Towards a Framework for Non-Linear Thermal Models in Shell Domains

Francisco Chinesta, Adrien Leygue, Marianne Béringhier, Linh Tuan Nguyen, Jean-Claude Grandidier, Bernhard Schrefler, Francisco Pasavento

Towards a framework for non-linear thermal models in shell domains

Francisco Chinesta and Adrien Leygue

GeM, Ecole Centrale Nantes, Nantes, France

Marianne Beringhier, Linh Tuan Nguyen and Jean-Claude Grandidier

Institut P', ENSMA, Poitiers, France, and

Bernhard Schrefler and Francisco Pesavento

Department of Civil, Environmental and Architectural Engineering, University of Padova, Padova, Italy

Abstract

Purpose – The purpose of this paper is to solve non-linear parametric thermal models defined in degenerated geometries, such as plate and shell geometries.

Design/methodology/approach – The work presented in this paper is based in a combination of the proper generalized decomposition (PGD) that proceeds to a separated representation of the involved fields and advanced non-linear solvers. A particular emphasis is put on the asymptotic numerical method.

Findings – The authors demonstrate that this approach is valid for computing the solution of challenging thermal models and parametric models.

Originality/value – This is the first time that PGD is combined with advanced non-linear solvers in the context of non-linear transient parametric thermal models.

Keywords Numerical methods, Mathematical modelling, Model reduction, Proper generalized decomposition, Thermal model, Shell geometries, Asymptotic numerical method

Paper type Research paper

1. Introduction

Today many problems in science and engineering remain intractable, in spite of the impressive progresses attained in mechanical modelling, numerical analysis, discretization techniques and computer science during the last decade, because their numerical complexity is simply unimaginable.

We can distinguish different challenging scenarios for efficient numerical simulations, all them needing urgently new proposals:

- The first one concerns models defined in high-dimensional spaces, usually encountered in quantum chemistry and kinetic theory descriptions of complex fluids. Models defined in high-dimensional spaces suffer the so-called curse of dimensionality, because the number of the degrees of freedom involved in a mesh beased discretization increases exponentially with the model dimension.

- A second category of problems involves multi-scale problems not necessarily defined in high-dimensional spaces, but whose spectrum of characteristic times

or lengths is so wide that standard incremental discretization techniques cannot be applied. In such time-multi-scale problems, for instance, the time step is extremely small as a consequence of numerical stability requirements. Thus, simulations over the much larger time interval of interest, which typically requires the solution of a large linear algebraic system at each time step, simply become impossible. Multiscale models involving a wide range of characteristic times abound in many fields. Reaction-diffusion models of the degradation of plastic materials, for example, describe chemical reactions occurring within microseconds coupled to diffusion of chemical substances taking place over years. In processes involving microwaves, ultrasounds [...] or materials exhibiting different relaxation times, the difficulty related to time integration is crucial. The same scenario is found in solid mechanics where the constitutive equations are strongly non-linear and coupled, involving many scales and different characteristic times.

- Other challenging problems are defined in degenerated geometrical domains. By this we mean that at least one of the characteristic dimensions of the domain is smaller by several orders of magnitude than the others. This is the case of bar, plate or shell-like domains typical of structures or materials processing applications. In simple situations, such problems are readily transformed into reduced, 1D or 2D approximate theories (e.g. the classical elastic plate theory). When geometrical or material non-linearities are present, however, it is usually impossible to derive lower-dimensional models of sufficient validity. Standard mesh-based discretization methods then quickly become impractical, in view of the compulsory discretization of the small length scales that yield extremely fine meshes.

- Many problems in process control, parametric modelling, inverse identification, and process or shape optimization, usually require, when approached with standard techniques, the direct computation of a very large number of solutions of the concerned model for particular values of the problem parameters. Consider for example the optimization of a process where optimal parameter values must be determined for process operating conditions (e.g. speed, position and temperature of heaters) and material properties (e.g. thermal and rheological properties of the materials). Clearly, it would be useful to be able to simulate this process at once for all values of these parameters within a prescribed interval, and then perform data mining within this rather general solution to identify optimal values.

- Traditionally, simulation-based engineering sciences (SBES) relied on the use of static data inputs to perform the simulations. These data could be parameters of the model(s) or boundary conditions. The word static is intended to mean here that these data could not be modified during the simulation. A new paradigm in the field of applied sciences and engineering has emerged in the last decade. Dynamic data-driven application systems (DDDAS) constitute nowadays one of the most challenging applications of SBES. By DDDAS we mean a set of techniques that allow the linkage of simulation tools with measurement devices for real-time control of simulations. DDDAS entails the ability to dynamically incorporate additional data into an executing application, and in reverse, the ability of an application to dynamically steer the measurement process.

In this context, real-time simulators are needed in many applications. One of the most challenging situations is that of haptic surgery simulators, where forces acting on the surgical tool must be translated to the peripheral device at a rate of $500\,\text{Hz}$. Control, malfunctioning identification and reconfiguration of malfunctioning systems also need to run in real time. All these problems can be seen as typical examples of DDDAS.

- Augmented reality is another area in which efficient (fast and accurate) simulation is urgently needed. The idea is supplying in real-time appropriate information to the reality perceived by the user. Augmented reality could be an excellent tool in many branches of science and engineering.

- Light computing platforms are appealing alternatives to heavy computing platforms that in general are expensive and whose use requires technical knowledge. One can imagine that the off-line parametric solution of many models will make possible the on-line manipulation of those general solutions by using very light computing platforms, as for example smartphones or tablets.

- All the previous scenarios becomes more complex when the models that they involve are non-linear, as is it usually the case in science and engineering.

While the previous list is by no means exhaustive, it includes a set of problems with apparent no relationship among them that can be, however, treated in an unified manner. Their common ingredient is our lack of capabilities (or knowledge) to solve them numerically in a direct, traditional way. In order to obtain a solution, some kind of model order reduction is thus compulsory.

