COMMUN. MATH. SCI. © 2010 International Press
Vol. 8, No. 3, pp. 671–695

# NON INCREMENTAL STRATEGIES BASED ON SEPARATED REPRESENTATIONS: APPLICATIONS IN COMPUTATIONAL RHEOLOGY*

A. AMMAR[†], M. NORMANDIN[‡], F. DAIM[§], D. GONZÁLEZ[¶], E. CUETO[‖], AND F. CHINESTA[**]

**Key words.** Complex fluids, computational rheology, kinetic theory, separated representations, finite sums decomposition, reduced approximation basis.

**AMS subject classifications.** 76A05, 65M60

**Abstract.** Description of complex materials, in particular complex fluids, involves numerous computational challenges. Today, atomistic descriptions are too expensive from the computational point of view, motivating that this kind of analysis is restricted to extremely small systems. The next scale introduces some simplificative hypotheses, leading to coarse grained descriptions. At this level, Brownian dynamics simulations are usually employed. However, this level of description requires intensive computation resources with significant unfavorable impact on the simulation performances (CPU time). For these reasons sometimes kinetic theory descriptions are preferred. In those descriptions, the molecular conformation is described from a probability distribution function (PDF) whose evolution is governed by the Fokker-Planck equation. This approach, despite its mathematical simplicity introduces some computational challenges. First, the distribution function is defined in a multidimensional space, and then the associated partial differential equations must be solved in a multidimensional domain (some times involving thousands dimensions). Secondly, the analysis of transient models needs intensive computation, in particular when the system response under small amplitude oscillations (of high and very high frequency) is concerned. In some of our former works [A. Ammar, B. Mokdad, F. Chinesta and R. Keunings, J. Non-Newtonian Fluid Mech., 139, 153-176, 2006], [A. Ammar, B. Mokdad, F. Chinesta and R. Keunings, J. Non-Newtonian Fluid Mech., 144, 98-121, 2007] we proposed a technique based on the separated representation of the unknown field able to circumvent the curse of dimensionality. In this paper, we are addressing the second challenging point, the one related to the transient behavior. For this purpose we propose a separated representation of transient models leading to a non-incremental strategy, allowing impressive CPU time savings.

## 1. Introduction

Some models encountered in science and engineering are sometimes defined in multidimensional spaces (as the ones involved in quantum mechanics or kinetic theory descriptions of materials, including complex fluids) that exhibit the terrific curse of dimensionality when usual mesh-based discretization techniques are applied. Other times, models involve transient fields that, even when they are defined in three-

[*] Received: August 25, 2008; accepted (in revised from): May 19, 2009.

[†] Laboratoire de Rhéologie, UMR CNRS - INPG - UJF, 1301 rue de la piscine, BP 53 Domaine Universitaire, F-38041 Grenoble Cedex 9, France (Amine.Ammar@ujf-grenoble.fr).

[‡] Laboratoire de Rhéologie, UMR CNRS - INPG - UJF, 1301 rue de la piscine, BP 53 Domaine Universitaire, F-38041 Grenoble Cedex 9, France (normandin@ujf-grenoble.fr).

[§] LMSP – Laboratoire de Mécanique des Systèmes et des Procédés, UMR CNRS - ENSAM, 151 Boulevard de l'Hôpital, F-75013 Paris, France (fatima.daim@mines.paristech.fr).

[¶] Group of Structural Mechanics and Material Modelling, Aragón Institute of Engineering Research, I3A. Universidad de Zaragoza, Edificio Betancourt, Mara de Luna, 7. E-50012 Zaragoza, Spain (gonzal@unizar.es).

[‖] Group of Structural Mechanics and Material Modelling, Aragón Institute of Engineering Research, I3A. Universidad de Zaragoza, Edificio Betancourt. Mara de Luna, 7. E-50012 Zaragoza, Spain (ecueto@unizar.es).

[**] EADS Corporate International Chair, GeM – Institut de Recherche en Génie Civil et Mécanique, UMR CNRS - ECN - Université de Nantes, 1 rue de la Noe, BP 92101. F-44321 Nantes Cedex 3, France (Francisco.Chinesta@ec-nantes.fr).

dimensional physical spaces, must be solved in large time intervals using very small time steps.

In the first kind of models the difficulty is quite natural and the solution of such models requires new strategies. One possibility lies in the use of sparse grids [7]. However, as argued in [1], the use of sparse grids 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 [17, 5] for some mathematical results on this topic). Basically, the separated representation of a generic function $u(x_1,\dots,x_D)$ (also known as finite sums decomposition) is written as

$$u(x_1,\dots,x_D) \approx \sum_{i=1}^{i=N} F_1^i(x_1) \times \cdots \times F_D^i(x_D). \tag{1.1}$$

This kind of representation is not new, it was widely employed in the last decades in the framework of quantum chemistry. In particular the Hartree-Fock (that involves a single product of functions) and post-Hartree-Fock approaches (as the MCSCF that involves a finite number of sums) made use of a separated representation of the wavefunction.

We recently proposed a technique able to construct, in a way completely transparent for the user, the separated representation of the unknown field involved in a partial differential equation. This technique, originally described and applied to multi-bead-spring FENE models of polymeric systems in [3], was extended to transient models of such complex fluids in [4]. Other more complex models (involving different couplings and non-linearities) based on the reptation theory of polymeric liquids were analyzed in [15].

We must recall that the technique that we proposed in the papers just referred is not a universal strategy able to solve any kind of multidimensional partial differential equation (PDE). Thus, the efficient application of the technique that we proposed in [3] requires the separability of all the fields involved in the model. Obviously, this separability is not always possible because some functions need a tremendous number of sums. On the other hand, even when the field is separable (one could perform this separation by invoking for example the SVD or the multidimensional SVD [13, 11]) the finite sums decomposition of general multidimensional functions is not realistic because the amount of memory needed for storing the discrete form of such functions before applying the multidimensional SVD.

In general any function can be separated by using the SVD. In 2D or 3D the computing cost is reasonable. Certain functions accept a separated representation consisting of a reduced number of functional products, but it is not always the case. For example, a step moving along the x-axis cannot be written as a sum of a reduced number of products. There are many other examples.

Another major difficulty consists in the enforcement of boundary conditions. The simplest way to enforce non homogeneous boundary conditions (e.g. $u = u_g$ in $\partial\Omega$) consists in defining a function $\hat{u}(\mathbf{x})$, $\mathbf{x} \in \Omega \subset \mathbb{R}^D$, with the requested regularity, verifying the boundary conditions, i.e., $\hat{u}(\mathbf{x} \in \partial\Omega) = u_g(\mathbf{x} \in \partial\Omega)$. Now, one could perform the separation of $\hat{u}$, by invoking the SVD or the multidimensional SVD, leading to

$$\hat{u}(x_1,\dots,x_D) \approx \sum_{i=1}^{i=M} G_1^i(x_1) \times \cdots \times G_D^i(x_D) \tag{1.2}$$

and then write

$$u(x_1,\ldots,x_D) \approx \sum_{i=1}^{i=M} G_1^i(x_1)\times\cdots\times G_D^i(x_D) + \sum_{i=1}^{i=N} F_1^i(x_1)\times\cdots\times F_D^i(x_D), \qquad (1.3)$$

where the second part of the right member $\sum_{i=1}^{i=N} F_1^i(x_1)\times\cdots\times F_D^i(x_D)$ is subjected to homogenous boundary conditions.

In high dimensions the main difficulty in applying this procedure is precisely the separation of $\hat{u}$ that becomes unrealistic as just indicated.

Despite all these difficulties, numerous models encountered in science and engineering are quite simple because the boundary conditions are simple (sometimes homogeneous) and the fields involved in the models are separated or can be separated easily. An introductive overview on the application of separated representations in the multi-scale modelling of materials can be found in [10].

The present work will focus on the second kind of models described at the beginning of this section. For the sake of simplicity in what follows we consider models defined in moderate dimensions ($d$D, $d=1,2,3$, physical or conformation spaces) but whose solutions evolve in large time intervals. In this context, 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 the stability requirements, and the simulation time interval is large enough, the 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 of the order of 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 techniques. Other scenarios found in computational rheology will be described and analyzed later.

In this paper we are exploring some new possibilities based on separated space-time representations that are closely inspired from some existing and well established strategies [4, 12]. In the next section we motivate the use of separated representations and its connection with other more experienced techniques, as the model reduction techniques based on the use of proper orthogonal decompositions. In section 3 we illustrate the implementation of the proposed technique through some academic models (parabolic and hyperbolic equations) where the problems related to the non-linearities will be also addressed. In section 4 we develop a compact and unified representation that allows to consider and easily implement different kind of models. Finally, in section 5, we consider a transient FENE model of a complex fluid that is efficiently solved by using the separated conformation-time representation.

In [4] we addressed the issue related to transient models. However in those cases we only addressed some simple parabolic linear models and we never compared incremental versus separated representation of such transient models. The main contributions of the present work are: (i) to motivate the construction of a separated representation of a model solution when this solution accepts a separated form by applying a singular value decomposition; (ii) the analysis of more complex transient models, e.g., hyperbolic models, ...; (iii) to propose different strategies for solving non linear models; and (iv) to give some preliminary comparison between standard incremental techniques and ones making use of the separated representations with its inherent non-incremental character.

