



On the solution of the heat equation in very thin tapes

Etienne Pruliere, Francisco Chinesta, Amine Ammar, Adrien Leygue, Arnaud Poitou

► To cite this version:

Etienne Pruliere, Francisco Chinesta, Amine Ammar, Adrien Leygue, Arnaud Poitou. On the solution of the heat equation in very thin tapes. International Journal of Thermal Sciences, 2013, 65, pp.148-157. 10.1016/j.ijthermalsci.2012.10.017 . hal-01062911

HAL Id: hal-01062911

<https://hal.science/hal-01062911>

Submitted on 10 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Science Arts & Métiers (SAM)

is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://sam.ensam.eu>
Handle ID: <http://hdl.handle.net/10985/8490>

To cite this version :

Etienne PRULIERE, Fancisco CHINESTA, Amine AMMAR, Adrien LEYGUE, Arnaud POITOU -
On the solution of the heat equation in very thin tapes - International Journal of Thermal Sciences
- Vol. 65, p.148–157 - 2013

Any correspondence concerning this service should be sent to the repository

Administrator : archiveouverte@ensam.eu

On the solution of the heat equation in very thin tapes

E. Prulière¹, F. Chinesta², A. Ammar³, A. Leygue², A. Poitou²

¹ *Institut de Mécanique et d'Ingénierie de Bordeaux*

16 avenue Pey Berland, 33607 PESSAC Cedex - France

etienne.pruliere@lamef.bordeaux.ensam.fr

² *GEM UMR CNRS - Ecole Centrale de Nantes*

EADS Corporate Foundation International Chair

1 rue de la Noe, BP 92101, F-44321 Nantes cedex 3, France

{Francisco.Chinesta;Adrien.Leygue;Arnaud.Poitou}@ec-nantes.fr

³ *Arts et Métiers ParisTech*

2 Boulevard du Ronceray, BP 93525, F-49035 Angers cedex 01, France

Amine.Ammar@ensam.eu

Abstract

This paper addresses two issues usually encountered when simulating thermal processes in forming processes involving tape-type geometries, as is the case of tape or tow placement, surface treatments, ... The first issue concerns the necessity of solving the transient model a huge number of times because the thermal loads are moving very fast on the surface of the part and the thermal model is usually non-linear. The second issue concerns the degenerate geometry that we consider in which

the thickness is usually much lower than the in-plane characteristic length. The solution of such 3D models involving fine meshes in all the directions becomes rapidly intractable despite the huge recent progresses in computer sciences. In this paper we propose to consider a reduced and fully space-time separated representation of the unknown field. This choice allows circumventing both issues allowing the solution of extremely fine models very fast, sometimes in real time.

Key words: Heat equation; Model reduction; Proper Generalized Decomposition; Composites manufacturing processes

1 Introduction

Industrial processes generally need efficient numerical simulations in order to optimize the process parameters. In the case of composite materials, even if the thermo-mechanical models are nowadays well established, efficient simulations need for further developments.

In this work we are considering some issues, analyzed from a methodological point of view, without considering its industrial counterpart that requires the coupling of different numerical procedures and richer physics.

Thermal models involved in the numerical modeling of composite tape placement processes introduce, despite its geometrical simplicity, a certain number of numerical difficulties related to: (i) the very fine mesh required due to the small domain thickness with respect to the other characteristic dimensions as well as to the presence of the a thermal source moving on the domain surface; and (ii) the long simulation times induced by the small thermal conductivity of polymers and the movement of the heat source;

The solution by using standard discretization techniques can be extremely expensive from the computing time point of view. For example, if one wants to simulate a thermal problem in a ply whose thickness is 1000 times lower than its length (which is a quite common ratio), the use of only 100 nodes in the thickness will lead to use 10^5 nodes in the length to ensure the geometrical quality of the mesh on which standard discretization techniques, like the finite element method, proceed. The total amount of nodes is then 10 millions even when considering a 2D thermal model. In this situation solving a 3D model seems a challenge. Indeed, when the model involves 10^{12} (that implies a reasonable number of nodes, of the order of 10^4 , in each coordinate direction of a 3D model) numerical complexity reaches the current computer capabilities. In addition, in transient non-linear models the problem must be solved at least once at each time step, time step that can be extremely small due to stability constraints

An efficient way to enhance the simulation capabilities is to reduce the size of the approximation basis employed for approximating the unknown field. In the finite elements method, at least one approximation function is associated to each node. Thus, the number of degrees of freedom scales with the number of nodes. Reduced modeling lies in using a reduced number of "appropriate" approximation functions defined in general in the whole domain and able to approximate up to a certain level of accuracy the problem solution at each time. Thus, the numbers of approximation functions (and by the way the number of degrees of freedom) becomes independent of the mesh size. The arising issue is how to calculate these "appropriate" functions defining the reduced approximation basis?

There are several possibilities. A first possibility lies in the use of the Proper

Orthogonal Decomposition - POD - that was employed in a former work [7] for addressing similar issues to the ones concerned by the present work. In what follows we are describing how the POD extracts relevant information for building-up a reduced approximation basis.

