A new strategy using the Proper Generalized Decomposition to model time evolving spatial domains

Max Beckermann, Ronan Scanff, Massimiliano Cremonesi, Andrea Barbarulo

Computers and Structures 316 (2025) 107860

Contents lists available at ScienceDirect

Computers and Structures

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

A new strategy using the Proper Generalized Decomposition to model time evolving spatial domains

Max Beckermann [a,b], Ronan Scanff [a], Massimiliano Cremonesi [b], Andrea Barbarulo [a,*]

[a] Université Paris-Saclay, CentraleSupélec, ENS Paris-Saclay, CNRS, LMPS - Laboratoire de Mécanique Paris-Saclay, 91190, Gif-Sur-Yvette, France
[b] Departement of Civil and Environmental Engineering, Politecnico di Milano, Piazza Leonardo da Vinci, 32, 20133, Milan, Italy

HIGHLIGHTS

- Uses the Proper Generalized Decomposition, a powerful Model Order Reduction method.
- Extends effectively to - hardly separable - time evolving space domains.
- Computes space modes in an expanded domain, and the time modes are piecewise.
- Additive manufacturing simulation with sequential activation of hot elements.

ARTICLE INFO

Keywords:
Proper generalized decomposition
Reduced order modeling
Evolving domains
Additive manufacturing

ABSTRACT

In this paper, we propose a new approach to adapt the Proper Generalized Decomposition (PGD) to problems containing space domains that are evolving over time. PGD shines with high-order parametrized and nonlinear problems, drastically reducing their computational time. It was proven highly effective in a wide range of problems, but the space domain has always remained fixed within the parametric manifold of interest. In this work, we adapt the PGD to non-constant domains that change over time at given discrete time instances. More specifically, we focus on time evolving space domains and separate the solution along space and time. The space modes are calculated in an expanded space that comprises all the degrees of freedom throughout the simulation. To visualize the solution, the modes are then projected onto the current physical representation. The time modes are solved in a piecewise manner, dividing the time domain into intervals and initializing the time modes to zero at the beginning of each interval. This work is illustrated with an additive manufacturing-inspired example in which the hot boundary elements are sequentially activated to simulate the addition of material. This aligns perfectly with the previously described strategy as it involves an expanding boundary. The impact of the mesh division and the initialization of the new points is discussed.

1. Introduction

Nowadays, in disciplines related to science and engineering, there is a growing need for accurate large-scale numerical simulations of complex physical phenomena. Moreover, we are faced with a finite and often limited amount of resources to compute the numerical models emanating from the associated partial differential equations. To face this ever-growing demand and to make these simulations faster, mathematical tools called Model Order Reduction (MOR) techniques, have been developed. These techniques generally use the fact that simulation results will always remain highly correlated when varying the problem's parameters. That is to say, the full solution expressed in a high-order space lives in a low-order manifold of reduced dimensions. The goal of these MOR techniques is to identify this low-order manifold and express the original problem in a lower dimension.

Most MOR methods correspond to so-called *a posteriori* techniques and first compute a limited series of high-order simulations to establish the low-order manifold. They then project the equations within this reduced dimensional space, therefore obtaining a solver of reduced dimension. Multiple methods exist for different applications [1–5], with the Proper Orthogonal Decomposition (POD) and Reduced Basis (RB)

* Corresponding author.
Email addresses: max.beckermann@centralesupelec.fr (M. Beckermann), ronan.scanff@ens-paris-saclay.fr (R. Scanff), massimiliano.cremonesi@polimi.it (M. Cremonesi), andrea.barbarulo@centralesupelec.fr (A. Barbarulo).

https://doi.org/10.1016/j.compstruc.2025.107860
Received 14 January 2025; Accepted 9 June 2025
Available online 11 July 2025

method being the most widely used. These techniques differ in the construction of the low-order manifold. In this work, we will focus instead on an *a priori* reduction technique, namely the Proper Generalized Decomposition, or PGD. This technique was introduced by Ladevéze in the 80 s as part of his non-incremental and nonlinear (LATIN) solver [6] and has since then been used in a wide range of domains [7]. Contrary to the previous techniques, it does not require any prior knowledge of solutions. Instead, it iteratively builds a more and more accurate representation of both the low-order manifold and the solution, solving several problems of lower dimension [8]. Several formulations can be found in the bibliography[9,10], the Galerkin definition remains the most popular and is at best equivalent to POD in degenerate cases. PGD is generally characterized by a poor yet monotonic convergence, but at the same time guarantees cheap iterations, the ability to deal with nonlinearities globally, and the possibility to manage a large number of parameters and therefore the curse of dimensionality. PGD was used successfully in nonlinear structural mechanics [6], large scale nonlinear thermal simulations [11,12], complex fluid flows [13–15], data driven control strategies of industrial applications [16], hyperbolic equations and dynamical applications, and wave equations [17–20], amongst many others. More recently, PGD has also been used for nonlinear and preconditioned diffusive equations [21,22], phase transition and phase separation dynamics [23], or with the inviscid Burgers' equation [24]. What all these examples have in common is that they can easily be separated within their parametric manifold of interest, whether it is along time, space, or any additional parameter.

As a matter of fact, MOR for hardly separable domains is still in its infancy and is widely admitted to be the next important frontier in this community [25]. It naturally harbors a lot of interest and methods have been proposed to tackle this issue within the different MOR communities. To cite a few examples, fixed geometrical parametrization and mappings have been used successfully for both RB [26,27] and POD [28]. More recently, mapping approaches aiming to construct the optimal map and the most-reducible snapshot manifold for *a posteriori* approaches have received a lot of interest. To construct those mappings, a range of methods have been used: optimal transport [29], registration-based methods [30], and velocity-driven approaches [31]. Regarding PGD, the advancements have been notably more sparse and would benefit from more attention from the community. Nonetheless, one should highlight the recent work made to combine a NURBS-based geometry description with PGD [32]. It demonstrated that the geometrical parameters can be used in the decomposition itself, which is part of the unique strengths of PGD.

In this work, we will focus on space-time separability for PGD. We will write the approximation of order N of the solution $u(x,t)$ as:

$$u(x,t) \approx \sum_{i=1}^{N} \Lambda_i(x)\lambda_i(t)\,,$$

with $\Lambda_i(x)$ the so-called *space* modes and $\lambda_i(t)$ the *time* modes. The goal of this paper is to look at problems that are by nature not separable because the space domain expands over time, at given discrete time instances. According to the current state of the art in terms of the implementation of PGD [33,34], these hardly separable domains are typical examples for which the PGD still requires further development. We will adapt the PGD formulation to expanding space domains with the important assumption that we know beforehand how the space domain evolves and that this evolution occurs at discrete time instances. Two major problems arise:

- We need to compute global space modes on an expanding domain with a varying number of degrees of freedom. Our solution is to calculate global space modes, accumulating all the degrees of freedom that may exist throughout the simulation in an expanded domain.
- All the points within our space are not initialized simultaneously. Therefore, we decided to calculate the time-modes in a piecewise manner, enforcing exactly the initialization constraints, for every number of modes.

To demonstrate the proposed approach, we took inspiration from what is done in the field of Additive Manufacturing – also called AM – simulations. AM is comprised of a very large range of different processes [35,36] that all have their own modeling method. To just name a few, there are laser-based melting processes such as Selective Laser Sintering [37] or Selective Laser Melting [38], there are thermal extrusion-based processes such as Fused Deposition Modeling [39] or material jetting processes such as 3DP printing [40]. The chosen numerical model also drastically changes depending on the model's scope and goal: it can be a very macro view of the entire manufacturing process, a zoom on a single pass of the addition tool, or a micro view of the heating tip. Our problem will be a modified version of [39] in which we removed the nonlinearities and only considered the thermal physics.

The paper is organized in the following way: in Section 2, we first introduce the problem that we will use throughout the paper and its finite element discretization. In Section 3, the PGD is described, and we then introduce the expanded PGD approach as well as the piecewise approach used for the time modes. In Section 4, the convergence of the method is discussed for a chosen activation scheme. We show the effectiveness of the interpolation operator, compare the results with other activation scenarios and establish the potential computational performance of this new formulation. In Section 5, conclusions are drawn and further developments are suggested.

2. Description of the problem

2.1. The heat equation in an evolving domain

We have opted to examine a simplified 3D representation of an additive manufacturing process in which hot boundary elements are sequentially activated to simulate the addition of material. Looking at Fig. 1, first Ω_1 is activated during a certain period with the laser on its top surface. After some period of time, Ω_2 is activated at a given temperature, the laser moves on its top surface and the process continues in the direction given by the gray arrow. This simplified model is far from an ideal representation of an AM process, but it is a good benchmark for the potential of the proposed technique. Within the activated domains, let $\Omega \subset \mathbb{R}^3$ be an open subset of boundary $\delta\Omega$. For $t \in I = [0,\tau]$ with $\tau \in \mathbb{R}_*^+$ the simulation length, we respect the following heat equation for the thermal field $T(x,t)$

$$c_p\rho\frac{\partial T}{\partial t}(x,t) - \lambda_c \Delta T(x,t) = 0\,, \tag{1}$$

with c_p the constant specific heat capacity, ρ the density, and λ_c the thermal conductivity. For the sake of simplicity, we are working with a fully linear equation, but note that the PGD methodology is not limited to linear cases.

The boundary $\delta\Omega$ can additionally be split into subsets,

$$\delta\Omega = \delta\Omega_D \cup \delta\Omega_s \cup \delta\Omega_h\,,$$

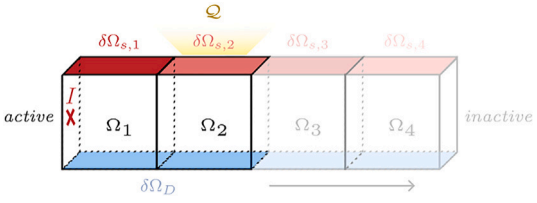


Fig. 1. An example with 4 mesh divisions and 3 sequential activations at time $\{t_2,t_3,t_4\}$, currently at time $t_2 \le t < t_3$, Ω_1 and Ω_2 are active while Ω_3 and Ω_4 are inactive.

with no intersection between each subset. $\delta\Omega_D$ denotes the subset on which Dirichlet boundary conditions have been imposed, $\delta\Omega_h$ corresponds to homogeneous Neumann boundary conditions and $\delta\Omega_s$ is the boundary on which we have the heat source or, in other words, the path of the laser. Therefore, the boundary conditions are:

$$\begin{aligned} &T(x,t) = T_d \quad \forall x \in \delta\Omega_D \ \forall t \in I\,, \\ &\frac{\partial T}{\partial n}(x,t) = 0 \quad \forall x \in \delta\Omega_h \ \forall t \in I\,, \\ &\lambda_c \frac{\partial T}{\partial n}(x,t) = S(x,t) \quad \forall x \in \delta\Omega_s \ \forall t \in I\,, \end{aligned} \tag{2}$$

with T_d the fixed Dirichlet boundary condition temperature, $\frac{\partial \bullet}{\partial n}$ the gradient operator relative to the outgoing surface normal vector, and $S(x,t)$ the surface heat source.
Let $N_d \in \mathbb{N}^*$ be the number of activated domains, let $\{\Omega_j\}_{j=1}^{N_d}$ be the non-overlapping, sequentially activated, partition of Ω. An activation partition can also be made on the time interval I, corresponding to

$$\forall j \in [\![1, N_d]\!],\ I_j = [t_j, t_{j+1}] \text{ with } t_1 = 0 < t_2 < \cdots < t_{N_d} < t_{N_d+1} = \tau. \tag{3}$$

Regarding the heat source term, we will use a constant heat flux term only on the recently activated subdivision of $\delta\Omega_s$, representing the progression of a laser

$$\begin{cases} S(x,t) = \mathcal{Q} & \forall x \in \delta\Omega_{s,j} \\ S(x,t) = 0 & \forall k \in \mathbb{N}^* | k < j \ \ \forall x \in \delta\Omega_{s,k} \end{cases} \quad \forall j \in [\![1, N_d]\!] \ \ \forall t \in I_j\,, \tag{4}$$

with $\delta\Omega_{s,j} = \delta\Omega_s \cap \Omega_j$. Note that it is not required to define a heat source on the domain that has not been activated yet. An example of the activation procedure is given in Fig. 1.
To initialize the problem correctly, at each activation and at $t = t_1 = 0$, the new material will be considered at a fixed temperature T_{new}

$$T(x,t_j) = T_{new} \quad \forall j \in [\![1, N_d]\!] \quad \forall x \in \Omega_j \tag{5}$$

