On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition

Etienne Pruliere, Francisco Chinesta, Amine Ammar

On the deterministic solution of multidimensional parametric models using the Proper Generalized Decomposition

E. Pruliere [1], F. Chinesta [2,3], A. Ammar [4]

[1] *Arts et Métiers ParisTech*
Esplanade des Arts et Métiers, 33405 Talence cedex, France

[2] *GeM: UMR CNRS – Centrale Nantes*
1 rue de la Noë, BP 92101, F-44321 Nantes cedex 3, France

[3] *EADS Corporate Foundation Internantional Chair – GeM – Centrale Nantes*

[4] *Laboratoire de Rhéologie: UMR CNRS – UJF – INPG*
1301 rue de la piscine, BP 53, F-38041 Grenoble, France

Abstract: This paper focuses on the efficient solution of models defined in high dimensional spaces. Those models involve numerous numerical challenges because of their associated curse of dimensionality. It is well known that in mesh-based discrete models the complexity (degrees of freedom) scales exponentially with the dimension of the space. Many models encountered in computational science and engineering involve numerous dimensions called configurational coordinates. Some examples are the models encountered in biology making use of the chemical master equation, quantum chemistry involving the solution of the Schrödinger or Dirac equations, kinetic theory descriptions of complex systems based on the solution of the so-called Fokker-Planck equation, stochastic models in which the random variables are included as new coordinates, financial mathematics, ... This paper revisits the curse of dimensionality and proposes an efficient strategy for circumventing such challenging issue. This strategy, based on the use of a Proper Generalized Decomposition, is specially well suited to treat the multidimensional parametric equations.

Keywords: Multidimensional models; Curse of dimensionality; Parametric models; Proper Generalized Decompositions; Separated representations

1 Introduction: revisiting the Proper Generalized Decomposition

The efficient solution of complex models involving an impressive number of degrees of freedom could be addressed by performing high performance computing (in general making use of parallel computing platforms) or by speeding up the calculation by using preconditioning, domain decomposition, ...

In the case of transient models the use of model reduction can alleviate significantly the solution procedure. The main ingredient of model reduction techniques based on the use of proper orthogonal decompositions -POD- consists of extracting a reduced number of functions able to represent the whole time evolution of the solution, that could be then used to make-up a reduced model. This extraction can be performed by invoking the POD. The reduced model can be then used for solving a similar model, i.e. a model slightly different to the one that served to extract the reduced approximation basis, or for solving the original model in a time interval larger than the one that served for constructing the reduced basis. The main issue in this procedure consists in the evaluation of the reduced basis quality when it applies in conditions beyond its natural interval of applicability. In order to ensure the solution accuracy one should proceed to enrich the reduced approximation basis, and the definition of optimal, or at least efficient, enrichment procedures is a difficult task that remains at present an open issue.

Such model reduction strategies were successfully applied, in some of our former works, to solve kinetic theory models [5] [23] allowing impressive computing-time savings. The main conclusion of our former works was the fact that an accurate description of a complex system evolution can, in general, be performed from the linear combination of a reduced number of space functions (defined in the whole space domain). The coefficients of that linear combination evolve in time. Thus, during the resolution of the evolution problem, an efficient algorithm has to compute the approximation coefficients and to enrich the approximation basis at the same time. An important drawback of one such approach is the fact that the approximation functions are defined in the space domain. Until now, the simplest form to represent one such function is to give its values in some points of the domain of interest, and to define its values in any other point by interpolation. However, sometimes models are defined in multidimensional spaces and in this case the possibility of describing functions from their values at the nodes of a mesh (or a grid in the domain of interest) can become prohibitory.

Many models encountered in computational science and engineering involve numerous dimensions called configurational coordinates. Some examples are the models encountered in biology making use of the chemical master equation, quantum chemistry involving the solution of the Schrödinger or Dirac equations, kinetic theory descriptions of complex materials and systems based on the solution of the so-called Fokker-Planck equation, stochastic models in which the random variables are included as new coordinates, financial mathematics model-

ing credit risk in credit markets (multi-dimensional Black and Scholes equation), ... The numerical solution of those models introduces some specific challenges related to the impressive number of degrees of freedom required because of the highly dimensional spaces in which those models are defined. Despite the fact that spectacular progresses have been accomplished in the context of computational mechanics in the last decades, the treatment of those models, as we describe in the present work, requires further developments.

The brut force approach cannot be considered as a possibility for treating this kind of models. Thus, in the context of quantum chemistry, the Nobel Prize laureate R.B. Laughlin, affirmed that no computer existing, or that will ever exist, can break the barriers found in the solution of the Schrödinger equation in multi-particle systems, because of the multidimensionality of this equation [15].

We can understand the catastrophe of dimension by assuming a model defined in a hyper-cube in a space of dimension D, $\Omega =]-L, L[^D$. In fact, if we define a grid to discretize the model, as it is usually performed in the vast majority of numerical methods (finite differences, finite elements, finite volumes, spectral methods, etc.), consisting of N nodes on each direction, the total number of nodes will be N^D. If we assume that for example $N \approx 10$ (an extremely coarse description) and $D \approx 80$ (much lower than the usual dimensions required in quantum or statistical mechanics), the number of nodes involved in the discrete model reaches the astronomical value of 10^{80} that represents the presumed number of elementary particles in the universe! Thus, progresses on this field need the proposal of new ideas and methods in the context of computational physics.

A first solution is the use of sparse grids methods [8], however as argued in [1], this strategy fails when it applies for the solutions of models defined in spaces whose dimension are about 20.

Another possible alternative for circumventing, or at least alleviating the curse of dimensionality issue, consists of using separated representations within the context of the so-called Proper Generalized Decomposition. We proposed recently a technique able to construct, in a completely transparent way for the user, the separated representation of the unknown field involved in a partial differential equation. This technique, originally described and applied to multi-bead-spring FENE models of polymeric systems in [3], was extended to transient models of such complex fluids in [4]. Other more complex models (involving different couplings and non-linearities) based on the reptation theory of polymeric liquids were analyzed in [17]. This technique was also applied in the fine description of the structure and mechanics of materials [11], including quantum chemistry [2], and in materials homogenization [12]. Some numerical results concerning this kind of approximation were addressed in [22] and [7].

Basically, the separated representation of a generic function $u(x_1, \cdots, x_D)$ (also known as finite sums decomposition) writes:

$$u(x_1, \cdots, x_D) \approx \sum_{i=1}^{i=N} F_1^i(x_1) \times \cdots \times F_D^i(x_D) \tag{1}$$

This kind of representation is not new, it was widely employed in the last decades in the framework of quantum chemistry. In particular, the Hartree-Fock (that involves a single product of functions) and the post-Hartree-Fock approaches (as the MCSCF that involves a finite number of sums) are based on a separated representation of the wavefunction [10]. Moreover, Pierre Ladeveze [14] proposed many years ago a space-time separated representation (that he called radial approximation) within the context of the non-incremental non-linear solver LATIN. This technique allowed impressive computing time savings in the simulation of usual 3D transient models because of its intrinsic non-incremental character.

Separated approximations are an appealing choice for addressing the solution of stochastic models [19]. Usually, stochastic equations are based on simple deterministic solvers in the context of Monte-Carlo methods [9] [20] or some equivalent methods [21] [6]. The main difficulty of this kind of methods is the need of a large amount of deterministic computations to approach the response of a given probability distribution. An alternative to these methods consists on the use of direct simulations describing explicitly the stochastic variables with a Galerkin approximation [13] [16]. Obviously, these strategies are restricted by the curse of dimensionality. The use of Proper Generalized Decomposition can extend the application field of deterministic solvers to models including many stochastic parameters, that will be introduced as new model coordinates. The interested reader can refer to the excellent review of A. Nouy [18].

In this present paper the PGD technique is revisited and then used to address some linear and non-linear parametric models.

