On the use of proper generalized decompositions for solving the multidimensional chemical master equation

Francisco Chinesta* — Amine Ammar — Elías Cueto*****

** EADS Corporate Fundation Internantional Chair*
GEM, UMR CNRS - Centrale Nantes
1 rue de la Noe, BP 92101, 44321 Nantes cedex 3, France

*** Laboratoire de Rhéologie*
1301 rue de la piscine, BP 53 Domaine universitaire
38041 Grenoble cedex 9, France

**** Group of Structural Mechanics and Materials Modelling*
Aragón Institute of Engineering Research (I3A)
Universidad de Zaragoza. Maria de Luna, 3
E-50018 Zaragoza, Spain

ABSTRACT. *In this paper we review the possibilities associated with the use of Proper Generalized Decompositions for solving models established in highly multidimensional spaces. This technique has also been recently extended to problems that can be, under some circumstances, seen as multidimensional.*

RÉSUMÉ. *Dans ce travail on passe en revue les possibilités offertes dans le cadre de la PGD (Proper Generalized Decomposition) pour la résolution de modèles définis dans des espaces multidimensionnels. L'application de cette dernière technique est étendue ici aux réactions chimiques avec un petit nombre de molécules réactives. Ces modèles sont gouvernés par des équations appelées « équations maîtresses ». Ils présentent une difficulté liée à un caractère fortement multidimensionnel que nous sommes parvenus à surmonter.*

KEYWORDS: *proper generalized decomposition, multidimensional models, separated representation, cell signalling.*

MOTS-CLÉS : *proper generalized decomposition, modèles multidimensionnels, représentation séparée, signalisation cellulaire.*

DOI:10.3166/EJCM.19.53-64 © 2010 Lavoisier, Paris

1. Introduction

Models defined in highly multidimensional spaces are ubiquitous in many branches of Sciences and Engineering. Beginning from the most classical description of atomic structure, arising from Schrödinger equation, plenty of models present this unique feature of being defined in spaces with a number of dimensions notably high than three. Other examples of this feature include many models for complex fluid modelling, or the modelling of chemical reactions at very low concentrations (*i.e.*, those governed by the Chemical Master Equation).

The main difficulty related to these models concerns the difficulty of establishing a (finite element, finite difference) mesh in such a high number of dimensions. Consider, for instance, a one-dimensional problem whose numerical description involves, say, ten finite elements. If the model is extended to two-dimensional settings, the mesh will be composed by 10×10 elements. If the problem becomes three-dimensional the mesh increases to 10^3 elements, and so on. In the limit, for an 80-dimensional space, a hardly imaginable mesh of 10^{80} elements would be necessary to solve the problem. But 10^{80} is precissely the presumed number of elementary particles in the universe, so " No computer existing, or that will ever exist, can break this barrier because it is a catastrophe of dimension" (Laughlin *et al.*, 2000).

This frustrating characteristic of highly-dimensional models has given rise to the so-called *curse of dimensionality*. One possible solution lies in the use of sparse grids (Bungartz *et al.*, 2004). However the use of sparse grid is restricted to models with moderate multidimensionality (up to 20). Another technique able to circumvent, or at least alleviate, the curse of dimensionality consists of using a separated representation of the unknown field. Basically, the separated representation of a generic function $u(\mathbf{x}_1, \cdots, \mathbf{x}_D)$ (also known as finite sum decomposition) writes :

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

Note that the coordinates $\mathbf{x}_i$, $i = 1, \cdots, D$, are defined in spaces of moderate dimension, *i.e.* $\mathbf{x}_i \in \Omega_i \subset \mathbb{R}^{d_i}$, $d_i \leq 3$. Thus, the dimension of the model results $\sum_{i=1}^{i=D} d_i$. Eventually, one of these coordinates could be the time $t \in \mathcal{I} \subset \mathbb{R}^+$.

This kind of representation is not new, it was widely employed in the last decades in the framework of quantum chemistry. In particular the Hartree-Fock (that involves a single product of functions) and post-Hartree-Fock approaches (as the MCSCF that involves a finite number of sums) made use of a separated representation of the wavefunction (Cancès *et al.*, 2003) (Chinesta *et al.*, 2008).

