11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

TOWARDS A PARAMETRISED AND TRANSIENT MODEL OF THE AUTOMATED FIBRE PLACEMENT

Nicolas Bur[1*], Saeid Aghighi[1], Pierre Joyot[1], Francisco Chinesta[2] and Pierre Villon[3]

[1]ESTIA, F-64210 Bidart, France
{n.bur, m.aghighi, p.joyot}@estia.fr

[2]ESI International Chair, GEM CNRS-ECN
1 rue de la Noë BP 92101, 44321 Nantes cedex 3, France
francisco.chinesta@ec-nantes.fr

[3]UTC-Roberval UMR 6253
Centre de recherches de Royallieu, CS 60319, 60203 Compiègne cedex, France
pierre.villon@utc.fr

Key words: Composite materials, Reduced-Order Modelling, Proper Generalized Decomposition, Transient model, Parametrisation.

Abstract. The control of the Laser heat source is critical for the Automated Fibre Placement (AFP) process of composite materials. In order to optimise this process, we propose a new PGD-based solution of the transient heat equation related to a thermal source moving with non uniform velocity. We parametrise the velocity by including its coefficients as model extra-coordinates, allowing to obtain a pertinent abacus with a higher number of parameters.

1 Introduction

Today, the Automated Fibre Placement (AFP) process provides new opportunities in the industry of composite materials. In particular, it makes possible to automatise the creation of small pieces with complex geometry, ideally manufactured by single configured machine. This one operates the composite fibre placement according to the following procedure:

The strip-shaped fibres hike in a sheath up to the working head of the robot that guides them and drapes them over the mould. A laser flux, driven by optical fibre and focused on the suitable point, melts the thermoplastic matrix that coats the fibres.
Simultaneously, the compacting roller, as its name suggests, ensures the cohesion between the substrate (the already draped layers) and the fibre newly melted and laid.

When the robot reaches the edge of the piece, the fibres are cut and the robot repositions itself in order to drape a new tape of material, as illustrated in Fig. 1.

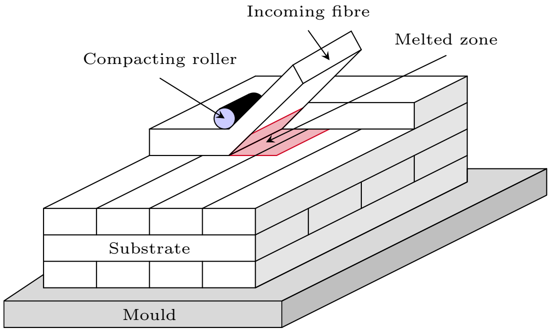


Figure 1: Representation of the stacking

In this process, temperature must be high enough to melt the matrix and ensure a good cohesion between layers, but not too high to avoid burning and degrading the material. That is why it is very crucial to control drastically the laser heat source.

Moreover the AFP process depends on many parameters such as draping velocity, number and orientation of the plies or materials parameters. By the way, using standard Finite Elements Methods (FEM) is not an option since one should have to solve a complex model for a huge set of parameters. That is why we are interested in reduced order model, and more precisely to one of them, the so called Proper Generalised Decomposition (PGD).

Up to now, the AFP process has been studied in the stationary case, in the frame related to the moving laser, with constant velocity, allowing the calculation of the solution at each point of the part [1, 2].

In our study we propose to solve the transient heat equation related to a thermal source moving with non uniform velocity. The velocity is parametrised: we integrate the coefficients involved in it as model extra-coordinates, allowing the calculation of a general parametric solution.

2 Modelling the Automated fibre placement

In this work, the improvement of performances of the AFP process passes through the modelling of the temperature field in the piece under construction. This simulation can involve many phenomena, increasing thereby its complexity. In our study we restrict to a simple problem:
We consider a domain Ω, as shown in Fig. 2, modelling a section of the stacking layers along the thickness. In addition, we assume that the heat source moves on the upper edge of this domain supposed homogeneous.

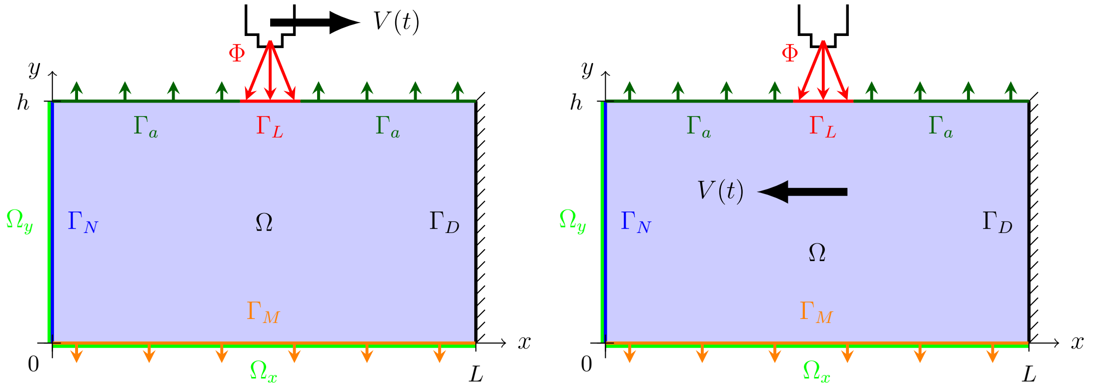
As an initial condition, the material has a constant known temperature. During heating, we also take into account the convection with air as well as the conduction between the piece and the mould, which is kept at constant temperature.

