Reduced order modeling via PGD for highly transient thermal evolutions in additive manufacturing

B. Favoretto, C.A. de Hillerin, O. Bettinotti, V. Oancea, Andrea Barbarulo

Reduced Order Modeling via PGD for Highly Transient Thermal Evolutions in Additive Manufacturing

B. Favoretto[a,b], C.A. de Hillerin[b], O. Bettinotti[c], V. Oancea[c], A. Barbarulo[a]

[a] *Laboratoire MSSMat, CentraleSupélec/CNRS UMR 8579/ Université Paris Saclay, 91190 Gif-sur-Yvette, France*

[b] *Dassault Systèmes SIMULIA Corp., Vélizy-Villacoublay, France*

[c] *Dassault Systèmes SIMULIA Corp., Johnston, RI, USA*

Abstract

In this paper, a highly performing model order reduction technique called Proper Generalized Decomposition (PGD) is applied to the numerical modeling of highly transient non-linear thermal phenomena associated with additive manufacturing (AM) powder bed fabrication (PBF) processes. The manufacturing process allows for unprecedented design freedom but fabricated parts often suffer from lower quality mechanical properties associated with the fast transients and high temperature gradients during the localized melting-solidification process. For this reason, an accurate numerical model for the thermal evolutions is a major necessity. This work focuses on providing a low-cost/high accuracy prediction of the high gradient thermal field occurring in a material under the action of a concentrated moving laser source, while accounting for phase changes, material non-linearities and time and space-dependent boundary conditions. An extensive numerical simulation campaign shows that the use of PGD in this context enables a remarkable reduction in the total number of global matrix inversions (5 times less or better) compared to standard techniques when simulating realistic AM PBF scenarios.

Keywords: AM Process Simulation, Powder Bed Fabrication, Reduced Order Modeling, Proper Generalized Decomposition

1. Introduction

Since their inception, selective laser melting (SLM) and electron beam melting (EBM) powder bed fabrication (PBF), as prime examples in addi-

tive manufacturing (AM), proved to be a paradigm shift for manufacturing processes. They consist in selective melting of superposed layers of metal powder using a machine-controlled moving high energy source. Due to their nature, these processes allow unprecedented freedom in designing, personalization and optimization of mechanical parts. Moreover, they are particularly suited for software-hardware integration when the desired geometry is conceived with a Computer Assisted Design (CAD) tool and directly produced by an automated process, removing many of the intermediate steps between the designer's vision and the physical world.

Effective numerical methods capable of predicting final characteristics of the part, spotting critical points during the process and helping the design process itself are often invaluable in gaining insight in the process. Since SLM and EBM encompass complex multiphysics (thermal, mechanical, phase change, metallurgic) and pose a gigantic multi-scale problem in both space and time, special consideration is required in numerical analysis (see [1, 2, 3])including fast executing AM-specialized approaches (see [4]).

This work focuses on the highly non-linear thermal phenomena occurring in the immediate proximity of the fast moving heat source where temperature evolution rates, phase changes and thermal gradients are the most intense, all happening on a very small scale (see [5]). The goal is to provide a lower-cost / high accuracy simulation of this important region. To provide this solution, a technique belonging to Reduced Order Model (ROM) family (see [6, 7, 8, 9, 10, 11]) called Proper Generalized Decomposition (PGD) (see [12, 13, 14, 15, 16, 17]) has been adapted for this problem to consider temperature-dependent material properties and latent heat associated with phase change. Heating is accomplished via a laser model that moves rapidly along the path while heat removal is modeled via temperature dependent thermal conductivity and rapidly evolving Neumann boundary conditions. This model order reduction technique allows computing a reduced base for each variable without solving the full eigenvalue problem. In order to properly keep into account the non-linearities and the boundary condition of highly transient thermal evolutions, a dedicated PGD algorithm and methodology have been developed, representing the main contributions of this work. Thanks to it, computational cost is significantly reduced and variable separation is achieved enabling a highly meaningful reduced basis. Benchmarking against the full transient finite element solutions are performed. For the purpose of benchmarking, a linear finite element full integration formulation was chosen. An in-depth study on PGD controls (number of modes, number

of iterations, etc.) and on how they can be best selected for efficient computations is included and accuracy and performance findings are carefully tabulated.

2. Problem description

For the purpose of this study, a patch of material of unit thickness is subjected to intense heating from a heat source moving over its surface (see Figure 1). While the heating is three-dimensional in nature and the build

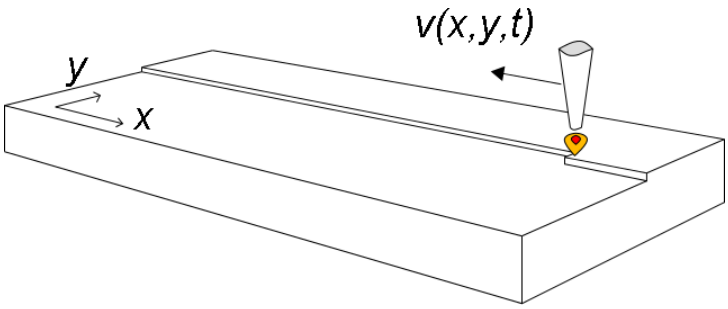


Figure 1: Problem under study

direction involves a very important heat transport mechanism, for simplicity, we have modeled it as a 2D problem in the xy plane (all units are modified accordingly). Despite this simplification, all core non-linearities associated with additive manufacturing heat transfer computations including material properties, high temperature gradients and high heating and cooling rates are included and therefore the setup serves the core purpose of the paper. The patch is a square region of given length (L) made of Ti-6Al-4V, a popular Titanium-based alloy used widely in powder bed fabrication additive manufacturing (see [18]). The material definition includes temperature-dependent properties (such as the density (ρ), the heat capacity (C_p) and the thermal conductivity (k)); the heat source is a laser beam with prescribed power density input (P) in the form of a Gaussian distribution initially centered at point (X_0, Y_0) relative to a (O, x, y) coordinate system aligned with the plate edges with origin at the lower left corner of the plate. The heat source is moving over the patch in the negative x-direction with a given speed (v). The patch loses heat to the surrounding environment at temperature (T_{env}) through its surface according to a film condition with heat transfer coefficient (h). In addition heat is also lost through its boundaries according to a

prescribed outward flux ($\overline{q}^{out}(x,y,t)$), where t denotes the time. The values of all these parameters are subject to change depending on the particular test scenario presented in the Results and Discussion section.

This problem can be formulated as a boundary value problem governed by a parabolic partial differential equation (PDE) for the scalar temperature field $T(x,y,t)$, with material non-linearities and a non-linear varying source, in a 2D domain $\Omega = [0,L] \times [0,L]$ over a time interval $I = [0, t_{end}]$, with prescribed Neumann boundary conditions and uniform initial conditions. In its most general form, the boundary value problem is formulated as follows:

$$\begin{cases} \frac{\partial(\rho C_p(T)T(x,y,t))}{\partial t} + \nabla \cdot (-k(T)\nabla T(x,y,t)) = Q^{in}(x,y,t) - Q^{out}(T) \text{ in } \Omega \times I \\ -k(T)\nabla T(x,y,t) \cdot \mathbf{n}^{out} = \overline{q}^{out}(x,y,t) \text{ on } \partial\Omega \times I \\ T(x,y,0) = T_0 \text{ in } \Omega \end{cases} \tag{1}$$

where

$$\begin{cases} Q^{in}(x,y,t) = \frac{2P}{\pi R_0^2} \exp\left(\frac{-2}{R_0^2}((x - x_0(t))^2 + (y - y_0(t))^2)\right) \\ Q^{out}(T) = h\ (T - T_{env}) \end{cases}. \tag{2}$$

Here $\mathbf{n}^{out}$ represents the outer unit normal to the domain boundary, T_0 represents the prescribed uniform initial temperature field, R_0 represents a characteristic radius for the Gaussian distribution and

$$\begin{cases} x_0(t) = X_0 + f_x(t) \\ y_0(t) = Y_0 + f_y(t) \end{cases} \tag{3}$$

represent the prescribed energy source motion. A description of the Gaussian source model can be found in [19].

3. Methods and Algorithms

Since the PGD results are compared to Finite Element Method (FEM) results (both in terms of accuracy and performance), the FEM and PGD methods will be described in parallel, for each new simulation capability added in order to solve the proposed problem.

We start by describing the linear case, then move to comprehensive non-linear cases, including temperature-dependent properties and latent heat associated with phase changes, fast moving time-dependent heat sources and temperature-dependent Neumann boundary conditions.

3.1. Linear case

The FEM and PGD approaches are first briefly reviewed in the context of a linear problem (parameters such as ρ, C_p and k are considered independent of the temperature) with zero Neumann boundary conditions (insulated) and a stationary and temperature independent heat source. Consequently, $\overline{q}^{out} = Q^{out} = f_x = f_y = 0 \ \forall t \in I$, and Q^{in} is independent of time:

$$Q^{in}(x,y) = \frac{2P}{\pi R_0^2} \exp\left(\frac{-2}{R_0^2}((x-X_0)^2 + (y-Y_0)^2)\right). \tag{4}$$

So, for the linear problem, Equation (1) can be reformulated as:

$$\begin{cases} \rho C_p \frac{\partial T}{\partial t} - k\nabla^2 T = Q^{in}(x,y) \text{ in } \Omega \times I \\ k\nabla T \cdot \mathbf{n}^{out} = 0 \text{ on } \partial\Omega \times I \\ T(x,y,0) = T_0 \text{ in } \Omega \end{cases} . \tag{5}$$

The problem is solved by classical FEM (see [20]). A backward-Euler finite difference scheme is used for implicit time integration in an incremental approach.

For PGD, the temperature field is assumed to be decomposable in the form of a finite sum of products between space and time functions:

$$T(x,y,t) \approx \sum_{i=1}^{N_{mod}} \Phi_i(x,y)\lambda_i(t) \tag{6}$$

where N_{mod} is the number of products, determined based on a convergence criterion. The space functions are discretized according to $\Phi_i(x,y) \approx \mathbb{N}(x,y)\mathbf{\Phi}_i$, where $\mathbf{\Phi}_i$ represents the i^{th} nodal solution vector and $\mathbb{N}(x,y)$ represents a set of interpolation functions. In a way, this is similar to the FEM approach, where the solution field is discretized as $T(x,y,t) \approx \mathbb{N}(x,y)\mathbf{T}(t)$, with $\mathbf{T}(t)$ representing the solution vector at the nodes of the FEM mesh.

The nodal solution vector is thus expressed as a separated representation of the form $\mathbf{T}(t) \approx \sum_{i=1}^{N_{mod}} \mathbf{\Phi}_i\lambda_i(t)$, where the vectors $\mathbf{\Phi}_i$ can be viewed as "basis vectors" and the time functions $\lambda_i(t)$ can be viewed as coefficients in a linear combination analogous to FEM-based eigenmode superposition. Alternatively, each couple $(\mathbf{\Phi}_i, \lambda_i(t))$ can be viewed as a "mode" in this superposition. This terminology will be used in the Results and Discussion section.

In contrast to usual methods (which include FEM eigenmode analysis and other order reduction methods such as the Proper Orthogonal Decomposition - POD method), in which the projection basis is known a priori, the PGD basis vectors and time functions are both unknown and will be computed on the fly, for the particular problem at hand.

The basis is progressively enriched by the addition, at stage $n < N_{mod}$, of a new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ which can be viewed as a correction to the previous result $\mathbf{T}_{n-1}(t)$:

$$\mathbf{T}_n(t) = \sum_{i=1}^{n} \mathbf{\Phi}_i \lambda_i(t) = \sum_{i=1}^{n-1} \mathbf{\Phi}_i \lambda_i(t) + \mathbf{\Phi}_n \lambda_n(t) = \mathbf{T}_{n-1}(t) + \mathbf{\Phi}_n \lambda_n(t). \quad (7)$$

The basis computation proceeds in successive stages. At each stage there are two steps (see [15]):

- An enrichment step: a new couple consisting of a basis vector and a time function $(\mathbf{\Phi}_n, \lambda_n(t))$ is computed;
- An update step: all time functions are recomputed.