2 Illustrating the solution of multidimensional parametric models by using the PGD

In what follows we are illustrating the construction of the Proper Generalized Decomposition by considering a quite simple problem, the parametric heat transfer equation:

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0 \tag{2}$$

where $(\mathbf{x}, t, k) \in \Omega \times I \times \Im$ and for the sake of simplicity the source term is assumed constant, i.e.$f = cte$. Because the conductivity is considered unknown, it is assumed as a new coordinate defined in the interval $\Im$. Thus, instead of solving the thermal model for different values of the conductivity parameter we prefer introducing it as a new coordinate. The price to be paid is the increase of the model dimensionality; however, as the complexity of PGD scales linearly with the space dimension the consideration of the conductivity as a new coordinate allows faster and cheaper solutions.

We look to the solution of Eq. (2) as:

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

For the following equation the approximation at iteration n is supposed known:

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

Thus, 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 required. The simplest choice consists in using an alternating directions fixed point algorithm. It proceeds 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 $W(k)$ we can update $S(t)$, and finally from the just computed $R(\mathbf{x})$ and $S(t)$ we compute $W(k)$. The procedure continues until reaching convergence. The converged functions $R(\mathbf{x})$, $S(t)$ and $W(k)$ allow defining the searched functions: $X_{n+1}(\mathbf{x}) = R(\mathbf{x})$, $T_{n+1}(t) = S(t)$ and $K_{n+1}(k) = W(k)$. These three steps can be detailed as follows.

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

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

$$\int\limits_{\Omega \times I \times \Im} u^* \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) \, d\mathbf{x} \, dt \, dk = 0 \tag{5}$$

where the trial and test functions write respectively:

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

and

$$u^*(\mathbf{x},t,k) = R^*(\mathbf{x}) \cdot S(t) \cdot W(k) \tag{7}$$

Introducing (6) and (7) into (5) it results

$$\begin{array}{l} \int\limits_{\Omega \times I \times \Im} R^* \cdot S \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \, d\mathbf{x} \, dt \, dk = \\ = - \int\limits_{\Omega \times I \times \Im} R^* \cdot S \cdot W \cdot \mathcal{R}^n \, d\mathbf{x} \, dt \, dk \end{array} \tag{8}$$

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

$$\mathcal{R}^n = \sum_{i=1}^{i=n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{i=n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f \tag{9}$$

Now, knowing all the functions involving time and parametric coordinate, we can integrate terms of Eq. (8) in their respective domains $I \times \Im$. By integrating over $I \times \Im$ and taking into account the notations:

$$\left[\begin{array}{lll} w_1 = \int\limits_{\Im} W^2 dk & s_1 = \int\limits_{I} S^2 dt & r_1 = \int\limits_{\Omega} R^2 d\mathbf{x} \\ w_2 = \int\limits_{\Im} kW^2 dk & s_2 = \int\limits_{I} S \cdot \frac{dS}{dt} dt & r_2 = \int\limits_{\Omega} R \cdot \Delta R \; d\mathbf{x} \\ w_3 = \int\limits_{\Im} W \; dk & s_3 = \int\limits_{I} S \; dt & r_3 = \int\limits_{\Omega} R \; d\mathbf{x} \\ w_4^i = \int\limits_{\Im} W \cdot K_i \; dk & s_4^i = \int\limits_{I} S \cdot \frac{dT_i}{dt} dt & r_4^i = \int\limits_{\Omega} R \cdot \Delta X_i \; d\mathbf{x} \\ w_5^i = \int\limits_{\Im} kW \cdot K_i \; dk & s_5^i = \int\limits_{I} S \cdot T_i \; dt & r_5^i = \int\limits_{\Omega} R \cdot X_i \; d\mathbf{x} \end{array}\right] \tag{10}$$

the Eq. (8) reduces to:

$$\begin{array}{l} \int\limits_{\Omega} R^* \cdot \left(w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R\right) \; d\mathbf{x} \; = \\ = - \int\limits_{\Omega} R^* \cdot \left(\sum\limits_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum\limits_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \; d\mathbf{x} \end{array} \tag{11}$$

Eq. (11) defines an elliptic steady state boundary value problem that can be solved by using any discretization technique operating on the model weak form (finite elements, finite volumes ...). Another possibility consists in coming back to the strong form of Eq. (11):

$$\begin{array}{c} w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R = \\ = - \left(\sum\limits_{i=1}^{i=n} w_4^i \cdot s_4^i \cdot X_i - \sum\limits_{i=1}^{i=n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \end{array} \tag{12}$$

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

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

In the present case the test function writes:

$$u^*(\mathbf{x}, t, k) = S^*(t) \cdot R(\mathbf{x}) \cdot W(k) \tag{13}$$

Now, the weak form writes:

$$\begin{array}{l} \int\limits_{\Omega \times I \times \Im} S^* \cdot R \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \; d\mathbf{x} \; dt \; dk = \\ = - \int\limits_{\Omega \times I \times \Im} S^* \cdot R \cdot W \cdot \mathcal{R}^n \; d\mathbf{x} \; dt \; dk \end{array} \tag{14}$$

After integration in the space $\Omega \times \Im$ and taking into account the notation (10) we obtain:

$$\begin{array}{l} \int\limits_{I} S^* \cdot \left(w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S \right) \; dt = \\ = - \int\limits_{I} S^* \cdot \left(\sum\limits_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum\limits_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \; dt \end{array} \tag{15}$$

Eq. (15) 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. (15) is:

$$w_1 \cdot r_1 \cdot \frac{dS}{dt} - w_2 \cdot r_2 \cdot S =$$

$$= - \left(\sum_{i=1}^{i=n} w_4^i \cdot r_5^i \cdot \frac{dT_i}{dt} - \sum_{i=1}^{i=n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \tag{16}$$

This equation 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 the present case the test function writes:

$$u^*(\mathbf{x}, t, k) = W^*(k) \cdot R(\mathbf{x}) \cdot S(t) \tag{17}$$

Now, the weak form reads

$$\begin{array}{l} \int\limits_{\Omega \times I \times \Im} W^* \cdot R \cdot S \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \, d\mathbf{x} \, dt \, dk = \\ = - \int\limits_{\Omega \times I \times \Im} W^* \cdot R \cdot S \cdot \mathcal{R}^n \, d\mathbf{x} \, dt \, dk \end{array} \tag{18}$$

that integrating in the space $\Omega \times I$ and taking into account the notation (10) results:

$$\begin{array}{l} \int\limits_{\Im} W^* \cdot (r_1 \cdot s_2 \cdot W - r_2 \cdot s_1 \cdot W) \, dk = \\ = - \int\limits_{\Im} W^* \cdot \left(\sum\limits_{i=1}^{i=n} r_5^i \cdot s_4^i \cdot K_i - \sum\limits_{i=1}^{i=n} r_4^i \cdot s_5^i \cdot K_i - r_3 \cdot s_3 \cdot f \right) \, dk \end{array} \tag{19}$$

Eq. (19) does not involve any differential operator. The strong form of Eq. (19) is:

$$(r_1 \cdot s_2 - r_2 \cdot s_1) \cdot W = - \left(\sum_{i=1}^{i=n} \left(r_5^i \cdot s_4^i - r_4^i \cdot s_5^i \right) \cdot K_i - r_3 \cdot s_3 \cdot f \right) \tag{20}$$

that represents an algebraic equation. Thus, the introduction of parameters as additional model coordinates has not a noticeable effect in the computational cost, because the original equation does not contain derivatives with respect to those parameters.

There are other minimization strategies more robust and exhibiting faster convergence for building-up the PGD, that we introduce in the next section in which the PGD constructor is presented in a matrix form.