Mathematically, we have to solve the heat equation associated with boundary and initial conditions and presented in Eq.(1).

$$\begin{cases} \rho C_p \partial_t u(x,y,t) - div\left(K \cdot \nabla u(x,y,t)\right) = 0 & \forall (x,y,t) \in \Omega \times \Omega_t; \\ u(x,y,t) = u_0(x,y) & \text{on } \Omega \times \{t_{min}\}\,; \\ u(x,y,t) = \overline{u}(y,t) & \text{on } \Gamma_D \times \Omega_t; \\ K \cdot \partial_n u(x,y,t) = h_m \left(u(x,t) - u_m(x,t)\right) & \text{on } \Gamma_M \times \Omega_t; \\ K \cdot \partial_n u(x,y,t) = h_a \left(u(x,t) - u_a(x,t)\right) & \text{on } \Gamma_a \times \Omega_t; \\ K \cdot \partial_n u(x,y,t) = 0 & \text{on } \Gamma_N \times \Omega_t; \\ K \cdot \partial_n u(x,y,t) = \Phi(x,t) & \text{on } \Gamma_L \times \Omega_t; \end{cases} \tag{1}$$

with

- $K = \begin{bmatrix} k_\parallel & 0 \\ 0 & k_\perp \end{bmatrix}$, tensor of thermal conductivities (W·m^{-1}·K^{-1}),
- ρ density (kg·m^{-3}),
- C_p specific heat (J·kg^{-1}·K^{-1}),
- u_0, $\overline{u}$, u_a, u_m respectively initial, fibres, air and mould temperatures (K),
- h_m (resp. h_a) thermal conductance between composite and mould (resp. air)(W·m^{-2}·K^{-1}),
- Φ source heat flux (W·m^{-2}), depending on the velocity.



(a) Lagrangian description

(b) Eulerian representation

Figure 2: Domain of study

This is for the Lagrangian representation, where the observer is static and sees the source moving (Fig. 2a). For us, due to the method we are going to use for solving the problem, we take advantage of the Eulerian description: the observer is on the source, the domain moves (Fig. 2b).

Applying the change of reference frame leads to the new equation

$$\rho C_p \partial_t u(x,y,t) - div\left(K \cdot \nabla u(x,y,t)\right) + \rho C_p V \cdot \nabla u(x,y,t) = 0, \qquad \forall (x,y,t) \in \Omega \times \Omega_t \quad (2)$$

with the same initial and boundary conditions and the same notations as for the Lagrangian description, but for the heat source term Φ that depends no more on the velocity.

3 Reduced order Model

Among the reduced order models, the PGD method has a particular mention, since it constructs its basis *on the fly* and not *a posteriori* as the POD (Proper Orthogonalised Decomposition [3]) and RB (Reduced Basis [4, 5]).

The PGD is a generalisation of the radial approximation introduced by P. Ladevèze in the 80's [6] which approximation separates space and time.

In the PGD frame, any coordinate can be separated. Thus, physical parameters, initial or boundary conditions can be added as extra-coordinates. For example, if the model involves space, time and k parameters $p^1, \dots, p^k$, the separated form reads

$$U(x,y,t,p^1,\dots,p^k) \approx \sum_{i=1}^{N} \left\{ X_i(x) Y_i(y) T_i(t) \prod_{j=1}^{k} F_i^j(p^j) \right\}. \quad (3)$$

Before express the PDG method in the general case, let us explicit it in a very simple case by resolving the Poisson's equation.

3.1 Poisson's equation

In this section, we are interested in solving the Poisson's equation given by Eq.(4)

$$\begin{cases} -\Delta u(x,y) = Q(x,y) & \forall (x,y) \in \Omega = \Omega_x \times \Omega_y, \\ u(x,y) = 0 & \text{on } \Gamma = \partial\Omega. \end{cases} \quad (4)$$

We assume a separated representation of the temperature field u along the two spaces variables. Therefore we write u as the following sum:

$$u = \sum_{i=1}^{\infty} X^i(x) Y^i(y). \quad (5)$$

Similarly we suppose $Q(x,y) = Q_x(x) \times Q_y(y)$, without loss of generality.

