



**HAL**  
open science

# Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion

Chady Ghnatios, Francisco Chinesta, Elías Cueto, Adrien Leygue, Piotr Breitkopf, Pierre Villon

## ► To cite this version:

Chady Ghnatios, Francisco Chinesta, Elías Cueto, Adrien Leygue, Piotr Breitkopf, et al.. Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion. *Composites Part A: Applied Science and Manufacturing*, 2011, 42 (9), pp.1169-1178. 10.1016/j.compositesa.2011.05.001 . hal-01007033

**HAL Id: hal-01007033**

**<https://hal.science/hal-01007033>**

Submitted on 2 Apr 2017

**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.

Public Domain

# Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: Application to pultrusion

Ch. Ghnatios<sup>a</sup>, F. Chinesta<sup>a</sup>, E. Cueto<sup>b</sup>, A. Leygue<sup>a</sup>, A. Poitou<sup>a</sup>, P. Breitskopf<sup>c</sup>, P. Villon<sup>c</sup>

<sup>a</sup>GEM: UMR CNRS-Centrale de Nantes-Université de Nantes, 1 rue de la Noe, BP 92101, F-44321 Nantes Cedex 3, France

<sup>b</sup>Aragón Institute of Engineering Research (I3A), Universidad de Zaragoza, María de Luna, 3, E-50018 Zaragoza, Spain

<sup>c</sup>UTC-Roberval, UMR 6253, F-60200 Compiègne, France

Optimization of manufacturing processes involves the optimal choice of many process parameters. Usual strategies proceed by defining a trial choice of those parameters and then solving the resulting model. Then, an appropriate cost function is evaluated and its optimality checked. While the optimum is not reached, the process parameters should be updated by using an appropriate optimization procedure, and then the model must be solved again for the updated process parameters. Thus, a direct numerical solution is needed for each choice of the process parameters, with the subsequent impact on the computing time. In this work we propose a methodological approach to the efficient numerical modeling and optimization of thermal processes taking place in a die. This scenario is usually encountered in polymer and composites processing where material flows inside a die equipped with different heating devices. An example of such kind of processes concerns the pultrusion of composites. The main aim of this work is to describe an original approach for modeling and then optimizing the thermal process by solving only once the thermal model, and then, optimizing the process without the necessity of performing new solutions of the thermal model. For this purpose we introduce the temperatures of the heaters as extra-coordinates in the thermal model. The solution of the resulting multi-dimensional heat equation gives the temperature field for any choice of the temperature prescribed in the heaters. The curse of dimensionality is circumvented by invoking the Proper Generalized Decomposition – PGD – introduced in our former works but never until now used in the framework of process optimization.

*Keywords:*  
Proper Generalized Decomposition  
Model reduction  
Optimization  
Separated representations

## 1. Introduction

The main aim of this work is proposing an original methodological approach to efficient numerical modeling and optimization. More specifically we are focusing on thermal processes taking place in a die, although the methodology proposed in this paper could easily be extended to many other processes.

In general optimization implies the definition of a cost function and the search of the optimum process parameters (e.g. temperatures of the heaters, temperature of the material coming to the die, flow rate, ... in the case of materials flowing into a heated die) defining the minimum of that cost function. The process starts by choosing a tentative set of process parameters. Then the process is simulated by discretizing the equations defining the model of the process. The solution of the model is the most costly step of

the optimization procedure. As soon as that solution is available, the cost function can be evaluated and its optimality checked. If the chosen parameters do not define a minimum (at least local) of the cost function, the process parameters should be updated and the solution recomputed. The procedure continues until reaching the minimum of the cost function. Obviously, nothing ensures that such minimum is global, so more sophisticated procedures exist in order to explore the domain defined by the parameters and escape from local minimums traps. The parameter updating is carried out in a way ensuring the reduction of the cost function. Many techniques update the model parameters in order to move along the cost function gradient. However, for identifying the direction of the gradient one should compute not only the fields involved in the model but also the derivatives of such fields with respect to the different process parameters. The evaluation of these derivatives is not in general an easy task. Conceptually, one could imagine that by perturbing slightly only one of the parameters involved in the process optimization and then solving the resulting model, one could estimate, using a finite difference formula, the derivative of the cost function with respect to the perturbed parameter. By perturbing sequentially all the parameters we could have access

to the derivatives of the cost function with respect to all the process parameters (that is, the sensibilities) that define the cost function gradient, on which the new trial set of parameters should be chosen. There are many strategies for updating the set of process parameters and the interested reader can find most of them in the books focusing on optimization procedures.

Our interest here is not the discussion on particular optimization strategies, but pointing out that standard optimization strategies need numerous direct solutions of the problem that represents the process, one solution for each tentative choice of the process parameters. The solution of such models is a tricky task that demands important computational resources and usually implies extremely large computing times.

In this paper we propose a radically different approach, to the authors' knowledge never explored. The approach here proposed considers the unknown process parameters as new coordinates of the model. In fact, coordinates, or space dimensions, represent the (non-necessarily physical) locations at which the solution is to be represented. Thus, strictly speaking, one could compute the solution of the problem for any value of the unknown parameters (in a bounded interval). This converts those unknown parameters in new dimensions of the space in which the model is defined. This idea seems exciting but it involves a major difficulty.

This strategy faces a challenging problem if the number of parameters of the model increases. It is well known that the number of degrees of freedom for a mesh-based discretization technique (say, finite element, finite difference, ...) increases exponentially with the number of dimensions. Thus, for a hypercubic domain, the number of degrees of freedom scales with the number of nodes along each spatial direction to the power of the number of dimensions. For instance, in 2D and considering 100 nodes along each direction with a single degree of freedom per node, the resulting number of degrees of freedom becomes  $100^2$ . In 3D, the number of degrees of freedom rises to  $100^3$  and so on. This exponential increase of the number of degrees of freedom can be literally out of reach for nowadays computers even if the number of dimensions increases only moderately. This phenomenon is known as *curse of dimensionality*.

Of course, to efficiently deal with this problem a strategy different of mesh-based discretization methods should be employed. Although efficient techniques exist for moderate number of spatial dimensions, such as sparse grid methods [4], they fail when the dimensionality increases. Here, we consider the use of Proper Generalized Decompositions (PGD) [1,2,7,8]. PGD techniques construct an approximation of the solution by means of a sequence of products of separable functions. These functions are determined "on the fly", as the method proceeds, with no initial assumption on their form.

The PGD method, while it can be considered as a model reduction technique (and hence its name, a generalization of the Proper Orthogonal Decomposition – POD –) can deal very efficiently with highly multi-dimensional problems, since only a sequence of low-dimensional problems is solved. Details of the technique are provided in Section 2 of this paper.

Once an efficient strategy of dealing with high-dimensional solutions has been defined, the numerical solution of problems with unknown data becomes straightforward. As mentioned before, the strategy here proposed consists of considering the unknown parameters as new coordinates of the model. Thus, the solution is computed only once and it allows to have access to the unknown field, as well as to the explicit expression of its derivatives, for any possible choice of the model parameters by a simple particularization of the parametric solution, that is, by a simple postprocessing.