PGD Enrichment step (addition of a new couple)

Assuming that the first $n-1$ couples $(\mathbf{\Phi}_i, \lambda_i(t))_{i=1,\dots,n-1}$ have been previously obtained, the current approximation of the field is written in the form $\mathbf{T}_n(t) = \mathbf{T}_{n-1}(t) + \mathbf{\Phi}_n \lambda_n(t)$, with the new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ unknown.

The new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ is obtained by applying an alternating directions fixed point iterations method , which is the standard choice for PGD algorithms (see [15]). In contrast with the FEM, the time marching increment has less influence on convergence. Other factors, such as a bad choice of the initialization function (for example a null function) can have a more detrimental effect.

The process starts with the initialization of the time function $\lambda_n^{(0)}(t)$. This choice is arbitrary, since several functions can result in converged results. We opted for using a linear time function. This was determined on physical grounds, based on the fact that in the absence of transients the solution of the boundary value problem with constant source and no Neumann boundary conditions behaves as an unbounded growth of the temperature field.

After the initialization, the fixed-point iterations consist in computing in sequence: $\mathbf{\Phi}_n^{(1)}$ and $\lambda_n^{(1)}(t)$ (first iteration), then $\mathbf{\Phi}_n^{(2)}$ and $\lambda_n^{(2)}(t)$ (second iteration), etc., until convergence to the couple $(\mathbf{\Phi}_n, \lambda_n(t))$.

At the k^{th} iteration, consistently with the basic premise of separation of variables, one thus solves in sequence:

- A space problem for $\mathbf{\Phi}_n^{(k)}$ (with $\lambda_n^{(k-1)}(t)$ known);
- A time problem for $\lambda_n^{(k)}(t)$ (with $\mathbf{\Phi}_n^{(k)}$ known).

Space problem

Assuming that $\lambda_n^{(k-1)}(t)$ is known, one obtains $\mathbf{\Phi}_n^{(k)}$ by assuming that $\mathbf{T}_n(t) = \mathbf{T}_{n-1}(t) + \mathbf{\Phi}_n^{(k)} \lambda_n^{(k-1)}(t)$ and by choosing a test function for the weak formulation of the problem in the form $\mathbf{T}^*(t) = \mathbf{\Phi}^* \lambda_n^{(k-1)}(t)$.

Based on the work of Nouy (see [15]) and Néron (see [16]), by employing the virtual work machinery, the space problem is then given by the equation:

$$\left(\int_I \lambda_n^{(k-1)}(t) \dot{\lambda}_n^{(k-1)}(t)\, dt\ \mathbb{M} + \int_I {\lambda_n^{(k-1)}}^2(t)\, dt\ \mathbb{K} \right) \mathbf{\Phi}_n^{(k)} = \int_I \lambda_n^{(k-1)}(t) \mathbf{F}_{Q_{in}}\, dt - \sum_{i=1}^{n-1} \left(\int_I \lambda_n^{(k-1)}(t) \dot{\lambda}_i(t)\, dt\ \mathbb{M} + \int_I \lambda_n^{(k-1)}(t) \lambda_i(t)\, dt\ \mathbb{K} \right) \mathbf{\Phi}_i \tag{8}$$

where

$$\begin{cases} \mathbb{M} = \int_\Omega \mathbb{N}^T(x,y) \rho C_p \mathbb{N}(x,y)\, dS \\ \mathbb{K} = \int_\Omega (\nabla \mathbb{N})^T(x,y) k \nabla \mathbb{N}(x,y)\, dS \\ \mathbf{F}_{Q_{in}} = \int_\Omega \mathbb{N}^T(x,y) Q^{in}(x,y)\, dS \end{cases} . \tag{9}$$

Letting:

$$\begin{cases} a_i = \int_I \lambda_n^{(k-1)}(t) \dot{\lambda}_i(t)\, dt \\ b_i = \int_I \lambda_n^{(k-1)}(t) \lambda_i(t)\, dt \end{cases} \tag{10}$$

for $i = 1, \ldots, n-1$ and

$$\begin{cases} a = \int_I \lambda_n^{(k-1)}(t) \dot{\lambda}_n^{(k-1)}(t)\, dt \\ b = \int_I {\lambda_n^{(k-1)}}^2(t)\, dt \\ \mathbf{c} = \int_I \lambda_n^{(k-1)}(t) \mathbf{F}_{Q_{in}}\, dt - \sum_{i=1}^{n-1} (a_i \mathbb{M} + b_i \mathbb{K}) \mathbf{\Phi}_i \end{cases} \tag{11}$$

this problem takes the form:

$$(a\mathbb{M} + b\mathbb{K}) \mathbf{\Phi}_n^{(k)} = \mathbf{c}. \tag{12}$$

Time problem

Assuming that $\mathbf{\Phi}_n^{(k)}$ is known, one obtains $\lambda_n^{(k)}(t)$ by assuming that $\mathbf{T}_n(t) = \mathbf{T}_{n-1}(t) + \mathbf{\Phi}_n^{(k)}(t)\lambda_n^{(k)}(t)$ and by choosing a test function for the weak formulation of the problem in the form $\mathbf{T}^*(t) = \mathbf{\Phi}_n^{(k)}\lambda^*(t)$.

Based again on the work of Nouy (see [15]) and Ladevèze (see [21]), the time problem is then given by the equation:

$$\begin{aligned} \mathbf{\Phi}_n^{(k)^T}\mathbb{M}\mathbf{\Phi}_n^{(k)}\ \dot{\lambda}_n^{(k)}(t) + \mathbf{\Phi}_n^{(k)^T}\mathbb{K}\mathbf{\Phi}_n^{(k)}\ \lambda_n^{(k)}(t) = \\ \mathbf{\Phi}_n^{(k)^T}\mathbf{F}_{Q_{in}} - \sum_{i=1}^{n-1}(\mathbf{\Phi}_n^{(k)^T}\mathbb{M}\mathbf{\Phi}_i\ \dot{\lambda}_i(t) + \mathbf{\Phi}_n^{(k)^T}\mathbb{K}\mathbf{\Phi}_i\ \lambda_i(t)). \end{aligned} \tag{13}$$

Letting:

$$\begin{cases} {a_i}' = \mathbf{\Phi}_n^{(k)^T}\mathbb{M}\mathbf{\Phi}_i \\ {b_i}' = \mathbf{\Phi}_n^{(k)^T}\mathbb{K}\mathbf{\Phi}_i \end{cases} \tag{14}$$

for $i = 1, \ldots, n-1$ and

$$\begin{cases} a' = \mathbf{\Phi}_n^{(k)^T}\mathbb{M}\mathbf{\Phi}_n^{(k)} \\ b' = \mathbf{\Phi}_n^{(k)^T}\mathbb{K}\mathbf{\Phi}_n^{(k)} \\ c'(t) = \mathbf{\Phi}_n^{(k)^T}\mathbf{F}_{Q_{in}} - \sum_{i=1}^{n-1}({a_i}'\dot{\lambda}_i(t) + {b_i}'\lambda_i(t)) \end{cases} \tag{15}$$

this problem takes the form:

$$a'\dot{\lambda}_n^{(k)}(t) + b'\lambda_n^{(k)}(t) = c'(t). \tag{16}$$

This standard ordinary differential equation in time is solved here for $\lambda_n^{(k)}(t)$ by numerical integration in time making use of Backward-Euler finite difference scheme with large time increments.

Convergence criterion

The new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ is obtained after convergence of the fixed point iterations for the sequence $\mathbf{\Phi}_n^{(k)}, \lambda_n^{(k)}(t)$, based on the relative error criterion chosen as:

$$\epsilon = \frac{2\int_I (\lambda_n^{(k)}(t) - \lambda_n^{(k-1)}(t))^2\, dt}{\int_I (\lambda_n^{(k)}(t) + \lambda_n^{(k-1)}(t))^2\, dt} < tol_{PGD} \tag{17}$$

where tol_{PGD} is a small tolerance. The optimal value of tol_{PGD} is case-dependent. On one hand, a small tolerance will lead to few PDG modes at the price of many fixed point iterations. On the other hand, a large value of tol_{PGD} will need fewer fixed point iterations but more PDG couples in order to reach convergence. An in-depth analysis of this behavior will be presented in Section 4.

PGD Update step (re-computation of all time functions)

Based again on [15], once the new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ has been computed, all time functions $\{\lambda_i(t)\}_{i=1,...,n}$ are updated based on all currently known space vectors $\{\mathbf{\Phi}_i\}_{i=1,...,n}$ by requiring that:

$$\sum_{i=1}^{n} \mathbf{\Phi}_j^T \mathbb{M} \mathbf{\Phi}_i \ \dot{\lambda}_i(t) + \sum_{i=1}^{n} \mathbf{\Phi}_j^T \mathbb{K} \mathbf{\Phi}_i \ \lambda_i(t) = {\mathbf{\Phi}_j}^T \mathbf{F}_{Q_{in}} \text{ for } j{=}1,...,n \ . \tag{18}$$

Letting:

$$\begin{cases} a'_{j,i} = {\mathbf{\Phi}_j}^T \mathbb{M} \mathbf{\Phi}_i \\ b'_{j,i} = {\mathbf{\Phi}_j}^T \mathbb{K} \mathbf{\Phi}_i \\ c'_j = {\mathbf{\Phi}_j}^T \mathbf{F}_{Q_{in}} \end{cases} \tag{19}$$

this problem takes the form:

$$\sum_{i=1}^{n} a'_{j,i} \dot{\lambda}_i(t) + \sum_{i=1}^{n} b'_{j,i} \lambda_i(t) = c'_j \ \text{ for } j{=}1,...,n \ . \tag{20}$$

This coupled ordinary differential equations system in time is solved for $\{\lambda_1(t), ..., \lambda_n(t)\}$ by numerical integration making use of Backward-Euler finite difference scheme with large time increments over the entire time interval $I = [0, t_{end}]$. For the purpose of this study, the interval I was always discretized via 100 equally sized increments.

3.2. Extension to the non-linear case - non-linear materials

The extension of the FEM and PGD approaches to the non-linear case is briefly described next. PDG has seen applications in non-linear cases, mainly by the use of linearization schemes or of asymptotic numerical methods as in [22], [23] and [24] . Nevertheless, the PGD algorithm and methodology proposed in this section are an original adaptation of PGD for highly transient thermal evolution. The material parameters ρ, C_p and k are now assumed

to be prescribed functions of the temperature. Since the capacitance matrix depends on the product $\rho C_P(T)$, it becomes a function of temperature $\mathbb{M}(T)$. Likewise, since the conductivity matrix depends on $k(T)$, it becomes a function of temperature $\mathbb{K}(T)$.

For the FEM, the equation becomes:

$$\mathbb{M}(T)\dot{\mathbf{T}}(t) + \mathbb{K}(T)\mathbf{T}(t) = \mathbf{F}_{Q_{in}}, \tag{21}$$

which is solved incrementally using implicit time integration with Backward-Euler finite differences and Newton iterations (where a small tolerance Tol_{FEM} must be used) at each time increment. More details can be found in [20].

For the PGD, the non-linearities are taken into account at each stage by using all the known information about the current content of the solution. The computation of each new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ takes place again in two steps.

PGD Enrichment step

This step proceeds again by fixed point iterations.

Since

$$\mathbf{T}_n(t) = \sum_{i=1}^{n} \mathbf{\Phi}_i \lambda_i(t) = \mathbf{T}_{n-1}(t) + \mathbf{\Phi}_n \lambda_n(t), \tag{22}$$

where $\mathbf{T}_{n-1}(t)$ is not known for the first couple, one proceeds by using all the available information at each iteration.

To compute the first couple $(\mathbf{\Phi}_1, \lambda_1(t))$, the temperature field $T(x, y, t)$ is initialized to T_0 and used to compute initial values for the capacitance and conductivity matrices $\mathbb{M}_0^{(0)} = \mathbb{M}(\rho C_p(T_0))$ and $\mathbb{K}_0^{(0)} = \mathbb{K}(k(T_0))$. These matrices are then used to compute the first iterate $\{\mathbf{\Phi}_1^{(1)}, \lambda_1^{(1)}(t)\}$. After each new iteration, these matrices are updated with the temperature field of the current iteration.