3 A general formalism for the PGD

3.1 Separated representations

In what follows we are summarizing the main ideas that the Proper Generalized Decomposition (PGD) technique involves in a general formalism. For that purpose we suppose the following discrete form:

$$\mathcal{U}^{*T}\mathcal{A}\mathcal{U} = \mathcal{U}^{*T}\mathcal{B} \tag{21}$$

$\mathcal{U}$ and $\mathcal{U}^{*T}$ are the discrete description of both the trial and the test fields respectively. We assume that the problem is defined in a space of dimension D and can be written in a separated form:

$$\begin{aligned}
\mathcal{A} &= \sum_{i=1}^{n_A} \mathbf{A}_1^i \otimes \mathbf{A}_2^i \otimes \cdots \otimes \mathbf{A}_D^i \\
\mathcal{B} &= \sum_{i=1}^{n_B} \mathbf{B}_1^i \otimes \mathbf{B}_2^i \otimes \cdots \otimes \mathbf{B}_D^i \\
\mathcal{U} &= \sum_{i=1}^{n} \mathbf{u}_1^i \otimes \mathbf{u}_2^i \otimes \cdots \otimes \mathbf{u}_D^i
\end{aligned} \tag{22}$$

The separated representation of $\mathcal{A}$ and $\mathcal{B}$ comes directly from the differential operators involved in the PDE weak form.

3.2 Building-up the separated representation

At iteration n, vectors $\mathbf{u}_j^i$, $\forall i \leq n$ and $\forall j \leq D$ are assumed to be known. Now we are looking for an enrichment:

$$\mathcal{U} = \sum_{i=1}^{n} \mathbf{u}_1^i \otimes \cdots \otimes \mathbf{u}_D^i + \mathbf{R}_1 \otimes \cdots \otimes \mathbf{R}_D \tag{23}$$

where $\mathbf{R}_i$, $i = 1, \cdots, D$, are the unknown enrichment vectors. We assume the following form of the test field:

$$\mathcal{U}^* = \mathbf{R}_1^* \otimes \mathbf{R}_2 \otimes \cdots \otimes \mathbf{R}_D + \cdots + \mathbf{R}_1 \otimes \cdots \otimes \mathbf{R}_{D-1} \otimes \mathbf{R}_D^* \tag{24}$$

Introducing the enriched approximation into the weak form, the following discrete form results:

$$\sum_{i=1}^{n_A}\sum_{j=1}^{n} (\mathbf{R}_1^*)^T \mathbf{A}_1^i \mathbf{u}_1^j \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{u}_D^j + \cdots +$$

$$+ \sum_{i=1}^{n_A}\sum_{j=1}^{n} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{u}_1^j \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{A}_D^i \mathbf{u}_D^j +$$

$$+\sum_{i=1}^{n_A}(\mathbf{R}_1^*)^T\mathbf{A}_1^i\mathbf{R}_1\times\cdots\times(\mathbf{R}_D)^T\mathbf{A}_D^i\mathbf{R}_D+\cdots+$$
$$+\sum_{i=1}^{n_A}(\mathbf{R}_1)^T\mathbf{A}_1^i\mathbf{R}_1\times\cdots\times(\mathbf{R}_D^*)^T\mathbf{A}_D^i\mathbf{R}_D\ =$$
$$=\sum_{i=1}^{n_B}\left((\mathbf{R}_1^*)^T\mathbf{B}_1^i\times\cdots\times(\mathbf{R}_D)^T\mathbf{B}_D^i\ +\cdots+(\mathbf{R}_1)^T\mathbf{B}_1^i\times\cdots\times(\mathbf{R}_D^*)^T\mathbf{B}_D^i\right) \tag{25}$$

For the sake of clarity we introduce the following notation:

$$\sum_{i=1}^{n_C}\mathbf{C}_1^i\otimes\cdots\otimes\mathbf{C}_D^i=\sum_{i=1}^{n_B}\mathbf{B}_1^i\otimes\cdots\otimes\mathbf{B}_D^i-\sum_{i=1}^{n_A}\sum_{j=1}^{n}\mathbf{A}_1^i\mathbf{u}_1^j\otimes\cdots\otimes\mathbf{A}_D^i\mathbf{u}_D^j \tag{26}$$

where $n_C=n_B+n_A\times n$. This sum contains all the known fields. Thus Eq. (25) can be written as:

$$+\sum_{i=1}^{n_A}(\mathbf{R}_1^*)^T\mathbf{A}_1^i\mathbf{R}_1\times\cdots\times(\mathbf{R}_D)^T\mathbf{A}_D^i\mathbf{R}_D+\cdots+$$
$$+\sum_{i=1}^{n_A}(\mathbf{R}_1)^T\mathbf{A}_1^i\mathbf{R}_1\times\cdots\times(\mathbf{R}_D^*)^T\mathbf{A}_D^i\mathbf{R}_D\ =$$
$$=\sum_{i=1}^{n_C}\left((\mathbf{R}_1^*)^T\mathbf{C}_1^i\times\cdots\times(\mathbf{R}_D)^T\mathbf{C}_D^i\ +\cdots+(\mathbf{R}_1)^T\mathbf{C}_1^i\times\cdots\times(\mathbf{R}_D^*)^T\mathbf{C}_D^i\right) \tag{27}$$

This problem is strongly non linear. To solve it, an alternated directions scheme is applied. The idea consists to start with the trial vectors $\mathbf{R}_i^{(0)}$, $i=1,\cdots,D$ or to assume that these vector are known at iteration p, $\mathbf{R}_i^{(p)}$, $i=1,\cdots,D$, and to update them gradually using an appropriate strategy. We can either:

- Update vectors $\mathbf{R}_i^{(p+1)},\ \forall i$, from $\mathbf{R}_1^{(p)},\cdots,\mathbf{R}_{i-1}^{(p)},\mathbf{R}_{i+1}^{(p)},\cdots,\mathbf{R}_D^{(p)}$.

or:

- Update vectors $\mathbf{R}_i^{(p+1)},\ \forall i$, from $\mathbf{R}_1^{(p+1)},\cdots,\mathbf{R}_{i-1}^{(p+1)},\mathbf{R}_{i+1}^{(p)},\cdots,\mathbf{R}_D^{(p)}$.

The last strategy converges faster but the advantage of the first one is the possibility of updating each vector simultaneously making use of a parallel computing platform. The fixed point of this iteration algorithm allows defining the enrichment vectors $\mathbf{u}_i^{n+1}=\mathbf{R}_i,\ i=1,\cdots,D$.

When we look for vector $\mathbf{R}_k$ assuming that all the others $\mathbf{R}_i$, $i\neq k$ are known, the test field reduces to:

$$\mathcal{U}^{*T}=\mathbf{R}_1\otimes\cdots\otimes\mathbf{R}_{k-1}\otimes\mathbf{R}_k^*\otimes\mathbf{R}_{k+1}\cdots\otimes\mathbf{R}_D \tag{28}$$

The resulting discrete weak form writes:

$$\sum_{i=1}^{n_A} \left(\mathbf{R}_1^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times \mathbf{R}_k^{*T} \mathbf{A}_k^i \mathbf{R}_k \times \cdots \times \mathbf{R}_D^T \mathbf{A}_D^i \mathbf{R}_D \right) =$$

$$= \sum_{i=1}^{n_C} \mathbf{R}_1^T \mathbf{C}_1^i \times \cdots \times \mathbf{R}_k^{*T} \mathbf{C}_k^i \times \cdots \times \mathbf{R}_D^T \mathbf{C}_D^i \tag{29}$$

By applying the arbitrariness of $\mathbf{R}_K^*$ the following linear system can be easily obtained:

$$\left(\sum_{i=1}^{n_A} \left(\prod_{j=1, j\neq k}^{D} \mathbf{R}_j^T \mathbf{A}_j^i \mathbf{R}_j \right) \mathbf{A}_k^i \right) \mathbf{R}_k = \sum_{i=1}^{n_C} \left(\prod_{j=1, j\neq k}^{D} \mathbf{R}_j^T \mathbf{C}_j^i \right) \mathbf{C}_k^i \tag{30}$$

which can be easily solved.

