Arch Comput Methods Eng (2010) 17: 327–350
DOI 10.1007/s11831-010-9049-y

ORIGINAL PAPER

Recent Advances and New Challenges in the Use of the Proper Generalized Decomposition for Solving Multidimensional Models

Francisco Chinesta · Amine Ammar · Elías Cueto

Received: 15 March 2010 / Accepted: 15 March 2010 / Published online: 7 October 2010

Abstract This paper revisits a powerful discretization technique, the Proper Generalized Decomposition—PGD, illustrating its ability for solving highly multidimensional models. This technique operates by constructing a separated representation of the solution, such that the solution complexity scales linearly with the dimension of the space in which the model is defined, instead the exponentially-growing complexity characteristic of mesh based discretization strategies. The PGD makes possible the efficient solution of models defined in multidimensional spaces, as the ones encountered in quantum chemistry, kinetic theory description of complex fluids, genetics (chemical master equation), financial mathematics, … but also those, classically defined in the standard space and time, to which we can add new extra-coordinates (parametric models, …) opening numerous possibilities (optimization, inverse identification, real time simulations, …).

1 Introduction

Many models encountered in science and engineering are defined in multidimensional spaces, as the ones involved in quantum chemistry, kinetic theory descriptions of materials (including complex fluids), the chemical master equation governing many biological processes (e.g. cell signaling), models of financial mathematics (e.g. option pricing), among many others. These models exhibit the redoubtable curse of dimensionality when usual mesh-based discretization techniques are applied. Other times, standard models can become multidimensional if some of the parameters that they involve are considered as new coordinates. This possibility is specially attractive when these coefficients are not well known, they have a stochastic nature, or when one is interested in optimization or inverse identification.

The difficulty related to the solution of multidimensional models is quite obvious and it needs the proposal of new appropriate strategies able to circumvent the curse of dimensionality. One possibility lies in the use of sparse grids [12]. However, as argued in [1], the use of sparse grid is restricted to models with moderate multidimensionality (up to 20). Another technique able to circumvent, or at least alleviate, the curse of dimensionality consists of using a separated representation of the unknown field (see [10, 44] for some numerical elements on this topic). Basically, the separated representation of a generic function $u(\mathbf{x}_1, \ldots, \mathbf{x}_D)$ (also known as finite sum decomposition) writes:

$$u(\mathbf{x}_1, \ldots, \mathbf{x}_D) \approx \sum_{i=1}^{i=N} F_i^1(\mathbf{x}_1) \times \cdots \times F_i^D(\mathbf{x}_D) \qquad (1)$$

Remark 1 The coordinates $\mathbf{x}_i$, $i = 1, \ldots, D$, are defined in spaces of moderate dimension, i.e. $\mathbf{x}_i \in \Omega_i \subset \mathbb{R}^{d_i}$,

Research partially supported by the Spanish Ministry of Science ad Innovation through grant CICYT-DPI2008-00918.

F. Chinesta (✉)
EADS Corporate Fundation International Chair,
GEM, UMR CNRS–Centrale Nantes, 1 rue de la Noe, BP 92101,
44321 Nantes Cedex 3, France
e-mail: Francisco.Chinesta@ec-nantes.fr

A. Ammar
Arts et Métiers ParisTech, 2 Boulevard du Ronceray, BP 93525,
49035 Angers Cedex 01, France
e-mail: Amine.AMMAR@ensam.eu

E. Cueto
Group of Structural Mechanics and Materials Modelling,
Aragón Institute of Engineering Research (I3A),
Universidad de Zaragoza, Maria de Luna, 3, 50018 Zaragoza,
Spain
e-mail: ecueto@unizar.es

$d_i \leq 3$. Thus, the dimension of the model results $\sum_{i=1}^{i=D} d_i$. Eventually, one of these coordinates could be the time $t \in \mathcal{I} \subset \mathbb{R}^+$.

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 post-Hartree-Fock approaches (as the MCSCF that involves a finite number of sums) made use of a separated representation of the wavefunction [13, 19].

We proposed recently a technique able to construct, in a way completely transparent for the user, the separated representation of the unknown field involved in a multidimensional partial differential equation. This technique, originally described and applied to multi-bead-spring FENE models of polymeric liquids in [3], was extended to transient models of such complex fluids in [4]. More complex models (involving different couplings and non-linearities) based on the reptation theory of polymeric liquids were analyzed in [34].

In the current life when one is buying a car there are many choices concerning the color, the power, ... In some cases the number of possible variants reaches astronomical values: 10^{50} in the case of a highly appreciated luxury car. Thus, the construction of a high dimensional matrix representing all the possible variants is simply impossible (we must recall that the presumed number of elementary particles in our universe is of around 10^{80}!). Obviously, the only possibility of following the sales of such a car is using a sparse storage structure. This is well known and widely used in practice. However, our present interest is not only in storing information, but in solving models defined in multidimensional spaces.

When the solution of highly multidimensional models explores the whole space there is no possibility to follow or represent such a system. This is a well known phenomenon in quantum systems. However, one could track the evolution of such a system in a time interval of moderate length by using a "sparse" representation. In our knowledge two possibilities exist, the first one consists in using Monte Carlo simulations within a stochastic framework. The second one, within a purely deterministic framework, lies in using separated representations like the one expressed by (1) where the number of sums increases as the solution explores the more and more larger domains. On the other hand, if the solution only explores a small enough subdomain of the whole domain the use of sparse grids or separated representations are some appealing candidates for discretizing such models.

Coming back to models defined in spaces of moderate dimension ($d \times D$, $d = 1, 2, 3$) but whose solutions evolve in large time intervals, if one uses standard incremental time-discretizations, in the general case (models involving time-dependent parameters, non-linear models, ...), one must solve at least a linear system at each time step. When the time step becomes too small as a consequence of stability requirements, and the simulation time interval is large enough, standard incremental simulation becomes inefficient. To illustrate this scenario, one could imagine the simple reaction-diffusion model that describes the degradation of plastic materials, where the characteristic time of the chemical reaction involved in the material degradation is in the order of some microseconds and the one related to the diffusion of chemical substances (that also represents the material degradation characteristic time itself) is of the order of years. In this case standard incremental techniques must be replaced by other more efficient strategies.

One possibility consists again in performing a separated representation of the unknown field, that in the present case reduces to:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \tag{2}$$

that allows, as we describe later, to non-incremental time integration strategies, which can reduce spectacularly the CPU time.

This space-time separated representation is not a new proposal. In fact such decompositions were proposed many year ago by Pierre Ladeveze as an ingredient of the powerful non-linear-non-incremental LATIN solver that he proposed in the 80s. During the last twenty years many works were successfully accomplished by the Ladeveze's group. The interested reader can refer to [29, 30, 32] and the references therein. In the radial approximation approach (the name given in the pioneer works of Ladeveze) functions depending on space and the ones depending on time were a priori unknown, and they were computed by an appropriate minimization technique.

In what follows we are reporting the recent advances in the solution of multidimensional models by applying the *Proper Generalized Decomposition* (PGD), that is, a separated representation of the unknown fields.

1.1 Motivating the Use of Separated Representations

From a historical point of view, separated representations were extensively used. The analytical solution of PDEs (elliptic, parabolic and hyperbolic) that the reader can find in any book devoted to the solution of PDEs starts assuming a separated representation: the method of separation of variables. After some manipulations, simple but sometimes quite technical, the final solution could be some times found in the form expressed by (1).

Another well established and widely employed technique allowing to define a separated representation of a given space-time function is based on the application of a proper orthogonal decomposition. We are illustrating the main ideas related to this technique.

Let $u(\mathbf{x}, t)$ be the solution of a certain transient model (in what follows $\mathbf{x} \in \Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$, and $t \in \mathcal{I} \subset \mathbb{R}^+$). We are also assuming that this field is known in a discrete manner, that is, at some points $\mathbf{x}_i$ (the nodes of a mesh or a grid) and at certain times $t_p = p \times \Delta t$, where $i \in [1, \ldots, N_n]$ and $p \in [1, \ldots, P]$.

Now, we introduce the notation $u_i^p \equiv u(\mathbf{x}_i, t_p)$ and construct the matrix $\mathbf{Q}$ that contains the snapshots:

$$\mathbf{Q} = \begin{pmatrix} u_1^1 & u_1^2 & \cdots & u_1^P \\ u_2^1 & u_2^2 & \cdots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_n}^1 & u_{N_n}^2 & \cdots & u_{N_n}^P \end{pmatrix} \tag{3}$$

The proper orthogonal decomposition (POD) of this discrete field consists in solving the eigenvalue problem:

$$\left(\mathbf{Q}\mathbf{Q}^T\right) \boldsymbol{\phi} = \lambda \boldsymbol{\phi} \tag{4}$$

that results in N_n couples eigenvalue-eigenvector $(\lambda_i, \boldsymbol{\phi}_i)$, $i = 1, \ldots, N_n$.

When the field evolves smoothly, the magnitude of the eigenvalues decreases very fast, fact that reveals that the evolution of the field can be approximated from a reduced number of modes (eigenvectors). Thus, if we define a cutoff value ϵ ($\epsilon = 10^{-8} \times \lambda_1$ in practice, λ_1 being the highest eigenvalue) only a reduced number of modes are retained. Let R ($R \ll N_n$) be the number of modes retained, i.e. $\lambda_i \geq 10^{-8} \times \lambda_1, i = 1, \ldots, R$ and $\lambda_i < 10^{-8} \times \lambda_1, \forall i > R$ (the eigenvalues are assumed ordered). Thus, one could write:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{i=R} \phi_i(\mathbf{x}) \cdot T_i(t) \equiv \sum_{i=1}^{i=R} X_i(\mathbf{x}) \cdot T_i(t) \tag{5}$$

where for the sake of clarity the space modes $\phi_i(\mathbf{x})$ will be, from now on, denoted as $X_i(\mathbf{x})$. Equation (5) represents a natural separated representation (also known as finite sums decomposition).

These modes could be now used to solve other "similar" problems, that is, models involving slight changes in the boundary conditions, model parameters, … [33, 39, 46]. Other possibility is to compute the reduced basis from the standard transient solution within a short time interval (with respect to the whole time interval in which the model is defined) and then solve the remaining part of the time interval by employing the reduced basis. Obviously, both strategies induce the introduction of an error whose evaluation, control and reduction is a challenging issue.

One possibility to construct an adaptive reduced approximation basis, that should be the best reduced approximation basis for the treated problem, consists in alternating a reduction step (based on the application of the proper orthogonal decomposition) and an enrichment stage to improve the quality of the reduced approximation basis in order to capture all the solution features. We proposed recently an enrichment technique based on the use of some Krylov's subspaces generated by the equation residual. This technique known as "a priori" model reduction was originally proposed in [47], widely described in [48] and successfully applied for solving complex fluid flows within the kinetic theory framework [2, 17] and for speeding up thermomechanical simulations [18]. However, some difficulties were noticed in the application of this strategy: (i) the enrichment based on the use of the Krylov's subspaces is far to be optimal in a variety of models (e.g. the wave equation); (ii) the incremental nature of the algorithm; … .

From the previous analysis we can conclude: (i) the transient solution of numerous models can be expressed using a very reduced number of products each one involving a function of time and a function of space; and (ii) the functions involved in these functional products should be determined simultaneously by applying an appropriate algorithm to guarantee robustness and optimality.

In what follows we are illustrating the simplest strategy able to compute these separated functional couples.

2 Illustrating the Proper Generalized Decomposition

In this section we are illustrating the discretization of partial differential equations using a separated representation (radial approximation in the Ladeveze's terminology) of the unknown field.

Let us consider the advection-diffusion equation

$$\frac{\partial u}{\partial t} - a \Delta u + \mathbf{v} \cdot \nabla u = f(\mathbf{x}, t) \quad \text{in } \Omega \times (0, T_{max}] \tag{6}$$

with the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x}, 0) = u^0, & \mathbf{x} \in \Omega \\ u(\mathbf{x}, t) = u_g, & (\mathbf{x}, t) \in \partial\Omega \times (0, T_{max}] \end{cases} \tag{7}$$

where a is the diffusion coefficient and $\mathbf{v}$ the velocity field, $\Omega \subset \mathbb{R}^d, 1 \leq d \leq 3$, $T_{max} > 0$. The aim of the separated representation method is to compute N couples of functions $\{(X_i, T_i)\}_{i=1,\ldots,N}$ such that $\{X_i\}_{i=1,\ldots,N}$ and $\{T_i\}_{i=1,\ldots,N}$ are defined respectively in $\overline{\Omega}$ and $[0, T_{max}]$ and the solution u of this problem can be written in the separate form

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} T_i(t) \cdot X_i(\mathbf{x}) \tag{8}$$

The weak form of problem (6) yields:

Find $u(\mathbf{x}, t)$ verifying the boundary conditions (7) such that

$$\int_0^{T_{max}} \int_{\Omega} u^\star \left(\frac{\partial u}{\partial t} - a\Delta u + \mathbf{v} \cdot \nabla u - f(\mathbf{x}, t) \right) d\mathbf{x}\, dt = 0 \tag{9}$$

for all the functions $u^\star(x, t)$ in an appropriate functional space.