1.1 *Extracting relevant information by applying the Proper Orthogonal Decomposition*

We assume that the field of interest $u(\mathbf{x}, t)$ is known at the nodes $\mathbf{x}_i$ of a spatial mesh for discrete times $t_m = m \cdot \Delta t$, with $i \in [1, \dots, M]$ and $m \in [0, \dots, P]$. We use the notation $u(\mathbf{x}_i, t_m) \equiv u^m(\mathbf{x}_i) \equiv u_i^m$ and define $\mathbf{u}^m$ as the vector of nodal values u_i^m at time t_m . The main objective of the POD is to obtain the most typical or characteristic structure $X(\mathbf{x})$ among these $u^m(\mathbf{x})$, $\forall m$. For this purpose, we solve the following eigenvalue problem [23]:

$$\mathbf{C}\mathbf{X} = \alpha\mathbf{X}. \quad (1)$$

Here, the components of vector $\mathbf{X}$ are $X(\mathbf{x}_i)$, and $\mathbf{C}$ is the two-point correlation matrix

$$\mathbf{C}_{ij} = \sum_{m=1}^P u^m(\mathbf{x}_i) \cdot u^m(\mathbf{x}_j), \quad (2)$$

whose matrix form reads:

$$\mathbf{C} = \sum_{m=1}^P \mathbf{u}^m \cdot (\mathbf{u}^m)^T, \quad (3)$$

which is symmetric and positive definite. With the matrix $\mathbf{Q}$ defined as

$$\mathbf{Q} = (\mathbf{u}^1, \dots, \mathbf{u}^P) \quad (4)$$

we have

$$\mathbf{C} = \mathbf{Q} \cdot \mathbf{Q}^T. \quad (5)$$

1.2 Building the POD reduced-order model

In order to obtain a reduced model, we first solve the eigenvalue problem Eq. (1) and select the N eigenvectors $\mathbf{X}_i$, $i = 1, \dots, N$, associated with the N eigenvalues belonging to the interval defined by the highest eigenvalue α_1 and α_1 divided by a large enough number (e.g. 10^8). In practice, N is found to be much lower than M . These N eigenfunctions $\mathbf{X}_i$ are then used to approximate the solution $u^m(\mathbf{x})$, $\forall m$. To this end, let us define the matrix $\mathbf{B} = (\mathbf{X}_1 \cdots \mathbf{X}_N)$.

Now, let us assume for illustrative purposes that an explicit time-stepping scheme is used to compute the discrete solution $\mathbf{u}^{m+1}$ at time t^{m+1} . One must thus solve a linear algebraic system of the form

$$\mathbf{G}^m \mathbf{u}^{m+1} = \mathbf{H}^m. \quad (6)$$

A reduced-order model is then obtained by approximating $\mathbf{u}^{m+1}$ in the subspace defined by the N eigenvectors $\mathbf{X}_i$, i.e.

$$\mathbf{u}^{m+1} \approx \sum_{i=1}^N \mathbf{X}_i \cdot T_i^{m+1} = \mathbf{B} \cdot \mathbf{T}^{m+1}. \quad (7)$$

Equation (6) then reads

$$\mathbf{G}^m \cdot \mathbf{B} \cdot \mathbf{T}^{m+1} = \mathbf{H}^m, \quad (8)$$

or equivalently

$$\mathbf{B}^T \cdot \mathbf{G}^m \cdot \mathbf{B} \cdot \mathbf{T}^{m+1} = \mathbf{B}^T \cdot \mathbf{H}^m. \quad (9)$$

The coefficients $\mathbf{T}^{m+1}$ defining the solution of the reduced-order model at the time step $m + 1$ are thus obtained by solving an algebraic system of size N instead of M . When $N \ll M$, as is the case in numerous applications, the solution of Eq. (9) is thus preferred because of its much reduced size.

Remark 1 *The reduced-order model Eq. (9) is built a posteriori by means of the already-computed discrete field evolution. Thus, one could wonder about the interest of the whole exercise. In fact, two beneficial approaches are widely considered (see e.g. [5],[6],[12],[15],[19],[22],[23] [17] [18]). The first approach consists in solving the large original model over a short time interval, thus allowing for the extraction of the characteristic structure that defines the reduced model. The latter is then solved over larger time intervals, with the associated computing time savings. The other approach consists in solving the original model over the entire time interval, and then using the corresponding reduced model to solve very efficiently similar problems with, for example, slight variations in material parameters or boundary conditions. We considered some years ago an adaptive technique for constructing the reduced basis without an "a priori" knowledge [22] [23] [2], following the original proposal in [21].*

Remark 2 *The application of the POD allows to express the unknown function $u(\mathbf{x}, t)$ in the reduced space-time separated form*

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{i=N} T_i(t) \cdot X_i(\mathbf{x}) \quad (10)$$

where $X_i(\mathbf{x})$ are space dependent function (the eigenfunctions resulting from the application of the POD) and $T_i(t)$ are its coefficients that only depend on time.

Remark 3 *Despite the fact of having proposed techniques able to define the*

reduced basis without an "a priori" knowledge, the robustness of such strategies is not ensured and in some cases these strategies do not converge. In that case one could consider as starting point a separated representation of the problem solution $u(\mathbf{x}, t)$

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{i=N} T_i(t) \cdot X_i(\mathbf{x}) \quad (11)$$

and then inject it in the weak form of the problem. This procedure allows computing the functions involved in the separated approximation without any "a priori" knowledge. This strategy was proposed by Pierre Ladeveze in the 80's, and he called it radial approximation [13] [14] [16].

Inspired by this procedure one could try to generalize this representation to the multidimensional fields as was proposed in [1] [3]. This generalized formulation was called Proper Generalized Decomposition –PGD–. See [8] for a recent review. In the PGD framework, when the domain is hexahedral an appealing separated representation of $u(\mathbf{x}, t)$ consists of a full separation, i.e.

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{i=N} T_i(t) \cdot X_i(x) \cdot Y_i(y) \cdot Z_i(z) \quad (12)$$

In the case of non hexahedral domains, a fully separated representation is always possible as proved in [11] but it involves some technical points.

In the present paper we are applying a fully separated representation of the temperature field defined in an hexahedral space-time domain on which a thermal source is moving.

2 Proper Generalized Decomposition of a thermal model defined in rectangular domain

For the sake of simplicity in the description of the technique we consider the application of the PGD for solving the transient heat equation in a 2D rectangular spacial domain (3D results will be presented later) because its generalization for addressing multidimensional problems is straightforward.