3.3 Residual minimization

We have noticed that if instead of using the alternated directions iteration within the Galerkin framework for computing vectors $\mathbf{R}_i$, we compute these vectors by minimizing the residual:

$$\mathrm{Res} = \sum_{i=1}^{n_A} \mathbf{A}_1^i \mathbf{R}_1 \otimes \cdots \otimes \mathbf{A}_D^i \mathbf{R}_D - \sum_{i=1}^{n_C} \mathbf{C}_1^i \otimes \cdots \otimes \mathbf{C}_D^i \tag{31}$$

the convergence is significantly enhanced specifically for non symmetric operators $\mathbf{A}$.

We denote $\langle ., . \rangle$ a scalar product and $\|.\|$ its associated norm. Using this notation the residual norm writes:

$$\begin{aligned} \|\mathrm{Res}\|^2 \quad = \quad & \sum_{i=1}^{n_A} \sum_{j=1}^{n_A} \left(\left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{A}_1^j \mathbf{R}_1 \right\rangle \times \cdots \times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{A}_D^j \mathbf{R}_D \right\rangle \right) - \\ & -2 \sum_{i=1}^{n_A} \sum_{j=1}^{n_C} \left(\left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{C}_1^j \right\rangle \times \cdots \times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{C}_D^j \right\rangle \right) + \\ & + \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} \left(\left\langle \mathbf{C}_1^i, \mathbf{C}_1^j \right\rangle \times \cdots \times \left\langle \mathbf{C}_D^i, \mathbf{C}_D^j \right\rangle \right) \end{aligned} \tag{32}$$

The minimization problem with respect to $\mathbf{R}_k$ reads:

$$\frac{\partial}{\partial \mathbf{R}_k} \langle \mathrm{Res}, \mathrm{Res} \rangle = \mathbf{0} \tag{33}$$

or:

$$\sum_{i=1}^{n_A}\sum_{j=1}^{n_A}\left\langle \mathbf{A}_1^i\mathbf{R}_1, \mathbf{A}_1^j\mathbf{R}_1\right\rangle \times\cdots\times\left\langle \mathbf{A}_{k-1}^i\mathbf{R}_{k-1}, \mathbf{A}_{k-1}^j\mathbf{R}_{k-1}\right\rangle \times$$
$$\times\left\langle \mathbf{A}_k^i, \mathbf{A}_k^j\mathbf{R}_k\right\rangle \times\left\langle \mathbf{A}_{k+1}^i\mathbf{R}_{k+1}, \mathbf{A}_{k+1}^j\times\mathbf{R}_{k+1}\right\rangle \times\cdots\times\left\langle \mathbf{A}_D^i\mathbf{R}_D, \mathbf{A}_D^j\mathbf{R}_D\right\rangle -$$
$$-\sum_{i=1}^{n_A}\sum_{j=1}^{n_C}\left\langle \mathbf{A}_1^i\mathbf{R}_1, \mathbf{C}_1^j\right\rangle \times\cdots\times\left\langle \mathbf{A}_{k-1}^i\mathbf{R}_{k-1}, \mathbf{C}_{k-1}^j\right\rangle \times$$
$$\times\left\langle \mathbf{A}_k^i, \mathbf{C}_k^j\right\rangle \times\left\langle \mathbf{A}_{k+1}^i\mathbf{R}_{k+1}, \mathbf{C}_{k+1}^j\right\rangle \times\cdots\times\left\langle \mathbf{A}_D^i\mathbf{R}_D, \mathbf{C}_D^j\right\rangle = \mathbf{0} \tag{34}$$

The stopping criterion is defined from the residual norm:

$$\left\|\sum_{i=1}^{n_C}\mathbf{C}_1^i\otimes\cdots\otimes\mathbf{C}_D^i\right\|_2 < \epsilon \tag{35}$$

3.4 Remarks

It is important to notice that a general approximation in a domain consists in the full tensor product of the corresponding one-dimensional basis. A non separable function is a function needing all the terms of such full tensor product to be accurately approximated. However, many functions can be approximated by using a reduced number of the terms related to such tensor product, and then they can be called separable functions. The constructor that we describe in the present paper consists in computing functional products in order to guarantee a given accuracy. If the solution that we are trying to approximate is non separable, the enriching procedure continues until introducing the same number of functional products that the full tensor product involves. Thus, in the worst case, the proposed algorithm converges to the solution that could be obtained by using a full tensor product of the one-dimensional basis.

The proposed technique does not need an 'a priori' knowledge of the solution behavior. The functions involved in the approximations are constructed by the solver itself. However, the definition of the approximation functions over the different coordinates requires an approximation scheme. In our simulations we consider the simplest choice, a linear finite element representation.

We can also notice that the impact of the time step on transient models has no significant impact on the solver efficiency because the PGD solver reduces the time problem to a one dimensional first order differential equation whose integration can be performed efficiently even for very fine time discretizations. However, in the analyzed models involving time dependent parameters, if that dependence involves very small time steps the dimensionality of the model increases with the associated impact in the solver efficiency.

The PGD is specially appropriate for solving multidimensional models whose solution accepts a separated representation. It is difficult to establish 'a priori'

the separability of the model solution, but as there are no alternatives for circumventing the curse of dimensionality (except Monte Carlo simulations that are not free of statistical noise) our proposal is quite pragmatic: try and see!

4 Parametric models

4.1 A simple local problem

We focus here on a simple local problem for illustrating the main difficulties of parametric models and to test some numerical strategies to overcome these difficulties. The considered problem writes:

$$\mu \frac{du}{dt} = 1 \text{ with } u(t=0) = 0 \tag{36}$$

where u is a function of t and μ is a badly known parameter. The main difficulty in this kind of problem is that we can not solve it directly because of the uncertainty on the value of μ. Thus, it may be interesting to solve it for every μ in the incertitude interval. This could be performed by taking μ as a new coordinate (at the same level as time). Thus, the problem to solve writes:

$$y \frac{du(t,y)}{dt} = 1 \quad \forall t \in (0, t_{\max}] \quad \forall y \in [y_{\min}, y_{\max}] \tag{37}$$

where y is the coordinate associated to the parameter μ. Unfortunately this strategy increases significantly the complexity of the solution procedure because when the number of parameters increases, the model becomes highly multidimensional suffering from the so-called curse of dimensionality.

We are analyzing the capabilities of Proper Generalized Decomposition for computing the solution at each time t and for each value of the parameter μ that is, $u(t, \mu)$. We are considering two cases: (i) μ time independent; and (ii) μ time dependent.

4.1.1 Time-independent parameter

To overcome the difficulty related to the computing cost associated to the dimensionality increase, we are solving Eq. (37) using a separated representation. We consider the weak form:

$$\int_\Omega u^* y \frac{du(t,y)}{dt} d\Omega = \int_\Omega u^* d\Omega \tag{38}$$

where $\Omega = (0, t_{\max}] \times [y_{\min}, y_{\max}] = \Omega_t \times \Omega_y$.