We compute now the functions involved in the sum (8). We suppose that the set of functional couples $\{(X_i, T_i)\}_{i=1,\ldots,n}$ with $0 \le n < N$ are already known (they have been previously computed) and that at the present iteration we search the enrichment couple $(R(t), S(\mathbf{x}))$ by applying an alternating directions fixed point algorithm that after convergence will constitute the next functional couple (X_{n+1}, T_{n+1}). Hence, at the present iteration, n, we assume the separated representation

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{n} T_i(t) \cdot X_i(\mathbf{x}) + R(t) \cdot S(\mathbf{x}) \tag{10}$$

The weighting function $u^\star$ is then assumed as

$$u^\star = S \cdot R^\star + R \cdot S^\star \tag{11}$$

Introducing (10) and (11) into (9) it results

$$\begin{aligned} &\int_0^{T_{max}} \int_{\Omega} (S \cdot R^\star + R \cdot S^\star) \\ &\quad \cdot \left(S \cdot \frac{\partial R}{\partial t} - a\Delta S \cdot R + (\mathbf{v} \cdot \nabla S) \cdot R \right) d\mathbf{x}\, dt \\ &= \int_0^{T_{max}} \int_{\Omega} (S \cdot R^\star + R \cdot S^\star) \cdot \left(f(\mathbf{x}, t) - \sum_{i=1}^{n} X_i \cdot \frac{\partial T_i}{\partial t} \right. \\ &\quad \left. + a \sum_{i=1}^{n} \Delta X_i \cdot T_i - \sum_{i=1}^{n} (\mathbf{v} \cdot \nabla X_i) \cdot T_i \right) d\mathbf{x}\, dt \end{aligned} \tag{12}$$

We apply an alternating directions fixed point algorithm to compute the couple of functions (R, S):

- *Computing the function $S(\mathbf{x})$.*
  First, we suppose that R is known, implying that $R^\star$ vanishes in (11). Thus, (12) writes

$$\begin{aligned} &\int_{\Omega} S^\star \cdot (\alpha_t S - a\beta_t \Delta S + \beta_t\, \mathbf{v} \cdot \nabla S)\, d\mathbf{x} \\ &= \int_{\Omega} S^\star \cdot \left(\gamma_t(\mathbf{x}) - \sum_{i=1}^{n} \alpha_t^i X_i + a \sum_{i=1}^{n} \beta_t^i \Delta X_i \right. \\ &\quad \left. - \sum_{i=1}^{n} \beta_t^i\, \mathbf{v} \cdot \nabla X_i \right) d\mathbf{x} \end{aligned} \tag{13}$$

where

$$\begin{cases} \alpha_t = \displaystyle\int_0^{T_{max}} R(t) \cdot \frac{\partial R}{\partial t}(t)\, dt \\ \alpha_t^i = \displaystyle\int_0^{T_{max}} R(t) \cdot \frac{\partial T_i}{\partial t}(t)\, dt \\ \beta_t = \displaystyle\int_0^{T_{max}} R^2(t)\, dt \\ \beta_t^i = \displaystyle\int_0^{T_{max}} R(t) \cdot T_i(t)\, dt \\ \gamma_t(\mathbf{x}) = \displaystyle\int_0^{T_{max}} R(t) \cdot f(\mathbf{x}, t)\, dt; \quad \forall \mathbf{x} \in \Omega \end{cases} \tag{14}$$

The weak formulation (13) is satisfied for all $S^\star$, therefore we could come back to the associated strong formulation

$$\begin{aligned} &\alpha_t S - a\beta_t \Delta S + \beta_t\, \mathbf{v} \cdot \nabla S \\ &= \gamma_t - \sum_{i=1}^{n} \alpha_t^i X_i + a \sum_{i=1}^{n} \beta_t^i \Delta X_i \\ &\quad - \sum_{i=1}^{n} \beta_t^i\, \mathbf{v} \cdot \nabla X_i \end{aligned} \tag{15}$$

that one could solve by using any appropriate discretization technique.

- *Computing the function $R(t)$.*
  From the function $S(\mathbf{x})$ just computed, we search $R(t)$. In this case $S^\star$ vanishes in (11) and (12) reduces to

$$\begin{aligned} &\int_0^{T_{max}} \int_{\Omega} (S \cdot R^\star) \\ &\quad \cdot \left(S \cdot \frac{\partial R}{\partial t} - a\Delta S \cdot R + (\mathbf{v} \cdot \nabla S) \cdot R \right) d\mathbf{x}\, dt \\ &= \int_0^{T_{max}} \int_{\Omega} (S \cdot R^\star) \cdot \left(f(x, t) - \sum_{i=1}^{n} X_i \cdot \frac{\partial T_i}{\partial t} \right. \\ &\quad \left. + a \sum_{i=1}^{n} \Delta X_i \cdot T_i - \sum_{i=1}^{n} (\mathbf{v} \cdot \nabla X_i) \cdot T_i \right) d\mathbf{x}\, dt \end{aligned} \tag{16}$$

where all the spatial functions can be integrated in Ω. Thus, by using the following notations

$$\begin{cases} \alpha_x = \int_\Omega S(\mathbf{x}) \cdot \Delta S(\mathbf{x})\, d\mathbf{x} \\ \alpha_x^i = \int_\Omega S(\mathbf{x}) \cdot \Delta X_i(\mathbf{x})\, d\mathbf{x} \\ \beta_x = \int_\Omega S^2(\mathbf{x})\, d\mathbf{x} \\ \beta_x^i = \int_\Omega S(\mathbf{x}) \cdot X_i(\mathbf{x})\, d\mathbf{x} \\ \lambda_x = \int_\Omega S(\mathbf{x}) \cdot (\mathbf{v} \cdot \nabla S(\mathbf{x}))\, d\mathbf{x} \\ \lambda_x^i = \int_\Omega S(\mathbf{x}) \cdot (\mathbf{v} \cdot \nabla X_i(\mathbf{x}))\, d\mathbf{x}, \\ \gamma_x(t) = \int_\Omega S(\mathbf{x}) \cdot f(\mathbf{x}, t)\, d\mathbf{x}; \quad \forall t \end{cases} \tag{17}$$

(16) reads

$$\int_0^{T_{max}} R^\star \cdot \left(\beta_x \frac{\partial R}{\partial t} + (\lambda_x - a\alpha_x) R - \gamma_x(t) + \sum_{i=1}^n \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^n (\lambda_x^i - a\alpha_x^i) T_i \right) dt = 0 \tag{18}$$

As (18) holds for all $S^\star$, we could come back to the strong formulation

$$\beta_x \frac{\partial R}{\partial t} = (a\, \alpha_x - \lambda_x)\, R + \gamma_x(t) - \sum_{i=1}^n \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^n (a\, \alpha_x^i - \lambda_x^i) T_i \tag{19}$$

which is a first order ordinary differential equation that can be solved easily (even for extremely small time steps) from its initial condition.

These two steps must be repeated until convergence, that is, until verifying that both functions reach a fixed point. If we denote by $R^{(q)}(t)$ and $R^{(q-1)}(t)$ the computed functions $R(t)$ at the present and previous iteration respectively, and the same for the space functions: $S^{(q)}(\mathbf{x})$ and $S^{(q-1)}(\mathbf{x})$, the stoping criterion used in this work writes:

$$e = \left\| R^{(q)}(t) \cdot S^{(q)}(\mathbf{x}) - R^{(q-1)}(t) \cdot S^{(q-1)}(\mathbf{x}) \right\|_2 < 10^{-8} \tag{20}$$

where 10^{-8} represents the square root of the machine precision.

We denote by Q_{n+1} the number of iterations for solving this non-linear problem to determine the enrichment couple of functions $X_{n+1}(\mathbf{x})$ and $T_{n+1}(t)$. After reaching convergence we write $X_{n+1}(\mathbf{x}) = S(\mathbf{x})$ and $T_{n+1}(t) = R(t)$. The enrichment procedure must continue until reaching the convergence of the enrichment global procedure at iteration N, when the separated representation of the unknown field writes:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^N X_i(\mathbf{x}) \cdot T_i(t) \tag{21}$$

The more usual global stopping criteria are:

- For models whose exact solution u^{ref} is known:

$$E = \frac{\|u - u^{ref}\|_2}{\|u^{ref}\|_2} < \epsilon \tag{22}$$

- For models whose exact solution is not known:

$$E = \frac{\| \frac{\partial u}{\partial t} - a\Delta u + \mathbf{v} \cdot \nabla u - f(\mathbf{x}, t) \|_2}{\| f(\mathbf{x}, t) \|_2} < \epsilon \tag{23}$$

with ϵ a small enough parameter ($\epsilon = 10^{-8}$ in our simulations and the L_2-norm applies in the whole space-time domain).

Discussion on the Space-Time Separated Representations The just proposed strategy needs for the solution of about $N \times Q$ space and time problems (with $Q = (Q_1 + \cdots + Q_N)/N$ and N the number of functional couples needed to approximate, up to the desired precision, the searched solution). Thus, one must compute $N \times Q$ d-D problems, $d = 1, 2, 3$, whose complexity depends on the spatial mesh considered, and also $N \times Q$ 1D problems (defined in the time interval $\mathcal{I}$) that only need the solution of an ordinary differential equation from its initial condition. Obviously, even for extremely small time steps, the solution of these transient 1D problems does not introduce any difficulty.

If instead the separated representation just discussed, one performs a standard incremental solution, P dD models, $d = 1, 2, 3$, must be solved (P being the number of time steps, i.e. $P = T_{max}/\Delta t$, where the time step Δt must be chosen for verifying the stability conditions).

In all the analyzed cases N and Q are of the order of tens that implies the solution of about hundred three-dimensional problems defined in Ω, instead the thousands (or even millions) needed for solving those models using standard incremental solvers.

A first comparison between both kind of approaches (the one based on the separated representation and the one based on standard incremental strategies) was presented in [6].

3 Recent Advances

3.1 Some Advances in Kinetic Theory, Quantum Chemistry and Cell Signaling

3.1.1 Solving the Fokker-Planck Equation

The first works focusing in the solution of multidimensional models by applying the PGD described in the previous section concerned the modeling of polymeric liquids. These fluids are composed of an unimaginable number of molecules, that are modeled as a series of dumbbells (on which the different forces arising from the flow drag, interactions, solvent molecules bombardment, … apply) connected by linear or non-linear springs that represent the stiffness between each two consecutive dumbbells. Let $\mathbf{q}_1, \ldots, \mathbf{q}_D$ be the vectors defining the orientation and extension of each one of these springs. A kinetic theory approach of the molecular nature of such fluids consists of introducing a distribution function $\Psi(\mathbf{x}, t, \mathbf{q}_1, \ldots, \mathbf{q}_D)$ that represents the fraction of molecules that at position $\mathbf{x}$ and time t have a conformation given by vectors $\mathbf{q}_1, \ldots, \mathbf{q}_D$. As $\mathbf{x} \in \Omega_x \subset \mathbb{R}^3$, $t \in \mathcal{I} \equiv \mathbb{R}^+$ and $\mathbf{q}_i \in \Omega_q \subset \mathbb{R}^3, \forall i \in [1, \ldots, D]$, the distribution function is then defined in a space of dimension $3 + 1 + 3 \times D$, i.e. $\Psi : \Omega_x \times \Omega_t \times (\Omega_q)^D \rightarrow \mathbb{R}^+$.

Remark 2 In the case of linear springs $\Omega_q = \mathbb{R}^3$, for non-linear springs of finite extension (let b be the maximum spring extension) Ω_q reduces to the ball of radius b, $\mathcal{B}(0, b)$. Finally, in the case of rigid rods of unit length, Ω_q reduces to the surface of the unit sphere $\mathcal{S}(0, 1)$.

The evolution of the molecular configuration $\dot{\mathbf{q}}_i$, $\forall i$, is due to both advective and diffusive terms, the first ones arising from the presence of a prescribed flow with non-zero gradient of velocities. The system should verify the momentum balance at any time, i.e. the equilibrium of forces in absence of inertia terms. For the particular expressions of $\dot{\mathbf{q}}_i$ the interested reader can refer to [11] that constitutes a huge catalogue of physical systems.

The balance equation for the distribution function (also known as the Fokker-Planck equation) writes:

$$\frac{D\Psi}{Dt} = -\sum_{i=1}^{i=D} \frac{\partial}{\partial \mathbf{q}_i} (\dot{\mathbf{q}}_i \cdot \Psi) \tag{24}$$

where $\frac{D}{Dt}$ denotes the material derivative.

The solution of this multidimensional equation by applying the PGD assumes a separated representation of the distribution function

$$\Psi(\mathbf{x}, t, \mathbf{q}_1, \ldots, \mathbf{q}_D) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot \left(\prod_{j=1}^{j=D} Q_i^j(\mathbf{q}_j) \right) \tag{25}$$

The case of steady-state multi-bead-finite-extension-spring models of polymeric liquids in homogeneous flows was addressed in [3]. Later, we considered the transient case, in which the time was introduced as an additional coordinate being careful on its upwinding discretization [4]. The case of short fiber suspensions in which the beads connector is inextensible was addressed in [40]. All these models are linear even when the spring behavior is strongly non-linear (the resulting Fokker-Planck equations (24) are linear with respect to the unknown function Ψ).

In more complex systems like liquid crystalline polymers or entangled polymeric systems involving double reptation and convective constraint release, as the one proposed in [23], the resulting models are no more linear and a linearization procedure is compulsory. In [34] we considered the simplest one, a fixed point strategy for solving some models associated with entangled polymeric systems.

Finally the case of non-homogeneous flows was deeply analyzed in [40] and [35]. The main issue in treating non-homogenous flows using the fully separated representation (25) including physical space, time and conformation spaces lies in the convective stabilization issues, because the stabilization in the conformation space depends on the gradient of velocities that is defined in the physical space. Thus, the adequate separated representation of the stabilization term remains an open issue.

In [17] we addressed the difficulty related to the reduced modeling of systems composed of moving particles within the framework of Brownian dynamics simulations. We proved that classical model reduction techniques based on the use of the proper orthogonal decomposition fails, and that POD or PGD based discretizations perfectly run as soon as the models are rewritten as Brownian configuration fields. Thus, the issue related to updated Lagrangian simulations (the reference configuration is changing in time)—of capital importance in material forming simulations, finite transformations thermomechanic or molecular dynamics—moved to the first plane. We come back to this challenging issue later.

3.1.2 Solving the Schrödinger Equation

After treating kinetic theory models we were interested in models arising from quantum chemistry. These models are redoubtable because the equations that govern the electronic distribution, the Schrödinger equation or its fully relativistic counterpart—the Dirac's equation—are defined in spaces whose dimensionality scales with the number of elementary particles involved in the quantum system. The first one, within the Born-Oppenheimer hypothesis (the wavefunction depends parametrically on the nuclei positions), writes:

$$i\hbar \frac{\partial \Psi}{\partial t} = -\frac{\hbar^2}{2m_e} \sum_{e=1}^{e=D_e} \nabla_e^2 \Psi + \sum_{e=1}^{e=D_e-1} \sum_{e'=e+1}^{e'=D_e} V_{ee'} \Psi$$

$$+\sum_{e=1}^{e=D_e}\sum_{n=1}^{n=D_n} V_{en}\Psi \tag{26}$$

where $i=\sqrt{-1}$, $\hbar$ is the Planck's constant divided by 2π, m_e is the electron mass, D_e and D_n are the number of electrons and nuclei in the system respectively, $\Psi(\mathbf{x}_1,\ldots,\mathbf{x}_{D_e},t;\mathbf{X}_1,\ldots,\mathbf{X}_{D_n})$ is the wavefunction, being $\mathbf{x}_e$ the space in which the electron e lives ($\mathbf{x}_e\in\mathbb{R}^3$), and $\mathbf{X}_n$ the position of nuclei n. The differential operator ∇_e refers to the gradient with respect to the coordinates $\mathbf{x}_e$. Finally $V_{ee'}$ and V_{en} represent the electron-electron and electron-nuclei Coulomb potentials respectively.