As can be readily noticed, the potential of the technique for inverse identification, optimization, etc. seems to be huge. In this pa-

per we propose a methodological approach in this direction. In what follows we are considering a thermal model of a material moving through a die equipped with some heating devices on the die walls. We could consider as process parameters the temperatures prescribed in the different heaters, the flow rate, the temperature of the material coming into the die, etc. For the sake of simplicity in what follows we are restricting the parametric space to the heating devices temperatures. The choice of the cost function depends on each particular process. There are many choices and because in this work we are more interested in proposing and illustrating a new methodological modeling and optimization approach than in analyzing deeply a particular process, we will restrict our analysis to a simplified model of pultrusion processes. Pultrusion is a continuous process to produce constant cross-sectional profile composites. During this process, fiber reinforcements are saturated with resin, which are then pulled through a heated die. The resin gradually cures inside the die while generating heat. At the exit, pullers draw the composite out and a traveling cut-off saw cuts it at the designed length. We consider the process conditions described in [5] and sketched in Fig. 1.<sup>1</sup>

For decades, engineers have relied on experience to define optimal parameters for pultrusion processes, pushed by the popularization of this technique in industry [10,12]. Nowadays, this process has been extended to thermoplastic resins as well as to reactive systems, in which the monomers polymerize inside the die, an efficient route for considering high viscosity thermoplastic resins. In these scenarios new efficient optimization procedures are urgently needed. Standard optimization approaches and others based on the use of genetic algorithms have been recently proposed and applied (see for example [5] and the references therein). However the efficiency of those approaches seems to be limited to a reduced number of process parameters because one must solve a thermal model for each choice of the process parameters, and it is well known that when the dimension of the parametric space increases the exploration of the space defined by the process parameters becomes more and more arduous, needing for numerous, sometimes excessive, solutions of the model governing the process.

If we consider the thermal model related to the pultrusion process sketched in Fig. 1, whose parametric space reduces to the temperatures prescribed at the three heating devices,  $\theta_1$ ,  $\theta_2$  and  $\theta_3$ , we could summarize traditional optimization procedures as follows:

- *Until reaching a minimum of the cost function  $\mathcal{C}(\theta_1, \theta_2, \theta_3)$  proceed by:*
  1. Computing the temperature field related to the trial choice of the process parameters, i.e.  $u(\mathbf{x}; \theta_1, \theta_2, \theta_3)$ .
  2. Computing the cost function  $\mathcal{C}(\theta_1, \theta_2, \theta_3)$  from the just calculated thermal field.
  3. Checking the optimality. While the optimum is not reached, update the process parameters by using an appropriate strategy and comeback to step 1 for another solution of the thermal model for the process parameters just updated.

In the approach that we propose in this work the procedure is substantially different. It proceeds as follows:

- Compute the thermal field for any possible choice of the process parameters:  $u(\mathbf{x}, \theta_1, \theta_2, \theta_3)$  (here the heaters temperatures play the same role that the space coordinates), the problem becoming multi-dimensional.

<sup>1</sup> For interpretation of color in all figures, the reader is referred to the web version of this article.

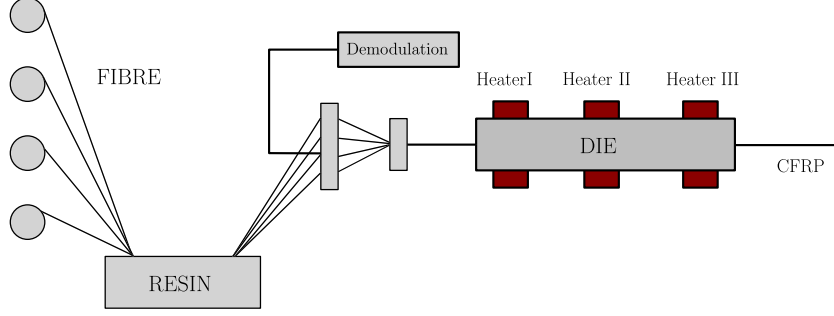


Fig. 1. Pultrusion process.

• Until reaching a minimum of the cost function  $\mathcal{C}(\theta_1, \theta_2, \theta_3)$  proceed by:

1. Particularizing the parametric solution to the considered values of the process parameters.
2. Computing the cost function  $\mathcal{C}(\theta_1, \theta_2, \theta_3)$  from the just calculated thermal field.
3. Checking the optimality. While the optimum is not reached, update the process parameters by using an appropriate strategy and comeback to step 1 for another particularization of the parametric solution.

Thus, in our proposal the thermal model is solved only once and then it is particularized for any choice of the process parameters. The price to pay is the necessity of solving a multi-dimensional thermal model that now has as coordinates the physical space  $\mathbf{x}$  and all the process parameters, i.e. the three heaters temperatures in the example addressed here.

Obviously, the solution of the resulting multi-dimensional model is a tricky task if one consider a standard mesh based discretization strategy because the number of degrees of freedom increases exponentially with the dimensionality of the model. To circumvent this serious difficulty, also known as *curse of dimensionality*, we consider a separated representation of the temperature field in the PGD framework, in which the temperature reads:

$$u(\mathbf{x}, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(\mathbf{x}) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (1)$$

To build up such separated representation we only need to compute the functions defined in the space domain  $F_i(\mathbf{x})$  and the one-dimensional functions  $\Theta_{ji}(\theta_j)$ ,  $j=1,2,3$ , defined in the intervals in which the heaters temperatures can evolve.

In Section 2 we revisit the main ideas involved in the construction of such separated representation. In Section 3 we describe the procedure to include the boundary conditions as extra-coordinates. In Section 4 a simplified pultrusion model is addressed. Then, in Section 5 we consider one possible optimization strategy based on the combination of the already computed parametric solution with a minimization strategy based on the use of moving least squares on the response surface. Due to the methodological purposes of this paper a simple cost function will be considered, enforcing a constant temperature of the flowing material on die outlet. Despite the simple scenario here addressed the reader can appreciate the huge potentialities that such approach represents.

## 2. Illustrating the solution of multi-dimensional parametric models by using the PGD

Imagine for example that you are interested in solving the heat equation but that you do not know the material thermal conductivity, because it has a stochastic nature or simply be-

cause prior to solve the thermal model you should measure it. You have three possibilities: (i) you wait to know the conductivity before solving the heat equation (a conservative solution!); (ii) you solve the equation for many values of the conductivity (a sort of Monte Carlo) and then the work is done (a sort of brute force approach!); or (iii) you solve the heat equation only once for any value of the conductivity (the cleverest alternative!). Obviously the third alternative is the most exciting one. To compute this “magic” solution it suffices to introduce the conductivity as an extra coordinate, playing the same role than the standard space and time coordinates, even if the model does not involve derivatives with respect to this extra-coordinate. This procedure runs very well, and can be extended for introducing many other extra-coordinates: the source term, initial condition, etc. It is easy to understand that after performing this type of calculations, a posteriori inverse identification or optimization can be easily handled, but we will come back to these potential applications later.

In what follows, we illustrate the construction of the Proper Generalized Decomposition [11,1,2,7,14] of the solution by considering the following simple parametric heat transfer equation:

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0 \quad (2)$$

where  $(\mathbf{x}, t, k) \in \Omega \times I \times \mathfrak{I}$ .  $\Omega \subset \mathbb{R}^d$  is the physical space in which the model is defined,  $I$  is the time interval in which the solution is found and finally  $\mathfrak{I}$  represents the interval in which the conductivity could take its value.

For the sake of simplicity the source term is assumed to be constant, i.e.  $f = \text{cte}$ . Because the conductivity is considered unknown, it is assumed to be a new coordinate defined in the interval  $\mathfrak{I}$ . Thus, instead of solving the thermal model for different values of the conductivity parameter we are introducing it as a new coordinate. Therefore, we solve a more general problem, but obviously the price to pay is the increase of the model dimensionality. However, as the complexity of PGD scales only linearly (and not exponentially) with the space dimension, the consideration of the conductivity as a new coordinate still allows one to obtain a fast and cheap solution. Thus, in this case, the solution of Eq. (2) is searched under the form:

$$u(\mathbf{x}, t, k) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \quad (3)$$

To start the PGD algorithm, we assume that the approximation at iteration  $n$  is already known:

$$u^n(\mathbf{x}, t, k) = \sum_{i=1}^{i=n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \quad (4)$$

and that at the present iteration we look for the next functional product  $X_{n+1}(\mathbf{x}) \cdot T_{n+1}(t) \cdot K_{n+1}(k)$  that for alleviating the notation will

be denoted by  $R(\mathbf{x}) \cdot S(t) \cdot W(k)$ . Before solving the resulting non-linear model related to the calculation of these three functions, a model linearization is performed. The simplest choice consists in using an alternating-directions fixed-point algorithm. First of all, we proceed by assuming  $S(t)$  and  $W(k)$  given at the previous iteration of the non-linear solver and then computing  $R(\mathbf{x})$ . From the just updated  $R(\mathbf{x})$  and the previously used  $W(k)$  we can update  $S(t)$ . Finally from the just computed  $R(\mathbf{x})$  and  $S(t)$  we update  $W(k)$ . The procedure continues until reaching convergence. The converged functions  $R(\mathbf{x})$ ,  $S(t)$  and  $W(k)$  allow us to define all the needed functions:  $X_{n+1}(\mathbf{x}) = R(\mathbf{x})$ ,  $T_{n+1}(t) = S(t)$  and  $K_{n+1}(k) = W(k)$ . The explicit form of each one of the just referred steps is widely described in the Appendix A.