We make the assumption that the solution can be written in the following separated form:

$$u = \sum_{i=1}^{\infty} T_i(t) \cdot Y_i(y) \tag{39}$$

Numerous models accept a separated representation consisting of a reduced number of functional products:

$$u \approx \sum_{i=1}^{n} T_i(t) \cdot Y_i(y) \tag{40}$$

The zero initial condition can be imposed by enforcing

$$T_i(t=0) = 0, \quad \forall i \tag{41}$$

If we want to enforce a non-zero initial condition, a variable change could be applied. We will discuss this point later. The introduction of the separated form (40) into (38) leads to:

$$\int_{\Omega} u^* \left(y \sum_{i=1}^{n} \frac{dT_i}{dt} \cdot Y_i - 1 \right) d\Omega = 0 \tag{42}$$

After discretization this equation can be rewritten formally:

$$\mathcal{U}^{*T} \mathcal{A} \mathcal{U} = \mathcal{U}^{*T} \mathcal{B} \tag{43}$$

with:

$$\begin{cases} \mathcal{A} = \mathbf{A}^t \otimes \mathbf{A}^y \\ \mathcal{B} = \mathbf{B}^t \otimes \mathbf{B}^y \\ \mathcal{U} = \sum_{i=1}^{n} \mathbf{T}_i \otimes \mathbf{Y}_i \end{cases} \tag{44}$$

Vectors $\mathbf{T}_i$ and $\mathbf{Y}_i$ contain the nodal values of the corresponding functions that are approximated using a standard linear finite element approximation. The components of matrices in Eq. (44) write:

$$\begin{cases} A^t_{ij} = \int_{\Omega_t} N^t_i \dfrac{dN^t_j}{dt} dt \\ A^y_{ij} = \int_{\Omega_y} N^y_i y N^y_j dy \\ B^t_i = \int_{\Omega_t} N^t_i dt \\ B^y_i = \int_{\Omega_y} N^y_i dy \end{cases} \tag{45}$$

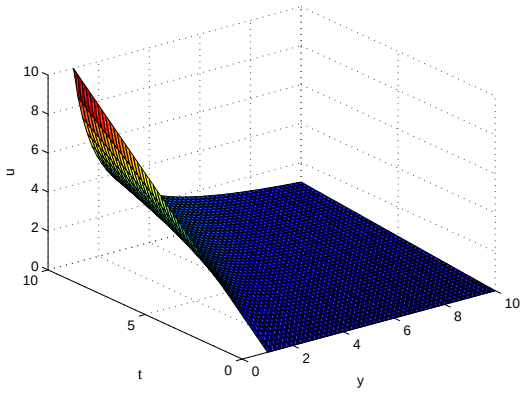
where N^t_i and N^y_i are the approximation shape functions. Once the global operators $\mathcal{A}$, $\mathcal{B}$ and the approximation $\mathcal{U}$ are defined, the Proper Generalized Decomposition can be applied. In the case here addressed of non symmetric differential operator, the minimization strategy converges faster.

In this case the exact solution of Eq. (37) is:

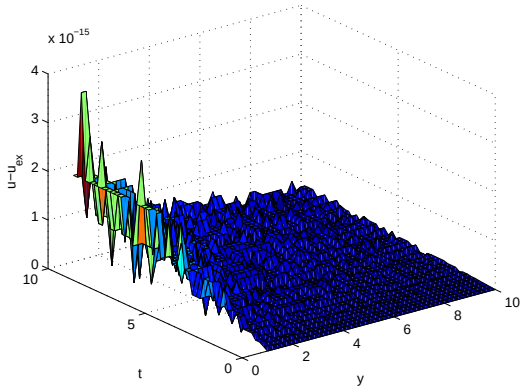
$$u_{ex} = \frac{t}{y} \tag{46}$$

that consists of a single functions product. Fig. 1 depicts the numerical solution associated with the Proper Generalized Decomposition previously described. Fig. 1 also shows the difference between the numerical and the analytical solutions. It can be noticed that the PGD solution is extremely accurate with an error of the order of 10^{-15}. The convergence was reached after a single iteration, the exact solution is a single functions product:

$$u_{ex} = T(t) \cdot Y(y) \quad \text{with} \quad T(t) = t \quad \text{and} \quad Y(y) = \frac{1}{y} \tag{47}$$



a)



b)

Figure 1: a) Computed solution. b) Comparison with the exact solution.

4.1.2 Time-dependent parameter

In what follows we are assuming:

$$y = 1 + t \tag{48}$$

The associated exact solution of Eq. (37) writes:

$$u_{ex} = \ln(1 + t) \tag{49}$$

For the sake of generality we are assuming a polynomial expression of y where the coefficients could be assumed unknown:

$$y = \sum_i a_i \cdot t^i \tag{50}$$

Obviously, each coefficient in the polynomial expansion can be assumed as a new coordinate. The model dimensionality depends on the number of coefficients.

The interest of such description is obvious. If one solves this problem, one has access to the most general solution $u(t, a_0, a_1, \cdots)$. Now, if an experimental curve is done, parameters a_i can be identified for minimizing $\|u(t, a_0, a_1, \cdots) - u^{exp}(t)\|$.

We are considering a simple linear evolution of y:

$$y = 1 + at \tag{51}$$

Now, the new additional coordinate is a instead of y. Eq. (37) leads to:

$$(1 + at)\frac{du(t,a)}{dt} = 1, \quad \forall t \in (0, t_{\max}], \quad \forall a \in [a_{\min}, a_{\max}] \tag{52}$$

Then, the separated representation of the unknown field writes:

$$u \approx \sum_{i=1}^{n} T_i(t) \cdot A_i(a) \tag{53}$$

that allows the use of the Proper Generalized Decomposition.

Fig. 2 depicts the computed solution $u(t, a)$. The solution accuracy is quantified by comparing the computed solution with the exact one. Fig. 3 compares both solutions in the case of $a = 0$ (that implies $y = 1$) and $a = 1$ (leading to $y = 1 + t$). The computed solution is in good agreement with the exact one. The error calculated using:

$$E(a) = \frac{1}{\|\Omega_t\|} \int_{\Omega_t} (u(t,a) - u_{ex}(t,a))^2 \, dt \tag{54}$$

was in order of 10^{-4} by using 10 terms in the finite sums decomposition.

The numerical error depends on the time step used in the time discretization. Tab. 1 shows its evolution with the time step considered whereas Fig. 4

depicts the evolution of the error with the number of terms n in the separated representation of $u(t, a)$. Obviously, this error decreases as the number of terms in the decomposition increases, but it approaches to the error associated to a full tensor product approximation. Further reduction of the error needs the use of finer 1D discretizations (meshes).

Δt	0.01	0.05	0.1	1
E	$1\ 10^{-5}$	$2.5\ 10^{-4}$	$1\ 10^{-3}$	$8\ 10^{-2}$

Table 1: Evolution of the error with the time step.

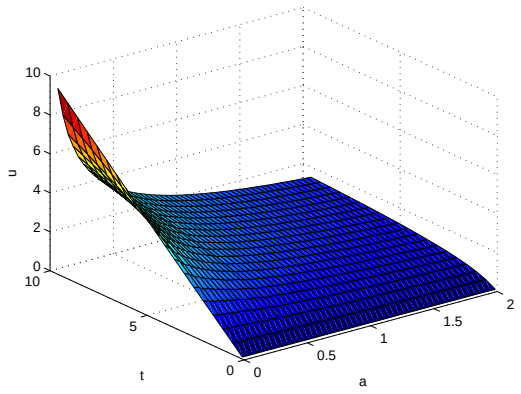


Figure 2: Numerical solution computed by applying the PGD.

4.1.3 Accounting for non-homogeneous initial conditions in local models

In order to enforce non-zero initial conditions the simplest way consists of applying a variable change. Therefore, we define a new unknown $\tilde{u}$ as:

$$\tilde{u} = u - u_0 \tag{55}$$

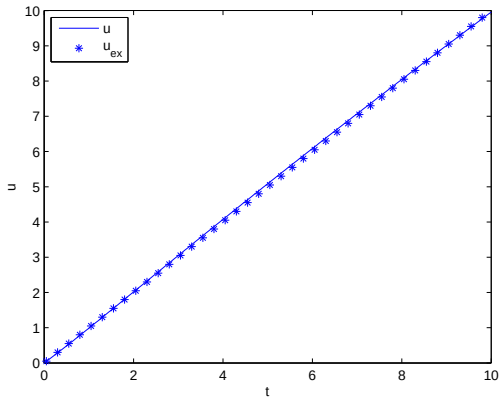
where u_0 is the initial condition. Indeed, enforcing $\tilde{u} = 0$ is equivalent to enforce $u = u_0$.