The Schrödinger and Dirac equations are defined in a space that scales with the number of elementary particles involved in the system, each one living in $\mathbb{R}^3$. Thus, when one considers few particles D, the time-independent Schrödinger equation is defined in a space of dimension $3\times D$. However, we have proven in [19] that in many systems the real issue is not the model multi-dimensionality, that could be circumvented by using the PGD, but the Pauli's exclusion principle that enforces the anti-symmetry of the wavefunction. For enforcing this constraint, the use of Slater's determinants is the simplest alternative, but the number of terms that these determinants involve explodes as the number of concerned particles grows. The interested reader can refer to [5, 19] for more details concerning the treatment of quantum systems by using the PGD.

3.1.3 Solving the Chemical Master Equation

We also considered recently in [7] the treatment of the chemical master equation, modeling, among many other physics, the cell signaling. In chemical systems involving some (or several) species each one composed by a few number of individuals, the state of the system is given by a probability (the deterministic approach fails when the number of individuals is too reduced for introducing the concept of chemical concentration): $P(\mathbf{z},t|\mathbf{z}_0,t_0)$ where $\mathbf{z}=(\#s_1,\ldots,\#s_D)$ and $\#s_i$ refers to the number of individuals of each species s_i. Obviously $\#s_i$ is changing in time because the different chemical reactions that are producing or destroying individuals in the population s_i. The chemical master equation writes:

$$\frac{\partial P(\mathbf{z},t|\mathbf{z}_0,t_0)}{\partial t}=\sum_j\big(a_j(\mathbf{z}-\mathbf{v}_j)P(\mathbf{z}-\mathbf{v}_j,t|\mathbf{z}_0,t_0)$$
$$-a_j(\mathbf{z})P(\mathbf{z},t|\mathbf{z}_0,t_0)\big) \tag{27}$$

where a_j are the propensities (the chemical reaction j operates growing the population $\mathbf{z}$ from $\mathbf{z}-\mathbf{v}_j$: $\mathbf{z}-\mathbf{v}_j\rightarrow\mathbf{z}$, and simultaneously population $\mathbf{z}$ decreases originating individuals of other populations).

Despite the fact that this equation does not imply partial derivatives other that the temporal one, we assumed a separated representation

$$P(\mathbf{z},t)\approx\sum_{i=1}^{i=N}T_i(t)\cdot\left(\prod_{j=1}^{j=D}S_i^j(\#sj)\right) \tag{28}$$

and we proceed in a standard manner as described in [7].

3.2 Further Considerations on Non-linear Models

For illustrating the treatment of non linear models we are treating the following simple nonlinear parabolic problem, that we considered in [6]

$$\begin{cases}\frac{\partial u}{\partial t}-a\Delta u=u^2+f(\mathbf{x},t) & \text{in } \Omega\times(0,T_{max}]\\ u(\mathbf{x},t)=0 & \text{on } \partial\Omega\times(0,T_{max}]\\ u(\mathbf{x},0)=0 & \text{in } \Omega\end{cases} \tag{29}$$

where $\Omega\subset\mathbb{R}^d$, $d\geq 1$, $T_{max}>0$ and $a>0$ is the diffusion coefficient. To build-up the approximated solution of (29) by using a separated representation, we propose two alternatives

- An incremental linearization
- A Newton linearization

which we describe in sections below.

3.2.1 Incremental Linearization

We look for writing the solution of problem (29) 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 $R(t)\cdot S(\mathbf{x})$ such that the updated approximation writes

$$u(\mathbf{x},t)\approx\sum_{i=1}^{n}X_i(\mathbf{x})\cdot T_i(t)+S(\mathbf{x})\cdot R(t) \tag{30}$$

The weak form of problem (29) writes

$$\int_0^{T_{max}}\int_\Omega u^\star\left(\frac{\partial u}{\partial t}-a\Delta u-u^2-f(\mathbf{x},t)\right)d\mathbf{x}\,dt=0;\quad \forall u^\star \tag{31}$$

The alternating directions scheme proceed by calculating $S(\mathbf{x})$ from the temporal function $R(t)$ just computed, and then, updating $R(t)$ from the just computed $S(\mathbf{x})$, as we described in Sect. 2. 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

$$u^2 \approx \left(\sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 \tag{32}$$

Thus, we can compute $S(\mathbf{x})$ from

$$\int_{\Omega} S^\star \cdot \left(\alpha_t S - a\beta_t \Delta S + \sum_{i=1}^{n} \alpha_t^i X_i - a \sum_{i=1}^{n} \beta_t^i \Delta X_i - \Phi(\mathbf{x}) - \gamma_t(\mathbf{x}) \right) d\mathbf{x} = 0 \tag{33}$$

where the notation introduced in Sect. 2 is used again, and where $\Phi(\mathbf{x})$ and $\Gamma(\mathbf{x})$ are given by

$$\Phi(\mathbf{x}) = \int_0^{T_{max}} R(t) \cdot \left(\sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 dt \tag{34}$$

The associated strong form writes

$$\alpha_t S - a\beta_t \Delta S = -\sum_{i=1}^{n} \alpha_t^i X_i + a \sum_{i=1}^{n} \beta_t^i \Delta X_i + \Phi(\mathbf{x}) + \gamma_t(\mathbf{x}) \tag{35}$$

From this solution $S(\mathbf{x})$, we can update the temporal function $R(t)$, by solving its associated strong form

$$\beta_x \frac{\partial R}{\partial t} - a\alpha_x R + \sum_{i=1}^{n} \beta_x^i \frac{\partial T_i}{\partial t} - a \sum_{i=1}^{n} \alpha_x^i T_i - \Psi(t) - \gamma_x(t) = 0 \tag{36}$$

where

$$\Psi(t) = \int_{\Omega} S(\mathbf{x}) \cdot \left(\sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 d\mathbf{x} \tag{37}$$

3.2.2 Newton Linearization

From now on we denote by u^n the solution computed at iteration n, i.e.

$$u^n(\mathbf{x}, t) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \tag{38}$$

Now, after linearization, the solution at the next iteration can be written as $u^{n+1} = u^n + \widetilde{u}$ where $\widetilde{u}$ is the solution of the problem

$$\frac{\partial \widetilde{u}}{\partial t} - a\Delta \widetilde{u} - 2u^n \widetilde{u} = -\left(\frac{\partial u^n}{\partial t} - a\Delta u^n - (u^n)^2 - f(\mathbf{x}, t) \right) \tag{39}$$

whose weak formulation writes:

$$\int_0^{T_{max}} \int_{\Omega} \widetilde{u}^\star \left(\frac{\partial \widetilde{u}}{\partial t} - a\Delta \widetilde{u} - 2u^n \widetilde{u} \right) d\mathbf{x}\, dt = \int_0^{T_{max}} \int_{\Omega} \widetilde{u}^\star \left(-\frac{\partial u^n}{\partial t} + a\Delta u^n + (u^n)^2 + f(\mathbf{x}, t) \right) d\mathbf{x}\, dt, \quad \forall \widetilde{u}^\star \tag{40}$$

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

3.2.3 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 some linear models, if we solve the problem and then apply the POD (for a given precision) we obtain

$$u^{POD,N}(\mathbf{x}, t) = \sum_{i=1}^{N} X_i^{POD}(\mathbf{x}) \cdot T_i^{POG}(t) \tag{41}$$

When the numerical solution is computed using the PGD described in Sect. 2, the solution obtained using N sums $u^{PGD,N}$

$$u^{PGD,N}(\mathbf{x}, t) = \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t) \tag{42}$$

is very close to $u^{POD,N}$ even if the space functions X_i^{POD} and X_i, and the time functions T_i^{POD} and T_i are very different.

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.

$$u^{ex}(\mathbf{x}, t) = X^{ex}(\mathbf{x}) \cdot T^{ex}(t) \tag{43}$$

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

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t) \tag{44}$$

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 [42] we analyzed many 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 (the one for computing the couple $R(t)$ and $S(\mathbf{x})$) according to:

$$u^2 \approx \left(\sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) + R^{(q-1)}(t) \cdot S^{(q-1)}(\mathbf{x}) \right)^2 \tag{45}$$

then we proved, in the case described above, that the solver converges after computing the first functional couple. In that sense the solver is optimal (only one couple was required for representing the solution that contains a single functional couple) but the computing time is similar to the one required by using the standard fixed point or the Newton strategy previously described.

3.3 PGD Tensor Form

The procedure described in section 2 can be generalized by using a tensor notation [8, 42].

We assume that the discrete problem, that we writes formally as $\mathcal{U}^{*T}\mathcal{A}\mathcal{U} = \mathcal{U}^{*T}\mathcal{B}$, can be written in a separated form:

$$\begin{aligned}
\mathcal{A} &= \sum_{i=1}^{n_A} \mathbf{A}_1^i \otimes \mathbf{A}_2^i \otimes \cdots \otimes \mathbf{A}_D^i \\
\mathcal{B} &= \sum_{i=1}^{n_B} \mathbf{B}_1^i \otimes \mathbf{B}_2^i \otimes \cdots \otimes \mathbf{B}_D^i \\
\mathcal{U} &\approx \sum_{i=1}^{N} \mathbf{u}_1^i \otimes \mathbf{u}_2^i \otimes \cdots \otimes \mathbf{u}_D^i
\end{aligned} \tag{46}$$

where $\mathbf{A}_i$, $\mathbf{B}_i$ and $\mathbf{u}_i$ involve only the coordinate $\mathbf{x}_i$.

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

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

$$\mathcal{U} = \sum_{i=1}^{n} \mathbf{u}_1^i \otimes \cdots \otimes \mathbf{u}_D^i + \mathbf{R}_1 \otimes \cdots \otimes \mathbf{R}_D \tag{47}$$

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

$$\mathcal{U}^* = \mathbf{R}_1^* \otimes \mathbf{R}_2 \otimes \cdots \otimes \mathbf{R}_D + \cdots + \mathbf{R}_1 \otimes \cdots \otimes \mathbf{R}_{D-1} \otimes \mathbf{R}_D^* \tag{48}$$

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

$$\begin{aligned}
&\sum_{i=1}^{n_A}\sum_{j=1}^{n} (\mathbf{R}_1^*)^T \mathbf{A}_1^i \mathbf{u}_1^j \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{u}_D^j + \cdots \\
&\quad + \sum_{i=1}^{n_A}\sum_{j=1}^{n} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{u}_1^j \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{A}_D^i \mathbf{u}_D^j \\
&\quad + \sum_{i=1}^{n_A} (\mathbf{R}_1^*)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{R}_D + \cdots \\
&\quad + \sum_{i=1}^{n_A} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{A}_D^i \mathbf{R}_D \\
&\quad = \sum_{i=1}^{n_B} \Big((\mathbf{R}_1^*)^T \mathbf{B}_1^i \times \cdots \times (\mathbf{R}_D)^T \mathbf{B}_D^i + \cdots \\
&\quad + (\mathbf{R}_1)^T \mathbf{B}_1^i \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{B}_D^i \Big)
\end{aligned} \tag{49}$$

For alleviating the notation we define:

$$\begin{aligned}
\sum_{i=1}^{n_C} \mathbf{C}_1^i \otimes \cdots \otimes \mathbf{C}_D^i = \sum_{i=1}^{n_B} \mathbf{B}_1^i \otimes \cdots \otimes \mathbf{B}_D^i \\
- \sum_{i=1}^{n_A}\sum_{j=1}^{n} \mathbf{A}_1^i \mathbf{u}_1^j \otimes \cdots \otimes \mathbf{A}_D^i \mathbf{u}_D^j
\end{aligned} \tag{50}$$

where $n_C = n_B + n_A \times n$. This sum only contains known fields. Thus (49) can be written as:

$$\begin{aligned}
&\sum_{i=1}^{n_A} (\mathbf{R}_1^*)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_D)^T \mathbf{A}_D^i \mathbf{R}_D + \cdots \\
&\quad + \sum_{i=1}^{n_A} (\mathbf{R}_1)^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{A}_D^i \mathbf{R}_D \\
&\quad = \sum_{i=1}^{n_C} \Big((\mathbf{R}_1^*)^T \mathbf{C}_1^i \times \cdots \times (\mathbf{R}_D)^T \mathbf{C}_D^i + \cdots \\
&\quad + (\mathbf{R}_1)^T \mathbf{C}_1^i \times \cdots \times (\mathbf{R}_D^*)^T \mathbf{C}_D^i \Big)
\end{aligned} \tag{51}$$

This problem is strongly non linear. To solve it, a method of alternated directions can be applied. The idea is, starting with the trial vectors $\mathbf{R}_i^{(0)}$, $i = 1, \ldots, D$ or assuming known these vector at iteration $r-1$, $\mathbf{R}_i^{(r-1)}$, $i = 1, \ldots, D$, update them using an appropriate strategy. The simplest alternatives consist of:

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

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 $\mathbf{u}_i^{n+1} = R_i$, $i = 1, \ldots, D$.

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

$$\mathcal{U}^{*T} = \mathbf{R}_1 \otimes \cdots \otimes \mathbf{R}_{k-1} \otimes \mathbf{R}_k^* \otimes \mathbf{R}_{k+1} \cdots \otimes \mathbf{R}_D \tag{52}$$

The resulting discrete weak form writes:

$$\sum_{i=1}^{n_A} \left(\mathbf{R}_1^T \mathbf{A}_1^i \mathbf{R}_1 \times \cdots \times \mathbf{R}_k^{*T} \mathbf{A}_k^i \mathbf{R}_k \times \cdots \times \mathbf{R}_D^T \mathbf{A}_D^i \mathbf{R}_D \right) = \sum_{i=1}^{n_C} \mathbf{R}_1^T \mathbf{C}_1^i \times \cdots \times \mathbf{R}_k^{*T} \mathbf{C}_k^i \times \cdots \times \mathbf{R}_D^T \mathbf{C}_D^i \tag{53}$$

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

$$\left(\sum_{i=1}^{n_A} \left(\prod_{j=1, j\neq k}^{D} \mathbf{R}_j^T \mathbf{A}_j^i \mathbf{R}_j \right) \mathbf{A}_k^i \right) \mathbf{R}_k = \sum_{i=1}^{n_C} \left(\prod_{j=1, j\neq k}^{D} \mathbf{R}_j^T \mathbf{C}_j^i \right) \mathbf{C}_k^i \tag{54}$$

which can be easily solved.