Thus, after the k^{th} iteration, yielding $(\mathbf{\Phi}_1^{(k)}, \lambda_1^{(k)}(t))$, the matrices are updated according to

$$\begin{cases} \mathbb{M}_0^{(k)}(t) = \mathbb{M}(\rho C_p(\mathbf{\Phi}_1^{(k)} \lambda_1^{(k)}(t))) \\ \mathbb{K}_0^{(k)}(t) = \mathbb{K}(k(\mathbf{\Phi}_1^{(k)} \lambda_1^{(k)}(t))) \end{cases} \tag{23}$$

and used to compute the next iterate $(\mathbf{\Phi}_1^{(k+1)}, \lambda_1^{(k+1)}(t))$. This process continues until convergence of the first pair.

To compute the other couples $\{(\mathbf{\Phi}_i, \lambda_i(t))\}_{i=2,\dots,n-1}$, the temperature vector is replaced by the sum of the products of the currently known terms and the capacitance and conductivity matrices are pre-computed once based on all previous information and then stored. Thus, for the computation of the n^{th} pair $(\mathbf{\Phi}_n, \lambda_n(t))$, $\mathbf{T}(t)$ is replaced by the known quantity $\mathbf{T}_{n-1}(t) = \sum_{i=1}^{n-1} \mathbf{\Phi}_i \lambda_i(t)$, where all the time functions have been updated, and the capacitance and conductivity matrices are updated via:

$$\begin{cases} \mathbb{M}_{n-1}(t) = \mathbb{M}(\rho C_p(\mathbf{T}_{n-1}(t))) \\ \mathbb{K}_{n-1}(t) = \mathbb{K}(k(\mathbf{T}_{n-1}(t))) \end{cases} . \tag{24}$$

Aside from the special treatment for the first couple, the process is repetitive.

After initializing the process for the n^{th} couple $(\mathbf{\Phi}_n, \lambda_n(t))$ with an arbitrary time function $\lambda_n^{(0)}(t)$, each iteration consists of solving in sequence:

- A space problem for $\mathbf{\Phi}_n^{(k)}$ (with $\lambda_n^{(k-1)}(t)$ known). Letting:

$$\begin{cases} \mathbb{A}_i = \int_I \lambda_n^{(k-1)}(t) \mathbb{M}_{n-1}(t) \dot{\lambda}_i(t)\, dt \\ \mathbb{B}_i = \int_I \lambda_n^{(k-1)}(t) \mathbb{K}_{n-1}(t) \lambda_i(t)\, dt \end{cases} \tag{25}$$

  for $i = 1, \dots, n-1$ and

$$\begin{cases} \mathbb{A} = \int_I \lambda_n^{(k-1)}(t) \mathbb{M}_{n-1}(t) \dot{\lambda}_n^{(k-1)}(t)\, dt \\ \mathbb{B} = \int_I {\lambda_n^{(k-1)}}^2(t) \mathbb{K}_{n-1}(t)\, dt, \\ \mathbf{c} = \int_I \lambda_n^{(k-1)}(t) \mathbf{F}_{Q_{in}}\, dt - \sum_{i=1}^{n-1} ([\mathbb{A}_i] + [\mathbb{B}_i]) \mathbf{\Phi}_i \end{cases} \tag{26}$$

  the space problem takes the form:

$$(\mathbb{A} + \mathbb{B}) \mathbf{\Phi}_n^{(k)} = \mathbf{c}; \tag{27}$$

- A time problem for $\lambda_n^{(k)}(t)$ (with $\mathbf{\Phi}_n^{(k)}$ known). Letting:

$$\begin{cases} {a_i}'(t) = {\mathbf{\Phi}_n^{(k)}}^T \mathbb{M}_{n-1}(t) \mathbf{\Phi}_i \\ {b_i}'(t) = {\mathbf{\Phi}_n^{(k)}}^T \mathbb{K}_{n-1}(t) \mathbf{\Phi}_i \end{cases} \tag{28}$$

for $i = 1, \ldots, n-1$ and

$$\begin{cases} a'(t) = {\boldsymbol{\Phi}_n^{(k)}}^T \mathbb{M}_{n-1}(t) \boldsymbol{\Phi}_n^{(k)} \\ b'(t) = {\boldsymbol{\Phi}_n^{(k)}}^T \mathbb{K}_{n-1}(t) \boldsymbol{\Phi}_n^{(k)} \\ c'(t) = {\boldsymbol{\Phi}_n^{(k)}}^T \mathbf{F}_{Q_{in}} - \sum_{i=1}^{n-1} (a_i{}'(t) \dot{\lambda}_i(t) + b_i{}'(t) \lambda_i(t)) \end{cases} \tag{29}$$

the time problem takes the form:

$$a'(t) \dot{\lambda}_n^{(k)}(t) + b'(t) \lambda_n^{(k)}(t) = c'(t). \tag{30}$$

PGD Update step

This step re-computes all time functions. The capacitance and conductivity matrices are first updated in terms of all current information $\{(\boldsymbol{\Phi}_i, \lambda_i(t))\}_{i=1,\ldots,n}$, by assuming that $\mathbf{T}_n(t) = \sum_{i=1}^{n} \boldsymbol{\Phi}_i \lambda_i(t)$ according to:

$$\begin{cases} \mathbb{M}_n(t) = \mathbb{M}(\rho C_p(\mathbf{T}_n(t))) \\ \mathbb{K}_n(t) = \mathbb{K}(k(\mathbf{T}_n(t))) \end{cases} . \tag{31}$$

Letting:

$$\begin{cases} a'_{j,i}(t) = {\boldsymbol{\Phi}_j}^T \mathbb{M}_n(t) \boldsymbol{\Phi}_i \\ b'_{j,i}(t) = {\boldsymbol{\Phi}_j}^T \mathbb{K}_n(t) \boldsymbol{\Phi}_i \\ c'_j = {\boldsymbol{\Phi}_j}^T \mathbf{F}_{Q_{in}} \end{cases} \tag{32}$$

the update problem takes the form:

$$\sum_{i=1}^{n} a'_{j,i}(t) \dot{\lambda}_i(t) + \sum_{i=1}^{n} b'_{j,i}(t) \lambda_i(t) = c'_j \quad \text{for } j=1,...,n \ . \tag{33}$$

3.3. Extension to phase change by latent heat capacity, time- and temperature-dependent source, time and space-dependent boundary conditions

The next step is to build a model that includes the latent change of phase of the patch material located in the close vicinity of the laser source which is now allowed to move inside the patch. Realistic Neumann boundary conditions at the patch edges and a convective heat loss throughout the patch surface are also taken into account.

In SLM processes, phase changes from raw materials (e.g. powder) to liquid followed by solidification occur at very high speed ($10^{-6} - 10^{-3}$ s).

Upon melting, the local internal energy increases significantly while the temperature typically increases over a small range between T_S (solidus) and T_L (liquidus). Upon solidification, the opposite takes place: a large amount of energy is conducted/convected/radiated away from the "action" zone while temperatures decrease over the same range between liquidus and solidus. These high temperature cooling/heating rates and large temperature gradients are the dominant factors in determining at microstructural level grain morphologies and are the primary driver behind metallurgical phase transformations upon solidification. These have a direct influence on the mechanical properties of the finished product, and therefore it is critical to capture reasonably well these highly transient events (see [19]).

The change in internal energy as function of temperature can then be interpreted as a rather very non-linear specific heat relationship given by $C_p(T) = dU/dT$ as illustrated in Figure 2. Only for the sake of simplifying the illustration of this relationship, we have considered that otherwise the specific heat is independent of temperature outside of range $T_L - T_S$, a premise that is not considered in the results shown in the following section.

If the source is time-dependent, i.e. the laser source is no longer stationary, the the right-hand side of the FEM equation for the problem becomes:

$$\mathbf{F}_{Q_{in}}(t) = \int_{\Omega} \mathbb{N}^T(x,y) Q^{in}(x,y,t)\, dS. \tag{34}$$

With the impetus of global-local FEM-based modeling techniques, the ability of robustly handling arbitrary Neumann boundary conditions on local models (like the ones illustrated in this work) is mandatory. While it is not the purpose of this work to dive into the global-local coupling algorithms, if time-dependent non-homogeneous Neumann boundary conditions are imposed (on this local model), an additional term must be added to the right-hand side of the FEM equation:

$$\mathbf{F}_{\overline{q}^{out}}(t) = \int_{\partial\Omega} \mathbb{N}^T(x,y) \overline{q}^{out}(x,y,t)\, dl. \tag{35}$$

Furthermore, if an additional temperature-dependent source, representing an outward convective flux similar to a 3D film condition is taken into account:

$$Q^{out}(T) = h\ (T - T_{env}) \tag{36}$$

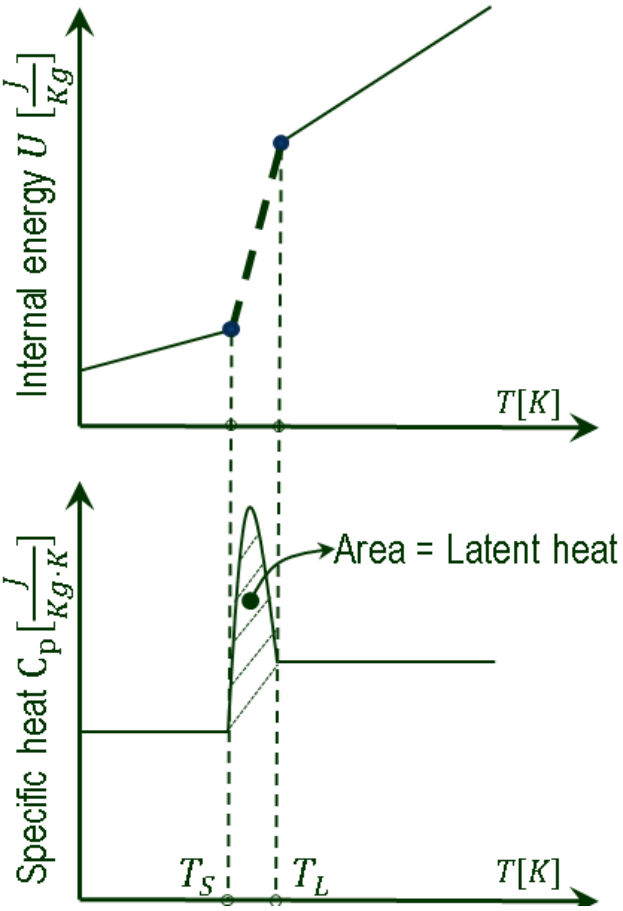


Figure 2: Internal energy for phase change and equivalent non-linear specific heat

a convective conductivity matrix must be added to the non-linear conductivity matrix:

$$\mathbb{K}_h = \int_\Omega \mathbb{N}^T(x,y)\ h\ \mathbb{N}(x,y)\, dS \tag{37}$$

and a convective source term must be added to the right-hand side of the FEM equation:

$$\mathbf{F_h}(T_{env}) = \int_\Omega \mathbb{N}^T(x,y)\ h\ T_{env}\, dS. \tag{38}$$

The FEM equation finally becomes:

$$\mathbb{M}(T)\dot{\mathbf{T}}(t) + (\mathbb{K}(T) + \mathbb{K}_h)\mathbf{T}(t) = \mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F_h}(T_{env}). \tag{39}$$

The FEM formulation remains incremental, with Newton iterations at each time increment as described in Section 3.2.

The PGD formulation also remains a succession of stages consisting of an enrichment step followed by an update step, as described in Section 3.2, but some terms are modified, as presented below.