We proposed recently a technique able to construct, in a way completely transparent for the user, the separated representation of the unknown field involved in a multidimensional partial differential equation. This technique, originally described and applied to multi-bead-spring FENE models of polymeric liquids in Ammar *et al.* (2006), was extended to transient models of such complex fluids in Ammar *et* *al.* (2007). More complex models (involving different couplings and non-linearities) based on the reptation theory of polymeric liquids were analyzed in Mokdad *et al.* (2007).

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

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

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

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

This space-time separated representation is not a new proposal. In fact such decompositions were proposed many year ago by Pierre Ladeveze as an ingredient of the powerful non-linear-non-incremental LATIN solver that he proposed in the 80s. During the last twenty years many works were successfully accomplished by the Ladeveze's group. The interested reader can refer to (Ladeveze, 1999) and the references therein. In the radial approximation approach (the name given in the pioneer works of Ladeveze) functions depending on space and the ones depending on time were a priori unknown, and they were computed by an appropriate minimization technique.

This paper reviews some of the basic features of the method, together with some interesting applications in different fields.

2. Basics of the proper generalized decomposition

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

$$u(x_1, \cdots, x_D) \approx \sum_{i=1}^{i=N} F_1^i(x_1) \times \cdots \times F_D^i(x_D) \qquad [3]$$

This kind of approximation only needs 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.

The technique that we proposed for computing the different functions involved in Equation [3)] consists of an alternating directions linearization strategy that we summarize here. For the sake of clarity, and without any loss of generality, we restrict our discussion to the D-dimensional Poisson's equation :

$$\triangle u = -f(x_1, x_2, ..., x_D), \tag{4}$$

where u is a scalar function of $(x_1, x_2, ..., x_D)$. Problem [4] is defined in the domain $(x_1, x_2, ..., x_D) \in \Omega = (-L, +L)^D$ with vanishing essential boundary conditions. The problem solution can be written in the form :

$$u(x_1, x_2, ..., x_D) = \sum_{j=1}^{\infty} \alpha_j \prod_{k=1}^{D} F_{kj}(x_k), \tag{5}$$

where F_{kj} is the j^{th} basis function, with unit norm, which only depends on the k^{th} coordinate.

It is well known that the solution of numerous problems can be accurately approximated using a finite (sometimes very reduced) number (N) of approximation functions, *i.e.* :

$$u(x_1, x_2, ..., x_D) \approx \sum_{j=1}^{N} \alpha_j \prod_{k=1}^{D} F_{kj}(x_k). \tag{6}$$

The previous expression implies the same number of approximation functions in each dimension, but each one of these functions could be expressed in a discrete form using different number of parameters (nodes of the 1D grids).

Now, an appropriate numerical procedure is needed for computing the coefficients α_j as well as the N approximations functions in each dimension.

The proposed numerical scheme consists of an iteration procedure that solves at each iteration n the following three steps :

Step 1 : *Projection of the solution in a discrete basis*

If we assume the functions $F_{kj} (\forall j \in [1, ..., n]; \forall k \in [1, ..., D])$ known (verifying the boundary conditions), the coefficients α_j can be computed by introducing the approximation of u into the Galerkin variational formulation associated with Equation [4] :

$$\int_\Omega \nabla u^* \cdot \nabla u d\Omega = \int_\Omega u^* f \, d\Omega \tag{7}$$

Introducing the approximation of u and u^* :

$$u(x_1, x_2, ..., x_D) = \sum_{j=1}^{n} \alpha_j \prod_{k=1}^{D} F_{kj}(x_k) \qquad [8]$$

and

$$u^*(x_1, x_2, ..., x_D) = \sum_{j=1}^{n} \alpha_j^* \prod_{k=1}^{D} F_{kj}(x_k) \qquad [9]$$

we have

$$\int_\Omega \nabla \left(\sum_{j=1}^{n} \alpha_j^* \prod_{k=1}^{D} F_{kj}(x_k) \right) \cdot \nabla \left(\sum_{j=1}^{n} \alpha_j \prod_{k=1}^{D} F_{kj}(x_k) \right) d\Omega =$$

$$= \int_\Omega \left(\sum_{j=1}^{n} \alpha_j^* \prod_{k=1}^{D} F_{kj}(x_k) \right) f \; d\Omega \qquad [10]$$