The Dirichlet's condition on boundary Γ is set on the separated form:

$$\forall i \in [1, \infty[\,, \begin{cases} X^i(0) = X^i(1) = 0, \\ Y^i(0) = Y^i(1) = 0. \end{cases} \quad (6)$$

We then look for an approximation u^p of u, defined as follows:

$$u^p = \sum_{i=1}^{p} X^i(x)Y^i(y) = \sum_{i=1}^{p-1} X^i(x)Y^i(y) + R(x)S(y) = u^{p-1} + R(x)S(y). \tag{7}$$

3.1.1 Classical approach

The approached solution u^p given by relation (7) and test function chosen as $u^* = X^*Y + XY^*$ leads to the coupled problem

$$\begin{cases} \begin{cases} \text{find } R \in H_0^1\left(\Omega_x\right) \text{ such that } \forall X^* \in H_0^1\left(\Omega_x\right) \\ \int\limits_{\Omega} -\Delta\left(RS\right) X^*Y dxdy = \int\limits_{\Omega} QX^*Y dxdy + \int\limits_{\Omega} \Delta\left(u^{p-1}\right) X^*Y dxdy; \end{cases} & (8) \\ \begin{cases} \text{find } S \in H_0^1\left(\Omega_y\right) \text{ such that } \forall Y^* \in H_0^1\left(\Omega_y\right) \\ \int\limits_{\Omega} -\Delta\left(RS\right) XY^* dxdy = \int\limits_{\Omega} QXY^* dxdy + \int\limits_{\Omega} \Delta\left(u^{p-1}\right) XY^* dxdy. \end{cases} & (9) \end{cases}$$

where the dependencies on x and y are omitted for the sake of clarity.

To solve this non-linear integral problem (since each unknown function never appears isolated), we use a fixed-point algorithm. Each iteration counts two steps, repeated until reaching convergence.

The first step assumes S known from a previous iteration, and computes an update for R from Eq.(8). From this just updated R, we then update S with Eq.(9). This procedure continues until convergence reached when $\|R_n - R_{n+1}\| \leq \tilde{\varepsilon}$, the indices denoting the fixed loop iteration. The converged functions define the new term in the expansion (7) of $u^p(x, y)$. If necessary, in order to get the wanted precision, another mode is computed.

The convergence criterion over the modes is the next one:
At the end of the fixed-point loop, the product RS is computed. If the norm of this matrix is lower than a user-defined ε, we assume the convergence, for-as-much as the new mode adds only "ε" to the global solution.

Let $\mathbf{N}_X$ and $\mathbf{N}_Y$ be the vectors containing the shape functions associated respectively with each space. We define the mass and stiffness matrices

$$\begin{cases} M_X = \int_{\Omega_X} \mathbf{N}_X \mathbf{N}_X^T dx \quad \text{and} \quad K_X = \int_{\Omega_X} \mathbf{N}_{X,x} \mathbf{N}_{X,x}^T dx; \\ M_Y = \int_{\Omega_Y} \mathbf{N}_Y \mathbf{N}_Y^T dy \quad \text{and} \quad K_Y = \int_{\Omega_Y} \mathbf{N}_{Y,y} \mathbf{N}_{Y,y}^T dy. \end{cases} \tag{10}$$

The continuous expressions (8) and (9) can now be written in the discrete form

$$\begin{aligned} \left(\mathbf{S}^T.M_Y.\mathbf{S}\right) K_X.\mathbf{R} + \left(\mathbf{S}^T.K_Y.\mathbf{S}\right) M_X.\mathbf{R} = \left(\mathbf{S}^T.M_Y.\mathbf{Q}_Y\right) M_X \mathbf{Q}_X - \\ - \sum_{i=1}^{p-1} \left\{ \left(\mathbf{S}^T.M_Y.\mathbf{Y}^i\right) K_X \mathbf{X}^i + \left(\mathbf{S}^T.K_Y.\mathbf{Y}^i\right) M_X.\mathbf{X}^i \right\}; \end{aligned} \tag{11}$$

$$\begin{aligned}\left(\mathbf{R}^T.M_X.\mathbf{R}\right) K_Y.\mathbf{S} + \left(\mathbf{R}^T.K_X.\mathbf{R}\right) M_Y.\mathbf{S} = \left(\mathbf{R}^T.M_X.\mathbf{Q}_X\right) M_Y\mathbf{Q}_Y - \\ - \sum_{i=1}^{p-1}\left\{\left(\mathbf{R}^T.M_X.\mathbf{X}^i\right) K_Y\mathbf{Y}^i + \left(\mathbf{R}^T.K_X.\mathbf{X}^i\right) M_Y.\mathbf{Y}^i\right\},\end{aligned} \tag{12}$$

where the bold font denotes the discrete form of continuous functions.

When it converges, this algorithm is pretty fast. However, in many cases we have to use another approach based on the residual minimisation.

