problems. Control and optimization with PDE constraints. Springer Basel. Internat Ser Numer Math. 2013; 79-92.

[8] R. Ibáñez, A. Ammar, E. Cueto, A. Huerta, J.L. Duval, F. Chinesta. Multiscale Proper Generalized Decomposition based on the Partition of Unity. Int J Numer Methods Eng. 2019; 120/6: 727-747.

[9] F. Chinesta, R. Keunings, A. Leygue. The proper generalized decomposition for advanced numerical simulations. A primer. Springerbriefs in applied sciences and technology. Cham: Springer. 2014.

[10] F. Chinesta, A. Ammar, E. Cueto. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. Arch Comput Methods Eng. 2010; 17/4: 327–50.

[11] F. Chinesta, E. Cueto. PGD-Based Modeling of Materials, Structures and Processes. ESAFORM Bookseries on Material Forming. Cham: Springer. 2014.

[12] A. Ammar. The proper generalized decomposition: a powerful tool for model reduction. Int J Mater Form. 2010; 3: 89-102.

[13] A. Ammar, F. Chinesta, P. Díez and A. Huerta,"An Error Estimator for Separated Representations of Highly Multidimensional Models", Comput Methods Appl Mech Eng. 2010; 199: 1872-1880.

[14] L. Boucinha, A. Gravouil and A. Ammar. Space-time proper generalized decompositions for the resolution of transient elastodynamic models. Comput Methods Appl Mech Eng. 2013; 255: 67-88.

[15] V. de Silva and LH Lim. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. SIAM J Matrix Anal Appl. 2006; 30/3: 1084–1127.

[16] L. Lathauwer and B. De Moor. A Multi-Linear Singular Value Decomposition. J Soc Ind Appl Math. 2000; 21: 1253-1278.

[17] N. M. Newmark. A Method of Computation for Structural Dynamics. J Eng Mech. 1959; 85: 67-94.

[18] A. Ammar, F. Chinesta and A. Falcó. On the Convergence of a Greedy Rank-One Update Algorithm for a Class of Linear Systems. Arch Comput Methods Eng. Springer Verlag. 2010. 17/4: 473-486.

[19] S. Perotto, M.G. Carlino and F. Ballarin. Model reduction by separation of variables: a comparison between Hierarchical Model Reduction and Proper Generalized Decomposition. In Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018. Series: Lect. Notes Comput. Sci. Eng. Springer Nature Switzerland, S.J. Sherwin, D. Moxey, J. Peirò, P.E. Vincent, C. Schwab Eds. 2020; 134: 61-77.

[20] G. Lindfield, J. Penny, Numerical methods - using Matlab. Academic Press. 2019.

[21] 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 Eng. Elsevier. 2008; 197/5: 400-413.

[22] F. Chinesta, P. Ladevèze, E. Cueto. A Short Review in Model Order Reduction Based on Proper Generalized Decomposition. Arch Comput Methods Eng. Springer Verlag. 2011; 18/4: 395-404.

[23] P. Ladevèze, A. Nouy. On a Multiscale Computational Strategy with Time and Space Homogenization for Structural Mechanics. Comput Methods Appl Mech Eng. Elsevier. 2003; 192/28-30: 3061-3087.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

16/2021 Salvador, M.; Dede', L.; Manzoni, A.
Non intrusive reduced order modeling of parametrized PDEs by kernel POD and neural networks

17/2021 Chew, R.; Benacchio, T.; Hastermann, G.; Klein, R.
Balanced data assimilation with a blended numerical model

18/2021 Gigante, G.; Vergara, C.
On the choice of interface parameters in Robin-Robin loosely coupled schemes for fluid-structure interaction

19/2021 Gillard, M.; Benacchio, T.
FT-GCR: a fault-tolerant generalized conjugate residual elliptic solver

13/2021 Ferro, N.; Perotto, S.; Cangiani, A.
An anisotropic recovery-based error estimator for adaptive discontinuous Galerkin methods

14/2021 Peli, R.; Menafoglio, A.; Cervino, M.; Dovera, L.; Secchi, P;
Physics-based Residual Kriging for dynamically evolving functional random fields

15/2021 Fumagalli, A.; Patacchini, F.S.
Model adaptation in a discrete fracture network: existence of solutions and numerical strategies

12/2021 di Cristofaro, D.; Galimberti, C.; Bianchi, D.; Ferrante, R.; Ferro, N.; Mannisi, M.; Perotto, S
Adaptive topology optimization for innovative 3D printed metamaterials

11/2021 Antonietti,P.F.; Manzini, G.; Mazzieri, I.; Scacchi, S.; Verani, M.
The conforming virtual element method for polyharmonic and elastodynamics problems: a review

10/2021 Di Michele, F.; May, J.; Pera, D.; Kastelic, V.; Carafa, M.; Smerzini, C.; Mazzieri, I.; Rubino,
Spectral elements numerical simulation of the 2009 L'Aquila earthquake on a detailed reconstructed domain