3.4 Decomposition Optimality: Residual Minimization

The alternating direction strategy that we used in Sect. 2 as well as in Sect. 3.3 converges very fast in the case of symmetric differential operators. However, when we include strong non-linearities, advection terms, parameters as additional coordinates, … the alternating directions strategy fails (the computed functional couples do not improve noticeably the solution). As Pierre Ladeveze proposed many years ago in the framework of the LATIN method, in that case a more efficient strategy consists of minimizing the residual [42]:

$$\text{Res} = \sum_{i=1}^{n_A} \mathbf{A}_1^i \mathbf{R}_1 \otimes \cdots \otimes \mathbf{A}_D^i \mathbf{R}_D - \sum_{i=1}^{n_C} \mathbf{C}_1^i \otimes \cdots \otimes \mathbf{C}_D^i \tag{55}$$

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

$$\begin{aligned}\|\text{Res}\|^2 = &\sum_{i=1}^{n_A} \sum_{j=1}^{n_A} \left(\left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{A}_1^j \mathbf{R}_1 \right\rangle \times \cdots \times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{A}_D^j \mathbf{R}_D \right\rangle \right) \\ &- 2 \sum_{i=1}^{n_A} \sum_{j=1}^{n_C} \left(\left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{C}_1^j \right\rangle \times \cdots \times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{C}_D^j \right\rangle \right) \\ &+ \sum_{i=1}^{n_C} \sum_{j=1}^{n_C} \left(\left\langle \mathbf{C}_1^i, \mathbf{C}_1^j \right\rangle \times \cdots \times \left\langle \mathbf{C}_D^i, \mathbf{C}_D^j \right\rangle \right)\end{aligned} \tag{56}$$

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

$$\frac{\partial}{\partial \mathbf{R}_k} \langle \text{Res}, \text{Res} \rangle = \mathbf{0} \tag{57}$$

or:

$$\begin{aligned}&\sum_{i=1}^{n_A} \sum_{j=1}^{n_A} \left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{A}_1^j \mathbf{R}_1 \right\rangle \times \cdots \times \left\langle \mathbf{A}_{k-1}^i \mathbf{R}_{k-1}, \mathbf{A}_{k-1}^j \mathbf{R}_{k-1} \right\rangle \\ &\times \left\langle \mathbf{A}_k^i, \mathbf{A}_k^j \mathbf{R}_k \right\rangle \times \left\langle \mathbf{A}_{k+1}^i \mathbf{R}_{k+1}, \mathbf{A}_{k+1}^j \times \mathbf{R}_{k+1} \right\rangle \times \cdots \\ &\times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{A}_D^j \mathbf{R}_D \right\rangle \\ &- \sum_{i=1}^{n_A} \sum_{j=1}^{n_C} \left\langle \mathbf{A}_1^i \mathbf{R}_1, \mathbf{C}_1^j \right\rangle \times \cdots \times \left\langle \mathbf{A}_{k-1}^i \mathbf{R}_{k-1}, \mathbf{C}_{k-1}^j \right\rangle \\ &\times \left\langle \mathbf{A}_k^i, \mathbf{C}_k^j \right\rangle \times \left\langle \mathbf{A}_{k+1}^i \mathbf{R}_{k+1}, \mathbf{C}_{k+1}^j \right\rangle \times \cdots \\ &\times \left\langle \mathbf{A}_D^i \mathbf{R}_D, \mathbf{C}_D^j \right\rangle = \mathbf{0}\end{aligned} \tag{58}$$

The stoping criterion is defined from the residual norm:

$$\left\| \sum_{i=1}^{n_C} \mathbf{C}_1^i \otimes \cdots \otimes \mathbf{C}_D^i \right\|_2 < \epsilon \tag{59}$$

Despite the fact that this strategy is much more efficient than the alternating directions strategy for non-symmetric operators, it is far to be optimal. Thus, in some particular cases, even if the exact solution accepts a separated representation, the PGD coupled with the residual minimization produces a solution that contains much more functional products that the exact one.

We are at present exploring other strategies, based on concepts of information theory (maximum entropy, …). Other possibility consists in looking at each iteration for some functional products (instead of a single one at each iteration). This strategy improves the convergence rate reducing the number of sums in the computed decomposition, but at present no strategy allows for a general optimal decomposition.

The reason of this lack of optimality was understood by performing a numerical analysis (see [21] in the present issue). However, when we tried to enforce the enrichment maximizing the residual diminution the numerical procedure exhibits locking.

Previously we identified two major issues of the PGD, we cited the convective stabilization and the treatment of models defined in moving domains, now we should add a third issue that concerns the decomposition optimality.

3.5 Non Homogeneous Essential Boundary Conditions and Complex Domains

In [25] we addressed another different topic, the one related to the enforcement of non-homogeneous boundary conditions and the treatment of complex geometries.

3.5.1 Non Homogeneous Essential Boundary Conditions

The enforcement of homogeneous boundary conditions is quite simple, it suffices to enforce at each enrichment stage that the first and last components of vectors $\mathbf{R}_i$, $i = 1, \ldots, D$, vanish. However, consider the 3D problem:

$$-\Delta u = f(\mathbf{x}) \tag{60}$$

where $\mathbf{x} = (x, y, z) \in \Omega = (0, L)^3$, with

$$u(\mathbf{x}) = u_g \neq 0, \quad \mathbf{x} \in \Gamma \equiv \partial\Omega \tag{61}$$

In [25] we proposed to determine a function ψ, regular enough (continuous in $\overline{\Omega}$ and $\Delta\psi \in L_2(\Omega)$) such that ψ verifies the essential boundary condition, i.e. $\psi(\mathbf{x}) = u_g$, $\mathbf{x} \in \Gamma$. This function can be constructed by using the transfinite interpolation method [45].

Now, by applying the change of variable $u = \psi + \Upsilon$, the problem (60) writes:

$$-\Delta\Upsilon = f(\mathbf{x}) + \Delta\psi \tag{62}$$

where $\Upsilon(\mathbf{x}) = 0$, $\mathbf{x} \in \Gamma$. Thus, the solution of problem (62) reduces to the ones that we considered previously subjected to homogenous essential boundary conditions, where the unknown field is searched in the separated form:

$$\Upsilon \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \tag{63}$$

The solution procedure simplifies if function ψ is given in a separated form, i.e.

$$\psi \approx \sum_{i=1}^{i=M} F_i(x) \cdot G_i(y) \cdot H_i(z) \tag{64}$$

This decomposition can be efficiently performed by invoking the SVD or its multidimensional counterpart.

3.5.2 Complex Domains

In [25] we also addressed the question of complex domains. Until now, the whole domain Ω was considered as the direct product of the domains in which each coordinate is defined, $\Omega = \Omega_1 \times \cdots \times \Omega_D$, i.e. Ω_i, $\forall i$, does not depend on the other coordinates $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_D$.

Due to this difficulty when we considered in Sect. 3.1.1 models defined in the physical ($\mathbf{x}$) and conformation spaces ($\mathbf{q}_1, \ldots, \mathbf{q}_D$) we considered the decomposition [3]

$$u(\mathbf{x}, \mathbf{q}_1, \ldots, \mathbf{q}_D) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot \left(\prod_{j=1}^{j=D} Q_i^j(\mathbf{q}_j) \right) \tag{65}$$

where $\mathbf{x} \in \Omega \subset \mathbb{R}^3$. Thus, this decomposition is not affected at all by the complexity of Ω.

However, a full decomposition of the physical space would result in:

$$u(\mathbf{x}, \mathbf{q}_1, \ldots, \mathbf{q}_D) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \cdot \left(\prod_{j=1}^{j=D} Q_i^j(\mathbf{q}_j) \right) \tag{66}$$

that can be used without major difficulties if Ω is a parallelepiped, i.e. $\Omega = \Omega_x \times \Omega_y \times \Omega_z$. However, when it is not the case, a special treatment is required.

We considered again the problem (60) now with homogeneous boundary conditions on $\partial\Omega \equiv \Gamma$, with Ω quite complex in the sense discussed above.

A fictitious domain technique is considered. Thus, the real domain Ω is placed inside a larger domain ω ($\Omega \subset \omega$ and $\overline{\Omega} \cap \overline{\varpi} = \emptyset$), with $\omega = \omega_x \times \omega_y \times \omega_z$.

Thus, we must solve the problem

$$-\Delta u = f(\mathbf{x}) \quad \text{in } \omega \tag{67}$$

with the unknown field u extended to the domain ω by using $u = \Phi \cdot \Upsilon$. Function Φ is regular enough, defined in the extended domain ω and it vanishes on $\Gamma \equiv \partial\Omega$ in order to enforce the boundary conditions. In [25] we proposed the use of the R-functions [45] for computing function Φ than should be rewritten again in a separated form by invoking the SVD. Function Υ is searched again in the separated form

$$\Upsilon \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) \cdot Z_i(z) \tag{68}$$

3.6 Localization, Interfaces and FE Coupling

In this section we are modifying the technique just presented in order to capture localized behaviors such as high gradients or weak and strong discontinuities.

This kind of behaviors induces too many functional products within a standard separated representation. However, it is well known that they can be easily taken into account within the finite element framework, by using adaptive local mesh refinement in the first case or any of the techniques able to represent weak and strong discontinuities (meshes compatible with the interfaces or enriched approximations when the meshes are not compatible with the interfaces).

Thus, the main goal is how combining separated representations, able to approximate the solution in the most part of the domain where the model is defined, reducing significantly the computational cost, and a localized finite element description, that even if it is expensive in nature, it only applies in a small region of the whole domain where the solution is expected to exhibit localized behavior. To this end, we proposed in [9] a multi-scale approach that somewhat resembles the s-version of the finite element method by J. Fish [24] or the multi-scale FEM proposed by Rank [43], but in this case in a global (Ritz) and separated basis function setting.

For the sake of simplicity we consider a generic 2D problem

$$\mathcal{K}(u(x,y)) + \mathcal{L}(u(x,y)) = 0 \tag{69}$$

where $\mathcal{K}$ and $\mathcal{L}$ are two differential operators. We are assuming that the first one only involves derivatives with respect to the x-coordinate, the second one involving the derivatives with respect to the other coordinate.

This model is defined in $\Omega = \Omega_x \times \Omega_y$ and we imagine that the solution $u(x,y)$ is smooth enough everywhere except in a small region $\Omega_l \subset \Omega$. Now, we could imagine an approximation combining both, the separated representation and a finite element approximation, where the last one only applies in Ω_l. We assume a mesh on Ω_l composed of N_l nodes. We also assume that the finite element shape functions related to the finite element approximation cannot be expressed from the tensor product of the one dimensional bases employed to build-up the separated representation to avoid rank deficiency.

Now, the approximation in the whole domain can be written as:

$$u(x,y) = u^{SR}(x,y) + u^{FE}(x,y) \tag{70}$$

where $u^{SR}(x,y)$ is defined in the whole domain $\Omega = \Omega_x \times \Omega_y$ whereas the finite element enrichment $u^{FE}(x,y)$ is only defined within Ω_l. In order to ensure the continuity of the resulting approximation we must enforce the nullity of the enrichment $u^{FE}(x,y)$ on the boundary of Ω_l, $\partial\Omega_l$.

The resulting approximation writes:

$$u(x,y) \approx \sum_{i=1}^{i=N} X_i(x) \cdot Y_i(y) + \sum_{j=1}^{j=M} N_j(x,y) u_j \tag{71}$$

where $N_j(x,y)$ are the standard finite element shape functions and u_j the associated weights. Because of the contribution of $u^{SR}(x,y)$ in Ω_l, u_j does not correspond to the values of the unknown field $u(x,y)$ at the nodal positions (x_j, y_j).

Remark 3 We are assuming that the enrichment is performed by using standard finite elements, but in fact any kind of compact support approximation could be used. In order to represent interfaces involving weak discontinuities one could proceed within the standard finite element method by ensuring that the interface coincides with the elements edges. If the interface passes across the elements an enriched version of the finite element method should be considered (e.g. the extended finite element [49]).

In [9] we illustrated the application of this enriched separated approximation when model solutions exhibit localization or for gluing subdomains in which different separated representations are defined, ...

3.7 On the Coupling of Local and Global Models: The Local Problem Globalization

We are considering again the model given by (6) in absence of advection, i.e. $\mathbf{v} = \mathbf{0}$, and in a one-dimensional physical space Ω:

$$\frac{\partial u}{\partial t} - a\Delta u = f(\mathbf{x}, t) \quad \text{in } \Omega \times (0, T_{max}] \tag{72}$$

with the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x}, 0) = u^0, & \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) = u_g, & (\mathbf{x}, t) \in \partial\Omega \times (0, T_{max}] \end{cases} \tag{73}$$

We are assuming that the source term depends on the local value of r fields $C_i(t, \mathbf{x})$, $i = 1, \dots, r$:

$$f(\mathbf{x}, t) = \sum_{i=1}^{i=r} \gamma_i \cdot C_i(\mathbf{x}, t) \tag{74}$$

where the time evolution of the r fields $C_i(\mathbf{x}, t)$ is governed by r simultaneous ordinary differential equations (the so-called kinetic model). For the sake of simplicity we consider the linear case, the non-linear one reduces to a sequence of linear problems by applying an appropriate linearization strategy [6]. The system of linear ODEs writes at each point $\mathbf{x} \in \Omega$:

$$\frac{dC_i(\mathbf{x}, t)}{dt} = \sum_{j=1}^{j=r} \alpha_{ij}(\mathbf{x})\, C_j(\mathbf{x}, t) \tag{75}$$

We are assuming that the kinetic coefficients α_{ij} evolve smoothly in Ω, because in practical applications these coefficients depend on the solution of the diffusion problem,

$u(\mathbf{x},t)$. For the sake of simplicity and without loss of generality, from now on we assume those coefficients constant (they were assumed evolving linearly in the physical space in [20]).

Now, we are describing three possible coupled solution of (72) and (75).