3.1.2 Residual minimisation

When we include convection terms leading to non symmetric operator, the alternating direction strategy that we used in Sect. 3.1.1 can fail: the computed functional couples do not improve noticeably the solution. As Pierre Ladevèze proposed many years ago in the framework of the LATIN method, a more efficient strategy consists of minimizing the residual.

Considering the Poisson's equation (4), the associated discrete residual reads

$$\begin{aligned}\mathcal{R} = \left(K_X.\mathbf{R}\right) \otimes \left(M_Y\mathbf{S}\right) + \left(M_X.\mathbf{R}\right) \otimes \left(K_Y.\mathbf{S}\right) + \left(M_X.\mathbf{Q}_X\right) \otimes \left(M_Y.\mathbf{Q}_Y\right) + \\ + \sum_{i=1}^{p-1}\left(K_X.\mathbf{X}^i\right) \otimes \left(M_Y.\mathbf{Y}^i\right) + \sum_{i=1}^{p-1}\left(M_X.\mathbf{X}^i\right) \otimes \left(K_Y.\mathbf{Y}^i\right).\end{aligned} \tag{13}$$

The stationarity conditions are $\dfrac{\partial \left\|\mathcal{R}\right\|^2}{\partial R} = 0$ et $\dfrac{\partial \left\|\mathcal{R}\right\|^2}{\partial S} = 0$.
Since R and S have symmetric roles, we just write the first stationarity condition:

$$\begin{aligned}&\left(\mathbf{S}^T.M_Y^T.M_Y.\mathbf{S}\right) K_X^T.K_X.\mathbf{R} + \left(\mathbf{S}^T.K_Y^T.M_Y.\mathbf{S}\right) M_X^T.K_X.\mathbf{R} \\ &\quad + \left(\mathbf{S}^T.M_Y^T.K_Y.\mathbf{S}\right) K_X^T.M_X.\mathbf{R} + \left(\mathbf{S}^T.K_Y^T.K_Y.\mathbf{S}\right) M_X^T.M_X.\mathbf{R} = \\ &\quad = \left(\mathbf{S}^T.M_Y^T.M_Y\mathbf{Q}_Y\right) K_X^T.M_X.\mathbf{Q}_X + \left(\mathbf{S}^T.K_Y^T.M_Y\mathbf{Q}_Y\right) M_X^T.M_X.\mathbf{Q}_X + \\ &\quad + \sum_{i=1}^{p-1}\left[\left(\mathbf{S}^T.M_Y^T.M_Y.\mathbf{Y}^i\right) K_X^T.K_X.\mathbf{X}^i\right] + \sum_{i=1}^{p-1}\left[\left(\mathbf{S}^T.K_Y^T.M_Y.\mathbf{Y}^i\right) M_X^T.K_X.\mathbf{X}^i\right] + \\ &\quad + \sum_{i=1}^{p-1}\left[\left(\mathbf{S}^T.M_Y^T.K_Y.\mathbf{Y}^i\right) K_X^T.M_X.\mathbf{X}^i\right] + \sum_{i=1}^{p-1}\left[\left(\mathbf{S}^T.K_Y^T.K_Y.\mathbf{Y}^i\right) M_X^T.M_X.\mathbf{X}^i\right].\end{aligned} \tag{14}$$

3.2 Tensor notation for general framework

Let Ω be a multidimensional domain involving the coordinates (each one not necessarily one dimensional). We consider a weak form of a linear problem given by:

$$a\left(\Psi\left(x_1, x_2, \ldots, x_N\right), \Psi^*\left(x_1, x_2, \ldots, x_N\right)\right) = b\left(\Psi^*\left(x_1, x_2, \ldots, x_N\right)\right), \tag{15}$$

where we are looking to an approximated solution that writes in the continuous form as

$$\Psi\left(x_1, x_2, \ldots, x_N\right) = \sum_{i=1}^{n_F} \alpha^i F_1^i(x_1).F_2^i(x_2).\ \ldots\ .F_N^i(x_N). \tag{16}$$

The separated representation is built-up from a projection-enrichment iteration scheme. For this purpose we need to write the continuous expression in a discrete form using the nodal values of each function. This discrete form is given by

$$\Psi = \sum_{i=1}^{n_F} \alpha^i F_1^i(x_1) \otimes F_2^i(x_2) \otimes\ \ldots\ \otimes F_N^i(x_N) \tag{17}$$

(for the sake of clarity we keep the same notation as for continuous functions) for this purpose the weak form must be transformed into a discrete form written as

$$\Psi^{*T}\mathcal{A}\Psi = \Psi^{*T}\mathcal{B}, \tag{18}$$

with

$$\begin{cases} \mathcal{A} = \sum_{j=1}^{n_A} A_1^j \otimes A_2^j \otimes \ldots \otimes A_N^j, & (19) \\ \mathcal{B} = \sum_{j=1}^{n_B} B_1^j \otimes B_2^j \otimes \ldots \otimes B_N^j. & (20) \end{cases}$$