## 2. Motivating the use of separated space-time representations

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 $t^p$, 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

$$\mathbf{Q}=\begin{pmatrix} u_1^1 & u_1^2 & \ldots & u_1^P \\ u_2^1 & u_2^2 & \ldots & u_2^P \\ \vdots & \vdots & \ddots & \vdots \\ u_{N_n}^1 & u_{N_n}^2 & \ldots & u_{N_n}^P \end{pmatrix}. \tag{2.1}$$

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

$$\left(\mathbf{Q}\mathbf{Q}^T\right)\boldsymbol{\phi}=\lambda\boldsymbol{\phi} \tag{2.2}$$

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

When the field evolves smoothly, the magnitude of the eigenvalues decreases very fast, a 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 $\epsilon$ ($\epsilon=10^{-8}\times\lambda_1$ in practice, $\lambda_1$ being the highest eigenvalue) only a reduced number of modes are retained. Let $R$ ($R<<N_n$) 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 to be ordered). Thus, one could write

$$u(\mathbf{x},t)\approx\sum_{i=1}^{i=R}\phi_i(\mathbf{x})\cdot T_i(t)\equiv\sum_{i=1}^{i=R}X_i(\mathbf{x})\cdot T_i(t), \tag{2.3}$$

where for the sake of clarity the space modes $\phi_i(\mathbf{x})$ will be from now on denoted as $\phi_i(\mathbf{x})\equiv X_i(\mathbf{x})$. Equation (2.3) 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, ... [16, 14, 18]. One other possibility is computing 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 is to construct an adaptive reduced approximation basis, which should be the best reduced approximation basis for the treated problem. It 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 recently proposed an enrichment technique based on the use of some Krylov subspaces generated by the equation residual. This technique known as "a priori" model reduction was originally proposed in [19], widely described in [20], and successfully applied for solving complex fluid flows within the kinetic theory framework [2, 8] and for speeding up thermomechanical simulations [9]. However, some difficulties were noticed in the application of this strategy: (i) the enrichment based on the use of the Krylov subspaces is far from 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 can be determined simultaneously by applying an appropriate algorithm.

In what follows we are describing a possible strategy able to compute these separated functional couples. From our numerical experiments we noticed that this technique converged in the different models analyzed at present, and that the final decomposition is not so far to the one that results from the application of the proper orthogonal decomposition on the model solution, that is, the number of functional couples was quite similar, being the optimal decomposition the one performed by applying the POD. However, at present we cannot prove this empirical observation for general models.

## 3. Illustrating the discretization based on separated representations

In this section we are illustrating the discretization of partial differential equations using a separated representation of the unknown field.

Let us consider the generic partial differential equation

$$\mathcal{L}(u(\mathbf{x},t)) = f(\mathbf{x},t) \quad \text{ in } \Omega\times(0,T_{max}],$$

with the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x},0) = u_0 \ \mathbf{x}\in\Omega, \\ u(\mathbf{x},t) = u_D \ (\mathbf{x},t)\in\partial\Omega\times(0,T_{max}], \end{cases} \tag{3.1}$$

where $\Omega\subset\mathbb{R}^d, d\geq 1$, $T_{max}>0$. 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

$$u(\mathbf{x},t) \approx \sum_{i=1}^{N} T_i(t)\cdot X_i(\mathbf{x}). \tag{3.2}$$

We will first illustrate the strategy for constructing these functional products in the case of some academic transient problems whose exact solutions are sometimes known: (i) the advection-diffusion equation; (ii) the wave equation; and finally (iii) a non-linear parabolic problem.

### 3.1. Advection-diffusion problem.

We consider the advection-diffusion equation

$$\frac{\partial u}{\partial t} - a\Delta u + \mathbf{v}\cdot\nabla u = f(\mathbf{x},t) \quad \text{ in } \Omega\times(0,T_{max}], \tag{3.3}$$

with the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x},0) = u_0 \ \mathbf{x}\in\Omega, \\ u(\mathbf{x},t) = u_g \ (\mathbf{x},t)\in\partial\Omega\times(0,T_{max}], \end{cases} \tag{3.4}$$

where $a$ is the diffusion coefficient and $\mathbf{v}$ the velocity field. The weak formulation yields:

*Find $u(\mathbf{x},t)$ verifying the boundary conditions (3.4) such that*

$$\int_0^{T_{max}} \int_\Omega u^\star \left( \frac{\partial u}{\partial t} - a\Delta u + \mathbf{v}\cdot\nabla u - f(\mathbf{x},t) \right) d\mathbf{x}\, dt = 0 \tag{3.5}$$

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

We compute now the functions involved in the sum (3.2). 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

$$u(\mathbf{x},t) \approx \sum_{i=1}^{n} T_i(t)\cdot X_i(\mathbf{x}) + R(t)\cdot S(\mathbf{x}). \tag{3.6}$$

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

$$u^\star = S\cdot R^\star + R\cdot S^\star. \tag{3.7}$$

Introducing (3.6) and (3.7) into (3.5) results in

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega (S\cdot R^\star + R\cdot S^\star)\cdot \left( S\cdot\frac{\partial R}{\partial t} - a\Delta S\cdot R + (\mathbf{v}\cdot\nabla S)\cdot R \right) d\mathbf{x}\, dt \\ &= \int_0^{T_{max}} \int_\Omega (S\cdot R^\star + R\cdot S^\star)\cdot \Bigg( f(\mathbf{x},t) - \sum_{i=1}^{n} X_i\cdot\frac{\partial T_i}{\partial t} \\ &\quad + a\sum_{i=1}^{n}\Delta X_i\cdot T_i - \sum_{i=1}^{n}(\mathbf{v}\cdot\nabla X_i)\cdot T_i \Bigg) d\mathbf{x}\, dt. \end{aligned} \tag{3.8}$$

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 (3.7). Thus, equation (3.8) is written as

$$\begin{aligned} &\int_\Omega S^\star\cdot(\alpha_t S - a\beta_t\Delta S + \beta_t\ \mathbf{v}\cdot\nabla S)\, d\mathbf{x} \\ &= \int_\Omega S^\star\cdot\left( \gamma_t(\mathbf{x}) - \sum_{i=1}^{n}\alpha_t^i X_i + a\sum_{i=1}^{n}\beta_t^i\Delta X_i - \sum_{i=1}^{n}\beta_t^i\ \mathbf{v}\cdot\nabla X_i \right) d\mathbf{x}, \end{aligned} \tag{3.9}$$

where

$$\begin{cases} \alpha_t = \displaystyle\int_0^{T_{max}} R(t)\cdot\frac{\partial R}{\partial t}(t)\,dt \\ \alpha_t^i = \displaystyle\int_0^{T_{max}} R(t)\cdot\frac{\partial T_i}{\partial t}(t)\,dt \\ \beta_t = \displaystyle\int_0^{T_{max}} R^2(t)\,dt \\ \beta_t^i = \displaystyle\int_0^{T_{max}} R(t)\cdot T_i(t)\,dt \\ \gamma_t(\mathbf{x}) = \displaystyle\int_0^{T_{max}} R(t)\cdot f(\mathbf{x},t)\ dt;\ \forall \mathbf{x}\in\Omega. \end{cases} \tag{3.10}$$

The weak formulation (3.9) is satisfied for all $S^\star$, therefore we could come back to the associated strong formulation

$$\begin{aligned} &\alpha_t S - a\beta_t \Delta S + \beta_t\ \mathbf{v}\cdot\nabla S \\ &= \gamma_t - \sum_{i=1}^n \alpha_t^i X_i + a\sum_{i=1}^n \beta_t^i \Delta X_i - \sum_{i=1}^n \beta_t^i\ \mathbf{v}\cdot\nabla X_i, \end{aligned} \tag{3.11}$$

that one could solve by using any appropriate discretization technique for computing the space function $S(\mathbf{x})$.