**Remark 1.** The construction of each term in Eq. (3) needs a certain number of iterations because of the non-linearity of the problem related to the approximation given by Eq. (4). Denoting by  $m_i$  the number of iterations needed for computing the  $i$ th sum in Eq. (3), let  $m = \sum_{i=1}^{i=N} m_i$  be the total number of iterations involved in the construction of the separated approximation, Eq. (3). It is easy to note that the solution procedure involves the solution of  $m$   $dD$  ( $d=1,2,3$ ) problems related to the construction of the space functions  $X_i(\mathbf{x})$ ,  $i=1, \dots, N$ ;  $m$  1D ordinary differential equations related to the construction of functions  $T_i(t)$  and, finally,  $m$  linear systems related to the definition of functions  $K_i(k)$ . In general  $m_i$  rarely exceeds ten. On the other hand the number  $N$  of sums needed to approximate the solution of a given problem depends on the solution regularity itself, but all the experiments carried out so far reveal that this number ranges from few 10s to slightly more than one hundred. Thus, we can conclude that the complexity of the solution procedure is of some tens of solutions defined in the  $dD$  space (the cost related to the one dimensional time problems being negligible with respect to the one related to the space problems [6]). On the contrary, if we follow a classical approach, we should solve a  $dD$  problem at each time step and for each value of the parameter  $k$ . In usual applications the complexity can easily reach millions of space solutions. In [8] the authors proved that the CPU time savings by applying the PGD can be of several orders of magnitude.

**Remark 2.** Note also that another possibility exists consisting in the separation of the three-dimensional physical space into a sequence of one-dimensional ones:

$$u(x, y, z, t, k) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \cdot T_i(t) \cdot K_i(k) \quad (5)$$

This possibility further reduces the complexity mentioned before, allowing to solve models with some  $10^9$  degrees of freedom in few minutes in a laptop. Techniques to cope in this framework with non-paralelepipedic domains in three or more dimensions have been analyzed in [9], in which a domain penalization strategy was proposed.

### 3. Parametric boundary conditions

Very often, in the optimization of an industrial process, it is necessary to solve the problem for different boundary conditions. Boundary conditions do not behave as any other parameter in the PGD, and therefore deserve some additional comments. In general, it is needed to perform a change of variable to introduce the boundary condition into the differential equation and then define it as an extra coordinate. To illustrate this procedure we consider, for the sake of simplicity and without any loss of generality, the following simple problem:

$$-\Delta u = f \text{ in } \Omega \quad (6)$$

subjected to the boundary conditions:

$$u = g \neq 0 \text{ on } \Gamma \equiv \partial\Omega \quad (7)$$

Let us assume that we are able to find a function  $\psi$ , continuous in the closure of  $\Omega$ ,  $\bar{\Omega}$ , such that  $-\Delta \psi \in L_2(\Omega)$  verifying Eq. (7). Then, the solution of the problem given by Eqs. (6), (7) can be obtained straightforwardly by

$$u = \psi + z, \quad (8)$$

where we thus face a problem in the  $z$  variable

$$-\Delta z = f + \Delta\psi \text{ in } \Omega \quad (9)$$

$$z = 0 \text{ on } \Gamma \quad (10)$$

easily solvable by the PGD method presented before. This procedure was deeply analyzed in [9].

The introduction of the value of the boundary condition ( $g$  in Eq. (7)) as an extra-coordinate generates an extra dimension in the problem, which means dramatically increasing the computation cost of the problem solution in classical mesh-based numerical methods for discretizing partial differential equations.

#### 3.1. Illustrating the main ideas from a numerical example

To fix the ideas, let us start by considering the one dimensional differential equation that follows:

$$-\frac{\partial^2 u}{\partial x^2} = 1 \quad (11)$$

to be solved in the domain  $x \in \Omega = (0, L = 10)$  and for the following boundary conditions:

$$\begin{cases} u(0) = 0 \\ u(L) = y \end{cases} \quad (12)$$

where  $y$  is a parameter taking values within the interval  $\alpha$ . To prescribe non-homogeneous boundary conditions in the framework of the PGD, it is a common practice, as just argued, to perform a change of variable to transform the non-homogeneous boundary conditions into homogeneous ones [9]. To this end, often simple procedures suffice. For instance, the following function  $g(x, y)$  satisfies all the prescribed boundary conditions (12):

$$g(x, y) = \frac{x \cdot y}{L} \quad (13)$$

Since, as we have seen in Section 2, the problem solution  $u(x, y)$  can be written in the separated form:

$$u(x, y) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \quad (14)$$

we can introduce the function verifying the essential boundary conditions,  $g(x, y)$ , as the first mode of the solution  $X_1(x) \cdot Y_1(y)$ . In other words,

$$X_1(x) \cdot Y_1(y) = \frac{x \cdot y}{L} \quad (15)$$

and

$$u(x, y) \approx X_1(x) \cdot Y_1(y) + \sum_{i=2}^{i=N} X_i(x) \cdot Y_i(y) \quad (16)$$

Now we proceed as shown in Section 2. The trial function after  $n$  iterations reads:

$$u(x, y) = X_1(x) \cdot Y_1(y) + \sum_{i=2}^{i=n} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y) \quad (17)$$

By replacing Eq. (17) into the weak form of Eq. (11), and repeating the same procedure used in Section 2 and the Appendix A we have:

- Computing  $R(x)$  from  $S(y)$ . In this case the test function is written as:

$$u^*(x, y) = R^*(x) \cdot S(y) \quad (18)$$

and the resulting weak form reads:

$$\int_{\Omega} \int_{\alpha} R^* \cdot S \cdot Y_1 \cdot \frac{d^2 X_1}{dx^2} + R^* \cdot S \cdot \sum_{i=2}^{i=n} \frac{d^2 X_i}{dx^2} \cdot Y_i \left( + R^* \cdot S \cdot \frac{d^2 R}{dx^2} \cdot S + R^* \cdot S \right) dy \cdot dx = 0 \quad (19)$$

As functions involving the parametric coordinate  $y$  are assumed known in the present step, integrals in the parametric domain  $\alpha$  can be calculated, leading to a problem defined in the space domain  $\Omega$  whose solution is precisely the searched unknown function  $R(x)$ .

- Computing  $S(y)$  from  $R(x)$ . In this case the test function is:

$$u^*(x, y) = R(x) \cdot S^*(y) \quad (20)$$

and the weak function now reads:

$$\int_{\Omega} \int_{\alpha} S^* \cdot R \cdot Y_1 \cdot \frac{d^2 X_1}{dx^2} + S^* \cdot R \cdot \sum_{i=2}^{i=n} \frac{d^2 X_i}{dx^2} \cdot Y_i \left( + S^* \cdot R \cdot \frac{d^2 R}{dx^2} \cdot S + S^* \cdot R \right) dy \cdot dx = 0 \quad (21)$$

As functions involving the space coordinate  $x$  are assumed known in the present step, integrals in the domain  $\Omega$  can be calculated, leading to a problem defined in the parametric domain  $\alpha$  whose solution is precisely the searched unknown function  $S(y)$ .

