Towards a parametrised and transient model of the automated fibre placement

Nicolas Bur, Saeid Aghighi, Pierre Joyot, Francisco Chinesta, Pierre Villon

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–Recherche
Estia, Technopole Izarbel, 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, Non-linearity, Transient model.

Abstract. In our study we propose to solve the transient heat equation related to a thermal source moving with non uniform velocity, using the so called PGD method. The velocity will be parametrized and the coefficients involved in it included as model extra-coordinates, allowing the calculation of a general parametric solution.

1 Introduction

The Automated Fibre Placement (AFP) process seems to be an interesting approach to industrialise the manufacturing of composite materials. Actually this method should allow to create wide pieces of complex geometry without requiring a baking in an expensive autoclave.

For this, a robot as shown on Fig.1 drapes composite fibres in a wide variety of configurations. This is possible not only thanks to the great freedom of movement of the robot, but also due to the small size of the working head that as been designed by Coriolis Composites.

The operating principle is the next one:
The fibres, in the form of strips, hike in a sheath up to the working head of the robot that guides them and drape them over the mould. A Laser flux, driven by optical fibre and focused on the suitable point, melt the thermoplastic matrix that coats the fibres.
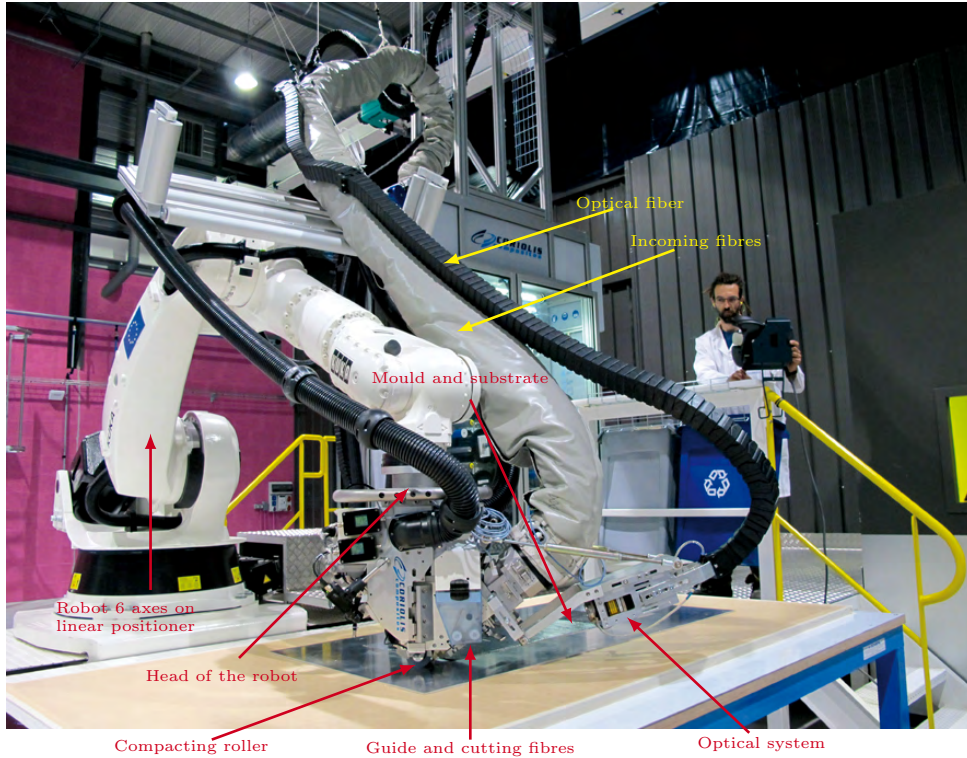
Figure 1: Presentation of the robot

Simultaneously, the compacting roller, as its name suggests, ensure 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.

The master elements indicated are shown on Fig.1. Schematically one can represent the stacking just as on Fig.2.

Figure 2: Representation of the stacking

The aim of modelling such a process is to optimise it. Indeed, to ensure a good cohesion between layers, temperature must be high enough to melt the matrix. However in order not to degrade the material, one must avoid burning it. Therefore the Laser heat source has to be controlled drastically.