Now, we assume that $f(x_1, \cdots, x_D)$ can be written in the form

$$f(x_1, \cdots, x_D) \approx \sum_{h=1}^{m} \prod_{k=1}^{D} f_{kh}(x_k) \qquad [11]$$

Equation [10] involves integrals of a product of D functions each one defined in a different coordinate. Let $\prod_{k=1}^{D} g_k(x_k)$ be one of these functions to be integrated. The integral over Ω can be performed by integrating each function in its definition interval and then multiplying the D computed integrals according to :

$$\int_\Omega \prod_{k=1}^{D} g_k(x_k) \; d\Omega = \prod_{k=1}^{D} \int_{-L}^{L} g_k(x_k) dx_k \qquad [12]$$

which makes possible the numerical integration in highly dimensional spaces.

Now, due to the arbitrariness of the coefficients α_j^*, Equation [10] allows to compute the n-approximation coefficients α_j, solving the resulting linear system of size $n \times n$. This problem is linear and moreover rarely exceeds the order of tens of degrees of freedom. Thus, even if the resulting coefficient matrix is densely populated, the time required for its solution is negligible with respect to the one required for performing the approximation basis enrichment (step 3).

Step 2 : *Checking convergence*

From the solution of u at iteration n given by Equation [8] we compute the residual Re related to Equation [4] :

$$Re = \frac{\sqrt{\int_\Omega \left(\triangle u + f(x_1, \cdots, x_D)\right)^2}}{\|u\|} \tag{13}$$

If $Re < \epsilon$ (epsilon is a small enough parameter) the iteration process stops, yielding the solution $u(x_1, \cdots, x_D)$ given by Equation [8]. Otherwise, the iteration procedure continues.

The integral in Equation [13] can be written as the product of one-dimensional integrals by performing a separated representation of the square of the residual.

Step 3 : *Enrichement of the approximation basis*

From the coefficients α_j just computed the approximation basis can be enriched by adding the new function $\prod_{k=1}^{D} F_{k(n+1)}(x_k)$. For this purpose we solve the non-linear Galerkin variational formulation related to Equation [4] :

$$\int_\Omega \nabla u^* \cdot \nabla u d\Omega = \int_\Omega u^* f \; d\Omega \tag{14}$$

using the approximation of u given by :

$$u(x_1, x_2, ..., x_D) = \sum_{j=1}^{n} \alpha_j \prod_{k=1}^{D} F_{kj}(x_k) + \prod_{k=1}^{D} R_k(x_k) \tag{15}$$

The weighting function can be expressed as :

$$u^*(x_1, x_2, ..., x_D) =$$

$$= R_1^*(x_1) \times R_2(x_2) \times \cdots \times R_D(x_D) + \cdots + R_1(x_1) \times R_2(x_2) \times \cdots \times R_D^*(x_D) \tag{16}$$

This leads to a non-linear variational problem, whose solution allows to compute the D functions $R_k(x_k)$. Functions $F_{k(n+1)}(x_k)$ are finally obtained by normalizing, after convergence of the non-linear solver, the functions $R_1, R_2, ..., R_D$.

To solve this problem we introduce a discretization of those functions $R_k(x_k)$. Each one of these functions is approximated using a 1D finite element description. If we assume than p_k nodes are used to construct the interpolation of function $R_k(x_k)$

in the interval $[-L, L]$, then the size of the resulting discrete non-linear problem is $\sum_{k=1}^{k=D} p_k$. The price to pay for avoiding a whole mesh in the multidimensional domain is the solution of a non-linear problem. However, even in high dimensions the size of the non-linear problems remains moderate and no particular difficulties have been found in its solution up to hundreds dimensions. Concerning the computation time, even when the non-linear solver converges quickly, this step consumes the main part of the global computing time.