PGD Enrichment step

In the enrichment step, a new couple $(\mathbf{\Phi}_n, \lambda_n(t))$ is again obtained with fixed point iterations by alternating sequentially between:

- The space problem, consisting of solving the algebraic system:

$$(\mathbb{A} + \mathbb{B})\mathbf{\Phi}_n^{(k)} = \mathbf{c} \tag{40}$$

with:

$$\begin{cases} \mathbb{A}_i \textit{ and } \mathbb{A} \textit{ unchanged} \\ \mathbb{B}_i = \int_I \lambda_n^{(k-1)}(t)(\mathbb{K}_{n-1}(t) + \mathbb{K}_h)\lambda_i(t)\, dt \quad \text{for } i = 1, \dots, n-1 \\ \mathbb{B} = \int_I {\lambda_n^{(k-1)}}^2(t)(\mathbb{K}_{n-1}(t) + \mathbb{K}_h)\, dt \\ \mathbf{c} = \int_I \lambda_n(t)(\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F}_{\mathbf{h}}(T_{env}))\, dt - \sum_{i=1}^{n-1}(\mathbb{A}_i + \mathbb{B}_i)\mathbf{\Phi}_i \end{cases} \tag{41}$$

- The time problem, consisting of integrating the ordinary differential equation:

$$a'(t)\dot{\lambda}_n^{(k)}(t) + b'(t)\lambda_n^{(k)}(t) = c'(t) \tag{42}$$

with:

$$\begin{cases} {a_i}'(t) \textit{ and } a'(t) \textit{ unchanged} \\ {b_i}'(t) = {\mathbf{\Phi}_n^{(k)}}^T(\mathbb{K}_{n-1}(t) + \mathbb{K}_h)\mathbf{\Phi}_i \quad \text{for } i = 1, \dots, n-1 \\ b'(t) = {\mathbf{\Phi}_n^{(k)}}^T(\mathbb{K}_{n-1}(t) + \mathbb{K}_h)\mathbf{\Phi}_n^{(k)} \\ c'(t) = \mathbf{\Phi}_n^T(\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F}_{\mathbf{h}}(T_{env})) - \sum_{i=1}^{n-1}({a_i}'(t)\dot{\lambda}_i(t) + {b_i}'(t)\lambda_i(t)). \end{cases} \tag{43}$$

PGD Update step

In the update step, the set of time functions $\{\lambda_1(t), ..., \lambda_n(t)\}$ is again simultaneously updated by integrating the ordinary differential system:

$$\sum_{i=1}^{n} a'_{j,i}\dot{\lambda}_i(t) + \sum_{i=1}^{n} b'_{j,i}\lambda_i(t) = c'_j(t) \text{ for } j = 1, \ldots, n \tag{44}$$

with:

$$\begin{cases} a'_{j,i}(t) \textit{ unchanged} \\ b'_{j,i}(t) = \mathbf{\Phi}_j{}^T(\mathbb{K}_n(t) + \mathbb{K}_h)\mathbf{\Phi}_i \\ c'_j(t) = \mathbf{\Phi_j}{}^T(\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\overline{q}^{out}}(t) + \mathbf{F_h}(T_{env})). \end{cases} \tag{45}$$

The computations for stage $n(> 1)$ in this most general case are summarized in Algorithm 1.

4. Results and Discussion

In this Section, the most relevant results are shown in order to compare PGD with FEM. All the simulations were done in Matlab. To assure the correct answers, Abaqus was used to verify FEM Matlab results. Meshes were generated by Gmsh. The coefficients a, b, c (see Equation (11)) and $\mathbb{A}$, $\mathbb{B}$, $\mathbf{c}$ (see Equations (26) and (41)) were calculated by numerical integration using a rectangular method with 100 discretization points in all the following examples.

4.1. Example 1: linear case

Even though no computational gain is expected in the linear case, it is a good starting point to better understand the PGD behavior when applied to an additive manufacture problem.

The domain to be analyzed consists of a 2 $mm \times 2$ mm square plate made of Ti-6Al-4V. The laser is stationary and heats the plate during 0.1 s. The walls are insulated and there are no Dirichlet boundary conditions. A non-uniform triangular mesh with 1894 degrees of freedom was used, with an element size of 10^{-6} m in the laser region and 10^{-4} m in the edges.

Further information is presented in Table 1 and in Figure 3.

Some of the cases tested are presented in Table 2 and Figure 4. Note that in all cases the temperature is analyzed at the node where the laser is located.

Algorithm 1 Non-linear PGD

Require: $\mathbb{M}_{n-1}(t)$, $\mathbb{K}_{n-1}(t)$, $\mathbb{K}_h$, $\mathbf{F}_{Q^{in}}(t)$, $\mathbf{F}_{\bar{q}^{out}}(t)$, $\mathbf{F}_h(T_{env})$, tol_{PGD}, $(\mathbf{\Phi}_{n-1}, \lambda_{n-1}(t))$

for $n = 2$ to N_{mod} **do**

Enrichment step (Fixed-point iterations) to **Compute** $(\mathbf{\Phi}_n, \lambda_n(t))$

Initialize $\lambda_n^{(k=0)}$(t)

while $\epsilon = \frac{\int_I (\lambda_n^{(k)}(t) - \lambda_n^{(k-1)}(t))^2 \, dt}{\frac{1}{2}\int_I (\lambda_n^{(k)}(t) + \lambda_n^{(k-1)}(t))^2 \, dt} > tol_{PGD}$ **do**

$k = k + 1$

Space problem: Compute $\mathbf{\Phi}_n^{(k)}$ from a known $\lambda_n^{(k-1)}(t)$

$$\begin{cases} \mathbb{A}_i = \int_I \lambda_n^{(k-1)}(t) \mathbb{M}_{n-1}(t) \dot{\lambda}_i(t) \, dt \\ \mathbb{B}_i = \int_I \lambda_n^{(k-1)}(t) (\mathbb{K}_{n-1}(t) + \mathbb{K}_h) \lambda_i(t) \, dt \end{cases}$$

for $i = 1, \ldots, n-1$ and

$$\begin{cases} \mathbb{A} = \int_I \lambda_n^{(k-1)}(t) \mathbb{M}_{n-1}(t) \dot{\lambda}_n^{(k-1)}(t) \, dt \\ \mathbb{B} = \int_I {\lambda_n^{(k-1)}}^2(t) (\mathbb{K}_{n-1}(t) + \mathbb{K}_h) \, dt, \\ \mathbf{c} = \int_I \lambda_n^{(k-1)}(t) (\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F_h}(T_{env})) \, dt - \sum_{i=1}^{n-1} ([\mathbb{A}_i] + [\mathbb{B}_i]) \mathbf{\Phi}_i \end{cases}$$

Solve for $\mathbf{\Phi}_n^{(k)}$: $(\mathbb{A} + \mathbb{B}) \mathbf{\Phi}_n^{(k)} = \mathbf{c}$

Time problem: Compute $\lambda_n^{(k)}(t)$ from a known $\mathbf{\Phi}_n^{(k)}$

$$\begin{cases} {a_i}'(t) = {\mathbf{\Phi}_n^{(k)}}^T \mathbb{M}_{n-1}(t) \mathbf{\Phi}_i \\ {b_i}'(t) = {\mathbf{\Phi}_n^{(k)}}^T (\mathbb{K}_{n-1}(t) + \mathbb{K}_h) \mathbf{\Phi}_i \end{cases}$$

for $i = 1, \ldots, n-1$ and

$$\begin{cases} a'(t) = {\mathbf{\Phi}_n^{(k)}}^T \mathbb{M}_{n-1}(t) \mathbf{\Phi}_n^{(k)} \\ b'(t) = {\mathbf{\Phi}_n^{(k)}}^T (\mathbb{K}_{n-1}(t) + \mathbb{K}_h) \mathbf{\Phi}_n^{(k)} \\ c'(t) = {\mathbf{\Phi}_n^{(k)}}^T (\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F_h}(T_{env})) - \sum_{i=1}^{n-1} ({a_i}'(t) \dot{\lambda}_i(t) + {b_i}'(t) \lambda_i(t)) \end{cases}$$

Solve for $\lambda_n^{(k)}(t)$: $a'(t) \dot{\lambda}_n^{(k)}(t) + b'(t) \lambda_n^{(k)}(t) = c'(t)$

Check ϵ

if $\epsilon \leq tol_{PGD}$ **then**

Set: $(\mathbf{\Phi}_n, \lambda_n(t)) = (\mathbf{\Phi}_n^{(k)}, \lambda_n^{(k)}(t))$

end if

end while

Update step (Re-computation of all time functions $\{\lambda_1(t), ..., \lambda_n(t)\}$)

$$\begin{cases} a'_{j,i}(t) = {\mathbf{\Phi}_j}^T \mathbb{M}_n(t) \mathbf{\Phi}_i \quad \text{for } i = 1, \ldots, n \\ b'_{j,i}(t) = {\mathbf{\Phi}_j}^T (\mathbb{K}_n(t) + \mathbb{K}_h) \mathbf{\Phi}_i \quad \text{for } i = 1, \ldots, n \\ c'_j(t) = {\mathbf{\Phi_j}}^T (\mathbf{F}_{Q_{in}}(t) - \mathbf{F}_{\bar{q}^{out}}(t) + \mathbf{F_h}(T_{env})). \end{cases}$$

Solve for $\{\lambda_1(t), ..., \lambda_n(t)\}$: $a'_{j,i} \dot{\lambda}_i(t) + b'_{j,i} \lambda_i(t) = c'_j(t)$ for i=1,...,n

end for

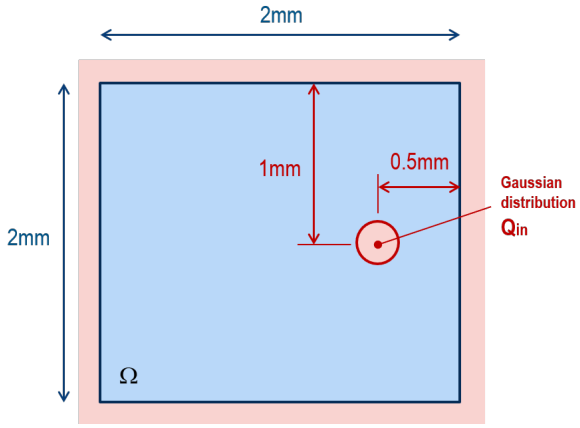
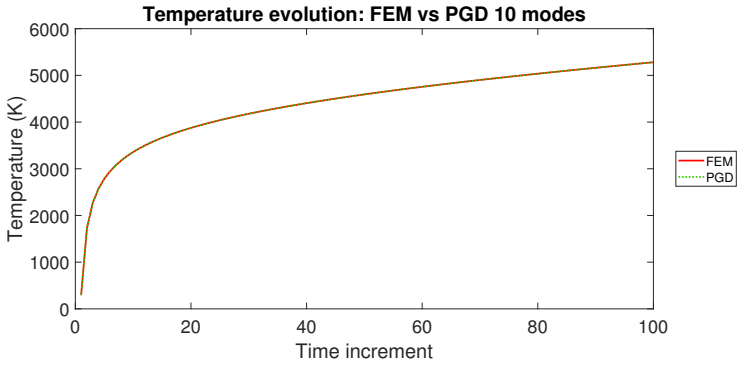


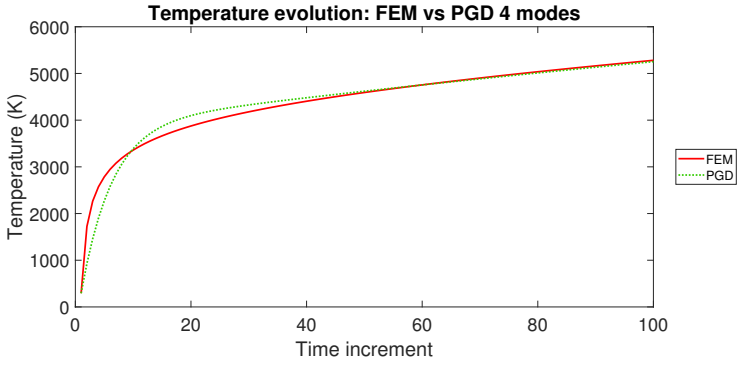
Figure 3: Representative schema of Example 1

Table 1: General properties for Example 1

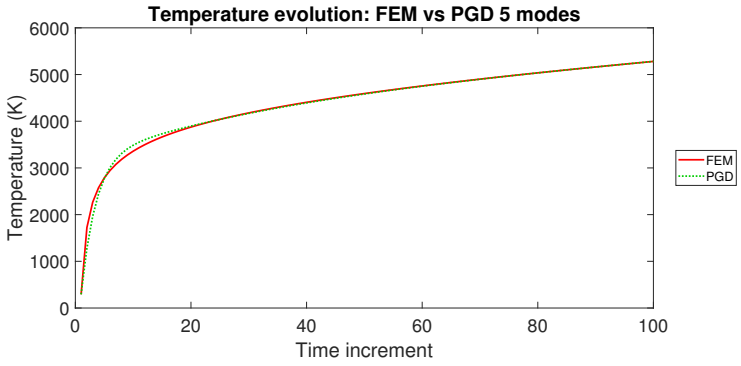
General properties	**Values**
Density (ρ)	4500 kg/m^2
Conductivity (k)	12 W/K
Heat capacity (C_p)	700 $J/K/kg$
Laser power (P)	100 kW
Laser radius (R_0)	$5 \cdot 10^{-5}$ m
Total time (t_{end})	0.1 s
Time increment (Δt)	0.001 s