Functions  $R$  and  $S$  vanish in the boundary of the domain. That is mandatory, because the boundary conditions are already satisfied by the first mode. The new modes improve the solution inside the domain. The reconstructed solution is depicted in Fig. 2 for any value of  $y \in [0, 5]$ .

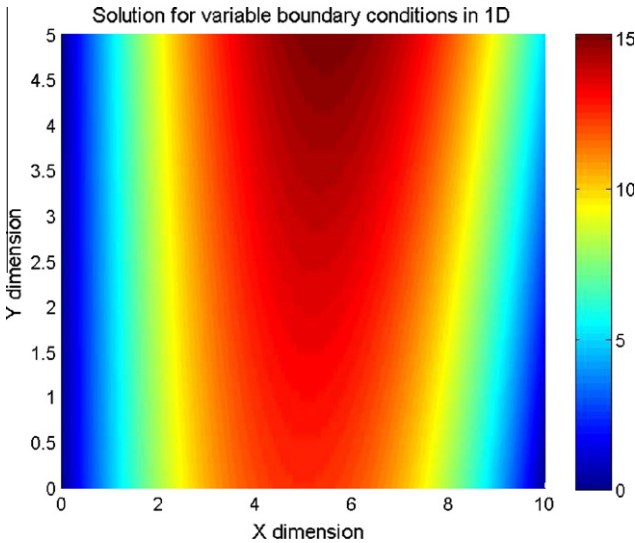


Fig. 2. Solution of the parametric Eq. (11). The solution  $u(x)$  is depicted for any possible boundary condition  $y \in [0, 5]$ .

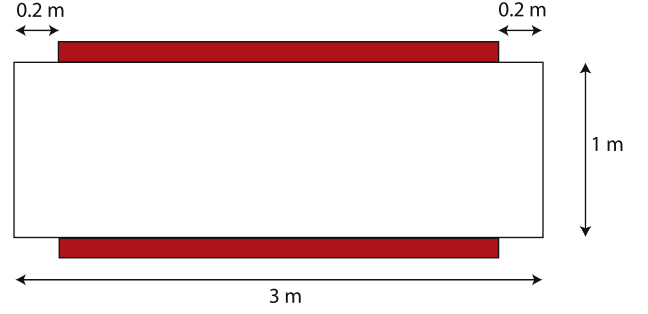


Fig. 3. Domain geometry and boundary conditions.

### 3.2. Boundary conditions as parameters

Once the main ingredient of the method has been introduced, we are generalizing the procedure for treating more complex models. For the sake of simplicity, we describe the solution of a two dimensional model defined in  $\Omega = (0, L_1 = 3) \times (0, L_2 = 1)$ . Let us consider the basic steady-state heat transfer equation:

$$-K \cdot \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = q \quad (22)$$

On the domain boundary  $\Gamma \equiv \partial\Omega$  either the temperature or the heat flux are enforced, the first one on the part of the boundary  $\Gamma_D$  and the second one on the remaining part  $\Gamma_N = \Gamma - \Gamma_D$ . Fig. 3 shows a sketch of the domain geometry and the boundary conditions.

We assume the prescribed temperatures given by:

$$\begin{cases} u(0, y) = U_0 \\ u(\epsilon \leq x \leq L_1 - \epsilon, y = 0) = \theta \\ u(\epsilon \leq x \leq L_1 - \epsilon, y = L_2) = \theta \end{cases} \quad (23)$$

where  $\epsilon = 0.2$  m,  $U_0 = 300$  K is a constant value and  $\theta$  any value defined in a certain interval  $\alpha$ .

A null heat flux is enforced in the remaining part of the domain boundary, i.e.  $\nabla u \cdot \mathbf{n}|_{\Gamma_N} = 0$ , where  $\mathbf{n}$  represents the outward unit vector defined on the domain boundary.

Traditionally, this type of problems is solved by considering different values of that parameter and solving the resulting model for each one of these values, with the consequent impact on the computing time. Here we propose to introduce this parameter  $\theta$  as a new extra-coordinate, and then solving only once the resulting multi-dimensional model. The solution  $u(x, y, \theta)$  allows one to have access to the value of the field  $u$  at each space position  $(x, y)$  and for each value of the parameter  $\theta$  in the interval  $\alpha$  in which it is defined ( $\theta \in (300, 500)$  in this example).

Prior to solve the problem it is necessary to find a function satisfying the essential boundary conditions, in order to perform the change of variable illustrated in the previous section. For arbitrarily-shaped domains a somewhat more sophisticated technique was introduced by the authors in [9] and will not be reproduced here. The interested reader can refer to this reference.

Therefore, the solution is assumed to be of the form:

$$u(x, y, \theta) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_i(\theta) \quad (24)$$

In the present case, it is difficult to find a single function satisfying all boundary conditions, unlike the previous example. Two initial modes have been considered in the separated representation of the solution. The first mode fixes the value of the solution at  $x = 0$  by prescribing the nodal values at that location, and vanishing elsewhere. The second one fixes the boundary conditions at  $y = 0$  and  $y = L_2$ , also vanishing at the remaining nodes of the domain. This

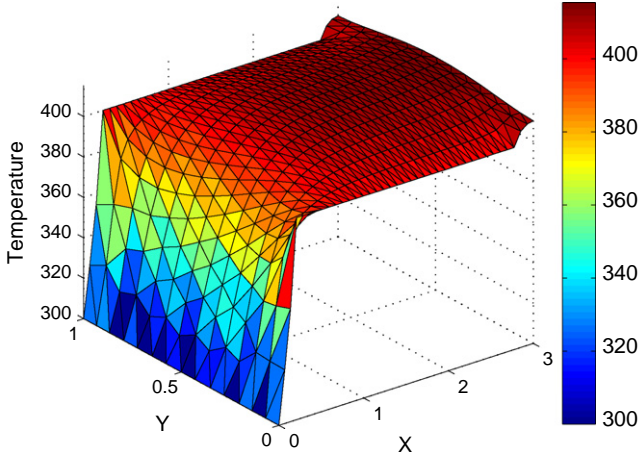


Fig. 4. Temperature field for  $\theta = 400$  K.

*ad hoc* procedure is not always easy to implement. For more sophisticated cases, a special type of transfinite interpolation was introduced in [9].

After applying the change of variable related to the imposition of essential boundary conditions, we solve Eq. (22) with homogeneous Dirichlet boundary conditions. The solution is represented in Fig. 4 for  $\theta = 400$  K.

Comparing this solution to the standard finite element one, which is computed for the precise boundary conditions mentioned before to avoid the increase of the dimensionality, a very small difference is found, lower than  $10^{-12}$  using the  $L_2$ -norm. For more details on the accuracy of the technique in general settings, we address the reader to [7].

#### 4. Parametric thermal model of a heated die in pultrusion processes

In modeling the pultrusion process, as sketched in Fig. 1, we consider the thermal process within the die as modeled by the following two-dimensional convection–diffusion equation:

$$\rho \cdot C \left( \frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} \right) = k \Delta u + q \quad (25)$$

where  $k$  is the thermal conductivity,  $q$  is the internal heat generated by the resin curing reaction,  $\rho$  is the density,  $C$  is the specific heat and  $v$  is the extruded profile speed. The material flowing inside the die is in contact with the die wall. Thus, conduction is the only heat transfer mechanism taking place.

The die is equipped with three heaters as depicted in Fig. 5 whose temperatures constitute the process parameters to be optimized. For the sake of simplicity we consider constant profile velocity  $v$  and inlet temperature  $U_0$ , all of them assumed known. The curing kinetics was coupled with the thermal field as described in [15].