- *Computing the function $R(t)$.*

From the function $S(\mathbf{x})$ just computed, we search $R(t)$. In this case $S^\star$ vanishes in (3.7) and equation (3.8) reduces to

$$\begin{aligned} &\int_0^{T_{max}}\int_\Omega (S\cdot R^\star)\cdot\left(S\cdot\frac{\partial R}{\partial t} - a\Delta S\cdot R + (\mathbf{v}\cdot\nabla S)\cdot R\right) d\mathbf{x}dt \\ &= \int_0^{T_{max}}\int_\Omega (S\cdot R^\star)\cdot\left(f(x,t) - \sum_{i=1}^n X_i\cdot\frac{\partial T_i}{\partial t}\right. \\ &\qquad \left. + a\sum_{i=1}^n \Delta X_i\cdot T_i - \sum_{i=1}^n (\mathbf{v}\cdot\nabla X_i)\cdot T_i\right) d\mathbf{x}dt, \end{aligned} \tag{3.12}$$

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

$$\begin{cases} \alpha_x = \displaystyle\int_\Omega S(\mathbf{x})\cdot\Delta S(\mathbf{x})\,d\mathbf{x} \\ \alpha_x^i = \displaystyle\int_\Omega S(\mathbf{x})\cdot\Delta X_i(\mathbf{x})\,d\mathbf{x} \\ \beta_x = \displaystyle\int_\Omega S^2(\mathbf{x})\,d\mathbf{x} \\ \beta_x^i = \displaystyle\int_\Omega S(\mathbf{x})\cdot X_i(\mathbf{x})\,d\mathbf{x} \\ \lambda_x = \displaystyle\int_\Omega S(\mathbf{x})\cdot(\mathbf{v}\cdot\nabla S(\mathbf{x}))\,d\mathbf{x} \\ \lambda_x^i = \displaystyle\int_\Omega S(\mathbf{x})\cdot(\mathbf{v}\cdot\nabla X_i(\mathbf{x}))\,d\mathbf{x}, \\ \gamma_x(t) = \displaystyle\int_\Omega S(\mathbf{x})\cdot f(\mathbf{x},t)\ d\mathbf{x};\ \ \forall t, \end{cases} \tag{3.13}$$

equation (3.12) reads

$$\int_0^{T_{max}} R^\star \cdot \left( \beta_x \frac{\partial R}{\partial t} + (\lambda_x - a\alpha_x)R - \gamma_x(t) \right.$$
$$\left. + \sum_{i=1}^{n} \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^{n} (\lambda_x^i - a\alpha_x^i) T_i \right) dt = 0. \tag{3.14}$$

As equation (3.14) holds for all $S^\star$, we could come back to the strong formulation

$$\beta_x \frac{\partial R}{\partial t} = (a\ \alpha_x - \lambda_x) R + \gamma_x(t) - \sum_{i=1}^{n} \beta_x^i \frac{\partial T_i}{\partial t} + \sum_{i=1}^{n} (a\ \alpha_x^i - \lambda_x^i) T_i, \tag{3.15}$$

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 $R^{(q)}(t)$ and $R^{(q-1)}(t)$ to be 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 is

$$e = \left\| R^{(q)}(t) \cdot S^{(q)}(\mathbf{x}) - R^{(q-1)}(t) \cdot S^{(q-1)}(\mathbf{x}) \right\|_2 < 10^{-8}, \tag{3.16}$$

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

We denote $Q_{n+1}$ to be 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 is written as

$$u(\mathbf{x},t) \approx \sum_{i=1}^{N} X_i(\mathbf{x}) \cdot T_i(t). \tag{3.17}$$

In all the numerical solutions computed in this work the global stopping criterion was

- For models whose exact solution $u^{ref}$ was known,

$$E = \frac{\|u - u^{ref}\|_2}{\|u^{ref}\|_2} < \epsilon, \tag{3.18}$$

- For models whose exact solution was not known,

$$E = \frac{\left\| \frac{\partial u}{\partial t} - a\Delta u + \mathbf{v} \cdot \nabla u - f(\mathbf{x},t) \right\|_2}{\|f(\mathbf{x},t)\|_2} < \epsilon, \tag{3.19}$$

with $\epsilon = 10^{-6}$ in all the simulations reported in this work.

**3.1.1. Discussion.** The strategy proposed above needs for the solution 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 desired 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 of the separated representation just discussed, one performs a standard incremental solution, $P$ $d$D models, $d=1,2,3$, must be solved ($P$ being the number of time steps, i.e. $P=T_{max}/\Delta t$). The time step in incremental strategies has a direct impact on the convergence and stability of the numerical scheme.

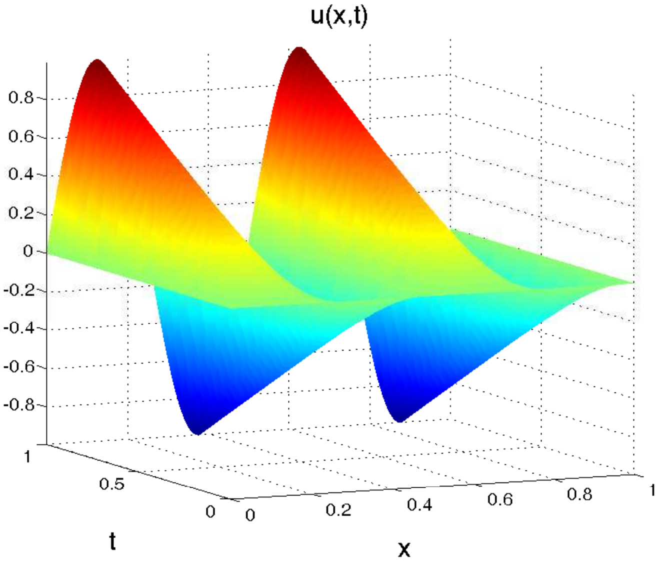

FIG. 3.1. *Reference solution $u(x,t)$*

In all the analyzed cases $N$ and $Q$ are of the order of tens which implies the solution of about hundred three-dimensional problems defined in $\Omega$, 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) is presented in section 3.1.3.

**3.1.2. Numerical results.** We consider the problem defined by equation (3.3) and (3.4) where $d=1$, $\Omega=(0,1)$, $\mathcal{I}=(0,T_{max}]$, $T_{max}=1$, $f=(16a\pi^2t+1)\sin(4\pi x)+4tv\pi\cos(4\pi x)$, $a=1$, $v=1$, $\Delta t=10^{-3}$ and $\Delta x=10^{-3}$ ($\Delta x$, and $\Delta t$ being the space and time steps respectively), whose exact solution $u_{ref}=t\sin(4\pi x)$ is depicted in figure 3.1.

A regular grid composed of 1001 nodes was considered on $\Omega$. For the analyzed model the number of time steps is not relevant because of the linear evolution of the exact solution on time. After the first enrichment the error was $7.3\cdot 10^{-7}$ (see figure 3.2). This fast convergence was expected because the model solution consists

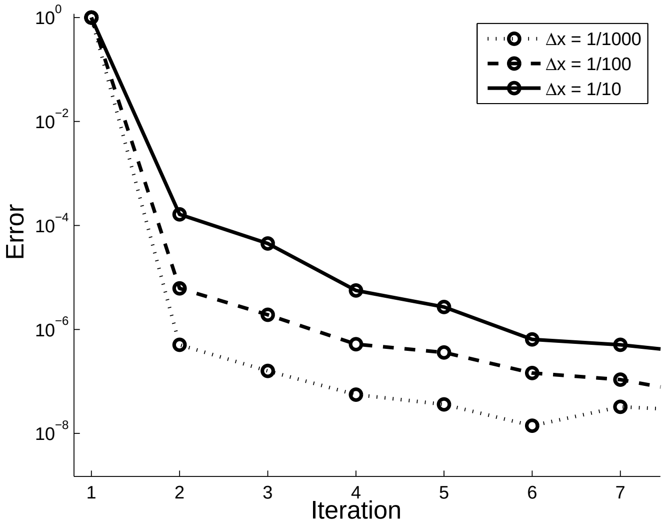

Fig. 3.2. *Convergence analysis*

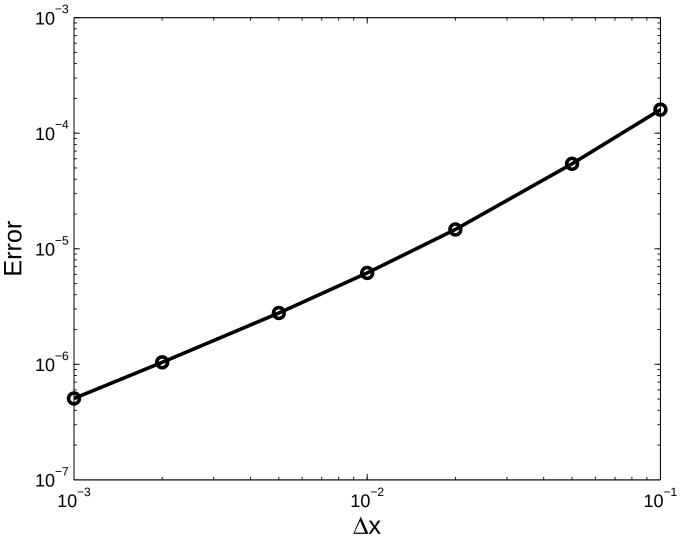

Fig. 3.3. *Error versus mesh density*

of a single product of a space and a time functions. The remaining error is induced by the discretization error. It is important to note that the error associated with the solution computed by using the separated representation is higher than the one associated to the a discrete solution related to a fully tensorial product of the space and time approximation bases.

The computation of other functional products do not significantly improve the error, because a single product of functions can represent the most features of the exact solution. However, as a piecewise linear approximation cannot represent properly the exact solution, a certain amount of error remains that cannot be suppressed by adding more sums in the separated decomposition. The only way to improve the accuracy lies in using finer meshes in space and time (in the present case as the solution is linear in time, finer meshes in time do not contribute to improve the accuracy). figure 3.2 illustrates the just described behavior: (i) the finer is the mesh the higher is the accuracy; and (ii) the first product reduces in some orders of magnitude the error, but when new products are added the error reduces slowly, approaching a horizontal asymptote. Finally, figure 3.3 depicts the evolution of the error with the mesh density. In this case as the threshold value defined in equation (3.18) is never reached, the iteration process is stopped as soon as the error cannot be reduced by adding additional products of functions in the finite sums decomposition.