Non-homogeneous activation procedures will be discussed later in Section 3.2. The Dirichlet boundary conditions Eq. (2) along with the activation temperatures Eq. (5) are combined in a time-constant lifting term defined on the entire domain Ω that we call $T_L(x)$, verifying

$$\begin{cases} T_L(x) = T_d & \forall x \in \delta\Omega_D \\ T_L(x) = T_{new} & \forall j \in [\![1, N_d]\!] \quad \forall x \in \Omega_j. \end{cases} \tag{6}$$

2.2. Finite element formulation

Eq. (1) is first multiplied by a virtual thermal field θ and integrated over a domain Ω to find the associated weak form. We search $T \in \mathcal{U} = L^2(I, H^1\{\Omega\}) = L^2(I, \mathcal{V})$ such that,

$$\forall \theta \in \mathcal{U}_0,\ c_p\rho \int_\Omega \frac{\partial T}{\partial t}\theta dV + \lambda_c \int_\Omega \boldsymbol{\nabla} T.\boldsymbol{\nabla}\theta dV = \int_{\delta\Omega_s} S(x,t)\theta dS. \tag{7}$$

Note that the free surface contribution on $\delta\Omega_h$, contrary to the term on $\delta\Omega_s$, vanishes in the weak form. For all functional spaces, we will use the subscript $\bullet_0$ for their associated homogeneous space, in which the functions are enforced to zero along the Dirichlet boundary conditions. Their respective finite-dimensional space-discretized subspaces will be written with straight letters (e.g., $\mathbb{U}$ for $\mathcal{U}$). Furthermore, the superscript $\bullet^{(k)}$ is used for any space-discretized tensor or function, of space dimension n_k, relevant to the time interval I_k. Introducing the finite element discretization, for every time interval I_k, we have $T \in \mathbb{U} \subset \mathcal{U}$ defined as:

$$T(x,t) = \{\boldsymbol{N}^{(k)}(x)\}^T \cdot \boldsymbol{T}^{(k)}(t) \quad \forall t \in I_k \quad \forall x \in \bigcup_{i=1}^{k} \Omega_i. \tag{8}$$

As the spatial domain evolves, both the shape function arrays $\boldsymbol{N}^{(k)} \in \mathbb{V}^{n_k}$, containing the finite element basis functions that span $\mathbb{V}$, and the nodal temperature arrays $\forall t \in I_k, \boldsymbol{T}^{(k)}(t) \in \mathbb{R}^{n_k}$, have non constant sizes. Note that for the FE operators, contrary to Ω_k, $\bullet^{(k)}$ refers to all the domains activated up to time t_k. With the addition of the lifting term Eq. (6), the homogenized discrete thermal problem is then to find $\forall k \in [\![1, N_d]\!],\ \forall t \in I_k,\ \boldsymbol{T}^{(k)}(t) \in \mathbb{R}_0^{n_k}$, such that:

$$\forall \boldsymbol{\theta} \in \mathbb{R}_0^{n_k},\ \boldsymbol{\theta}^T \left\{ \boldsymbol{M}^{(k)}\dot{\boldsymbol{T}}^{(k)}(t) + \boldsymbol{K}^{(k)}\boldsymbol{T}^{(k)}(t) - \boldsymbol{Q}^{(k)}(t) \right\} = 0\,, \tag{9}$$

with $\boldsymbol{M}^{(k)} \in \mathbb{R}^{n_k \times n_k}$, $\boldsymbol{K}^{(k)} \in \mathbb{R}^{n_k \times n_k}$ and $\forall t \in I_k,\ \boldsymbol{Q}^{(k)}(t) \in \mathbb{R}^{n_k}$ as:

$$\begin{cases} \boldsymbol{M}^{(k)} = \rho c_p \displaystyle\int_{\bigcup_{i=1}^{k}\Omega_i} \boldsymbol{N}^{(k)}(x)\left\{\boldsymbol{N}^{(k)}(x)\right\}^T dV \\ \boldsymbol{K}^{(k)} = \lambda_c \displaystyle\int_{\bigcup_{i=1}^{k}\Omega_i} \boldsymbol{\nabla}\boldsymbol{N}^{(k)}(x)\boldsymbol{\nabla}\left\{\boldsymbol{N}^{(k)}(x)\right\}^T dV \\ \boldsymbol{Q}^{(k)}(t) = \displaystyle\int_{\bigcup_{i=1}^{k}\delta\Omega_{s,k}} S(x,t)\left\{\boldsymbol{N}^{(k)}(x)\right\}^T dS - \boldsymbol{K}^{(k)}\boldsymbol{T}_L^{(k)}. \end{cases} \tag{10}$$

3. The methodology

In this section, we will extend the standard PGD formulation to tackle evolving domains with interpolation constraints at each activation step. After a succinct presentation of the standard PGD, we will introduce the expanded PGD procedure, which enables the handling of evolving space domains. Following that, we will introduce interpolation constraints at each activation step and combine these with the PGD method using a novel piecewise strategy.

3.1. The standard PGD approach

First, let's introduce some fundamental notations and provide an overview of the key steps in the classical [8,9,41] space-time separated PGD formulation. If we start with a single fixed domain, the approximation of order N for the thermal field is given in the form:

$$\forall (x,t) \in \Omega \times I,\ \ T_N(x,t) = \sum_{i=1}^{N} \Lambda_i(x)\lambda_i(t)\,, \tag{11}$$

with $T_N \in \mathcal{U}_0 = L^2(I, H_0^1\{\Omega\}) = L^2(I, \mathcal{V}_0)$ the approximated thermal field, $\Lambda_i \in \mathcal{V}_0$ the space modes, and $\lambda_i \in \mathcal{I} = L^2(I)$ the time modes.

Using finite elements to discretize space, we are now looking for $\boldsymbol{T}_N(t) \in \mathbb{R}_0^n$, such that:

$$\boldsymbol{T}_N(t) = \sum_{i=1}^{N} \boldsymbol{\Lambda_i}\lambda_i(t).$$

In the PGD algorithm, new pairs of modes are built one by one. Let us now assume that we are currently at step N and have already calculated N-1 space and time modes. We are therefore looking for $\boldsymbol{\Lambda}_N \in \mathbb{R}_0^n$ and $\lambda_N \in \mathcal{I}$ such that:

$$\boldsymbol{T}_N(t) = \boldsymbol{\Lambda}_N\lambda_N(t) + \sum_{i=1}^{N-1} \boldsymbol{\Lambda}_i\lambda_i(t) = \boldsymbol{\Lambda}_N\lambda_N(t) + \boldsymbol{R}_{N-1}(t)\,,$$

with $\boldsymbol{R}_{N-1}(t)$ a known term from the previous PGD iteration. We can introduce this new form of the unknown into the original discrete thermal problem Eq. (9) for any value of $k \in [\![1, N_d]\!]$. We will first imagine that we do not change our discretization over time and remove the k dependency in the next equation. Using the standard Galerkin approach, which is extensively detailed in [8,9,41], we arrive at two interdependent functional equations: a linear system fordetermining the spatial mode $\boldsymbol{\Lambda}_N$ and an ordinary differential equation for determining the time mode λ_N:

$$\begin{cases} \forall \boldsymbol{\Lambda}^* \in \mathbb{R}_0^n & \boldsymbol{\Lambda}^{*T} \left\{ (\int_I \lambda_N \dot{\lambda}_N \boldsymbol{M} dt + \int_I \lambda_N^2 \boldsymbol{K} dt) \boldsymbol{\Lambda}_{\boldsymbol{N}} - \int_I \lambda_N \boldsymbol{Q}_{\boldsymbol{N}}(t) dt \right\} = 0 \\ & (\boldsymbol{\Lambda}_{\boldsymbol{N}}^T \boldsymbol{M} \boldsymbol{\Lambda}_{\boldsymbol{N}}) \dot{\lambda}_N + (\boldsymbol{\Lambda}_{\boldsymbol{N}}^T \boldsymbol{K} \boldsymbol{\Lambda}_{\boldsymbol{N}}) \lambda_N - \boldsymbol{\Lambda}_{\boldsymbol{N}}^T \boldsymbol{Q}_{\boldsymbol{N}}(t) = 0 \,, \end{cases} \tag{12}$$

with $\forall N > 2$:

$$\boldsymbol{Q}_{\boldsymbol{N}}(t) = \boldsymbol{Q}(t) - \boldsymbol{M} \dot{\boldsymbol{R}}_{N-1}(t) - \boldsymbol{K} \boldsymbol{R}_{N-1}(t) \,,$$

the source term containing both the information about the heat input and the previous modes. This system is typically solved using a fixed-point algorithm that iterates between calculating each mode while treating the other as constant until convergence is achieved, as determined by a specified criterion. The ordinary time equation in time is solved by making use of a Backward–Euler finite difference scheme. To ensure a unique pair of modes within the fixed-point loop, we choose to normalize the spatial mode. It is worth noting that all PGD formulations developed within this article are still based on the same mesh as the original FEM problem: ultimately we converge to the FE solution, which remains in general approximate, with an error not addressed in our approach. We will neither study the effect of different mesh discretizations nor different time discretizations on the FE solution as it simply is outside the scope of this paper.

3.2. An expanded PGD approach

We will now introduce an expanded PGD approach to handle problems involving evolving space domains, such as the one defined in Section 2. The goal is to generalize the PGD formulation Eq. (12), which can be adapted for any time interval I_k, to the full time interval I containing activations. The core concept is to incorporate every point that exists throughout the simulation into a new global space discretization on which space modes can be computed. For our evolving AM example, the expanded space is the full domain Ω. We can then compute an expanded PGD solution that provides a solution to every point of Ω at all times. Finally, writing this expanded solution as part of the current physical representation, one can obtain the actual solution. All tensors defined within the expanded space have the same space dimension, n, and are denoted by a subscript $\bullet_e$. The new expanded discretization can be written:

$$\forall k \in [\![1, N_d]\!], \quad \forall t \in I_k, \quad \forall x \in \bigcup_{i=1}^{k} \Omega_i, \quad T(x,t) = \boldsymbol{N}_e^{(k)T}(x).\boldsymbol{T}_e(t) \,, \tag{13}$$

with $\boldsymbol{T}_e(t) \in \mathbb{R}^n$ constant in size over time. The expanded shape functions $\boldsymbol{N}_e^{(k)}(x)$ can be seen as a sparse combination of all of the previously defined $\left\{ \boldsymbol{N}^{(k)}(x) \right\}_{k \in [\![1, N_d]\!]}$:

$$\boldsymbol{N}_e^{(k)}(x) = \begin{matrix} \Omega_1 \\ \vdots \\ \Omega_k \\ \Omega_{k+1} \\ \vdots \\ \Omega_{N_d} \end{matrix} \begin{pmatrix} \\ \boldsymbol{N}^{(k)}(x) \\ \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix}. \tag{14}$$

The blue terms here highlight the sparsity, as we have zeros for the inactive nodes going from Ω_{k+1} to Ω_{N_d}. Integrating the expanded shape functions (14) as we did in (10), we obtain expanded finite element operators (see example in Appendix A). We also get an expanded discrete thermal problem in which we are looking $\forall k \in [\![1, N_d]\!], \forall t \in I_k$, for $\boldsymbol{T}_e(t) \in \mathbb{R}_0^n$ verifying:

$$\forall \boldsymbol{\theta} \in \mathbb{R}_0^n \; \boldsymbol{\theta}^T \{ \boldsymbol{M}_e^{(k)} \dot{\boldsymbol{T}}_e(t) + \boldsymbol{K}_e^{(k)} \boldsymbol{T}_e(t) - \boldsymbol{Q}_e^{(k)}(t) \} = 0 \,, \tag{15}$$

with $\boldsymbol{M}_e^{(k)} \in \mathbb{R}^{n \times n}$, $\boldsymbol{K}_e^{(k)} \in \mathbb{R}^{n \times n}$, and $\boldsymbol{Q}_e^{(k)}(t) \in \mathbb{R}^n$ the expanded version of each respective matrix defined in Section 2.2. System Eq. (15) is not solvable using conventional step-by-step time solvers. However, we should be able to obtain modes and a solution, by integrating it with the PGD framework. The PGD approximation of order N of $\boldsymbol{T}_e(t)$ is:

$$\boldsymbol{T}_{e,N}(t) = \sum_{i=1}^{N} \boldsymbol{\Lambda}_{e,i} \lambda_i(t) \quad \forall k \in [\![1, N_d]\!] \quad \forall t \in I_k. \tag{16}$$

The expanded solution $\boldsymbol{T}_{e,N}$ is then defined at every point of space and time, even when that part of the domain is not yet activated. To retrieve the actual approximation of the thermal field, you project onto the current physical representation with $T_N(x,t) = \boldsymbol{N}_e^{(k)T}(x).\boldsymbol{T}_{e,N}(t)$.