Many years ago, in the 1980s, Pierre Ladeveze proposed a separated representation of the space and time coordinates:

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

for performing efficient solutions of complex non-linear thermo-mechanical models. The radial approximation (1) was one of the main blocks of the powerful non-incremental and non-linear LArge Time INcrement (LATIN) solver. The corpus of literature devoted to this technique is vast, as proved in the book by Ladeveze (1999) on the topic, but remained in the form of space-time separations for many years.

A more general separated representation (in this case in the conformation space of complex fluids descriptions) was more recently employed by Ammar *et al.* (2006, 2007) for approximating the solution of multi-dimensional partial differential equations. Nouy (2009) considered also such separated representations for solving stochastic equations where the deterministic coordinates and the stochastic ones were separated, very much like in the radial, space-time approximation. Proper generalized decomposition (PGD) is the common name recently coined for techniques using such separated representations. The nature of the problem, the involved coordinates and the constructor of such approximations can be very different.

The general form of the separated representation involved in the PGD reads:

$$u(x_1, \ldots, x_D) \approx \sum_{i=1}^{i=N} F_i^1(x_1) \ldots F_i^D(x_D) \tag{2}$$

where x_i denote a scalar or vector coordinate defined in a domain Ω_i of moderate dimension $\Omega_i \subset \mathbb{R}^d$, with $d \leq 3$, in general.

The PGD approximation is thus a sum of N functional products involving each a number D of functions $F_i^j(x_j)$ that are unknown a priori. It is constructed by successive enrichment, whereby each functional product is determined in sequence. At a particular enrichment step $n + 1$, the functions $F_i^j(x_j)$ are known for $i \leq n$ from the previous steps, and one must compute the new product involving the D unknown functions $F_{n+1}^j(x_j)$. This is achieved by invoking the weak form of the problem under consideration. The resulting discrete system is non-linear, which implies that iterations are needed at each enrichment step. A low-dimensional problem can thus be defined in Ω_j for each of the D functions $F_{n+1}^j(x_j)$.

If M nodes are used to discretize each coordinate, the total number of PGD unknowns is $N \times M \times D$ instead of the M^D degrees of freedom involved in standard mesh-based discretizations. Moreover, all numerical experiments carried out to date with the PGD show that the number of terms N required to obtain an accurate solution is not a function of the problem dimension D, but it rather depends on the regularity of the searched solution and the separated representation constructor. The PGD thus avoids the exponential complexity with respect to the problem dimension.

This paper focuses on the solution of non-linear parametric models defined in shell geometries. Previous works focussed on:

- The solution of non-linear models. Complex thermo-mechanical models were efficiently solved within the LATIN framework (Ladeveze *et al.*, 2009, 2010; Neron and Ladeveze, 2010). More standard linearization strategies were also analyzed in Ammar *et al.* (2010) and Pruliere *et al.* (2010a) and applied for solving non-linear thermo-mechanical models in Pruliere *et al.* (2010b).

- The solution of parametric models. They were addressed in our former works within the PGD framework by including those parameters as extra-coordinates (Pruliere *et al.*, 2010a; Lamari *et al.*, 2010; Ghnatios *et al.*, 2011, 2012). See for a complete and recent review (Chinesta *et al.*, 2011) and the references therein.

- When these models are moreover defined in degenerated domains (plates or shells) additional difficulties arise related to the necessity of fully 3D solutions in domains involving different characteristic lengths. Efficient separation of the physical space was considered in Chinesta *et al.* (2008) that were generalized in Bognet *et al.* (2012) for treating complex plate geometries.

Despite these noticeable progresses, the efficient solution of non-linear parametric models remains today an open problem where different approaches are being explored. An interesting parametric LATIN approach has been recently proposed by Neron and coworkers (Heyberger *et al.*, 2012). Approaches based intensively developed by Maday and coworkers and Patera and coworkers, on the use of reduced bases (Prud'homme *et al.*, 2002) seem an appealing framework for addressing these challenging models.

This paper explores the possibility of combining separated representations in the physical and parametric spaces with asymptotic expansions for addressing coupled non-linear models in shell domains.

The paper is organized as follows. In the next section the PGD is applied for building-up the solution of the parametric heat transfer problem, which allows to

define a framework circumventing the issues related to incremental and parametric models as discussed in Section 3. Section 4 proposes a framework for addressing non-linearities and non-linear couplings by making use of asymptotic developments. Sections 5 integrates all the ingredients introduced in the previous sections for solving some numerical examples.

2. PGD of a generic linear parametric model

In this section, we illustrate the PGD on a simple parametric heat transfer problem. More details on the recent developments and applications of the method can be found in Chinesta *et al.* (2011), while additional details on its convergence and accuracy can be found in Ammar *et al.* (2010). Let us consider the following parametric heat transfer equation:

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0. \tag{3}$$

with homogeneous initial and boundary conditions. The enforcement of non-homogeneous initial and boundary conditions was deeply treated in Gonzalez *et al.* (2010) and Chinesta *et al.* (2010a).

Here $(\mathbf{x}, t, k) \in \Omega \times I \times \Im$, and the source term f is assumed constant. The conductivity k is viewed as a new coordinate defined in the interval $\Im$. Thus, instead of solving the thermal model for different discrete values of the conductivity parameter, we wish to solve at once a more general problem, the price to pay being an increase of the problem dimensionality. However, as the complexity of the PGD scales only linearly (and not exponentially) with the space dimension, consideration of the conductivity as a new coordinate still allows one to efficiently obtain an accurate solution.

The weak form related to equation (3) reads:

$$\int_{\Omega \times I \times \Im} u^* \cdot \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) d\mathbf{x} \ dt \ dk = 0, \tag{4}$$

for all test functions u^* selected in an appropriate functional space.

The PGD solution is sought in the form:

$$u(\mathbf{x}, t, k) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k). \tag{5}$$

At enrichment step n of the PGD algorithm, the following approximation is already known:

$$u^n(\mathbf{x}, t, k) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k). \tag{6}$$

We wish to compute the next functional product $X_{n+1}(\mathbf{x}) \cdot T_{n+1}(t) \cdot K_{n+1}(k)$, which we write as $R(\mathbf{x}) \cdot S(t) \cdot W(k)$ for notational simplicity.