The transient thermal model is defined in $\Omega \times \mathcal{I}$, $\Omega = \Omega_x \times \Omega_y$ ($\Omega_x = (0, L)$ and $\Omega_y = (0, H)$) and $\mathcal{I} = (0, t_{max}]$. The evolution of the temperature field $u(x, y, t)$ is governed by the heat equation

$$\frac{\partial u}{\partial t} - \nabla \cdot (\mathbf{K} \cdot \nabla u) = 0 \quad (13)$$

where $\mathbf{K}$ represents the diffusivity tensor, assumed, without loss of generality, constant. If we proceed in the coordinate system associated with the principal directions of $\mathbf{K}$, the diffusivity tensor becomes diagonal, being its components k_x and k_y . In that system of coordinates the previous equation reduced to:

$$\frac{\partial u}{\partial t} - k_x \frac{\partial^2 u}{\partial x^2} - k_y \frac{\partial^2 u}{\partial y^2} = 0 \quad (14)$$

We assume, without loss of generality, a constant initial temperature

$$u(x, y, t = 0) = u^0 \quad (15)$$

and we prescribe the heat flux on the whole boundary $\Gamma \equiv \partial\Omega$, $\Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \cup \Gamma_4$, $\Gamma_1 = (x = 0, y \in \Omega_y)$, $\Gamma_2 = (x \in \Omega_x, y = 0)$, $\Gamma_3 = (x = L, y \in \Omega_y)$

and $\Gamma_4 = (x \in \Omega_x, y = H)$:

$$\left\{ \begin{array}{l} \frac{du}{dx}|_{\mathbf{x} \in \Gamma_1} = 0 \\ \frac{du}{dy}|_{\mathbf{x} \in \Gamma_2} = 0 \\ \frac{du}{dx}|_{\mathbf{x} \in \Gamma_3} = 0 \\ \frac{du}{dy}|_{\mathbf{x} \in \Gamma_4} = q(x, t) \end{array} \right. \quad (16)$$

where $q(x, t)$ represents the heating source that moves on the surface Γ_4 .

The weak form related to Eq. (14) reads:

$$\int_{\Omega \times \mathcal{I}} u^* \cdot \left(\frac{\partial u}{\partial t} - k_x \frac{\partial^2 u}{\partial x^2} - k_y \frac{\partial^2 u}{\partial y^2} \right) d\Omega \cdot dt = 0 \quad (17)$$

$\forall u^*$ in an appropriate functional space.

In order to transfer the boundary condition into the integral formulation (17)

we perform an spatial integration by parts, which results in

$$\begin{aligned} & \int_{\Omega \times \mathcal{I}} u^* \cdot \frac{\partial u}{\partial t} d\Omega \cdot dt + \int_{\Omega \times \mathcal{I}} k_x \frac{\partial u^*}{\partial x} \cdot \frac{\partial u}{\partial x} d\Omega \cdot dt + \int_{\Omega \times \mathcal{I}} k_y \frac{\partial u^*}{\partial y} \cdot \frac{\partial u}{\partial y} d\Omega \cdot dt - \\ & - \int_{\Gamma_4} u^* \cdot q(x, t) dx \cdot dt = 0 \end{aligned} \quad (18)$$

Now, we assume a separated representation of the temperature field

$$u(x, y, t) \approx \sum_{i=1}^{i=N} T_i(t) \cdot X_i(x) \cdot Y_i(y) \quad (19)$$

In order to construct such representation we proceed iteratively, by computing a term of the finite sum at each iteration. If we assume that at iteration n , functions $X_i(x)$, $Y_i(y)$ and $T_i(t)$, $i = 1, \dots, n$, were already computed, the

solution at iteration n , $u^n(x, y, t)$ writes:

$$u^n(x, y, t) = \sum_{i=1}^{i=n} T_i(t) \cdot X_i(x) \cdot Y_i(y) \quad (20)$$

Remark 4 *In order to ensure the verification of the initial condition (the Neunman's boundary ones are implicit in the weak formulation) we could consider that the first term of the finite sum decomposition is given by $T_1(t) = u_0$ and $X_1(x) = Y_1(y) = 1$. In more complex situations the interested reader can refer to [11].*

At iteration $n+1$ we look for the new functions $X_{n+1}(x)$, $Y_{n+1}(y)$ and $T_{n+1}(t)$ that for the sake of clarity will be denoted by $R(x)$, $S(y)$ and $W(t)$. Thus, we can write:

$$u^{n+1}(x, y, t) = u^n(x, y, t) + R(x) \cdot S(y) \cdot W(t) \quad (21)$$

The associated weighting function u^* reads:

$$u^*(x, y, t) = R^*(x) \cdot S(y) \cdot W(t) + R(x) \cdot S^*(y) \cdot W(t) + R(x) \cdot S(y) \cdot W^*(t) \quad (22)$$

Introducing Eqs. (21) and (22) into Eq. (18) yields a non-linear integral problem because each unknown function ($R(x)$, $S(y)$ and $W(t)$) never appear isolated but is always multiplying several unknown functions.

A linearization strategy is compulsory. In our earlier papers [1] and [3], we used Newton's method. Simpler linearization strategies can also be applied. The simplest one is an alternating direction, fixed-point algorithm, which was found remarkably robust in the present context. Each iteration consists of three steps that are repeated until reaching convergence, that is, until reaching the fixed point. The first step assumes $S(y)$ and $W(t)$ known from the previous iteration and compute an update for $R(x)$ (in this case the test function reduces to

$R^*(x) \cdot S(y) \cdot W(t)$). From the just-updated $R(x)$ and the previously-used $W(t)$, we can update $S(y)$ (with $u^* = R(x) \cdot S^*(y) \cdot W(t)$). Finally, from the just-computed $R(x)$ and $S(y)$, we update $W(t)$ (with $u^* = R(x) \cdot S(y) \cdot W^*(t)$). This iterative procedure continues until convergence. The converged functions define the new term in the expansion 19 of $u(x, y, t)$: $X_{n+1}(x) = R(x)$, $Y_{n+1}(y) = S(y)$ and $T_{n+1}(t) = W(t)$.

