An overview of the proper generalized decomposition with applications in computational rheology

F. Chinesta[a,*], A. Ammar[b], A. Leygue[a], R. Keunings[c]

[a] *EADS Corporate Foundation International Chair, GEM, UMR CNRS-Centrale Nantes, 1 rue de la Noe, BP 92101, F-44321 Nantes Cedex 3, France*
[b] *Arts et Métiers ParisTech, 2 Boulevard du Ronceray, BP 93525, F-49035 Angers Cedex 01, France*
[c] *ICTEAM, Université catholique de Louvain, Bat. Euler, Av. Georges Lemaitre 4, B-1348 Louvain-la-Neuve, Belgium*

ABSTRACT

We review the foundations and applications of the proper generalized decomposition (PGD), a powerful model reduction technique that computes *a priori* by means of successive enrichment a separated representation of the unknown field. The computational complexity of the PGD scales linearly with the dimension of the space wherein the model is defined, which is in marked contrast with the exponential scaling of standard grid-based methods. First introduced in the context of computational rheology by Ammar et al. [3,4], the PGD has since been further developed and applied in a variety of applications ranging from the solution of the Schrödinger equation of quantum mechanics to the analysis of laminate composites. In this paper, we illustrate the use of the PGD in four problem categories related to computational rheology: (i) the direct solution of the Fokker-Planck equation for complex fluids in configuration spaces of high dimension, (ii) the development of very efficient non-incremental algorithms for transient problems, (iii) the fully three-dimensional solution of problems defined in degenerate plate or shell-like domains often encountered in polymer processing or composites manufacturing, and finally (iv) the solution of multidimensional parametric models obtained by introducing various sources of problem variability as additional coordinates.

Keywords:
Complex fluids
Numerical modeling
Model reduction
Proper orthogonal decomposition
Proper generalized decomposition
Kinetic theory
Parametric models
Optimization
Inverse identification

1. Introduction

The direct solution of many problems in scientific computing has long been thought intractable in view of the so-called curse of dimensionality. Consider for example the quantum-mechanical description of a physical system made of N particles. The evolution of the associated wavefunction is governed by the Schrödinger equation (or its relativistic Dirac counterpart). This defines a transient problem to be solved in a space of dimension $d = 3N + 1$. A typical grid-based discretization with M nodes for each coordinate would yield a total number of discrete unknowns of order M^d. A rather coarse discretization ($M = 10^3$) of a modest atomic system ($d = 30$) would thus involve 10^{90} unknowns. This is a gigantic number indeed, larger than the presumed number 10^{80} of elementary particles in the universe, according to the estimate put forward in the 1920s by the famous astronomer A.S. Eddington.

Problems defined in high-dimensional spaces abound. For example, the atomistic and mesoscopic models of theoretical rheology usually involve a large number of configurational coordinates. They thus also constitute a rich source of mathematical problems defined in high-dimensional spaces. In particular, coarse-grained models of kinetic theory result in a Fokker-Planck equation for the distribution function that must be solved in both configuration space, physical space and temporal domain. Until recently, the *direct* numerical solution of the Fokker-Planck equation has been limited to models having but few (2 or 3) configurational degrees of freedom (see e.g. the review [22]).

In two recent papers [3,4], we have proposed a technique able to circumvent, or at least alleviate, the curse of dimensionality. This method is based on the use of separated representations. It basically consists in constructing by successive enrichment an approximation of the solution in the form of a finite sum of N functional products involving d functions of each coordinate. In contrast with the shape functions of classical methods, these individual functions are unknown *a priori*. They are obtained by introducing the approximate separated representation into the weak formulation of the original problem and solving the resulting non-linear equations. If M nodes are used to discretize each coordinate, the total number of unknowns amounts to $N \times M \times d$ instead of the M^d degrees of free-

[*] Corresponding author.
E-mail addresses: Francisco.Chinesta@ec-nantes.fr (F. Chinesta), Amine.Ammar@ensam.eu (A. Ammar), Adrien.Leygue@ec-nantes.fr (A. Leygue), Roland.Keunings@uclouvain.be (R. Keunings).

dom of classical mesh-based methods. Thus, the complexity of the method grows linearly with the dimension d of the space wherein the problem is defined, in vast contrast with the exponential growth of classical mesh-based techniques.

In [3], for example, this new technique has allowed us to compute solutions of the Fokker-Planck equation in configuration spaces of dimension 20 using the multi-bead-FENE spring model of dilute polymer solutions.

The method was later coined proper generalized decomposition (or PGD), as in many cases the number N of terms in the separated representation needed to obtain an accurate solution is found to be close to that of the optimal decomposition obtained by applying *a posteriori* the proper orthogonal decomposition to the problem solution.

The goal of the present review paper is twofold. First, we wish to describe the PGD with sufficient detail and generality in order to allow the interested reader to grasp its main features and to implement it for her or his particular application. Second, we illustrate recent developments of the PGD for the solution of four problem categories that are typical of computational rheology: (1) the direct solution of the Fokker-Planck equation for complex fluids in configuration spaces of high dimension, (2) the development of very efficient non-incremental algorithms for transient problems, (3) the fully three-dimensional solution of problems defined in degenerate plate or shell-like domains often encountered in polymer processing or composites manufacturing, and finally (4) the solution of multidimensional parametric models obtained by introducing various sources of problem variability as additional coordinates. We also point to the recent literature where other applications of the PGD have been reported.

Use of the PGD is by no means restricted to computational rheology. In fact, each of the above problem categories instantiates a significant challenge in scientific computing that the PGD can address efficiently whereas standard techniques either cannot be used at all or are computationally very expensive indeed:

(1) Quantum mechanics and molecular modeling of complex fluids are not the only branches of science that suffer from the curse of dimensionality. Consider for example a chemical process involving so few molecules of the reacting species that use of the continuum concept of concentration is not valid. This situation is often found in genetic processes such as expression of genes. The state of such a discrete system is given by a probability distribution for the number of individual molecules of each one of the d coexisting species. The balance equation governing the evolution of the system, the so-called master equation, is again defined in a high-dimensional space that prevents direct solution by means of standard grid-based techniques. There are of course alternative methods to address these high-dimensional problems indirectly, stochastic simulations being one of the foremost approaches. Stochastic techniques have their own challenges, however. While variance reduction is always an issue, it is impossible with a stochastic technique to implement parametric or sensitivity studies that go beyond the brute force approach of computing a large number of expensive, individual simulations.

(2) The second problem category involves time-dependent problems not necessarily defined in high-dimensional spaces, but whose spectrum of characteristic times is so wide that standard incremental time discretization techniques cannot be applied. In such cases, the time step is extremely small as a consequence of numerical stability requirements. Thus, simulations over the much larger time interval of interest, which typically requires the solution of a large linear algebraic system at each time step, simply become impossible. Multiscale models involving a wide range of characteristic times abound in many fields. Reaction-diffusion models of the degradation of plastic materials, for example, describe chemical reactions occurring within microseconds coupled to diffusion of chemical substances that takes place over years.

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

(4) Finally, many problems in process control, parametric modeling, inverse identification, and process or shape optimisation, usually require, when approached with standard techniques, the direct computation of a very large number of solutions of the concerned model for particular values of the problem parameters. Consider for example the optimization of a pultrusion process where optimal parameter values must be determined for process operating conditions (e.g. pultrusion speed, position and temperature of heaters) and material properties (e.g. thermal and rheological properties of the resin). Clearly, it would be useful to be able to simulate this process *at once* for *all* values of these parameters within a prescribed interval, and then perform data mining within this rather general solution to identify optimal values. As we shall see, this can be achieved with the PGD by viewing all sources of problem variability as additional coordinates of a higher-dimensional problem.

The paper is organized as follows. We begin with a brief discussion of model reduction and illustrate the use of the standard POD technique to build a reduced-order model *a posteriori*. The PGD is then described at a glance in Section 3. Technical details are given in Section 4 for the solution of a parametric heat transfer problem. The four problem categories are further discussed in Section 5, and their individual PGD treatment is illustrated in the four subsequent sections.

2. Model reduction: information versus relevant information

Consider a mesh having M nodes, and associate to each node an approximation function (e.g. a shape function in the framework of finite elements), we implicitly define an approximation space wherein a discrete solution of the problem is sought. For a transient problem, one must thus compute at each time step M values (the nodal values in the finite element framework). For non-linear problems, this implies the solution of at least one linear algebraic system of size M at each time step, which becomes computationally expensive when M increases.

In many cases, however, the problem solution lives in a subspace of dimension much smaller than M, and it makes sense to look for a reduced-order model whose solution is computationally much cheaper to obtain. This constitutes the main idea behind the proper orthogonal decomposition (POD) reduced modeling approach, which we revisit in what follows.

2.1. Extracting relevant information: the proper orthogonal decomposition

We assume that the field of interest $u(\mathbf{x}, t)$ is known at the nodes $\mathbf{x}_i$ of a spatial mesh for discrete times $t_m = m \cdot \Delta t$, with $i \in [1, \ldots, M]$ and $m \in [0, \ldots P]$. We use the notation $u(\mathbf{x}_i, t_m) \equiv u^m(\mathbf{x}_i) \equiv u_i^m$ and define $\{\mathbf{u}\}^m$ as the vector of nodal values u_i^m at time t_m. The main objective of the POD is to obtain the most typical or characteristic structure $\phi(\mathbf{x})$ among these $u^m(\mathbf{x})$, $\forall m$ [39]. For this purpose, we maximize the scalar quantity

$$\alpha = \frac{\displaystyle\sum_{m=1}^{P} \left[\sum_{i=1}^{M} \phi(\mathbf{x}_i) u^m(\mathbf{x}_i) \right]^2}{\displaystyle\sum_{i=1}^{M} (\phi(\mathbf{x}_i))^2}, \tag{1}$$

which amounts to solve the following eigenvalue problem:

$$\mathbf{c}\boldsymbol{\phi} = \alpha \boldsymbol{\phi}. \tag{2}$$

Here, the vector $\boldsymbol{\phi}$ has i-component $\phi(\mathbf{x}_i)$, and $\{\mathbf{c}\}$ is the two-point correlation matrix

$$c_{ij} = \sum_{m=1}^{P} u^m(\mathbf{x}_i) u^m(\mathbf{x}_j) = \sum_{m=1}^{P} \mathbf{u}^m \cdot (\mathbf{u}^m)^T, \tag{3}$$

which is symmetric and positive definite. With the matrix $\{\mathbf{Q}\}$ defined as

$$\mathbf{Q} = \begin{pmatrix} u_1^1 & u_1^2 & \cdots & u_1^P \\ u_2^1 & u_2^2 & \cdots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_M^1 & u_M^2 & \cdots & u_M^P \end{pmatrix}, \tag{4}$$

we have

$$\mathbf{c} = \mathbf{Q} \cdot \mathbf{Q}^T. \tag{5}$$

2.2. Building the POD reduced-order model

In order to obtain a reduced model, we first solve the eigenvalue problem Eq. (2) and select the N eigenvectors $\boldsymbol{\phi}_i$ associated with the eigenvalues belonging to the interval defined by the highest eigenvalue α_1 and α_1 divided by a large enough number (e.g. 10^8). In practice, N is found to be much lower than M. These N eigenfunctions $\boldsymbol{\phi}_i$ are then used to approximate the solution $u^m(\mathbf{x})$, $\forall m$. To this end, let us define the matrix $\{\mathbf{B}\} = [\boldsymbol{\phi}_1 \ldots \boldsymbol{\phi}_N]$, i.e.

$$\mathbf{B} = \begin{pmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_M) & \phi_2(\mathbf{x}_M) & \cdots & \phi_N(\mathbf{x}_M) \end{pmatrix}. \tag{6}$$

Now, let us assume for illustrative purposes that an explicit time-stepping scheme is used to compute the discrete solution $\{\mathbf{u}\}^{m+1}$ at time t^{m+1}. One must thus solve a linear algebraic system of the form

$$\mathbf{G}^m \, \mathbf{u}^{m+1} = \mathbf{H}^m. \tag{7}$$

A reduced-order model is then obtained by approximating $\{\mathbf{u}\}^{m+1}$ in the subspace defined by the N eigenvectors $\boldsymbol{\phi}_i$, i.e.

$$\mathbf{u}^{m+1} \approx \sum_{i=1}^{N} \boldsymbol{\phi}_i \, \zeta_i^{m+1} = \mathbf{B} \, \boldsymbol{\zeta}^{m+1}. \tag{8}$$

Eq. (7) then reads

$$\mathbf{G}^m \, \mathbf{B} \, \boldsymbol{\zeta}^{m+1} = \mathbf{H}^m, \tag{9}$$

or equivalently

$$\mathbf{B}^T \mathbf{G}^m \, \mathbf{B} \, \boldsymbol{\zeta}^{m+1} = \mathbf{B}^T \mathbf{H}^m. \tag{10}$$

The coefficients $\boldsymbol{\zeta}^{m+1}$ defining the solution of the reduced-order model are thus obtained by solving an algebraic system of size N instead of M. When $N \ll M$, as is the case in numerous applications, the solution of Eq. (10) is thus preferred because of its much reduced size.