In term of minimisation, the problem to solve writes

$$\mathcal{A}^T\mathcal{A}\Psi = \mathcal{A}^T\mathcal{B}. \tag{21}$$

The enrichment stage includes new candidates for enriching the reduced separated approximation basis

$$\Psi = \underbrace{\sum_{i=1}^{n_F} \alpha^i F_1^i(x_1) \otimes F_2^i(x_2) \otimes\ \ldots\ \otimes F_N^i(x_N)}_{\Psi_F} + \underbrace{R_1(x_1) \otimes R_2(x_2) \otimes\ \ldots\ \otimes R_N(x_N)}_{\Psi_R} \tag{22}$$

Within a fixed point alternating direction algorithm, we look at each iteration for the computation of a single discrete function R_j assuming all the others known. Thus, when we are looking for R_j the test function writes

$$\Psi^* = R_1 \otimes \ldots \otimes R_{j_1} \otimes R_j^* \otimes R_{j+1} \otimes \ldots \otimes R_N. \tag{23}$$

The system to be solved within a fixed point strategy writes

$$\mathcal{A}^T\mathcal{A}\Psi_R + \mathcal{A}^T\mathcal{A}\Psi_F = \mathcal{A}^T\mathcal{B}, \tag{24}$$

with

$$\mathcal{A}^T\mathcal{A}\Psi_R = \sum_{i=1}^{n_A}\sum_{j=1}^{n_A}\left(A_k^iA_k^jR_k\prod_{\substack{l=1\\l\neq k}}^{N} R_l^T A_l^i A_l^j R_l\right), \tag{25}$$

$$\mathcal{A}^T\mathcal{A}\Psi_F = \sum_{h=1}^{n_F}\sum_{i=1}^{n_A}\sum_{j=1}^{n_A}\left(\alpha^h A_k^iA_k^jF_k^h\prod_{\substack{l=1\\l\neq k}}^{N} R_l^T A_l^i A_l^j F_l^h\right), \tag{26}$$

$$\mathcal{A}^T\mathcal{B} = \sum_{i=1}^{n_A}\sum_{j=1}^{n_A}\left(A_k^iB_k^j\prod_{\substack{l=1\\l\neq k}}^{N} R_l^T A_l^i B_l^j\right). \tag{27}$$

4 Solving the heat equation

In this section, we apply the PGD framework to Eq.(2) we are interested in. Since the robot accelerates and slows down, we consider the velocity presents a ramp as shown in Fig. 3.

t_0, t_1, v_0 and v_1 are parameters given by user: in a classical FEM approach, changing one of them requires a new simulation to get the associated temperature field.

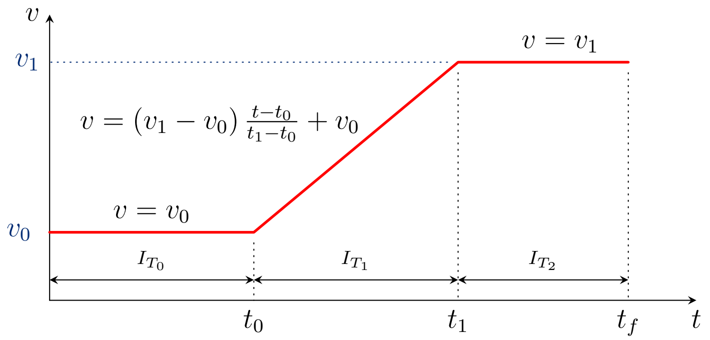


Figure 3: Velocity parametrisation

4.1 Separating $\boldsymbol{X}$, $\boldsymbol{Z}$ and $\boldsymbol{T}$

We seek for an approximation of the temperature field under the form

$$u(x,z,t) \approx \sum_{\alpha=1}^{N} X^\alpha(x)Z^\alpha(z)T^\alpha(t).$$

In order to write the tensor representation of the problem, we need some notations. Let $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{T}$ be the vectors containing the shape functions associated respectively with x, z and t direction space.

On $\boldsymbol{X}$ $\Omega_X = [0,\ L]$

$M_X = \int_{\Omega_X} \mathbf{X}\mathbf{X}^T dx$	$H_X = \int_{\Omega_X} \mathbf{X}_{,x}\mathbf{X}^T dx$	$K_X = \int_{\Omega_X} \mathbf{X}_{,x}\mathbf{X}^T_{,x} dx$
$\Phi_X = \int_{\Omega_X} \Phi\mathbf{X}^T dx$	$U_{m_X} = \int_{\Omega_X} u_m\mathbf{X}^T dx$	$U_{a_X} = \int_{\Omega_X} u_a\mathbf{X}^T dx$