The PGD approach remains the same as Eq. (12) but now with the expanded operators, at iteration N we are looking for the expanded space mode $\boldsymbol{\Lambda}_{e,N}$ and time mode λ_N such that:

$$\begin{cases} \forall \boldsymbol{\Lambda}_e^* \in \mathbb{R}_0^n & \boldsymbol{\Lambda}_e^{*T} \sum_{k=1}^{N_d} \left\{ \left[\boldsymbol{M}_e^{(k)} \int_{I_k} \lambda_N \dot{\lambda}_N dt + \boldsymbol{K}_e^{(k)} \int_{I_k} \lambda_N^2 dt \right] \boldsymbol{\Lambda}_{e,N} \right. \\ & \left. - \int_{I_k} \lambda_N \boldsymbol{Q}_{e,N}^{(k)}(t) dt \right\} = 0 \\ \forall k \in [\![1, N_d]\!], \forall t \in I_k & (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{M}_e^{(k)} \boldsymbol{\Lambda}_{e,N}) \dot{\lambda}_N(t) + (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{K}_e^{(k)} \boldsymbol{\Lambda}_{e,N}) \lambda_N(t) \\ & - \boldsymbol{\Lambda}_{e,N}^T \boldsymbol{Q}_{e,N}^{(k)}(t) = 0 \,, \end{cases} \tag{17}$$

with $\forall N > 2$:

$$\boldsymbol{Q}_{e,N}^{(k)}(t) = \boldsymbol{Q}_e^{(k)}(t) - \boldsymbol{M}_e^{(k)} \dot{\boldsymbol{R}}_{e,N-1}(t) - \boldsymbol{K}_e^{(k)} \boldsymbol{R}_{e,N-1}(t).$$

Practically, while we will be able to compute modes and a solution with the formulation shown in Eq. (17), we are missing the proper initializations that will ensure the convergence to the correct solution. Indeed, if all new points are added with a fixed temperature T_{new} – see Eq. (5) – we require, at convergence, to verify

$$\forall k \in [\![1, N_d]\!] \quad \{ \boldsymbol{T}_e(t_k^+) \} |_{\#k} = 0. \tag{18}$$

where $\bullet|_{\#k}$ designates the nodes added at time t_k and contained within Ω_k. This should be verified, and the solution should remain continuous for the rest of the already existing nodes.

One can go further and generalize the discrete constraints Eq. (18) to more complex activation scenarios. In fact, in many physical phenomena that contain domain extensions, the values given to the new nodes depend linearly on the values of the already existing ones. For example, imposing the conservation of a chosen quantity over our space domain at the activation instances can lead to linear constraints. This is why we will generalize the constraints Eq. (18) to include all linear dependencies and instead write

$$\forall k \in [\![1, N_d]\!] \quad \boldsymbol{T}_e(t_k^+) = \boldsymbol{\mathcal{H}}_k \boldsymbol{T}_e(t_k^-). \tag{19}$$

with $\boldsymbol{\mathcal{H}}_k \in \mathbb{R}^{n \times n}$ a general interpolation matrix associated with activation k. The construction of linear interpolation operators $\boldsymbol{\mathcal{H}}_k$ based on the conservation of internal energy is presented in Appendix B. In the rest of the manuscript, we refer to the general expression (19) and use this aforementioned conservation of internal energy.

Looking back at our original PGD formulation (17), the only way to verify Eq. (19) for every mode number N and at every activation instance is to ensure that all of the time modes λ_i are equal to zero at these specific instances. As there is currently no mechanism to ensure this, new nodes will simply not appear with the proper temperature. To comply with these new discrete constraints, in the upcoming Section 3.3, we will introduce a novel piecewise approach that will deviate from (17).

3.3. The piecewise PGD approach

Imposing the time-discrete constraints Eq. (19), with any number of modes, is done via the lifting term, i.e., when the PGD time modes are equal to zero. Therefore, we will now calculate the time modes in a piecewise manner allowing for a reset at the activation instances. Let us imagine we have built $N_d - 1$ interpolation matrices $\{\boldsymbol{\mathcal{H}}_k\}_{k\in[[2,N_d]]}$, one for each activation, and derive the corresponding PGD formulation. We are looking for $\{\boldsymbol{\Lambda}_{e,i}\}_{i\in[[1,N]]} \in \mathbb{R}_0^{n\times N}$ and $\{\lambda_i^{(1)}(t),\dots,\lambda_i^{(N_d)}(t)\}_{i\in[[1,N]]} \in \mathcal{I}_{I_1}^N \times \cdots \times \mathcal{I}_{I_{N_d}}^N$ such that the approximate homogenized temperature field of order N is written as:

$$\forall t \in I \quad \boldsymbol{T}_e^N(t) = \sum_{i=1}^{N}\sum_{k=1}^{N_d} \mathbb{1}_{t\in I_k}(t)\left[\lambda_i^{(k)}(t) + \boldsymbol{\mathcal{W}}_i^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})\right]\boldsymbol{\Lambda}_{e,i}. \tag{20}$$

with the indicator function $\mathbb{1}_{t\in I_k}(t) = 1$ if $t \in I_k$, 0 if $t \notin I_k$ and where by $\{t_j\}_{j>1}$ we will always designate the activation instances defined in (3). The $\boldsymbol{\mathcal{W}}_i^{(k)} \in \mathbb{R}^{n\times n}$ are defined as:

$$\forall i \in [\![1,N]\!] \quad \begin{cases} \boldsymbol{\mathcal{W}}_i^{(1)} = 0 \\ \forall k \in [\![2,N_d]\!] \quad \boldsymbol{\mathcal{W}}_i^{(k)} = \boldsymbol{\mathcal{H}}_k\left[\lambda_i^{(k-1)}(t_k) + \boldsymbol{\mathcal{W}}_i^{(k-1)}\right]. \end{cases} \tag{21}$$

We also have an extension of the time modes such that:

$$\forall i \in [\![1,N]\!] \quad \forall k \in [\![1,N_d]\!] \quad \lambda_i^{(k)}(t_k) = 0.$$

This new formulation allows for the constraint (19) to be verified for all the activation times $\{t_k\}_{k\in[[2,N_d]]}$ and with any number of modes:

$$\begin{aligned} \boldsymbol{T}_e^N(t_k^+) &= \sum_{i=1}^{N}\left[\lambda_i^{(k)}(t_k) + \boldsymbol{\mathcal{W}}_i^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})\right]\boldsymbol{\Lambda}_{e,i} \\ &= \sum_{i=1}^{N}\boldsymbol{\mathcal{W}}_i^{(k)}\boldsymbol{\Lambda}_{e,i} \\ &= \sum_{i=1}^{N}\boldsymbol{\mathcal{H}}_k[\lambda_i^{(k-1)}(t_k) + \boldsymbol{\mathcal{W}}_i^{(k-1)}]\boldsymbol{\Lambda}_{e,i} \\ &= \boldsymbol{\mathcal{H}}_k\boldsymbol{T}_e^N(t_k^-). \end{aligned} \tag{22}$$

And the solution is continuous for every node within its activation interval. Let us now assume we have already calculated $N-1$ expanded space modes and $\{N-1\}\times N_d$ time modes, we are looking for $\boldsymbol{\Lambda}_{e,N} \in \mathbb{R}_0^n$ and $\{\lambda_N^{(1)}(t),\dots,\lambda_N^{(N_d)}(t)\} \in \mathcal{I}_{I_1}\times\cdots\times\mathcal{I}_{I_{N_d}}$ such that:

$$\begin{aligned} \boldsymbol{T}_e^N(t) &= \sum_{i=1}^{N}\sum_{k=1}^{N_d}\mathbb{1}_{t\in I_k}(t)\left[\lambda_i^{(k)}(t) + \boldsymbol{\mathcal{W}}_i^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})\right]\boldsymbol{\Lambda}_{e,i} \\ &= \sum_{k=1}^{N_d}\mathbb{1}_{t\in I_k}(t)\left[\lambda_N^{(k)}(t) + \boldsymbol{\mathcal{W}}_N^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})\right]\boldsymbol{\Lambda}_{e,N} + \sum_{k=1}^{N_d}\mathbb{1}_{t\in I_k}(t)\boldsymbol{R}_{e,N-1}^{(k)}(t) \end{aligned} \tag{23}$$

Introducing the virtual temperature field $\boldsymbol{T}_e^*(t)$ for the Galerkin formulation:

$$\begin{aligned} &\forall k \in [\![1,N_d]\!] \ \forall \lambda^{*(k)}(t) \in \mathcal{I}_{I_k} \ \forall \Lambda_e^* \in \mathbb{V}_{e,0} \ \boldsymbol{T}_e^*(t) \\ &= \sum_{k=1}^{N_d}\mathbb{1}_{t\in I_k}(t)\left[\lambda_N^{(k)}(t) + \boldsymbol{\mathcal{W}}_N^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})\right]\boldsymbol{\Lambda}_e^* \\ &\quad + \sum_{k=1}^{N_d}\mathbb{1}_{t\in I_k}(t)\left[\lambda^{*(k)}(t)\right]\boldsymbol{\Lambda}_{e,N}\ . \end{aligned} \tag{24}$$

Let us recombine it with the discretized PDE (15), we are looking for $\boldsymbol{\Lambda}_{e,N} \in \mathbb{R}_0^n$ and $\{\lambda_N^{(1)}(t),\dots,\lambda_N^{(N_d)}(t)\} \in \mathcal{I}_{I_1}\times\cdots\times\mathcal{I}_{I_{N_d}}$ such that:

$$\forall k \in [\![1,N_d]\!] \ \forall \lambda^{*(k)}(t) \in \mathcal{I}_{I_k}, \ \forall \Lambda_e^* \in \mathbb{R}_0^n$$

$$\sum_{k=1}^{N_d}\int_{I_k}\boldsymbol{T}_e^{*T}(t)\left\{\boldsymbol{M}_e^{(k)}\dot{\boldsymbol{T}}_e^N(t) + \boldsymbol{K}_e^{(k)}\boldsymbol{T}_e^N(t) - \boldsymbol{Q}_{e,N}^{(k)}(t)\right\}dt = 0\ , \tag{25}$$

with for $N > 2$:

$$\boldsymbol{Q}_{e,N}^{(k)}(t) = \boldsymbol{Q}_e^{(k)}(t) - \boldsymbol{M}_e^{(k)}\dot{\boldsymbol{R}}_{e,N-1}^{(k)}(t) - \boldsymbol{K}_e^{(k)}\boldsymbol{R}_{e,N-1}^{(k)}(t). \tag{26}$$

After carrying out the calculations and isolating both the space and time systems, we are looking for $\boldsymbol{\Lambda}_{e,N} \in \mathbb{R}_0^n$, $\{\lambda_N^{(1)}(t),\dots,\lambda_N^{(N_d)}(t)\} \in \mathcal{I}_{I_1}\times\cdots\times\mathcal{I}_{I_{N_d}}$ that verify:

$$\begin{cases} \forall \Lambda_e^* \in \mathbb{R}_0^n \ \boldsymbol{\Lambda}_e^{*T}\left\{\sum_{k=1}^{N_d}\left[\boldsymbol{M}_e^{(k)}\int_{I_k}\lambda_N^{(k)}\dot{\lambda}_N^{(k)}dt + \boldsymbol{K}_e^{(k)}\int_{I_k}(\lambda_N^{(k)})^2 dt\right.\right. \\ \qquad + [\boldsymbol{\mathcal{W}}_N^{(k)}]^T\boldsymbol{M}_e^{(k)}\int_{I_k}\dot{\lambda}_N^{(k)}dt + [\boldsymbol{\mathcal{W}}_N^{(k)}]^T\boldsymbol{K}_e^{(k)}\int_{I_k}\lambda_N^{(k)}dt \\ \qquad \left.+\boldsymbol{K}_e^{(k)}\boldsymbol{\mathcal{W}}_N^{(k)}\int_{I_k}\lambda_N^{(k)}dt + [\boldsymbol{\mathcal{W}}_N^{(k)}]^T\boldsymbol{K}_e^{(k)}\boldsymbol{\mathcal{W}}_N^{(k)}(t_{k+1}-t_k)\right]\boldsymbol{\Lambda}_{e,N} \\ \qquad \left.-\int_{I_k}\lambda_N^{(k)}\boldsymbol{Q}_{e,N}^{(k)}(t)dt - [\boldsymbol{\mathcal{W}}_N^{(k)}]^T\int_{I_k}\boldsymbol{Q}_{e,N}^{(k)}(t)dt\right\} = 0 \\ \forall k \in [[1,N_d]] \ \forall t \in I_k \ (\boldsymbol{\Lambda}_{e,N}^T\boldsymbol{M}_e^{(k)}\boldsymbol{\Lambda}_{e,N})\dot{\lambda}_N^{(k)}(t) + (\boldsymbol{\Lambda}_{e,N}^T\boldsymbol{K}_e^{(k)}\boldsymbol{\Lambda}_{e,N})\lambda_N^{(k)}(t) \\ \qquad - \boldsymbol{\Lambda}_{e,N}^T\boldsymbol{Q}_{e,N}^{(k)}(t) + \boldsymbol{\Lambda}_{e,N}^T\boldsymbol{K}_e^{(k)}[\boldsymbol{\mathcal{W}}_N^{(k)}(\{t_j\}_{\substack{j\le k\\ j>1}})]\boldsymbol{\Lambda}_{e,N} = 0. \end{cases} \tag{27}$$

We end up with a linear system for the space mode and N_d ordinary differential equations that can be solved one after the other for the time modes, updating $\boldsymbol{\mathcal{W}}_N^{(k)}$ after each activation. While the assembly cost is higher as the matrices are more complex, solving the space system only costs as much as the largest space system in the time-step approach. In the space system, the $\boldsymbol{\mathcal{W}}_N^{(k)}$ terms are dependent on $\{\lambda_N^{(j-1)}(t_j)\}_{\substack{j\le k\\ j>1}}$ and they can be further developed. We demonstrate this for the case $N_d = 2$ in Section 3.4. Another important consequence of the evolving shape functions, is that the normalization of the space modes required to converge to a unique set of space and time modes inside the fixed-point algorithm becomes non-trivial as the modes exist in the extended space that lacks a physical representation. We observed that computationally any of the standard $||\boldsymbol{\Lambda}_e||_{\boldsymbol{k}} = ||\boldsymbol{\Lambda}_e||_{L^2(\bigcup_{i=1}^k \Omega_i)}$ norms can be used and all lead to convergence. The individual norm choice will be discussed in the results section.