Thus, the solution at enrichment step $n + 1$ reads:

$$u^{n+1} = u^n + R(\mathbf{x}) \cdot S(t) \cdot W(k). \tag{7}$$

We propose the simplest choice for the test functions u^* used in equation (4):

$$u^* = R^*(\mathbf{x}) \cdot S(t) \cdot W(k) + R(\mathbf{x}) \cdot S^*(t) \cdot W(k) + R(\mathbf{x}) \cdot S(t) \cdot W^*(k). \qquad (8)$$

With the trial and test functions given by equations (7) and (8), respectively, equation (4) is a non-linear problem that must be solved by means of a suitable iterative scheme. In our earlier papers (Ammar $et\,al.$, 2006, 2007), we used Newton's method. Simpler linearization strategies can also be applied, however. 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 $S(t)$ and $W(k)$ known from the previous iteration and compute an update for $R(\mathbf{x})$ (in this case the test function reduces to $R^*(\mathbf{x}) \cdot S(t) \cdot W(k)$). From the just-updated $R(\mathbf{x})$ and the previously-used $W(k)$, we can update $S(t)$ (with $u^* = R(\mathbf{x}) \cdot S^*(t) \cdot W(k)$). Finally, from the just computed $R(\mathbf{x})$ and $S(t)$, we update $W(k)$ (with $u^* = R(\mathbf{x}) \cdot S(t) \cdot W^*(k)$). This iterative procedure continues until reaching convergence. The converged functions $R(\mathbf{x})$, $S(t)$ and $W(k)$ yield the new functional product of the current enrichment step: $X_{n+1}(\mathbf{x}) = R(\mathbf{x})$, $T_{n+1}(t) = S(t)$ and $K_{n+1}(k) = W(k)$. Other procedures based on the residual minimization were considered for solving non-symmetric problems (Chinesta $et\,al.$, 2010b).

We have seen that at each enrichment step the construction of the new functional product in equation (5) requires non-linear iterations. If m_i denotes the number of iterations needed at enrichment step i, 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 m 3D problems for the functions $X_i(\mathbf{x})$, m 1D problems for the functions $T_i(t)$ and m algebraic systems for the functions $K_i(k)$. In general, m 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 (5) is of some tens of 3D steady-state problems (the cost related to the 1D and algebraic problems being negligible with respect to the 3D problems). In a classical approach, one must solve for each particular value of the parameter k a 3D problem at each time step. In usual applications, this often implies the computation of several millions of 3D solutions. Clearly, the CPU time savings by applying the PGD can be of several orders of magnitude.

3. Pushing forward simulation limits with the PGD

Before considering in some detail a number of applications of the PGD, let us briefly revisit some of the issues discussed in the introductory section:

(1) Efficient solvers for transient problems can be defined by applying a space-time separation:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t). \qquad (9)$$

The constructor of that separated representation was illustrated in the previous section (it suffices to ignore the existence of the parametric extra-coordinate). We cited previously the pioneering works of Ladeveze's team in the field

of structural mechanics. Space-time separated representations were also considered in the context of the multi-scale coupling of diffusion and kinetic models endowed with very different characteristic times (Chinesta *et al.*, 2010a) as well as in the context of time multi-scale models (Ammar *et al.*, 2012).

(2) The fully 3D solution of models defined in degenerate domains is also an appealing field of application of the PGD. Consider the unknown field $u(\mathbf{x})$ defined in a domain Ξ. Two approaches come to mind:

- Complete decomposition:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \cdot Z_i(z). \tag{10}$$

This strategy is particularly suitable for separable domains, i.e. $\Xi = \Omega_x \times \Omega_y \times \Omega_z$ (Chinesta *et al.*, 2008; Lamari *et al.*, 2010). For general domains, embedding Ξ into a larger separable domain $\Omega_x \times \Omega_y \times \Omega_z$ can also be done, as described in Gonzalez *et al.* (2010).

- Plate-type or profile-type decomposition:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} X_i(x, y) \cdot Z_i(z). \tag{11}$$

This strategy is particularly suitable when $\Xi = \Omega \times \mathscr{I}$, with $\Omega \subset \mathcal{R}^2$ and $\mathscr{I} \subset \mathcal{R}$. In plate-type domains $\mathscr{I}$ denotes the plate thickness dimension whereas in extruded profiles it denotes the extrusion direction being Ω the transversal section.

(3) Finally, for applications requiring many solutions of a particular model, it suffices to introduce all sources of variability as extra-coordinates. The solution of the resulting parametric multi-dimensional model is then sought in the separated form:

$$u(\mathbf{x}, t, p_1, \dots p_Q) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t) \cdot P_i^1(p_1) \dots P_i^Q(p_Q), \tag{12}$$

where the p_i's denote the different problem parameters such as material parameters, boundary conditions, applied loads, initial conditions, and geometrical parameters (see Chinesta *et al.* (2010b, 2011) and the references therein).

Let us illustrate the above discussion by considering the simple example of a heat transfer problem in an extruded shell domain. Even in the case of a simple physics, some problems are simply intractable due to the extremely large amount of degrees of freedom that they involve. When the physical domain is such that it is possible to separate some space coordinates from each other, the PGD offers tremendous computational gains. Figure 1 shows a tunnel-like domain on which we solve the following transient heat transfer problem:

$$\frac{\partial u}{\partial t} - k\Delta u = 0, \tag{13}$$

with homogeneous initial conditions, homogeneous boundary conditions on the exterior surfaces and a non-zero heat flux on the interior surface of the tunnel. The problem geometry is taken from the previous work of Schrefler *et al.* (2011) in which they studied the coupling of the fluid flow related to a fire inside the tunnel with the structural response of the concrete. The specific heat flux on the interior surface of the tunnel is modeled from the solution of the coupled solution presented in Schrefler *et al.* (2011) and corresponds to a one hour 20 MW fire. Solving the 3D transient thermal problem, even linear and uncoupled, in such shell geometry with a classical finite elements method is a challenge since the problem requires a very fine discretization along the tunnel thickness.
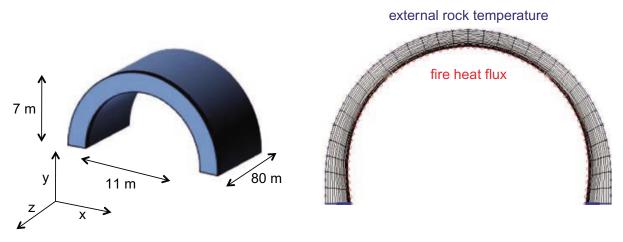
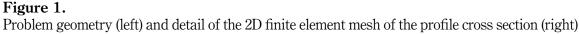We overcome this problem by seeking a PGD solution in the following separated form:

$$u(\mathbf{x}, z, t) \approx \sum_{i=1}^{N} X_i(x, y) \cdot Z_i(z) \cdot T_i(t), \tag{14}$$

with $(\mathbf{x}, z, t) \in \Omega \times \mathscr{I} \times I$ and $\Omega \subset \mathscr{R}^2$ and $\mathscr{I} \subset \mathscr{R}, I \subset \mathscr{R}$. The symbol $\mathscr{I}$ denotes the tunnel length, Ω the tunnel cross section while I is the time interval. In the (x, y) plane we use a biquadratic finite element mesh with 640 elements (2,665 nodes), while the z-direction is discretized using 800 linear elements. The equivalent 3D finite element mesh would therefore comprise over two millions nodes.

As explained before, the solution of this 4D problem will require the solution of several 2D and 1D elliptic boundary value problems (BVP) in Ω and $\mathscr{I}$, respectively, as well as the solution of several ODEs in I. The 2D and 1D BVPs are solved using standard (bi-)quadratic finite elements and the ODEs are solved using an implicit Euler scheme. Due to the non-symmetric nature of the problem because the first order time derivative, a variant of the PGD algorithm known as residual minimization (Chinesta *et al.*, 2010b) has been used in this case.

The PGD converges after computing only 14 terms of the finite sum decomposition of $u(\mathbf{x}, z, t)$. The required CPU time is about three minutes on a simple laptop. Figure 2 shows the reconstructed temperature field at a certain time instant. The equivalent
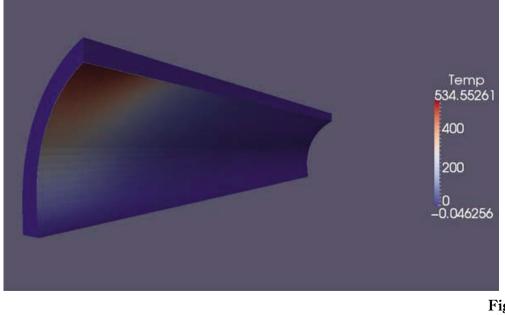

Figure 1.
Problem geometry (left) and detail of the 2D finite element mesh of the profile cross section (right)

Figure 2.
Reconstructed
temperature field

solution with classical finite elements would have required the solution of a problem with around two million nodes and 3,600 time steps (using an implicit time integration strategy).

We can conclude that the solution of conservation equations involving discretizations of the whole physical domain does not introduce major difficulties even in the case of degenerated domains, in which at least one of the characteristic dimensions in much lower than the other ones. The main issue in solving such models concerns the treatment of strong non-linearities and multi-physic couplings. In the sections that follow we are exploring the possibility of treating such non-linearities within the asymptotic numerical method (ANM) framework.

4. Accounting for non-linearities

The treatment of non-linear models by using some standard linearization strategies was addressed in some of our former works (Ammar *et al.*, 2010; Pruliere *et al.*, 2010a). We start by considering such procedures in order to point out their inherent limitations before considering an alternative approach based on the use of the ANM.

We consider the same model that we considered in our former works:

$$
\begin{cases}
\frac{\partial u}{\partial t} - k\Delta u = u^2 + f(\mathbf{x}, t) & \text{in } \Omega \times (0, T_{max}] \\
\\
u(\mathbf{x}, t) = 0 & \text{on } \partial\Omega \times (0, T_{max}] \\
\\
u(\mathbf{x}, 0) = 0 & \text{in } \Omega
\end{cases}
\tag{15}
$$

where $\Omega \subset \mathbb{R}^d, d \geq 1, T_{max} > 0$ and $k > 0$ is the diffusion coefficient. To build-up the approximated solution of equation (15) by using a separated representation, we considered two approaches: (i) an incremental linearization and (ii) a Newton linearization, both described in sections below.

4.1 Incremental linearization

We write the solution of problem (15) in the separated form:

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

We suppose that at iteration n, with $n < N$, the n first modes (X_i, T_i), $i = 1, \ldots n$, are already known and that at present iteration we search the new enrichment functional product $R(\mathbf{x}) \cdot S(t)$ such that the updated approximation writes:

$$u^{n+1}(\mathbf{x}, t) = u^n(\mathbf{x}, t) + R(\mathbf{x}) \cdot S(t) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) + R(\mathbf{x}) \cdot S(t) \qquad (16)$$

The weak form of problem (15) writes:

$$\int_0^{T_{max}} \int_\Omega u^* \left(\frac{\partial u}{\partial t} - k\Delta u - u^2 - f(\mathbf{x}, t) \right) d\mathbf{x} \ dt = 0; \quad \forall u^* \qquad (17)$$

The alternating directions scheme proceed by calculating $R(\mathbf{x})$ from the temporal function $S(t)$ just computed, and then, updating $R(\mathbf{x})$ from the just computed $S(t)$ as we described in the previous section. The iteration procedure should continue until reaching convergence. Here, the novelty is the treatment of the non-linear term u^2. The simplest possibility consists in computing this term at the previous iteration, that is, assuming at the present iteration the following approximation of the non-linear term:

$$u^2 \approx \left(\sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 \qquad (18)$$

4.2 Newton linearization

From the solution at iteration n, u^n:

$$u^n(\mathbf{x}, t) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \qquad (19)$$

the solution at the next iteration can be written as $u^{n+1} = u^n + \tilde{u}$ where $\tilde{u}$ is the solution of the linearized problem:

$$\frac{\partial \tilde{u}}{\partial t} - k\Delta \tilde{u} - 2u^n \tilde{u} = -\mathscr{R}(u^n) \qquad (20)$$

where the residual $\mathcal{R}(u^n)$ reads:

$$\mathcal{R}(u^n) \equiv \left(\frac{\partial u^n}{\partial t} - k\Delta u^n - (u^n)^2 - f(\mathbf{x}, t) \right) \qquad (21)$$