Introducing variable change in Eq. (37) the evolution of $\tilde{u}$ is governed by:

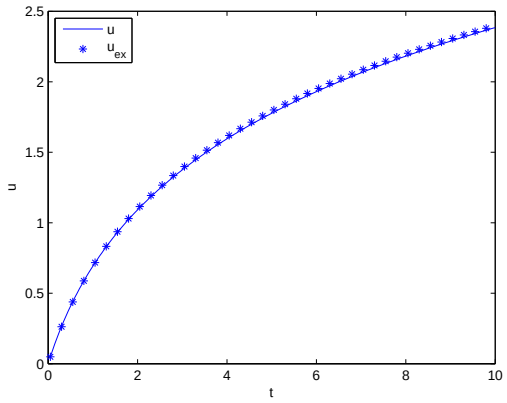
$$y\left(\frac{d\tilde{u}(t,y)}{dt} + \frac{du_0(t,y)}{dt}\right) = 1 \quad \forall t \in (0, t_{\max}] \quad \forall y \in [y_{\min}, y_{\max}] \tag{56}$$

But as u_0 is time independent, the equation becomes:

$$y\frac{d\tilde{u}(t,y)}{dt} = 1 \quad \forall t \in (0, t_{\max}] \quad \forall y \in [y_{\min}, y_{\max}] \tag{57}$$



a)



b)

Figure 3: Numerical versus exact solution of $u(t)$: a) for $a = 0$; b) for $a = 1$.

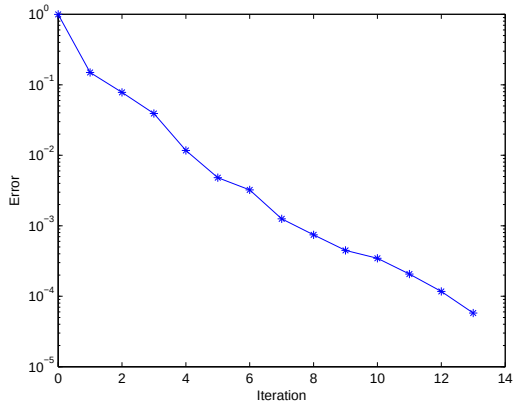


Figure 4: Error versus the number of sums in the finite sums decomposition.

which doesn't depend on u_0. Thus, just solving this equation is sufficient to know u for every value of u_0 from Eq. (55). Now, the separated representation based strategy can be used again for solving Eq. (57). In fact, it has already been performed because Eq. (57) is similar to Eq (37).

In a more general case, if u_0 appears in the equation governing the evolution of $\tilde{u}$ (after having performed the variable change), we could introduce u_0 as a new model coordinate. Thus, in that case, the separated representation writes:

$$\tilde{u}(t, y, u_0) \approx \sum_{i=1}^{n} T_i(t) \cdot Y_i(y) \cdot U_i(u_0) \tag{58}$$

This expression is valid because u_0 neither depends on time nor on y. The PGD leads to the solution of u at any t, y, u_0, i.e. $u(t, y, u_0)$.

The solution can be expressed by: $u = f(t, y, u_0)$. When the coordinate y is time dependent, it could be assumed piecewise constant, linear or polynomial. If we assume a piecewise constant variation, the solution $u = f(t, y, u_0)$ could give the solution at each step of y. If we assume a length Δt_y of the y steps, then $u = f(t, y_1, u_0)$ is valid in $[0, \Delta t_y]$. Now, if we assume that $u_1 = f(\Delta t_y, y_1, u_0)$, then the solution for the next value of y, y_2, is given by: $u = f(t - \Delta t_y, y_2, u_1)$ and so on. Thus, only the solution of the model in the time interval $]0, \Delta t_y]$ is required. Obviously, if $\Delta t_y = \Delta t$ (Δt being the time step used in the time discretization) there is not computing-time reduction because the number of calculations becomes the same as the one involved in a standard incremental solution. However, if $\Delta t_y >> \Delta t$, then the computing-time savings can be significant.

Eqs. (57) and (55) allow computing the solution for all t, y and u_0. Now, we are considering the previous algorithm to build-up the solution for a time evolving parameter. Fig. 5 depicts the computed solution for an evolution of y given by $y = 1 + t$. We can notice that the solution is in perfect agreement

with the exact solution. The corresponding error is 1.5×10^{-5}. This solution was computed using $\Delta t_y = \Delta t$ allowing a good accuracy but in fact the solution is absolutely equivalent to a backward standard time integration. However, when the time evolution of y is smooth enough we can approximate it from a piecewise constant description. Thus, the function y is assumed constant in each interval of length $\Delta t_y >> \Delta t$ defined by a partition t_i^y of the whole time interval ($t_{i+1}^y - t_i^y = \Delta t_y,\ t_0^y = 0$).

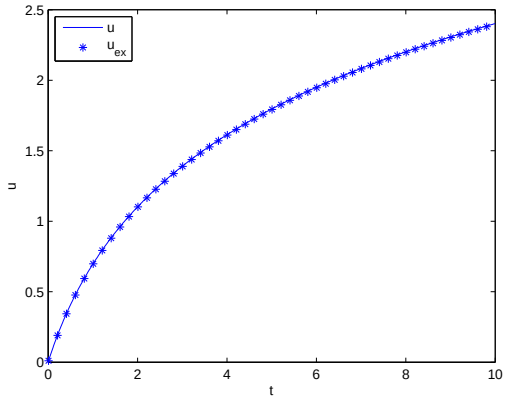


Figure 5: Numerical versus exact solution $u(t)$ with $y = 1 + t$.

In this case the solution reconstruction from the separated representation $u(t, y, u_0)$ is obtained from:

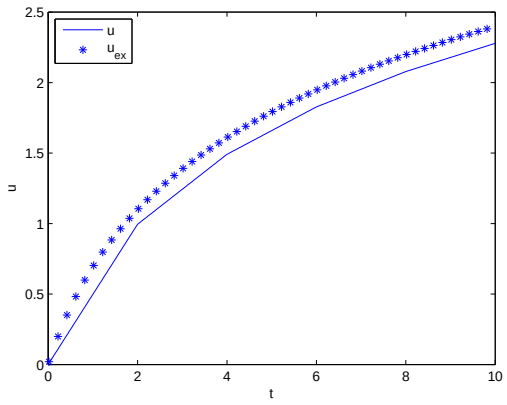
$$u(t_{i-1}^y \leq t \leq t_i^y) = f\left(t - t_{i-1}^y, y_i, u(t_{i-1}^y)\right), \quad \forall i \geq 1 \tag{59}$$

By using this strategy the computing time savings increase as the ratio $\Delta t_y/\Delta t$ increases. Fig. 6 compares the computed and the exact solutions for $y = 1 + t$ and for two different sizes of Δt_y. As expected, the solution accuracy increases as the number of intervals increases. With 10 intervals (each one containing 20 time steps, i.e. $\Delta t_y = 20 \cdot \Delta t$), the numerical solution fits the exact one whereas the number of iterations is reduced in the order of 20. The error when using 10 intervals was $2 \cdot 10^{-3}$ and the one when using 5 intervals was $2 \cdot 10^{-2}$.

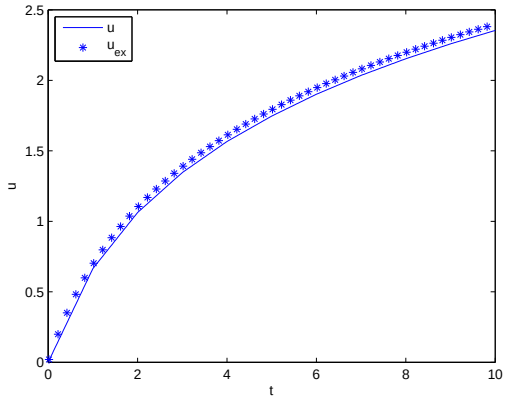
Further improvements can be attained by performing a polynomial approximation of y on each interval Δt_y.