Everything can now be summarized in Algorithm 1. It allows space-time separated PGD to work with any evolving space domain containing linear interpolation at the activation instances, at the expense of a more expensive space system. Note the presence of *tol* and $N_{\bar{t}}$, respectively, the fixed-point tolerance and the fixed-point iteration threshold. Together they manage the iterations within the fixed-point loop; their impact will be discussed in Section 4.

3.4. A developed example for $N_d = 2$

Let us look at our example with $N_d = 2$, also presented in Appendix A, and develop the $\boldsymbol{\mathcal{W}}_N^{(k)}$ terms inside the integrals. The approximated thermal field of order N (20) becomes $\forall t \in I$:

$$\boldsymbol{T}_e^N(t) = \sum_{i=1}^{N}\left\{\mathbb{1}_{t\in I_1}(t)\lambda_i^{(1)}(t)\boldsymbol{\Lambda}_{e,i} + \mathbb{1}_{t\in I_2}(t)[\lambda_i^{(2)}(t) + \lambda_i^{(1)}(t_2)\boldsymbol{\mathcal{H}}_2]\boldsymbol{\Lambda}_{e,i}\right\}. \tag{28}$$

At step N we are looking for $\boldsymbol{\Lambda}_{e,N} \in \mathbb{R}_0^n$, $\lambda_N^{(1)}(t) \in \mathcal{I}_{I_1}$ and $\lambda_N^{(2)}(t) \in \mathcal{I}_{I_2}$ such that:

Algorithm 1 space-time separated expanded piecewise PGD algorithm.

Require: $\boldsymbol{M}_e(t)$,$\boldsymbol{K}_e(t)$, $\boldsymbol{Q}_e(t)$, $\{\boldsymbol{\mathcal{H}}_a\}_{a\in[[2,N_d]]}$ (15) and (B.14)
  for $i = 1$ to N_{max} **do**
    $k = 0$
    Initialize current time modes $\{\lambda_i^{(1)},\dots,\lambda_i^{(N_d)}\}^0$
    Calculate $\{\boldsymbol{\mathcal{W}}_i^{(a)}\}_{a\in[[1,N_d]]}$ with $\{\lambda_i^{(1)},\dots,\lambda_i^{(N_d)}\}^0$ (21)
    Assemble the piecewise $\{\lambda_i^{(1)},\dots,\lambda_i^{(N_d)}\}^0$ into one continuous $\lambda_i^{C,0}$ defined on all I
    Update $\boldsymbol{Q}_{e,i}(t)$ (23) and (26)
    while $\epsilon > tol$ or $k < N_{\bar{t}}$ **do**
      $k = k + 1$
      Space Linear system: Compute $\boldsymbol{\Lambda}_{e,i}^k$ with $\{\lambda_i^{(1)},\dots,\lambda_i^{(N_d)}\}^{k-1}$ and $\{\boldsymbol{\mathcal{W}}_i^{(a)}\}_{a\in[[1,N_d]]}$ (27)
      Normalize $\boldsymbol{\Lambda}_{e,i}^k$
      $\boldsymbol{\mathcal{W}}_i^{(1)} = 0$
      for $j = 1$ to N_d **do**
        Time ODE: Compute $\lambda_i^{(j),k}$ with $\boldsymbol{\Lambda}_{e,i}^k$ and $\boldsymbol{\mathcal{W}}_i^{(j)}$ (27)
        Calculate $\boldsymbol{\mathcal{W}}_i^{(j)}$ with $\boldsymbol{\mathcal{W}}_i^{(j-1)}$ and $\lambda_i^{(j-1),k}(t_j)$ (21)
      end for
      Assemble the piecewise $\{\lambda_i^{(1)},\dots,\lambda_i^{(N_d)}\}^k$ into one continuous $\lambda_i^{C,k}$ defined on all I
      Compute $\epsilon = \frac{2*\int_I \lambda^{C,k}(t)-\lambda^{C,k-1}(t)dt}{\int_I \lambda^{C,k}(t)+\lambda^{C,k-1}(t)dt}$
    end while
  end for

$$
\begin{cases}
\forall \Lambda_e^* \in \mathbb{R}_0^n \;\; \boldsymbol{\Lambda}_e^{*T} \Big\{ \Big[\boldsymbol{M}_e^{(1)} \int_{I_1} \lambda_N^{(1)} \dot{\lambda}_N^{(1)} dt + \boldsymbol{K}_e^{(1)} \int_{I_1} (\lambda_N^{(1)})^2 dt + \boldsymbol{M}_e^{(2)} \int_{I_2} \lambda_N^{(2)} \dot{\lambda}_N^{(2)} dt \\
\qquad + \boldsymbol{K}_e^{(2)} \int_{I_2} (\lambda_N^{(2)})^2 dt + \lambda_N^{(1)}(t_2) \boldsymbol{\mathcal{H}}_2^T \boldsymbol{M}_e^{(2)} \int_{I_2} \dot{\lambda}_N^{(2)} dt \\
\qquad + \lambda_N^{(1)}(t_2) \boldsymbol{\mathcal{H}}_2^T \boldsymbol{K}_e^{(2)} \int_{I_2} \lambda_N^{(2)} dt + \lambda_N^{(1)}(t_2) \boldsymbol{K}_e^{(2)} \boldsymbol{\mathcal{H}}_2 \int_{I_2} \lambda_N^{(2)} dt \\
\qquad + (\lambda_N^{(1)}(t_2))^2 (T - t_2) \boldsymbol{\mathcal{H}}_2^T \boldsymbol{K}_e^{(2)} \boldsymbol{\mathcal{H}}_2 \Big] \boldsymbol{\Lambda}_{e,N} \\
\qquad - \int_{I_1} \lambda_N^{(1)} \boldsymbol{Q}_{e,N}^{(1)}(t) dt - \int_{I_2} \lambda_N^{(2)} \boldsymbol{Q}_{e,N}^{(2)}(t) dt \\
\qquad - \lambda_N^{(1)}(t_2) \boldsymbol{\mathcal{H}}_2^T \int_{I_2} \boldsymbol{Q}_{e,N}^{(2)}(t) dt \Big\} = 0 \\
\forall t \in I_1 \quad (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{M}_e^{(1)} \boldsymbol{\Lambda}_{e,N}) \dot{\lambda}_N^{(1)}(t) + (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{K}_e^{(1)} \boldsymbol{\Lambda}_{e,N}) \lambda_N^{(1)}(t) \\
\qquad - \boldsymbol{\Lambda}_{e,N}^T \boldsymbol{Q}_{e,N}^{(1)}(t) = 0 \\
\forall t \in I_2 \quad (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{M}_e^{(2)} \boldsymbol{\Lambda}_{e,N}) \dot{\lambda}_N^{(2)}(t) + (\boldsymbol{\Lambda}_{e,N}^T \boldsymbol{K}_e^{(2)} \boldsymbol{\Lambda}_{e,N}) \lambda_N^{(2)}(t) \\
\qquad - \boldsymbol{\Lambda}_{e,N}^T \boldsymbol{Q}_{e,N}^{(2)}(t) + \lambda_N^{(1)}(t_2) \boldsymbol{\Lambda}_{e,N}^T \boldsymbol{K}_e^{(2)} \boldsymbol{\mathcal{H}}_2 \boldsymbol{\Lambda}_{e,N} = 0 \,.
\end{cases}
\tag{29}
$$

We retrieve finite element operators for our space system that can be calculated before any PGD iterations. Ideally, this should always be done as it is computationally very beneficial. Doing this decomposition for the general case with N_d activations, we end up with $\sum_{k=1}^{N_d}(2\,k+k^2) = \mathrm{O}(N_d^3)$ operators to precalculate. This exponential increase in the number of operators due to the nature of the Galerkin formulation could be a computational burden for the general case (see Section 4.6).

4. Numerical results

In this section, the numerical results of the method discussed in the previous part are shown, comparing the PGD solver with classical time-stepping FEM. A 1D preliminary example is presented with a single activation corresponding to the refinement of the mesh at a particular time instance. We will use it to discuss the normalization procedure and the possible use of an update phase to accelerate convergence. Then we will proceed to our full additive manufacturing example, which we introduced earlier. The PGD results will first be laid out for a domain with 4 divisions, the effect of the mean energy constraints will also be shown, a comparison between the different possible partitions will then be made, and finally, the potential computational speedup of the method will be discussed.

4.1. Preliminary 1D example and the normalization choice

To confirm the convergence and to assess the effect of the norm choice, we first look at the case of a heated 1D bar. This example will not have any addition of material but will nonetheless have a time-evolving discretization. We study (1), with $\Omega = [0, L]$, $I = [0, \tau]$, $L = 1$ m, $\tau = 2$ s, and $\frac{\lambda_c}{\rho c_p} = 0.01$ m^{2}s^{-1}. We have no heat source term but two time-dependent Dirichlet boundary conditions:

$$\forall t \in I,\; T(0,t) = \sin\left(\frac{3\pi t}{2}\right) \qquad T(L,t) = \sin(\pi t).$$

We use linear finite elements of equal length in space, starting with a coarse 20-node mesh $\forall t \in [0, \frac{\tau}{2}]$ and refining this mesh, doubling its number of elements to 39 nodes $\forall t \in [\frac{\tau}{2}, \tau]$. We will calculate PGD modes on this time-evolving discretization with the exact algorithm detailed in Section 3.4. The interpolation matrix $\boldsymbol{\mathcal{H}}_2$ is chosen so that the new nodes are exactly interpolated between two already existing nodes. The lifting term will be chosen as the linear interpolation between the two time-dependent Dirichlet boundary conditions. Fig. 2 shows the convergence of the algorithm using the following norm:

$$\mathcal{E}_{PGD} = \sqrt{\frac{\int_0^{\tau/2} ||\boldsymbol{T}_{PGD} - \boldsymbol{T}_{FEM}||^2_{\boldsymbol{M}_c} + \int_{\tau/2}^{\tau} ||\boldsymbol{T}_{PGD} - \boldsymbol{T}_{FEM}||^2_{\boldsymbol{M}_f}}{\int_0^{\tau/2} ||\boldsymbol{T}_{FEM}||^2_{\boldsymbol{M}_c} + \int_{\tau/2}^{\tau} ||\boldsymbol{T}_{FEM}||^2_{\boldsymbol{M}_f}}},$$

where $\boldsymbol{M}_c$ and $\boldsymbol{M}_f$ are the mass matrices of the coarse and fine mesh, respectively. The error is calculated with respect to a reference FEM solution with 1000 nodes and Fig. 3 presents the space and time modes. Note that the space modes are shown on the fine mesh but the coarse mesh is indicated with small crosses. Likewise, the piecewise time modes $\lambda_N^{(k)}$ are here assembled into one continuous mode.