Remark 1. The reduced-order model Eq. (10) is built a posteriori by means of the already-computed discrete field evolution. Thus, one could wonder about the interest of the whole exercice. In fact, two beneficial approaches are widely considered (see e.g. [10,13,21,28,34,37–39]). The first approach consists in solving the large original model over a short time interval, thus allowing for the extraction of the characteristic structure that defines the reduced model. The latter is then solved over larger time intervals, with the associated computing time savings. The other approach consists in solving the original model over the entire time interval, and then using the corresponding reduced model to solve very efficiently similar problems with, for example, slight variations in material parameters or boundary conditions.

2.3. Illustrating the construction of a reduced-order model

We consider the following one-dimensional heat transfer problem, written in dimensionless form:

$$\frac{\partial u}{\partial t} = \lambda \frac{\partial^2 u}{\partial x^2}, \tag{11}$$

with constant thermal diffusivity $\lambda = 0.01$, $t \in (0, 30]$ and $x \in (0, 1)$. The initial condition is $u(x, t = 0) = 1$ and the boundary conditions are given by $\left.\frac{\partial u}{\partial x}\right|_{x=0,t} = q(t)$ and $\left.\frac{\partial u}{\partial x}\right|_{x=1,t} = 0$.

Eq. (11) is discretized by using an implicit finite element method on a mesh with $M = 100$ nodes, where a linear approximation is defined in each of the $M_e = 99$ elements. The time step is set to $\Delta t = 0.1$. The resulting discrete system can be written as:

$$\mathbf{K} \, \mathbf{u}^{m+1} = \mathbf{M} \, \mathbf{u}^m + \mathbf{q}^{m+1}, \tag{12}$$

where the vector $\{\mathbf{q}\}^{m+1}$ accounts for the boundary heat flux source at t_{m+1}.

First, we consider the following boundary heat source:

$$q(t) = \begin{cases} 1 & t \le 10 \\ 0 & t > 10 \end{cases}. \tag{13}$$

The computed temperature profiles are depicted in Fig. 1 at discrete times $t_m = m$, for $m = 1, 2, \ldots, 30$. The red curves correspond to the heating stage up to $t = 10$, while the blue curves for $t > 10$ illustrate the heat transfer by conduction from the warmest zones towards the coldest ones.

From these 30 discrete temperature profiles, we compute the matrices $\{\mathbf{Q}\}$ and $\{\mathbf{c}\}$ in order to build the eigenvalue problem (2). The 3 largest eigenvalues are found to be $\alpha_1 = 1790, \alpha_2 = 1.1, \alpha_3 = 0.1$, while the remaining eigenvalues are such that $\alpha_j < \alpha_1 \times 10^{-8}$, $4 \le j \le 100$. A reduced model involving a linear combination of the 3 eigenvectors related to the first 3 largest eigenvalues should thus be able to approximate the solution with great accuracy. In order to account for the initial condition, it is convenient to include the initial condition in the approximation basis (even though it is then no longer orthogonal). Fig. 2 shows the resulting approximation func-

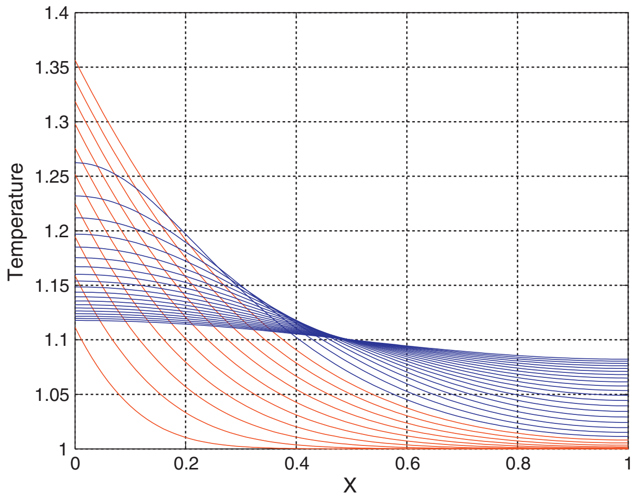

Fig. 1. Temperature profiles corresponding to the source term (13) at discrete times $t_m = m$, for $m = 1, 2, \ldots, 30$. The red curves correspond to the heating stage up to $t = 10$, while the blue curves for $t > 10$ illustrate the heat transfer by conduction from the warmest zones towards the coldest ones. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

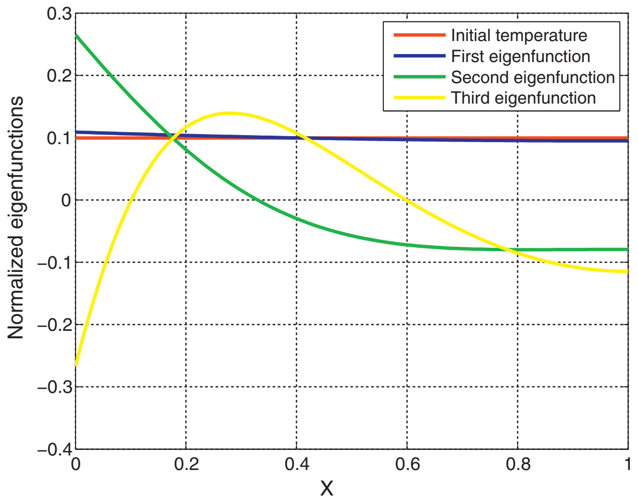

Fig. 2. Reduced-order approximation basis involving the initial condition and the eigenvectors corresponding to the three largest eigenvalues.

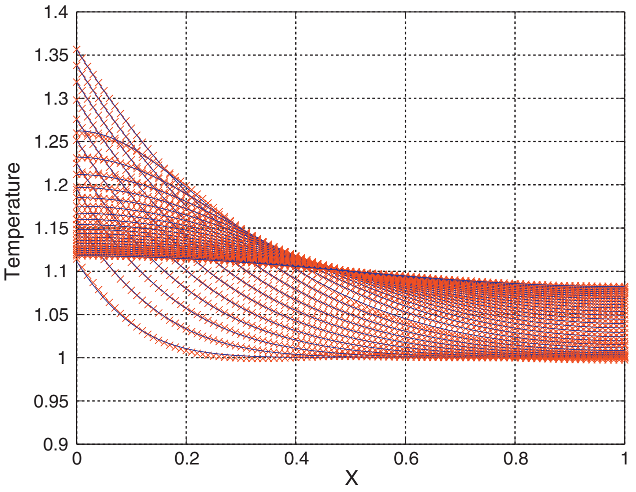

Fig. 3. Global (continuous line) versus reduced-order (symbols) model solutions.

tions in normalized form, i.e. $\frac{\boldsymbol{\phi}_j}{\|\boldsymbol{\phi}_j\|}$ $(j = 1, 2, 3)$ and $\frac{\mathbf{u}^0}{\|\mathbf{u}^0\|}$. Defining the matrix $\{\mathbf{B}\}$ as follows

$$\mathbf{B} = \left[\frac{\mathbf{u}^0}{\|\mathbf{u}^0\|} \; \frac{\boldsymbol{\phi}_1}{\|\boldsymbol{\phi}_1\|} \; \frac{\boldsymbol{\phi}_2}{\|\boldsymbol{\phi}_2\|} \; \frac{\boldsymbol{\phi}_3}{\|\boldsymbol{\phi}_3\|} \right], \tag{14}$$

we obtain the reduced model related to Eq. (12),

$$\mathbf{B}^T\mathbf{K}\mathbf{B}\,\boldsymbol{\zeta}^{m+1} = \mathbf{B}^T\mathbf{M}\mathbf{B}\,\boldsymbol{\zeta}^m + \mathbf{B}^T\mathbf{q}^{m+1}, \tag{15}$$

which involves 4 degrees of freedom only. The initial condition in the reduced basis is $(\boldsymbol{\zeta}^0)^T = (1, 0, 0, 0)$.

Eq. (15) and the relationship $\{\mathbf{u}\}^{m+1} = \{\mathbf{B}\}\boldsymbol{\zeta}^{m+1}$ then yield approximate solution profiles at a very low cost indeed. The results are shown in Fig. 3 and they cannot be distinguished at the scale of the drawing from those of the complete problem (12).

In order to illustrate our Remark 1, let us now use the reduced model (15) *as such* to solve a problem *different* from the one that served to derive it. While keeping all other specifications identical, we now impose instead of (13) a substantially different boundary heat source:

$$q(t) = \begin{cases} \dfrac{t}{20} & t \leq 20 \\ \dfrac{t-30}{5} & t > 20 \end{cases}. \tag{16}$$

The solution of the reduced model is compared to that of the complete problem in Fig. 4. Even though the reduced approximation basis functions are those obtained from the thermal model related to the boundary condition (13), the reduced model yields a very accurate representation of the solution of this rather different problem.

2.4. Discussion

The above example illustrates the significant value of model reduction. Of course, one would ideally want to be able to build a reduced-order approximation *a priori*, i.e. without relying on the knowledge of the (approximate) solution of the complete problem. One would then want to be able to assess the accuracy of the reduced-order solution and, if necessary, to enrich the reduced approximation basis in order to improve accuracy (see e.g. our ear-

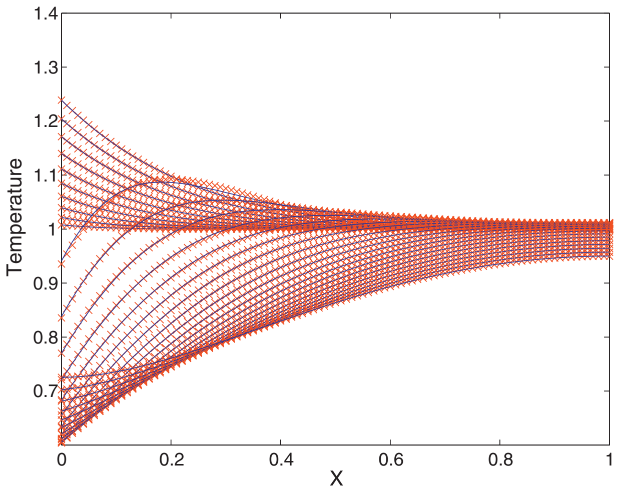

Fig. 4. Global (continuous line) versus reduced-order (symbols) model solutions for the source term (16). The reduced-order approximation basis is that obtained from the solution of a different thermal problem, with the source term (13).

lier studies [2,5,39]). The proper generalized decomposition (PGD), which we describe in general terms in the next section, is an efficient answer to these questions.