The associated weak formulation writes:

$$\int_0^{T_{max}} \int_\Omega \tilde{u}^* \left(\frac{\partial \tilde{u}}{\partial t} - a\Delta\tilde{u} - 2u^n\tilde{u} \right) d\mathbf{x} \; dt$$

$$= \int_0^{T_{max}} \int_\Omega \tilde{u}^* \left(-\frac{\partial u^n}{\partial t} + k\Delta u^n + (u^n)^2 + f(\mathbf{x}, t)) \right) d\mathbf{x} \; dt, \quad \forall \tilde{u}^* \tag{22}$$

Now, we assume $u(\mathbf{x}, t) = R(\mathbf{x}) \cdot S(t)$ and $\tilde{u}^* = S \cdot R^* + R \cdot S^*$. To compute both functions $R(\mathbf{x})$ and $S(t)$ we apply again the alternating directions method deeply described in the previous sections.

4.3 Discussion

Both procedures converge but no significant differences in the number of required iterations were noticed. The convergence rate and the computing time were similar.

In the case of linear models and symmetric differential operators, if we solve the problem and then apply the POD (for a given precision) we obtain an optimal separated representation:

$$u^{POD,N}(\mathbf{x}, t) = \sum_{i=1}^N X_i^{POD}(\mathbf{x}) \cdot T_i^{POD}(t) \tag{23}$$

When the numerical solution is computed using the PGD, the solution obtained using N sums $u^{PGD,N}$ is very close to $u^{POD,N}$. However, in the case of non-linear models the situation is radically different. Even when the exact solution can be represented by a single functional product, i.e.:

$$u^{ex}(\mathbf{x}, t) = X^{ex}(\mathbf{x}) \cdot T^{ex}(t) \tag{24}$$

the non-linear solver produces a solution composed of many sums:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^N X_i(\mathbf{x}) \cdot T_i(t) \tag{25}$$

with $N > 1$. The main reason is that the number of sums is in this case subsidiary of the convergence rate of the non-linear solver.

In Pruliere *et al.* (2010a) we analyzed other linearization schemes. When we considered the improved fixed point, in which the non-linear term is approximated at iteration q of the enrichment step n, according to:

$$u^2 \approx (u^n + R^{(q-1)}(\mathbf{x}) \cdot S^{(q-1)}(t))^2 \tag{26}$$

then we proved, in the case described above whose exact solution consists of a single product, that the solver converges after computing the first functional couple. In that sense the solver is optimal but the computing time is similar to the one required by using the standard fixed point or the Newton strategy previously described.

The main difficulty related to the use of standard linearizations lies in the necessity of evaluating the non-linear term. Because the necessity of arriving to a separated representation of such term, one possibility consists in performing a polynomial expansion and then compute the different powers of the solution at the previous step.

However, it is easy to understand that the power p of u^n involves too many terms when p or n increase as can be noticed from:

$$(u^n)^p = \left(\sum_{i=1}^{i=n} F_i^1(x_1) \times \cdots \times F_i^D(x_D) \right)^p \tag{27}$$

The issue related to the high powers could be alleviated by introducing new variables v^i such that $v^{(2)} = u \cdot u$, $v^{(3)} = u^3 = v^{(2)} \cdot u$, ... that allows never compute more than powers of degree 2.

However, in general the differential operators related to the searched functional product involve the more and more terms. Thus, when D, p or N increases the solution procedure described in the previous sections become rapidly inefficient. The next section explores the use of an alternative approach, the ANM that allows guarantee a constant number of terms to represent the differential operator.

4.4 The ANM

In this section we propose an alternative approach, within the ANM framework (Cochelin *et al.*, 1994a, b), able to include the parametric dimensions as extra-coordinates without modifying the differential operator representation.

We consider again the model defined by:

$$\frac{\partial u}{\partial t} - k\Delta u = u^2 + f(\mathbf{x}, t) \tag{28}$$

The non-linear term can be affected by a "loading" parameter λ:

$$\frac{\partial u}{\partial t} - k\Delta u = \lambda \cdot u^2 + f(\mathbf{x}, t) \tag{29}$$

We denote by u_0 the solution related to $\lambda = \lambda_0 = 0$ that can be computed easily because it corresponds to the solution of the linear problem. The searched solution is the one related to $\lambda = 1$.

Now, we define an asymptotic expansion of the unknown field u as well as of the loading parameter λ by considering powers of the expansion parameter a:

$$\begin{cases} u = u_0 + a \cdot u_1 + a^2 \cdot u_2 + \cdots \\ \lambda = \lambda_0 + a \cdot \lambda_1 + a^2 \cdot \lambda_2 + \cdots \end{cases} \tag{30}$$

The non-linear term can be written as:

$$u^2 = (u^2)_0 + a \cdot (u^2)_1 + a^2 \cdot (u^2)_2 + \cdots + a^p \cdot (u^2)_p + \cdots \tag{31}$$

where $(u^2)_p$ reads:

$$(u^2)_p = \sum_{i=0}^{i=p} u_i \cdot u_{p-i} = 2 \cdot u_0 \cdot u_p + \sum_{i=1}^{i=p-1} u_i \cdot u_{p-i} \tag{32}$$

Introducing equations (30) and (32) into equation (29) and identifying the different powers of a, it results:

- Power $p = 0$. Solution u_0 is assumed known because it corresponds to the solution of the linear problem resulting from $\lambda = \lambda_0 = 0$.
- Power $p = 1$. We should solve:

$$\frac{\partial u_1}{\partial t} - k\Delta u_1 = \lambda_0 \cdot (u^2)_1 + \lambda_1 \cdot (u^2)_0 \tag{33}$$

and because $\lambda_0 = 0$, the evolution equation for u_1 reduces to:

$$\frac{\partial u_1}{\partial t} - k\Delta u_1 = \lambda_1 \cdot (u^2)_0 \tag{34}$$