First of all, we can observe that our method successfully converges for all configurations. Importantly, the norm taken for the normalization procedure in our PGD algorithm does not seem to significantly affect the convergence, as both the coarse mesh norm $||\cdot||_c$ and the fine mesh norm $||\cdot||_f$ allow for quasi-monotonous convergence. An update phase was also implemented, which is a common addition to the PGD procedure [9] and involves recalculating the time modes while keeping the space modes unchanged, aiming to approximate the real solution better. It is typically done after the calculation of every new space-time mode pair. The update phase works as it should and significantly improves convergence.

While the norm choice does not directly affect convergence, it has an effect on the visualization of the modes, as seen in Fig. 3. By choosing $||\cdot||_c$, the amplitude of the time modes decreases monotonously as it converges, but the space modes can explode in value for higher mode numbers. On the contrary, had we chosen $||\cdot||_f$, the space modes would have remained reasonable, but the time modes would have been completely out of scale and exploded in value for higher mode numbers. Note that while the condition number of the main system (27) is not affected by the norm choice, the conditioning of the update phase is. Choosing a specific norm will cause the condition number of the opposing update phase to blow up near convergence. In here, it only becomes a problem when we have already reached convergence and $\mathcal{E}_{PGD} \approx 10^{-4}$, it is therefore mostly harmless. However, we confirmed that this conditioning problem worsens with more mesh activations; we lose the benefits of the update phase entirely for higher N_d. With this in mind, we present the following example containing up to $N_d = 16$ activations without an update phase. A mode-dependent norm will also be used to avoid extreme values in the visualization of the modes.

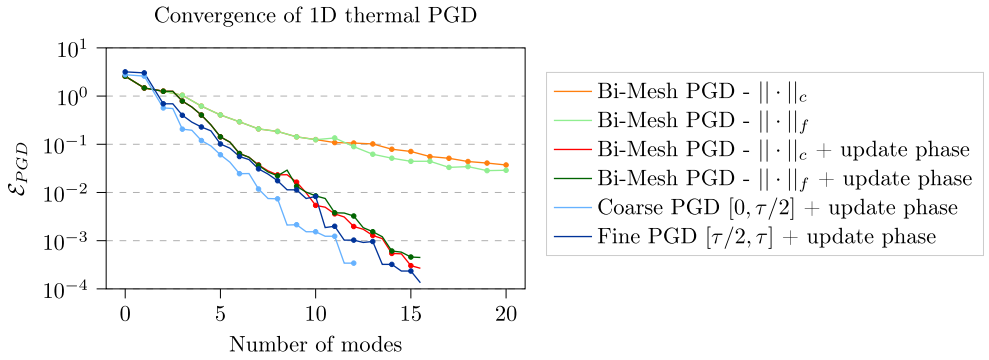


Fig. 2. PGD convergence error with respect to the reference FEM for our 1D example.

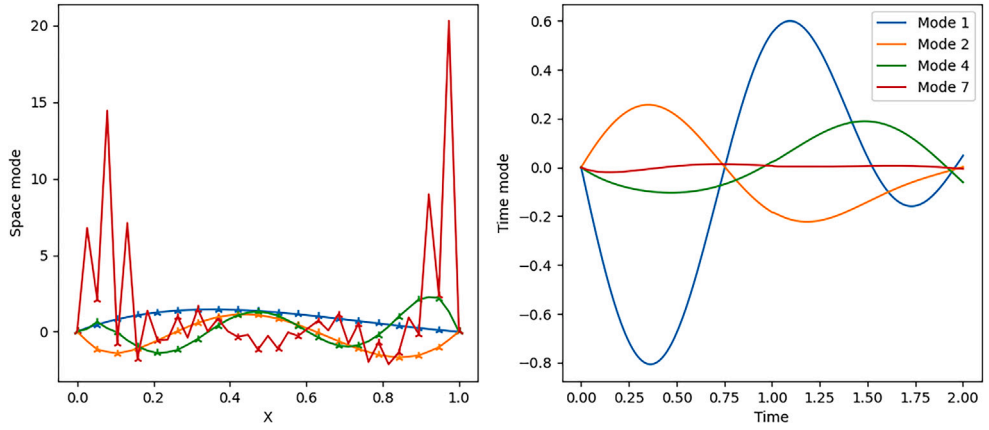


Fig. 3. Space and time modes for the 1D PGD solve, orthonormalized with $||\cdot||_c$.

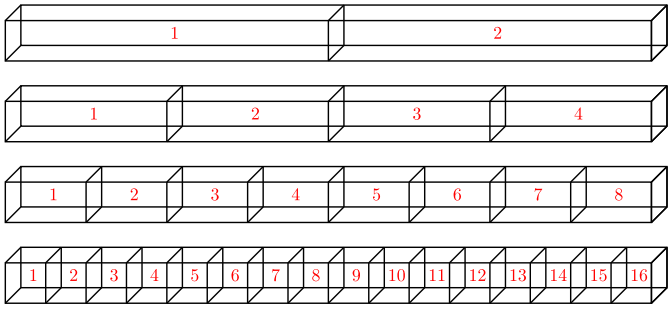


Fig. 4. The domain can be partitioned into 4 different configurations.

Table 1
Simulation parameters.

Symbol	Parameter	Value
a_1	Sample length (mm)	160
a_2	Sample width (mm)	10
a_3	Sample height (mm)	10
P	Total laser power (W)	10
N_d	Number of domains	$\{2, 4, 8, 16\}$
Q	Thermal heat source (W m^{-2})	$\frac{P}{N_d a_2 a_3}$
λ_c	Thermal conductivity (Wm^{-1}K^{-1})	43
ρ	Mass density (kg m^{-3})	7850
c_p	Specific heat (J kg^{-1}K^{-1})	500
τ_c	Thermal characteristic time (s)	$\frac{(a_3)^2 \rho c_p}{\lambda_c} \approx 9$
τ	Simulation length (s)	10
n_t	Number of time steps	100
T_{new}	Activation temperature (°C)	0
T_d	Dirichlet boundary condition (°C)	0

4.2. The additive manufacturing example

Let us now go back to our additive manufacturing example. All simulations were done in Matlab, and the mesh was generated with Gmsh [42]. To correctly compare the results, we chose to study multiple activation partitions of a fixed geometrical and time domain. The geometrical domain was chosen such that it can be divided into 4 different configurations with activated domains of equal size (see Fig. 4). The time partitions were made equal with every configuration. For the fully activated mesh we constructed a single mesh, as explained at the end of Section 3.1, and we are using 3048, constant-size, linear tetrahedral elements. This adds up to a total of 1004 nodes and roughly 190 elements per potential activation domain. When generating the mesh, it is important to ensure that the mesh is conformal at every possible activation interface. The total power of the input laser is constant with every configuration in order to simplify the energy comparisons. All the simulation parameters can be found in Table 1.

4.3. PGD convergence with 4 domains

First, we look at a case with $N_d = 4$ and explore the usual PGD considerations. We will then compare the result with those obtained using

an implicit step-by-step direct FEM solver, using the following norm:

$$\mathcal{E}_{PGD} = \frac{||\boldsymbol{T}_{PGD} - \boldsymbol{T}_{FEM}||_{L^2\left(\bigcup_{k=1}^{N_d}\{I_k \times \bigcup_{i=1}^{k}\Omega_i\}\right)}}{||\boldsymbol{T}_{FEM}||_{L^2\left(\bigcup_{k=1}^{N_d}\{I_k \times \bigcup_{i=1}^{k}\Omega_i\}\right)}} .$$

Figure Fig. 5a confirms that we converge to the reference solution with this method, as the error decreases quasi-monotonously with the number of modes. One can also observe single-point convergence of the thermal field in Fig.5b. Similar to what we noticed in our previous example, the chosen norm doesn't have much of an effect on convergence but has an important effect on the modal amplitudes. Indeed, we observed that using $||\boldsymbol{\Lambda}_e||_4 = ||\boldsymbol{\Lambda}_e||_{L^2(\bigcup_{i=1}^{4}\Omega_i)}$ allows for a correct comparison of the space modes; however, the amplitudes of the $\{\lambda_i^{(k)}\}_{\{\forall k<4\}}$ functions scale to infinite values for large mode numbers i. Conversely, using $||\boldsymbol{\Lambda}_e||_1 = ||\boldsymbol{\Lambda}_e||_{L^2(\Omega_1)}$ restores the monotonous convergence in all of the time modes but makes the space mode scale to infinite values within $\{\Omega_k\}_{k>1}$. As a means to properly visualize the modes in similar scales for both Figs. 6 and 7, we arbitrarily decided to use a mode-dependent space norm that is a weighted sum of all possible norms:

$$||\boldsymbol{\Lambda}_e||^2(\lambda) = \left[\sum_{k=1}^{N_d} ||\boldsymbol{\Lambda}_e||_k^2 ||\lambda^{(k)}||^2_{L^2(I_k)}\right] / \left[\sum_{k=1}^{N_d} ||\lambda^{(k)}||^2_{L^2(I_k)}\right] .$$

Note that this also means we can neither compare the space mode amplitudes nor observe the convergence of the space mode on those graphs.

Table 2 compares the PGD accuracy for various mode numbers N and various numbers of iterations per mode within the fixed point loop (controlled by tol and $N_{\bar{t}}$). First, we can see that having more fixed-point iterations per mode calculation improves accuracy at the cost of solving more linear systems. In order to increase accuracy, it is however significantly more advantageous to increase the number of modes with 2 iterations per mode than to increase the number of iterations per mode with a lower number of computed modes. These trends can also be observed with standard PGD convergence results [11].

4.4. The mean internal energy

We know from the mean power equilibrium that the mean internal energy will increase with the heat provided by the laser and decrease because of the Dirichlet boundary condition. Let us observe the PGD convergence for 4 different mode numbers, looking at the energy loss ratio over time. It corresponds to the ratio of the accumulated dissipated energy over the accumulated input energy.

$$\varepsilon_r(t) = \frac{\mathcal{E}_{in}(t) - \langle \mathcal{E}(\boldsymbol{T}_N(t))\rangle_V}{\mathcal{E}_{in}(t)} ,$$

where $\langle \mathcal{E}(\boldsymbol{T}_N(t))\rangle_V$ is the mean accumulated internal energy (B.3) and $\mathcal{E}_{in}(t)$ is the accumulated energy brought by the laser of Power Q.

We can confirm with Fig.8a that the PGD is converging to the reference FEM solution, as both converge towards a loss ratio of 60 %. We can also observe the effect of the piecewise time mode definition, as the derivative of the energy is not continuous at the activation instances, creating small bumps within each interval. Crucially, the interpolation operators ensure the continuity of the energy, not only for the reference FEM solution towards which the PGD converges but also for any mode number in between.

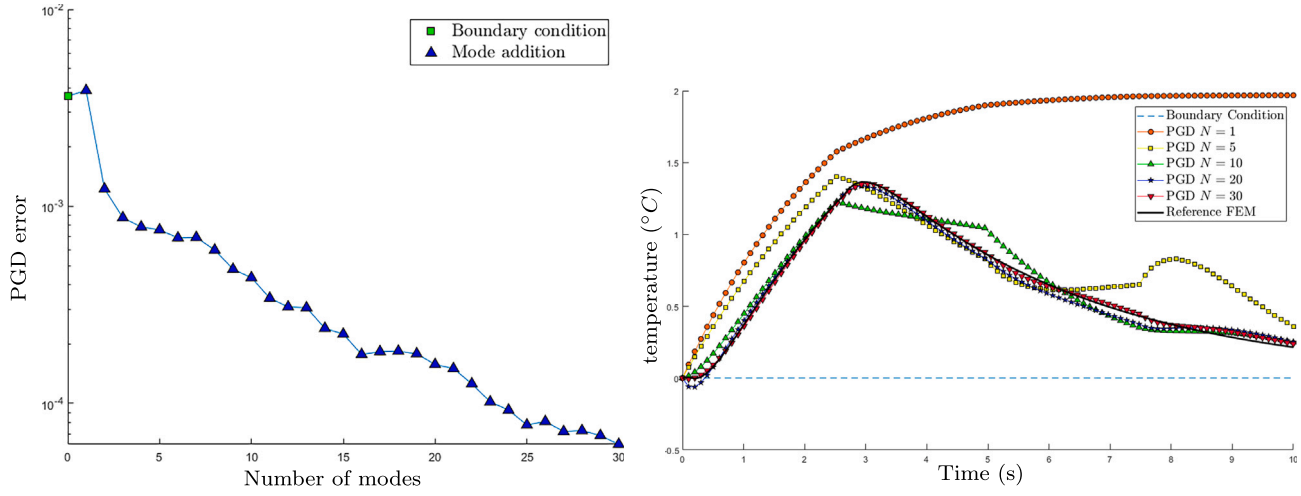
If we now ignore the interpolation constraint and force the interpolation operators $\boldsymbol{h}_k$ (B.13) to zero inside the $\boldsymbol{\mathcal{H}}_k$ matrices, we lose the continuity of the energy at the activation instances – see Fig.8b – confirming the effectiveness of the operators.