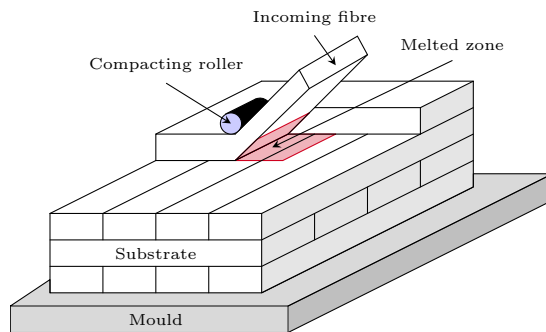Moreover the AFP process depends on many parameters such as draping velocity, number and orientation of the plies or materials parameters. By the way, using standard finite elements methods (FEM) is not an option since one should have to solve a complex model for a huge set of parameters. That is why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

Some work has been carried out, in particular at École Centrale Nantes [1, 2]. In our study, we consider the transient heat equation with non constant velocity. In addition, we will try to add this velocity as extra-coordinate of the problem.

2 Modelling the Automated fibre placement

As presented, we are interesting in modelling the temperature field in the piece under construction.

To simulate the AFP process, many elements have to be taken into account. However, we restrict to quite a simple problem:
In this framework, we study a section Ω of the piece, along the thickness, figuring a stacking of some layers. We consider that the heat source moves on the upper edge of this domain which is assumed to be homogeneous. Before the source heats the material, this one is at a known and constant temperature. We take into account the conduction between the piece and the mould (assumed to be at constant temperature) as well as the convection with air.

Mathematically, we have to solve the heat equation associated with boundary and initial conditions and presented in Eq.; Fig.3 shows the domain of study.

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

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}),

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.

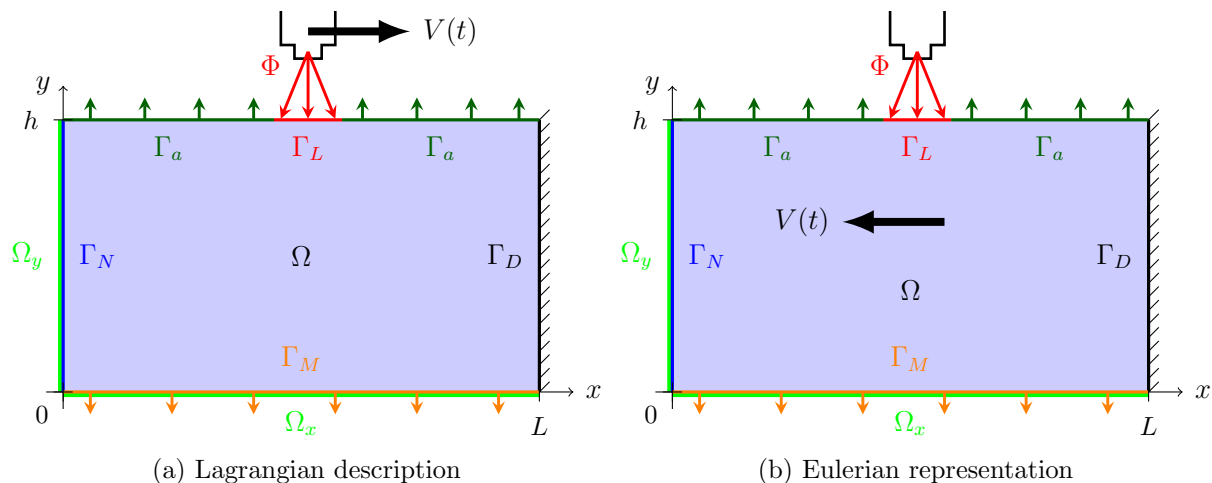(a) Lagrangian description   (b) Eulerian representation

Figure 3: Domain of study

This is for the Lagrangian representation, where the observer is static and sees the source moving. For us, due to the method we are going to use for solving the problem, we have to consider the Eulerian description: the observer is on the source, the domain moves.

Applying the change of reference frame leads to a new equation

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

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 (Proper Generalized Decomposition) method has a particular mention, since this method 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

4

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

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

In order to introduce the Proper Generalised Decomposition (PGD) method and before writing it in the general case, let us consider a very simple example.