The first functional product was computed after 20 iterations, that is, after solving 20 problems defined in space and other 20 defined in the time interval $\mathcal{I}$. Classical approaches based on incremental algorithms need the solution of $P = T_{max}/\Delta t = 1000$ linear systems (one at each time step) if one considers the same time step in both solvers. A deeper analysis will be carried out in the next section.

**3.1.3. Separated representation solver versus incremental strategies.** In this section we would like to discuss the comparison between the discretization technique based on the space-time separated representations and the standard incremental one.

| Time step | | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ | 0.2 | 0.5 |
|---|---|---|---|---|---|---|
| $\epsilon = 10^{-5}$ | Error | $3.35\ 10^{-7}$ | $1.25\ 10^{-7}$ | $1.9\ 10^{-5}$ | $1.1\ 10^{-3}$ | 0.066 |
| | N | 33 | 22 | 12 | 7 | 4 |
| | CPU | 10.58 | 2.62 | 0.76 | 0.34 | 0.22 |
| $\epsilon = 10^{-3}$ | Error | $7.01\ 10^{-7}$ | $1.14\ 10^{-5}$ | $2.98\ 10^{-5}$ | $1.2\ 10^{-3}$ | 0.067 |
| | N | 14 | 9 | 6 | 4 | 2 |
| | CPU | 2.14 | 0.59 | 0.34 | 0.22 | 0.17 |
| N=5 | Error | $2.95\ 10^{-5}$ | $1.87\ 10^{-5}$ | $9.26\ 10^{-5}$ | $1.1\ 10^{-3}$ | 0.066 |
| | N | 5 | 5 | 5 | 5 | 5 |
| | CPU | 0.46 | 0.406 | 0.328 | 0.296 | 0.296 |
| Implicit FEM | Error | $1.83\ 10^{-6}$ | $2.24\ 10^{-5}$ | $4.72\ 10^{-4}$ | $1.4\ 10^{-3}$ | $4.7\ 10^{-3}$ |
| | CPU | 42.95 | 4.296 | 0.468 | 0.265 | 0.125 |

TABLE 3.1. *Separated representation versus incremental strategies (the error is evaluated according to equation (3.18) and the CPU time is given in seconds).*

First we analyze problem (3.3) in $\Omega = (0,1) \times (0, T_{max} = 1]$, with the term source $f(x,t) = 1$ and a null velocity $v = 0$. Table 3.1 summarizes different simulation scenarios. These results do not seem very favorable to the separated representation technique, but we must recall that in the present comparison the space problem was one-dimensional. Obviously, the higher is the dimension of the physical space the more the expected CPU time savings are. We can also notice that the lower the time step, the higher the CPU time savings when using the separated representation. Obviously, the separated representation strategy is especially efficient when the solution can be represented by a reduced number of functional products. When it is not the case the standard incremental algorithms can be faster. We illustrate this situation in figure 3.4 in which we compare the convergence of the purely diffusion model ($a=1$ and $v=0$) — the dashed line — and the purely advection one ($a=0$ and $v=1$) — the continuous line. As we can notice the convergence rate is higher the the more elliptic the problem is. When the advection character increases the separated strategy could be more expensive that the standard one, but we must recall that in high dimensions the only practical possibility is the use of a separated representation because its complexity scales linearly with the space dimension, instead the exponential growing of standard mesh based discretization techniques.

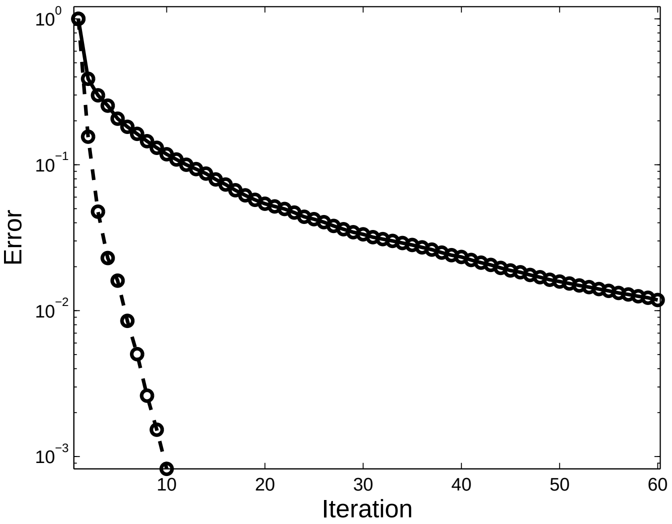

Fig. 3.4. *Convergence analysis: purely diffusion model (dashed line) versus purely advection model (continuous line).*

**3.2. Beam equation.** In this second example we consider the equation governing the deflection of a beam subjected to an evolving load which is the solution of an advection-diffusion problem. The beam problem, defined by

$$\frac{\partial^2 u}{\partial t^2}+a\frac{\partial^4 u}{\partial x^4}=f(x,t) \quad \text{in } \Omega\times\mathcal{I}=(0,L)\times(0,T_{max}], \tag{3.20}$$

and subjected to the following initial and boundary conditions

$$\begin{cases} u(\mathbf{x},t)=0 & (\mathbf{x},t)\in\partial\Omega\times(0,T_{max}] \\ \dfrac{\partial u}{\partial x}(\mathbf{x},t)=0 & (\mathbf{x},t)\in\partial\Omega\times(0,T_{max}] \\ u(\mathbf{x},0)=0 & \mathbf{x}\in\Omega \\ \dfrac{\partial u}{\partial t}(\mathbf{x},0)=0 & \mathbf{x}\in\Omega, \end{cases} \tag{3.21}$$

can easily be solved by applying the procedure described in Subsection 3.1.

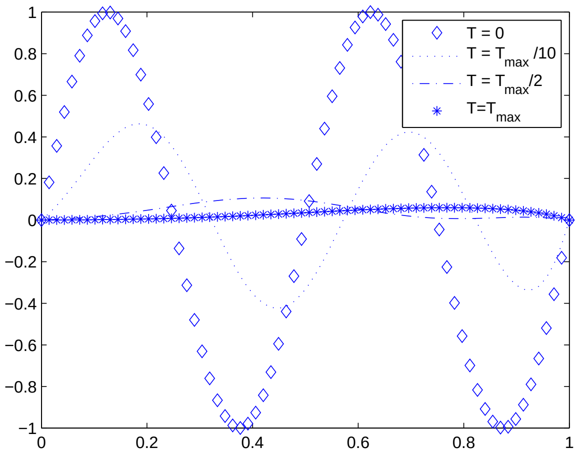

Fig. 3.5. *Evolving load at times:* $t=0$, $t=\frac{T_{max}}{10}$, $t=\frac{T_{max}}{2}$, *and* $t=T_{max}$

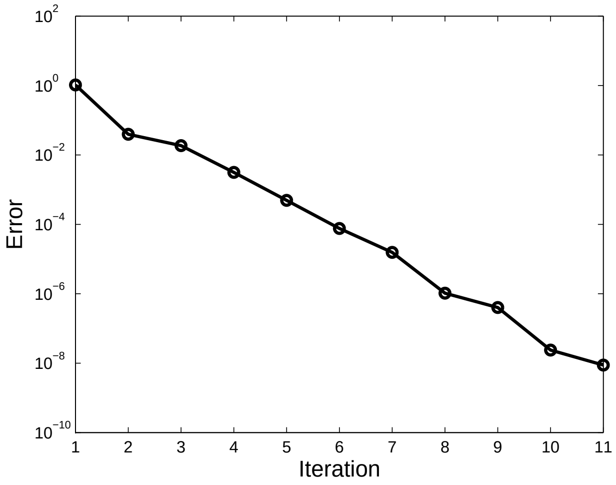

Fig. 3.6. *Convergence analysis for the beam bending problem*

We consider the problem defined by equation (3.20) and (3.21) where $\Omega=(0,1)$, $T_{max}=0.5$, $a=10^{-1}$, $\Delta t=10^{-3}$ and $\Delta x=1/70$. The load $f$ is the solution of an advection-diffusion problem whose solution is illustrated at different times in figure 3.5. The choice of this function was motivated by the interest of having a smooth function evolving in time and moving in space. The reference solution was in this case computed by using a fine enough finite different scheme.