(a) PGD 10 modes, tol_{PGD}= 10^{-8}



(b) PGD 4 modes, tol_{PGD}= 10^{-2}



(c) PGD 5 modes, 1 iteration per mode

Figure 4: Comparison between PGD and FEM at the laser node for Example 1

Table 2: PGD performance at the laser node for Example 1

Simulation				
Type	Modes	Error	Criterion	Solve Total
PGD	4	0.0327	1 iteration/mode	4
PGD	4	0.0280	tol_{PGD}=0.01	8
PGD	5	0.0098	1 iteration/mode	5
PGD	5	0.0092	tol_{PGD}=0.01	9
PGD	10	$3.31{\cdot}10^{-5}$	tol_{PGD}=10^{-8}	40

Here:

$$\text{Error} = \frac{\|T_{FEM} - T_{PGD}\|_{L^2(\Omega\times[0,t_{end}])}}{\|T_{FEM}\|_{L^2(\Omega\times[0,t_{end}])}} \tag{46}$$

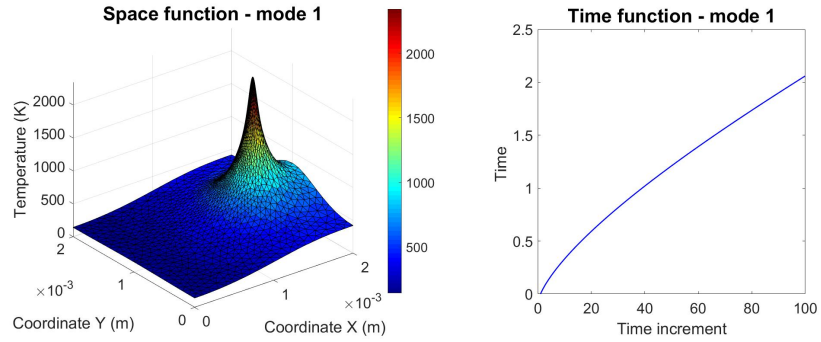
is the expression used to compute the error. It measures the difference between FEM and PGD results according to the L^2 norm, normalized by the L^2 norm of the FEM answer. Solve Total represents the number of times a linear system was solved by matrix inversion.

Comparing the accuracy of the PGD results for various numbers of retained modes and for various numbers of iterations per mode, we can make some observations. First, very few iterations are needed per mode. The error criterion can be replaced by a strategy with 1 iteration per mode. This shows that the method is self-correcting. Besides, since each iteration requires the resolution of a potentially large algebraic system (along with the integration of an ODE), for a given number of matrix factorizations, it is more advantageous to increase the number of modes with just a few (one or two) iterations per mode than to increase the number of iterations per mode with a lower number of computed modes. This is illustrated by comparing Figures 4b (PGD 4 modes, tol_{PGD}=0.01) and 4c (PGD 5 modes, 1 iteration per mode). Both represent an improvement of the PGD 4 modes, 1 iteration per mode solution. But, in the first case, it is necessary 8 iterations for an error equals to 0.028, while for the second one, less iterations (5) give better result (error = 0.0098).

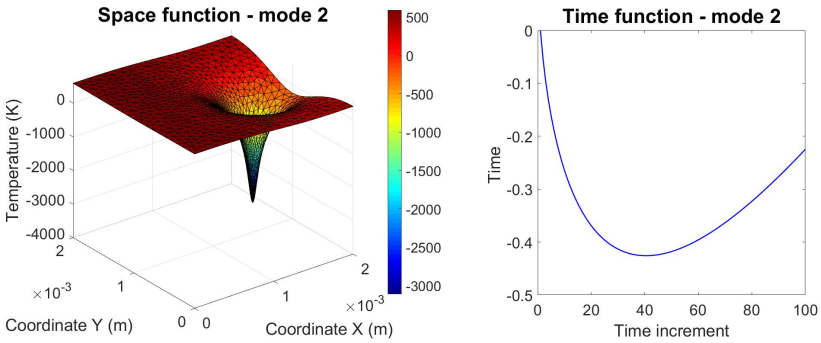
Furthermore, Figure 4a confirms that as the number of modes increases, the PGD solution converges to the FEM solution.

Finally, to illustrate the principle of the PGD, a few couples of basis vectors and time functions are presented in Figure 5. The basis vectors have

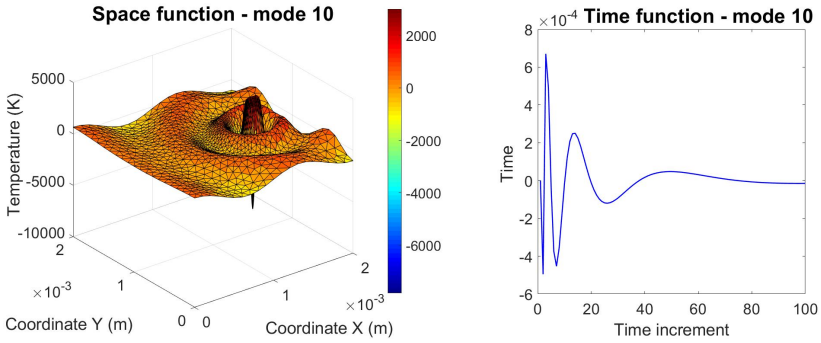
been normalized with respect to the capacitance matrix in order to maintain consistent orders of magnitude for the time functions. The time functions were initialized by assuming a linear time dependence.



(a) Space and time functions for the first PGD mode



(b) Space and time functions for the second PGD mode



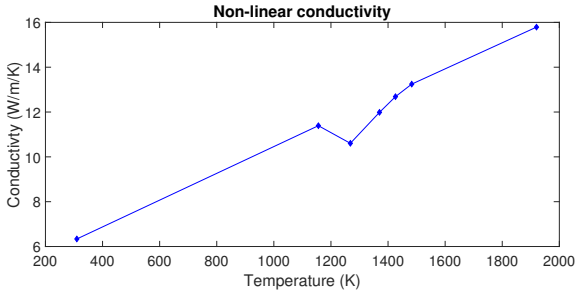
(c) Space and time functions for the tenth PGD mode

Figure 5: PGD space and time functions

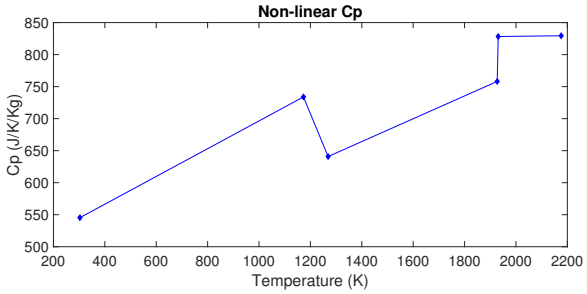
4.2. Example 2: non-linear materials

As expected, the PGD-computed response is very accurate in the linear cases. So, in order to increase the complexity of the model, non-linearities

must be introduced. First, we will consider material non-linearities, i.e. temperature-dependent conductivity and heat capacity. These values were obtained in [25] and they are shown in Figure 6.



(a) Conductivity as function of the temperature



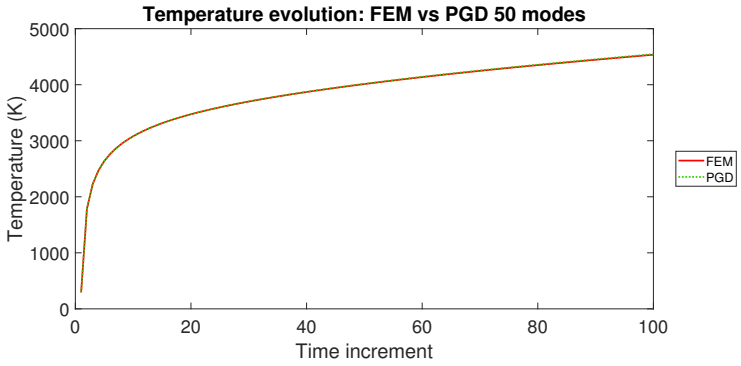
(b) Thermal capacity as function of the temperature

Figure 6: Non-linear material properties for Ti-6Al-4V

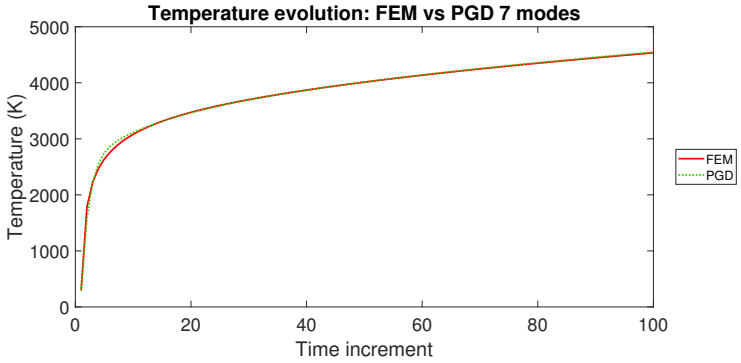
Several tests were run, varying the number of modes and convergence criterion. The results are presented in Figure 7 and Table 3, again at the node where the laser is located.

The PGD error measure is the same as in Equation (46). The FEM tolerance, on the other hand, is chosen so as to yield a reference FEM solution of the same precision when compared to Abaqus. The reported Speed-up Factor is the ratio of the total number of solves needed for our FEM solution by the total number of solves needed for our PGD solution. Due to the fact that Abaqus requires only roughly half the number of solves per increment, a more conservative way to measure the PGD gain would be to divide the reported Speed-up Factor by two.

First of all, the 50 modes PGD solution (Figure 7a) is very accurate, confirming the assumption that if the number of PGD modes is large enough,



(a) PGD 50 modes, 2 iterations per mode



(b) PGD 7 modes, 2 iterations in the 1^{st} mode and 1 iteration in the others

Figure 7: Performance comparison between PGD and FEM at the laser node for Example 2

the PGD solution tends towards the reference solution, even when some non-linearities are introduced.

Moreover, regarding the modes convergence criterion, the same conclusions as in the linear case can be applied here. It is possible, and preferable, to impose a fixed number of iterations per mode instead of using a tolerance. For example, one can compare PGD 6 modes with tol_{PGD}=0.01 and PGD 7 modes with 2 iterations for the first mode and 1 iteration for the others. In the first case, the total number of iterations is not known a priori. The error ϵ from Equation (17) must be smaller than 0.01 in order to calculate the next mode. In the second case, the number of iterations of a specific mode is imposed, and Equation (17) is ignored. One concludes that a few iterations suffice, since the first case has more iterations (meaning larger computational

Table 3: Performance comparison between PGD and FEM at the laser node for example 2

Simulation				Solve	
Type	Modes	Error	Criterion	Total	Speed-up Factor
FEM	-	-	Tol_{FEM}=0.0001	658	-
PGD	5	0.0168	2 it. in the first mode and 1 it. in the others	6	109.67
PGD	6	0.0077	tol_{PGD}=0.01	17	38.71
PGD	6	0.0083	2 it. in the first mode and 1 it. in the others	7	94.00
PGD	7	0.0070	tol_{PGD}=0.01	19	34.63
PGD	7	0.0063	2 it. in the first mode and 1 it. in the others	8	82.25
PGD	8	0.0059	2 it. in the first mode and 1 it. in the others	9	73.11
PGD	50	0.0025	tol_{PGD}=0.01	105	6.26

costs) and its results are less accurate. 2 iterations for the first mode and 1 iteration for the others is the minimum necessary number of iterations, since the first mode has a larger contribution to the final result - so it must have a special treatment - and each new mode seems to correct partially the flaws of previous modes. If a given result is not precise enough, it is preferable to add an extra mode (with just one iteration) than to perform more iterations.