Thus, the temperature field  $u$  depends on five different coordinates, the two space coordinates  $(x, y)$  and the three temperatures prescribed in three regions on the die wall. In fact  $u = u(x, y, \theta_1, \theta_2, \theta_3)$ , where  $\theta_1, \theta_2$  and  $\theta_3$  are the temperature of the heaters I, II and III, depicted in Fig. 5 respectively. The separated representation of  $u$  reads:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (26)$$

Temperatures are assumed prescribed as Dirichlet boundary conditions on  $\Gamma_D$  whereas a null flux is prescribed on the complementary part  $\Gamma_N = \Gamma - \Gamma_D$ , i.e.  $\nabla u \cdot \mathbf{n}|_{\Gamma_N} = 0$ .

The prescribed temperatures write:

$$\begin{cases} u(x = 0, y, \theta_1, \theta_2, \theta_3) = U_0 \\ u(x \in L_1, y = 0 \text{ or } y = h, \theta_1, \theta_2, \theta_3) = \theta_1 \\ u(x \in L_2, y = 0 \text{ or } y = h, \theta_1, \theta_2, \theta_3) = \theta_2 \\ u(x \in L_3, y = 0 \text{ or } y = h, \theta_1, \theta_2, \theta_3) = \theta_3 \end{cases} \quad (27)$$

where  $L_1, L_2$  and  $L_3$  are the intervals of the  $x$ -coordinate related to heaters I, II and III respectively.  $L = 0.5$  m is the length of the die and  $h = 0.04$  m its width. The parameters  $\theta_1, \theta_2$  and  $\theta_3$  take values in the intervals  $\mathcal{I}_1, \mathcal{I}_2$  and  $\mathcal{I}_3$  respectively.

To solve this problem, we need to define some functional products in order to introduce the boundary conditions (27) into the partial differential equation (four modes suffices for this end) and then proceed by applying the Proper Generalized Decomposition as described in Section 2 and the Appendix A.

#### 5. Optimization strategy

In this section we consider the parametric solution already computed:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (28)$$

The objective of the optimization procedure consists of the determination of the process parameters  $\theta_1, \theta_2$  and  $\theta_3$  in order to minimize an appropriate cost function depending on the considered physics. In this work we are only interested in describing the main ingredients involved in the proposed optimization strategy, and for this reason, in what follows, we consider a cost function without any particular physical significance.

The cost function is defined from the difference between the temperature on the outflow boundary  $x = L$ ,  $u(x = L, y; \theta_1, \theta_2, \theta_3)$ , and the desired one  $\bar{u}(y)$ . Thus, the cost function writes:

$$\mathcal{C}(\theta_1, \theta_2, \theta_3) = \frac{1}{2} \int_0^h (u(x = L, y; \theta_1, \theta_2, \theta_3) - \bar{u}(y))^2 dy \quad (29)$$

We look for a uniform profile of temperatures on the outlet. This kind of condition is quite difficult to fulfill due to the heat conduction mechanism as well as the internal heat generation. We consider this cost function only to prove the robustness of the proposed optimization approach.

The parametric domain is defined by  $\mathcal{I} = \mathcal{I}_1 \times \mathcal{I}_2 \times \mathcal{I}_3$  in which the optimal solution is sought. The optimal solution constitutes a minimum of the cost function (29). We denote points in the parametric domain by  $\theta$  with components  $(\theta_1, \theta_2, \theta_3)$ .

The algorithm starts by considering an arbitrary point within  $\mathcal{I}$ . Then, the gradient and Hessian defined by the cost function are computed in order to apply a Newton strategy. For computing both, we need to define an appropriate approximation of the cost function in the vicinity of the evaluation point. By defining a quadratic approximation we could compute the gradient and the Hessian, and because the approximation is quadratic the Newton algorithm is an appropriate choice for performing the minimization.

Let  $\theta^0 \in \mathcal{I}$  be the starting point. We consider a small paralelepipedic volume  $\mathcal{P}^0$  in the parametric space centered at that point in which the cost function will be approximated using a polynomial approximation of a certain order, and whose edges lengths  $l_1, l_2$  and  $l_3$ , in the directions of the axes  $\theta_1, \theta_2$  and  $\theta_3$  respectively, depend on the desired order of the approximation as well as on the mesh used for discretizing the intervals  $\mathcal{I}_1, \mathcal{I}_2$  and  $\mathcal{I}_3$  in which  $\theta_1, \theta_2$  and  $\theta_3$ , respectively, are defined. Then, a number of points  $\theta_i^0, i = 1, \dots, n_{LH}$ , are considered within  $\mathcal{P}^0$  according to the LATIN hypercube technique.

The cost function  $\mathcal{C}_i^0$ ,  $i = 1, \dots, n_{LH}$  is computed at those points  $\theta_i^0$ ,  $i = 1, \dots, n_{LH}$ . Now, a quadratic approximation of  $\mathcal{C}(\theta_1, \theta_2, \theta_3)$  could be defined in  $\mathcal{P}^0$ ,  $\mathcal{C}^h(\theta \in \mathcal{P}^0)$ , and then both the gradient and the Hessian computed.

Obtaining a smooth enough, quadratic approximation of a field given by scattered data is not always easy. For this purpose we make use of a centered moving least square (MLS) strategy that we revisit in the next section.

### 5.1. The Moving Least Squares approximation revisited

In this section we revisit the Moving Least Squares – MLS – approximation introduced by Nayroles et al. [13] that will be then used for defining the optimization strategy.

We look for a local approximation of a generic function  $u(\mathbf{x})$  at point  $\mathbf{x}$ ,  $u^h(\mathbf{x})$ , based on the nodal values  $u_i$  of the function  $u(\mathbf{x})$  at a limited number  $n$  of points  $\mathbf{x}_i$  close to  $\mathbf{x}$ . Thus, the unknown field is approximated in the vicinity of  $\mathbf{x}$  by:

$$\mathbf{u}^h(\mathbf{x}) = \mathbf{p}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \quad (30)$$

with  $\mathbf{p}(\mathbf{x})$  a polynomial basis, i.e.  $\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, x_1 \cdot x_2]$  and  $\mathbf{p}^T(\mathbf{x}) = [1, x_1, x_2, x_1 \cdot x_2, x_1^2, x_2^2]$  for a bilinear and quadratic basis, respectively, in 2D, and  $\mathbf{a}(\mathbf{x})$  a vector of unknown coefficients. In order to determine  $\mathbf{a}(\mathbf{x})$ , we define the functional  $J(\mathbf{a})$  that must be minimized with respect to  $\mathbf{a}(\mathbf{x})$ :

$$J(\mathbf{a}) = \frac{1}{2} \sum_{i=1}^n w_i(\mathbf{x}) (\mathbf{p}(\mathbf{x}_i) \cdot \mathbf{a}(\mathbf{x}) - u_i)^2 \quad (31)$$

where  $u_i$  are the nodal unknowns associated with the neighboring nodes  $\mathbf{x}_i$  of point  $\mathbf{x}$  and  $w_i(\mathbf{x})$  is a weighting function whose value decreases as the distance between  $\mathbf{x}_i$  and  $\mathbf{x}$  increases. The weight function  $w_i(\mathbf{x})$  vanishes at a finite distance from  $\mathbf{x}_i$ , called radius of influence. In general it is positive, takes its maximum value (usually unit) at position  $\mathbf{x}_i$  and decreases continuously over the domain of influence. There are many choices according to the problem to be solved. The reader can refer to [3] for more details on the properties of this function and the most used ones. The minimization of  $J(\mathbf{a})$  with respect to the unknown coefficient  $a_j(\mathbf{x})$  leads to:

$$\frac{\partial J(\mathbf{a})}{\partial a_j(\mathbf{x})} = \sum_{k=1}^n a_k \left[ \sum_{i=1}^n w_i(\mathbf{x}) p_j(\mathbf{x}_i) p_k(\mathbf{x}_i) \right] - \sum_{i=1}^n w_i(\mathbf{x}) p_j(\mathbf{x}_i) u_i = 0 \quad (32)$$

which leads to the linear system:

$$\mathbf{A}(\mathbf{x}) \mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x}) \mathbf{u} \quad (33)$$

where matrices  $\mathbf{A}(\mathbf{x})$  and  $\mathbf{B}(\mathbf{x})$  are defined by:

$$A_{jk}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \cdot p_j(\mathbf{x}_i) \cdot p_k(\mathbf{x}_i) \quad (34)$$

$$B_{ij}(\mathbf{x}) = w_i(\mathbf{x}) \cdot p_j(\mathbf{x}_i) \quad (35)$$

Substituting  $\mathbf{a}(\mathbf{x})$  in Eq. (30), results in:

$$u^h(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}) \mathbf{u} \quad (36)$$