The above POD results also tell us that an accurate approximate solution can often be written as a separated representation involving but few terms. Indeed, when the field evolves smoothly, the magnitude of the (ordered) eigenvalues α_i decreases very fast with increasing index i, and the evolution of the field can be approximated from a reduced number of modes. Thus, if we define a cutoff value ϵ (e.g. $\epsilon = 10^{-8} \cdot \alpha_1$, α_1 being the highest eigenvalue), only a small number N of modes are retained ($N << M$) such that $\alpha_i \geq \epsilon$, for $i \leq N$, and $\alpha_i < \epsilon$, for $i > N$. Thus, one can write:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} \phi_i(\mathbf{x}) \cdot T_i(t) \equiv \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t). \tag{17}$$

For the sake of clarity, the space modes $\phi_i(\mathbf{x})$ will be denoted in the sequel as $X_i(\mathbf{x})$. Eq. (17) represents a natural *separated representation*, also known as finite sum decomposition. The solution that depends on space and time can be approximated as a sum of a *small number* of functional products, with one of the functions depending on the space coordinates and the other one on time. Use of separated representations like (17) is at the heart of the PGD.

Thus, we expect that the transient solution of numerous problems of interest can be expressed using a very reduced number of functional products involving each a function of time and a function of space. Ideally, the functions involved in these functional products should be determined simultaneously by applying an appropriate algorithm to guarantee robustness and optimality; in view of the non-linear nature of the separated representation, this will require a suitable iterative process.

To our knowledge, the unique precedent to the PGD algorithm for building a separated space–time representation is the so-called radial approximation introduced by Ladeveze ([23,24,31]) in the context of computational solid mechanics.

In terms of performance, the verdict is simply impressive. Consider a typical transient problem defined in 3 D physical space. Use of a standard incremental strategy with P time steps (P is of order of millions in industrial applications) requires the solution of P three-dimensional problems. By contrast, using the space-time separated representation (17), we must solve $N \cdot m$ three-dimensional problems for computing the space functions $X_i(\mathbf{x})$, and $N \cdot m$ one-dimensional problems for computing the time functions $T_i(t)$. Here, m is the number of non-linear iterations needed for computing each term of the finite sum. For many problems of practical interest, we find that $N \cdot m$ is of order 100. The computing time savings afforded by the separated representation can thus reach many orders of magnitude.

3. The proper generalized decomposition at a glance

Consider a problem defined in a space of dimension d for the unknown field $u(x_1, \ldots, x_d)$. Here, the coordinates x_i denote any usual coordinate (scalar or vectorial) related to physical space, time, or conformation space, for example, but they could also include problem parameters such as boundary conditions or material parameters. We seek a solution for $(x_1, \ldots, x_d) \in \Omega_1 \times \cdots \times \Omega_d$.

The PGD yields an approximate solution in the separated form:

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{N} F_i^1(x_1) \times \cdots \times F_i^d(x_d). \tag{18}$$

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

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

In many applications studied to date, N is found to be as small as a few tens, and in all cases the approximation converges towards the solution associated with the complete tensor product of the approximation bases considered in each Ω_j. Thus, we can be confident about the generality of the separated representation (18), but its optimality depends on the solution regularity. When an exact solution of a particular problem can be represented with enough accuracy by a reduced number of functional products, the PGD approximation is optimal. If the solution is a non-separable function for the particular coordinate system used, the PGD solver proceeds to enrich the approximation until including all the elements of the functional space, i.e. the M^d functions involved in the full tensor product of the approximation bases in each Ω_j.

Let us now consider in more detail a specific example.

4. Proper generalized decomposition of a generic parametric model

In this section, we illustrate the PGD by considering the following parametric heat transfer equation:

$$\frac{\partial u}{\partial t} - k\Delta u - f = 0, \tag{19}$$

with homogeneous initial and boundary conditions. Enforcement of non-homogeneous initial and boundary conditions is discussed in [20,17].

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

The weak form related to Eq. (19) reads:

$$\int_{\Omega \times I \times \Im} u^* \cdot \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k = 0, \tag{20}$$

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

The PGD solution is sought in the form:

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

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

$$u^n\left(\mathbf{x}, t, k\right) = \sum_{i=1}^{n} X_i\left(\mathbf{x}\right) \cdot T_i\left(t\right) \cdot K_i\left(k\right). \tag{22}$$

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

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

$$u^{n+1} = u^n + R(\mathbf{x}) \cdot S(t) \cdot W(k). \tag{23}$$

We propose the simplest choice for the test functions u^* used in Eq. (20):

$$u^* = R^*(\mathbf{x}) \cdot S(t) \cdot W(k) + R(\mathbf{x}) \cdot S^*(t) \cdot W(k) + R(\mathbf{x}) \cdot S(t) \cdot W^*(k). \tag{24}$$

With the trial and test functions given by Eqs. (23) and (24) respectively, Eq. (20) is a non-linear problem that must be solved by means of a suitable iterative scheme. In our earlier papers [3,4], we used Newton's method. Simpler linearization strategies can also be applied, however. The simplest one is an alternating direction, fixed-point algorithm, which was found remarkably robust in the present context. Each iteration consists of three steps that are repeated until reaching convergence, that is, until reaching the fixed point. The first step assumes $S(t)$ and $W(k)$ known from the previous iteration and compute an update for $R(\mathbf{x})$ (in this case the test function reduces to $R^*(\mathbf{x}) \cdot S(t) \cdot W(k)$). From the just-updated $R(\mathbf{x})$ and the previously-used $W(k)$, we can update $S(t)$ (with $u^* = R(\mathbf{x}) \cdot S^*(t) \cdot W(k)$). Finally, from the just-computed $R(\mathbf{x})$ and $S(t)$, we update $W(k)$ (with $u^* = R(\mathbf{x}) \cdot S(t) \cdot W^*(k)$). This iterative procedure continues until reaching convergence. The converged functions $R(\mathbf{x})$, $S(t)$ and $W(k)$ yield the new functional product of the current enrichment step: $X_{n+1}(\mathbf{x}) = R(\mathbf{x})$, $T_{n+1}(t) = S(t)$ and $K_{n+1}(k) = W(k)$. The explicit form of these operations is described below.

4.1. Computing $R(\boldsymbol{x})$ from $S(t)$ and $W(k)$

We consider the weak form of Eq. (19):

$$\int_{\Omega \times I \times \Im} u^* \left(\frac{\partial u}{\partial t} - k\Delta u - f \right) \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k = 0. \tag{25}$$

Here, the trial function is given by

$$u(\mathbf{x}, t, k) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \cdot K_i(k) + R(\mathbf{x}) \cdot S(t) \cdot W(k). \tag{26}$$

Since S and W are known from the previous iteration, the test function reads

$$u^*(\mathbf{x}, t, k) = R^*(\mathbf{x}) \cdot S(t) \cdot W(k). \tag{27}$$

Introducing (26) and (27) into (25) yields

$$\begin{aligned} &\int_{\Omega \times I \times \Im} R^* \cdot S \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k \\ &\quad = - \int_{\Omega \times I \times \Im} R^* \cdot S \cdot W \cdot \mathcal{R}^n\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k, \end{aligned} \tag{28}$$

where $\mathcal{R}^n$ is the residual at enrichment step n:

$$\mathcal{R}^n = \sum_{i=1}^{n} X_i \cdot \frac{\partial T_i}{\partial t} \cdot K_i - \sum_{i=1}^{n} k \cdot \Delta X_i \cdot T_i \cdot K_i - f. \tag{29}$$

Since all functions involving time and conductivity have been determined, we can integrate Eq. (28) over $I \times \Im$. With the following notations,

$$\begin{bmatrix} w_1 = \displaystyle\int_{\Im} W^2\, \mathrm{d}k & s_1 = \displaystyle\int_{I} S^2\, \mathrm{d}t & r_1 = \displaystyle\int_{\Omega} R^2\, \mathrm{d}\mathbf{x} \\ w_2 = \displaystyle\int_{\Im} kW^2\, \mathrm{d}k & s_2 = \displaystyle\int_{I} S \cdot \frac{\mathrm{d}S}{\mathrm{d}t} \mathrm{d}t & r_2 = \displaystyle\int_{\Omega} R \cdot \Delta R\, \mathrm{d}\mathbf{x} \\ w_3 = \displaystyle\int_{\Im} W\, \mathrm{d}k & s_3 = \displaystyle\int_{I} S\, \mathrm{d}t & r_3 = \displaystyle\int_{\Omega} R\, \mathrm{d}\mathbf{x} \\ w_4^i = \displaystyle\int_{\Im} W \cdot K_i\, \mathrm{d}k & s_4^i = \displaystyle\int_{I} S \cdot \frac{\mathrm{d}T_i}{\mathrm{d}t} \mathrm{d}t & r_4^i = \displaystyle\int_{\Omega} R \cdot \Delta X_i\, \mathrm{d}\mathbf{x} \\ w_5^i = \displaystyle\int_{\Im} kW \cdot K_i\, \mathrm{d}k & s_5^i = \displaystyle\int_{I} S \cdot T_i\, \mathrm{d}t & r_5^i = \displaystyle\int_{\Omega} R \cdot X_i\, \mathrm{d}\mathbf{x} \end{bmatrix}, \tag{30}$$

Eq. (28) reduces to

$$\begin{aligned} &\int_{\Omega} R^* \cdot (w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R)\, \mathrm{d}\mathbf{x} \\ &\quad = - \int_{\Omega} R^* \cdot \left(\sum_{i=1}^{n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right) \mathrm{d}\mathbf{x}. \end{aligned} \tag{31}$$

Eq. (31) defines in weak form an elliptic steady-state boundary value problem for the unknown function R that can be solved by using any suitable discretization technique (finite elements, finite volumes, ...). Another possibility consists in coming back to the strong form of Eq. (31):

$$\begin{aligned} &w_1 \cdot s_2 \cdot R - w_2 \cdot s_1 \cdot \Delta R \\ &\quad = - \left(\sum_{i=1}^{n} w_4^i \cdot s_4^i \cdot X_i - \sum_{i=1}^{n} w_5^i \cdot s_5^i \cdot \Delta X_i - w_3 \cdot s_3 \cdot f \right), \end{aligned} \tag{32}$$

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

4.2. Computing $S(t)$ from $R(\boldsymbol{x})$ and $W(k)$

In the present case, the test function is written as

$$u^*(\mathbf{x}, t, k) = S^*(t) \cdot R(\mathbf{x}) \cdot W(k), \tag{33}$$

and the weak form becomes

$$\begin{aligned} &\int_{\Omega \times I \times \Im} S^* \cdot R \cdot W \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W \right) \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k \\ &\quad = - \int_{\Omega \times I \times \Im} S^* \cdot R \cdot W \cdot \mathcal{R}^n\, \mathrm{d}\mathbf{x}\, \mathrm{d}t\, \mathrm{d}k. \end{aligned} \tag{34}$$

Integrating over $\Omega \times \Im$, one obtains

$$\begin{aligned} &\int_{I} S^* \cdot \left(w_1 \cdot r_1 \cdot \frac{\mathrm{d}S}{\mathrm{d}t} - w_2 \cdot r_2 \cdot S \right) \mathrm{d}t \\ &\quad = - \int_{I} S^* \cdot \left(\sum_{i=1}^{n} w_4^i \cdot r_5^i \cdot \frac{\mathrm{d}T_i}{\mathrm{d}t} - \sum_{i=1}^{n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f \right) \mathrm{d}t. \end{aligned} \tag{35}$$

Eq. (35) represents the weak form of the ODE defining the time evolution of the field S that can be solved by using any stabilized discretization technique (SU, Discontinuous Galerkin, ...). The strong form of Eq. (35) reads

$$w_1 \cdot r_1 \cdot \frac{\mathrm{d}S}{\mathrm{d}t} - w_2 \cdot r_2 \cdot S$$

$$= -\left(\sum_{i=1}^{n} w_4^i \cdot r_5^i \cdot \frac{\mathrm{d}T_i}{\mathrm{d}t} - \sum_{i=1}^{n} w_5^i \cdot r_4^i \cdot T_i - w_3 \cdot r_3 \cdot f\right). \tag{36}$$

Eq. (36) can be solved by using backward finite differences, or higher order Runge–Kutta schemes, among many other possibilities.

