A NEW REDUCED APPROACH TO SIMULATE CURING IN COMPOSITE FORMING PROCESSES

N. Bur[1], S. Aghighi[1], P. Joyot*[1], F. Chinesta[2], P. Villon[3]

[1]*ESTIA, F-64210 Bidart, France*
[2]*ESI International Chair, GEM CNRS-ECN, 1 rue de la Noë BP 92101, 44321 Nantes cedex 3, France*
[3]*UTC-Roberval UMR 6253, Centre de recherches de Royallieu, CS 60319, 60203 Compiègne cedex, France*
* *Corresponding Author: p.joyot@estia.fr*

Keywords: Composite materials, Curing, Reduced-Order Modelling, Proper Generalized Decomposition, Non-linearity.

Abstract
We propose an algorithm to solve the curing equation by writing the fields under their separated form. This approach enables having a non-incremental method in time, less expensive than classical approaches. We illustrate this new process with an example based on Bailleul's model.

1. Introduction

An efficient simulation of thermosetting curing process must be able to accurately predict the temperature and the degree of reticulation through the entire process. This allows to ensure that the curing is homogeneous and avoid some defaults in the finished parts. From a numerical point of view, there are several difficulties related to the simulation of composites processing. A first one is related to the growing size and complexity of composite parts. The number of time steps needed in the time discretization increases drastically due to the differences of the characteristic times of heat diffusion and chemical kinetics. Many other difficulties should be addressed, being the most important the multi-physics coupling and the strong non linearity of the resulting model. In order to reduce the computation time related to numerical simulations, a new family of solvers, called the Proper Generalized Decomposition method, or in short the PGD method, has been developed. It is an *a priori* resolution technique, that derives from the EDP of the problem. The solution is progressed by resolving some spatial problems and temporal problems (ordinary differential equation). E. Prulière [1] uses the PGD approach to solve this problem. This technique may be improved, particularly at the level of the non-linear part of the equations. Indeed, non-linear terms break down the separable nature of the fields, and therefore restrict the global efficiency of the algorithm applied on problems over large domains. We propose a new approach in order to treat the non-linearity, that doesn't break down the separable nature of the fields. In this way, "industrial" cases, that generally involve huge domains, can be addressed.

In this paper we only illustrate the approach on the curing equation. Firstly we propose a method for solving this equation taking into account the separated form of the variables. In a second

step we propose a PGD-based algorithm to deal with non linear terms.

2. Solving curing equation on space and time

2.1. Curing equations

Among the models describing the evolution of the degree of curing, denoted α, the most used is the one proposed by Kamal and Sourour [2]. The present work deals with the Bailleul's model [3] that can be written in a simple polynomial form:

$$\frac{\partial}{\partial t}\alpha\left(x,t\right) = K\left(u\left(x,t\right)\right)G\left(\alpha\left(x,t\right)\right), \tag{1}$$

$$\text{with} \begin{cases} K\left(u\right) = K_{ref}e^{-A\left(\frac{T_{ref}}{u-273}-1\right)} & (2) \\ G\left(\alpha\right) = \sum_{i}^{n_g} g_i\alpha^i & (3) \end{cases}$$

2.2. Separated Picard method

We propose an original scheme to solve the curing equation (1). The starting point is based on the Picard's method:

$$\alpha_{n+1}\left(x,t\right) = \alpha_0\left(x\right) + \int_{t_0}^{t} K\left(u\left(x,s\right)\right)G\left(\alpha_n\left(x,s\right)\right)\ ds, \tag{4}$$

where the indices denote the Picard's iterations. In order to ssume the convergence of the fixed point algorithm, $t - t_0$ has to be small enough.