The difference between the diffuse finite element and the element free Galerkin schemes comes from the evaluation of the shape function derivatives. In the first scheme only the term  $\mathbf{p}(\mathbf{x})$  in Eq. (30) is derived

$$\frac{\partial u^h(\mathbf{x})}{\partial x_i} = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_i} \cdot \mathbf{a}(\mathbf{x}), \quad i = 1, 2 \quad (37)$$

whereas all terms depending on  $\mathbf{x}$  are derived in the element free Galerkin approach, according to:

$$\frac{\partial u^h(\mathbf{x})}{\partial x_i} = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_i} \cdot \mathbf{a}(\mathbf{x}) + \mathbf{p}(\mathbf{x}) \cdot \frac{\partial \mathbf{a}(\mathbf{x})}{\partial x_i}, \quad i = 1, 2. \quad (38)$$

### 5.2. Centered moving least squares

We define the polynomial basis  $\mathbf{q}$  centered at the evaluation point  $\mathbf{x}$ . For a node  $\mathbf{x}_i$  we have

$$\mathbf{q}^T(\mathbf{x}^i - \mathbf{x}) = [1, (x_1^i - x_1), (x_2^i - x_2), (x_1^i - x_1) \cdot (x_2^i - x_2), \dots] \quad (39)$$

For the sake of clarity, and without loss of generality, from now on we consider the bilinear basis  $\mathbf{q}^T = [1, (x_1^i - x_1), (x_2^i - x_2), (x_1^i - x_1) \cdot (x_2^i - x_2)]$ .

It is easy to verify the relationship

$$\mathbf{Q}(\mathbf{x}) \cdot \mathbf{p}(\mathbf{x}_i) = \mathbf{q}(\mathbf{x}^i - \mathbf{x}) \quad (40)$$

with the matrix  $\mathbf{Q}(\mathbf{x})$  given by:

$$\mathbf{Q}(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -x_1 & 1 & 0 & 0 \\ -x_2 & 0 & 1 & 0 \\ x_1 \cdot x_2 & -x_2 & -x_1 & 1 \end{pmatrix} \quad (41)$$

Because matrix  $\mathbf{Q}(\mathbf{x})$  is nonsingular (it corresponds to a change of basis in a polynomial vector space), we can write:

$$\mathbf{p}(\mathbf{x}_i) = \mathbf{Q}^{-1}(\mathbf{x}) \mathbf{q}(\mathbf{x}^i - \mathbf{x}) \quad (42)$$

from which the approximation  $u(\mathbf{x}_i)$  reads:

$$u(\mathbf{x}_i) = \mathbf{p}(\mathbf{x}_i) \cdot \mathbf{a}(\mathbf{x}) = \mathbf{q}^T(\mathbf{x}^i - \mathbf{x}) \mathbf{Q}^{-T}(\mathbf{x}) \mathbf{a}(\mathbf{x}) = \mathbf{q}^T(\mathbf{x}^i - \mathbf{x}) \boldsymbol{\alpha}(\mathbf{x}) \quad (43)$$

where

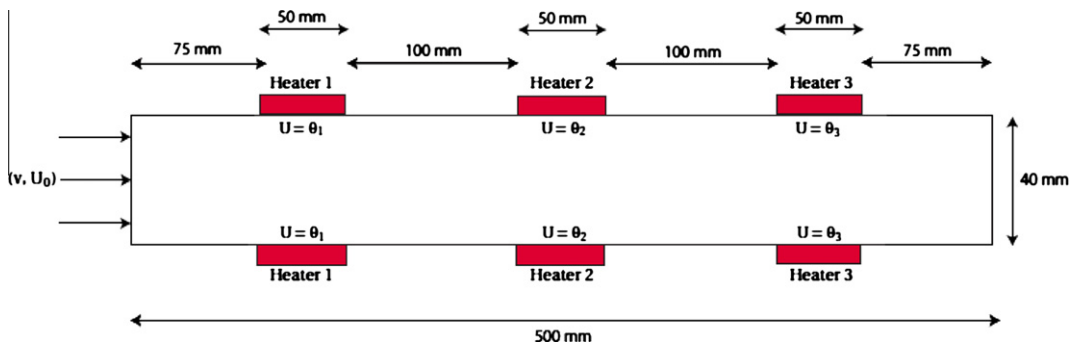


Fig. 5. Domain geometry and boundary conditions.



$$\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{Q}^{-T}(\mathbf{x})\mathbf{a}(\mathbf{x}) \quad (44)$$

with

$$\mathbf{Q}^{-T}(\mathbf{x}) = \begin{pmatrix} 1 & x_1 & x_2 & x_1 \cdot x_2 \\ 0 & 1 & 0 & x_2 \\ 0 & 0 & 1 & x_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (45)$$

Now, defining the functional

$$J(\boldsymbol{\alpha}) = \frac{1}{2} \sum_{i=1}^n w_i(\mathbf{x})(\mathbf{q}(\mathbf{x}_i - \mathbf{x}) \cdot \boldsymbol{\alpha}(\mathbf{x}) - u_i)^2 \quad (46)$$

and proceeding as described in the previous section, we can compute  $\boldsymbol{\alpha}(\mathbf{x})$  from

$$\boldsymbol{\alpha}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{u} \quad (47)$$

where

$$A_{jk}(\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) \cdot q_j(\mathbf{x}_i - \mathbf{x}) \cdot q_k(\mathbf{x}_i - \mathbf{x}) \quad (48)$$

$$B_{ij}(\mathbf{x}) = w_i(\mathbf{x}) \cdot q_j(\mathbf{x}_i - \mathbf{x}) \quad (49)$$

Now, by computing the matrix–vector product (44) we obtain:

$$\begin{cases} \alpha_1 = a_1 + a_2 \cdot x_1 + a_3 \cdot x_2 + a_4 \cdot x_1 \cdot x_2 \\ \alpha_2 = a_2 + a_4 \cdot x_2 \\ \alpha_3 = a_3 + a_4 \cdot x_1 \\ \alpha_4 = a_4 \end{cases} \quad (50)$$

that corresponds to

$$\begin{cases} \alpha_1 = \mathbf{p}(\mathbf{x}) \cdot \mathbf{a}(\mathbf{x}) \equiv u^h(\mathbf{x}) \\ \alpha_2 = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_1} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta u^h(\mathbf{x})}{\delta x_1} \\ \alpha_3 = \frac{\partial \mathbf{p}(\mathbf{x})}{\partial x_2} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta u^h(\mathbf{x})}{\delta x_2} \\ \alpha_4 = \frac{\partial^2 \mathbf{p}(\mathbf{x})}{\partial x_1 \partial x_2} \cdot \mathbf{a}(\mathbf{x}) \equiv \frac{\delta^2 u^h(\mathbf{x})}{\delta x_1 \delta x_2} \end{cases} \quad (51)$$

Thus, by employing a centered basis, we have a direct access to the different diffuse derivatives that could be used within an optimization algorithm for computing the gradient vector and the Hessian matrix.

### 5.3. Numerical example

In order to illustrate the above presented procedure, we consider the geometry shown in Fig. 5 and the following material and process parameters:  $v = 0.26$ ,  $\rho = 1560$ ,  $C = 1700$  and  $k = 3.7 \cdot 10^{-7}$ , all these values expressed in the metric system.