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

The test function is now given by

$$u^*(\mathbf{x}, t, k) = W^*(k) \cdot R(\mathbf{x}) \cdot S(t), \tag{37}$$

and the weak form becomes

$$\int_{\Omega\times I\times\Im} W^* \cdot R \cdot S \cdot \left(R \cdot \frac{\partial S}{\partial t} \cdot W - k \cdot \Delta R \cdot S \cdot W\right) \mathrm{d}\mathbf{x}\,\mathrm{d}t\,\mathrm{d}k$$
$$= -\int_{\Omega\times I\times\Im} W^* \cdot R \cdot S \cdot \mathcal{R}^n \,\mathrm{d}\mathbf{x}\,\mathrm{d}t\,\mathrm{d}k. \tag{38}$$

Integration over $\Omega \times I$ yields

$$\int_{\Im} W^* \cdot (r_1 \cdot s_2 \cdot W - r_2 \cdot s_1 \cdot k \cdot W)\,\mathrm{d}k$$
$$= -\int_{\Im} W^* \cdot \left(\sum_{i=1}^{n} r_5^i \cdot s_4^i \cdot K_i - \sum_{i=1}^{n} r_4^i \cdot s_5^i \cdot k \cdot K_i - r_3 \cdot s_3 \cdot f\right) \mathrm{d}k. \tag{39}$$

Eq. (39) does not involve any differential operator. The corresponding strong form reads

$$(r_1 \cdot s_2 - r_2 \cdot s_1 \cdot k) \cdot W$$
$$= -\left(\sum_{i=1}^{n} \left(r_5^i \cdot s_4^i - r_4^i \cdot s_5^i \cdot k\right) \cdot K_i - r_3 \cdot s_3 \cdot f\right). \tag{40}$$

This is an algebraic problem, which is hardly a surprise since the original Eq. (19) does not contain derivatives with respect to the parameter k. Introduction of the parameter k as additional model coordinate does not increase the cost of a particular enrichment step. It does however necessitate more enrichment steps, i.e. more terms (higher N) in the decomposition (21).

We have seen that at each enrichment step the construction of the new functional product in Eq. (21) requires non-linear iterations. If m_i denotes the number of iterations needed at enrichment step i, the total number of iterations involved in the construction of the PGD approximation is $m = \sum_{i=1}^{i=N} m_i$. In the above example, the entire procedure thus involves the solution of m three-dimensional problems for the functions $X_i(\mathbf{x})$, m one-dimensional problems for the functions $T_i(t)$ and m algebraic systems for the functions $K_i(k)$. In general, m rarely exceeds 10. The number N of functional products needed to approximate the solution with enough accuracy depends on the solution regularity. All numerical experiments carried to date reveal that N ranges between a few tens and one hundred. Thus, we can conclude that the complexity of the PGD procedure to compute the approximation (21) is of some tens of 3 D steady-state problems (the cost related to the 1 D and algebraic problems being negligible with respect to the 3 D problems). In a classical approach, one must solve for each particular value of the parameter k a 3 D problem at each time step. In usual applications, this often implies the computation of several millions of 3 D solutions. Clearly, the CPU time savings by applying the PGD can be of several orders of magnitude.

5. Pushing forward simulation limits with the PGD

Before considering in some detail a number of applications of the PGD, let us briefly revisit the four challenges discussed in the introductory section.

(1) High-dimensional problems are readily handled by invoking the PGD separated representation:

$$u(x_1, \ldots, x_d) \approx \sum_{i=1}^{N} F_i^1(x_1) \times \cdots \times F_i^d(x_d), \tag{41}$$

with $(x_1, \ldots, x_d) \in \Omega_1 \times \cdots \times \Omega_d$. The PGD procedure is identical to that described in the previous section. At each enrichment step, introduction of the separated representation into the problem's weak form and use of non-linear iterations yield the equations required for computing each one of the functions $F_i^j(x_j)$ in their respective, low-dimensional domain Ω_j.

This strategy was successfully applied in our studies of the kinetic theory description of complex fluids. A multidimensional separated representation of the linear steady-state Fokker-Planck equation was introduced in the seminal work [3] and later in [27], further extended to transient simulations in [4] and non-linear Fokker-Planck equations in [29]. In [30,35], we considered the solution of Fokker-Planck equations in complex flows, where space, time and conformation coordinates coexist. We have also applied the same approach for solving the Schrödinger equation [16], the chemical master equation [19] or kinetic theory models formulated within the Brownian configurations fields framework [15].

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

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

The constructor of that separated representation was illustrated in the previous section (it suffices to ignore the existence of the parametric extra-coordinate). We cited previously the pioneering works of Ladeveze's team in the field of structural mechanics. Space-time separated representations were also considered in the context of computational rheology in [6]. In [18], they were applied to the multiscale coupling of diffusion and kinetic models endowed with very different characteristic times.

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

Complete decomposition:

$$u(\mathbf{x}, t) \approx \sum_{i=1}^{N} X_i(x) \cdot Y_i(y) \cdot Z_i(z). \tag{43}$$

This strategy is particularly suitable for separable domains, i.e. $\Xi = \Omega_x \times \Omega_y \times \Omega_z$. For general domains, embedding Ξ into a larger separable domain $\Omega_x \times \Omega_y \times \Omega_z$ can also be done, as described in [20].

Plate-type decomposition:

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

This strategy is particularly suitable when $\Xi = \Omega \times \mathcal{I}$, with $\Omega \subset \mathcal{R}^2$ and $\mathcal{I} \subset \mathcal{R}$. More complex domains (e.g. plates with a varying

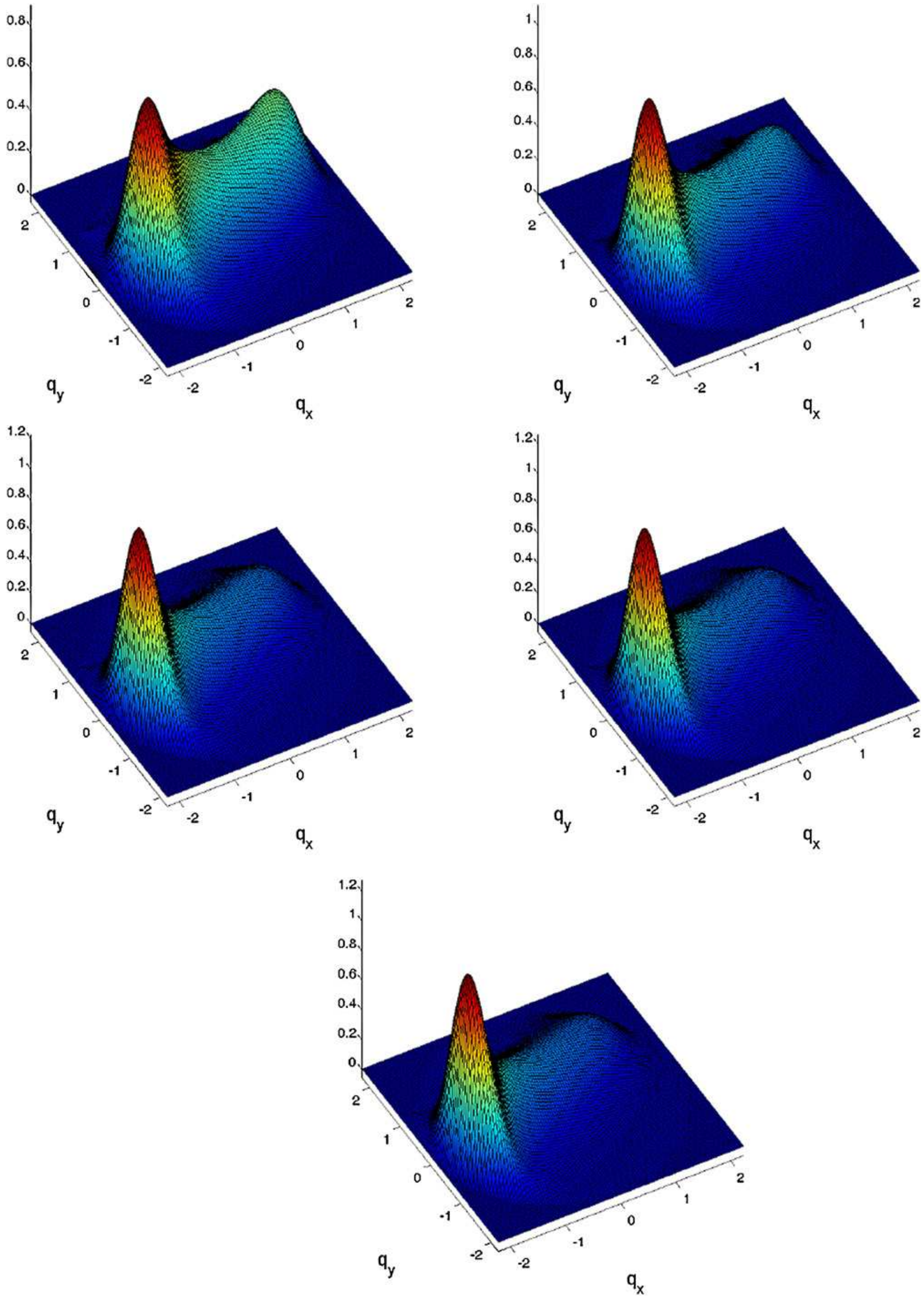

Fig. 5. MBS model consisting of 10 FENE springs in a two-dimensional physical space. First reduced approximation functions: (top-left) first spring; (top-right) second spring; (middle-left) third spring; (middle-right) fourth spring and (bottom) central spring.

thickness) can be treated by using an appropriate change of variable.

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

$$u(\mathbf{x}, t, p_1, \ldots, p_Q) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t) \cdot P_i^1(p_1) \cdots P_i^Q(p_Q), \qquad (45)$$

where the p_i s denote the different problem parameters such as material parameters, boundary conditions, applied loads, initial conditions, and geometrical parameters [36].

In the remaining sections, we illustrate each of these problem categories.

6. Solution of the multidimensional Fokker-Planck equation

6.1. Multi-bead-spring FENE model of dilute polymer solutions

The MBS FENE chain consists of $d+1$ beads connected by d non-linear entropic springs. For homogeneous flows, the molecular conformations are described by a probability distribution $\Psi(\mathbf{q}_1,$

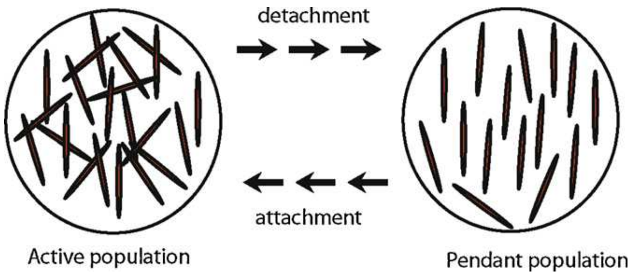

Fig. 6. Flow induced aggregation/disaggregation mechanisms.