4.5. Comparing different domain activations schemes

We now want to understand how the number of domain divisions N_d affects the convergence speed. It is important to recall that a different number of domains leads to different physical problems and that some will be easier to solve than others. Nonetheless, to be able to compare the different simulations energetically, we will remove the Dirichlet boundary conditions, the energy loss ratio will therefore now converge to zero at all times and for all the values of N_d.

The convergence graphs in Fig. 9 confirm that all partitions converge to zero loss represented by the dotted blue line. However, they converge at different speeds, the smaller the number of partitions the faster it converges. Indeed, for both $N = 1$ and $N = 10$, the case with the least number of activations, $N_d = 2$, is the closest to the line of zero energy loss, whereas $N_d = 16$ is the furthest away. This difference in convergence speed can also be observed with the full thermal field. Looking at different single-point evaluations of the thermal field in Fig. 10, there is a clear difference in the required number of modes to achieve convergence. With 4 domains, you can barely differentiate the reference FEM thermal profile from the 50 PGD modes profile whereas with 16 domains, the two profiles are very distinguishable.

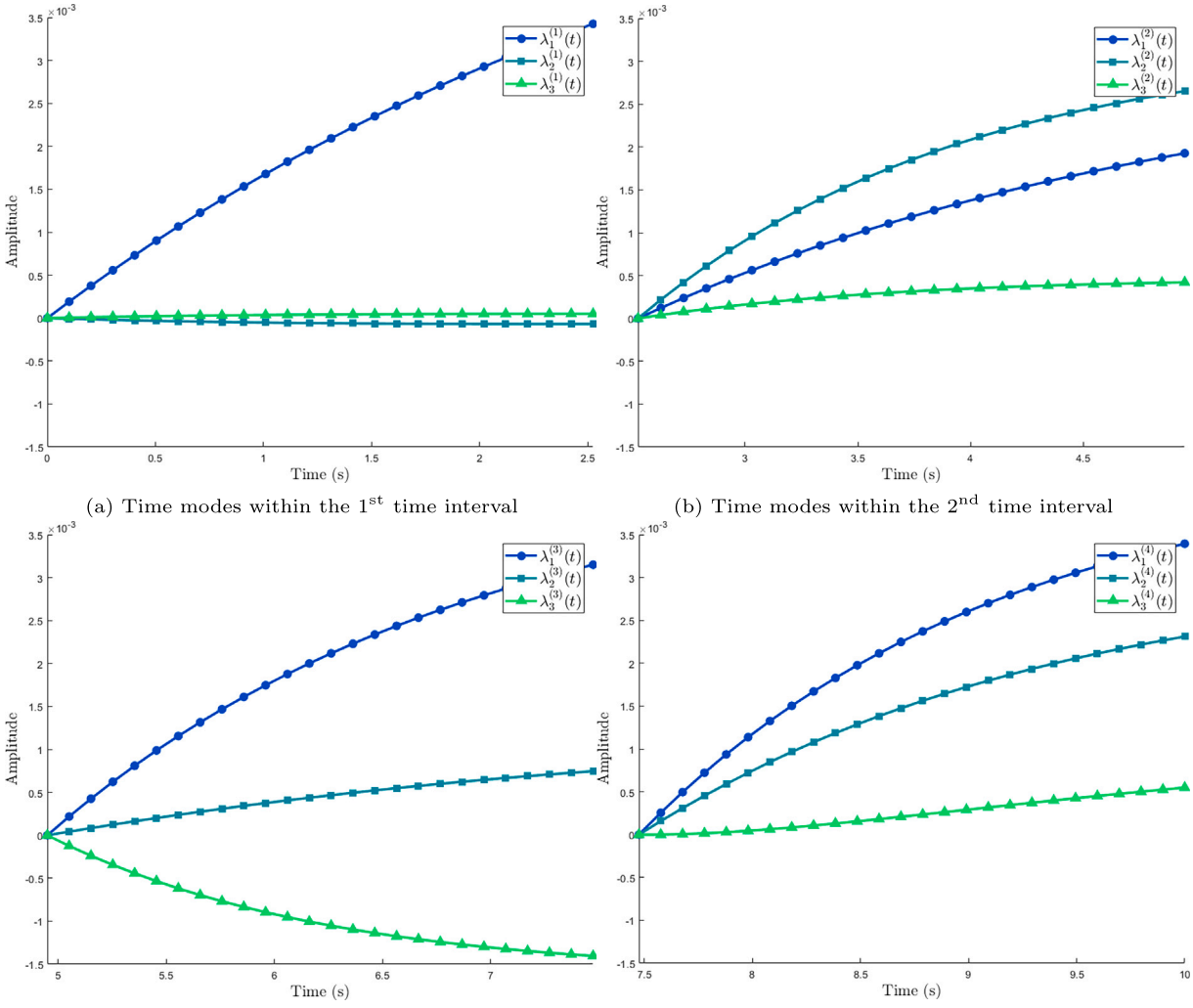
If we now examine the number of modes required to achieve a PGD error smaller than 10^{-3} in Table 3, the results confirm the previous trend



(a) Evolution of the error $\mathcal{E}_{PGD}$

(b) Temperature at point I (see Fig. 1)

Fig. 5. Convergence with 4 domains, $N = 30, N_d = 4, tol = 0.01$, and $N_{\bar{t}} = 2$.



(a) Time modes within the 1st time interval

(b) Time modes within the 2nd time interval

(c) Time modes within the 3rd time interval

(d) Time modes within the 4th time interval

Fig. 6. Representation of the first 3 piecewise time modes. $N_d = 4, tol = 0.01, N_{\bar{t}} = 2$.

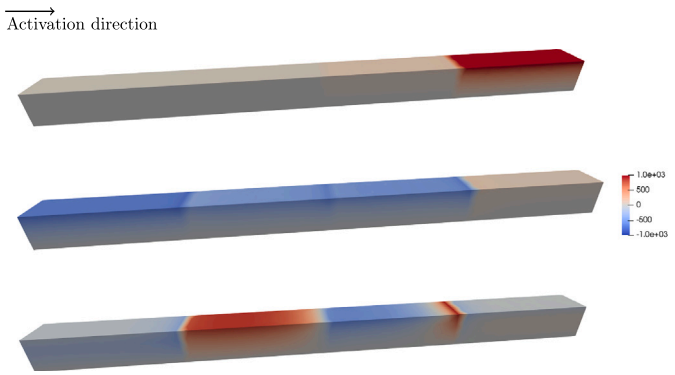


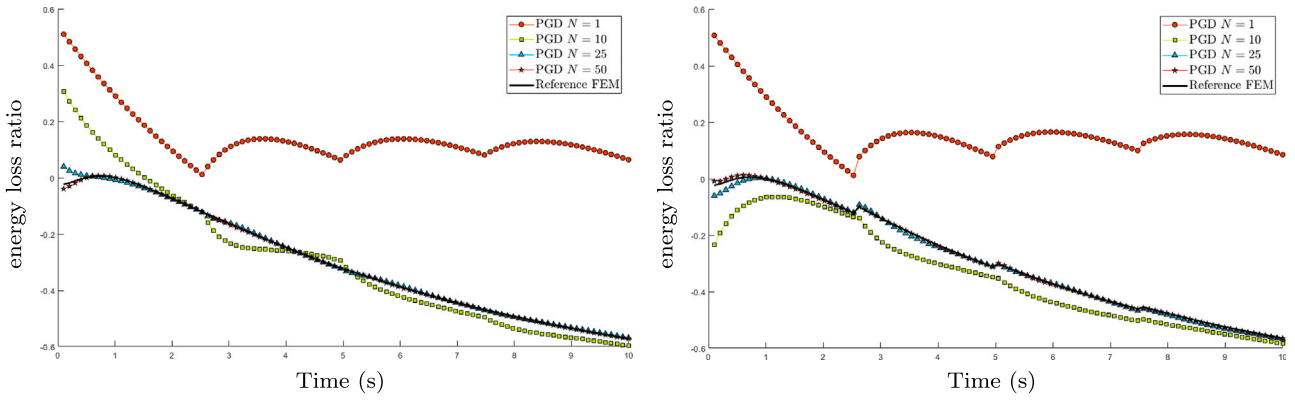
Fig. 7. Representation of the first three expanded space modes, from top to bottom, $\Lambda_{e,1}$, $\Lambda_{e,2}$, and $\Lambda_{e,3}$.

Table 2
PGD Convergence comparison for 4 domains.

Solver	N_d	N	$N_{\bar{t}}$	tol	Total iterations	$\mathcal{E}_{PGD}$
Reference	4	–	–	–	n_t	–
PGD	4	10	2	0.01	20	4.3×10^{-4}
PGD	4	10	10	0.01	54	1.6×10^{-4}
PGD	4	25	2	0.01	50	7.8×10^{-5}

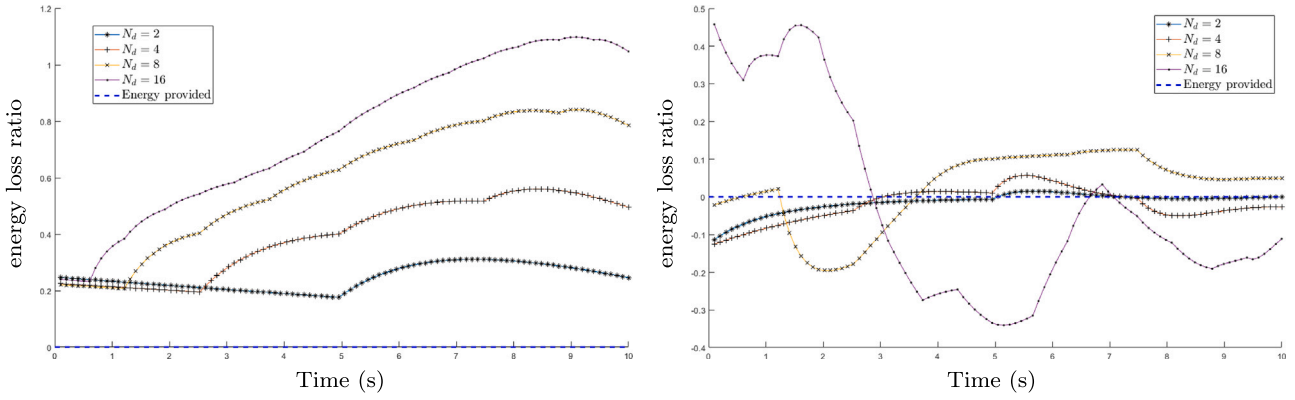
and we seem to need at least twice as many modes N as number of domains N_d. This requirement increase should however not only be blamed on the piecewise formulation but also on the increase in complexity of the laser input. For $N_d = 16$, the laser moves 16 times, hence it seems reasonable to expect the requirement of 16 space modes to represent its motion accurately. It has mathematically been shown that highly discontinuous transport phenomena, such as this traveling discrete heat source, have a slow decay in terms of Kolmogorov N-width [25]. They are therefore not easily reducible and require many modes to be represented accurately.

Finally, attempts have been made to draw conclusions regarding the impact on convergence of the complexity of the current $\boldsymbol{\mathcal{H}}_k$ operator. Forcing the interpolation operators $\boldsymbol{h}_k$ (B.13) to zero reduces the required number of modes by one or two for higher N_d. Making $\boldsymbol{h}_k$ denser - i.e., not just dependent on the last boundary - increases the requirement by a few modes for higher N_d. However, these differences do not appear significant enough to conclude with certainty that more and denser $\boldsymbol{\mathcal{H}}_k$ matrices have an adverse effect on convergence.