Now we take into account the separated representation of the variables. That is to say we suppose it is possible and easy to get a separated form of $K\left(u\right)G\left(\alpha_n\right)$. We note

$$H_n = K\left(u\right)G\left(\alpha_n\right) \tag{5}$$

we can write, separating space and time, as

$$H_n\left(x,t\right) = \sum_{i}^{n_h} H_{n,i}^{x}\left(x\right)H_{n,i}^{t}\left(t\right) \tag{6}$$

Replacing H_n by (6) within (4) leads to

$$\alpha_{n+1}\left(x,t\right) = \alpha_0\left(x\right) + \sum_{i}^{n_h} H_{n,i}^{x}\left(x\right)\int_{t_0}^{t} H_{n,i}^{t}\left(s\right)\ ds \tag{7}$$

Computing $\int_{t_0}^{t} H_{n,\alpha}^{t}\left(s\right)\ ds$ is similar to solve the ODE

$$\frac{\partial}{\partial t}h_{n,i}\left(t\right) = H_{n,i}^{t}\left(t\right), \tag{8}$$

with homogeneous initial condition. At each iteration of the Picard's algorithm, we have to separate H_n and solve the n_h ODEs.

This algorithm avoid solving Eq.(1) at each point of the space domain, what is done in the classical method. In our opinion, it appears much more efficient and can be extended to the multiparametric case.

Let us now consider how to separate efficiently H_n.

3. Separation of the non linear term $H(x,t)$

For the sake of clarity, we note H instead of H_n.

H is the product of a polynomial $G(\alpha(x,t))$ by an Arrhenius function $K(u(x,t))$. We are going to study how to separate both functions, since it is simple to compute the product of two separated functions.

3.1. Separation of polynomial expression

We guess the separated form of $\alpha^q(x,t)$ is known, i.e

$$p_q(x,t) = \alpha^q(x,t) = \sum_{i=1}^{n_{p,q}} p^x_{q,i}(x)\, p^t_{q,i}(t)\,, \tag{9}$$

where p_q is introduced to avoid misunderstanding between powers and superscripts.

We wish to calculate the separated form of $\alpha^{q+1}(x,t)$, i.e

$$p_{q+1}(x,t) = \alpha^{q+1}(x,t) = \sum_{i=1}^{n_{p,q+1}} p^x_{q+1,i}(x)\, p^t_{q+1,i}(t)\,. \tag{10}$$

Moreover we have

$$p_{q+1}(x,t) = p_q(x,t)\,\alpha(x,t) \tag{11}$$

The PGD framework applied to (11) writes

$$\begin{aligned}\int_\Omega p^x_{q+1}(x)\, {p^x_{q+1}}^\star(x)\; dx \int_\tau p^t_{q+1}(t)\, p^t_{q+1}(t)\; dt = \\ \sum_{j=1}^{n_\alpha}\sum_{i=1}^{n_{p,q}} \int_\Omega p^x_{q,i}(x)\,\alpha^x_j(x)\, {p^x_{q+1}}^\star(x)\; dx \int_\tau p^t_{q,i}(t)\,\alpha^t_j(t)\, p^t_{q+1}(t)\; dt \\ - \sum_{i=1}^{n-1} \int_\Omega p^x_{q+1,i}(x)\, {p^x_{q+1}}^\star(x)\; dx \int_\tau p^t_{q+1,i}(t)\, p^t_{q+1}(t)\; dt \quad \forall {p^x_{q+1}}^\star\end{aligned} \tag{12}$$

$$\int_\Omega p^x_{q+1}(x)\, p^x_{q+1}(x)\ dx \int_\tau p^t_{q+1}(t)\, {p^t_{q+1}}^\star(t)\ dt = \sum_{j=1}^{n_\alpha}\sum_{i=1}^{n_{p,q}} \int_\Omega p^x_{q,i}(x)\,\alpha^x_j(x)\, p^x_{q+1}(x)\ dx \int_\tau p^t_{q,i}(t)\,\alpha^t_j(t)\, {p^t_{q+1}}^\star(t)\ dt - \sum_{i=1}^{n-1}\int_\Omega p^x_{q+1,i}(x)\, p^x_{q+1}(x)\ dx \int_\tau p^t_{q+1,i}(t)\, {p^t_{q+1}}^\star(t)\ dt \quad \forall {p^t_{q+1}}^\star \tag{13}$$

We get easily the separation of $p_i(x,t)$. We repeat this process in order to obtain the separated representation of functions from $p_2(x,t)$ to $p_{n_g}(x,t)$.