$\ldots, \mathbf{q}_d, t)$ governed by the Fokker-Planck equation

$$\frac{\partial \Psi(\mathbf{q}_1, \ldots, \mathbf{q}_d, t)}{\partial t} = -\sum_{k=1}^{d} \left(\frac{\partial}{\partial \mathbf{q}_k} \left(\dot{\mathbf{q}}_k \cdot \Psi(\mathbf{q}_1, \ldots, \mathbf{q}_d, t) \right) \right), \tag{46}$$

with a suitable governing equation for $\dot{\mathbf{q}}_k$ [11]. In order to demonstrate the applicability of the PGD, we considered in [3] steady-state, simple shear flow calculations for chains with 10 springs having 2D orientation. The distribution function, thus defined in a 20 D space, is sought in the separated form

$$\Psi(\mathbf{q}_1, \ldots, \mathbf{q}_{10}) \approx \sum_{j=1}^{N} F_j^1(\mathbf{q}_1) \times \cdots \times F_j^{10}(\mathbf{q}_{10}). \tag{47}$$

A mesh consisting of 10^4 nodes was used for approximating each function $F_j^k(\mathbf{q}_k)$. A simple shear flow was applied ($We = \sqrt{2}$) and 3 terms in the separated representation were found sufficient to accurately describe the steady-state distribution function. We depict in Fig. 5 the functions defining the first mode, i.e. functions $F_1^k(\mathbf{q}_k)$, $k = 1, \ldots, 5$. We notice that the results are sharper at the center of the chain, indicating that the central springs are more stretched than the ones located near the chain ends.

This simulation implied $10 \times 10^4 = 10^5$ degrees of freedom (10 spring connectors whose approximation functions were defined using a mesh of 10^4 nodes). A standard finite element solution would have required of the order of $(10^4)^{10} = 10^{40}$ degrees of freedom (nodes) for computing an equivalent solution.

6.2. Rod-like aggregating suspension in complex flows

We consider a suspension of rod-like particles which can flocculate to create aggregates. These aggregates are continuously broken by the flow. Thus, aggregation and disaggregation mechanisms coexist and two populations of particles can be identified: the one related to free rods (pendant population) and the one associated with the aggregated rods (active population). The model is inspired from the those developed for associative polymers. Fig. 6 depicts both populations and the flow induced aggregation/disaggregation.

The kinetic theory description of such systems contains two coupled Fokker-Planck equations for the orientation distribution $\Psi(\mathbf{x}, t, \mathbf{p})$ and $\Phi(\mathbf{x}, t, \mathbf{p})$ of the active and pendant rods, respectively [30]:

$$\frac{D\Psi}{Dt} = -\frac{\partial}{\partial \mathbf{p}} \left(\dot{\mathbf{p}} \Psi \right) + D_{r1} \frac{\partial^2 \Psi}{\partial \mathbf{p}^2} - V_d \Psi + V_c \Phi, \tag{48}$$

$$\frac{D\Phi}{Dt} = -\frac{\partial}{\partial \mathbf{p}} \left(\dot{\mathbf{p}} \Phi \right) + D_{r2} \frac{\partial^2 \Phi}{\partial \mathbf{p}^2} + V_d \Psi - V_c \Phi. \tag{49}$$

Here, $\dot{\mathbf{p}}$ is the flow induced orientation modeled by Jeffery's equation, D_{r1} and D_{r2} are the rotary diffusion coefficient of both populations taking into account Brownian effects and hydrodynamic interactions, and finally V_d and V_c are the velocity of destruction and construction of the active population, respectively. Note that a material derivative acts on the distribution functions, since we are dealing with complex flows.

We consider the flow in a converging channel. The steady-state flow kinematics (assumed undisturbed by the presence of the suspended particles) were computed by solving the Stokes equations. The Fokker-Planck equations were integrated along particular flow streamlines. The separated representation of both orientation distribution functions reads:

$$\begin{pmatrix} \Psi_{st}(s, \mathbf{p}) \\ \Phi_{st}(s, \mathbf{p}) \end{pmatrix} \approx \sum_{i=1}^{N} \begin{pmatrix} P_i^{st,\Psi}(\mathbf{p}) \cdot S_i^{st,\Psi}(s) \\ P_i^{st,\Phi}(\mathbf{p}) \cdot S_i^{st,\Phi}(s) \end{pmatrix}, \tag{50}$$

where s denotes the curvilinear coordinate along each individual streamline "*st*". For additional details, see [30].

Fig. 7 depicts the resulting orientation distribution of both populations at particular points along individual flow streamlines. The orientation distribution is directly depicted on the unit surface, and the color scheme indicates the intensity of the orientation in each direction.

7. Space-time separation for efficient transient simulations

As an illustration of the efficiency of the PGD to address transient problems, we consider the computation of the linear viscoelastic moduli G' and G'' via the direct solution of the Fokker-Planck equation for FENE dumbbells [6]. Thus, several small-amplitude oscillatory flows must be solved for a wide range of frequencies covering several decades. For each value of the applied frequency, the flow kinematics is given by

$$\nabla \mathbf{v} = \begin{pmatrix} 0 & g\sin(\omega t) \\ 0 & 0 \end{pmatrix}, \tag{51}$$

where g is the maximum shear rate and ω the applied frequency.

This constitutes a challenge for standard incremental numerical methods: many transient simulations are needed, and, for each frequency, the transient simulation must be performed over a time interval large enough to ensure the response stabilization. Moreover, the higher the frequency, the shorter the time step.

Applying the PGD, we compute the probability distribution function in the separated form

$$\Psi(\mathbf{q}, t) \approx \sum_{i=1}^{N} G_i(\mathbf{q}) \cdot F_i(t). \tag{52}$$

We use 1000 time steps uniformly distributed over 10 periods of oscillation and about 3000 nodes in Ω_q. Fig. 8 illustrates the three most significant conformation modes $G_i(\mathbf{q})$, $i = 1, 2, 3$, the first few time modes $F_i(t)$, the resulting probability distribution function (from which the probability distribution function at equilibrium $\Psi_0(\mathbf{q})$ was removed) at the final time ($t = T_{\max}$) and the shear stress for $\omega = 2.55$ and $g = 0.1$.

In this application, the PGD yields a significant reduction in CPU time (of order of hundreds) relative to standard incremental methods. The separated representation allows one to treat one or more periods using almost the same number of functions and consequently the same CPU time. The CPU time of a standard incremental method, however, would grow linearly with the number of periods. The advantage of using the PGD lies in the fact that consideration of a very fine discretization of the time axis does not affect significantly the overall simulation time since the cost of computing the functions of time $F_i(t)$ (solution of ordinary differential equations) is negligible relative to that for the conformation dependent functions $G_i(\mathbf{q})$ (solution of partial differential equations).

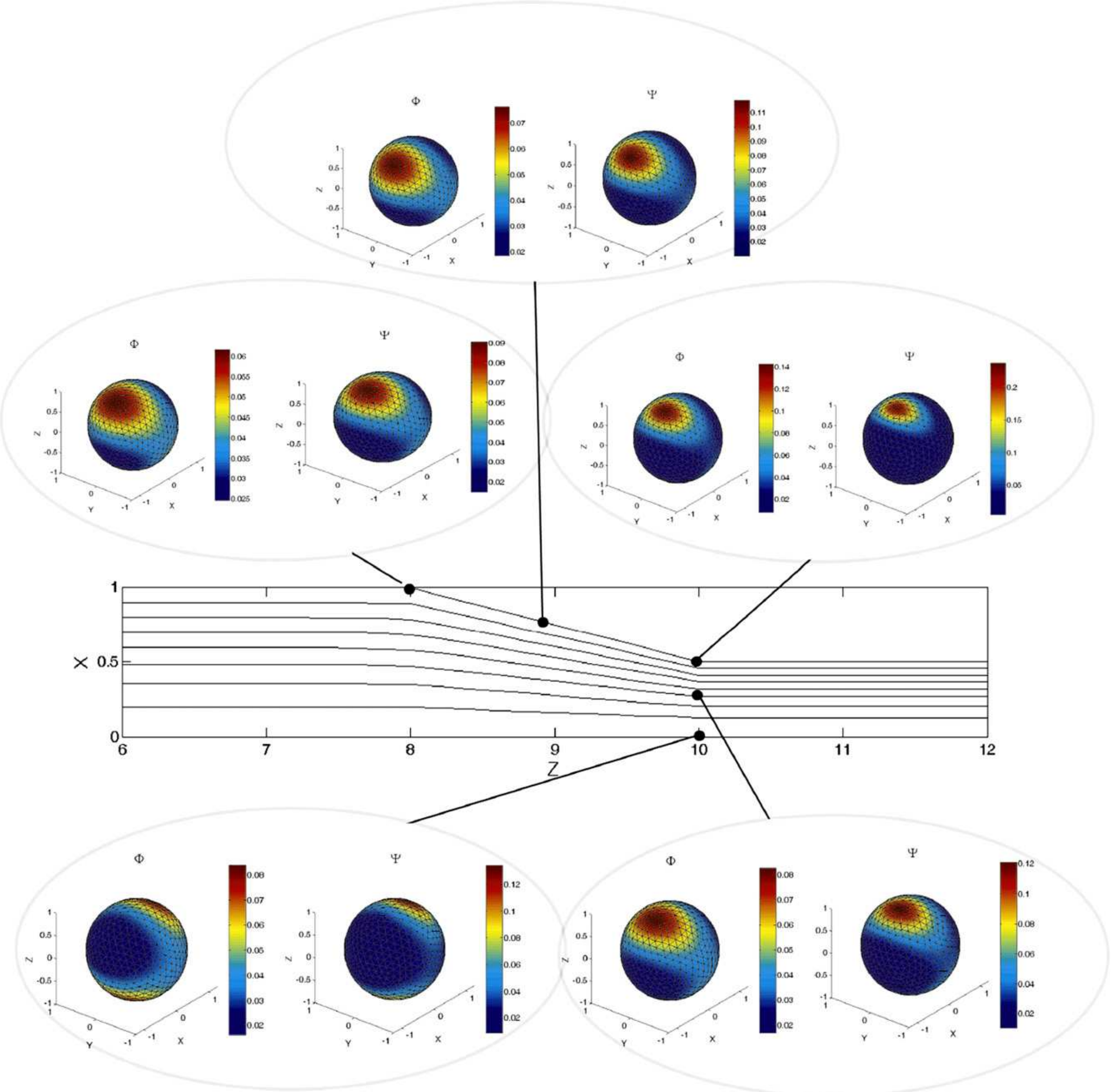

Fig. 7. Orientation distribution of active and pendant populations in a contraction flow. The orientation distribution is represented on the unit sphere at various positions along individual streamlines.

8. Three-dimensional simulation of resin transfer moulding

8.1. Governing equations and PGD approach

We now illustrate in some detail the application of the PGD to the modeling of resin transfer moulding processes. We consider the flow within a porous medium in a plate domain $\Xi = \Omega \times \mathcal{I}$ with $\Omega \subset \mathcal{R}^2$ and $\mathcal{I} = [0, H] \subset \mathcal{R}$. The governing equation is obtained by combining Darcy's law, which relates the fluid velocity to the pressure gradient,

$$\mathbf{v} = -\mathbf{K} \cdot \nabla p, \tag{53}$$

and the incompressibility constraint,

$$\nabla \cdot \mathbf{v} = 0. \tag{54}$$

Introduction of Eq. (53) into Eq. (54) yields a single equation for the pressure field:

$$\nabla \cdot (\mathbf{K} \cdot \nabla p) = 0. \tag{55}$$

The mould contains a laminate preform composed of P different anisotropic plies of thickness h, each one characterized by a permeability tensor $\mathbf{K}_i(x, y)$ that is assumed constant through the ply thickness. We define a characteristic function

$$\xi_i(z) = \begin{cases} 1 & z_i \leq z \leq z_{i+1}, \\ 0 & \text{otherwise}, \end{cases} \tag{56}$$

where $z_i = (i - 1) \cdot h$ is the location of ply i in the plate thickness. The laminate's permeability is thus given in separated form as follows:

$$\mathbf{K}(x, y, z) = \sum_{i=1}^{P} \mathbf{K}_i(\mathbf{x}) \cdot \xi_i(z), \tag{57}$$

where $\mathbf{x}$ denotes the in-plane coordinates, i.e. $\mathbf{x} = (x, y) \in \Omega$.