Finally, PGD has responded well to the non-linear case. Satisfactory results were obtained with just a few iterations, as shown for instance in Figure 7b.

4.3. Example 3: non-linear material properties, latent heat and moving laser

From Example 2, it was observed that PGD responded well to the first non-linearities. So, one can increase the sources of non-linearities to investigate the limits of PGD. The idea is to introduce a moving torch for the laser and to take into account the latent heat and the melting pools.

The properties for this problem are shown in Table 4 and in Figures 9 and 6a. Some modifications were made in the domain as illustrated in Figure 8.

It is still a 2 $mm \times 2$ mm square patch, with the origin located at the bottom left corner and with the laser starting at point 5 located at position (1.5,1) mm. However, new points were introduced (points 6 to 10) to represent the path of the moving laser. At the instant $t = 0$ s (beginning of the simulation), the laser is at point 5 and starts moving during 1 ms with a speed of 0.5 m/s in the direction of point 10, located at position (0.5,1) mm. Afterwards, the laser is turned off ($Q_{in} = 0$), and cooling takes place during 1 ms. With this approach, it is possible to see the latent heat effects when the material undergoes phase transitions both from solid to liquid and from liquid to solid.

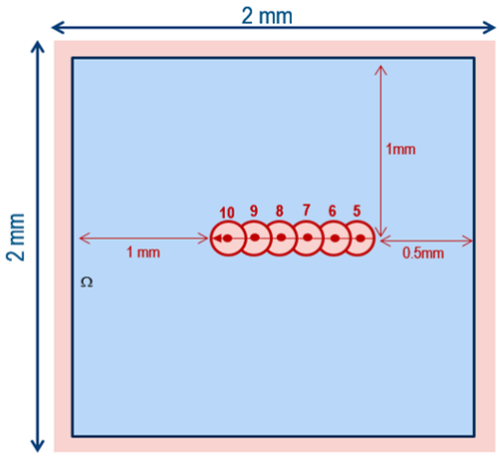


Figure 8: Representative schema of Example 3

Table 4: General properties for Example 3

General properties	**Values**
Density (ρ)	4500 kg/m^2
Laser power (P)	460 kW
Laser radius (R_0)	$5 \cdot 10^{-5}$ m
Torch speed (v)	0.5 m/s ($v_x = -0.5$ m/s , $v_y = 0$ m/s)
Laser initial position	$X_0 = 1.5$ mm and $Y_0 = 1.0$ mm
Total time (t_{end})	2 ms (1 ms with the laser on and 1 ms of cooling)
Time increment (Δt)	$2 \cdot 10^{-5}$ s

In addition, a non-uniform mesh is used. It has 1208 degrees of freedom,

with an element size of 10^{-4} m at the edges and 10^{-5} m along the laser path. The element size must be smaller than the laser radius in the path region, otherwise the heating effect caused by the laser is not well captured. Nevertheless, far from the laser path, there is no need to have a refined mesh, so it is possible to have coarser elements to save computational time.

4.3.1. Latent heat

The approach used for modeling latent heat is explained in Section 3.3. The phase transformation from solid to liquid and vice-versa occurs for Ti-6Al-4V over a narrow range of temperatures (of about 50 K) centered around 1920 K. In the finite element formulation, in order to achieve reasonable convergence when traversing this strong non-linearity, we chose to widen the phase transformation interval using the parameters presented in Table 5. Previous studies (see [26], [19]) suggest that such choice has almost no detrimental effect when numerical results are a compared to physical tests including melt pool in plane dimensions and depth. For consistency a similar approach is adopted in the PGD formulation.

Parameter	**Value**
$\mathcal{L}$ (latent heat)	440 kJ/kg
T_S (modified solid temperature)	1653 K
T_L (modified liquid temperature)	2153 K

Table 5: Latent heat parameters

In order to obtain a smooth variation of the total heat capacity, the latent heat capacity (henceforth C_λ) is interpolated using a fourth order degree polynomial:

$$C_\lambda = \begin{cases} 30(\frac{T-T_S}{T_L-T_S})^2(1-(\frac{T-T_S}{T_L-T_S}))^2\frac{\mathcal{L}}{T_L-T_S} \text{ if } T_S < T < T_L \\ 0 \text{ otherwise} \end{cases} \tag{47}$$

Adding the C_λ values from Equation (47) to the previous C_p defined in Figure 6b, one obtains the total C_p which will be used in the current analysis (Figure 9).

4.3.2. Moving source

The source is now allowed to move within the patch in the negative x-direction with speed $v = v_x$, so that $f_x(t) = v_x$ and $f_y(t) = 0$ (see Equations

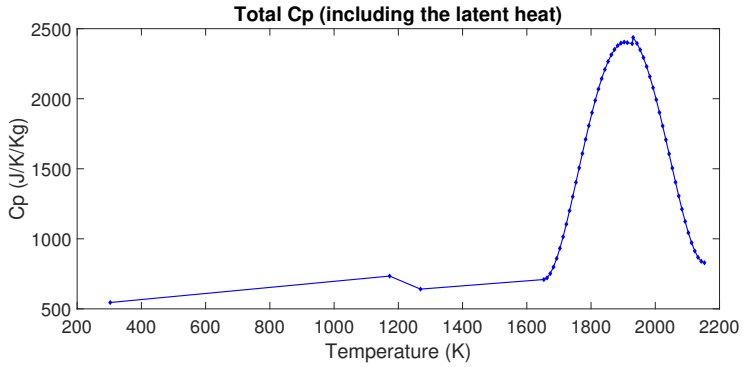


Figure 9: Total heat capacity (including latent heat) as a function of the temperature for Ti-6Al-4V

(2) and (3)). Consequently, the moving source expression becomes:

$$Q^{in}(x,y,t) = \frac{2P}{\pi R_0^2} \exp\left(\frac{-2}{R_0^2}((x-(X_0+\upsilon_x t))^2 + (y-Y_0)^2)\right). \tag{48}$$

As the source travels along the horizontal direction, the regions of high temperature are expected to follow its motion.

4.3.3. Results and Discussion for Example 3

Before comparing PGD and FEM results, it is interesting to better understand the effects of the latent heat. Figure 10 shows how the latent heat leads to a significant reduction of the temperature. Both curves (with and without latent heat) coincide as long as $T < T_S$. When the temperature reaches the value $T = T_S$, there is a decrease in the slope of the curve with latent heat, and the curves no longer coincide. However, when $T > T_L$, the slope of the curve with latent heat increases again and the curves become nearly parallel.

The results of the simulation are illustrated in Figure 11 and 12 and in Table 6.

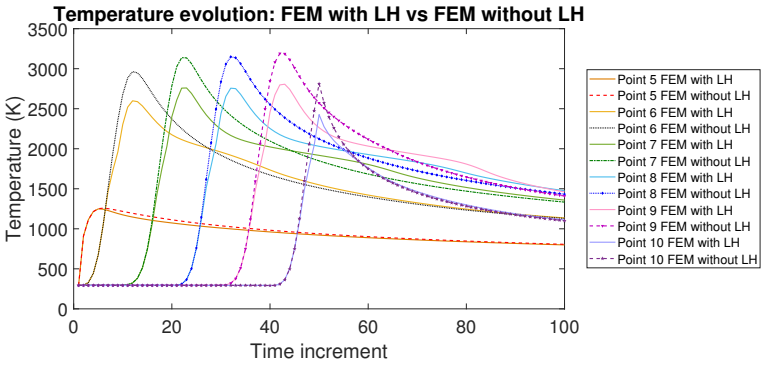
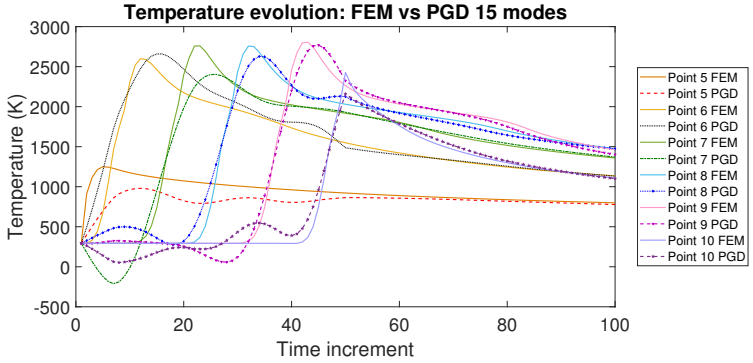
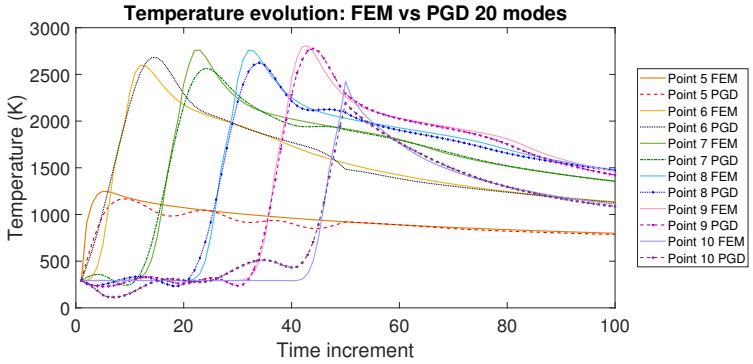


Figure 10: Comparison between FEM with latent heat and FEM without latent heat

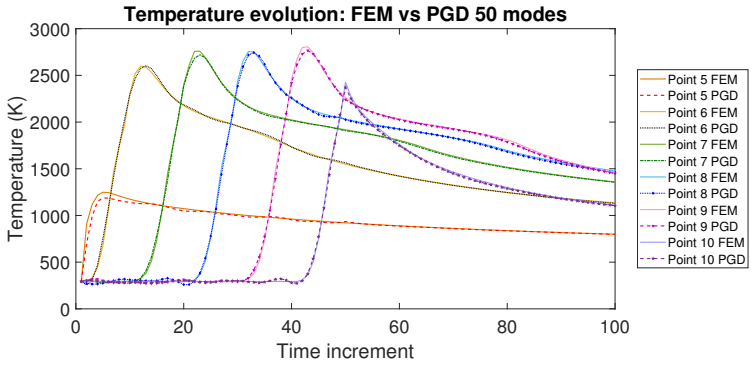


(a) PGD 15 modes, 2 iterations per mode

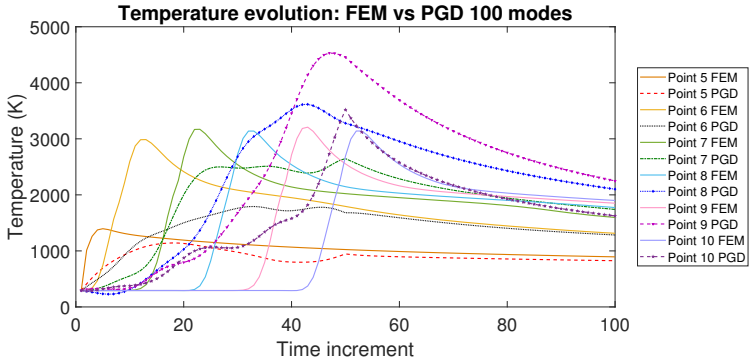


(b) PGD 20 modes, 2 iterations per mode

Figure 11: Comparison between PGD and FEM for Example 3 for a low number of modes



(a) PGD 50 modes, 2 iterations per mode



(b) PGD 100 modes, 3 iterations in the 1^{st} mode and 1 iteration in the others

Figure 12: Comparison between PGD and FEM for Example 3 for a high number of modes

Table 6: Performance comparison between PGD and FEM for Example 3

Simulation					Solve	
Type	Modes	Error 1	Error 2	Criterion	Total	Speed-up Factor
FEM	-	-	-	Tol_{FEM}=0.0001	994	-
PGD	10	0.0999	0.0642	2 it./mode	20	49.70
PGD	15	0.0685	0.0434	2 it./mode	30	33.13
PGD	20	0.0299	0.0190	2 it./mode	40	24.85
PGD	25	0.0264	0.0168	2 it./mode	50	19.88
PGD	30	0.0196	0.0125	2 it./mode	60	16.57
PGD	50	0.0070	0.0045	2 it./mode	100	9.94
PGD	100	0.1795	0.1206	3 it. in the 1st mode and 1 it. in the others	102	9.75

Here:

$$\text{Error 1} = \frac{\|T_{FEM} - T_{PGD}\|_{L^2(\Omega\times[0,t_{end}/2])}}{\|T_{FEM}\|_{L^2(\Omega\times[0,t_{end}/2])}} \tag{49}$$

is the error for the first half time of the simulation, where the laser is turned on and travels from point 5 to point 10, and

$$\text{Error 2} = \frac{\|T_{FEM} - T_{PGD}\|_{L^2(\Omega\times[0,t_{end}])}}{\|T_{FEM}\|_{L^2(\Omega\times[0,t_{end}])}} \tag{50}$$

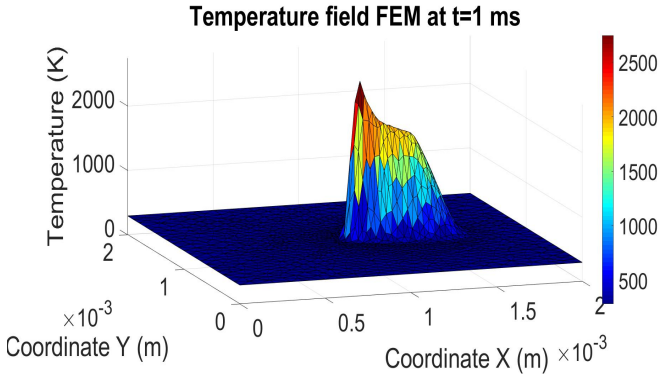
is the error for the total duration of the simulation, i.e., the laser on and the cooling (laser off) phases. Notice that the temperature gradients are higher in the laser "on" scenario than in the cooling scenario, meaning that relative global discrepancies between PGD and FEM results are expected to be larger in the first half of the simulation . This justifies the fact that Error 1 will always be larger than Error 2 in this example.