The separated form of $G(\alpha)$ comes from the separation of $p_i(x,t)$ and Eq.(3) using the following PGD algorithm

$$\begin{cases} \int_\Omega G^x(x)\, G^{x\star}(x)\ dx \int_\tau G^t(t)\, G^t(t)\ dt = \sum_{i=1}^{n_g}\sum_{j=1}^{n_{p,i}} g_i \int_\Omega p^x_{i,j}(x)\, G^{x\star}(x)\ dx \int_\tau p^t_{i,j}(x)\, G^t(t)\ dt \\ \qquad - \sum_{i=1}^{n-1}\int_\Omega G^x_i(x)\, G^{x\star}(x)\ dx \int_\tau G^t_i(t)\, G^t(t)\ dt \quad \forall G^{x\star} \end{cases} \tag{14}$$

$$\begin{cases} \int_\Omega G^x(x)\, G^x(x)\ dx \int_\tau G^t(t)\, G^{t\star}(t)\ dt = \sum_{i=1}^{n_g}\sum_{j=1}^{n_{p,i}} g_i \int_\Omega p^x_{i,j}(x)\, G^x(x)\ dx \int_\tau p^t_{i,j}(x)\, G^{t\star}\ dt \\ \qquad - \sum_{i=1}^{n-1}\int_\Omega G^x_i(x)\, G^x(x)\ dx \int_\tau G^t_i(t)\, G^{t\star}\ dt \quad \forall G^{t\star} \end{cases} \tag{15}$$

3.2. Separation of Arrhenius type equation

3.2.1. General case

Let us consider

$$v(x,t) = f(u(x,t)) \tag{16}$$

assuming known the separation $u(x,t) = \sum_\alpha^{n_u} u^x(x)\, u^t(x)$. We look for the separation of v under the form

$$v(x,t) = \sum_\alpha^{n_v} v^x(x)\, v^t(t)\,. \tag{17}$$

One solution is to rebuild the field u on the space-time domain, then apply the f function to u and run an SVD to get v. This approach is used in [1]. In this work we propose a method avoiding this rebuilding of u and the separation of $f(u)$.
We call this approach *separation by differentiation*.

We write

$$\frac{df(u)}{du} = wf(u) \tag{18}$$

Our approach will be efficient if w is simple to separate along space and time; we see this within the next section. According to (18) we have

$$v_{,t}(x,t) = u_{,t}(x,t)\, w(x,t)\, v(x,t)\,, \tag{19}$$

$$\Delta v\left(x,t\right) = \left(u_{,i}\left(x,t\right) w\left(x,t\right) v\left(x,t\right)\right)_{,i}, \tag{20}$$

or, with a parameter μ

$$v_{,t}\left(x,t\right) - \mu\Delta v\left(x,t\right) = u_{,t}\left(x,t\right) w\left(x,t\right) v\left(x,t\right) - \mu\left(u_{,i}\left(x,t\right) w\left(x,t\right) v\left(x,t\right)\right)_{,i} \tag{21}$$

To this last equation we add the boundary condition $v\left(x,t\right) = f\left(u\left(x,t\right)\right)$ for $x \in \Gamma$, where Γ is the boundary of the spatial domain, and the initial condition $v\left(x,t=0\right) = f\left(u\left(x,t=0\right)\right)$.

The v function solution of this differential equation verifies Eq. (16). We can apply the PGD method on this equation to get the separated form of v.

Since we use (18), in this method function f has to be C_1.

3.2.2. Arrhenius' function

In this section we consider f as an Arrhenius' function, i.e.

$$v\left(x,t\right) = f\left(u\right) = \exp\left(-\frac{a}{u\left(x,t\right)+u_0}\right) \tag{22}$$