The evolution of the residual with the number of functional products is depicted in figure 3.6 where the error was computed by using equation (3.18). As it can be noticed in this figure, only 8 functional couples were needed to approximate the

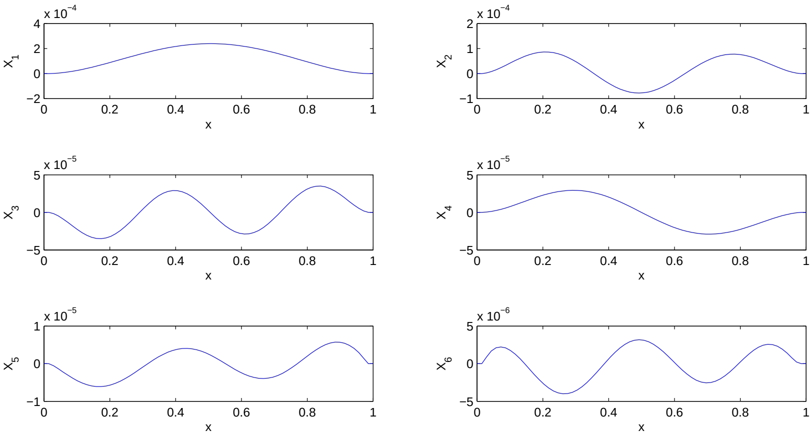

FIG. 3.7. *Functions* $X_1(x),\ldots,X_6(x)$

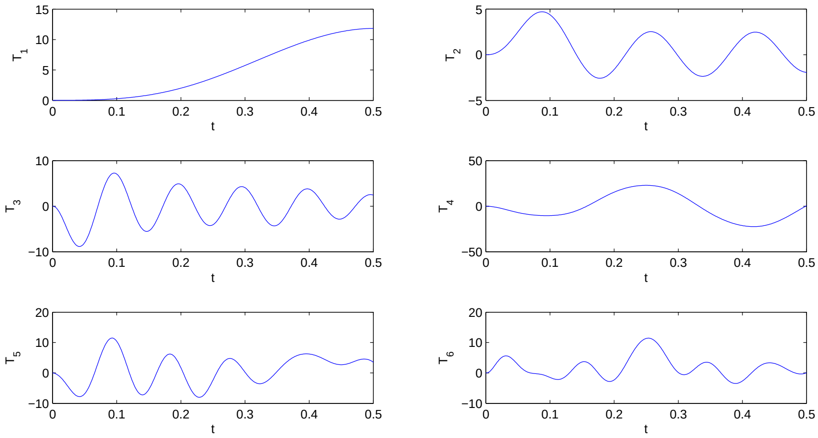

FIG. 3.8. *Functions* $T_1(t),\ldots,T_6(t)$

unknown field $u(x,t)$ with a precision of $10^{-6}$. The six most significant spatial and temporal functions are depicted in figures 3.7 and 3.8 respectively.

The solution procedure only required the solution of 87 space and time problems, for the given precision, instead the 500 required when using standard incremental algorithms, if the same time step is considered in both solution procedures.

**3.3. Accounting for non-linearities.** For illustrating the treatment of non linear models we are considering the following simple nonlinear parabolic problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} - a\Delta u = u^2 + f(\mathbf{x},t) & \text{in } \Omega \times (0,T_{max}], \\ u(\mathbf{x},t) \quad = 0 & \text{on } \partial\Omega \times (0,T_{max}], \\ u(\mathbf{x},0) \quad = 0 & \text{in } \Omega, \end{cases} \tag{3.22}$$

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 (3.22) by using a separated representation, we propose two alternatives

- An incremental linearization,
- A Newton linearization,

which we describe in sections below.

**3.3.1. Incremental linearization.** We look to write the solution of problem (3.22) 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 is written as

$$u(\mathbf{x},t) \approx \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) + S(\mathbf{x}) \cdot R(t). \tag{3.23}$$

The weak form of problem (3.22) is written as

$$\int_0^{T_{max}} \int_\Omega u^\star \left( \frac{\partial u}{\partial t} - a \Delta u - u^2 - f(\mathbf{x},t) \right) d\mathbf{x}\, dt = 0; \quad \forall u^\star. \tag{3.24}$$

The alternating directions scheme proceeds by calculating $S(\mathbf{x})$ from the temporal function $R(t)$ just computed and then updating $R(t)$ from the just computed $S(\mathbf{x})$. 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 of computing this term at the previous iteration, that is, assuming at the present iteration the following approximation of the non-linear term

$$u^2 \approx \left( \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2. \tag{3.25}$$

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

$$\int_\Omega S^\star \cdot \left( \alpha_t S - a \beta_t \Delta S + \sum_{i=1}^{n} \alpha_t^i X_i - a \sum_{i=1}^{n} \beta_t^i \Delta X_i - \Phi(\mathbf{x}) - \Gamma(\mathbf{x}) \right) d\mathbf{x} = 0, \tag{3.26}$$

where

$$\Phi(\mathbf{x}) = \int_0^{T_{max}} R(t) \cdot \left( \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 dt \tag{3.27}$$

and

$$\Gamma(\mathbf{x}) = \int_0^{T_{max}} R(t) \cdot f(\mathbf{x},t)\, dt, \tag{3.28}$$

whose strong form is written as

$$\alpha_t S - a \beta_t \Delta S = -\sum_{i=1}^{n} \alpha_t^i X_i + a \sum_{i=1}^{n} \beta_t^i \Delta X_i + \Phi(\mathbf{x}) + \Gamma(\mathbf{x}). \tag{3.29}$$

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

$$\beta_x \frac{\partial R}{\partial t} - a\alpha_x R + \sum_{i=1}^{n} \beta_x^i \frac{\partial T_i}{\partial t} - a\sum_{i=1}^{n} \alpha_x^i T_i - \Psi(t) - \Upsilon(t) = 0, \tag{3.30}$$

where

$$\Psi(t) = \int_\Omega S(\mathbf{x}) \cdot \left( \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) \right)^2 d\mathbf{x} \tag{3.31}$$

and

$$\Upsilon(t) = \int_\Omega S(\mathbf{x}) \cdot f(\mathbf{x},t)\, d\mathbf{x}. \tag{3.32}$$

**3.3.2. Newton linearization.** From now on we denote by $u^n$ the solution computed at iteration $n$, i.e.,

$$u^n(\mathbf{x},t) = \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t). \tag{3.33}$$

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

$$\frac{\partial \widetilde{u}}{\partial t} - a\Delta\widetilde{u} - 2u^n\widetilde{u} = -\left( \frac{\partial u^n}{\partial t} - a\Delta u^n - (u^n)^2 - f(\mathbf{x},t) \right), \tag{3.34}$$

whose weak formulation is written as

$$\begin{aligned} &\int_0^{T_{max}} \int_\Omega \widetilde{u}^\star \left( \frac{\partial \widetilde{u}}{\partial t} - a\Delta\widetilde{u} - 2u^n\widetilde{u} \right) d\mathbf{x}\,dt \\ &= \int_0^{T_{max}} \int_\Omega \widetilde{u}^\star \left( -\frac{\partial u^n}{\partial t} + a\Delta u^n + (u^n)^2 + f(\mathbf{x},t) \right) d\mathbf{x}\,dt, \quad \forall \widetilde{u}^\star. \end{aligned} \tag{3.35}$$

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 again apply the alternating directions method deeply described in the previous sections.

**3.3.3. Numerical results.** We consider the problem (3.22) where $d=1$, $\Omega = (0,1)$, $T_{max} = 1$, $f = (16a\pi^2 t + 1)\sin(4\pi x) - t^2 \sin^2(4\pi x)$, $\Delta t = 10^{-3}$, $\Delta x = 10^{-3}$, and $a = 1$, whose exact solution $u_{ref} = t\sin(4\pi x)$, is depicted in figure 3.1.

When we applied any of the just described strategies, the convergence was reached in 6 iterations as noticed in figure 3.9, that is, only 6 couples or functions were needed to approximate the solution at the prescribed precision. However, this problem deserves some additional comments. As just indicated the solution of this problem can be written as a single product of functions. At each iteration of the non-linear solver a new product of functions is added. Thus, as the non-linear solver converged in 6 iterations, 6 couples of functions were computed. Obviously, after convergence, if we apply the singular value decomposition to the computed solution (that consists of the sum of the 6 products of functions generated by the non-linear solver) the number of products is reduced to one.

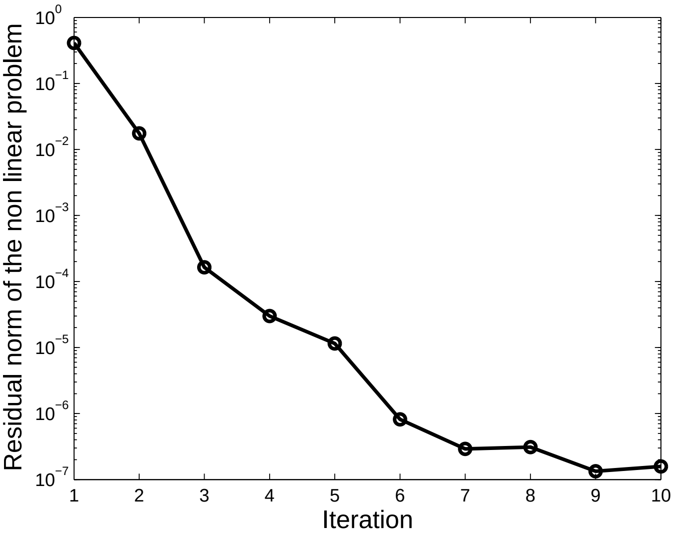

FIG. 3.9. *Convergence analysis of the non-linear model*

It is clear that in all cases, even for linear models, the separated representation algorithm is not optimal in the sense that even if the exact solution can be written from a certain number of functional products, the computed solution could involve a higher number of functional products because the approximation errors and the strategy used for computing the enrichment functions. However, in the case of non-linear models these effects are more noticeable because the number of functional products also depends strongly on the linearization scheme considered.