1. The simplest strategy consists in using a separated representation of the global problem solution (72) whereas the local problems are integrated in the whole time interval at each nodal position (or integration point). Obviously, this strategy implies the solution of r ordinary differential equations at each node (or integration point). Moreover, the resulting fields $C_i(\mathbf{x},t)$, $r = 1,\dots,r$, don't have a separated structure, and by this reason before of injecting these fields into the global problem (72) we should separate them by invoking, for example, the singular value decomposition (SVD) leading to:

$$C_i(\mathbf{x},t) \approx \sum_{q=1}^{q=m} X_q^{C,i}(\mathbf{x}) \cdot T_q^{C,i}(t) \tag{76}$$

   As soon as the source term has a separated structure, the procedure illustrated in previous sections can be applied again for computing the new trial solution of the global problem that writes:

$$u(\mathbf{x},t) \approx \sum_{i=1}^{N} X_i^u(\mathbf{x}) \cdot T_i^u(t) \tag{77}$$

   Thus, this coupling strategy requires the solution of many local problems (for all species) and at all nodal positions (or integration points). Moreover, after these solutions (that we recall that could be performed in parallel) a singular value decomposition must be applied in order to separate these solutions prior to inject them in the PGD solver of the global problem (72).
2. The second coupling strategy lies in globalizing the solution of the local problems. Thus, we assume that the field related to each species can be written in a separated form:

$$C_i(\mathbf{x},t) \approx \sum_{q=1}^{q=m} X_q^{C,i}(\mathbf{x}) \cdot T_q^{C,i}(t) \tag{78}$$

   and now, we apply the procedure described in Sect. 2 to build-up the reduced separated approximation, i.e. for constructing all the functions involved in (78). Thus, instead of solving the r ODEs in (76) (that define r one-dimensional problems) at each nodal position (or integration point), we should solve only r higher dimensional coupled models defined in the physical space and time. Obviously, if the number of nodes (or integration points) is important (mainly when 3D physical spaces are considered) the present coupling strategy could offer significant CPU time savings.

   This strategy allows computing directly a separated representation, and then, with respect to the previous one, the application of the SVD is avoided. However, if the number of species is high, the computational efforts can become important, because the space-time separated solver must be apply to each species.
3. The third alternative, that in our opinion is the more appealing one for solving models involving many species, as large as one wants, implies the definition of a new variable $C(\mathbf{x},t,c)$, that as we can notice contains an extra coordinate c, with discrete nature, and that takes integer values: $c = 1,\dots,r$, in such manner that $C(\mathbf{x},t,i) \equiv C_i(\mathbf{x},t)$, $i = 1,\dots,r$. Thus, we have increased the dimensionality of the problem, but now, only a single problem should be solved, instead of one for each species as was the case when using the previous strategy. This increase of the model dimensionality is not dramatic because as argued in the first section of this work, the separated representation allows circumventing the curse of dimensionality, allowing for fast and accurate solutions of highly multidimensional models. Now, the issue is the derivation of the governing equation for this new variable $C(\mathbf{x},t,c)$ and the separated representation constructor able to define the approximation:

$$C(\mathbf{x},t,c) \approx \sum_{q=1}^{q=S} X_q^C(\mathbf{x}) \cdot T_q^C(t) \cdot A_q(c) \tag{79}$$

As this strategy was retained in our simulations in [20] we are focusing in its associated computational aspects in the next section.

3.7.1 Fully Globalized Local Models

The third strategy just referred implies the solution of a single multidimensional model involving the field $C(\mathbf{x},t,c)$. This original introduction deserves some additional comments. The first one concerns the discrete nature of the kinetic equations

$$\frac{dC_i(\mathbf{x},t)}{dt} = \sum_{j=1}^{j=r} \alpha_{ij}(\mathbf{x}) \cdot C_j(\mathbf{x},t) \tag{80}$$

Now, by introducing $C(\mathbf{x},t,c)$, such that $C(\mathbf{x},t,i) \equiv C_i(\mathbf{x},t)$, the kinetic equations could be written as:

$$\frac{dC}{dt} = \mathcal{L}_c(C) \tag{81}$$

where $\mathcal{L}_c$ is an operator in the c-coordinate.

If for one second we try to discretize (81) by finite differences, we could write at each node $(\mathbf{x}_k, t_p, i)$:

$$\frac{C(\mathbf{x}_k, t_p, i) - C(\mathbf{x}_k, t_{p-1}, i)}{\Delta t} = \mathcal{L}_c(C)|_i \tag{82}$$

where

$$\mathcal{L}_c(C)|_i = \sum_{j=1}^{j=r} \alpha_{ij} \cdot C(\mathbf{x}_k, t_p, j) \tag{83}$$

represents the discrete form of the c-operator at point i.

Now, we come back to the separated representation constructor for defining the approximation:

$$C(\mathbf{x}, t, c) \approx \sum_{q=1}^{q=S} X_q^C(\mathbf{x}) \cdot T_q^C(t) \cdot A_q(c) \tag{84}$$

For defining such approximation one should repeat the procedure deeply illustrated in Sect. 2. As the operator here involved is less standard we are summarizing the main steps.

We assume that the first n iterations allowed computing the first n sums of (84)

$$C(\mathbf{x}, t, c) \approx \sum_{q=1}^{q=n} X_q^C(\mathbf{x}) \cdot T_q^C(t) \cdot A_q(c) \tag{85}$$

and now, we look for the enrichment $R(\mathbf{x}) \cdot S(t) \cdot W(c)$, such that

$$\begin{aligned} C(\mathbf{x}, t, c) &\approx \sum_{q=1}^{q=n} X_q^C(\mathbf{x}) \cdot T_q^C(t) \cdot A_q(c) \\ &\quad + R(\mathbf{x}) \cdot S(t) \cdot W(c) \\ &= C^n(\mathbf{x}, t, c) + R(\mathbf{x}) \cdot S(t) \cdot W(c) \end{aligned} \tag{86}$$

satisfies

$$\int_\Omega \int_0^{T_{max}} \int_0^r C^*(\mathbf{x}, t, c) \cdot \left(\frac{dC}{dt} - \mathcal{L}_c(C)\right) dc\ dt\ d\mathbf{x} = 0 \tag{87}$$

Obviously, due to the discrete character of the third coordinate, an integration quadrature consisting of r points, $c_1 = 1, \ldots, c_r = r$ will be considered later.

Now, for computing the three enrichment functions we are considering again (as in Sect. 2) an alternating directions strategy, that proceeds in three steps (that are repeated until reaching convergence):

1. Assuming functions $S(t)$ and $W(c)$ known, the trial function $C^*(\mathbf{x}, t, c)$ writes $C^*(\mathbf{x}, t, c) = R^*(\mathbf{x}) \cdot S(t) \cdot W(c)$. Thus the weak form (87) reads:

$$\begin{aligned} &\int_\Omega \int_0^{T_{max}} \int_0^r R^* \cdot S \cdot W \\ &\quad \cdot \big(R \cdot S' \cdot W - R \cdot S \cdot \mathcal{L}_c(W)\big)\ dc\ dt\ d\mathbf{x} \\ &= -\int_\Omega \int_0^{T_{max}} \int_0^r R^* \cdot S \cdot W \\ &\quad \cdot \sum_{q=1}^{q=n} \left(X_q^C \cdot (T_q^C)' \cdot A_q\right)\ dc\ dt\ d\mathbf{x} \\ &\quad + \int_\Omega \int_0^{T_{max}} \int_0^r R^* \cdot S \cdot W \\ &\quad \cdot \sum_{q=1}^{q=n} \left(X_q^C \cdot T_q^C \cdot \mathcal{L}_c(A_q)\right)\ dc\ dt\ d\mathbf{x} \end{aligned} \tag{88}$$

where $S' = \frac{dS}{dt}$ and $(T_q^C)' = \frac{dT_q^C}{dt}$.

Now, time integrals and the ones involving the c-coordinate can be performed. The ones involving the coordinate c write:

$$\int_0^r W \cdot W\ dc = \sum_{i=1}^{i=r} W(c_i)^2 \tag{89}$$

where as just mentioned $c_i = i, \forall i$,

$$\int_0^r W \cdot \mathcal{L}_c(W)\ dc = \sum_{i=1}^{i=r} \left(W(c_i) \cdot \sum_{j=1}^{j=r} \alpha_{ij} W(c_j)\right) \tag{90}$$

and similar expressions can be derived for the integrals involved in the right hand member.

Thus, finally it results:

$$\xi^x \int_{\Omega_x} R^* \cdot R\ d\mathbf{x} = \int_{\Omega_x} R^* F^x(\mathbf{x})\ d\mathbf{x} \tag{91}$$

where the coefficient ξ^x contains all the integrals in the time and c-coordinates related to the left hand member of (88) and $F^x(\mathbf{x})$ all the integrals appearing in the right hand member. The strong form related to (91) writes:

$$\xi^x R(\mathbf{x}) = F^x(\mathbf{x}) \tag{92}$$

whose algebraic nature describes the fact that the kinetic model is local and then it does not involve space derivatives.

2. Assuming functions $R(\mathbf{x})$ and $W(c)$ known, the trial function $C^*(\mathbf{x}, t, c)$ writes $C^*(\mathbf{x}, t, c) = R(\mathbf{x}) \cdot S^*(t) \cdot W(c)$. Thus the weak form (87) reads:

$$\begin{aligned} &\int_\Omega \int_0^{T_{max}} \int_0^r R \cdot S^* \cdot W \\ &\quad \cdot \big(R \cdot S' \cdot W - R \cdot S \cdot \mathcal{L}_c(W)\big)\ dc\ dt\ d\mathbf{x} \\ &= -\int_\Omega \int_0^{T_{max}} \int_0^r R \cdot S^* \cdot W \\ &\quad \cdot \sum_{q=1}^{q=n} \left(X_q^C \cdot (T_q^C)' \cdot A_q\right)\ dc\ dt\ d\mathbf{x} \end{aligned}$$

$$+\int_{\Omega}\int_0^{T_{max}}\int_0^r R\cdot S^*\cdot W \cdot\sum_{q=1}^{q=n}\left(X_q^C\cdot T_q^C\cdot\mathcal{L}_c(A_q)\right)\,dc\,dt\,d\mathbf{x} \tag{93}$$

Now, integrals defined in the physical space Ω must be computed, but this task does not involve additional difficulties.

Finally it results:

$$\int_0^{T_{max}} S^*\cdot\left(\xi^t S+\upsilon^t\frac{dS}{dt}\right)dt=\int_0^{T_{max}} S^* F^t(t)\,dt \tag{94}$$

where coefficients ξ^t and υ^t contain all the integrals in the space and the c-coordinate related to the left hand member of (93) and $F^t(t)$ the associated integrals appearing in the right hand member. The strong form related to (94) writes:

$$\upsilon^t\frac{dS}{dt}+\xi^t S(t)=F^t(t) \tag{95}$$

whose first order differential nature results from the first order time derivatives involved by the kinetic model.

3. Assuming functions $R(\mathbf{x})$ and $S(t)$ known, the trial function $C^*(\mathbf{x},t,c)$ writes $C^*(\mathbf{x},t,c)=R(\mathbf{x})\cdot S(t)\cdot W^*(c)$. Thus the weak form (87) reads:

$$\begin{aligned}&\int_{\Omega}\int_0^{T_{max}}\int_0^r R\cdot S\cdot W^*\\&\quad\cdot\left(R\cdot S'\cdot W-R\cdot S\cdot\mathcal{L}_c(W)\right)\,dc\,dt\,d\mathbf{x}\\&=-\int_{\Omega}\int_0^{T_{max}}\int_0^r R\cdot S\cdot W^*\\&\quad\cdot\sum_{q=1}^{q=n}\left(X_q^C\cdot(T_q^C)'\cdot A_q\right)\,dc\,dt\,d\mathbf{x}\\&\quad+\int_{\Omega}\int_0^{T_{max}}\int_0^r R\cdot S\cdot W^*\\&\quad\cdot\sum_{q=1}^{q=n}\left(X_q^C\cdot T_q^C\cdot\mathcal{L}_c(A_q)\right)\,dc\,dt\,d\mathbf{x}\end{aligned} \tag{96}$$

After performing integration in space and time, it results:

$$\int_0^r W^*\cdot\left(\xi^c\,W-\upsilon^c\,\mathcal{L}_c(W)\right)\,dc=\int_0^r W^* F^c(c)\,dc \tag{97}$$

where coefficients ξ^c and υ^c contain all the integrals in the space and time related to the left hand member of (96) and $F^c(c)$ the associated integrals appearing in the right hand member. The strong form related to (97) writes:

$$-\upsilon^c\,\mathcal{L}_c(W)+\xi^c W(c)=F^c(c) \tag{98}$$

that results in the algebraic system:

$$-\upsilon^c\sum_{j=1}^{j=r}\alpha_{ij}W(c_j)+\xi^c W(c_i)=F^c(c_i),\quad i=1,\ldots,r \tag{99}$$

Again, its algebraic nature comes from the nature of the kinetic model.

Remark 4 For the sake of simplicity we illustrated the solution procedure within the alternating directions framework, but the solutions reported in [20] were computed by enforcing the residual minimization described in Sect. 3.4.

3.8 Error Estimation

In this section we are summarizing an error estimator procedure proposed in [8] and based on the use of primal and dual formulations.

For this purpose we assume a generic multidimensional model whose weak form writes

$$a(u,u^*)=b(u^*) \tag{100}$$

defined in $\Omega=\Omega_1\times\cdots\times\Omega_D$, where each Ω_d ($d=1,2,\ldots,D$) involves a coordinate $\mathbf{x}_d$ not necessarily one-dimensional. From now on, the form (100) will be referred as primal form.

The discrete counterpart of (100) reads

$$\mathcal{A}\mathcal{U}=\mathcal{B} \tag{101}$$

where

$$\mathcal{A}=\sum_{j=1}^{n_A}\mathbf{A}_1^j\otimes\mathbf{A}_2^j\otimes\cdots\otimes\mathbf{A}_D^j \tag{102}$$

and

$$\mathcal{B}=\sum_{j=1}^{n_B}\mathbf{B}_1^j\otimes\mathbf{B}_2^j\otimes\cdots\otimes\mathbf{B}_D^j \tag{103}$$

The discrete separated representation of u at iteration n_u writes:

$$\mathcal{U}\approx\sum_{j=1}^{n_u}\mathbf{u}_1^j\otimes\mathbf{u}_2^j\otimes\cdots\otimes\mathbf{u}_D^j \tag{104}$$

where $\mathbf{u}_i^j$ is the discrete (nodal) form of $u_i^j(x_i)$.