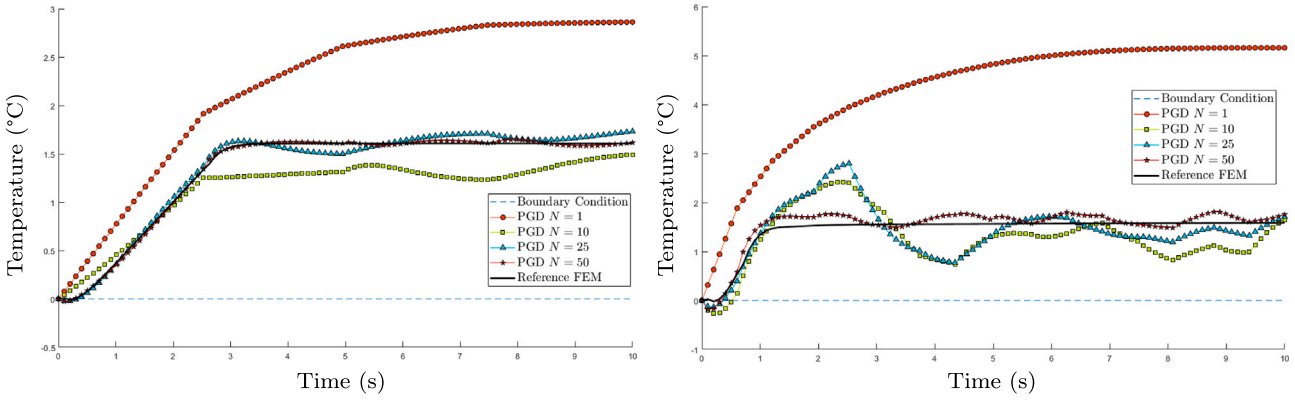
(a) With the interpolation operators (b) Without the interpolation operators

Fig. 8. PGD convergence of the energy loss ratio.



(a) $\epsilon_r(t)$ for N=1 modes (b) $\epsilon_r(t)$ for N=10 modes

Fig. 9. Comparison of PGD convergence with different partitions.



(a) Temperature at point I, (see Fig. 1), with $N_d = 4$ (b) Temperature at point I, (see Fig. 1), with $N_d = 16$

Fig. 10. PGD single point convergence comparison.

4.6. The potential computational speedup

Because our current problem is geometrically simple and linear by nature, in terms of computational costs the reference FEM solver will always remain faster. Space-time separated PGD shines when the original problem contains a large number of expensive linear systems, as it is first and foremost a tool to drastically reduce the required number of solves [11]. As we can see from our previous results in Table 3, we conserve this crucial reduction property with our new formulation. To have a good solution, we observed that $2N_d$ modes were sufficient. Therefore, we go from requiring one linear solve of varying size at every timestep to $2N_{\bar{t}}N_d$ large linear systems of size n, with n being the total number of degrees of freedom. Knowing that the cost to solve a linear system of

Table 3
PGD errors for different activation schemes.

Solver	N_d	N	$N_{\bar{t}}$	tol	Total iterations	$\mathcal{E}_{PGD} < 10^{-3}$
Reference	{2, 4, 8, 16}	–	–	–	n_t	–
PGD	2	3	2	0.01	6	4.5×10^{-4}
PGD	4	9	2	0.01	18	7.9×10^{-4}
PGD	8	14	2	0.01	28	9.9×10^{-4}
PGD	16	23	2	0.01	46	9.9×10^{-4}

size n is $\mathrm{O}(n^\gamma)$, we can estimate that our approach is worth doing if:

$$Cost(\text{PGD}) < Cost(\text{Reference})$$

$$2N_{\bar{t}}N_d\mathrm{O}(n^\gamma) < \sum_{k=1}^{N_d} \frac{n_t}{N_d}\mathrm{O}\left(\left(\frac{kn}{N_d}\right)^\gamma\right).$$

For a low number of activations, we observed that $2N_{\bar{t}}N_d \ll n_t$ therefore, the method can potentially become beneficial. To be computationally competitive, we must verify that doing a sequential run of N_d PGDs is not better. An immediate way to verify that is if the last activation containing the entire mesh requires around the same number of modes to reduce as the full simulation. This is problem dependent, we showed it is the case for our 1D example with Fig. 2. With the current formulation, the PGD finite element operators are notably more expensive to compute (see Eqs. 27 and 29). Indeed, to compute the space mode in our fixed point, we have an exponential increase of operations relative to the number of activations. The assembly time is also very dependent on the complexity of the $\boldsymbol{\mathcal{H}}_k$ matrices and drastically increases if the $\boldsymbol{\mathcal{H}}_k$ are more dense. However, as the mesh is known a priori, the most significant part of the assembly can be done before the fixed point loop. Additional Numerical strategies could therefore be developed to counterbalance the computational disadvantages listed before. A final advantage of low activation numbers is that we can use an update phase to accelerate convergence, as we did for the first example. Currently, its efficiency also decreases with the activation number.

All of these facts lead us to believe that we should have a considerable speedup with this method for larger nonlinear problems. Especially if the activation number is low or if the original problem has a favorable decay of the Kolmogorov N-width at higher activation numbers.

5. Conclusions

In this paper, a new expanded piecewise formulation has been introduced to allow for the application of the Proper Generalized Decomposition model order reduction technique to problems with a time evolving expanding spatial domain. We illustrated this work with a simplified additive manufacturing example with consecutively added hot boundary elements to mimic the addition process. After the detailed explanation of the new formulations, we presented coherent results: we compared the convergence of the method for different domain divisions and PGD parameters. We also showed the conservation of the mean internal energy at every activation instance and for any number of modes. Moreover, we showed that the method could lead to important computational savings under the right circumstances. This new formulation shows promising potential for the future as it preserves the main computational benefits of the original method and expands the possible applications for the PGD model order reduction method. Realistically, the current formulation has limited usability due to the expensive assembly of the operators (29) and the non-ideal performance at higher activation numbers. Nonetheless, it does have its niche use and is a significant first step towards adapting PGD to sequentially activated domains.

Regarding future perspectives, a priority would be to relax the stringent interpolation constraint present at every mode and every activation and achieve it instead only at convergence. This should reduce the number of operators and enable scenarios with a high number of activations. We could then implement this method in a more complex additive manufacturing case, containing nonlinearities or more parameters in the PGD decomposition, leading to measurable time savings. In the long term, one could implement a case in which the activation parameter itself is part of the PGD decomposition, exploiting fully the strengths of PGD. Finally, a comparison to experimental results or other computational results should also be done to analyze the practical use cases of such a tool.

CRediT authorship contribution statement

Max Beckermann: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Ronan Scanff:** Writing – review & editing, Validation, Software, Formal analysis, Conceptualization. **Massimiliano Cremonesi:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Andrea Barbarulo:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

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

Appendix A. Expanded operators for $N_d = 2$

To illustrate the methods and formulations presented within this paper, we will regularly refer back to the case seen in Fig. A.1 with a single activation at time t_2 and $N_d = 2$. Before the introduction of the expanded strategy in Section 3.2 we had two possible states for the shape function: $\boldsymbol{N}^{(1)}(x)$ of size n_1, $\forall t \in I_1$ and $\boldsymbol{N}^{(2)}(x)$ of size $n_2 > n_1$, $\forall t \in I_2$. The expanded shape function $\boldsymbol{N}_e(x,t)$ and all of the expanded tensors will therefore be of characteristic size n_2:

$$\begin{aligned}&\boldsymbol{N}_e(x,t) = \mathbb{1}_{t\in I_1}(t)\boldsymbol{N}_e^{(1)}(x) + \mathbb{1}_{t\in I_2}(t)\boldsymbol{N}_e^{(2)}(x)\,,\\ &\text{with}\quad \boldsymbol{N}_e^{(1)}(x) = \begin{matrix}\Omega_1\\ \Omega_2\end{matrix}\begin{pmatrix}\boldsymbol{N}^{(1)}(x)\\ \boldsymbol{0}\end{pmatrix}\quad\text{and}\quad \boldsymbol{N}_e^{(2)}(x) = \boldsymbol{N}^{(2)}(x)\,.\end{aligned} \tag{A.1}$$

$$\begin{aligned}&\boldsymbol{M}_e(t) = \mathbb{1}_{t\in I_1}(t)\boldsymbol{M}_e^{(1)} + \mathbb{1}_{t\in I_2}(t)\boldsymbol{M}_e^{(2)}\,,\\ &\text{with}\quad \boldsymbol{M}_e^{(1)} = \begin{matrix}\\ \Omega_1\\ \Omega_2\end{matrix}\begin{matrix}\Omega_1 & \Omega_2\\ \end{matrix}\begin{pmatrix}\boldsymbol{M}^{(1)} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{pmatrix}\quad\text{and}\quad \boldsymbol{M}_e^{(2)} = \boldsymbol{M}^{(2)}\,.\end{aligned} \tag{A.2}$$

The construction of the remaining finite element operators will be the same as the method used for the expanded mass matrix. Once again, note that for the FE operators $\bullet^{(2)}$ refers to all the domains activated up to time t_2 , not just Ω_2.

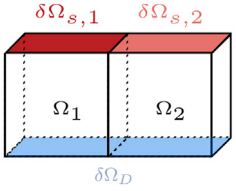


Fig. A.1. Geometry for $N_d = 2$.

Appendix B. An interpolation operator for the conservation of internal energy

To understand the energy conservation problem, let us start with the toy problem seen in Fig. B.1a, which stems from Fig. A.1. We have a block activating at temperature $T_{new} = 100$ °C next to a block of similar material and volume at initial temperature $T_{ini} = 0$ °C. Physically, as it is an adiabatic process, we should expect the two blocks to converge to the same temperature of $T_{eq} = 50$ °C. Numerically, as can be seen in Fig. B.1b, we are not converging to the correct temperature but slightly lower; there seems to be a loss of energy at the addition of the material. As a matter of fact, this can be explained by the temperature choices on the boundary of the two blocks, seen in green in Fig. B.1a. Naively, only the new points are initialized at T_{new} while the boundary remains at T_{ini}. There are multiple elements on the hot side in which the temperature will therefore be interpolated and have a lower initial temperature than T_{new}, thus the lower final equilibrium temperature.

Our idea to fix this numerical imbalance is to quantify the loss of energy and compensate it with a constant correction term over the new elements. Let us first look at the mean power equilibrium for a single domain Ω of our original problem:

$$c_p\rho \int_\Omega \frac{\partial T}{\partial t}(x,t)dV = \lambda_c \int_\Omega \Delta T(x,t)dV \quad \forall t \in I, \tag{B.1}$$

Transferring the equilibrium to the boundaries with Green's theorem, this translates to:

$$\begin{aligned} c_p\rho \int_\Omega \frac{\partial T}{\partial t}(x,t)dV &= \lambda_c \int_{\delta\Omega} \boldsymbol{\nabla} T(x,t).\boldsymbol{n} dS \\ &= \lambda_c \int_{\delta\Omega_u} \frac{\partial T}{\partial n}(x,t)dS + \mathcal{Q}S_{\delta\Omega_s} \quad \forall t \in I, \end{aligned} \tag{B.2}$$

Which means that the internal power either comes from the laser source of power $\mathcal{Q}S_{\delta\Omega_s}$ or is exchanged through the Dirichlet boundary condition. After time integration of the left-hand side term, we retrieve the mean internal energy:

$$\langle \mathcal{E}(t) \rangle_V = c_p\rho \int_\Omega \{T(x,t) - T_0(x)\}dV \quad \forall t \in I, \tag{B.3}$$

Where $T_0(x)$ is the thermal field of zero energy or the initial thermal field. Note that in the current example we are adiabatic, therefore all the source terms in the power equation (B.2) are null. We therefore also have a conservation of the mean internal energy at all times.

In case of a domain activation at time t_a, this internal energy should be conserved. T_0 then verifies the following:

$$\begin{cases} T_0(x) = T_{ini} & \forall x \in \Omega_1 \\ T_0(x) = T_{new} & \forall x \in \Omega_2 \end{cases}, \tag{B.4}$$

and the energy conservation at time t_a is then written:

$$\langle \mathcal{E}(t_a^-) \rangle_V = \langle \mathcal{E}(t_a^+) \rangle_V \tag{B.5}$$

$$c_p\rho \int_{\Omega_1} \{T(x,t_a^-) - T_{ini}\}dV = c_p\rho \int_{\Omega_1 \cup \Omega_2} \{T(x,t_a^+) - T_0(x)\}dV \tag{B.6}$$

$$= c_p\rho \int_{\Omega_1} \{T(x,t_a^+) - T_{ini}\}dV + c_p\rho \int_{\Omega_2} \{T(x,t_a^+) - T_{new}\}dV . \tag{B.7}$$

Considering we do not change the thermal field in Ω_1 at activation, $T(x,t_a^+) = T(x,t_a^-) \ \forall x \in \Omega_1$ and we must have:

$$c_p\rho \int_{\Omega_2} \{T(x,t_a^+) - T_{new}\}dV = 0 . \tag{B.8}$$

Introducing the expanded finite element discretization found in Appendix A and choosing the lifting term such that $T_0(x) = T_L(x)$ we get:

$$\boldsymbol{I}^M_{\Omega_2} . \boldsymbol{T}_e(t_a^+) = 0 \tag{B.9}$$

with $\boldsymbol{I}^M_{\Omega_2}$ the integration operator directly dependent on the mass operator $\boldsymbol{M}^{(2)}$. Let us denote by $|_{1\#}$ and $|_{\#2}$ the nodes only present in Ω_1 and Ω_2 respectively, the nodes on the boundaries are denoted by $|_{1\#2}$.