3.1 Poisson's equation

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

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

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

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

Similarly we suppose $Q = Q_x \times Q_y$, without loss of generality.
The Dirichlet's condition on Γ is set on the separated form:

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

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

$$\begin{cases} \begin{cases} \text{find } R \in H_0^1(\Omega_x) \text{ such that } \forall X^* \in H_0^1(\Omega_x) \\ \int\limits_{\Omega} -\Delta(RS) X^* Y \, dxdy = \int\limits_{\Omega} Q X^* Y \, dxdy + \int\limits_{\Omega} \Delta(u^{p-1}) X^* Y \, dxdy \end{cases} \\ \begin{cases} \text{find } S \in H_0^1(\Omega_y) \text{ such that } \forall Y^* \in H_0^1(\Omega_y) \\ \int\limits_{\Omega} -\Delta(RS) X Y^* \, dxdy = \int\limits_{\Omega} Q X Y^* \, dxdy + \int\limits_{\Omega} \Delta(u^{p-1}) X Y^* \, dxdy \end{cases} \end{cases} \tag{8} \tag{9}$$

5

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

To get a solution of 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 is known from the 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 an ε given by the user, 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

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

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

$$\begin{cases} \left(\mathbf{S}^T.M_Y.\mathbf{S}\right) K_X.\mathbf{R} + \left(\mathbf{S}^T.K_Y.\mathbf{S}\right) M_X.\mathbf{R} = \left(\mathbf{S}^T.M_Y.\mathbf{Q}_Y\right) M_X \mathbf{Q}_X - \\ \qquad\qquad - \sum_{i=1}^{p-1} \left\{ \left(\mathbf{S}^T.M_Y.\mathbf{Y}^i\right) K_X \mathbf{X}^i + \left(\mathbf{S}^T.K_Y.\mathbf{Y}^i\right) M_X.\mathbf{X}^i \right\} \qquad (11) \\ \left(\mathbf{R}^T.M_X.\mathbf{R}\right) K_Y.\mathbf{S} + \left(\mathbf{R}^T.K_X.\mathbf{R}\right) M_Y.\mathbf{S} = \left(\mathbf{R}^T.M_X.\mathbf{Q}_X\right) M_Y \mathbf{Q}_Y - \\ \qquad\qquad - \sum_{i=1}^{p-1} \left\{ \left(\mathbf{R}^T.M_X.\mathbf{X}^i\right) K_Y \mathbf{Y}^i + \left(\mathbf{R}^T.K_X.\mathbf{X}^i\right) M_Y.\mathbf{Y}^i \right\} \qquad (12) \end{cases}$$

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 strong non-linearities, advection terms, parameters as additional coordinates..., the alternating direction strategy that we used in Sect. 3.1.1 fails: the computed functional couples do not improve noticeably the solution. As Pierre Ladevèze proposed many years ago in the framework of the LATIN method, in that case a more efficient strategy consists of minimizing the residual.

Considering the Poisson's equation (4) and with the same notations as above, the associated discrete residual reads

$$\mathcal{R} = (K_X.\mathbf{R}) \otimes (M_Y \mathbf{S}) + (M_X.\mathbf{R}) \otimes (K_Y.\mathbf{S}) + (M_X.\mathbf{Q}_X) \otimes (M_Y.\mathbf{Q}_Y) +$$

$$+ \sum_{i=1}^{p-1} (K_X.\mathbf{X}^i) \otimes (M_Y.\mathbf{Y}^i) + \sum_{i=1}^{p-1} (M_X.\mathbf{X}^i) \otimes (K_Y.\mathbf{Y}^i) \quad (13)$$

The stationarity conditions are $\dfrac{\partial \|\mathcal{R}\|^2}{\partial R} = 0$ et $\dfrac{\partial \|\mathcal{R}\|^2}{\partial S} = 0$.

Since R and S have symmetric roles, we just write the first stationarity condition.