We can notice that the previous equation contains two unknowns, u_1 and λ_1. The solution procedure needs an extra-equation. The following equation is considered (Cochelin *et al.*, 1994b):

$$(u - u_0, u_1) + (\lambda - \lambda_0, \lambda_1) = a \tag{35}$$

where $(,)$ denotes the scalar product either in L^2 or in $\mathbb{R}$ depending on its arguments. The first order of equation (35) reads:

$$(u_1, u_1) + (\lambda_1)^2 = 1 \tag{36}$$

Because the linearity of equation (34) if $\hat{u}_1$ is the solution of:

$$\frac{\partial \hat{u}_1}{\partial t} - k\Delta \hat{u}_1 = (u^2)_0 \tag{37}$$

the searched solution u_1 results:

$$u_1 = \lambda_1 \cdot \hat{u}_1 \tag{38}$$

By introducing equation (38) into equation (35) it results:

$$(\lambda_1)^2 \cdot [(\hat{u}_1, \hat{u}_1) + 1] = 1 \tag{39}$$

that allows computing the value of λ_1 and then u_1 from equation (38).
- Power p, with $p > 1$. We should solve:

$$\frac{\partial u_p}{\partial t} - k\Delta u_p = \lambda_p \cdot (u^2)_0 + \sum_{i=1}^{i=p-1} \lambda_i \cdot (u^2)_{p-i} \tag{40}$$

where the fact that $\lambda_0 = 0$ was taken into account.

In this expression we can notice that the differential operator is always the same and that in the right hand member the term $\sum_{i=1}^{i=p-1} \lambda_i \cdot (u^2)_{p-i}$ has been already computed.

Equation (35) at order p reads:

$$(u_p, u_1) + (\lambda_p, \lambda_1) = 0 \tag{41}$$

Because the linearity of equation (40) if $\tilde{u}_p$ is the solution of:

$$\frac{\partial \tilde{u}_p}{\partial t} - k\Delta\tilde{u}_p = \sum_{i=1}^{i=p-1} \lambda_i \cdot (u^2)_{p-i} \tag{42}$$

and $\hat{u}_p$ the solution of:

$$\frac{\partial \tilde{u}_p}{\partial t} - k\Delta\hat{u}_p = (u^2)_0 \tag{43}$$

that as we can notice corresponds to $\hat{u}_1$, the searched solution u_p reads:

$$u_p = \tilde{u}_p + \lambda_p \cdot \hat{u}_1 \tag{44}$$

By introducing equation (44) into equation (41) it results:

$$(\tilde{u}_p, \hat{u}_1) + \lambda_p \cdot [(\hat{u}_1, \hat{u}_1) + 1] = 0 \tag{45}$$

that allows computing the value of λ_p and then u_p from equation (44).

5. Numerical results

In this section, we illustrate the previous developments on a few selected examples. For the sake of simplicity, we restrict ourselves to the solution of transient and steady-state heat transfer problems with a quadratic non-linear term, similarly to the previous sections. In particular, we show how the PGD can be used as a very efficient PDE solver that can be integrated into the ANM allowing the computation of multi-dimensional, non-incremental, parametric, non-linear solutions.

5.1 PGD-based ANM

Since we have seen that the PGD offers a powerful alternative to classical finite elements when one can take advantage of variable separation, we next investigate the use of the PGD as an alternative of finite elements in the solution of a non-linear heat transfer problem within the ANM framework. For this purpose we consider:

$$-k\Delta u - f + u^2 = 0. \tag{46}$$

Here $u(x, y) \in \Omega_x \times \Omega_y$, with $\Omega_x = \Omega_y = [0, 2]$, the source term being constant, $f = 1$ and the conductivity k is also constant and equal to 1. We consider the following non-homogeneous Dirichlet conditions along the boundary Γ of Ω: $u(x, y) = 2$ for $(x, y) \in \Gamma$. For this choice of boundary conditions, it is obvious that the non-linear term will greatly influence the solution. Indeed, accounting or not for the non-linearity could change the concavity of the solution.

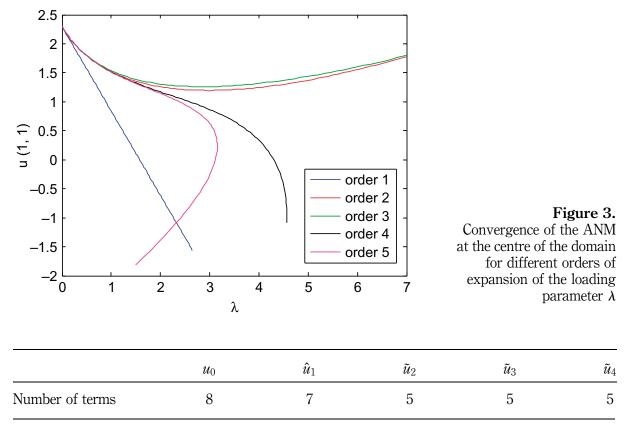The PGD solution is sought in the following separated form:

$$u(x, y) \approx \sum_{i=1}^{N} X_i(x) \cdot Y_i(y), \tag{47}$$

where we use a Galerkin finite elements method with continuous piecewise linear interpolation to solve the 1D BVP problems that appear in the PGD algorithm. In order to apply the ANM, we introduce the loading parameter a and modify the original problem in the following way:

$$-k\Delta u - f + \lambda u^2 = 0. \tag{48}$$

Now, the procedure described in Section 4.4 is applied. In Figure 3 we show the value solution in the centre of the domain $u(1, 1)$ for different values of the loading parameter λ and for different orders of expansion. The same problem solved with 2D finite elements on an equivalent mesh instead of the PGD predicts curves that exactly superimpose on the curves shown in Figure 3. We see that when the loading parameter is zero, the solution is greater than 2 while it becomes less than 2 for increasing values of λ. Note that the solution of the original non-linear problem corresponds to $\lambda = 1$. Furthermore, we see that the expansion up to order 4 is sufficient for capturing the non-linearity of the problem. The number of terms in the separated representation of the different solutions computed in the construction of the asymptotic expansion is reported in Table I.