Because it is the operator for Ω_2, we know that $\boldsymbol{I}^M_{\Omega_2}|_{1\#} = 0$. Moreover at activation, only the new points have a new temperature and they are all initialized at ε_T with ε_T the scalar correction term compensating for the loss of energy, therefore:

$$\boldsymbol{T}_e(t_a^+) = \boldsymbol{T}_e(t_a^-)|_{1\#} + \boldsymbol{T}_e(t_a^-)|_{1\#2} + \varepsilon_T|_{\#2} . \tag{B.10}$$

Reinserting $\boldsymbol{T}_e(t_a^+)$ into (B.9):

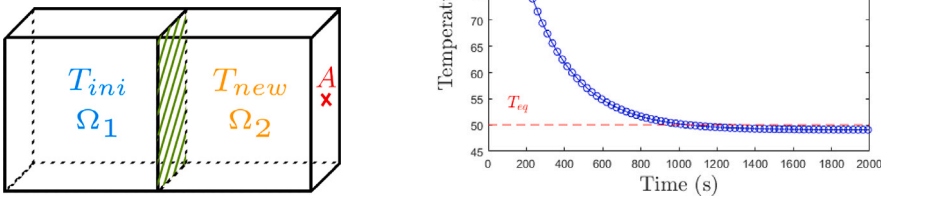
$$\boldsymbol{I}^M_{\Omega_2}|_{\#2} \cdot \varepsilon_T|_{\#2} = -\boldsymbol{I}^M_{\Omega_2}|_{1\#2} \cdot \boldsymbol{T}_e(t_a^-)|_{1\#2} , \tag{B.11}$$

$$\varepsilon_T \sum_{j \in \#2} \boldsymbol{I}^M_{\Omega_2,j} = -\boldsymbol{I}^M_{\Omega_2}|_{1\#2} \cdot \boldsymbol{T}_e(t_a^-)|_{1\#2} , \tag{B.12}$$

and isolating the scalar correction term we get:

$$\varepsilon_T = \frac{-\boldsymbol{I}^M_{\Omega_2}|_{1\#2}}{\sum_{j\in\#2} \boldsymbol{I}^M_{\Omega_2,j}} \cdot \boldsymbol{T}_e(t_a^-)|_{1\#2} = \boldsymbol{h}_2^T \cdot \boldsymbol{T}_e(t_a^-)|_{1\#2} , \tag{B.13}$$

where $\boldsymbol{h}_2$ is of the size of the boundary 1#2 and is the correction term interpolation operator associated with the activation of Ω_2. Note that



(a) Setup of the toy problem (b) Observed temperature profile with no conservation of energy

Fig. B.1. Representations to understand the energy conservation requirements.

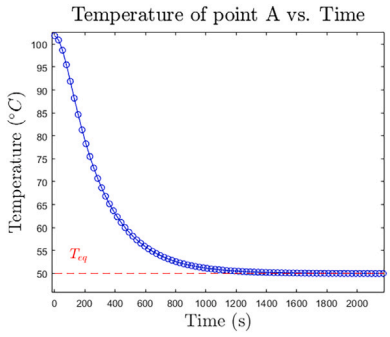


Fig. B.2. Temperature Profile with the correction term.

this operator can be calculated independently of the current solution and only depends on the geometry and material in our linear thermal case.

Using this newly established initialization, we can calculate the new temperature profile at point A as seen in Fig. B.2. We find that we are now correctly converging to 50 °C.

Let us go back to our original problem and introduce this conservation of mean internal energy at the activation instances. We can introduce $\mathcal{H}_2$ the full expanded interpolation matrix associated with $\boldsymbol{h}_2$ written as:

$$\mathcal{H}_2 \boldsymbol{T}_e = \begin{pmatrix} \boldsymbol{I}_{\{1\#\}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{\{1\#2\}} & \boldsymbol{0} \\ \boldsymbol{0} & \begin{pmatrix} \boldsymbol{h}_2^T \\ \vdots \\ \boldsymbol{h}_2^T \end{pmatrix}_{\{\#2\}\times\{1\#2\}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{T}|_{e\#1} \\ \boldsymbol{T}|_{e1\#2} \\ \boldsymbol{T}|_{e\#2} \end{pmatrix}, \tag{B.14}$$

with $\boldsymbol{I}_{\{\bullet\}}$ the identity matrix associated with domain $\{\bullet\}$. We therefore have at activation time t_2 with the expanded notation:

$$\boldsymbol{T}_e(t_2^+) = \mathcal{H}_2 \boldsymbol{T}_e(t_2^-). \tag{B.15}$$

This can then be generalized for all N_d to retrieve the interpolation matrices $\mathcal{H}_k$ that we required for Eq. (19). Note that (18) is a special case of (19) for which the interpolation operators $\boldsymbol{h}_k$ are null. As explained at the end of Section 3.2, to implement such a time-discrete constraint into the PGD framework that is global in time, a new formulation is required.

Data availability

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

References

[1] Volume 2 snapshot-based methods and algorithms. De Gruyter; 2021.
[2] Benner P, Ohlberger M, Cohen A, Willcox K. Model reduction and approximation. Society for Industrial and Applied Mathematics; 2017.
[3] Quarteroni A, Manzoni A, Negri F. Reduced basis methods for partial differential equations: an introduction. Springer International Publishing; 2015.
[4] Hesthaven JS, Rozza G, Stamm B. Certified reduced basis methods for parametrized partial differential equations. Cham: Springer International Publishing; 2016. SpringerBriefs in Mathematics.
[5] Cordier L, Bergmann M. Réduction de dynamique par décomposition orthogonale aux valeurs propres POD. Ec printemps OCET. 2006;7563:107.
[6] Ladevèze P. Nonlinear computational structural mechanics: new approaches and non-incremental methods of calculation. New York: Springer; 1999. Mechanical Engineering Series.
[7] Chinesta F, Ladeveze P, Cueto E. A short review on model order reduction based on proper generalized decomposition. Arch Comput Methods Eng. 2011;18(4):395–404.
[8] Chinesta F, Ladevèze P. 3 proper generalized decomposition. In: Volume 2 snapshot-based methods and algorithms. De Gruyter; 2021. pp. 97–138.
[9] Nouy A. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. Comput Methods Appl Mech Eng. 2010;199(23–24):1603.
[10] Boucinha L, Ammar A, Gravouil A, Nouy A. Ideal minimal residual-based proper generalized decomposition for non-symmetric multi-field models – application to transient elastodynamics in space-time domain. Comput Methods Appl Mech Eng. 2014;273:56–76.
[11] Favoretto B, de Hillerin CA, Bettinotti O, Oancea V, Barbarulo A. Reduced order modeling via PGD for highly transient thermal evolutions in additive manufacturing. Comput Methods Appl Mech Eng. 2019;349:405–30.
[12] Qin Z, Talleb H, Ren Z. A proper generalized decomposition-based solver for nonlinear magnetothermal problems. IEEE Trans Magn. 2016;52(2):1–9.
[13] Ammar A, Normandin M, Chinesta F. Solving parametric complex fluids models in rheometric flows. J Nonnewton Fluid Mech. 2010;165(23):1588–601.
[14] Dumon A, Allery C, Ammar A. Proper generalized decomposition method for incompressible flows in stream-vorticity formulation. Eur J Comput Mech. 2010;19(5–7):591–617.
[15] Fernandes JWD, Kuche RA, Barbarulo A. A stabilized mixed space–time proper generalized decomposition for the Navier–Stokes equations. Comput Methods Appl Mech Eng. 2021;386:114102.
[16] Ghnatios C, Masson F, Huerta A, Leygue A, Cueto E, Chinesta F. Proper generalized decomposition based dynamic data-driven control of thermal processes. Comput Methods Appl Mech Eng. 2012;213–216:29–41.
[17] Goutaudier D, Berthe L, Chinesta F. Proper generalized decomposition with time adaptive space separation for transient wave propagation problems in separable domains. Comput Methods Appl Mech Eng. 2021;380:113755.
[18] Vella C, Prudhomme S. PGD reduced-order modeling for structural dynamics applications. Comput Methods Appl Mech Eng. 2022;402:115736.
[19] Arjoune T, Markert B, Bamer F. Non-incremental response evaluation in geometrically nonlinear structural dynamics using a space-time stiffness operator. Comput Mech. 2022;70(2):309–33.
[20] Modesto D, Zlotnik S, Huerta A. Proper generalized decomposition for parameterized Helmholtz problems in heterogeneous and unbounded domains: application to harbor agitation. Comput Methods Appl Mech Eng. 2015;295:127–49.
[21] Deeb A, Kalaoun O, Belarbi R. Proper generalized decomposition using Taylor expansion for non-linear diffusion equations. Math Comput Simulat. 2023;208:71–94.
[22] Tang S, Guan X, Liu WK. Solving diffusive equations by proper generalized decomposition with preconditioner. Comput Mech. 2024;73(1):199–221.
[23] Ma W, Shen Y. A mixed formulation of proper generalized decomposition for solving the Allen-Cahn and Cahn-Hilliard equations. Finite Elem Anal Des. 2021;194:103560.
[24] Tang S, Xu H. Ruminated tensor decomposition algorithm for solving inviscid burgers' equation. J Comput Phys. 2025;523:113663.
[25] Ohlberger M, Rave S. Reduced basis methods: success, limitations and future challenges. 2016.
[26] Manzoni A, Quarteroni A, Rozza G. Model reduction techniques for fast blood flow simulation in parametrized geometries. Int J Numer Method Biomed Eng. 2012;28(6–7):604–25.
[27] Rozza G, Lassila T, Manzoni A. Reduced basis approximation for shape optimization in thermal flows with a parametrized polynomial geometric map. In: Hesthaven JS, Rønquist EM, editors. Spectral and high order methods for partial differential equations; Berlin, Heidelberg: Springer; 2011. p. 307–15.
[28] Stabile G, Zancanaro M, Rozza G. Efficient geometrical parametrization for finite-volume-based reduced order methods. Int J Numer Methods Eng. 2020;121(12):2655–82.
[29] Iollo A, Taddei T. Mapping of coherent structures in parameterized flows by learning optimal transportation with Gaussian models. J Comput Phys. 2022;471:111671.
[30] Taddei T. A registration method for model order reduction: data compression and geometry reduction. SIAM J On Sci Comput. 2020;42(2):A997–1027.
[31] Kabalan A, Casenave F, Bordeu F, Ehrlacher V, Ern A. Elasticity-based morphing technique and application to reduced-order modeling. Appl Math Model. 2025;141:115929.
[32] Kazemzadeh-Parsi M-J, Pasquale A, Di Lorenzo D, Champaney V, Ammar A, Chinesta F. NURBS-based shape parametrization enabling PGD-based space separability: methodology and application. Finite Elem Anal Des. 2023;227:104022.
[33] Chinesta F, Keunings R, Leygue A. The proper generalized decomposition for advanced numerical simulations: a primer, 1st ed. Cham, 1st Springer International Publishing: Imprint: Springer; 2014. SpringerBriefs in Applied Sciences and Technology.
[34] Ghnatios C, Abisset E, Ammar A, Cueto E, Duval JL, Chinesta F. Advanced separated spatial representations for hardly separable domains. Comput Methods Appl Mech Eng. 2019;354:802–19.
[35] Gibson I, Rosen D, Stucker B, Khorasani M. Additive manufacturing technologies. Springer Cham; 2020.
[36] Bikas H, Stavropoulos P, Chryssolouris G. Additive manufacturing methods and modelling approaches: a critical review. Int J Adv Manuf Technol. 2016;83(1):389–405.
[37] Dong L, Makradi A, Ahzi S, Remond Y. Three-dimensional transient finite element analysis of the selective laser sintering process. J Mater Process Technol. 2009;209(2):700–06.
[38] Michaleris P. Modeling metal deposition in heat transfer analyses of additive manufacturing processes. Finite Elem Anal Des. 2014;86:51–60.
[39] Zhang Y, Chou Y. Three-dimensional finite element analysis simulations of the fused deposition modelling process. Proc Inst Mech Eng, Part B: J Eng Manuf. 2006;220(10):1663–71.
[40] Sachs E, Vezzetti E. Numerical simulation of deposition process for a new 3DP printhead design. J Mater Process Technol. 2005;161(3):509–15.
[41] Cueto E, González D, Alfaro I. Proper generalized decompositions. Cham: Springer International Publishing; 2016. SpringerBriefs in Applied Sciences and Technology.
[42] Geuzaine C, Remacle J-F. Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Methods Eng. 2009;79(11):1309–31.