As can be seen from these curves, the response behavior is in accordance with expectations and shows good correlation between FEM and PGD results provided the number of retained modes is sufficient.

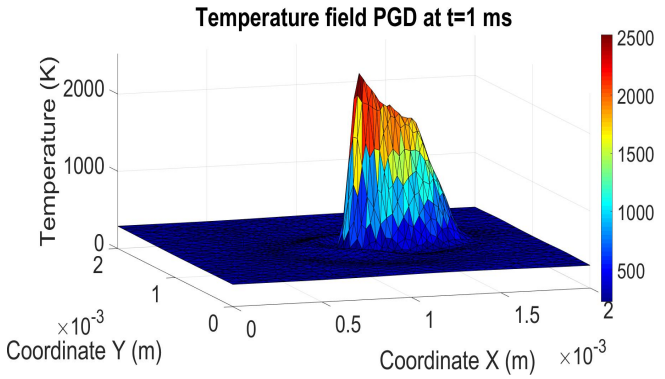
Important points can be highlighted from this example. First of all, in contrast with the case of a stationary source, the solution requires more than 1 iteration per mode, even when using several modes. For example, using 100 modes with 3 iterations in the first mode and 1 iteration in the others

(Figure 12b), the result is very different from the FEM answer and totally unacceptable. Nevertheless, when using 50 modes with 2 iterations per mode (Figure 12a), the result accuracy is excellent. In both cases, there are about 100 iterations in total, but in the first case, although more modes are used, the result is worse than in the second case. This might seem contradictory with the previous results and with the PGD assumptions. However, in the case of a moving laser, the higher modes have an important contribution to the response due to the non-separability of time and space. As the new expression for Q_{in} is such that it is not possible to separate time and space - due to the terms $v_x t$ and $v_y t$ in the exponential - the higher order modes are no longer mere corrections, but contribute as importantly to the solution as the lower ones. Since higher modes are very relevant for the solution in this new case, the convergence criterion needs to be more strict in order to obtain an accurate PGD solution. This explains the difference between 50 and 100 modes PGD solutions. Besides, it establishes that at least 2 iterations are necessary for each mode to have an acceptable solution. In fact, one can consider that 2 iterations per mode is the ideal mode convergence criterion for this example, and a larger number of iterations could add numerical pollution . Adding an extra mode is again better than adding more iterations per mode.

Moreover, the number of modes and iterations required to correctly capture the diffusion effect is much larger than for a stationary source, even though, with 20 modes, PGD represents a performance gain of 24.85 times compared to FEM. Qualitatively the result is quite acceptable as the very large temperature gradients, peak temperatures and overall temperature distributions are captured reasonably well, as shown in Figure 13.



(a) FEM

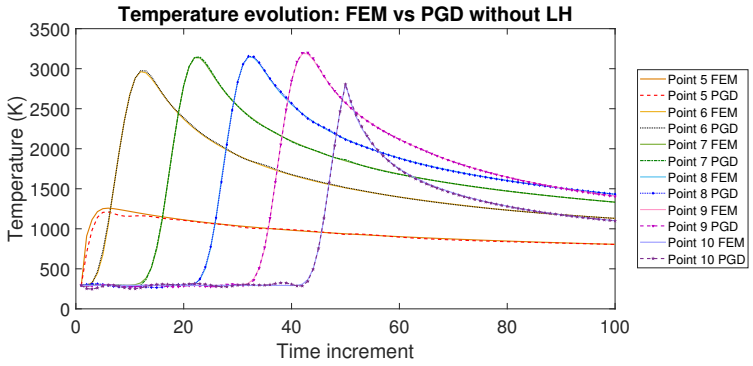


(b) PGD 20 modes 2 iterations per mode

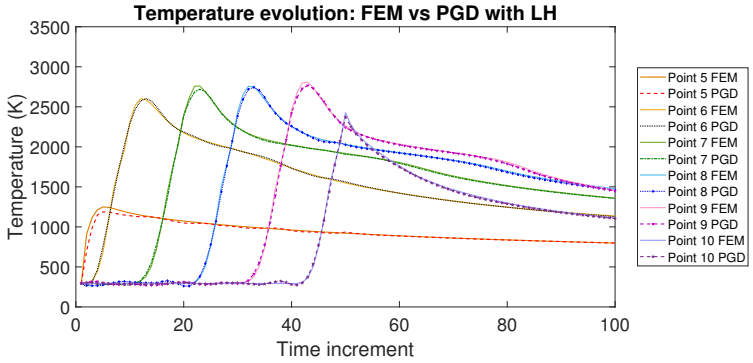
Figure 13: Temperature field in all the domain at t=1 ms for FEM and PGD

Summarizing the results of Sections 4.3 and 4.2, one can observe the influence of the latent heat with a moving laser on the solution in Figure 14.

Figures 14a and 14b support the verification of PGD, showing it is able to handle problems with moving source and latent heat, and the larger the number of modes, the better the solution (here 50 modes and 2 iterations per mode were used). One can note that the PGD solution without latent heat is more accurate than the one with latent heat. This is consistent, since the latent heat introduces a very strong non-linearity to the problem.



(a) Comparison between FEM and PGD without latent heat



(b) Comparison between FEM and PGD with latent heat

Figure 14: FEM and PGD behaviors for problems with and without latent heat

4.4. Example 4: non-linear material properties, latent heat, moving laser, time and space-dependent Neumann boundary conditions and convection

This is the final example that encompasses most different types of non-linearities a thermal problem in the field of additive manufacturing can have, by adding a film condition and time and space-dependent Neumann boundary conditions to the previous example. The notable exception is radiative heat loss which will be subject of future work.

The properties for this problem are the same as in Example 3, so they can be found in Table 4 and in Figures 9 and 6a. However, it was again necessary to modify the domain (to be explained later). The patch is now represented by a 1 mm x 0.1 mm rectangular uniform mesh (this region will be also called the small patch), with the origin at the bottom left corner and with triangular elements of size 10^{-5} m, totalizing 1378 degrees of freedom.

The source motion is the same as in the previous case: at the beginning of the simulation (t=0), the laser is at point 5 (now located at (0.85,0.05) mm). Then it starts moving towards point 10, traveling a distance of 0.5 mm with a constant speed of 0.5 m/s. Once arrived at point 10, the laser is turned off, and the cooling phenomenon is observed during 1 ms.

The previous description is illustrated in Figure 15.

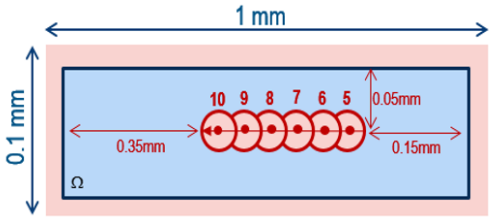


Figure 15: Representative schema of Example 4

4.4.1. Neumann Boundary Conditions

In all previous examples, it was considered that the domain was thermally insulated, so the Neumann boundary conditions were zero everywhere. However, in reality, not all the laser heat received remains inside the model. A fraction of the heat is lost through the edges due to non-homogeneous Neumann boundary conditions. In order to determine realistic Neumann boundary conditions (to be applied to our model in Example 4), a specific approach was used.

The procedure starts with the computation of the non-linear problem response (i.e., non-linear k, non-linear C_p, latent heat, convection, moving laser and insulated boundaries) in a larger patch. A 2 mm x 2 mm square domain (henceforth the super-patch), totally overlapping the small patch was created and used in a first simulation. It was assumed that the influence of the diffusion of the laser heat on the edges of the super-patch during the first 2 ms is negligible, so the insulated walls assumption is more realistic for this super-patch.

The second step was the extraction of the space and time-dependent outflux, that is represented in Figure 16. 30 points at and near the boundaries of the small patch (labeled "point 11" to "point 41") were created. For each pair of points (for example points 12 and 13), the heat flux was computed at each time increment. The result is a vector containing the time evolution of the heat flux for a specific point in space. With this methodology, one can

extract the time and space-dependent Neumann boundary conditions for the small patch. The criterion for choosing the width of the small patch equal to 0.1 mm was that 0.1 $Q_{in} < Q_{out} < 0.5\ Q_{in}$, so the influence of non-zero Neumann boundary conditions is significant.

The final step was to apply at the boundaries of the small patch the previously computed flux and to compute the new response.

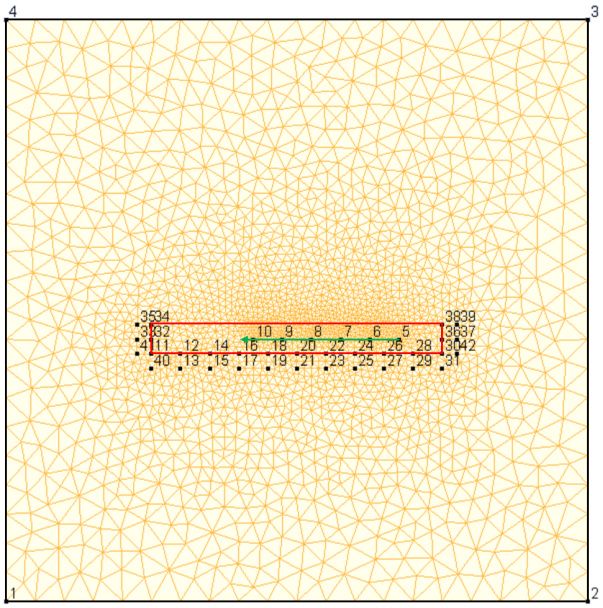


Figure 16: Approach to extract the space and time-dependent Neumann boundary conditions

The small patch is represented by the red rectangle and is exactly the same as in Figure 15. The green arrow is the laser trajectory. The super-patch is represented by the black square which contains the red rectangle. To save computation time, the super-patch was meshed non-uniformly. The element sizes are $10^{-4}\ m$ near the black edges, $2.5 \cdot 10^{-5}\ m$ near the red edges and $10^{-5}\ m$ along the laser path.

4.4.2. Surface convection (film condition)

In order to take into account heat exchanges between the model and the surrounding environment (air), a film condition was applied to the entire surface Ω - (not only redat the edges ($\partial\Omega$)). Thus, the film condition was treated as a body source rather than as a boundary condition. Its expression

is given by:

$$-\mathbf{n}^{out} \cdot k\nabla T = h(T - T_{env}) \tag{51}$$

where h= 18 $W/K/m$ (see [27]) is the heat transfer coefficient for the air-titanium interaction and T_{env}= 293 K is the air temperature, here considered the same as the room temperature.