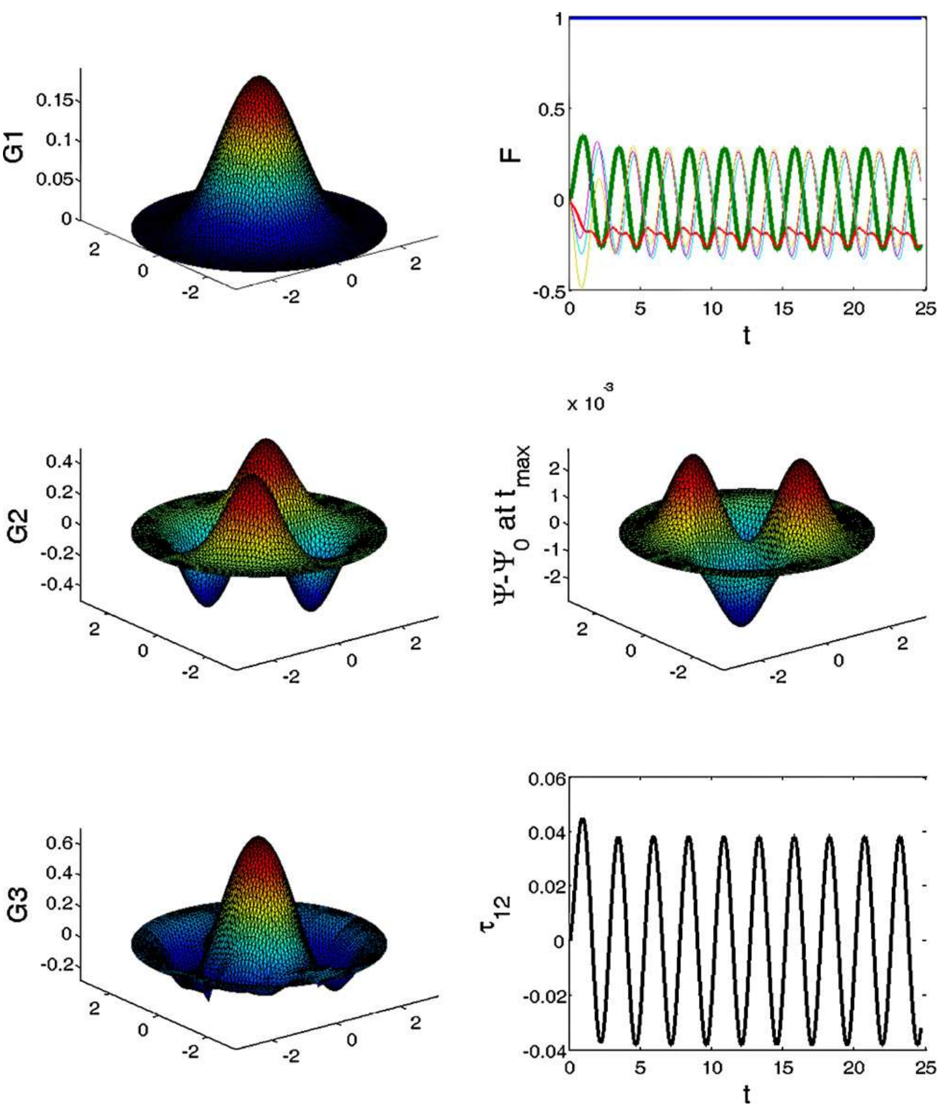

Fig. 8. Small amplitude oscillatory flow with $\omega = 2.55$ and $g = 0.1$. The three most significant conformation modes are depicted on the left. On the right are represented the first few time modes (top), the resulting orientation distribution (middle) and the shear stress (bottom).

The weak form of Eq. (55) reads:

$$\int_{\Xi} \nabla p^* \cdot (\mathbf{K} \cdot \nabla p)\, \mathrm{d}\Xi = 0, \tag{58}$$

for all test functions p^* selected in an appropriate functional space. Dirichlet boundary conditions are imposed for the pressure at the inlet and outlet of the flow domain, while zero flux (i.e. no flow) is imposed elsewhere as a weak boundary condition. We seek an approximate solution $p(x, y, z)$ in the PGD form:

$$p(\mathbf{x}, z) \approx \sum_{j=1}^{N} X_j(\mathbf{x}) \cdot Z_j(z). \tag{59}$$

The PGD algorithm then proceeds as follows. Assume that the first n functional products have been computed, i.e.

$$p^n(\mathbf{x}, z) = \sum_{j=1}^{n} X_j(\mathbf{x}) \cdot Z_j(z), \tag{60}$$

is a known quantity. We must now perform an enrichment step to obtain

$$p^{n+1}(\mathbf{x}, z) = p^n(\mathbf{x}, z) + R(\mathbf{x}) \cdot S(z). \tag{61}$$

The test function involved in the weak form is given by:

$$p^*(\mathbf{x}, z) = R^*(\mathbf{x}) \cdot S(z) + R(\mathbf{x}) \cdot S^*(z). \tag{62}$$

Introducing Eqs. (61) and (62) into Eq. (58), we obtain

$$\begin{aligned} &\int_{\Xi} \left(\begin{pmatrix} \tilde{\nabla} R^* \cdot S \\ R^* \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} + \begin{pmatrix} \tilde{\nabla} R \cdot S^* \\ R \cdot \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \right) \cdot \left(\mathbf{K} \cdot \begin{pmatrix} \tilde{\nabla} R \cdot S \\ R \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \right) \mathrm{d}\Xi \\ &\quad = -\int_{\Xi} \left(\begin{pmatrix} \tilde{\nabla} R^* \cdot S \\ R^* \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} + \begin{pmatrix} \tilde{\nabla} R \cdot S^* \\ R \cdot \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \right) \cdot \mathbf{Q}^n \,\mathrm{d}\Xi, \end{aligned} \tag{63}$$

where $\tilde{\nabla}$ denotes the plane component of the gradient operator, i.e. $\tilde{\nabla}^T = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$ and $\mathbf{Q}^n$ is a flux term known at step n:

$$\mathbf{Q}^n = \mathbf{K} \cdot \sum_{j=1}^{n} \begin{pmatrix} \tilde{\nabla} X_j(\mathbf{x}) \cdot Z_j(z) \\ X_j(\mathbf{x}) \cdot \dfrac{\mathrm{d}Z_j(z)}{\mathrm{d}z} \end{pmatrix}. \tag{64}$$

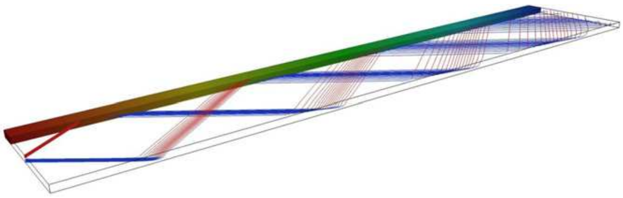

Fig. 9. Computed pressure field and flow pathlines for a two-ply rectangular laminate 1 m × 0.2 m × 0.01 m, $\Delta P = 0.05$ bar with 45°/− 45° orientation.

As discussed previously, each enrichment step of the PGD algorithm is a non-linear problem which must be performed by means of a suitable iterative process. Here, we compute the unknown functions $R(\mathbf{x})$ and $S(z)$ by applying an alternating direction fixed point algorithm. Thus, assuming $R(\mathbf{x})$ known, we compute $S(z)$, and then we update $R(\mathbf{x})$. The process continues until reaching convergence. The converged solutions provide the next functional product of the PGD: $R(\mathbf{x}) \rightarrow X_{n+1}(\mathbf{x})$ and $S(z) \rightarrow Z_{n+1}(z)$. The explicit form of these operations is given below.

8.2. Enrichment step

We now detail the computations yielding the functions $R(\mathbf{x})$ and $S(z)$.

(1) Computing $R(\mathbf{x})$ from $S(z)$:

When $S(z)$ is known, the test function reduces to:

$$p^*(\mathbf{x}, z) = R^*(\mathbf{x}) \cdot S(z), \tag{65}$$

and the weak form (63) becomes:

$$\int_{\Xi} \begin{pmatrix} \tilde{\nabla} R^* \cdot S \\ R^* \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \cdot \left(\mathbf{K} \cdot \begin{pmatrix} \tilde{\nabla} R \cdot S \\ R \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \right) \mathrm{d}\Xi$$
$$= - \int_{\Xi} \begin{pmatrix} \tilde{\nabla} R^* \cdot S \\ R^* \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \cdot \mathbf{Q}^n \, \mathrm{d}\Xi. \tag{66}$$

Now, as all functions involving the z coordinate are known, they can be integrated over $\mathcal{I} = [0, H]$. Thus, with the following notations:

$$\mathbf{K} = \begin{pmatrix} \mathbb{K} & \mathbf{k} \\ \mathbf{k}^T & \kappa \end{pmatrix}, \tag{67}$$

with

$$\mathbf{k} = \begin{pmatrix} \mathbf{K}_{xz} \\ \mathbf{K}_{yz} \end{pmatrix}, \tag{68}$$

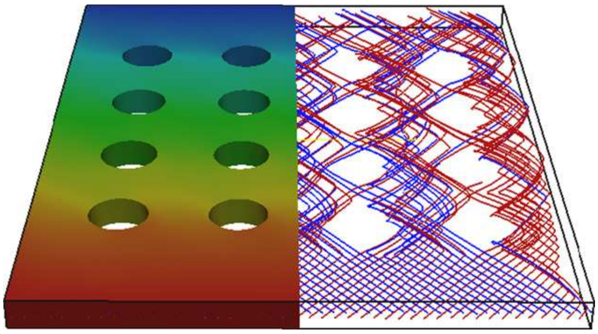

Fig. 10. Pressure field and flow trajectories in a complex laminate preform: 0.22 m × 0.18 m × 0.01 m, $\Delta P = 0.05$ bar.

and $\kappa = \mathbf{K}_{zz}$, we define:

$$\mathbf{K}^x = \begin{pmatrix} \displaystyle\int_{\mathcal{I}} \mathbb{K} \cdot S^2 \, \mathrm{d}z & \displaystyle\int_{\mathcal{I}} \mathbf{k} \cdot \frac{\mathrm{d}S}{\mathrm{d}z} \cdot S \, \mathrm{d}z \\ \displaystyle\int_{\mathcal{I}} \mathbf{k}^T \cdot \frac{\mathrm{d}S}{\mathrm{d}z} \cdot S \, \mathrm{d}z & \displaystyle\int_{\mathcal{I}} \kappa \cdot \left(\frac{\mathrm{d}S}{\mathrm{d}z} \right)^2 \mathrm{d}z \end{pmatrix}, \tag{69}$$

and

$$(\mathbf{Q}^x)^n$$
$$= \sum_{j=1}^{n} \left(\begin{pmatrix} \displaystyle\int_{\mathcal{I}} \mathbb{K} \cdot S \cdot Z_j \, \mathrm{d}z & \displaystyle\int_{\mathcal{I}} \mathbf{k} \cdot \frac{\mathrm{d}Z_j}{\mathrm{d}z} \cdot S \, \mathrm{d}z \\ \displaystyle\int_{\mathcal{I}} \mathbf{k}^T \cdot \frac{\mathrm{d}S}{\mathrm{d}z} \cdot Z_j \, \mathrm{d}z & \displaystyle\int_{\mathcal{I}} \kappa \cdot \frac{\mathrm{d}S}{\mathrm{d}z} \cdot \frac{Z_j}{\mathrm{d}z} \, \mathrm{d}z \end{pmatrix} \cdot \begin{pmatrix} \tilde{\nabla} X_j(\mathbf{x}) \\ X_j(\mathbf{x}) \end{pmatrix} \right). \tag{70}$$

Finally, we can write Eq. (66) as follows:

$$\int_{\Omega} \begin{pmatrix} \tilde{\nabla} R^* \\ R^* \end{pmatrix} \cdot \left(\mathbf{K}^x \cdot \begin{pmatrix} \tilde{\nabla} R \\ R \end{pmatrix} \right) \mathrm{d}\Omega = - \int_{\Omega} \begin{pmatrix} \tilde{\nabla} R^* \\ R^* \end{pmatrix} \cdot (\mathbf{Q}^x)^n \, \mathrm{d}\Omega. \tag{71}$$

This last equation defines an elliptic 2D problem (in weak form) for the unknown function R defined over Ω, i.e. the mid-plane of the preform.