On $\boldsymbol{Z}$ $\Omega_Z = [0,\ h]$

$M_Z = \int_{\Omega_Z} \mathbf{Z}\mathbf{Z}^T dz$	$K_Z = \int_{\Omega_Z} \mathbf{Z}_{,z}\mathbf{Z}^T_{,z} dz$	$H_{m_Z} = \int_{z=0} \mathbf{Z}\mathbf{Z}^T ds$
$H_{a_Z} = \int_{z=h} \mathbf{Z}\mathbf{Z}^T ds$	$B_{m_Z} = \int_{z=0} \mathbf{Z} ds$	$B_{a_Z} = \int_{z=h} \mathbf{Z} ds$

On $\boldsymbol{T}$ $I_T = [0,\ t_f]$

$M_T = \int_{I_T} \mathbf{T}\mathbf{T}^T dt$	$H_T = \int_{I_T} \mathbf{T}_{,t}\mathbf{T}^T dt$
$M_{Tv} = \int_{I_T} v(t)\mathbf{T}\mathbf{T}^T dt$	$B_T = \int_{I_T} \mathbf{T}^T dt$

According to the general framework given at §3.2, Table 1 gives the decompositions of the operators associated to Eq.(2), separating X, Z and T.

Table 1: Tensor decompositions of Eq.(2), separating X, Z and T

$\mathcal{A}_i^j$	$j=1$	2	3	4	5	6	$\mathcal{B}_i^j$	$j=1$	2	3
i=1 (x)	$k_\parallel K_X$	$k_\perp M_X$	$\rho C_p M_X$	$\rho C_p H_X$	$h_m M_X$	$h_a M_X$		Φ_X	$h_m U_{m_X}$	$h_a U_{a_X}$
2 (z)	M_Z	K_Z	M_Z	M_Z	H_{M_Z}	H_{A_Z}		B_{m_Z}	B_{m_Z}	B_{a_Z}
3 (t)	M_T	M_T	M_{Tv}	H_T	M_T	M_T		Bt	Bt	Bt

4.2 Velocity as extra-coordinate

In order to improve and expand the previous modelling, the parameters v_0 and v_1 involved in the expression of velocity are added as extra-coordinates. That way, instead of solving the problem each time we change these values, we will run the computation just once, and get the temperature distribution for any values of v_0 and v_1 (within a given interval). The unknown u is written under the separated approximation

$$u(x,z,t,v_0,v_1) \approx \sum_{\alpha=1}^{N} X^\alpha(x) Z^\alpha(z) T^\alpha(t) V_0^\alpha(v_0) V_1^\alpha(v_1).$$

The previous velocity (Fig. 3) can be written under the separated form

$$v(t,v_0,v_1) = v_0 \times f_{0,t_0} + v_1 \times f_{t_1,t_f} + v_0 \times \left(1 - \frac{t-t_0}{t_1-t_0}\right) \times f_{t_0,t_1} + v_1 \times \frac{t-t_0}{t_1-t_0} \times f_{t_0,t_1}, \quad (28)$$

with $f_{a,b}(t) = \begin{cases} 1 & \text{if } a \leq t \leq b, \\ 0 & \text{else.} \end{cases}$

Let $\mathbf{V_0}$ and $\mathbf{V_1}$ be the vectors containing the shape functions associated respectively with v_0 and v_1 direction space.

We add the following notations

On $\boldsymbol{T}$ $I_T = [0,\ t_f]$

$$M_{T_0} = \int_{I_{T_0}} \mathbf{T}\mathbf{T}^T dt \quad \Bigg\| \quad M_{T_1 v_0} = \int_{I_{T_1}} \left(1 - \frac{t-t_0}{t_1-t_0}\right) \mathbf{T}\mathbf{T}^T dt$$

$$M_{T_1 v_1} = \int_{I_{T_1}} \left(\frac{t-t_0}{t_1-t_0}\right) \mathbf{T}\mathbf{T}^T dt \quad \Bigg\| \quad M_{T_2} = \int_{I_{T_2}} \mathbf{T}\mathbf{T}^T dt$$

On $\boldsymbol{V_0}$ $I_{V_0} = [v_{0_{min}},\ v_{0_{max}}]$

$$M_{V_0} = \int_{I_{V_0}} \mathbf{V_0}\mathbf{V_0}^T dv_0 \quad \Big\| \quad M_{V_0 v} = \int_{I_{V_0}} v_0 \mathbf{V_0}\mathbf{V_0}^T dv_0 \quad \Big\| \quad B_{V_0} = \int_{I_{V_0}} \mathbf{V_0} dv_0$$

On $\boldsymbol{V_1}$ $I_{V_1} = [v_{1_{min}},\ v_{1_{max}}]$

$$M_{V_1} = \int_{I_{V_1}} \mathbf{V_1}\mathbf{V_1}^T dv_1 \quad \Big\| \quad M_{V_1 v} = \int_{I_{V_1}} v_1 \mathbf{V_1}\mathbf{V_1}^T dv_1 \quad \Big\| \quad B_{V_1} = \int_{I_{V_1}} \mathbf{V_1} dv_1$$

According to the general framework given at §3.2, Table 2 gives the decomposition of the operators associated to Eq.(2). As explained in §3.2, one has to keep in mind that the modes previously computed add "some" terms in the right hand side part.