If instead of considering both linearization schemes just described, one considers the incremental linearization where the non-linear term is assumed given at the previous iteration of the alternating directions enrichment solver, that is, replacing equation (3.25) by

$$u^2 \approx \left( \sum_{i=1}^{n} X_i(\mathbf{x}) \cdot T_i(t) + \overline{S}(\mathbf{x}) \cdot \overline{R}(t) \right)^2, \qquad (3.36)$$

where $\overline{S}(\mathbf{x})$ and $\overline{R}(\mathbf{t})$ are the just computed enrichment functions at the previous iteration of the non-linear enrichment solver, and fixing the same level of precision in the solution, only one product of functions results after convergence. Despite this reduction on the number of sums in the separated representation, the computing time is similar to the one associated to the other strategies.

## 4. A general framework

In this section we propose a more general form to formulate the procedure previously described by using a tensor product notation.

Let $\Omega$ be a multidimensional domain involving $D$ coordinates $x_1,\ldots,x_D$ and $a(\Psi^\star,\Psi)$ and $b(\Psi^*)$ a bilinear and linear forms respectively defined in that multidimensional domain representing the weak formulation of a partial differential equation

$$a\left(\Psi^\star(x_1,x_2,\ldots,x_D),\Psi(x_1,x_2,\ldots,x_D)\right) = b(\Psi^*(x_1,x_2,\ldots,x_D)), \qquad (4.1)$$

whose solution is searched in the separated form

$$\Psi(x_1,x_2,\dots,x_D)\approx\sum_{i=1}^{N}F_1^i(x_1)\times F_2^i(x_2)\times\dots\times F_D^i(x_D). \tag{4.2}$$

It is important to recall that these coordinates are not necessarily defined in $\mathbb{R}$. In general $x_i\in\Omega_i\subset\mathbb{R}^{d_i}$ $(\Omega=\Omega_1\times\cdots\times\Omega_D)$. Moreover, one of these coordinates could eventually be the time.

The discrete form is written as

$$\mathbf{\Psi}^{*T}\mathcal{A}\mathbf{\Psi}=\mathbf{\Psi}^{*T}\mathcal{B}, \tag{4.3}$$

where

$$\mathcal{A}=\sum_{j=1}^{n_A}\mathbb{A}_1^j\otimes\mathbb{A}_2^j\otimes\dots\otimes\mathbb{A}_D^j, \tag{4.4}$$

$$\mathcal{B}=\sum_{j=1}^{n_B}\mathbf{B}_1^j\otimes\mathbf{B}_2^j\otimes\dots\otimes\mathbf{B}_D^j, \tag{4.5}$$

and we look for a solution that is written as

$$\mathbf{\Psi}=\sum_{j=1}^{N}\mathbf{F}_1^j\otimes\mathbf{F}_2^j\otimes\dots\otimes\mathbf{F}_D^j, \tag{4.6}$$

where $\mathbf{F}_i^j$ is the discrete form of function $F_i^j(x_i)$

The enrichment stage looks for a new product of $D$ functions, that for the sake of clarity we denote by $R_1,\dots,R_D$, that results in

$$\mathbf{\Psi}=\underbrace{\sum_{j=1}^{n}\mathbf{F}_1^j\otimes\mathbf{F}_2^j\otimes\dots\otimes\mathbf{F}_D^j}_{\mathbf{\Psi}_F}+\underbrace{\mathbf{R}_1\otimes\mathbf{R}_2\otimes\dots\otimes\mathbf{R}_D}_{\mathbf{\Psi}_R}. \tag{4.7}$$

Now, the alternating direction scheme, widely illustrated in the previous sections, is used. Each iteration of this non-linear solver consists of $D$ steps, each one updating one of the $\mathbf{R}$ functions. Thus, for updating the function $\mathbf{R}_j$ all the other functions $\mathbf{R}_1,\dots,\mathbf{R}_{j-1},\mathbf{R}_{j+1},\dots,\mathbf{R}_D$ are assumed known. Thus, at the present step the weighting function is written as

$$\Psi^*=\mathbf{R}_1\otimes\dots\otimes\mathbf{R}_{j-1}\otimes\mathbf{R}_j^*\otimes\mathbf{R}_{j+1}\otimes\dots\otimes\mathbf{R}_D. \tag{4.8}$$

The discrete form involves

$$\mathbf{\Psi}^*\mathcal{A}\mathbf{\Psi}_R=\sum_{k=1}^{n_A}\left(\mathbf{R}_j^{*T}\mathbb{A}_j^k\mathbf{R}_j\prod_{\substack{h=1\\h\neq j}}^{D}\mathbf{R}_h^T\mathbb{A}_h^k\mathbf{R}_h\right)=\mathbf{R}_j^{*T}\mathbb{K}\mathbf{R}_j, \tag{4.9}$$

$$\mathbf{\Psi}^*\mathcal{A}\mathbf{\Psi}_F=\sum_{i=1}^{n}\sum_{k=1}^{n_A}\left(\mathbf{R}_j^{*T}\mathbb{A}_j^k\mathbf{F}_j^i\prod_{\substack{h=1\\h\neq j}}^{D}\mathbf{R}_h^T\mathbb{A}_h^k\mathbf{F}_h^i\right)=\mathbf{R}_j^{*T}\mathbf{V},\tag{4.10}$$

$$\mathbf{\Psi}^*\mathcal{B}=\sum_{k=1}^{n_B}\left(\mathbf{R}_j^{*T}\mathbf{B}_j^k\prod_{\substack{h=1\\h\neq j}}^{D}\mathbf{R}_h^T\mathbf{B}_h^k\right)=\mathbf{R}_j^{*T}\mathbf{W},\tag{4.11}$$

that leads to the linear system

$$\mathbb{K}\mathbf{R}_j+\mathbf{V}=\mathbf{W}.\tag{4.12}$$

## 5. Solving computational rheology models

Dilute polymers are usually modelled using dumbbell models. In what follows we are considering one of the most used models, the FENE one, in which molecules are represented by two beads connected by a non-linear spring of finite extension.

The kinetic theory description of this model results in an advection-diffusion equation governing the evolution of the probability distribution function (PDF) $\Psi(\mathbf{x},t,\mathbf{q})$ that represents the fraction of molecules that at point $\mathbf{x}$, and time $t$ have a conformation (orientation and extension) represented by vector $\mathbf{q}$.

The physics related to this model can be found in numerous books (see for example [6]). To reduce the computing time associated to its solution we proposed in [2] the use of reduced approximations basis that were build-up making use of the proper orthogonal decomposition. That incremental procedure allowed to spectacular computing time savings. In this paper we analyze the use of separated representations to build the reduced approximation basis in a non incremental way.

In what follows, for the sake of simplicity, we are considering homogeneous flows (characterized by its velocity gradient $\nabla\mathbf{v}$) and molecules "living" in a 2D space. The homogeneity of the flow implies that the PDF will depend only on the conformation space — also assumed 2D — and on the time, i.e. $\Psi(t,\mathbf{q})$.

The advection-diffusion equation governing the evolution of $\Psi$, also known as the Fokker-Planck equation, can be written in the form

$$\mathcal{K}(\Psi(t,\mathbf{q}))+\mathcal{L}(\Psi(t,\mathbf{q}))=0,\tag{5.1}$$

where $\mathbf{q}\in\Omega_q=\mathcal{B}^{2D}(0,\sqrt{b})$ ($\mathcal{B}^{2D}(0,\sqrt{b})$ being the 2D-ball centered at the origin and whose radius $\sqrt{b}$ represents the maximum molecule stretching). $\mathcal{K}$ and $\mathcal{L}$ are the two differential operators involved in the Fokker-Planck equation [6]

$$\mathcal{L}(\Psi)=\frac{\partial}{\partial\mathbf{q}}\left(\left[\ \nabla\mathbf{v}\cdot\mathbf{q}-\frac{1}{2}f(\mathbf{q})\ \right]\ \Psi\right)-\frac{1}{2}\frac{\partial^2\Psi}{\partial\mathbf{q}^2},\tag{5.2}$$

$$\mathcal{K}=\frac{\partial}{\partial t},\tag{5.3}$$

where $f(\mathbf{q})$ represents the spring constant that describes the molecule stiffness [6]:

$$f(\mathbf{q})=\frac{\mathbf{q}}{1-\frac{q^2}{b}},\tag{5.4}$$

with $q=\|\mathbf{q}\|_2$.

**5.1. Simple shear flow.** First we consider the simple shear flow

$$\nabla \mathbf{v} = \begin{pmatrix} 0 & g \\ 0 & 0 \end{pmatrix}, \tag{5.5}$$

where $g$ denotes the shear rate, and we look for the solution of the Fokker-Planck equation by assuming a separated representation of the PDF:

$$\Psi(t,\mathbf{q}) \approx \sum_{i=1}^{N} F_i(t) \cdot G_i(\mathbf{q}). \tag{5.6}$$

To build-up this approximation we must follow the iteration procedure previously described, that at iteration $n$ looks for the enrichment functional product $R(t) \cdot S(\mathbf{q})$.