Therefore we have

$$\frac{\partial f\left(u\right)}{\partial u} = \frac{a}{\left(u+u_0\right)^2} f\left(u\right) \tag{23}$$

and consequently

$$w = \frac{a}{\left(u+u_0\right)^2} \tag{24}$$

We use the PGD method to find the separation of w under the form $w\left(x,t\right) = \sum_i^{n_u} w_i^x\left(x\right) w_i^t\left(x\right)$. We have

$$\left\{\begin{array}{l} \displaystyle\sum_{i=1}^{n_h} \int_\Omega h^x\left(x\right) w^x\left(x\right) w^{x\star}\left(x\right)\ dx \int_\tau h^t\left(x\right) w^t\left(t\right) w^t\left(t\right)\ dt = a\int_\Omega w^{x\star}\left(x\right)\ dx \int_\tau w^t\left(t\right)\ dt \\ \qquad\qquad - \displaystyle\sum_{i=1}^{n-1}\sum_{j=1}^{n_h} \int_\Omega h_j^{x2}\left(x\right) w_i^x\left(x\right) w^{x\star}\left(x\right)\ dx \int_\tau h_j^{t\,2}\left(x\right) w_i^t\left(t\right) w^t\left(t\right)\ dt \quad \forall w^{x\star} \end{array}\right. \tag{25}$$

$$\left\{\begin{array}{l} \displaystyle\sum_{i=1}^{n_h} \int_\Omega h^x\left(x\right) w^x\left(x\right) w^x\left(x\right)\ dx \int_\tau h^t\left(x\right) w^t\left(t\right) w^{t\star}\left(t\right)\ dt = a\int_\Omega w^{x\star}\left(x\right)\ dx \int_\tau w^{t\star}\left(t\right)\ dt \\ \qquad\qquad - \displaystyle\sum_{i=1}^{n-1}\sum_{j=1}^{n_h} \int_\Omega h_j^x\left(x\right) w_i^x\left(x\right) w^x\left(x\right)\ dx \int_\tau h_j^t\left(x\right) w_i^t\left(t\right) w^{t\star}\left(t\right)\ dt \quad \forall w^{t\star}, \end{array}\right. \tag{26}$$

with $h\left(x,t\right) = \sum_i^{n_h} h_i^x\left(x\right) h_i^t\left(x\right)$ the separated form of $\left(u+u_0\right)^2$ that can be obtained easily as explained in 3.1.

3.3. Numerical validation

We consider a 1D spatial domain $\Omega = [0, x_{max}]$ and a time interval $\tau = [0, t_{max}]$, with $x_{max} = 1\,\text{mm}$ and $t_{max} = 2000\,\text{s}$. The temperature field is known as

$$u(x,t) = \frac{210 - 160}{x_{max}} x + 160. \tag{27}$$

The coefficients in the Bailleul's equation are $T_{ref} = 493.13\,\text{K}$, $K_{ref} = 0.001\,56\,\text{s}^{-1}$ and $A = 15.44$ for Eq.(2), and $g_1 = 0.44$, $g_2 = 10.03$, $g_3 = -80.35$, $g_4 = 364.46$, $g_5 = -1041.60$, $g_6 = 1831.43$, $g_7 = -1907.14$, $g_8 = 1076.45$, $g_9 = -253.73$ for Eq.(3)

In a first step we validate the numerical scheme proposed in 2.2.
Figure 1 shows the solution α of Eq.(1), and Fig. 2 plots the error at each iteration n. This error is computed as

$$e = \frac{\|\alpha - \alpha_{ref}\|_2}{\|\alpha_{ref}\|}, \tag{28}$$

where α_{ref} is the reference solution obtained from Eq.(1), on each point x of Ω.

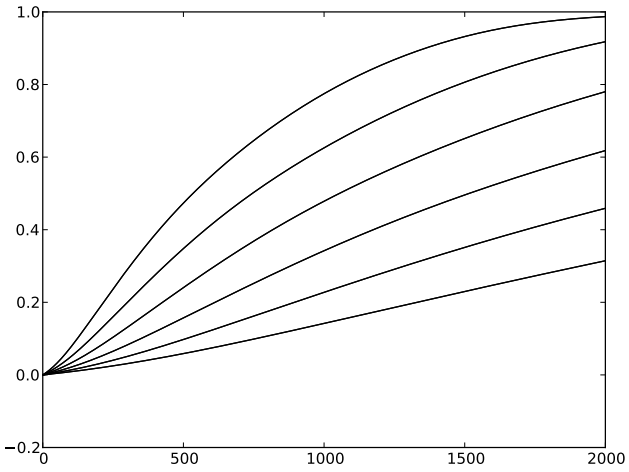


Figure 1. Field $\alpha(x,t)$

To confirm $H(x,t)$ separation, we assume $u(x,t)$ given by (27) and $\alpha(x,t)$ is the solution of the previous problem at time step t_{max}. The separation of $G(\alpha(x,t))$ is represented in Fig. 3(a). Similarly, $K(u(x,t))$ is shown in Fig. 3(b).