Now, we are interested in a certain function of u, $o(u)$, of physical interest (the model output). In what follows we assume that the operator defining the output is linear. Thus, we could write

$$o(u) = \mathcal{O}\left(\sum_{j=1}^{n_F} \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes \cdots \otimes \mathbf{u}_D^j\right) \tag{105}$$

If this operator accepts a separated representation, that is:

$$\mathcal{O} = \sum_{j=1}^{n_O} \mathbf{O}_1^j \otimes \mathbf{O}_2^j \otimes \cdots \otimes \mathbf{O}_D^j \tag{106}$$

then, the output can be evaluated from:

$$o(u) = \sum_{j=1}^{n_u}\sum_{i=1}^{n_O} \left(\mathbf{u}_1^{jT}\mathbf{O}_1^i\right)\cdot\left(\mathbf{u}_2^{jT}\mathbf{O}_2^i\right)\cdots\left(\mathbf{u}_D^{jT}\mathbf{O}_D^i\right) \tag{107}$$

Now, the error on the output, can be evaluated by solving the so-called dual problem (see [8] and the references therein):

$$a(u^*, v) = o(u^*) \tag{108}$$

whose discrete form writes:

$$\mathcal{A}^T\mathcal{V} = \mathcal{O} \tag{109}$$

where $\mathcal{O}$ was already defined and where $\mathcal{A}^T$ is given by

$$\mathcal{A}^T = \sum_{j=1}^{n_A} \mathbf{A}_1^{jT} \otimes \mathbf{A}_2^{jT} \otimes \cdots \otimes \mathbf{A}_D^{jT} \tag{110}$$

A good error estimation needs for an accurate solution of the dual problem. Within the separated representation framework the solution of the dual problem (109) (assumed accurate enough) can be written as:

$$\mathcal{V} \approx \sum_{j=1}^{n_v} \mathbf{v}_1^j \otimes \mathbf{v}_2^j \otimes \cdots \otimes \mathbf{v}_D^j \tag{111}$$

Now, the error in the output can be evaluated from:

$$o(e) = b(v) - a(u, v) \tag{112}$$

whose discrete counterpart writes:

$$o(e) = \mathcal{V}^T\mathcal{B} - \mathcal{V}^T\mathcal{A}\mathcal{U} \tag{113}$$

that is easily computed from:

$$\begin{aligned} o(e) = &\sum_{i=1}^{n_B}\sum_{j=1}^{n_v} \left(\mathbf{v}_1^{jT}\mathbf{B}_1^i\right)\cdot\left(\mathbf{v}_2^{jT}\mathbf{B}_2^i\right)\cdots\left(\mathbf{v}_D^{jT}\mathbf{B}_D^i\right) \\ &- \sum_{k=1}^{n_A}\sum_{i=1}^{n_u}\sum_{j=1}^{n_v}\left(\mathbf{v}_1^{jT}\mathbf{A}_1^k\mathbf{u}_1^i\right) \\ &\cdot\left(\mathbf{v}_2^{jT}\mathbf{A}_2^k\mathbf{u}_2^i\right)\cdots\left(\mathbf{v}_D^{jT}\mathbf{A}_D^k\mathbf{u}_D^i\right) \end{aligned} \tag{114}$$

3.9 Parametric Models

The case of parametric models was addressed in [42]. In this section we revisit the main ideas in the treatment of such models. For this purpose we consider again the heat equation but in which the thermal diffusivity is unknown. The most usual strategy in that case lies in solving that equation for many values of the thermal diffusivity (Monte Carlo procedure). If the statistical distribution of the realizations of the thermal diffusivity represents accurately the real thermal diffusivity distribution, then from the computed temperature fields we can infer different outputs (in a statistical sense).

However, this problem can also be solved in a fully deterministic way if the unknown parameter is considered as a new coordinate (as the physical ones—space and time). Thus, the dimensionality of the problem increases, but this fact is not a serious handicap for the PGD.

We consider again the one-dimensional heat equation

$$\frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(k\frac{\partial u}{\partial x}\right) = 0 \quad \forall t \in \Omega_t\ \forall x \in \Omega_x \tag{115}$$

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

$$k = au + b \tag{116}$$

where coefficients a and b are assumed unknown or badly known.

If we introduce the diffusivity expression into the heat equation (115) it results:

$$\frac{\partial u}{\partial t} - b\frac{\partial^2 u}{\partial x^2} - au\frac{\partial^2 u}{\partial x^2} - a\left(\frac{\partial u}{\partial x}\right)^2 = 0 \tag{117}$$

The goal in the solution 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 theirs domains of variability, i.e. $u(t, x, a, b)$:

$$u(t, x, a, b) \approx \sum_{i=1}^{N} T_i(t)\cdot X_i(x)\cdot A_i(a)\cdot B_i(b) \tag{118}$$

In [42] we considered 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 piecewise 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 less than one minute for solving this problem using a personal laptop.

There is no restriction on the number of parameters that can be transformed into additional model coordinates. In [42] we considered the thermal diffusivity, the source term and the initial condition as additional coordinates.

Remark 5 The introduction of model parameters as extra-coordinates has not an impact in the computational efforts, because as the departure model (the heat transfer equation in the present case) does not involve derivatives with respect to those parameters, when we compute the functions related to these coordinates (associated to the model parameters) only algebraic problems must be solved. Moreover, the use of the residual minimization described in Sect. 3.4 increases significantly the convergence rate of the PGD.

3.10 Solvers Coupling and New Integration Procedures

As the PGD operates by solving the different functions involved in the separated representation independently, one could imagine that different solvers could be applied for solving the problems in the different coordinates.

A direct consequence of this fact was the definition of new integration procedures. In [28] we considered the integration of transient models by applying the Boundary Element Method (BEM). In what follows we revisit the main ideas on this non-incremental boundary element strategy.

3.10.1 Non-incremental Boundary Element Discretizations

The Boundary Element Method allows for an efficient solution of partial differential equations whose kernel functions are known. The heat equation is one of these candidates. When the model involves large physical domains and time simulation intervals the amount of information that must be stored increases significantly.

We proposed in [28] an alternative radically different that leads to a separated solution of the space and time problems within a non-incremental integration strategy. The technique is based on the use of a space-time separated representation of the unknown field that introduced in the residual weighting formulation allows to define a separated solution of the resulting weak form. The spatial step can be then treated by invoking the standard BEM for solving the resulting steady state problem defined in the physical space. Then, the time problem that results in an ordinary first order differential equation is solved using any standard appropriate integration technique (e.g. backward finite differences).

In principle this procedure opens new possibilities for integrating transient models (linear or non-linear) whose space-time fundamental solution is not known but whose steady state fundamental solution is known.

We are illustrating the strategy for constructing these functional products in the case of an academic transient problem, the transient linear heat equation:

$$\frac{\partial u}{\partial t} - a\Delta u = f(\mathbf{x}, t) \quad \text{in } \Omega \times (0, T_{max}] \tag{119}$$

with the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x}, 0) = u^0, & \mathbf{x} \in \Omega \\ u(\mathbf{x}, t) = u_g, & (\mathbf{x}, t) \in \partial\Omega \times (0, T_{max}] \end{cases} \tag{120}$$

where a is the diffusion coefficient. The weak formulation yields:

Find $u(\mathbf{x}, t)$ verifying the boundary conditions (7) such that

$$\int_0^{T_{max}} \int_\Omega u^\star \left(\frac{\partial u}{\partial t} - a\Delta u - f(\mathbf{x}, t) \right) d\mathbf{x}\, dt = 0 \tag{121}$$

for all the functions $u^\star(x, t)$ in an appropriate functional space.

We compute now the functions involved in the sum (8). We suppose that the set of functional couples $\{(X_i, T_i)\}_{i=1,\dots,n}$ with $0 \le n < N$ are already known (they have been previously computed) and that at the present iteration we search the enrichment couple $(R(t), S(\mathbf{x}))$ by applying an alternating directions fixed point algorithm that after convergence will constitute the next functional couple (X_{n+1}, T_{n+1}). Hence, at the present iteration, n, we assume the separated representation

$$u(\mathbf{x}, t) \approx \sum_{i=1}^n T_i(t) \cdot X_i(\mathbf{x}) + R(t) \cdot S(\mathbf{x}) \tag{122}$$

The weighting function $u^\star$ is then assumed as

$$u^\star = S \cdot R^\star + R \cdot S^\star \tag{123}$$

Introducing (10) and (11) into (9) it results

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega (S \cdot R^\star + R \cdot S^\star) \cdot \left(S \cdot \frac{\partial R}{\partial t} - a\Delta S \cdot R \right) d\mathbf{x}\, dt \\ &\quad = \int_0^{T_{max}} \int_\Omega (S \cdot R^\star + R \cdot S^\star) \\ &\qquad \cdot \left(f(\mathbf{x}, t) - \sum_{i=1}^n X_i \cdot \frac{\partial T_i}{\partial t} + a \sum_{i=1}^n \Delta X_i \cdot T_i \right) d\mathbf{x}\, dt \end{aligned} \tag{124}$$

We apply an alternating directions fixed point algorithm to compute the couple of functions (R, S). Following the procedure described in Sect. 2, functions $S(\mathbf{x})$ and $R(t)$ satisfy:

$$\alpha_t S - a\beta_t \Delta S = \gamma_t - \sum_{i=1}^{n} \alpha_t^i X_i + a \sum_{i=1}^{n} \beta_t^i \Delta X_i \tag{125}$$

and

$$\beta_x \frac{\partial R}{\partial t} = a \cdot \alpha_x \cdot R + \gamma_x(t) - \sum_{i=1}^{n} \beta_x^i \cdot \frac{\partial T_i}{\partial t} + \sum_{i=1}^{n} a \cdot \alpha_x^i \cdot T_i \tag{126}$$

We can notice that (125) defines a steady-state elliptic equation with constant coefficients. Being the Green solution associated to that equation known, one could apply the BEM for solving it. The only issue in applying the BEM for solving (125) is the integration of the right hand member that implies volumetric integrations that need the definition of appropriate approximation everywhere in the domain. In [28] the approximation was performed by using a moving least square technique. The main advantage in using this strategy is the possibility of solving models for which the Green solution of the space model is known even when it is not the case for the space-time model.

The interested reader can refer to [28] for some numerical experiments and convergence analysis on this non-incremental boundary element technique.

3.11 Miscellaneous: High Resolution Solvers, Multi-physics and Mixed Formulations

Separated representations involving ordinary models (transient or steady state; 2D or 3D) allow in several cases impressive computing time savings. In the transient case is evident as discussed previously (see the discussion at the end of Sect. 2). In the case of rectangular domains is also possible to separate the different space coordinates x, y and z (in the case of more complex domains the separated representation can also be applied by using appropriate strategies, as the one proposed in [25] and here summarized in Sect. 3.5).

If a fully separated representation can be applied, only 1D problems have to be solved. These boundary-value problems ca be solved efficiently even for extremely large number of nodes distributed in the 1D interval, if the boundary value problem is transformed into an initial value problem by using the technique proposed in [14–16] that avoids the solution of any linear system. The combination of the PGD and these advanced 1D boundary value solvers allows for impressive CPU time savings of several orders of magnitude.

In [18] we considered linear computational homogenization of heterogeneous materials. A simple thermal model must be solved in a quite simple domain (a cube) but in order to capture all the microstructure details a extremely fine mesh is needed (high resolution). The use of a fully separated representation allows considering millions of nodes in each space direction, solving models, at present, beyond the finite elements expectations.

The same kind of techniques were applied for solving multi-physics models arising in the composites manufacturing processes, where curing kinetics are coupled with the non-linear thermal and thermo-mechanical behaviors [41]. The simplest coupling lies in the use of a fixed point strategy, but the LATIN framework or the monolithic approaches could be more efficient alternatives.

Recently we have also solved in a fully separated space description the incompressible infinitesimal strain elasticity equations (that are the same the ones that govern the flow of a Newtonian incompressible fluid—the Stokes equations). The treatment of these mixed formulations needs to satisfy the LBB conditions. At present there are no mathematical results on this topic, but in our numerical experiments we noticed that the one-dimensional meshes used for approximating the displacements field must be richer than the ones used for approximating the pressure field (Lagrange multiplier associated with the incompressibility constraint) and moreover, we must enrich the pressure separated representation one time for, at least, two displacement enrichments. The understanding of these observation represents a work in progress.

4 Future Developments

The separated representation methodology allows in many cases circumventing the redoubtable curse of dimensionality, or at least alleviating its impact. Thus, the PGD seems to be an appealing and powerful strategy for solving models defined in high dimensional spaces including physical (space and time) and configurational (also known as conformational) coordinates. Thus, the solution of many physical models suffering the curse of dimensionality illness is now possible, opening new possibilities in quantum chemistry, the kinetic theory description of materials within the statistical mechanics framework, the solution of the chemical master equation that is present in many branches of biology (genetics, cell-signaling, ...), several models encountered in financial mathematics, ...

However, there are numerous other possibilities based in transforming usual models (defined in moderate dimensions, generally space and time) into higher dimensional models that include all the desired model parameters as new model coordinates defined in theirs respective intervals. Thus, if the increase in the model dimensionality is no more a serious drawback (and it is the case if the PGD is applied) one could solve a parametric equation only one time and then particularize the solution for different choices of the parameters.

This procedure only implies post-processing, the computation is performed off-line and only one time! It is easy to infer the impact that such procedure could constitute in the context of optimization or inverse identification. There are many other possibilities, it suffices to change our mind, our way of modeling the physics, without trying to enforce the models of physics to be three-dimensional: the models of a physics taking place in the space and time could be multidimensional (the mathematics accepts it!).

In this section we are addressing some comments in different potential and exciting application of the Proper Generalized Decomposition.

- *Optimization and inverse identification*
Imagine that we are interested in optimizing a thermal process by choosing the optimal thermal diffusivity k (that in what follows we consider constant), or that we are interested in identifying this thermal diffusivity from the experimental data recorded by a thermocouple placed at a certain position. In both cases, when one uses any standard strategy, the solution of many direct problems is needed, as well as a minimization strategy able to search the optimum value with respect to a certain cost function.

Our approach consists in assuming the thermal diffusivity as a new coordinate of the model, as described in Sect. 3.9. By applying the PGD we can compute the general solution

$$u(\mathbf{x},t,k) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) \tag{127}$$

that represents the value at each position $\mathbf{x}$, time t and for each value of the thermal diffusivity k.