For applying the technique just described we consider the discrete form of all the functions involved in the separated representation of the PDF. Thus, functions $F_i(t)$ (respectively $G_i(\mathbf{q})$) and $R(t)$ (respectively $S(\mathbf{q})$) are defined using $1D$ (respectively $2D$) finite element interpolation in their associated domains $\Omega_t \equiv \mathcal{I} = (0, T_{max}]$ and $\Omega_q = \mathcal{B}^{2D}(0, \sqrt{b})$. We use for this purpose the vector $\mathbf{M}$ (respectively $\mathbf{N}$) containing the value of the shape functions associated to each space. Finally $\mathbf{F}_i$ (respectively $\mathbf{G}_i$) and $\mathbf{R}$ (respectively $\mathbf{S}$) are the nodal descriptions of the associated functions.

Then, we define the following matrix

$$\begin{aligned} &\mathbb{A}_1^1 = \int_{\Omega_t} \mathbf{M} \mathcal{K} \mathbf{M}^T d\Omega_t, \ \mathbb{A}_2^1 = \int_{\Omega_q} \mathbf{N}\mathbf{N}^T d\Omega_q, \\ &\mathbb{A}_1^2 = \int_{\Omega_t} \mathbf{M}\mathbf{M}^T d\Omega_t, \ \ \mathbb{A}_2^2 = \int_{\Omega_q} \mathbf{N} \mathcal{L} \mathbf{N}^T d\Omega_q. \end{aligned} \tag{5.7}$$

The former expressions contain all the needed stabilizations of advective terms, and the tensorial problem form is then given by

$$\begin{aligned} &\mathcal{B} = 0, \\ &\mathcal{A} = \mathbb{A}_1^1 \otimes \mathbb{A}_2^1 + \mathbb{A}_1^2 \otimes \mathbb{A}_2^2. \end{aligned} \tag{5.8}$$

**5.2. Small amplitude oscillatory flow.** The obtention of the so called by rheologists “linear viscoelastic data” is of great interest in rheology in order to characterize complex fluids. Here, linear viscoelasticity means that the flow only implies small deformations. In any case, and independently of the magnitude of deformations that we are applying (small or large), the Fokker-Planck equation that we are solving is always linear. In rheology the so called linear viscoelastic regime means that the enforced kinematics is a small amplitude oscillatory flow. However, as just indicated, one could solve the Fokker-Planck equation for any flow by using the separated representation discretization scheme previously described.

Thus, several small amplitude oscillatory flows must be solved, each one characterized by a frequency varying in a large interval of several decades. For each value of the applied frequency, the flow kinematics is given by

$$\nabla \mathbf{v} = \begin{pmatrix} 0 & g\sin(\omega t) \\ 0 & 0 \end{pmatrix}, \tag{5.9}$$

where $g$ is the maximum shearing rate and $\omega$ the applied frequency. The operator $\mathcal{L}(\Psi)$ can be decomposed as

$$\mathcal{L}(\Psi) = \mathcal{L}_0(\Psi) + g\sin(\omega t)\mathcal{L}_1(\Psi), \tag{5.10}$$

where both operators $\mathcal{L}_0$ and $\mathcal{L}_1$ are time independent. Thus, we can define the new matrices

$$\begin{aligned}
&\mathbb{A}_1^1 = \int_{\Omega_t} \mathbf{M}\mathcal{K}\mathbf{M}^T d\Omega_t \mathbb{A}_2^1 = \int_{\Omega_q} \mathbf{N}\mathbf{N}^T d\Omega_q,\\
&\mathbb{A}_1^2 = \int_{\Omega_t} \mathbf{M}\mathbf{M}^T d\Omega_t \mathbb{A}_2^2 = \int_{\Omega_q} \mathbf{N}\mathcal{L}_0\mathbf{N}^T d\Omega_q,\\
&\mathbb{A}_1^3 = \int_{\Omega_t} \sin(\omega t)\mathbf{M}\mathbf{M}^T d\Omega_t \mathbb{A}_2^3 = g\int_{\Omega_q} \mathbf{N}\mathcal{L}_1\mathbf{N}^T d\Omega_q.
\end{aligned} \tag{5.11}$$

Thus the tensorial form of the problem is given by

$$\begin{aligned}
&\mathcal{B} = 0,\\
&\mathcal{A} = \mathbb{A}_1^1 \otimes \mathbb{A}_2^1 + \mathbb{A}_1^2 \otimes \mathbb{A}_2^2 + \mathbb{A}_1^3 \otimes \mathbb{A}_2^3.
\end{aligned} \tag{5.12}$$

As soon as the PDF is known, the stresses can be computed by invoking Kramer's rule

$$\boldsymbol{\tau}(t) = \int_{\Omega_q} \Psi(t,\mathbf{q})\ (\mathbf{q} \otimes f(\mathbf{q}))\ d\Omega_q. \tag{5.13}$$

**5.3. Results and discussion.** A simple shear flow, with unit shear rate ($g=1$), was applied to a quiescent FENE model whose PDF is written as $\Psi_0(\mathbf{q})$ [6]

$$\Psi_0(\mathbf{q}) = \frac{\left(1-\frac{q^2}{b}\right)^{\frac{b}{2}}}{\frac{2\pi b}{2+b}}, \tag{5.14}$$

where again $q = \|\mathbf{q}\|_2$. In what follows the parameter $b$ was set to $b=10$.

The separated representation of the probability distribution function $\Psi(t,\mathbf{q})$ is sought in the form:

$$\Psi(t,\mathbf{q}) \approx F_1(t)\cdot G_1(\mathbf{q}) + \sum_{i=2}^{N} F_i(t)\cdot G_i(\mathbf{q}), \tag{5.15}$$

where

$$\begin{cases} F_1(t) = 1\\ G_1(\mathbf{q}) = \Psi_0(\mathbf{q}) \end{cases} \tag{5.16}$$

in order to facilitate the enforcement of the initial condition. Thus, during the enrichment stage $F_i(t=0)=0$, $\forall i \geq 2$.

Figure 5.1 proves that the initial condition was properly enforced. Obviously, after convergence, the probability distribution function is positive everywhere, but the different modes are not necessary strictly positive.

The separated representation allows us to compute the most significant functions of space and time that approximate the model solution. The discrete model consists of thousand time steps uniformly distributed over 10 periods and about 3000 nodes in $\Omega_q$. The five most significative functions are depicted in figure 5.1, that also depicts the PDF at $t=T_{max}$, the first normal stress difference and the shear stress evolutions.

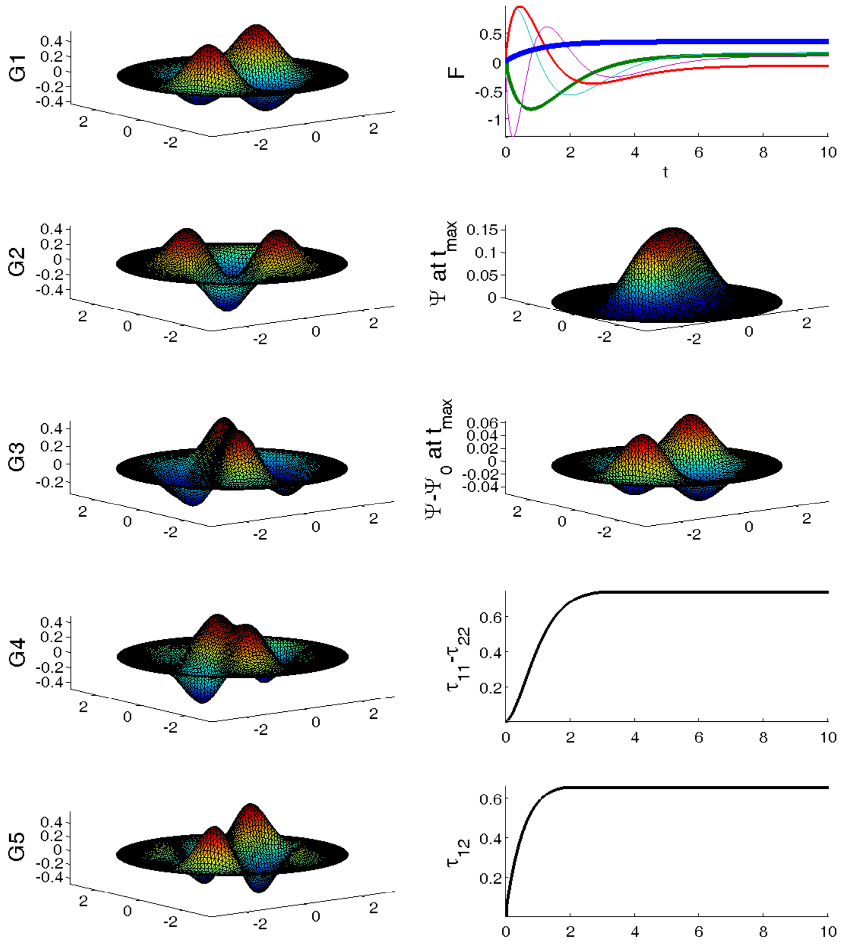

FIG. 5.1. *Start-up shear flow with unit shear rate*

This solution is in perfect agreement to the solution computed by using any standard incremental strategy.

The obtention of "linear viscoelasticity" data is a challenging problem when standard incremental methods are used because the flow related to numerous frequencies must be simulated (within an interval covering several decades) and for each frequency the transient simulation must be performed within a time interval large enough to ensure the response stabilization. Moreover, the higher is the frequency, the shorter the time step must be. Figure 5.2 illustrates the separated representation, the resulting PDF at the final time ($t=T_{max}$), and the shear stress when $\omega=2.55$ and $g=0.1$.