(2) Computing $S(z)$ from $R(\mathbf{x})$:

When $R(\mathbf{x})$ is known the test function reads:

$$p^*(\mathbf{x}, z) = R(\mathbf{x}) \cdot S^*(z), \tag{72}$$

and the weak form (63) reduces to:

$$\int_{\Xi} \begin{pmatrix} \tilde{\nabla} R \cdot S^* \\ R \cdot \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \cdot \left(\mathbf{K} \cdot \begin{pmatrix} \tilde{\nabla} R \cdot S \\ R \cdot \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \right) \mathrm{d}\Xi$$
$$= - \int_{\Xi} \begin{pmatrix} \tilde{\nabla} R \cdot S^* \\ R \cdot \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \cdot \mathbf{Q}^n \, \mathrm{d}\Xi. \tag{73}$$

Since all functions involving the in-plane coordinates $\mathbf{x} = (x, y)$ are known, they can be integrated over Ω. Thus, using the previous notation, we define:

$$\mathbf{K}^z = \begin{pmatrix} \displaystyle\int_{\Omega} (\tilde{\nabla} R) \cdot (\mathbb{K} \cdot \tilde{\nabla} R) \, \mathrm{d}\Omega & \displaystyle\int_{\Omega} (\tilde{\nabla} R) \cdot \mathbf{k} \cdot R \, \mathrm{d}\Omega \\ \displaystyle\int_{\Omega} (\tilde{\nabla} R) \cdot \mathbf{k} \cdot R \, \mathrm{d}\Omega & \displaystyle\int_{\Omega} \kappa \cdot R^2 \, \mathrm{d}\Omega \end{pmatrix}, \tag{74}$$

and

$$(\mathbf{Q}^z)^n$$
$$= \sum_{j=1}^{n} \left(\begin{pmatrix} \displaystyle\int_{\Omega} (\tilde{\nabla} R) \cdot (\mathbb{K} \cdot \tilde{\nabla} X_j) \, \mathrm{d}\Omega & \displaystyle\int_{\Omega} (\tilde{\nabla} R) \cdot \mathbf{k} \cdot X_j \, \mathrm{d}\Omega \\ \displaystyle\int_{\Omega} (\tilde{\nabla} X_j) \cdot \mathbf{k} \cdot R \, \mathrm{d}\Omega & \displaystyle\int_{\Omega} \kappa \cdot X_j \cdot R \, \mathrm{d}\Omega \end{pmatrix} \cdot \begin{pmatrix} Z_j(z) \\ \dfrac{\mathrm{d}Z_j}{\mathrm{d}z}(z) \end{pmatrix} \right) (.75)$$

We can thus write Eq. (73) in the form

$$\int_{\mathcal{I}} \begin{pmatrix} S^* \\ \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \cdot \left(\mathbf{K}^z \cdot \begin{pmatrix} S \\ \dfrac{\mathrm{d}S}{\mathrm{d}z} \end{pmatrix} \right) \mathrm{d}z = - \int_{\mathcal{I}} \begin{pmatrix} S^* \\ \dfrac{\mathrm{d}S^*}{\mathrm{d}z} \end{pmatrix} \cdot (\mathbf{Q}^z)^n \, \mathrm{d}z. \tag{76}$$

This equation defines a one-dimensional problem (in weak form) for the unknown function S.

8.3. Numerical example

Traditionally, the flow of a viscous fluid through a porous preform within a planar mould is assumed two-dimensional in order to make possible realistic simulations of industrial interest.

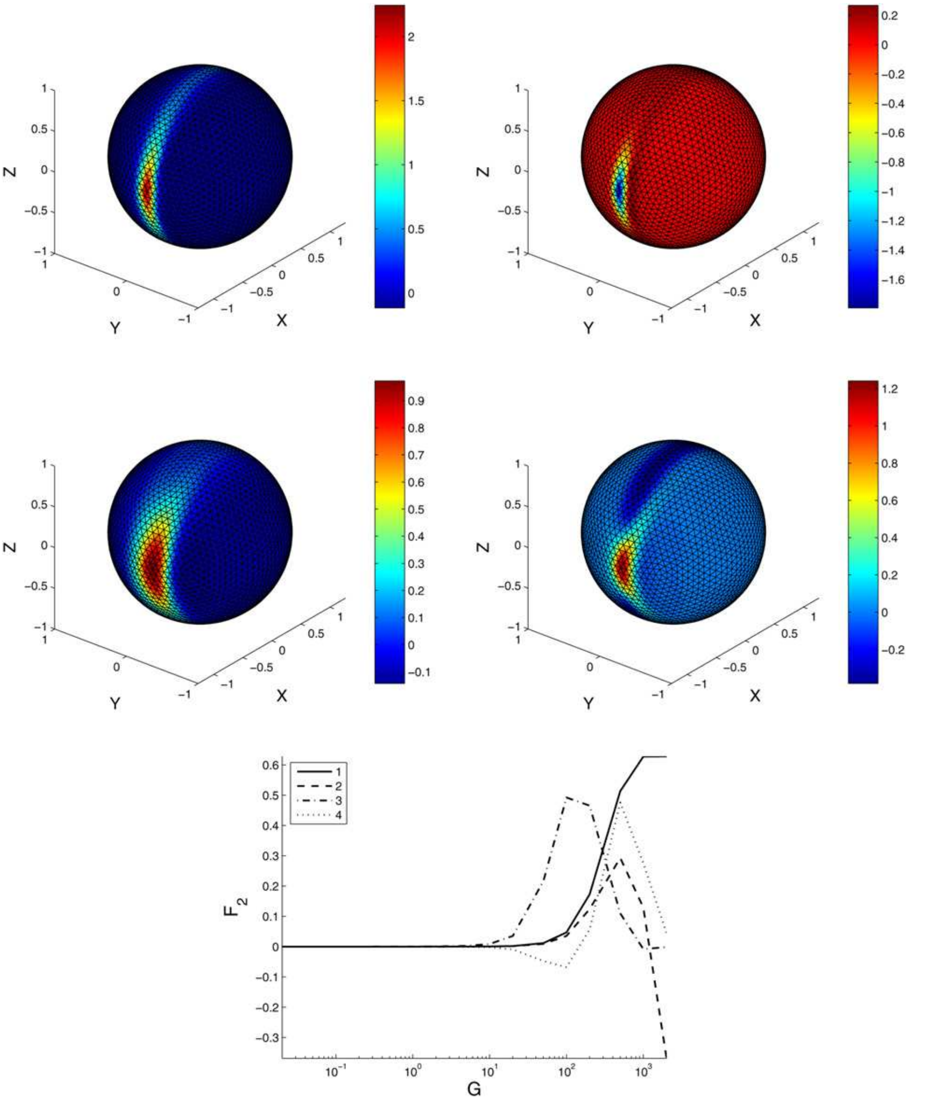

Fig. 11. Solution of the parametric Fokker-Planck equation for short fibers: functions involved in the four first terms of the PGD, namely $F_1^1(\mathbf{p})$ to $F_1^4(\mathbf{p})$ (top) and $F_2^1(G)$ to $F_2^4(G)$ (bottom).

When the mould consists of a laminate composed of several anisotropic plies with different principal directions of anisotropy, the definition of an equivalent permeability tensor representing the whole laminated is an important issue. One would expect that an appropriate equivalent permeability tensor could be defined by averaging through the thickness the permeability of the different plies that compose the laminate. In order to assess the validity of this approach, we consider a rectangular laminate composed of two unidirectional plies. The plies are identical but have different orientation.

In the principal anisotropy directions, the in-plane permeability tensor of each ply is given by

$$\mathbf{K} = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix}. \tag{77}$$

We adopt a coordinate system such that the x-coordinate axis is aligned in the direction of the longest plate edge, the y-coordinate defines the plate width and the z-coordinate its thickness. The first ply is turned an angle θ with respect to the x-axis, whereas the second ply is oriented at an angle $-\theta$. Thus, their in-plane permeabilities are given by

$$\mathbf{K}^1 = \mathbf{Q}_{(\theta)}^T \mathbf{K} \mathbf{Q}_{(\theta)}, \tag{78}$$

and

$$\mathbf{K}^2 = \mathbf{Q}_{(-\theta)}^T \mathbf{K} \mathbf{Q}_{(-\theta)}, \tag{79}$$

where $\mathbf{Q}_{(\theta)}$ and $\mathbf{Q}_{(-\theta)}$ are rotation tensors.

Defining the equivalent in-plane permeability of the laminate $\tilde{\mathbf{K}}$ from a simple through-the-thickness average of ply permeabilities, one obtains

$$\tilde{\mathbf{K}} = \begin{pmatrix} K_1 \cdot \cos^2(\theta) + K_2 \cdot \sin^2(\theta) & 0 \\ 0 & K_2 \cdot \cos^2(\theta) + K_1 \cdot \sin^2(\theta) \end{pmatrix}. \tag{80}$$

Now, if we apply a pressure drop ΔP between the inlet ($x = 0$) and outlet ($x = L$), the pressure distribution is strictly linear and the associated velocity field is predicted to be uniform and unidirectional:

$$\mathbf{v}(x, y, z) = \begin{pmatrix} -\tilde{\mathbf{K}} \cdot \dfrac{\Delta P}{L} \\ 0 \\ 0 \end{pmatrix}. \tag{81}$$

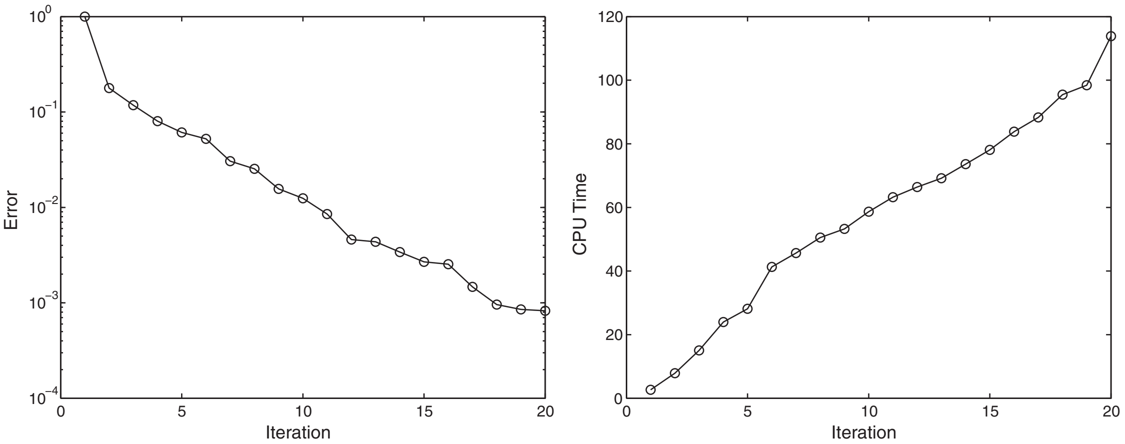

Fig. 12. Convergence and CPU time as a function of the number N of terms in the PGD approximation. The error is defined as the residual L^2 norm.