This approach is an attempt to be as realistic as possible, since we are representing a 2D top view model of a 3D problem.

4.4.3. Results and Discussion

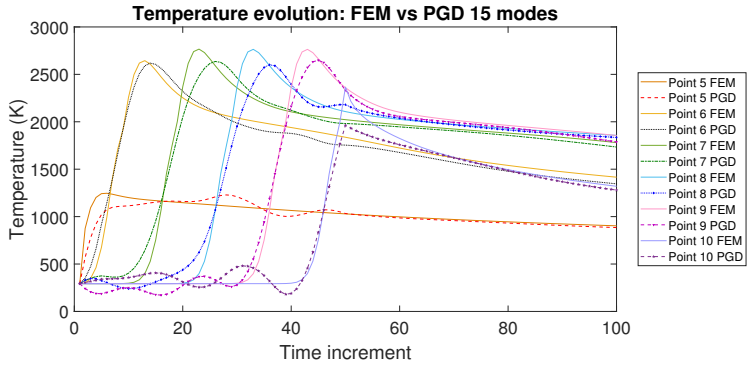
The simulation was done for FEM and for PGD with different numbers of modes (always with 2 iterations per mode). The results are shown in Figure 17 and in Table 7. Points 5 to 10 were chosen in order to analyze the temperature behavior. Because the laser crosses these points, the changes in temperature are highly transient, and so they are good candidates for comparing FEM with PGD.

Table 7: Performance comparison between PGD and FEM for Example 4

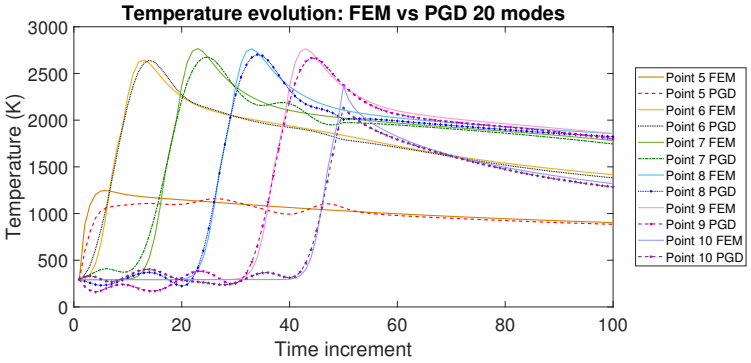
Simulation					Solve	
Type	Modes	Error 1	Error 2	Criterion	Total	Speed-up Factor
FEM				Tol_{FEM}=0.0001	880	
PGD	10	0.0827	0.0519	2 it./mode	20	44.00
PGD	15	0.0427	0.0274	2 it./mode	30	29.33
PGD	20	0.0355	0.0223	2 it./mode	40	22.00
PGD	25	0.0170	0.0121	2 it./mode	50	17.60
PGD	30	0.0147	0.0105	2 it./mode	60	14.67
PGD	50	0.0105	0.0098	2 it./mode	100	8.80

Here, Error 1 and Error 2 are the same as defined in Equations (49) and (50), respectively.

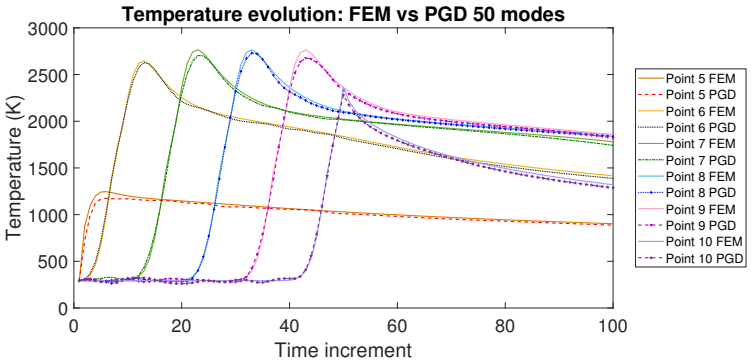
As expected, at least 2 iterations are needed per mode, because higher modes are important to correctly capture the effect of the traveling laser and time-varying Neumann boundary conditions. The result with 50 modes (Figure 17c) is extremely accurate, however the performance gain is not very high in this case. 10 or 15 modes (Figure 17a) are not sufficient, because the accuracy in the high temperature zone is poor. 20 modes (Figure 17b)



(a) PGD 15 modes, 2 iterations per mode



(b) PGD 20 modes, 2 iterations per mode



(c) PGD 50 modes, 2 iterations per mode

Figure 17: Performance comparison between PGD 50 modes and FEM for Example 4

seems a good compromise, even though there are some oscillations at the beginning of the simulation for the PGD response. The accuracy in the most

important regions (such as high temperatures or cooling) is acceptable and the number of matrix inversions in this case is reduced by 22 compared to the FEM. As the out-flux represents a considerable percentage of the laser energy, there is a remarkable drop in the temperature when compared to Example 3. However, this does not seem to be a problem for PGD. Adding the time and space-dependent Neumann boundary conditions and the film convection does not change the number of modes necessary for a reasonable PGD response. Previously, with 20 modes and 2 iterations per mode one obtained an error $_1 = 0.0299$ and an error $_2 = 0.0190$. Thus, although the errors have slightly increased, the choice of 20 modes seems a reasonable compromise. Finally, this example proves that PGD can handle highly non-linear and complex problems.

5. Conclusion

In this paper, a PGD model order reduction technique has been applied to the numerical model of highly transient non-linear thermal phenomena associated with melt pools in additive manufacturing powder bed fabrication. In previous work we have found that with minimal calibration, parabolic PDEs associated with transient heat transfer equations are sufficiently accurate/predictive and hence we have focused this work in that context.

After a brief introduction and setting the objectives in Section 1, the specific problem benchmark in this work is outlined in Section 2. Linear and non-linear PGD theory for transient heat transfer equations has been carefully described in Section 3. To develop a realistic AM thermal model, most relevant non-linearities associated with the parabolic PDE, such as material property non-linearities, phase change by latent heat, time-dependent source, temperature-dependent source and time and space-dependent boundary conditions have been introduced. Finally, Section 4 presents an extensive numerical campaign to validate the PDG-based approach against traditional FEM while thoroughly assessing the execution performance gains.

As demonstrated by these tests, model order reduction techniques such as PGD appear to be a very promising lead when tackling highly non-linear numerical simulation challenges in AM. Two key aspects should be highlighted: 1) the possibility to considerer several high complexity non-linearities in PGD (key for realistic models in AM process simulation) in this highly transient thermal analysis, and; 2) the remarkable computational time savings.

Short term perspectives of this work will cover implementation, validation and performance assessment of a 3D model, development of an *ad-hoc* PGD algorithm to further improve computational gains and implementation in commercial codes to solve complex industrial cases. On a broader long-term scale, future work should address the gigantic multi-physics and the multi-scale nature of the problem which could take great advantage from model order reduction techniques at both local (melt pool level) and global (part level) scales.

6. References

[1] C. Meier, R. W. Penny, Y. Zou, J. S. Gibbs, A. J. Hart, Thermophysical phenomena in metal additive manufacturing by selective laser melting: Fundamentals, modeling, simulation and experimentation, arXiv (2017).

[2] M. Markl, C. Körner, Multiscale modeling of powder bedbased additive manufacturing, Annual Review of Materials Research 46 (2016) 93–123.

[3] T. I. Zohdi, Modeling and simulation of functionalized materials for additive manufacturing and 3d printing: Continuous and discrete media, Springer International Publishing (2018).

[4] K. Z. D. Pal, N. Patil, B. Stucker, An integrated approach to additive manufacturing simulations using physics based, coupled multiscale process modeling, J. Manuf. Sci. Eng. 136 (2014) 061022.1061022.16.

[5] S. S. Al-Bermani, M. L. Blackmore, W. Zhang, I. Todd, The origin of microstructural diversity, texture, and mechanical properties in electron beam melted ti-6al-4v, Metallurgical and Materials Transactions A 41 (2010) 3422–3434.

[6] F. Chinesta, A. Huerta, G. Rozza, K. Willcox, Model reduction methods, Encyclopedia of Computational Mechanics Second Edition (2017) 1–36.

[7] G. Rozza, Reduced-basis methods for elliptic equations in sub-domains with a posteriori error bounds and adaptivity, Applied Numerical Mathematics 55 (2005) 403 – 424.

[8] K. Veroy, A. T. Patera, Certified real-time solution of the parametrized steady incompressible navierstokes equations: rigorous reduced-basis a posteriori error bounds, International Journal for Numerical Methods in Fluids 47 (2005) 773–788.

[9] B. Peherstorfer, K. Willcox, Online adaptive model reduction for nonlinear systems via low-rank updates, SIAM Journal on Scientific Computing 37 (2015) A2123–A2150.

[10] T. Lieu, C. Farhat, M. Lesoinne, Reduced-order fluid/structure modeling of a complete aircraft configuration, Computer Methods in Applied Mechanics and Engineering 195 (2006) 5730 – 5742. John H. Argyris Memorial Issue. Part II.

[11] D. Ryckelynck, F. Vincent, S. Cantournet, Multidimensional a priori hyper-reduction of mechanical models involving internal variables, Computer Methods in Applied Mechanics and Engineering 225-228 (2012) 28 – 43.

[12] P. Ladevèze, A. Nouy, On a multiscale computational strategy with time and space homogenization for structural mechanics, Computer Methods in Applied Mechanics and Engineering 192 (2003) 3061 – 3087.

[13] A. Huerta, E. Nadal, F. Chinesta, Proper generalized decomposition solutions within a domain decomposition strategy, International Journal for Numerical Methods in Engineering 113 (2017) 1972–1994.

[14] P.-B. Rubio, F. Louf, L. Chamoin, Fast model updating coupling bayesian inference and pgd model reduction, Computational Mechanics (2018).

[15] A. Nouy, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1603–1626.

[16] D. Néron, P. Ladevèze, Proper generalized decomposition for multiscale and multiphysics problems, Archives of Computational Methods in Engineering 17 (2010) 351–372.

[17] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. Gonzalez, I. Alfaro, A. Ammar, A. Huerta, Pgd-based computational vademecum for efficient design, optimization and control, Archives of Computational Methods in Engineering 20 (2013) 31–59.

[18] T. DebRoy, H. I. Wei, J. S. Suback, T. Mukherjee, J. W. Elmer, J. O. Milewski, A. M. Beese, A. Wilson-Heid, A. De, W. Zhang, Additive manufacturing of metallic components - process, structure and properties, Progress in Materials Science 92 (2018) 112–224.

[19] J. Goldak, A. Chakravarti, M. Bibby, A new finite element model for welding heat sources, Metallurgical Transactions B 15(2) (1984) 299–305.

[20] K. J. Bathe, Finite Elements Procedures in Engineering Analysis, Prentice-Hall, 1982.

[21] P. Ladevèze, A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations, Computer Methods in Applied Mechanics and Engineering 199 (2010) 1603–1626.

[22] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta, Non incremental strategies based on separated representations: applications in computational rheology, Commun. Math. Sci. 8 (2010) 671–695.

[23] M. Capaldo, P.-A. Guidault, D. Nron, P. Ladevze, The reference point method, a hyperreduction technique: Application to pgd-based nonlinear model reduction, Computer Methods in Applied Mechanics and Engineering 322 (2017) 483 – 514.

[24] F. Chinesta, A. Leygue, M. Beringhier, L. T. Nguyen, J. Grandidier, B. Schrefler, F. Pesavento, Towards a framework for nonlinear thermal models in shell domains, International Journal of Numerical Methods for Heat & Fluid Flow 23 (2013) 55–73.

[25] Z. Fan, F. Liou, Numerical modeling of the additive manufacturing (am) processes of titanium alloy - towards achieving enhanced properties for diversified applications, InTech (2012).

[26] J. Xie, V. Oancea, J. Hurtado, Phase transformations in metals during additive manufacturing processes, NAFEMS World Congress 2017, Stockholm, Sweden (2017).

[27] X. W. Bai, H. O. Zhang, G. L. Wang, Improving prediction accuracy of thermal analysis for weld-based additive manufacturing by calibrating input parameters using ir imaging,, International Journal of Advanced Manufacturing Technology vol. 69 (Nov 2013) 1087–1095.