Table 2: Tensor decompositions for LHS and RHS of Eq.(2)

$\mathcal{A}_i^j$	$j=1$	2	3	4	5	6	7	8	9	$\mathcal{B}_i^1$	$\mathcal{B}_i^2$	$\mathcal{B}_i^3$
i=1 (x)	$k_\parallel K_X$	$k_\perp M_X$	$\rho C_p M_X$	$\rho C_p H_X$	$\rho C_p H_X$	$\rho C_p H_X$	$\rho C_p H_X$	$h_m M_X$	$h_a M_X$	Φ_X	$h_m U_{m_X}$	$h_a U_{a_X}$
2 (z)	M_Z	K_Z	M_Z	M_Z	M_Z	M_Z	M_Z	H_{M_Z}	H_{A_Z}	B_{m_Z}	B_{m_Z}	B_{a_Z}
3 (t)	M_T	M_T	H_T	M_{T_0}	$M_{T_1 v_0}$	$M_{T_1 v_1}$	M_{T_2}	M_T	M_T	Bt	Bt	Bt
4 (v_0)	M_{V_0}	M_{V_0}	M_{V_0}	$M_{V_0 v}$	$M_{V_0 v}$	M_{V_0}	M_{V_0}	M_{V_0}	M_{V_0}	B_{V_0}	B_{V_0}	B_{V_0}
5 (v_1)	M_{V_1}	M_{V_1}	M_{V_1}	M_{V_1}	M_{V_1}	$M_{V_1 v}$	$M_{V_1 v}$	M_{V_1}	M_{V_1}	B_{V_1}	B_{V_1}	B_{V_1}