In what follows we detail the problems to be solved at each one of these three steps.

(1) Computing $R(x)$ being $S(y)$ and $W(t)$ given.

In the present case the test function reads:

$$u^* = R^*(x) \cdot S(y) \cdot W(t) \quad (23)$$

that introduced into the integral form (18) results in:

$$\begin{aligned} & \int_{\Omega_x \times \Omega_y \times \mathcal{I}} R^* \cdot S \cdot W \cdot R \cdot S \cdot \frac{dW}{dt} dx \cdot dy \cdot dt + \\ & + \int_{\Omega_x \times \Omega_y \times \mathcal{I}} k_x \cdot \frac{dR^*}{dx} \cdot S \cdot W \cdot \frac{dR}{dx} \cdot S \cdot W dx \cdot dy \cdot dt + \\ & + \int_{\Omega_x \times \Omega_y \times \mathcal{I}} k_y \cdot R^* \cdot \frac{dS}{dy} \cdot W \cdot R \cdot \frac{dS}{dy} \cdot W dx \cdot dy \cdot dt - \\ & - \int_{\Omega_x \times \mathcal{I}} R^* \cdot S(y = H) \cdot W \cdot q(x, t) dx \cdot dt = \\ & - \int_{\Omega_x \times \Omega_y \times \mathcal{I}} R^* \cdot S \cdot W \cdot \sum_{i=1}^{i=n} \left(X_i \cdot S_i \cdot \frac{dT_i}{dt} \right) dx \cdot dy \cdot dt - \\ & - \int_{\Omega_x \times \Omega_y \times \mathcal{I}} k_x \cdot \frac{dR^*}{dx} \cdot S \cdot W \cdot \sum_{i=1}^{i=n} \left(\frac{dX_i}{dx} \cdot Y_i \cdot T_i \right) dx \cdot dy \cdot dt - \\ & - \int_{\Omega_x \times \Omega_y \times \mathcal{I}} k_y \cdot R^* \cdot \frac{dS}{dy} \cdot W \cdot \sum_{i=1}^{i=n} \left(X_i \cdot \frac{dY_i}{dy} \cdot T_i \right) dx \cdot dy \cdot dt \quad (24) \end{aligned}$$

where the dependences of R , S and W on their respective coordinates were omitted for the sake of clarity.

As all the functions involving the y and t coordinates are known, we can integrate Eq. (24) in $\Omega_y \times \mathcal{I}$ leading to:

$$\int_{\Omega_x} \left(R^* \cdot \alpha^x \cdot R + \frac{dR^*}{dx} \cdot \beta^x \cdot \frac{dR}{dx} \right) dx =$$

$$= \int_{\Omega_x} \left(R^* \cdot \gamma^x(x) + \frac{dR^*}{dx} \cdot \delta^x(x) \right) dx \quad (25)$$

where α^x and β^x are two constants and $\gamma^x(x)$ and $\delta^x(x)$ are functions of x . Eq. (25) can be solved by using any standard technique, as for example a 1D finite element discretization.

Remark 5 *Efficient implementation requires a separated representation of the thermal source*

$$q(x, t) \approx \sum_{i=1}^{i=Q} F_i(x) \cdot G_i(t) \quad (26)$$

decomposition that can be performed by using the SVD (singular value decomposition).

- (2) Computing $S(y)$ being $R(x)$ and $W(t)$ given.

In this case and proceeding in a similar way that previously but integrating in $\Omega_x \times \mathcal{I}$ it results in

$$\begin{aligned} & \int_{\Omega_y} \left(S^* \cdot \alpha^y \cdot S + \frac{dS^*}{dy} \cdot \beta^y \cdot \frac{dS}{dy} \right) dy = \\ & = S^*(y = H) \cdot \gamma^y + \int_{\Omega_y} \frac{dS^*}{dy} \cdot \delta^y(y) dy \end{aligned} \quad (27)$$

In the preset case the integral on $\Gamma_4 \times \mathcal{I}$ results in a constant value γ^y .

- (3) Computing $W(t)$ being $R(x)$ and $S(y)$ given.

Now, the weak form is integrated in $\Omega_x \times \Omega_y$ to derive the equation given $W(t)$. In the present case it is easy to verify that the resulting equation reads:

$$\begin{aligned} & \int_{\mathcal{I}} W^* \cdot \left(\alpha^t \cdot \frac{dW}{dt} + \beta^t \cdot W \right) dt = \\ & = \int_{\mathcal{I}} W^* \cdot \delta^t(t) dt \end{aligned} \quad (28)$$

One could solve this weak form by using a stabilized discretization technique (e.g. discontinuous Galerkin) or coming back to its strong form

$$\alpha^t \cdot \frac{dW}{dt} + \beta^t \cdot W = \delta^t(t) \quad (29)$$

that can be solved by using any standard finite difference discretization (e.g. backward Euler, among many others).

We have seen that at each enrichment step the construction of the new functional product in Eq. (19) requires iterations. If m_i denotes the number of iterations needed at enrichment step i for computing $X_i(x)$, $Y_i(y)$ and $T_i(t)$, the total number of iterations involved in the construction of the PGD approximation is $m = \sum_{i=1}^{i=N} m_i$. In the above example, the entire procedure thus involves the solution of $2 \cdot m$ ($3 \cdot m$ in 3D thermal problems) one-dimensional boundary values problems for the functions $X_i(x)$ and $Y_i(y)$ and m one-dimensional initial values problems for the functions $T_i(t)$. In general, m_i rarely exceeds ten. The number N of functional products needed to approximate the solution with enough accuracy depends on the solution regularity. All numerical experiments carried to date reveal that N ranges between a few tens and one hundred. Thus, we can conclude that the complexity of the PGD procedure to compute the approximation (19) is of some tens of 1D problems. In a classical approach, one must solve a 2D problem at each time step. In usual applications, this often implies the computation of several millions of 2D solutions. Clearly, the CPU time savings by applying the PGD can be of several orders of magnitude.