$$\left(\mathbf{S}^T \cdot M_Y^T \cdot M_Y \cdot \mathbf{S}\right) K_X^T \cdot K_X \cdot \mathbf{R} + \left(\mathbf{S}^T \cdot K_Y^T \cdot M_Y \cdot \mathbf{S}\right) M_X^T \cdot K_X \cdot \mathbf{R}$$
$$+ \left(\mathbf{S}^T \cdot M_Y^T \cdot K_Y \cdot \mathbf{S}\right) K_X^T \cdot M_X \cdot \mathbf{R} + \left(\mathbf{S}^T \cdot K_Y^T \cdot K_Y \cdot \mathbf{S}\right) M_X^T \cdot M_X \cdot \mathbf{R} =$$
$$= \left(\mathbf{S}^T \cdot M_Y^T \cdot M_Y \mathbf{Q}_Y\right) K_X^T \cdot M_X \cdot \mathbf{Q}_X + \left(\mathbf{S}^T \cdot K_Y^T \cdot M_Y \mathbf{Q}_Y\right) M_X^T \cdot M_X \cdot \mathbf{Q}_X +$$
$$+ \sum_{i=1}^{p-1} \left[\left(\mathbf{S}^T \cdot M_Y^T \cdot M_Y \cdot \mathbf{Y}^i\right) K_X^T \cdot K_X \cdot \mathbf{X}^i\right] + \sum_{i=1}^{p-1} \left[\left(\mathbf{S}^T \cdot K_Y^T \cdot M_Y \cdot \mathbf{Y}^i\right) M_X^T \cdot K_X \cdot \mathbf{X}^i\right] +$$
$$+ \sum_{i=1}^{p-1} \left[\left(\mathbf{S}^T \cdot M_Y^T \cdot K_Y \cdot \mathbf{Y}^i\right) K_X^T \cdot M_X \cdot \mathbf{X}^i\right] + \sum_{i=1}^{p-1} \left[\left(\mathbf{S}^T \cdot K_Y^T \cdot K_Y \cdot \mathbf{Y}^i\right) M_X^T \cdot M_X \cdot \mathbf{X}^i\right]$$

$$(14)$$

3.2 Tensor notation for general framework

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

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

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

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

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

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

(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

$$\begin{cases} \mathcal{A} = \sum_{j=1}^{n_A} A_1^j \otimes A_2^j \otimes \ldots \otimes A_N^j & (19) \\[4mm] \mathcal{B} = \sum_{j=1}^{n_B} B_1^j \otimes B_2^j \otimes \ldots \otimes B_N^j & (20) \end{cases}$$

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

$$\begin{cases} \mathcal{A}^T\mathcal{A}\Psi_R = \sum_{i=1}^{n_A}\sum_{j=1}^{n_A} \left(A_k^i A_k^j R_k \prod_{\substack{l=1 \\ l \neq k}}^{N} R_l^T A_l^i A_l^j R_l \right) & (25) \\[6mm] \mathcal{A}^T\mathcal{A}\Psi_F = \sum_{h=1}^{n_F}\sum_{i=1}^{n_A}\sum_{j=1}^{n_A} \left(\alpha^h A_k^i A_k^j F_k^h \prod_{\substack{l=1 \\ l \neq k}}^{N} R_l^T A_l^i A_l^j F_l^h \right) & (26) \\[6mm] \mathcal{A}^T\mathcal{B} = \sum_{i=1}^{n_A}\sum_{j=1}^{n_A} \left(A_k^i B_k^j \prod_{\substack{l=1 \\ l \neq k}}^{N} R_l^T A_l^i B_l^j \right) & (27) \end{cases}$$

8

4 Solving the heat equation

In this section, we are going to apply the PGD framework to Eq.(2) we are interested in. See that the robot accelerates and slows down, we consider the velocity presents a ramp, as shown on Fig.4.
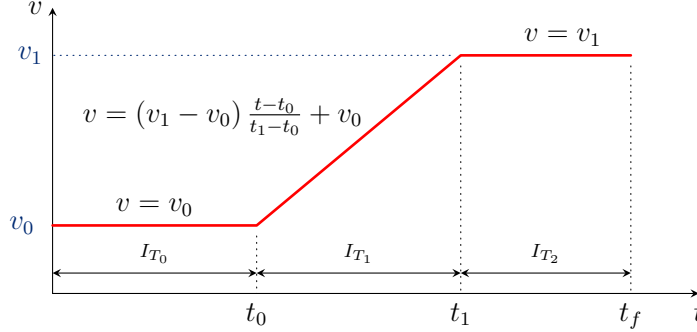

Figure 4: Velocity parametrisation

t_0, t_1, v_0 and v_1 are parameters given by user: changing one of them requires a new simulation.