As soon as a trial thermal diffusivity k^{trial} is generated by the minimization algorithm, the temperature field becomes defined at each point $\mathbf{x} \in \Omega_x$ and time $t \in \mathcal{I} \equiv \Omega_t$

$$u(\mathbf{x},t;k^{trial}) \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k^{trial}) \tag{128}$$

that allows an easy and fast calculation of the cost function and its gradient. If we denote the cost function $\mathcal{C}(u)$, its gradient reads:

$$\frac{\partial \mathcal{C}}{\partial k} = \frac{\partial \mathcal{C}}{\partial u} \cdot \frac{\partial u}{\partial k} \tag{129}$$

that can be easily determined from the cost function expression and from the derivative of (128):

$$\left.\frac{\partial u}{\partial k}\right|_{k^{trial}} \approx \sum_{i=1}^{i=N} X_i(\mathbf{x}) \cdot T_i(t) \cdot \left.\frac{dK_i(k)}{dk}\right|_{k^{trial}} \tag{130}$$

Moreover, for simple cost functions one could expect to find directly the extremum points, that is, the points at which the derivative with respect to the design variables (in the present case the thermal diffusivity) vanish:

$$\frac{\partial \mathcal{C}(u)}{\partial k} = \frac{\partial \mathcal{C}}{\partial u} \cdot \frac{\partial u}{\partial k} = 0 \tag{131}$$

Let $\mathcal{K}$ be the set of such extremum points ($\mathcal{K} = \{k_1, \ldots, k_M\}$). Now, it suffices calculating the cost functions for each value of the thermal diffusivity $k \in \mathcal{K}$: $\mathcal{C}_1, \ldots, \mathcal{C}_M$ and select the minimum value, i.e. $k_{op} = k_j$ such that $\mathcal{C}_j = \min\{\mathcal{C}_1, \ldots, \mathcal{C}_M\}$. Thus, we could compute the global minimum, instead of a simple minimum (not necessary the global one) obtained by using standard minimization techniques.

- *Evolving domains*
Some times we are interested in solving models defined in a domain that is evolving in time, i.e. the partial differential equations are defined in $\Omega_x(t)$. There are probably many ways to treat this kind of models, but a simple procedure consists in writing the problem in the reference domain Ω_x^{ref} (that could be identified to the domain at the initial time, i.e. $\Omega_x^{ref} = \Omega_x(t=0)$).

Now, if the kinematics is known, the function given the position at time t of a point that occupied the position $\mathbf{X}$ at time $t=0$ can be easily derived: $\mathbf{x} = \mathbf{x}(\mathbf{X},t)$. Now, the partial differential equation defined in $\Omega_x(t) \times \Omega_t$ can be transformed in its counterpart now defined in $\Omega_x^{ref} \times \Omega_t$. This procedure is quite standard in computational mechanics when one proceeds in the Lagrangian framework.

Now, with the problem defined in $(\mathbf{X},t) \in \Omega_x^{ref} \times \Omega_t$, the natural separated representation writes:

$$u(\mathbf{X},t) \approx \sum_{i=1}^{i=N} X_i(\mathbf{X}) \cdot T_i(t) \tag{132}$$

The global weak space-time formulation is first written, and then the functions of space and time searched after integration in the time and space domains respectively (we don't enforce the balance equation at any particular time step as is the case when using incremental strategies).

- *Towards real time simulations*
Many problems in engineering needs for very fast solutions, sometimes in the real time range. As the models in engineering are quite complex (non-linear, involving thousands or even millions degrees of freedom, ...) the real time constraint is an intractable issue in numerical simulation by using the standard procedures, despite the impressive recent progresses in the computational resources. Parallel platforms, high performance computing ... do not suffice, at least at present, for reaching the real time simulation requirements.

One area in which we are specially interested is the one that concerns surgery simulators. In these applications, real time is not a caprice, real time is needed to be a valuable surgery tool. Living tissues are non-linear, anisotropic, geometrically complex, and the applied loads move on their surfaces ... all them making difficult fast simulations.

In [36, 37] and [38] we explored the use of model reduction techniques based on the use of Proper Orthogonal Decompositions (POD) combined with an advanced non-linear solver (the asymptotic numerical method) that avoids the necessity of recomputing the tangent matrix at each load updating. Despite these advanced computational ingredients, real time requirements were reached with some difficulties.

At present we are considering the application of the PGD. We could solve the non-linear elasticity problem (involving material and geometrical non-linearities) including the moving load applying in the organ surface as a new model coordinate. Let **p** the vector representing the applied force on the organ surface. The point at which that force applies is designated by **y**. As we are considering quasi-static behaviors we don't need to specify the dependence $\mathbf{y}(t)$.

Now, the non-linear elasticity problem is solved assuming the separated representation of the displacement field given formally by:

$$\mathbf{u}(\mathbf{x},\mathbf{y},\mathbf{p}) \approx \sum_{i=1}^{i=N} \mathbf{X}_i(\mathbf{x}) \cdot \mathbf{Y}_i(\mathbf{y}) \cdot \mathbf{P}_i(\mathbf{p}) \tag{133}$$

that represents the displacement **u** at point **x** when a load **p** applies at position **y**.

Now, as soon as the force and its location are given, $\mathbf{p}^g$ and $\mathbf{y}^g$, the displacement field is calculated

$$\mathbf{u}(\mathbf{x};\mathbf{y}^g,\mathbf{p}^g) \approx \sum_{i=1}^{i=N} \mathbf{X}_i(\mathbf{x}) \cdot \mathbf{Y}_i(\mathbf{y}^g) \cdot \mathbf{P}_i(\mathbf{p}^g) \tag{134}$$

The construction of this general solution is made once, off line, and then used in-line as a simple post-processing allowing to fulfill real time constraints.

- *Homogenization*
Many processes and/or materials exhibit many scales. Microstructure in materials induces different space scales. Processes can also induce different time scales. Thus, for example when one considers processes like ultrasonic welding [31] the scale related to the load evolution is of some microseconds, whereas the characteristic time related to the welding process itself is of some seconds.

When these scales are well separated (turbulence is a nice couner-example!) we could consider separated (and independent) coordinates related to the different scales. For the sake of simplicity we consider two time scales. The fastest one implies a time τ being $\tilde{t}$ the one related to the slowest one. The characteristic times of both scales define the scale factor ϵ: $\tilde{t} = \epsilon\tau$.

Thus, the time dependence of a generic field $u(t)$ can be rewritten as $u(\tilde{t},\tau)$ and then

$$\frac{du}{dt} = \frac{\partial u}{\partial \tilde{t}} \cdot \frac{\partial \tilde{t}}{\partial t} + \frac{\partial u}{\partial \tau} \cdot \frac{\partial \tau}{\partial t} \tag{135}$$

that taking into account that $\tilde{t} = t$ reduced to:

$$\frac{du}{dt} = \frac{\partial u}{\partial \tilde{t}} + \frac{\partial u}{\partial \tau} \cdot \frac{1}{\epsilon} \tag{136}$$

and so on for the higher order derivatives. Thus, the partial differential equations involved in the thermo-mechanical model of the process can be rewritten making use of the two new time coordinates.

In general, for solving the resulting model an asymptotic expansion of the different fields is performed, and then by identifying the equation at the different orders the model is solved and the effects of the microscopic scale appears naturally in the macroscopic one. However, in some situations this procedure becomes quite technical.

The use of the PGD avoids the asymptotic expansion because the multidimensional problem that results by introducing the different independent time coordinates can be efficiently solved without major difficulties by applying the PGD. Thus, the solution is searched under the separated form:

$$u(\tau,\tilde{t}) \approx \sum_{i=1}^{i=N} \mathbf{\Theta}_i(\tau) \cdot \mathbf{T}_i(\tilde{t}) \tag{137}$$

In general this procedure can be extended to models involving more than two time scales, for the ones involving many space scales (rich microstructures with many different characteristic lengths) and for the ones combining many space and time scales.

- *Convective stabilization*
It is well-known that standard finite element (Galerkin) methods do not work well for convection-diffusion or convection-diffusion-reaction equations, since they lead to unstable, oscillating, solutions [22]. The first stabilization methods including upwinding in the convective term in order to produce an artificial diffusion that eventually lead to stable solutions were published in [26].

Among the very numerous methods that have been proposed for the stabilization of convection-diffusion equations, the streamline-upwind/Petrov-Galerkin (SUPG) method [27] is one of the most extended. The major drawback of this method is that it introduces some cross-wind artificial numerical diffusion when applied to two- or three-dimensional equations.

The use of PGD could circumvent, or at least alleviate this drawback. It has been noticed that the use of such an approximation leads to a sequence of different one-dimensional problems, for which "exact" SUPG stabilizations exist. In this way, we can apply a standard SUPG method to a sequence of one-dimensional problems to obtain a properly stabilized solution to a two- or three-dimensional convection-diffusion(-reaction) problem.

For simplicity, we consider the steady-state 2D convection-diffusion(-reaction) equation. This equation is given by

$$\boldsymbol{v} \cdot \nabla u - \nabla \cdot (a \nabla u) + \sigma u = s \quad \text{in } \Omega \subset \mathbb{R}^n \tag{138}$$

with $n = 2, 3$ and with boundary conditions

$$u = u_D \quad \text{on } \Gamma_D \tag{139a}$$

$$\boldsymbol{n} \cdot a \nabla u = a \frac{\partial u}{\partial \boldsymbol{n}} = t \quad \text{on } \Gamma_N \tag{139b}$$

where u is the scalar quantity to be transported and also the unknown field of the problem, $\boldsymbol{v}$ is the advective velocity, $a > 0$ the diffusivity, assumed constant, σ the reaction term and $s(\boldsymbol{x})$ a volumetric source term. The function u_D denotes the prescribed value of u on the Dirichlet portion of the boundary given by Γ_D and t denotes the value of the normal diffusive flux on the Neumann boundary Γ_N. In what follows, for the sake of simplicity, we shall omit the reaction term of the equation, since it does not imply any special difficulty for the technique here developed.

The weak form of the problem given by (138)–(139) is:

Find $u(\boldsymbol{x}) \in \mathcal{S} = \{u \in \mathcal{H}^1(\Omega) | u = u_D \text{ on } \Gamma_D\}$ *such that for all* $w \in \mathcal{V} = \{w \in \mathcal{H}^1(\Omega) | w = 0 \text{ on } \Gamma_D\}$

$$\int_\Omega w(\boldsymbol{v} \cdot \nabla u) d\Omega + \int_\Omega \nabla w \cdot (a \nabla u) d\Omega = \int_\Omega w s d\Omega + \int_{\Gamma_N} w t d\Gamma \tag{140}$$

which is very often expressed compactly with the help of the following notation:

$$a(w, u) = \int_\Omega \nabla w \cdot (a \nabla u) d\Omega \tag{141a}$$

$$c(\boldsymbol{v}; w, u) = \int_\Omega w(\boldsymbol{v} \cdot \nabla u) d\Omega \tag{141b}$$

$$(w, s) = \int_\Omega w s d\Omega \tag{141c}$$

$$(w, t)_{\Gamma_N} = \int_{\Gamma_N} w t d\Gamma \tag{141d}$$

giving rise to the following compact form of the equation:

$$a(w, u) + c(\boldsymbol{v}; w, u) = (w, s) + (w, t)_{\Gamma_N} \tag{142}$$

The general form of the *consistent stabilization* techniques is [22]

$$a(w, u) + c(\boldsymbol{v}; w, u) + \sum_e \int_{\Omega_e} \mathcal{P}(w) \tau \mathcal{R}(u) d\Omega = (w, s) + (w, t)_{\Gamma_N} = l(w) \tag{143}$$

where $\mathcal{P}(w)$ is some operator applied to the test functions and $\mathcal{R}(u) = \mathcal{L}(u) - s$ is the residual of the equation. In the SUPG method, $\mathcal{P}(w) = \boldsymbol{v} \cdot \nabla w$.

The exact nodal values for 1D convection-diffusion equation are obtained for a value

$$\tau = \frac{h}{2v} \left(\coth Pe - \frac{1}{Pe} \right) \tag{144}$$

where Pe is the mesh Péclet number, defined as $Pe = vh/2a$. h represents the mesh size parameter and v the modulus of the convective velocity. The problem with this type of stabilization is that, on one side, this value has been defined for one-dimensional problems and linear finite elements in order to represent exactly the problem solution.

Within the PGD framework, in the enrichment stage we look for an improved representation of the essential field in the form

$$u^{n+1}(x, y) = \sum_{i=1}^{n} X_i(x) \cdot Y_i(y) + R(x) \cdot S(y) \tag{145}$$

or, equivalently,

$$u^{n+1}(x, y) = u^n(x, y) + R(x) \cdot S(y) \tag{146}$$

The test function will then be given by

$$w^*(x, y) = R^*(x) \cdot S(y) + R(x) \cdot S^*(y) \tag{147}$$

When substituting (146) and (147) into the weak form of the problem, (143), we arrive at

$$\begin{aligned} & a\left(R^*S + RS^*, RS\right) + c\left(\boldsymbol{v}; R^*S + RS^*, RS\right) \\ & \quad = -a\left(R^*S + RS^*, u^n\right) - c\left(\boldsymbol{v}; R^*S + RS^*, u^n\right) \\ & \qquad + l(R^*S + RS^*) \\ & \qquad - \sum_e \int_{\Omega_e} \boldsymbol{v} \nabla (R^*S + RS^*) \tau \left[\mathcal{L}(u^{n+1}) - s\right] d\Omega \end{aligned} \tag{148}$$

where we have omitted, for clarity, the obvious dependence of R in x and S in y.

After applying such an approximation, the weak form of the problem given by (148) is solved by using the fixed-point algorithm (extensively used throughout the present paper) in which the R and S functions are sought iteratively.

For instance, assuming $R(x)$ known, the resulting expression will be

$$\begin{aligned}&a\left(RS^*, RS\right)+c\left(\boldsymbol{v}; RS^*, RS\right)\\&\quad=-a\left(RS^*, u^n\right)-c\left(\boldsymbol{v}; RS^*, u^n\right)\\&\qquad+l(RS^*)\\&\qquad-\sum_e\int_{\Omega_e}\boldsymbol{v}\nabla(RS^*)\tau\left[\mathcal{L}(u^{n+1})-s\right]d\Omega\end{aligned}\tag{149}$$

Assuming ν constant, we have

$$\begin{aligned}\mathcal{L}(u^{n+1})&=\boldsymbol{v}\cdot\nabla u^{n+1}-\nabla\cdot(\nu\nabla u^{n+1})\\&=\mathcal{L}(u^n)+\boldsymbol{v}\cdot\nabla RS-\nu\nabla^2 RS\end{aligned}$$

so that we arrive at

$$\begin{aligned}&a\left(RS^*, RS\right)+c\left(\boldsymbol{v}; RS^*, RS\right)\\&\quad=-a\left(RS^*, u^n\right)-c\left(\boldsymbol{v}; RS^*, u^n\right)+l(RS^*)\\&\qquad\underbrace{-\sum_e\int_{\Omega_e}\boldsymbol{v}\cdot\nabla(RS^*)\tau\left[\boldsymbol{v}\cdot\nabla RS-\nu\nabla^2 RS-s\right]d\Omega}_{A}\\&\qquad\underbrace{-\sum_e\int_{\Omega_e}\boldsymbol{v}\cdot\nabla(RS^*)\cdot\tau\cdot\mathcal{L}(u^n)\,d\Omega}_{B}\end{aligned}\tag{150}$$

After some lengthy, yet simple, algebra and after taking into account that second derivatives of functions R and S vanish if we approximate them by standard, first-order, finite elements, we obtain for the term A,

$$\begin{aligned}A=\sum_e\int_{\Omega_e}&\left(v_x\cdot\frac{dR}{dx}\cdot S^*+v_y\cdot R\cdot\frac{dS^*}{dy}\right)\\&\cdot\tau\cdot\left(v_x\cdot\frac{dR}{dx}\cdot S+v_y\cdot R\cdot\frac{dS}{dy}-s\right)d\Omega\end{aligned}\tag{151}$$

For the term B we obtain

$$\begin{aligned}B=\sum_e\int_{\Omega_e}&\left(v_x\cdot\frac{dR}{dx}\cdot S^*+v_y\cdot R\cdot\frac{dS^*}{dy}\right)\\&\cdot\tau\cdot\left(v_x\cdot\frac{\partial u^n}{dx}+v_y\cdot\frac{\partial u^n}{dy}\right)d\Omega\end{aligned}\tag{152}$$

The impact of the choice of τ is being evaluated and PGD solutions compared with fully 2D SUPG stabilized solutions.