In general, the solution of a model depends strongly on the initial condition. This fact motivates the introduction of the initial condition as a new model coordinate. We are now considering a bit more complex problem:

$$y\frac{du}{dt} + u = h \quad \text{with} \quad u(t = 0) = u_0 \tag{60}$$



a)



b)

Figure 6: Numerical versus exact solution of $u(t)$ with $y = 1+t$: a) $\Delta t_y = 20 \cdot \Delta t$; and b) $\Delta t_y = 10 \cdot \Delta t$.

where h corresponds to a source term that could be also considered unknown or badly known. When $h = 0$ the exact solution writes:

$$u = u_0 \exp\left(-\frac{t}{y}\right) \tag{61}$$

For solving Eq. (60) we introduce a change of variable:

$$\tilde{u} = u - u_0 \tag{62}$$

which leads to:

$$y\frac{d\tilde{u}}{dt} + \tilde{u} + u_0 = h \quad \text{with} \quad \tilde{u}(t=0) = 0 \tag{63}$$

Then a separated representation is performed including the coordinates $t,\ y,\ u_0$ and the unknown source term h. From this general solution $u(t, y, u_0, h)$ as soon as the different model parameters are given (y^g, u_0^g, h^g) one could extract the time solution evolution from $u(t, y^g, u_0^g, h^g)$. Figure 7 depicts the computed solution $u(t, u_0)$ for $h^g = 0$ and $y^g = 1$. The maximum difference between the computed and the exact solutions never exceeds the value of 0.04 which represents about 0.2% of the solution. Fig. 8 shows similar results for $u_0 = 5$. The error calculated using Eq. (54) is $8{\cdot}10^{-5}$.

5 Solving non-local non-linear parametric models

Until now, only local linear problems have been considered, however Proper Generalized Decomposition can be also successfully applied for solving non-local non-linear models. In what follows we are describing the solution of such models defined by parabolic non-linear partial differential equations containing some unknown or badly known parameters.

5.1 Treating non-linearities

For the sake of simplicity we consider the one-dimensional non-linear heat equation:

$$\frac{\partial u}{\partial t} - k\frac{\partial^2 u}{\partial x^2} = u^2 + f(t,x) \quad \forall t \in \Omega_t \quad \forall x \in \Omega_x \tag{64}$$

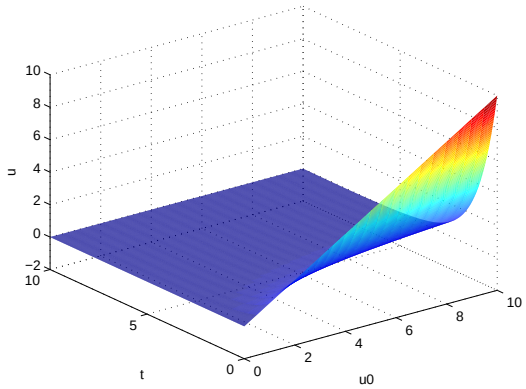
where u is the temperature field, k is the thermal diffusivity assumed constant and f is a source term. This model is defined in $\Omega_x = (x_{\min}, x_{\max}) \times \Omega_t = (0, t_{\max}]$.

The initial condition is:

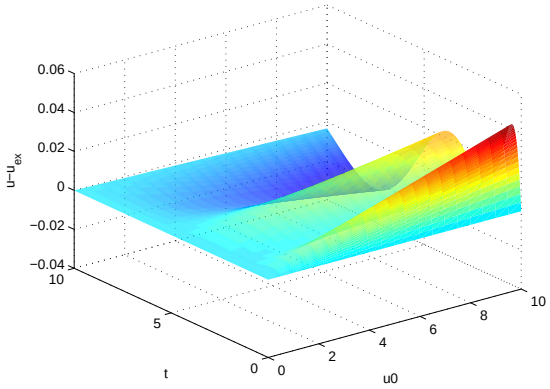
$$u(0,x) = 0, \quad \forall x \in \Omega_x \tag{65}$$

and the boundary conditions are:

$$u(t, x_{\min}) = u(t, x_{\max}) = 0, \quad \forall t \in \Omega_t \tag{66}$$



a)



b)

Figure 7: a) Computed solution $u(t, u_0)$. b) Computed and exact solutions comparison.

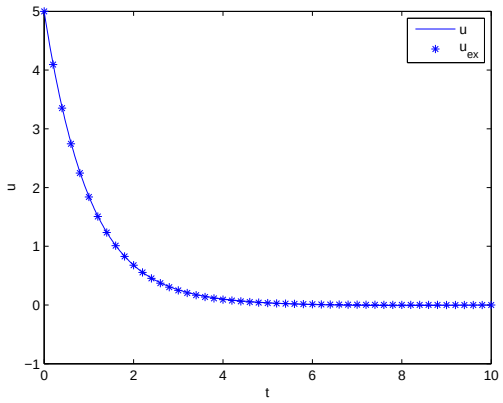


Figure 8: Computed solution $u(t, y = 1, u_0 = 5, h = 0)$.

The weak formulation results:

$$\int_{\Omega_t} \int_{\Omega_x} u^* \left(\frac{\partial u}{\partial t} - k \frac{\partial^2 u}{\partial x^2} - u^2 - f \right) d\Omega = 0 \tag{67}$$

This section focuses in the treatment of the non linearities, and then all the model parameters are assumed known. Thus, the separated representation of the unknown field writes:

$$u(t, x) = \sum_{i=1}^{\infty} T_i(t) \cdot X_i(x) \tag{68}$$

For building-up such separated approximation we look at iteration n for the functions $R(t)$ and $S(x)$:

$$u(t, x) = \sum_{i=1}^{n} T_i(t) \cdot X_i(x) + R(t) \cdot S(x) = u^n + R \cdot S \tag{69}$$

where functions T_i and X_i were computed at previous iterations.

Different strategies exist for treating the presence of the non-linear term u^2. We consider three possibilities:

- The non-linear term is evaluated from the solution at the previous iteration u^n:

$$u^2 \approx (u^n)^2 = \left(\sum_{i=1}^{n} T_i \cdot X_i \right)^2 \tag{70}$$

  It is direct to conclude that when the enrichment procedure converges, $\|u^n - u^{n-1}\| \leq \epsilon$, and then $\|(u^n)^2 - (u^{n-1})^2\| \leq \epsilon'$ that guarantees the solution of the non-linear model.

- Another possibility lies in partially using the solution just computed within the non-linear solver iteration scheme:

$$u^2 \approx \left(\sum_{i=1}^{n} T_i \cdot X_i\right) \cdot \left(\sum_{i=1}^{n} T_i \cdot X_i + R(t) \cdot S(x)\right) \tag{71}$$

- A third possibility is a variant of the previous one that considers:

$$u^2 \approx \left(\sum_{i=1}^{n} T_i \cdot X_i + R^{(k-1)} \cdot S^{(k-1)}\right) \left(\sum_{i=1}^{n} T_i \cdot X_i + R^{(k)} \cdot S^{(k)}\right) \tag{72}$$

  where k denotes the iteration of the non-linear solver used for computing the enrichment functions $R(t)$ and $S(x)$.

5.1.1 Numerical results

To test the different strategies we consider a problem with a known exact solution.

For this purpose we consider Eq. (64) with the source term:

$$f(t,x) = (16\pi^2 t + 1)\sin(4\pi x) - t^2 \sin(4\pi x)^2 \tag{73}$$

where the exact solution writes:

$$u_{ex}(t,x) = t\sin(4\pi x) \tag{74}$$

The exact solution (74) involves a single product of space and time functions.

The first strategy described in the previous section computes 9 couples of functions. In fact, the number of couples depends more on the efficiency of the linearization strategy than in the separability of the exact solution. The second strategy is expected to give better results because of the better representation of non-linearity. Thus, similar precision was reached with 8 functional couples. Finally, the third strategy computes a single couple of functions for the same precision. Fig. 9 compares the convergence rates of these three strategies.

The third strategy seems to be the best one where the number of functional couples only depends on the separability of the exact solution. However, a computational cost very close to the other ones.