4.1 Separating X, Z and T

We seek for an approximation of the temperature field by writing it 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 X $\quad \Omega_X = [0, \ L]$
$M_X = \int_{\Omega_X} \mathbf{X}\mathbf{X}^T dx$
$\Phi_X = \int_{\Omega_X} \Phi \mathbf{X}^T dx$
$\qquad \left\| \begin{array}{l} H_X = \int_{\Omega_X} \mathbf{X}_{,x}\mathbf{X}^T dx \\ U_{m_X} = \int_{\Omega_X} u_m \mathbf{X}^T dx \end{array} \right.$
$\qquad \left\| \begin{array}{l} K_X = \int_{\Omega_X} \mathbf{X}_{,x}\mathbf{X}_{,x}^T dx \\ U_{a_X} = \int_{\Omega_X} u_a \mathbf{X}^T dx \end{array} \right.$

On Z $\quad \Omega_Z = [0, \ h]$
$M_Z = \int_{\Omega_Z} \mathbf{Z}\mathbf{Z}^T dz$
$H_{a_Z} = \int_{z=h} \mathbf{Z}\mathbf{Z}^T ds$
$\qquad \left\| \begin{array}{l} K_Z = \int_{\Omega_Z} \mathbf{Z}_{,z}\mathbf{Z}_{,z}^T dz \\ B_{m_Z} = \int_{z=0} \mathbf{Z} ds \end{array} \right.$
$\qquad \left\| \begin{array}{l} H_{m_Z} = \int_{z=0} \mathbf{Z}\mathbf{Z}^T ds \\ B_{a_Z} = \int_{z=h} \mathbf{Z} ds \end{array} \right.$

On T $\quad I_T = [0, \ t_f]$
$M_T = \int_{I_T} \mathbf{T}\mathbf{T}^T dt$
$M_{Tv} = \int_{I_T} v(t)\mathbf{T}\mathbf{T}^T dt$
$\qquad \left\| \begin{array}{l} H_T = \int_{I_T} \mathbf{T}_{,t}\mathbf{T}^T dt \\ B_T = \int_{I_T} \mathbf{T}^T dt \end{array} \right.$

9

According to the general framework given at §3.2, Table 1 gives the decompositions of the operator associated to the left hand side and right hand side of 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 the 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 velocity 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 $f_{a,b} = \begin{cases} 1 & \text{if } a \leq t \leq b, \\ 0 & \text{else.} \end{cases}$

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

We add the following notations

On T $I_T = [0,\ t_f]$

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

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

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

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

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

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

According to the general framework given at §3.2, Table2 gives the decomposition of the operator associated to the left hand side of Eq.(2).

Similarly, Table3 gives the decomposition for the right hand side of 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 equation...

Table 2: Tensor decomposition for LHS of Eq.(2)

$\mathcal{A}_i^j$	$j=1$	2	3	4	5	6	7	8	9
i=1 (x)	$k_\parallel K_X$	$k_\perp M_X$	$\rho C_p M_X$	$\rho C_p H_X$	$\rho C_p H_X$	$\rho C_p H_X$	$\rho C_p H_X$	$h_m M_X$	$h_a M_X$
2 (z)	M_Z	K_Z	M_Z	M_Z	M_Z	M_Z	M_Z	H_{M_Z}	H_{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
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}
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}

Table 3: Tensor decomposition for RHS of Eq.(2)

$\mathcal{B}_i^j$	$j=1$	2	3
i=1 (x)	Φ_X	$h_m U_{m_X}$	$h_a U_{a_X}$
2 (z)	B_{m_Z}	B_{m_Z}	B_{a_Z}
3 (t)	Bt	Bt	Bt
4 (v_0)	B_{V_0}	B_{V_0}	B_{V_0}
5 (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 parameter, what can be done in real time (obviously once the simulation is done!)

Fig.5 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 one 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, 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.