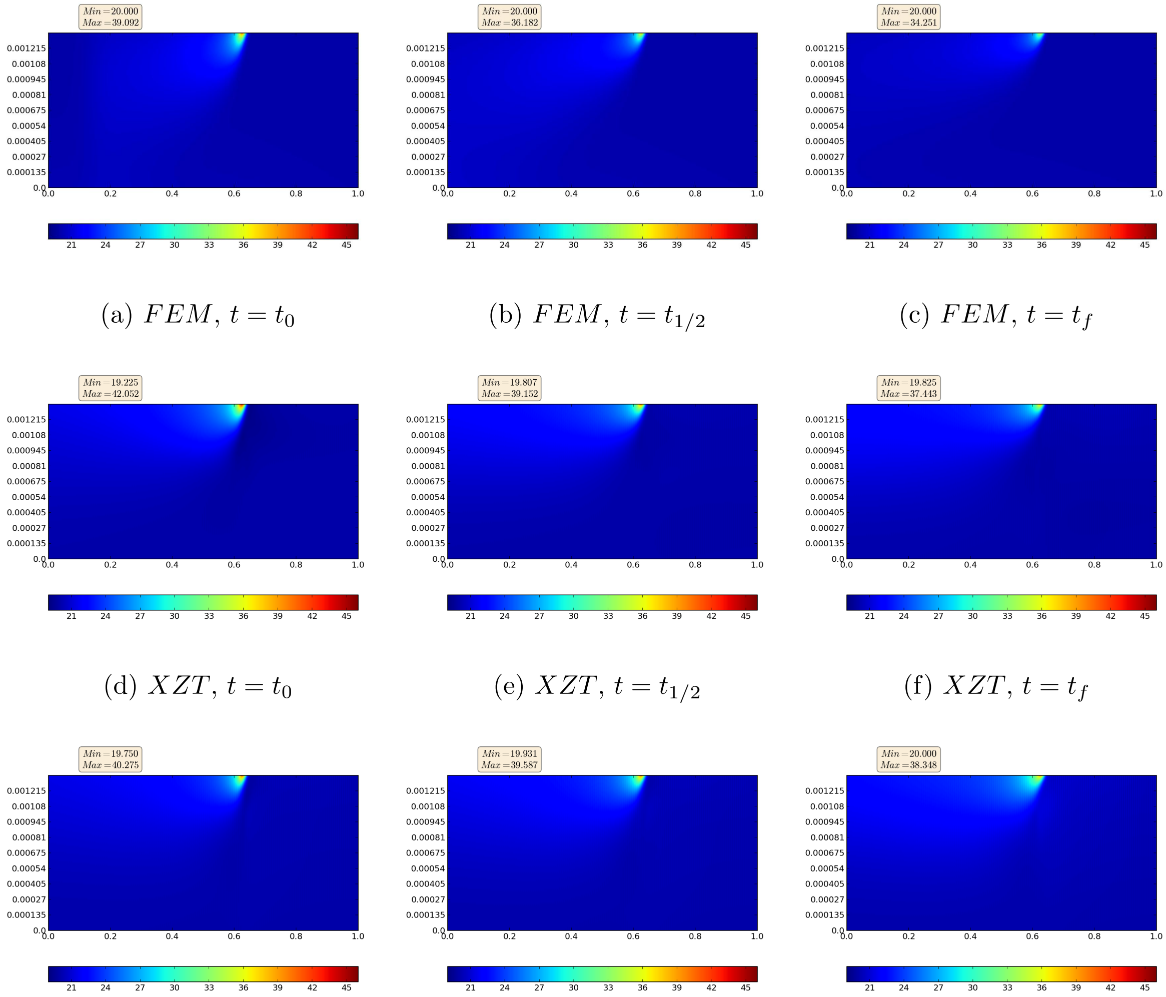
4.3 Results

Here we give some representations of the results obtained with our code, for different values of the parameters. Actually we have an abacus that is an hyper-cube of dimension 5, and we simply pick up the temperature field in it once we choose the parameters. When the simulation is run, browsing the abacus is done in real time.

Fig. 4 shows the results for the two PGD compared with a classical FEM simulation: the upper line gives the temperature fields obtained by FEM, at time step t_0 (the beginning of the ramp), $t_{1/2}$ (half the time interval) and t_f (final instant); the next line shows the results when separating x, y and t, at same time steps. We plot the same for the PGD over x, y, t, v_0 and v_1 on the lower line.

To make the comparison relevant between FEM and the two PGDs, the same discretisations have been used on x, y and t. The approximations count the same number of modes, computed with the same criteria of convergence, for the PGDs.

The slight differences between the two PGD approaches and the FEM computation come from the small number of modes used to construct the approximation of the temperature field. Actually, the more modes are added, the more accurate is the solution. In



(a) FEM, $t = t_0$

(b) FEM, $t = t_{1/2}$

(c) FEM, $t = t_f$

(d) XZT, $t = t_0$

(e) XZT, $t = t_{1/2}$

(f) XZT, $t = t_f$

(g) $XZTV_0V_1$, $t = t_0$

(h) $XZTV_0V_1$, $t = t_{1/2}$

(i) $XZTV_0V_1$, $t = t_f$

Figure 4: Comparison between the two PGD for $v_0 = -0.5$ and $v_1 = -1$

these results, we restrict to 15 modes, what is few enough to catch the physics, particularly the sharp incoming heat source flux.

For this case of study, the FEM runs in approximatively 2h37'. Comparatively, the first PGD separation returns the solution in 6 minutes.
We have to note that these two resolutions give access to the same kind of result: the temperature field in any space and time position, for a given couple $(v_0,\ v_1)$.

The more complex PGD adding the velocity as extra-coordinate allows to obtain a more pertinent abacus after... 6 minutes! We have to keep in mind that this last simulation provides the temperature field not only for any space and time position as the FEM and the first PGD, but also for any values of v_0 and v_1 (within given intervals). And once the computation has been done, the results are given in real time: it is only to supply the values of the parameters.

5 Conclusion

We presented a PGD-based solution in the framework of the heat equation by parametrising a non uniform velocity. We validate our approach on a simple problem with rather good results.

In the future works and in order to bring closer to real world conditions, we intend to consider other phenomena as the thermal conductance between the layers. Indeed, even if the roller compacts the last layer on the substrate, the contact is not perfect due to the roughness of the material, leading to jumps of temperature within the piece. In addition, we are working on the simulation of the AFP process with number and orientation of plies as extra-coordinates.

REFERENCES

[1] Chady Ghnatios. *Simulation avancée des problèmes thermiques rencontrés lors de la mise en forme des composites*. PhD thesis, École Centrale de Nantes, October 2012.

[2] A. Barasinski, A. Leygue, E. Soccard, and A. Poitou. In situ consolidation for thermoplastic tape placement process is not obvious. In *14th International ESAFORM Conference on Material Forming: ESAFORM 2011*, volume 1353, pages 948–953. AIP Publishing, May 2011.

[3] Michel Loeve. Fonctions aléatoires du second ordre. *Comptes Rendus de l'Académie des Sciences*, 1945.

[4] Dennis A. Nagy. Modal representation of geometrically nonlinear behavior by the finite element method. *Computers & Structures*, 10(4):683–688, August 1979.

[5] Ahmed K. Noor and Jeanne M. Peters. Reduced Basis Technique for Nonlinear Analysis of Structures. *AIAA Journal*, 18(4):455–462, April 1980.

[6] Pierre Ladevèze. New algorithms: mechanical framework and development (in french). *Technical report, LMT-Cachan*, 57, 1985.

Aknowledgement

This research is part of the Impala project, which is a FUI 11 project, funded by OSEO, Conseil régional d'Aquitaine and Conseil général des Pyrénées Atlantiques.