Remark 6 *The just proposed strategy also applies for solving non-linear models. In that case many standard linearization strategies can be considered. Thus, one could expect that when looking for the solution at iteration $n + 1$, u^{n+1} , all the non-linear terms could be considered at the previous iteration, by using u^n for evaluating all the non-linear contributions. This technique runs, as well as many other variants [4]. A non-conventional and specially appealing technique for addressing complex non linearities lies in the use of the LATIN method [13] [14] [16].*

Remark 7 *Because when using the PGD method the computational complex-*

ity scales linearly with the model dimensionality instead of the exponential growing characteristic of mesh based discretization techniques, one could introduce new extra-coordinates in the model, other than the usual space and time, without a significant impact on the CPU time. Thus, thermal parameters, initial and/or boundary conditions, geometrical parameters, ... can be considered as extra-coordinates. Then, by solving once the multidimensional resulting model, we have access to the space-time evolution for each value of the parameters that were introduced as extra-coordinates. The interested reader can refer to [9] [20] [10] and the references therein.

Remark 8 *Because we have decoupled in the solution algorithm the space and time problems, the meshes used for solving each one of the problems becomes uncorrelated. Thus, there is not stability constraints on the time step. Moreover, we could consider extremely small time steps without affecting the computation cost significantly, because that choice only affects the solution accuracy of the one-dimensional initial value problem serving to the calculation of functions $T_i(t)$.*

Remark 9 *Because the just argued decoupling, the problems that must be solved within the PGD framework at each step (the ones concerning the calculation of $R(x)$, $S(y)$ and $W(t)$) can be solved, if desired, by using different discretization methods for each one of them.*

Remark 10 *When the diffusivity becomes too small, the non-symmetry of the time differential operator requires a variant of the algorithm described above. In that case we should proceed to the residual minimization [8].*

3 Numerical results

Prior to perform some numerical test in 2D and 3D, we are focusing in the thermal source that will be considered and the issues related to its space-time separated representation.

3.1 Thermal source

We consider a thermal source moving along the surface $y = H$ with a velocity v . Because in many industrial applications such thermal source consists of a laser beam, we assume that the thermal flux on the upper surface is modeled from a gaussian distribution whose characteristic length will be denoted by l . In the 3D solutions addressed later, we will assume without loss of generality that this distribution is uniform in the z -direction. Thus, the thermal flux reads:

$$q(x, t) = A \cdot \frac{1}{l\sqrt{2\pi}} \cdot \exp\left(-\frac{(x - vt)^2}{2l^2}\right) \quad (30)$$

where A represents the thermal flux intensity.

In order to perform a separated representation description of $q(x, t)$ we compute the matrix $\mathbf{q}$ with components $\mathbf{q}_{j,r} = q(\tilde{x}_j, \tilde{t}_j)$, where $(\tilde{x}_j, \tilde{t}_j)$ are related to a coarse mesh consisting of $\tilde{M}$ nodes on the upper boundary $y = H$ and $\tilde{P}$ time steps.

As soon as matrix $\mathbf{q}$ is defined, we can apply a singular value decomposition – SVD – that allows to define its separated form representation on the coarse

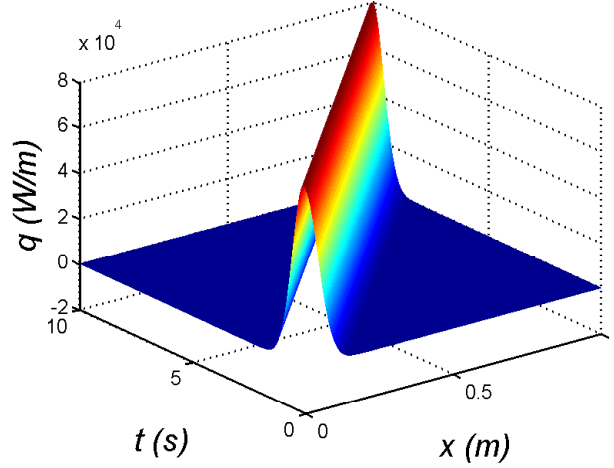


Fig. 1. Reconstructed heat flux consisting of a separated representation involving 15 terms.

mesh

$$q(\tilde{x}, \tilde{t}) \approx \sum_{i=1}^{i=Q} \tilde{F}_i(\tilde{x}) \cdot \tilde{G}_i(\tilde{t}) \quad (31)$$

By performing a projection of the functions involved in that representation on the fine calculation mesh, we obtain finally

$$q(x, t) \approx \sum_{i=1}^{i=Q} F_i(x) \cdot G_i(t) \quad (32)$$

When applying this procedure on the thermal flux (30) for $v = 0.1$, $A = 1$ and $l = 0.05$ (all the unit in the metric systems) the separated representation consisting of the $Q = 15$ most significant functions $F_i(x)$ and $G_i(t)$ exhibits an approximation error of 0.03% when comparing the reconstructed solution (32) depicted in Fig. 1 with its exact expression (30). For the application of the SVD, a coarse mesh consisting of $\tilde{M} = 100$ nodes in the x -direction and $\tilde{P} = 100$ in the time axis was considered (even if it can be applied efficiently on the finer mesh). The functions that resulted from the SVD application where projected on the fine calculation mesh consisting of $M = 1000$ and $P = 1000$.