5.2 Non-linear parametric models

In this section we introduce parameters in the model. We consider again the one-dimensional heat transfer equation

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) = 0 \quad \forall t \in \Omega_t \quad \forall x \in \Omega_x \tag{75}$$

where the thermal diffusivity is assumed depending on the temperature field, i.e. $k(u)$:

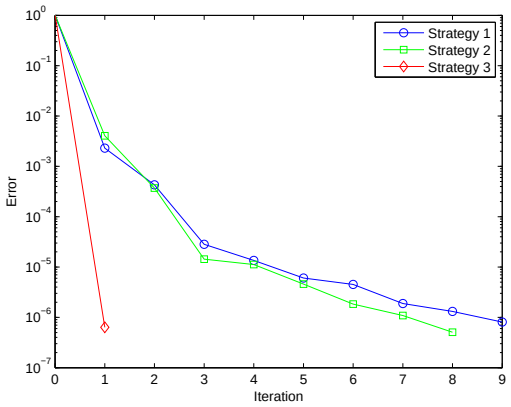


Figure 9: Convergence analysis for the three non-linear solution strategies.

$$k = au + b \tag{76}$$

where coefficients a and b are considered as parameters that will be associated to new model coordinates.

Introducing the diffusivity expression into the heat transfer equation (75) we obtain:

$$\frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} - au\frac{\partial^2 u}{\partial x^2} - a\left(\frac{\partial u}{\partial x}\right)^2 = 0 \tag{77}$$

The purpose of the resolution of that equation is the calculation of the temperature at each point and time, and for any value of the parameters a and b within their domains of variability, i.e. $u(t, x, a, b)$:

$$u(t, x, a, b) \approx \sum_{i=1}^{n} T_i(t) \cdot X_i(x) \cdot A_i(a) \cdot B_i(b) \tag{78}$$

with, in our numerical experiments, $t \in \Omega_t = (0, 1]$, $x \in \Omega_x = (-1, 1)$, $a \in \Omega_a = [-1, 1]$ and $b \in \Omega_b = [1, 5]$.

The initial condition writes:

$$u(t = 0, x, a, b) = 1 - x^{10} \tag{79}$$

and the temperature is assumed vanishing on the boundary of the spatial domain $x = -1$ and $x = 1$.

We consider the approximations of the different functions $T_i(t)$, $X_i(x)$, $A_i(a)$ and $B_i(b)$ performed by using standard one dimensional linear finite element shape functions on a uniform mesh consisting of 500 nodes in each 1D-domain Ω_t, Ω_x, Ω_a and Ω_b. If this problem is solved using a mesh-based strategy

in the whole domain the complexity scales with 500^4. However, the Proper Generalized Decomposition needed a single minute using a personal laptop. Fig. 10 illustrates the computed solution for $t = 1$ and $x = 0$.

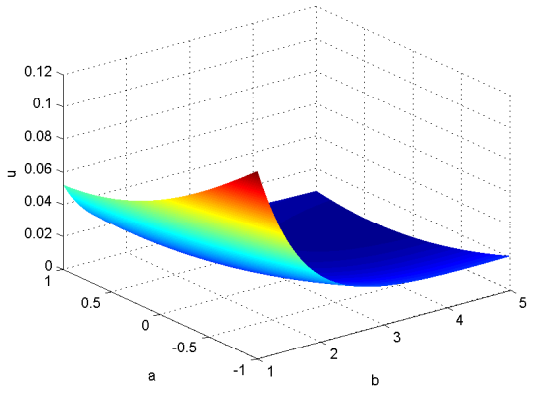


Figure 10: Temperature versus parameters a and b defining the thermal diffusivity for $t = 1$ and $x = 0$.

6 Conclusion

In this paper we revisited the Proper Generalized Decomposition technique. This discretization strategy is based on the use of a separated representation of the differential operators, the functions and the unknown fields involved in a partial differential equation. The numerical complexity scales linearly with the dimension of the space instead of the expected exponential scaling characteristic of mesh based discretization techniques.

The main advantage lies in the fact that in some models involving unknown or badly known parameters, these parameters could be included as additional coordinates. Despite the fact that the dimension of the model increases, the separated representation allows its efficient and accurate solution. We illustrated this procedure by considering some simple problems where bifurcations are not present. We have also analyzed the solution of non-linear parabolic models, where the eventual presence of uncertain parameters was also incorporated, proving the potentiality of the proper generalized decomposition for addressing complex thermomechanical models encountered in computational mechanics and engineering.

References

[1] Y. Achdou and O. Pironneau. Siam frontiers in applied mathematics. *Computational methods for option pricing*, 2005.

[2] A. Ammar and F. Chinesta. Circumventing curse of dimensionality in the solution of highly multidimensional models encountered in quantum mechanics using meshfree finite sums decomposition. *Lectures Notes on Computational Science and Engineering*, 65:1–17, 2008.

[3] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids. *J. Non-Newtonian Fluid Mech.*, 139:153–176, 2006.

[4] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings. 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-Newtonian Fluid Mech.*, 144(2-3):98–121, 2007.

[5] A. Ammar, D. Ryckelynck, F. Chinesta, and R. Keunings. On the reduction of kinetic theory models related to finitely extensible dumbbells. *J. Non-Newtonian Fluid Mech.*, 134:136–147, 2006.

[6] M. Berveiller, B. Sudret, and M. Lemaire. Stochastic finite element: a non-intrusive approach by regression. *Eur. J. Comput. Mech.*, 15:81–92, 2006.

[7] G. Beylkin and M. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM J. Sci. Com.*, 26(6):2133–2159, 2005.

[8] H.J. Bungartz and M. Griebel. Sparse grids. *Acta numerica*, 13:1–123, 2004.

[9] R.E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49, 1998.

[10] E. Cancès, M. Defranceschi, W. Kutzelnigg, C. Le Bris, and Y. Maday. Computational quantum chemistry: a primer. *Handbook of Numerical Analysis*, X:3–270, 2003.

[11] F. Chinesta, A. Ammar, and P. Joyot. The nanometric and micrometric scales of the structure and mechanics of materials revisited: An introduction to the challenges of fully deterministic numerical descriptions. *International Journal for Multiscale Computational Engineering*, 6(3):191–213, 2008.

[12] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene, and F. Boust. Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. *Comput. Methods Appl. Mech. Engrg.*, 197(5):400–413, 2008.

[13] R. Ghanem and P. Spanos. Stochastic finite elements: A spectral approach. *Springer, Berlin*, 1991.

[14] P. Ladeveze. Nonlinear computational structural mechanics. *Springer, NY.*, 1999.

[15] R.B. Laughlin. The theory of everything. In *Proceeding of the U.S.A National Academy of Science*, 2000.

[16] H.G. Matthies and A. Keese. Galerkin methods for linear and nonlinear elliptic stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 194(12-16):12951331, 2005.

[17] B. Mokdad, E. Pruliere, A. Ammar, and F. Chinesta. On the simulation of kinetic theory models of complex fluids using the fokker-planck approach. *Applied Rheology*, 17(2):26494, 1–14, 2007.

[18] A. Nouy. Recent developments in spectral stochastic methods for thenumerical solution of stochastic partial differential equations. *Archives of Computational Methods in Engineering*, In press.

[19] A. Nouy. A generalized spectral decomposition technique to solve a class of linear stochastic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 196:4521–4537, 2007.

[20] M. Papadrakakis and V. Papadopoulos. Robust and efficient methods for stochastic finite element analysis using monte carlo simulation. *Comput. Methods Appl. Mech. Engrg.*, 134:325–340, 1996.

[21] B. Puig, F. Poirion, and C. Soize. Non-gaussian simulation using hermite polynomial expansion: convergences. *Probab. Engrg. Mech.*, 17:252–264, 2002.

[22] T. M. Rassias and J. Simsa. Finite sums decompositions in mathematical analysis. *John Wiley and Sons Inc.*, 1995.

[23] D. Ryckelynck, F. Chinesta, E. Cueto, and A. Ammar. On the a priori model reduction: overview and recent developements. *Archives of compt. Meths in Engineering*, 13(1):91–128, 2006.