(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) $XZTV_0V_1$, $t = t_f$
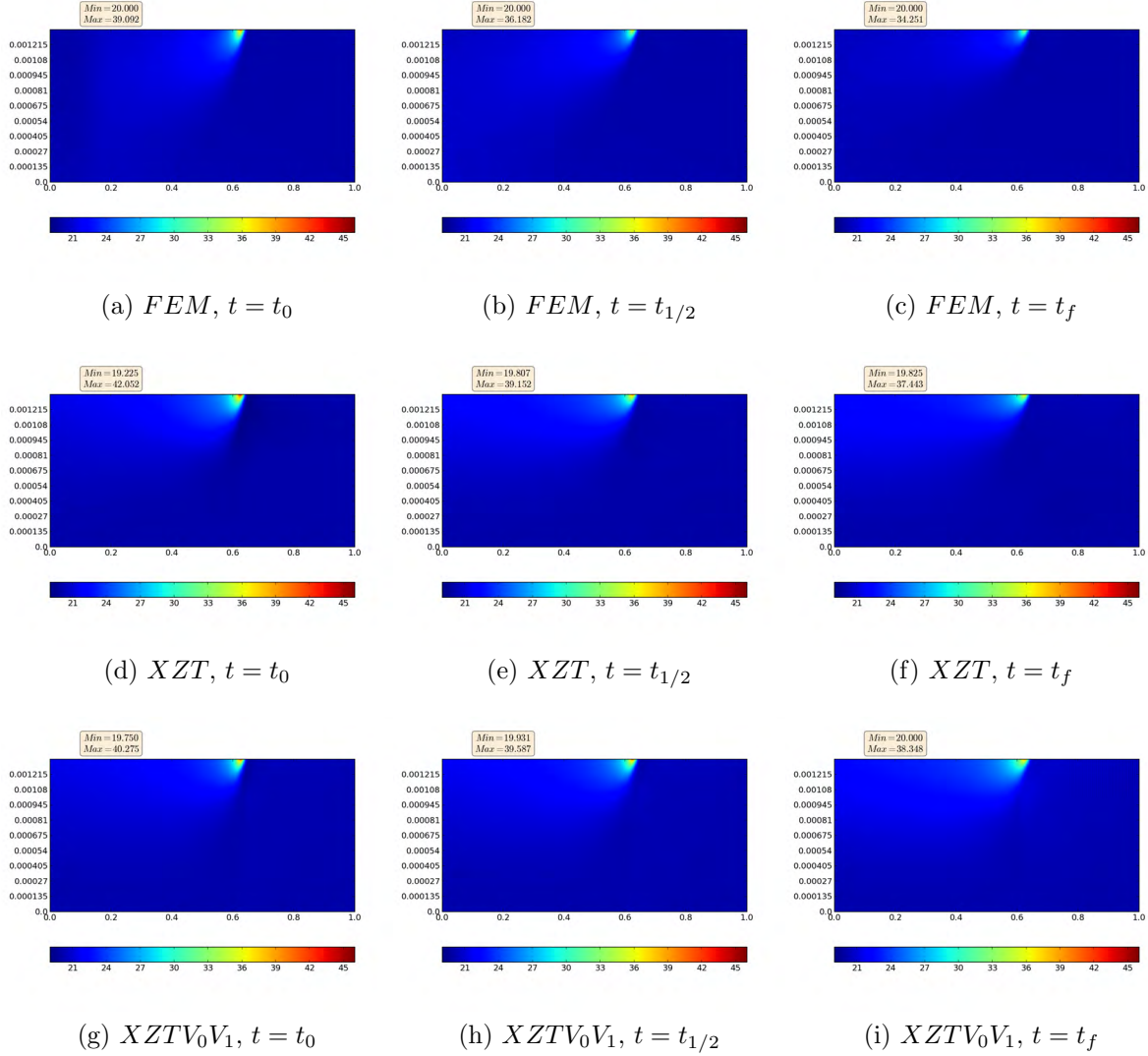
Figure 5: Comparison between the two PGD for $v_0 = -0.5$ and $v_1 = -1$

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

On this problem, it took approximatively XX for the FEM to run.
The first PGD separation returns the solution on YY.
The more complex PGD adding the velocity as extra-coordinate allow to obtain a more pertinent abacus after ZZ.

We have to keep in mind that this last simulation provides the temperature field not only for any space and time position, just as the FEM and the first PGD, but also for any values of v_0 and v_1! And once the computation is done, the results are given in real time: it is only to supply the values of the parameters.

5 Conclusion

This is quite a simple example.
In order to move closer to reality, other phenomena can be considered, 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. This leads to jumps of temperature within the piece.

Currently we work on this, plus simulate the process with the number of plies as extra coordinate.

However, we showed that the parametrisation of the velocity is possible with rather good results.

REFERENCES

[1] Chady Ghnatios. *Simulation avancée des problèmes thermiques rencontrés lors de la mise en forme des composites.* PhD thesis, Ecole 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 *THE 14TH INTERNATIONAL ESAFORM CONFERENCE ON MATERIAL FORMING: ESAFORM 2011*, volume 1353, pages 948–953. AIP Publishing, May 2011.

[3] Michel Loeve. Fonctions aleatoires 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.