When considering sharper thermal sources the number of required terms for approximating it up to a certain accuracy increases in a significant manner. For example when considering a moving step, all the modes are relevant and no reduction can be made by applying the SVD. In those cases one could proceed without performing a space-time separated representation of the thermal source. If we observe the terms affected by such choice in the procedure described in the previous section we notice that in its first step (the one related to the calculation of function $R(x)$ (Eq. (24)) the boundary integral writes:

$$\int_{\Omega_x \times \mathcal{I}} R^* \cdot S(y = H) \cdot W \cdot q(x, t) \, dx \cdot dt \quad (33)$$

that could be integrated numerically in the time interval $\mathcal{I}$.

In the second step, the one leading to the calculation of $S(y)$ it results:

$$\int_{\Omega_x \times \mathcal{I}} R \cdot S^*(y = H) \cdot W \cdot q(x, t) \, dx \cdot dt \quad (34)$$

to be integrated in $\Omega_x \times \mathcal{I}$, and finally in the third step, the ones leading to the calculation of $W(t)$ it results

$$\int_{\Omega_x \times \mathcal{I}} R \cdot S(y = H) \cdot W^* \cdot q(x, t) \, dx \cdot dt \quad (35)$$

that must be integrated in Ω_x .

When the thermal sources are localized in space, $q(x, t)$ vanishes in the most part of the domain $\Omega_x \times \mathcal{I}$ and in that case previous integrals can be performed without major difficulties in a reasonable time. On the other hand it can be noticed that when the thermal source can be separated, integrations can be carried out very fast because multidimensional integrals can be computed from the product of one-dimensional integrals.

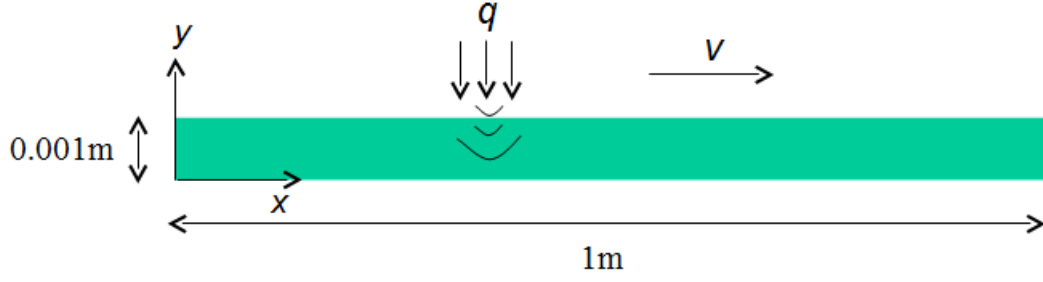


Fig. 2. Sketch of the geometry and process conditions.

3.2 2D numerical test

In this section we are considering the geometry and the process conditions sketched in Fig. 2. The calculation mesh consists of 1000 nodes in the length, 100 nodes in the thickness and 1000 time steps. We consider the previous gaussian flux with again $A = 1$, $v = 0.1$ and $l = 0.05$. The material thermal properties are $\rho = 1000 \text{ kg} \cdot \text{m}^{-3}$, $C_p = 1000 \text{ J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$, $\lambda_x = 5 \text{ W} \cdot \text{K}^{-1} \cdot \text{m}^{-1}$ and $\lambda_y = 0.5 \text{ W} \cdot \text{K}^{-1} \cdot \text{m}^{-1}$, which allows computing the diffusivity values: $k_x = \frac{\lambda_x}{\rho \cdot C_p}$ and $k_y = \frac{\lambda_y}{\rho \cdot C_p}$.

Figure 3 depicts the most relevant functions involved in the separated representation of the temperature field $X_i(x)$, $Y_i(y)$ and $T_i(t)$, $i = 1, \dots, 4$. The reconstructed solution obtained from these functions is depicted in Fig. 4 at different times that correspond to different positions of the thermal source moving on the surface $y = H = 0.001$.

3.3 3D numerical test

The procedure detailed above can be easily extended to 3D geometries. For that purpose it suffices to consider the space-time separated representation of