4. Conclusion

We proposed an algorithm to solve the curing equation, adapted to variables written under separated form. This algorithm requires to compute the separated form of a non linear function at each iteration. We gave a simple solution in case this function is polynomial. We also proposed a new solution for the Arrhenius' function. This approach, named *separation by differentiation* is generic and will be tried in the near future on other type of non linearities.

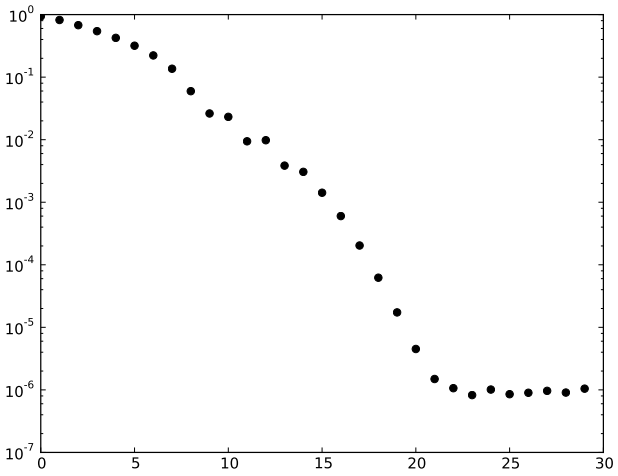
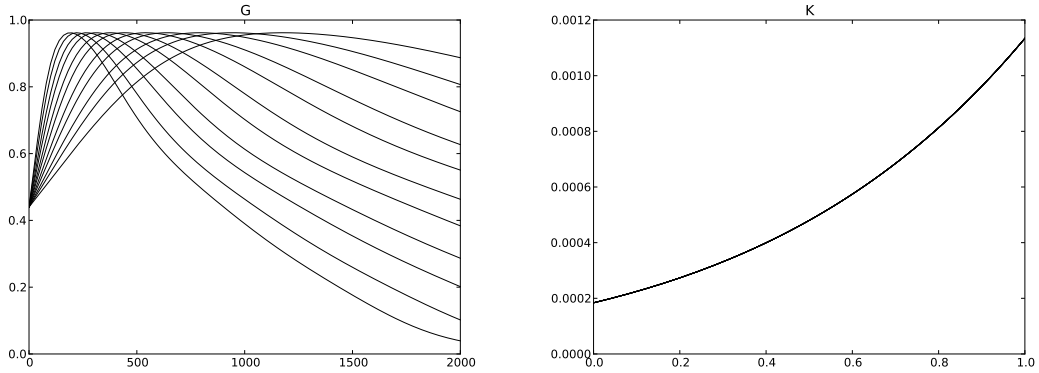


Figure 2. Error e evolution over iterations



(a) $G(\alpha(x,t))$ (b) $K(u(x,t))$

Figure 3. Modes representation

References

[1] Etienne Prulière, Julien Férec, Francisco Chinesta, and Amine Ammar. An efficient reduced simulation of residual stresses in composite forming processes. *International Journal of Material Forming*, 3(S2):1339–1350, January 2010.

[2] M. R. Kamal and S. Sourour. Kinetics and thermal characterization of thermoset cure. *Polymer Engineering and Science*, 13(1):59–64, January 1973.

[3] J. L. Bailleul, D. Delaunay, and Y. Jarny. Determination of Temperature Variable Properties of Composite Materials: Methodology and Experimental Results. *Journal of Reinforced Plastics and Composites*, 15(5):479–496, May 1996.