First, proceeding as indicated in the previous sections, we compute the solution for all possible boundary conditions:

$$u(x, y, \theta_1, \theta_2, \theta_3) \approx \sum_{i=1}^{i=N} F_i(x, y) \cdot \Theta_{1i}(\theta_1) \cdot \Theta_{2i}(\theta_2) \cdot \Theta_{3i}(\theta_3) \quad (52)$$

This parametric solution can then be particularized for any choice of the temperatures of the heaters. One of the possible solutions related to a particular choice of the heaters temperatures is depicted in Fig. 6.

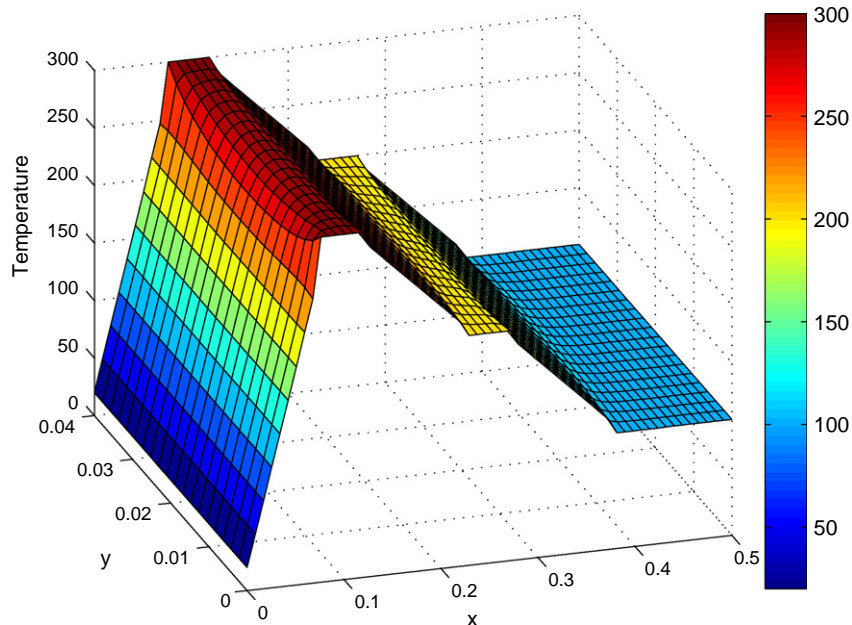
If we compare the solution represented in Fig. 6 to the one obtained by the finite element method for the same values of the heaters temperatures, the difference (still using the  $L_2$ -norm) is found to be lower than  $10^{-5}$  for  $N$  as little as 30 modes.

Now, we are considering the optimization process with respect to the cost function given by Eq. (29), where the target temperature at the outflow is fixed to  $\bar{u} = 150^\circ\text{C}$ .

The temperature distribution in the whole die for these optimized values is shown in Fig. 7, for the optimal temperatures of the three heaters:

$$\begin{aligned} \theta_1^{\text{opt}} &= 211.9^\circ\text{C} \\ \theta_2^{\text{opt}} &= 89.3^\circ\text{C} \\ \theta_3^{\text{opt}} &= 150.2^\circ\text{C} \end{aligned} \quad (53)$$

The value of the associated cost function when the three optimal temperatures were considered was of  $3.1 \times 10^{-7}$ .



**Fig. 6.** Multidimensional solution particularized for the heaters temperatures  $\theta_1 = 300^\circ\text{C}$ ,  $\theta_2 = 200^\circ\text{C}$  and  $\theta_3 = 100^\circ\text{C}$ .

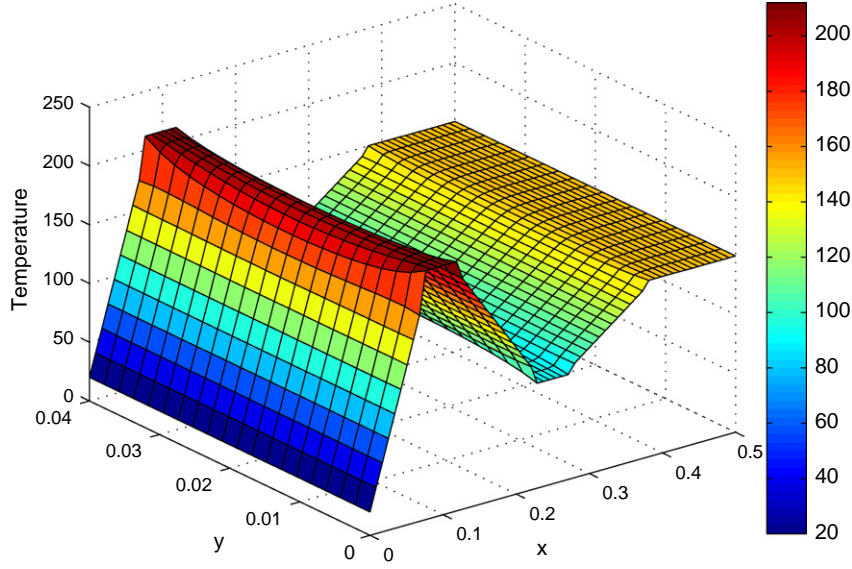


Fig. 7. Final, optimized distribution of temperatures.

**Remark 3.** Note that the values taken in this academic example do not correspond to any physically meaningful ones. The obtained temperatures could have an impact on the resin viscosity and therefore in its thermal conductivity. We are here mostly interested in introducing the optimization strategy.

The technique here proposed could allow impressive computing efficiencies, however it is too early for quantifying the reachable computing time savings. The present technique only needs one solution of a multi-dimensional model, that thanks to the PGD features can be performed easily and very fast. As soon as this parametric solution is known we can imagine a panoply of optimization problems, involving different cost functions and numerical strategies, performed on-line without the necessity of solving the model again. The optimization described above was carried out by using Matlab and a standard laptop in 0.0015 s.

## 6. Conclusion

In this paper we have presented the abilities of the Proper Generalized Decomposition for solving a multi-dimensional problem in which parameters involved in optimization procedures are introduced as extra-coordinates in the model. Using the PGD the solution complexity scales linearly with the dimension of the space. The method seems to be able to deal with optimization problems whose deterministic solutions were a dream until now.

Advanced modeling of thermal processes taking place in a heated die, as encountered in pultrusion processes, is achieved by introducing the heating devices temperatures as extra-coordinates. This fact allows by solving once the resulting multi-dimensional model, to have access to the solution for any choice of the process parameters. For circumventing the resulting curse of dimensionality a PGD based discretization was applied, whose computational complexity scales linearly with the dimension of the space, instead of the exponential growing characteristic of mesh-based discretization strategies.

The potentialities of such procedure seem extraordinary, but at present it is too early to conclude on its advantages and its limitations. The coupling of parametric off-line solutions with most optimization strategies could be suitable because we only need to particularize the pre-computed parametric solution for any choice

of the process parameters. Thus, optimization can be performed on-line and by using very light computing platforms, as for example smartphones.

## Acknowledgement

This work has been partially supported by the Spanish Ministry of Science and Innovation, through Grant No. CICYT-DPI2011-27778-C02-01.