In this specific example the use of PGD-based ANM has not any advantage since the problem is very simple. There is no real benefit in seeking a separated representation of the solutions of the various linear problems that have to be solved to compute the solution at the different orders within the ANM framework. Indeed as all the linear problems appearing in the ANM at its different orders involve the same differential operator, the computational cost of using 2D finite elements is dominated by the initial construction and factorization of the matrix representing the discretized operator. The successive solutions are then inexpensive. With the PGD, one cannot take advantage of this property because the coefficient related to the differential operators are changing at each iteration. However, it is clear that for larger problems, such as the previous tunnel example the PGD would offer a powerful alternative to finite elements within the ANM context.

Figure 3.
Convergence of the ANM at the centre of the domain for different orders of expansion of the loading parameter λ

	u_0	$\hat{u}_1$	$\tilde{u}_2$	$\tilde{u}_3$	$\tilde{u}_4$
Number of terms	8	7	5	5	5

Table I.

5.2 Parametric non-linear solution

We now turn to the case of parametric models, for which the PGD is well suited. We solve the same non-linear problem but consider this time the diffusivity k as an additional coordinate:

$$-k\Delta u - 1 + u^2 = 0. \tag{49}$$

Here $u(x, y, k) \in \Omega_x \times \Omega_y \times \Im$, with $\Omega_x = \Omega_y = [0, 2]$ and $k \in \Im = [1, 5]$. We consider the same Dirichlet condition as previously and the same expansion for the loading parameter. Applying the ANM directly to such model is only possible due to the PGD ability to treat multi-dimensional parametric models. The convergence of the method is shown in Figure 4 where we obviously see a better convergence in the case of higher values for k for which the non-linear term is of lesser importance. The number of terms in the separated representation of the different solutions computed in the construction of the asymptotic expansion is reported in Table II.

5.3 Space-time PGD-based ANM

Finally we turn to the case of a transient non-linear problem which will be solved in a non-incremental way by combining both the ANM and the PGD. We consider the following 1D PDE which is a particular case of the case treated in the previous sections:
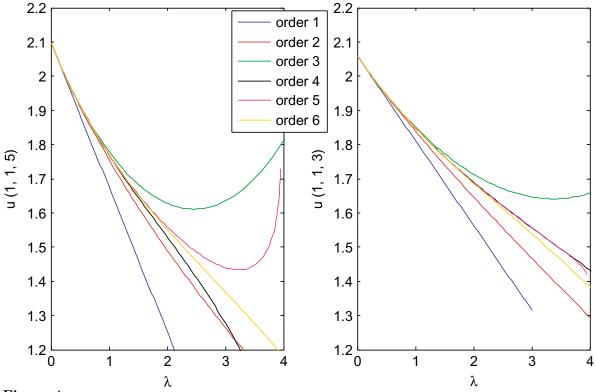

Figure 4.
Convergence of the ANM at the centre of the domain for different orders of expansion of the loading parameter λ and different values of k

	u_0	$\hat{u}_1$	$\tilde{u}_2$	$\tilde{u}_3$	$\tilde{u}_4$	$\tilde{u}_5$	$\tilde{u}_6$
Number of terms	6	7	5	4	4	5	5

Table II.

$$\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} u - 1 + u^2 = 0. \tag{50}$$

Here $u(x, t) \in \Omega_x \times \Im$, with $\Omega_x = [0, 2]$ and $t \in \Im = [0, 1]$, and we take the following boundary and initial conditions:
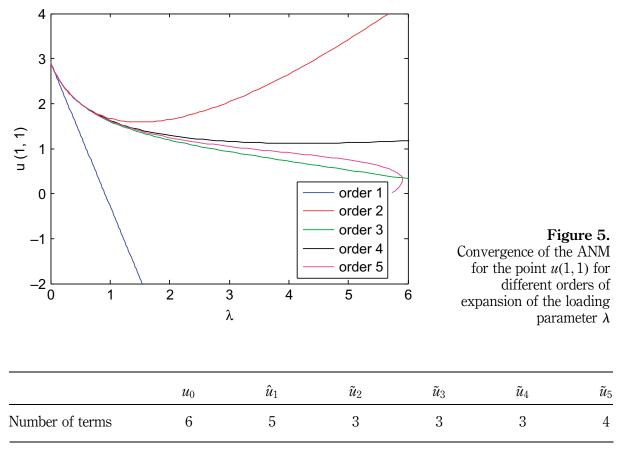
$$u(t, 0) = u(t, 2) = 2, \quad \forall t \in [0, 1],$$

$$u(x, 0) = 2, \quad \forall x \in [0, 2].$$

In Figure 5, we show the convergence of the ANM for the point $u(1, 1)$. We see that the space-time solution converges already for a moderate expansion order. The number of terms in the separated representation of the different solutions computed in the construction of the asymptotic expansion is reported in Table III.

6. Conclusions

By means of selected illustrations, we have demonstrated that the PGD-based ANM is particularly suitable for addressing coupled non-linear parametric models. It allows treating non-linear models with a unique representation of the differential operators involved at each order of the asymptotic development. Then, at each order, the resulting equation is solved by invoking the PGD method that allows separating the physical space for accounting for degenerated domains, including parameters as extra-coordinates for addressing non-linear parametric models and finally solving transient models using a non-incremental procedure.

Figure 5.
Convergence of the ANM for the point $u(1, 1)$ for different orders of expansion of the loading parameter λ

	u_0	$\hat{u}_1$	$\tilde{u}_2$	$\tilde{u}_3$	$\tilde{u}_4$	$\tilde{u}_5$
Number of terms	6	5	3	3	3	4

Table III.

Similarly to the LATIN method, we directly compute the non-linear space-time solution, but the approach proposed here could be readily extended to the case of parametric non-linear solutions.

Many questions remain open, but the first numerical experiments here reported allow to be confident on the potentialities of the proposed strategy.