Thus, one would expect fluid trajectories parallel to the x-axis and a residence time t_R such that $L = ||\mathbf{v}|| \cdot t_R$.

The results of the three-dimensional PGD simulation reveal a very different situation, which shows that use of an average permeability is not appropriate. Fig. 9 depicts the pressure field as well as some flow pathlines obtained in the 3D simulation. The flow is clearly far from unidirectional and uniform.

The PGD can be used to analyze more complex cases. Fig. 10 depicts the pressure fields and flow pathlines in a plate with cylindrical obstacles composed of 51 plies. The predicted flow trajectories are quite complex indeed. Consideration of such a large number of plies has no significant impact on the overall PGD computing time, in view of the relatively low computing cost of the one-dimensional problems defined through the plate thickness. In order to compute an equivalent solution by applying the finite element method, one should use about two million nodes. The PGD solution is computed in a few minutes on a laptop.

9. Parametric modeling in computational rheometry

In this final illustration, we consider the evolution of the distribution Ψ of orientation $\{\mathbf{p}\}$ for a suspension of short fibres. The corresponding steady-state Fokker-Planck equation reads in dimensionless form

$$\frac{\partial}{\partial \mathbf{p}} \left(\dot{\mathbf{p}} \Psi(\mathbf{p}, t) \right) - \frac{\partial}{\partial \mathbf{p}} \left(\frac{\partial \Psi(\mathbf{p}, t)}{\partial \mathbf{p}} \right) = 0, \tag{82}$$

with a suitable governing equation relating $\dot{\mathbf{p}}$ to the applied flow kinematics [11].

We study simple shear flow and wish to compute the solution at once for any value of the applied dimensionless shear rate G. With the PGD, it suffices to view G as an additional coordinate. We seek an approximation in the separated form:

$$\Psi(\mathbf{p}, G) \approx \sum_{i=1}^{N} F_1^i(\mathbf{p}\ ) \cdot F_2^i(G). \tag{83}$$

Thus, instead of solving a problem defined on the unit sphere for each G within a discrete set of values, we solve at once a single problem defined in a higher-dimensional space that now includes the unit sphere for $\{\mathbf{p}\}$ as well as the continuous interval of values for the applied shear rate G. With the PGD, the resulting increase in model dimensionality has a negligible impact on the overall computational effort [9].

Fig. 11 depicts the four most relevant functions of orientation, $F_1^1(\mathbf{p})$ to $F_1^4(\mathbf{p})$, and strain rate, $F_2^1(G)$ to $F_2^4(G)$.

Finally, Fig. 12 shows the evolution of the numerical error and total computing time on a laptop as a function of the number N of PGD terms. The error is defined as the L^2-norm of the residual of the Fokker-Planck equation in strong form.

10. Conclusions

In this review paper, we have revisited the proper generalized decomposition, its foundations, and some exciting applications.

By means of selected illustrations, we have demonstrated that the PGD separated representations are particularly suitable for addressing models defined in high-dimensional spaces. The PGD yields efficient, non-incremental time integration strategies of transient problems. It allows for the fast computation of fully 3 D solutions of problems defined in degenerate plate or shell-like domains typically encountered in materials and process engineering applications. Finally, separated representations make possible the increase of the problem dimensionality by introducing all sources of variability as extra-coordinates.

By avoiding the exponential complexity of standard grid-based discretization techniques, the PGD circumvents the curse of dimensionality in a variety of problems. With the PGD, the problem's usual coordinates (e.g. space, time, conformation), but also model parameters, boundary conditions, and other sources of variability can be viewed globally as coordinates of a high-dimensional space wherein an approximate solution can efficiently be computed at once.

The PGD is a recent technique and its mathematical foundations remain for the most part to be established rigorously. Theoretical results on the numerical analysis of the PGD are becoming available [8,26]. Empirical observations regarding the limitations of the PGD can be summarised as follows. We have not yet encountered a problem for which the PGD would behave less well than the corresponding finite element solution framework, but on the other hand, the PGD has been found extremely efficient in a wide variety of problems. When the solution is non-separable, irrespective of the nature (e.g. symmetric or not) of the underlying differential operator, the number of terms in the PGD expansion does grow in order to span the full tensor basis of approximation functions, and the PGD then offers no particular advantage over classical techniques.

In our opinion, PGD-based discretization techniques constitute a new paradigm in scientific computing. The way is open for innovative algorithms in simulation, parametric modeling, inverse identification, optimization and control of high-dimensional systems. Application of the PGD in computational rheology and non-Newtonian fluid mechanics has been shown in this review paper to have great potential indeed.

References

[2] A. Ammar, D. Ryckelynck, F. Chinesta, R. Keunings, On the reduction of kinetic theory models related to finitely extensible dumbbells, Journal of Non-Newtonian Fluid Mechanics 134 (2006) 136–147.

[3] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, Journal of Non-Newtonian Fluid Mechanics 139 (2006) 153–176.

[4] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, 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-Newtonian Fluid Mechanics 144 (2007) 98–121.

[5] A. Ammar, E. Pruliere, F. Chinesta, M. Laso, Reduced numerical modeling of flows involving liquid-crystalline polymeres, Journal of Non-Newtonian Fluid Mechanics 160 (2009) 140–156.

[6] A. Ammar, M. Normandin, F. Daim, D. Gonzalez, E. Cueto, F. Chinesta, Non-incremental strategies based on separated representations: applications in computational rheology, Communications in Mathematical Sciences 8/3 (2010) 671–695.

[8] A. Ammar, F. Chinesta, A. Falco, On the convergence of a greedy rank-one update algorithm for a class of linear systems, Archives of Computational Methods in Engineering 17 (4) (2010) 473–486.

[9] A. Ammar, M. Normandin, F. Chinesta, Solving parametric complex fluids models in rheometric flows, Journal of Non-Newtonian Fluid Mechanics 165 (2010) 1588–1601.

[10] R.A. Bialecki, A.J. Kassab, A. Fic, Proper orthogonal decomposition and modal analysis for acceleration of transient FEM thermal analysis, International Journal of Numerical Methods in Engineering 62 (2005) 774–797.

[11] R.B. Bird, C.F. Crutiss, R.C. Armstrong, O. Hassager, Dynamic of Polymeric Liquid, vol. 2: Kinetic Theory, John Wiley and Sons, 1987.

[13] J. Burkardt, M. Gunzburger, H.-Ch. Lee, POD and CVT-based reduced-order modeling of Navier-Stokes flows, Computer Methods in Applied Mechanics and Engineering 196 (2006) 337–355.

[15] F. Chinesta, A. Ammar, A. Falco, M. Laso, On the reduction of stochastic kinetic theory models of complex fluids, Modeling and Simulation in Materials Science and Engineering 15 (2007) 639–652.

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

[17] F. Chinesta, A. Ammar, E. Cueto, Recent advances in the use of the proper generalized decomposition for solving multidimensional models, Archives of Computational Methods in Engineering 17 (4) (2010) 327–350.

[18] F. Chinesta, A. Ammar, E. Cueto, Proper generalized decomposition of multiscale models, International Journal of Numerical Methods in Engineering 83 (8–9) (2010) 1114–1132.

[19] F. Chinesta, A. Ammar, E. Cueto, On the use of Proper Generalized Decompositions for solving the multidimensional chemical master equation, European Journal of Computational Mechanics 19 (2010) 53–64.

[20] D. Gonzalez, A. Ammar, F. Chinesta, E. Cueto, Recent advances in the use of separated representations, International Journal for Numerical Methods in Engineering 81 (5) (2010) 637–659.

[21] M.D. Gunzburger, J.S. Peterson, J.N. Shadid, Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data, Computer Methods in Applied Mechanics and Engineering 196 (2007) 1030–1047.

[22] R. Keunings, Micro-macro methods for the multiscale simulation viscoelastic flow using molecular models of kinetic theory, in: D.M. Binding, K. Walters (Eds.), Rheology Reviews, British Society of Rheology, 2004, pp. 67–98.

[23] P. Ladeveze, Nonlinear Computational Structural Mechanics, Springer, NY, 1999.

[24] P. Ladeveze, J.-C. Passieux, D. Neron, The LATIN multiscale computational method and the proper orthogonal decomposition, Computer Methods in Applied Mechanics and Engineering 199 (21–22) (2010) 1287–1296.

[26] C. Le Bris, T. Lelièvre, Y. Maday, Results and Questions on a Nonlinear Approximation Approach for Solving High-dimensional Partial Differential Equations, Constructive Approximation, 10.1007/s00365-009-9071-1.

[27] G.M. Leonenko, T.N. Phillips, On the solution of the Fokker-Planck equation using a high-order reduced basis approximation, Computer Methods in Applied Mechanics and Engineering 199 (1–4) (2009) 158–168.

[28] Y. Maday, E.M. Ronquist, The reduced basis element method: application to a thermal fin problem, SIAM Journal of Science and Computation 26/1 (2004) 240–258.

[29] B. Mokdad, E. Pruliere, A. Ammar, F. Chinesta, On the simulation of kinetic theory models of complex fluids using the Fokker-Planck approach, Applied Rheology 17/2 (26494) (2007) 1–14.

[30] B. Mokdad, A. Ammar, M. Normandin, F. Chinesta, J.R. Clermont, A fully deterministic micro-macro simulation of complex flows involving reversible network fluid models, Mathematics and Computer in Simulation 80 (2010) 1936–1961.

[31] A. Nouy, P. Ladeveze, Multiscale computational strategy with time and space homogenization: a radial-type approximation technique for solving microproblems, International Journal of Multiscale Computational Engineering 170/2 (2004).

[34] H.M. Park, D.H. Cho, The use of the Karhunen-Loéve decomposition for the modelling of distributed parameter systems, Chemical Engineering Science 51 (1996) 81–98.

[35] E. Pruliere, A. Ammar, N. El Kissi, F. Chinesta, Recirculating flows involving short fiber suspensions: numerical difficulties and efficient advanced micro-macro solvers, Archives of Computational Methods in Engineering, State of the Art Reviews 16 (2009) 1–30.

[36] E. Pruliere, F. Chinesta, A. Ammar, On the deterministic solution of parametric models by using the proper generalized decomposition, Mathematics and Computer Simulation 81 (2010) 791–810.

[37] D. Ryckelynck, A priori hyper-reduction method: an adaptive approach, Journal of Computational Physics 202 (2005) 346–366.

[38] D. Ryckelynck, L. Hermanns, F. Chinesta, E. Alarcon, An efficient "a priori" model reduction for boundary element models, Engineering Analysis with Boundary Elements 29 (2005) 796–801.

[39] D. Ryckelynck, F. Chinesta, E. Cueto, A. Ammar, On the a priori model reduction: overview and recent developments, Archives of Computational Methods in Engineering 13/1 (2006) 91–128.