## Appendix A. Alternating directions separated representation constructor

### A.1. Computing $R(\mathbf{x})$ from $S(t)$ and $W(k)$

We consider the global weak form of Eq. (2):

$$\int_{\Omega \times I \times \mathfrak{S}} u^* \left( \frac{\partial u}{\partial t} - k \Delta u - f \right) dx dt dk = 0 \quad (54)$$

where the trial and test functions write respectively:

$$u(\mathbf{x}, t, k) = \sum_{i=1}^{i=n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) + R(\mathbf{x}) \cdot S(t) \cdot W(k) \quad (55)$$

and, assuming  $S$  and  $W$  known from the previous iteration,

$$u^*(\mathbf{x}, t, k) = R^*(\mathbf{x}) \cdot S(t) \cdot W(k) \quad (56)$$

Introducing (55) and (56) into (54) it results:

$$\begin{aligned} \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \cdot \left( R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) dx dt dk \\ = - \int_{\Omega \times I \times \mathfrak{S}} R^* \cdot S \cdot W \cdot \mathcal{R}^n dx dt dk \end{aligned} \quad (57)$$

where  $\mathcal{R}^n$  defines the residual at iteration  $n$  that reads:

$$\mathcal{R}^n = \sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \quad (58)$$

Once all functions involving time and conductivity have been determined, we can integrate Eq. (57) along its respective domains  $I \times \mathfrak{S}$ , and by taking into account the following notations:

$$\left[ \begin{array}{lll} w_1 = \int_{\mathfrak{I}} W^2 dk & s_1 = \int_I S^2 dt & r_1 = \int_{\Omega} R^2 d\mathbf{x} \\ w_2 = \int_{\mathfrak{I}} kW^2 dk & s_2 = \int_I S \cdot \frac{dS}{dt} dt & r_2 = \int_{\Omega} R \cdot \Delta R d\mathbf{x} \\ w_3 = \int_{\mathfrak{I}} W dk & s_3 = \int_I S dt & r_3 = \int_{\Omega} R d\mathbf{x} \\ w_4^i = \int_{\mathfrak{I}} W \cdot K_i dk & s_4^i = \int_I S \cdot \frac{dT_i}{dt} dt & r_4^i = \int_{\Omega} R \cdot \Delta X_i d\mathbf{x} \\ w_5^i = \int_{\mathfrak{I}} kW \cdot K_i dk & s_5^i = \int_I S \cdot T_i dt & r_5^i = \int_{\Omega} R \cdot X_i d\mathbf{x} \end{array} \right], \quad (59)$$

Eq. (57) is reduced to:

$$\begin{aligned} & \int_{\Omega} R^* \cdot (w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R) d\mathbf{x} \\ & = - \int_{\Omega} R^* \cdot \left( \sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) d\mathbf{x} \end{aligned} \quad (60)$$

Eq. (60) defines an elliptic steady-state boundary value problem that can be solved by using any discretization technique operating on the weak form of the problem (finite elements, finite volumes, ...). Another possibility consists in coming back to the strong form of Eq. (60):

$$\begin{aligned} & w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R \\ & = - \sum_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \end{aligned} \quad (61)$$

that could be solved by using any classical collocation technique (finite differences, SPH, ...).

#### A.2. Computing $S(t)$ from $R(\mathbf{x})$ and $W(k)$ :

In the present case the test function is written as:

$$u^*(\mathbf{x}, t, k) = S^*(t) \cdot R(\mathbf{x}) \cdot W(k) \quad (62)$$

Now, the weak form becomes:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{I}} S^* \cdot R \cdot W \cdot \left( R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) d\mathbf{x} dt dk \\ & = - \int_{\Omega \times I \times \mathfrak{I}} S^* \cdot R \cdot W \cdot \mathcal{R}^n d\mathbf{x} dt dk \end{aligned} \quad (63)$$

Integrating in the space  $\Omega \times \mathfrak{I}$  and by taking into account the notation (59) gives:

$$\begin{aligned} & \int_I S^* \cdot \left( w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S \right) dt \\ & = - \int_I S^* \cdot \left( \sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) dt \end{aligned} \quad (64)$$

Eq. (64) represents the weak form of the ODE defining the time evolution of the field  $S$  that can be solved by using any stabilized discretization technique (SU, Discontinuous Galerkin, ...). The strong form of Eq. (64) reads:

$$\begin{aligned} & w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S \\ & = - \sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \end{aligned} \quad (65)$$

Eq. (65) can be solved by using backward finite differences, or higher order Runge–Kutta schemes, among many other possibilities.

Computing  $W(k)$  from  $R(\mathbf{x})$  and  $S(t)$ :

In this part of the algorithm, the test function is written as:

$$u^*(\mathbf{x}, t, k) = W^*(k) \cdot R(\mathbf{x}) \cdot S(t) \quad (66)$$

Now, the weak form becomes:

$$\begin{aligned} & \int_{\Omega \times I \times \mathfrak{I}} W^* \cdot R \cdot S \cdot \left( R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) d\mathbf{x} dt dk \\ & = - \int_{\Omega \times I \times \mathfrak{I}} W^* \cdot R \cdot S \cdot \mathcal{R}^n d\mathbf{x} dt dk \end{aligned} \quad (67)$$

Integrating the weak form, Eq. (67), in  $\Omega \times I$  and considering the notations given by Eq. (59) leads to:

$$\begin{aligned} & \int_{\mathfrak{I}} W^* \cdot (r_1 \cdot s_2 \cdot W - r_2 \cdot s_1 \cdot k \cdot W) dk \\ & = - \int_{\mathfrak{I}} W^* \cdot \left( \sum_{i=1}^{i=n} r_5^i \cdot s_4^i \cdot K_i - \sum_{i=1}^{i=n} r_4^i \cdot s_5^i \cdot k \cdot K_i - r_3 \cdot s_3 \cdot f \right) dk \end{aligned} \quad (68)$$

Eq. (68) does not involve any differential operator. The strong form of Eq. (68) is:

$$(r_1 \cdot s_2 - r_2 \cdot s_1 \cdot k) \cdot W = - \sum_{i=1}^{i=n} (r_5^i \cdot s_4^i - r_4^i \cdot s_5^i \cdot k) \cdot K_i - r_3 \cdot s_3 \cdot f \quad (69)$$

Therefore, Eq. (69) represents an algebraic equation. Despite the introduction of parameters as additional model coordinates, the change of computational cost will not be noticed. In fact, the original equation does not contain derivatives with respect to those parameters, so obviously solving this equation will not affect the total computation time in a dramatic way.

Note that other strategies for constructing the separated representation are available in the literature, leading to sometimes more robust and faster convergence [6,14].

## References

- [1] Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *J Non Newton Fluid Mech* 2006;139:153–76.
- [2] Ammar A, Mokdad B, Chinesta F, Keunings R. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. Part II: transient simulation using space-time separated representations. *J Non Newton Fluid Mech* 2007;144:98–121.
- [3] Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Comput Meth Appl Mech Eng* 1998;139:3–47.
- [4] Bungartz H-J, Griebel M. Sparse grids. *Acta Numer* 2004;13:1–123.
- [5] Chen X, Xie H, Chen H. Optimisation for CFRP pultrusion process based on genetic algorithm-neutral network. *Int J Mater Form* 2010;3:927–30.
- [6] Chinesta F, Ammar A, Cueto E. Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models. *Arch Comput Method Eng* 2009;17:327–50.
- [7] Chinesta F, Ammar A, Cueto E. Recent advances in the use of the proper generalized decomposition for solving multidimensional models. *Arch Comput Method Eng*, in press.
- [8] Chinesta F, Ammar A, Cueto E. Proper generalized decomposition of multiscale models. *Int J Numer Meth Eng* 2010;83:1114–32.
- [9] Gonzalez D, Ammar A, Chinesta F, Cueto E. Recent advances on the use of separated representations. *Int J Numer Meth Eng* 2010;85(5):637–59.
- [10] Jacob A. Globalisation of the pultrusion industry. *Reinf Plast* 2006;50(5):38–41.
- [11] Ladeveze P. *Nonlinear computational structural mechanics*. NY: Springer; 1999.
- [12] Martin J. Pultruded composites compete with traditional construction materials. *Reinf Plast* 2006;50(5):20–7.
- [13] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput Mech* 1992;10:307–18.
- [14] Nouy A. A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations. *Comput Meth Appl Mech Eng* 2010;199(23–24):1603–26.
- [15] Pruliere E, Ferec J, Chinesta F, Ammar A. An efficient reduced simulation of residual stresses in composites forming processes. *Int J Mater Form* 2010;3(2):1339–50.