References

Ammar, A., Chinesta, F., Cueto, E. and Doblare, M. (2012), "Proper generalized decomposition of time-multiscale models", *International Journal for Numerical Methods in Engineering*, Vol. 90 No. 5.

Ammar, A., Mokdad, B., Chinesta, F. and Keunings, R. (2006), "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*, Vol. 139, pp. 153-76.

Ammar, A., Mokdad, B., Chinesta, F. and Keunings, R. (2007), "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 representations", *Journal of Non-Newtonian Fluid Mechanics*, Vol. 144, pp. 98-121.

Ammar, A., Normandin, M., Daim, F., Gonzalez, D., Cueto, E. and Chinesta, F. (2010), "Non-incremental strategies based on separated representations: applications in computational rheology", *Communications in Mathematical Sciences*, Vol. 8 No. 3, pp. 671-95.

Bognet, B., Leygue, A., Chinesta, F., Poitou, A. and Bordeu, F. (2012), "Advanced simulation of models defined in plate geometries: 3D solutions with 2D computational complexity", *Computer Methods in Applied Mechanics and Engineering*, Vol. 201-204, pp. 1-12.

Chinesta, F., Ammar, A. and Cueto, E. (2010a), "Proper generalized decomposition of multiscale models", *International Journal for Numerical Methods in Engineering*, Vol. 83 Nos 8/9, pp. 1114-32.

Chinesta, F., Ammar, A. and Cueto, E. (2010b), "Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models", *Archives of Computational Methods in Engineering*, Vol. 17 No. 4, pp. 327-50.

Chinesta, F., Ladeveze, P. and Cueto, E. (2011), "A short review on model order reduction based on proper generalized decomposition", *Archives of Computational Methods in Engineering*, Vol. 18, pp. 395-404.

Chinesta, F., Ammar, A., Leygue, A. and Keunings, R. (2011), "An overview of the proper generalized decomposition with applications in computational rheology", *Journal of Non-Newtonian Fluid Mechanics*, Vol. 166, pp. 578-92.

Chinesta, F., Ammar, A., Lemarchand, F., Beauchene, P. and Boust, F. (2008), "Alleviating mesh constraints: model reduction, parallel time integration and high resolution homogenization", *Computer Methods in Applied Mechanics and Engineering*, Vol. 197 No. 5, pp. 400-13.

Cochelin, B., Damil, N. and Potier-Ferry, M. (1994a), "Asymptotic-numerical methods and pade approximants for non-linear elastic structures", *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 1187-213.

Cochelin, B., Damil, N. and Potier-Ferry, M. (1994b), "The asymptotic numerical method: an efficient perturbation technique for nonlinear structural mechanics", *Revue Europeenne des Elements Finis*, Vol. 3, pp. 281-97.

Ghnatios, C., Chinesta, F., Cueto, E., Leygue, A., Breitkopf, P. and Villon, P. (2011), "Methodological approach to efficient modeling and optimization of thermal processes taking place in a die: application to pultrusion", *Composites Part A*, Vol. 42, pp. 1169-78.

Ghnatios, C., Masson, F., Huerta, A., Cueto, E., Leygue, A. and Chinesta, F. (2012), "Proper generalized decomposition based dynamic data-driven control of thermal processes", *Computer Methods in Applied Mechanics and Engineering*, Vol. 213-216, pp. 29-41.

Gonzalez, D., Ammar, A., Chinesta, F. and Cueto, E. (2010), "Recent advances in the use of separated representations", *International Journal for Numerical Methods in Engineering*, Vol. 81 No. 5, pp. 637-59.

Heyberger, C., Boucard, P.-A. and Neron, D. (2012), "Multiparametric analysis within the proper generalized decomposition framework", *Computational Mechanics*, Vol. 49 No. 3, pp. 277-89.

Ladeveze, P. (1999), *Nonlinear Computational Structural Mechanics*, Springer, New York, NY.

Ladeveze, P., Neron, D. and Passieux, J.-C. (2009), "On multiscale computational mechanics with time-space homogenization", in Fish, J. (Ed.), *Multiscale Methods – Bridging the Scales in Science and Engineering*, Oxford University Press, Oxford, pp. 247-82, Chapter Space time scale bridging methods.

Ladeveze, P., Passieux, J.-C. and Neron, D. (2010), "The Latin multiscale computational method and the proper generalized decomposition", *Computer Methods in Applied Mechanics and Engineering*, Vol. 199 Nos 21/22, pp. 1287-96.

Lamari, H., Ammar, A., Cartraud, P., Legrain, G., Jacquemin, F. and Chinesta, F. (2010), "Routes for efficient computational homogenization of non-linear materials using the proper generalized decomposition", *Archives of Computational Methods in Engineering*, Vol. 17 No. 4, pp. 373-91.

Neron, D. and Ladeveze, P. (2010), "Proper generalized decomposition for multiscale and multiphysics problems", *Archives of Computational Methods in Engineering*, Vol. 17 No. 4, pp. 351-72.

Nouy, A. (2009), "Recent developments in spectral stochastic methods for the solution of stochastic partial differential equations", *Archives of Computational Methods in Engineering*, Vol. 16 No. 3, pp. 251-85.

Prud'homme, C., Rovas, D.V., Veroy, K., Machiels, L., Maday, Y., Patera, A.T. and Turinici, G. (2002), "Reliable real-time solution of parametrized partial differential equations: reduced-basis output bound methods", *Journal of Fluids Engineering*, Vol. 124, pp. 70-80.

Pruliere, E., Chinesta, F. and Ammar, A. (2010a), "On the deterministic solution of multidimensional parametric models by using the proper generalized decomposition", *Mathematics and Computers in Simulation*, Vol. 81, pp. 791-810.

Pruliere, E., Ferec, J., Chinesta, F. and Ammar, A. (2010b), "An efficient reduced simulation of residual stresses in composites forming processes", *International Journal of Material Forming*, Vol. 3 No. 2, pp. 1339-50.

Schrefler, B.A., Codina, R., Pesavento, F. and Principe, J. (2011), "Thermal coupling of fluid flow and structural response of a tunnel induced by fire", *International Journal of Numerical Methods in Engineering*, Vol. 87, pp. 361-85.