- *Separated representation of Molecular Dynamics*
The molecular dynamic problems are based on the resolution of a set of Newton's equations for a given number of particles denoted here by N. The system is described by a set of discrete positions varying during time $x_1(t), x_2(t), \ldots, x_N(t)$. The dynamics of each particle depends on the position of the surrounding particles through an interaction potential as well as on its own position in the case of an existence of an external field generating a force on each particle. The general equations are then written in the next form:

$$\frac{d^2x_i(t)}{dt^2}=\frac{1}{m_i}f_i(x_1,\ldots,x_N),\quad i=1,\ldots,N\tag{153}$$

m_i is the mass and f_i is the force applied on the ith particle which is non linearly dependent on the system state. These N equations can be gathered into a compact notation involving a vector $\mathbf{x}=(x_1,\ldots,x_N)^T$.

$$\frac{d^2\mathbf{x}}{dt^2}=\mathcal{L}(\mathbf{x})\tag{154}$$

where the vectorial discrete operator $\mathcal{L}$ contains all forces and masses contributions.

The main difficulty associated to the solution of this equations is associated to the high number of iteration steps for which we have to solve high number of discrete equations related to each particle. In the context of the PGD we can look for the solution directly as a product of time functions $T_j(t)$ by some configurations states of the particles given by a position coordinates vector $\mathbf{X}_j$. $\mathbf{X}_j$ could be interpreted as a discrete significant mode of the position vector during the time evolution that has been to be calculated 'a priori'. If we look for a general solution that writes

$$\mathbf{x}(t)\approx\sum_{j=1}^{R}\mathbf{X}_j.T_j(t)\tag{155}$$

and if we assume known the first n couples of configuration states and time functions ($n<R$) then the new couple $\mathbf{Y}$, $S(t)$ deriving from the enrichment process must satisfy

$$\sum_{j=1}^{n}\left(\mathbf{X}_j\frac{d^2T_j}{dt^2}\right)+\mathbf{Y}\frac{d^2S}{dt^2}=\mathcal{L}(\mathbf{X}_jT_j+\mathbf{Y}S)\tag{156}$$

The residual minimization of Sect. 3.4 could be applied onto this equation. The main difficulty is related to the updating of the discrete operator $\mathcal{L}$ that accounts for high non linearity. This requires a complete reconstitution of the time-space solution for each enrichment stage (if we consider the simplest updating of the non linearity described in Sect. 3.2). Even if this technique allows to find the solution with an 'a priori' estimation it costs, unfortunately, R times more expensive than the direct simulation.

- *New dreams*
The possibility of solving efficiently multidimensional models including hundreds of coordinates [4] opens new possibilities in that concerns the solution of models that

are in nature multidimensional (quantum chemistry, statistical mechanics, biology, computational finance, ...) but also many others (standard in nature) that could be transformed into highly multidimensional models.

We could expect that some non-linear problems could be linear in a higher dimensional spaces, making possible its solution in such spaces and then come back to the initial one solving a non-linear algebraic problem. Other possibility that we recently explored lies in the solution of some simple models that exhibit bifurcations in theirs solution. The treatment of these models needs some appropriate technique able to identify the bifurcation points and to follow each solution branch. Our idea was to immerse these physics in higher dimensional spaces, such that the solution can represent simultaneously all the solution richness. Thus, in the buckling of a column in 2D, we obtained a solution that after reaching the critical load exhibit all the possible solutions simultaneously. The price to be paid is obviously the consideration of many extra-coordinates. Thus, if a solver able to address highly multidimensional models is available, new and some times unimaginable possibilities appears suddenly!

5 Conclusions

In this paper we presented the ability of the Proper Generalized Decomposition—PGD—for solving highly multidimensional models. This technique operates by constructing a separated representation of the solution, such that, the solution complexity scales linearly with the dimension of the space in which the model is defined, instead the exponentially-growing complexity characteristic of mesh based discretization strategies.

The PGD makes possible the efficient solution of models defined in multidimensional spaces, as the ones encountered in quantum chemistry, kinetic theory description of complex fluids, genetics (chemical master equation), financial mathematics, ... but also those, classically defined in the standard space and time, to which we can add new extra-coordinates (parametric models, ...) opening numerous possibilities (optimization, inverse identification, real time simulations, ...).

Despite the first success in the treatment of the models described in this paper, this numerical proposal is too recent to conclude about its limits, ... Thus, the analysis should continue for identifying new opportunities but also its limitations.

Acknowledgements The authors acknowledge the contribution of many colleages to the works here presented: I. Alfaro, F. Daim, M. Doblaré, P. Diez, A. Falco, J. Ferec, D. Gonzalez, A. Hamdouni, A. Huerta, P. Joyot, P. Ladeveze, B. Mokdad, A. Nouy, A. Poitou, E. Pruliere and P. Villon.

References

1. Achdou Y, Pironneau O (2005) Computational methods for option pricing. SIAM frontiers in applied mathematics. SIAM, Philadelphia
2. Ammar A, Ryckelynck D, Chinesta F, Keunings R (2006) On the reduction of kinetic theory models related to finitely extensible dumbbells. J Non-Newton Fluid Mech 134:136–147
3. Ammar A, Mokdad B, Chinesta F, Keunings R (2006) A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. J Non-Newton Fluid Mech 139:153–176
4. Ammar A, Mokdad B, Chinesta F, 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. J Non-Newton Fluid Mech 144:98–121
5. Ammar A, Chinesta F (2008) Circumventing curse of dimensionality in the solution of highly multidimensional models encountered in quantum mechanics using meshfree finite sums decomposition. In: Lecture notes on computational science and engineering. Springer, Berlin, pp 1–17
6. Ammar A, Normandin M, Daim F, Gonzalez D, Cueto E, Chinesta F (2010) Non-incremental strategies based on separated representations: Applications in computational rheology. Commun Math Sci 8(3):671–695
7. Chinesta F, Ammar A, Cueto E (2010) On the use of Proper Generalized Decompositions for solving the multidimensional chemical master equation. Eur J Comput Mech 19:53–64
8. Ammar A, Chinesta F, Diez P, Huerta A (2010) An error estimator for separated representations of highly multidimensional models. Comput Methods Appl Mech Eng 199:1872–1880
9. Ammar A, Chinesta F, Cueto E (2010) Coupling finite elements and separated representations. Int J Multiscale Comput Eng (in press)
10. Beylkin G, Mohlenkamp M (2005) Algorithms for numerical analysis in high dimensions. SIAM J Sci Comput 26(6):2133–2159
11. Bird BB, Curtiss CF, Armstrong RC, Hassager O (1987) Dynamics of polymeric liquids. In: Kinetic theory, vol 2. Wiley, New York
12. Bungartz HJ, Griebel M (2004) Sparse grids. Acta Numer 13:1–123
13. Cancès E, Defranceschi M, Kutzelnigg W, Le Bris C, Maday Y (2003) Computational quantum chemistry: a primer. Handbook of numerical analysis, vol X. Elsevier, Amsterdam, pp 3–270
14. Chaidron G, Chinesta F (2002) On the steady solution of non-linear advection problems in steady recirculating flows. Comput Methods Appl Mech Eng 191:1159–1172
15. Chinesta F, Chaidron G (2001) On the steady solution of linear advection problems in steady recirculating flows. J Non Newton Fluid Mechanics 98:65–80
16. Chinesta F, Chaidron G (2003) A characteristics strategy for solving advection equations in 2D steady flows containing recirculating areas. Comput Methods Appl Mech Eng 192:37–38, 4217–4235
17. Chinesta F, Ammar A, Falco A, Laso M (2007) On the reduction of stochastic kinetic theory models of complex fluids. Model Simul Mater Sci Eng 15:639–652
18. Chinesta F, Ammar A, Lemarchand F, Beauchene P, Boust F (2008) Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization. Comput Methods Appl Mech Eng 197(5):400–413
19. Chinesta F, Ammar A, Joyot P (2008) The nanometric and micrometric scales of the structure and mechanics of materials revisited: An introduction to the challenges of fully deterministic numerical descriptions. Int J Multiscale Comput Eng 6(3):191–213

20. Chinesta F, Ammar A, Cueto E (2010) Proper generalized decomposition of multiscale models. Int J Numer Methods Eng 83(8–9):1114–1132
21. Ammar A, Chinesta F, Falco A (2010) On the convergence of a greedy rank-one update algorithm for a class of linear systems. Arch Comput Methods Eng (submitted). doi:10.1007/s11831-010-9048-z
22. Donea J, Huerta A (2002) Finite element methods for flow problems. Wiley, New York
23. Fang J, Kroger M, Ottinger HC (2000) A thermodynamically admissible reptation model for fast flows of entangled polymers. II. Model predictions for shear and extensional flows. J Rheol 40:1293–1318
24. Fish J, Guttal R (1996) The s-version of the finite element method for laminated composites. Int J Numer Methods Eng 39:3641–3662
25. Gonzalez D, Ammar A, Chinesta F, Cueto E (2010) Recent advances in the use of separated representations. Int J Numer Methods Eng (in press)
26. Heinrich JC, Zienkiewicz OC (1979) The finite element method and upwinding techniques in the numerical solution of convection dominated flow problems. In: AMD, vol 34. Am Soc Mech Eng, New York, pp 105–136
27. Hughes TJR, Brooks AN (1979) A multidimensional upwind scheme with no crosswind diffusion. In: Hughes TJR (ed) Finite element methods for convection dominated flows. AMD, vol 34. American Society of Mechanical Engineering, New York
28. Joyot P, Bonithon G, Chinesta F, Villon P (2010) Non-incremental boundary element discretization of parabolic models based on the use of proper generalized decompositions. Eng Anal Bound Elements (submitted). doi:10.1016/j.enganabound.2010.07.007
29. Ladeveze P (1999) Nonlinear computational structural mechanics. Springer, New York
30. Ladeveze P, Passieux JCh, Neron D (2010) The LATIN multiscale computational method and the proper orthogonal decomposition. Comput Methods Appl Mech Eng 199(21–22):1287–1296
31. Levy A, Le Corre S, Poitou A, Soccard E (2010) Ultrasonic welding of thermoplastic composites, modeling of the process using time homogenization. Int J Multiscale Comput Eng (in press)
32. Nouy A, Ladeveze P (2004) Multiscale computational strategy with time and space homogenization: A radial-type approximation technique for solving microproblems. Int J Multiscale Comput Eng 170(2)
33. Maday Y, Ronquist EM (2004) The reduced basis element method: application to a thermal fin problem. SIAM J Sci Comput 26(1):240–258
34. Mokdad B, Pruliere E, Ammar A, Chinesta F (2007) On the simulation of kinetic theory models of complex fluids using the Fokker-Planck approach. Appl Rheol 17(2):1–14
35. Mokdad B, Ammar A, Normandin M, Chinesta F, Clermont JR (2010) A fully deterministic micro-macro simulation of complex flows involving reversible network fluid models. Math Comput Simul 80:1936–1961
36. Niroomandi S, Alfaro I, Cueto E, Chinesta F (2008) Real-time deformable models of non-linear tissues by model reduction techniques. Comput Methods Programs Biomed 91:223–231
37. Niroomandi S, Alfaro I, Cueto E, Chinesta F (2010) Model order reduction for hyperelastic materials. Int J Numer Methods Eng 81(9):1180–1206
38. Niroomandi S, Alfaro I, Cueto E, Chinesta F (2010) Accounting for large deformations in real-time simulations of soft tissues based on reduced order models. Comput Methods Programs Biomed (submitted)
39. Park HM, Cho DH (1996) The use of the Karhunen-Loève decomposition for the modelling of distributed parameter systems. Chem Eng Sci 51:81–98
40. Pruliere E, Ammar A, El Kissi N, Chinesta F (2009) Recirculating flows involving short fiber suspensions: Numerical difficulties and efficient advanced micro-macro solvers. Arch Comput Methods Eng State Art Rev 16:1–30
41. Pruliere E, Ferec J, Chinesta F, Ammar A (2010) An efficient reduced simulation of residual stresses in composites forming processes. Int J Mater Form 3(2):1341–1352
42. Pruliere E, Chinesta F, Ammar A (2010) On the deterministic solution of parametric models by using the proper generalized decomposition. Math Comput Simul (submitted). doi:10.1016/j.matcom.2010.07.015
43. Rank E, Krause R (1987) A multiscale finite element method. Comput Struct 64(1–4):139–144
44. Rassias TM, Simsa J (1995) Finite sums decompositions in mathematical analysis. Wiley, New York
45. Rvachev VL, Sheiko TI (1995) R-functions in boundary value problems in mechanics. Appl Mech Rev 48:151–188
46. Ryckelynck D, Hermanns L, Chinesta F, Alarcon E (2005) An efficient “a priori” model reduction for boundary element models. Eng Anal Bound Elem 29:796–801
47. Ryckelynck D (2005) A priori hyperreduction method: an adaptive approach. J Comput Phys 202:346–366
48. Ryckelynck D, Chinesta F, Cueto E, Ammar A (2006) On the a priori model reduction: overview and recent developments. Arch Comput Methods Eng State Art Rev 13(1):91–128
49. Sukumar N, Chopp D, Moes N, Belytschko T (2001) Modeling holes and inclusions by level sets in the extended finite element method. Comput Methods Appl Mech Eng 190:6183–6200