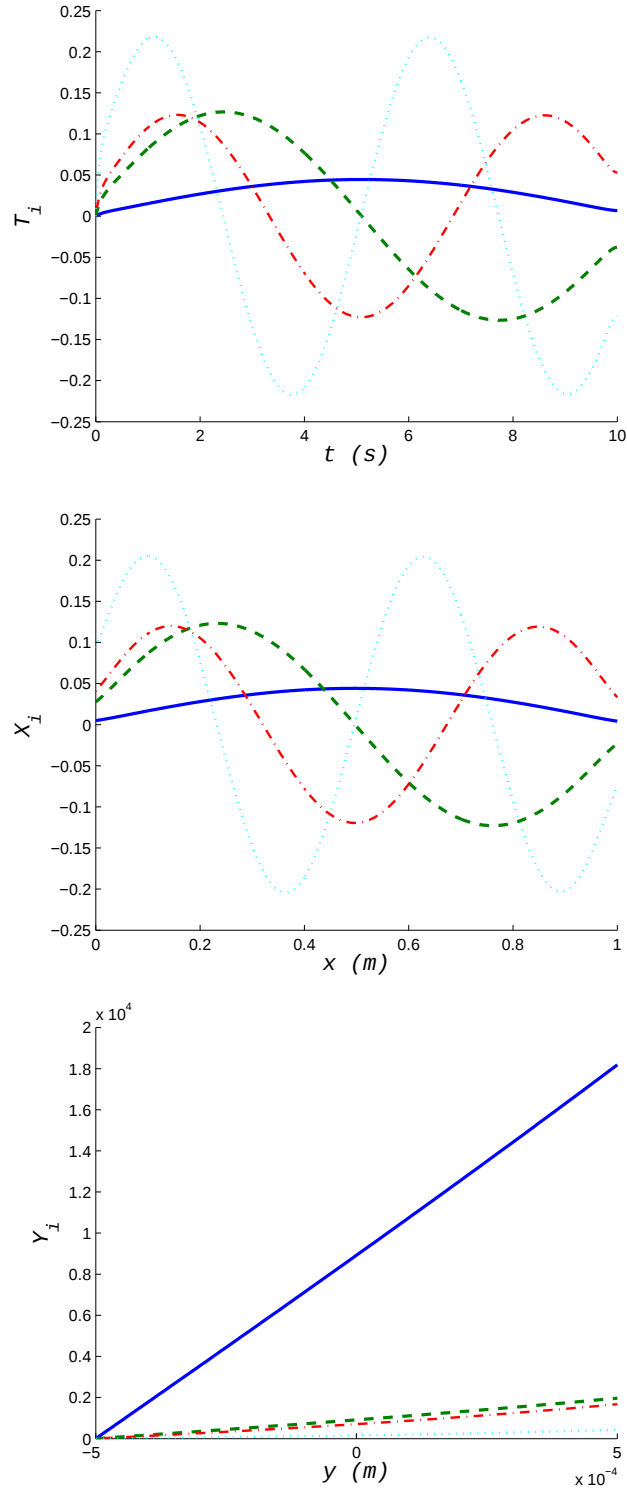


Fig. 3. Most significant functions involved in the separated representation of the temperature field.

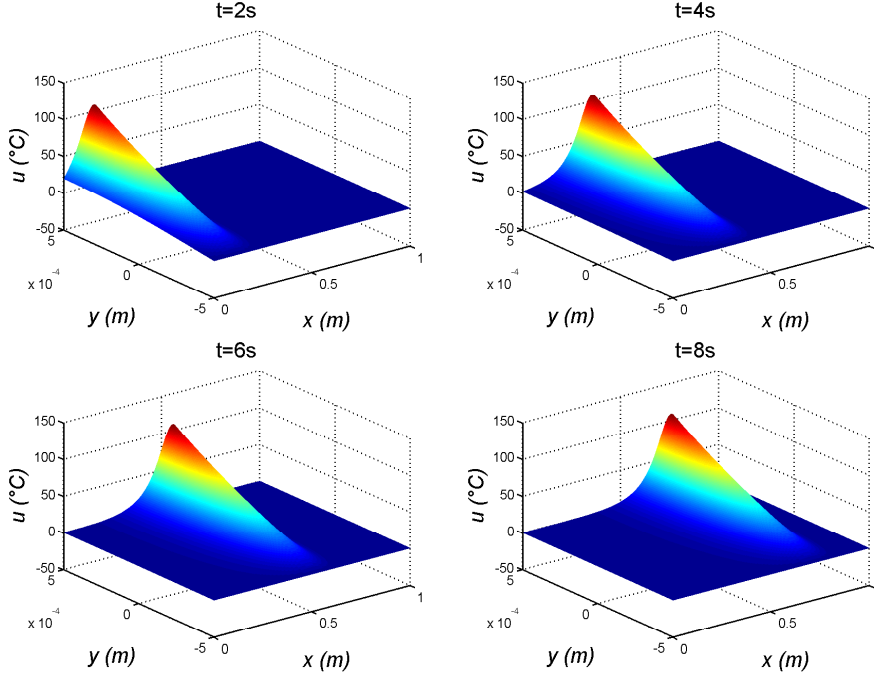


Fig. 4. Reconstructed thermal field at different times obtained from the separated representation whose functions are depicted in Fig. 3.

the temperature field in a hexahedral tape

$$u(x, y, z, t) \approx \sum_{i=1}^N X_i(x) \cdot Y_i(y) \cdot Z_i(z) \cdot T_i(t) \quad (36)$$

that is constructed by a simple extension of the iteration procedure described previously.

To prove the feasibility of such extension to higher dimensional models we consider the geometry addressed in the previous 2D example extruded in the z -direction with a depth of $0.2m$. Thus, $(x, y, z) \in \Omega$, $\Omega = \Omega_x \times \Omega_y \times \Omega_z$, with $\Omega_x = (0, L = 1m)$, $\Omega_y = (0, H = 0.001m)$ and $\Omega_z = (0, D = 0.2m)$. We consider that the thermal flux does not vary in the z -direction such that the expression previously considered remains valid:

$$q(x, z, t) = A \cdot \frac{1}{l\sqrt{2\pi}} \cdot \exp\left(-\frac{(x - vt)^2}{2l^2}\right) \quad (37)$$

whose separated representation reads again:

$$q(x, z, t) \approx \sum_{i=1}^{i=Q} F_i(x) \cdot H_i(z) \cdot G_i(t) \quad (38)$$

with $H_i(z) = 1, \forall i$.

We consider 1000, 100 and 1000 nodes for discretizing Ω_x , Ω_y and Ω_z respectively. Due to the uniformity of the solution in the z -direction, a very coarse discretization in that direction suffices, but we prefer to consider a mesh fine enough to highlight the capabilities of PGD and the interest of the coordinates separation. We consider as previously 1000 time steps. Of course, if one wants to solve the same problem using a standard mesh based discretization technique, the resulting model contains $1000 \times 100 \times 1000$, i.e. 10^8 nodes (degrees of freedom), and then, in the general case of non-linear material models one must solve 1000 times a system of size 10^8 , that is practically intractable.

By using the PGD this solution is computed in around one minute by using Matlab on a standard laptop. Instead of solving 1000 times, a systems of size 10^8 we must solve of the order of N 1D problems of size 1000 (leading to tridiagonal matrices) for computing functions $X_i(x)$ ($i = 1, \dots, N$), of the order of N 1D problems of size 100 for computing $Y_i(y)$, of the order of N 1D problems of size 1000 for calculating $Z_i(z)$ and of the order of N 1D initial value problems for computing $T_i(t)$. These calculation can be performed incredibly fast even in the non-linear case [4].

Figure 5 depicts the reconstructed solution, where for visualization purposes we represented the temperature field (using a color map) on different sections along the tap thickness, without respecting the geometrical scale.