By knowing the input and the output (strain amplitude, shear stress amplitude,

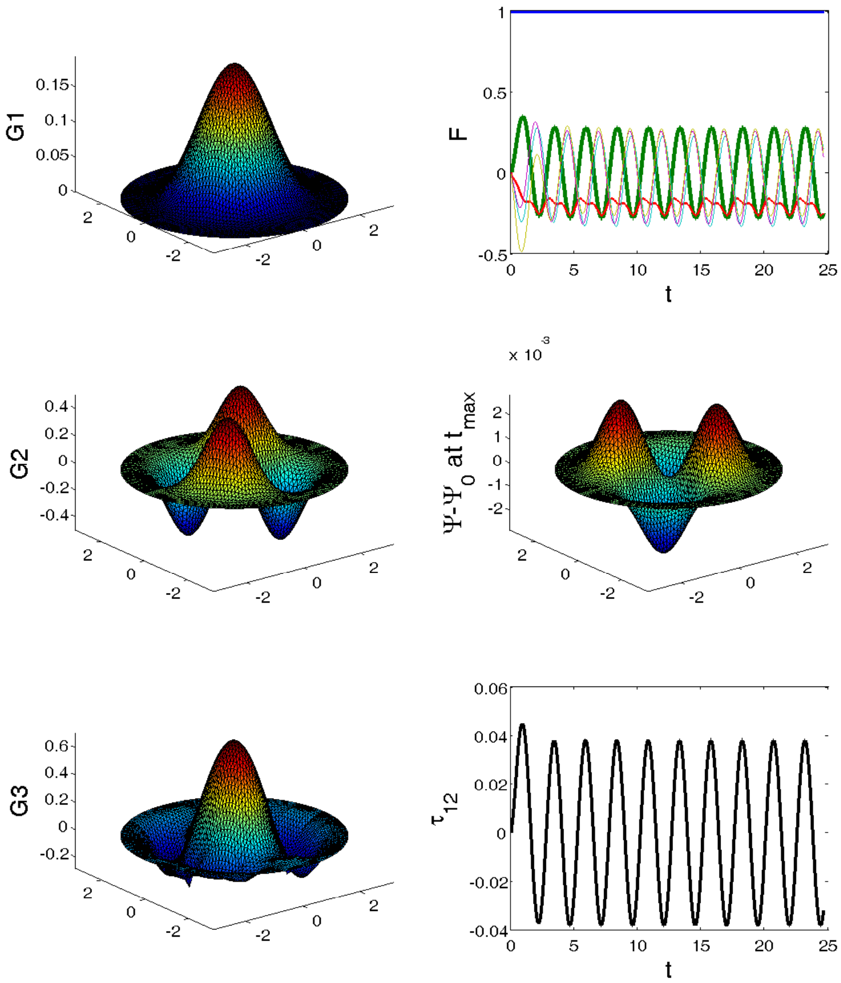

FIG. 5.2. *Small amplitude oscillatory flow with* $\omega = 2.55$ *and* $g = 0.1$

and phase angle) one could compute the elastic and viscous modulus $G'$ and $G''$ respectively (a key information for rheologists). Thus, the solution depicted in figure 5.2 (which requires an intensive computation) allows the obtention of one of the points of the modulus versus frequency curves depicted in figure 5.3. Obviously, the curves depicted in these figures needed the solution of the transient Fokker-Planck equation for numerous frequencies.

The separated representation allowed a fantastic reduction of the CPU time in the order of hundreds. Besides, the simulation using the separated representation allows to treat one, two, three, or ten periods using almost the same number of functions and consequently the same CPU time. However, the analysis of $n$-periods by using a

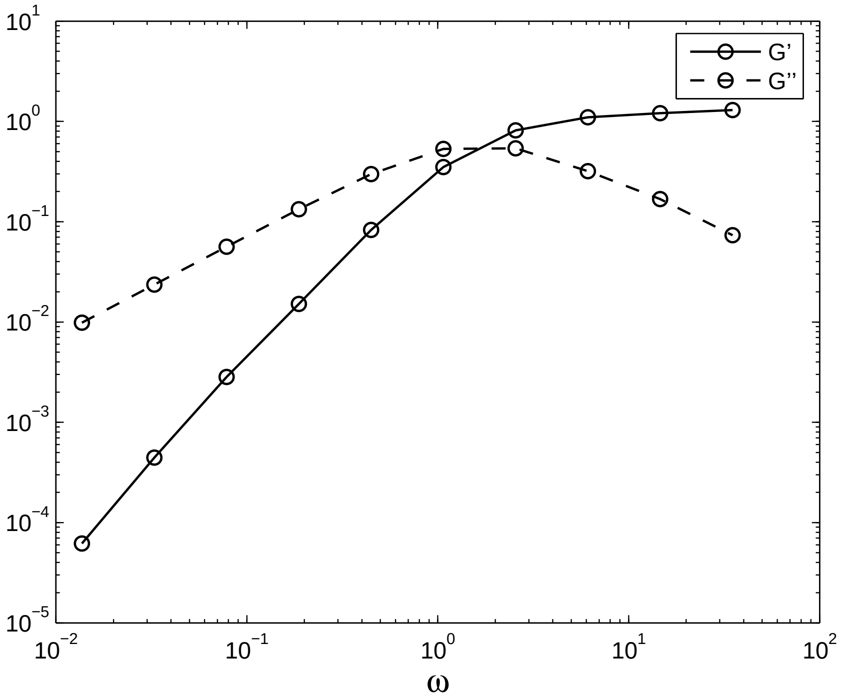

FIG. 5.3. *Real ($G'$) and complex ($G''$) elastic modulus*

standard incremental time integration scheme needs $n$-times the CPU time requested to analyze a single period.

## 6. Conclusions

Some equations encountered in science and engineering are susceptible to treatment by using a separated representation. We proposed in some of our former works a technique able to build-up a reduced separated representation of such models, and even if this procedure is not a universal solution strategy, at least it allows solving numerous models which until now required enormous computational resources. Thus, it allows to circumvent, in the cases in which it can be applied, the terrific curse of dimensionality associated to models defined in highly multidimensional spaces. On the other hand, even when models are defined in spaces of moderate dimensions, sometimes transient simulations must be computed in large time intervals using extremely small time steps because the stability constraints. In that case the separated representation could also alleviate the computational complexity by leading to efficient non-incremental strategies with the associated impressive computing time savings.

In this paper we illustrated the separated representation of transient models proving its potentiality for treating complex models coming from the modelling of complex fluids within the kinetic theory framework that result in highly multidimensional models whose transient solutions must be computed in large time intervals and for numerous loadings (linear viscoelastic analysis).

It is too early to quantify the computing time savings because at present the codes are far from optimized. However, from the nature of the algorithm it is quite natural expecting impressive computing time savings when the treated model accepts a separated representation. This consideration opens new questions: when does a model accept a separated representation? These kind of questions needs further mathematical analysis.

## REFERENCES

[1] Y. Achdou and O. Pironneau, *Computational Methods for Option Pricing*, SIAM Frontiers in Applied Mathematics, 2005.

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

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

[4] A. Ammar, B. Mokdad, F. Chinesta and R. Keunings, *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids. Part II: transient simulation using space-time separated representations*, J. Non-Newtonian Fluid Mech., 144, 98–121, 2007.

[5] G. Beylkin and M. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26/6, 2005, 2133–2159.

[6] B.B. Bird, C.F. Curtiss, R.C. Armstrong and O. Hassager, *Dynamics of Polymeric Liquids*, John Wiley & Sons, 2: Kinetic Theory, 1987.

[7] H.J. Bungartz and M. Griebel, *Sparse grids*, Acta Numerica, 13, 1–123, 2004.

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

[9] F. Chinesta, A. Ammar, F. Lemarchand, P. Beauchene and F. Boust, *Alleviating mesh constraints: Model reduction, parallel time integration and high resolution homogenization*, Comput, Meth. Appl. Mech. Engr., 197/5, 400–413, 2008.

[10] F. Chinesta, A. Ammar and P. Joyot, *The nanometric and micrometric scales of the structure and mechanics of materials revisited: an introduction to the challenges of fully deterministic numerical descriptions*, International Journal for Multiscale Computational Engineering, in press.

[11] T.G. Kolda and B.W. Bader, *Tensor decompositions and applications*, Technical Report SAND2007-6702, November, 2007.

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

[13] L. De Lathauwer, B. De Moor and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21/4, 1253–1279, 2000.

[14] Y. Maday and E.M. Ronquist, *The reduced basis element method: application to a thermal fin problem*, SIAM J. Sci. Comput., 26/1, 240–258, 2004.

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

[16] H.M. Park and D.H. Cho, *The use of the Karhunen-Love decomposition for the modelling of distributed parameter systems*, Chem. Engineer. Science, 51, 81–98, 1996.

[17] T.M. Rassias and J. Simsa, *Finite Sums Decompositions in Mathematical Analysis*, John Wiley and Sons Inc., 1995.

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

[19] D. Ryckelynck, *A priori hyperreduction method: an adaptive approach*, J. Comput. Phys., 202, 346–366, 2005.

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