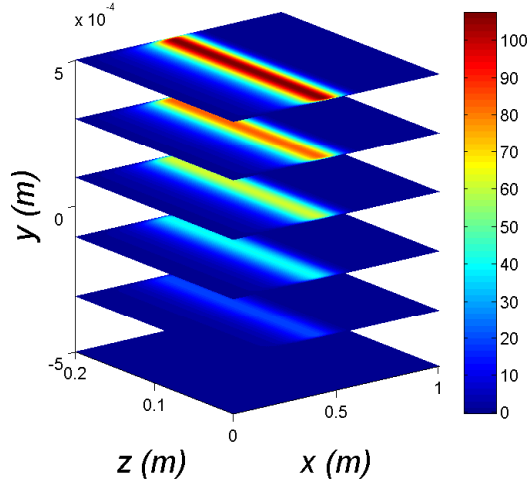


Fig. 5. Reconstructed temperature field at $t = 5s$.

4 Conclusion

In this paper we addressed the issue of fully space-time separated representations of thermal models defined in tape-type domains. These degenerate geometries are more and more considered in composite forming processes justifying the interest for fast and accurate simulations of processes, especially in the case of tricky process conditions involving moving thermal sources applied on the domain boundary.

We proposed a fully separated representation that transforms a three dimensional transient problem into a sequence of 4 one dimensional ones. The computing time savings can be simply spectacular, allowing the solution of models never solved until now due to the extremely large number of degrees of freedom.

The use of the PGD opens a number of unimaginable possibilities, some of them are being explored, others are waiting for deeper analysis.

References

- [1] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. *Journal of Non-Newtonian Fluid Mechanics*, 139, 153-176, 2006.
- [2] A. Ammar, D. Ryckelynck, F. Chinesta, R. Keunings, On the reduction of kinetic theory models related to finitely extensible dumbbells, *J. Non-Newtonian Fluid Mech.*, 134, 136-147, 2006.
- [3] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: Transient simulation using space-time separated representation. *Journal of Non-Newtonian Fluid Mechanics*, 144, 98-121, 2007.
- [4] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta. Non-incremental strategies based on separated representations: Applications in computational rheology. *Communications in Mathematical Sciences*, 8/3, 671-695, 2010.
- [5] R.A. Bialecki, A.J. Kassab, A. Fic, Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis, *Int. J. Numer. Meth. Engrg.*, 62, 774-797, 2005.
- [6] J. Burkardt, M. Gunzburger, H-Ch. Lee, POD and CVT-based reduced-order modeling of Navier-Stokes flows, *Comput. Methods Appl. Mech. Engrg.*, 196, 337-355, 2006.
- [7] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene, F. Boust. Alleviating mesh constraints: Model reduction, parallel time integration and high resolution

- homogenization. *Computer Methods in Applied Mechanics and Engineering*, 197/5, 400-413, 2008.
- [8] F. Chinesta, A. Ammar, E. Cueto. Recent advances and new challenges in the use of the Proper Generalized Decomposition for solving multidimensional models. *Archives of Computational Methods in Engineering*, 17/4, 327-350, 2010.
- [9] F. Chinesta, A. Ammar, A. Leygue, R. Keunings. An overview of the Proper Generalized Decomposition with applications in computational rheology. *Journal of Non Newtonian Fluid Mechanics*, 166, 578-592, 2011.
- [10] Ch. Ghnatios, F. Chinesta, E. Cueto, A. Leygue, P. Breitkopf, P. Villon. Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion. *Composites Part A*, .42, 11691178, 2011.
- [11] D. Gonzalez, A. Ammar, F. Chinesta, E. Cueto. Advances in the use of separated representations. *International Journal for Numerical Methods in Engineering*, 81/5, 637-659, 2010.
- [12] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data. *Comput. Methods Appl. Mech. Engrg.*, 196, 1030-1047, 2007.
- [13] P. Ladeveze, *Nonlinear computational structural mechanics*, Springer, NY, 1999.
- [14] P. Ladeveze, J.-C. Passieux, D. Néron. The LATIN multiscale computational method and the proper generalized decomposition. *Computer Methods In Applied Mechanics and Engineering*, 199/21-22, 1287-1296, 2010.
- [15] Y. Maday, E.M. Ronquist, The reduced basis element method: application to a thermal fin problem, *SIAM J. Sci. Comput.*, **26/1**, 240-258, 2004.

- [16] D. Néron, P. Ladevèze. Proper generalized decomposition for multiscale and multiphysics problems. *Archives of Computational Methods In Engineering*, 17/4, 351-372, 2010.
- [17] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta. Real-time deformable models of non-linear tissues by model reduction techniques. *Computer Methods and Programs in Biomedicine*, 91, 223-231, 2008.
- [18] S. Niroomandi, I. Alfaro, E. Cueto, F. Chinesta. Order Reduction for hyperelastic materials. *International Journal for Numerical Methods in Engineering*, 81/9, 1180-1206, 2010.
- [19] H.M. Park, D.H. Cho, The use of the Karhunen-Love decomposition for the modelling of distributed parameter systems, *Chem. Engineer. Science*, 51, 81-98, 1996.
- [20] E. Pruliere, F. Chinesta, A. Ammar. On the deterministic solution of parametric models by using the Proper Generalized Decomposition. *Mathematics and Computers in Simulation*, 81, 791-810, 2010.
- [21] D. Ryckelynck, A priori hyper-reduction method: an adaptive approach, *Journal of Computational Physics*, 202, 346-366, 2005.
- [22] D. Ryckelynck, L. Hermanns, F. Chinesta, E. Alarcon, An efficient "a priori" model reduction for boundary element models, *Engineering Analysis with Boundary Elements*, 29, 796-801, 2005.
- [23] D. Ryckelynck, F. Chinesta, E. Cueto, A. Ammar, On the a priori model reduction: overview and recent developments, *Archives of Computational Methods in Engineering*, 13/1, 91-128, 2006.