Different non-linear solvers have been analyzed : fixed-point, Newton or one based on an alternating directions scheme. In this work the last strategy was retained. Thus, in the enrichment step, function $R_1^{s+1}(x_1)$ is updated by assuming known all the others functions (given at the previous iteration of the non-linear solver $R_2^s(x_2), \cdots, R_D^s(x_D)$). Then, functions $R_1^{s+1}(x_1), R_3^s(x_3), \cdots, R_D^s(x_D)$ are assumed known for updating function $R_2^{s+1}(x_2)$, and so on until updating the last function $R_D^{s+1}(x_D)$. Now the convergence is checked by calculating $\sum_{i=1}^{i=D} \|R_i^{s+1}(x_i) - R_i^s(x_i)\|^2$. If this norm is small enough we can define the functions $F_{k(n+1)}(x_k)$ by normalizing the functions $R_1, R_2, ..., R_D$ and come back to step 1. On the contrary, if this norm is not small enough, a new iteration of the non-linear solver should be performed by updating functions $R_i^{s+2}(x_i), \ i = 1, \cdots, D$ and then checking again the convergence. Despite its simplicity, our experience proves that this strategy is in fact very robust.

3. Application of the PGD to the simulation of cell signaling processes

When chemically reacting species are present at very low concentrations (in the number of tens or hundreds of molecules, for instance) the resulting state can not be modeled accurately as deterministic and the inherent randomness of the system should be taken into account. This is the case, for instance, when modeling gene regulatory networks. It is well known that small numbers of molecules can alter these networks significantly (Hasty *et al.*, 2001) (Sreenath *et al.*, 2008).

It is also well known that under some circumstances (a well stirred mixture, fixed volume and fixed temperature), such a system can be considered Markovian, and that it is governed by the so-called Chemical Master Equation (CME) (Munsky *et al.*, 2006), which is a set of linear ordinary differential equations.

3.1. *The chemical master equation*

When dealing with chemical system in which the different species are present at very low copy numbers, it is necessary to work with the number of molecules present at each time instant, instead of working, as usual, with the concentration of each species. Thus, consider a well mixed system, at constant volume and temperature, described by the state vector

$$\boldsymbol{Z}(t) = (\#A, \#B, \#C, \#D, \ldots)^T, \qquad [17]$$

with initial state $\boldsymbol{Z}(t_0) = \boldsymbol{z}_0$. Here, A, B, etc., represent different chemical species. When some reaction r_j occurs, the system moves from $\boldsymbol{z}_0$ to $\boldsymbol{z}^*$, where the change in the number of molecules is equal to its stoichiometry in the reaction r_j :

$$\begin{aligned} r_j : {} & s^{\text{in}}_{1,j} x_1 + s^{\text{in}}_{2,j} x_2 + \cdots + s^{\text{in}}_{N,j} x_N \\ & \xrightarrow{k_j} s^{\text{out}}_{1,j} x_1 + s^{\text{out}}_{2,j} x_2 + \cdots + s^{\text{out}}_{N,j} x_N, \end{aligned} \qquad [18]$$

with k_j the rate constant of reaction r_j. In turn, $s^{\text{in,out}}_{i,j}$ represent the stoichiometric coefficients of the i-th species and x_k represent the concentrations of each biochemical species.

This state transition depends on the probability that the changes due to any reaction occur as described by the *propensity function*. For any reaction j :

$$a_j(\boldsymbol{z})dt \equiv \text{ the probability, given } \boldsymbol{Z}(t) = \boldsymbol{z}, \text{ that } r_j \text{ occurs in } [t, t+dt]$$

This state transition results in a change of molecules of each species :

$$\boldsymbol{v}_{ij} = \boldsymbol{s}^{\text{in}}_{i,j} + \boldsymbol{s}^{\text{out}}_{i,j},$$

or, equivalently,

$$\boldsymbol{z} - \boldsymbol{v}_j \xrightarrow{a_j(\boldsymbol{z} - \boldsymbol{v}_j)} \boldsymbol{z}, \qquad [19]$$

and

$$\boldsymbol{z} \xrightarrow{a_j(\boldsymbol{z})} \boldsymbol{z} + \boldsymbol{v}_j. \qquad [20]$$

Let us define the probability that each species exists in $\boldsymbol{z}$ number of molecules at any time t :

$$P(\boldsymbol{z}, t | \boldsymbol{z}_0, t_0) \equiv \text{Prob}\{\boldsymbol{Z}(t) = \boldsymbol{z}, \text{ given } \boldsymbol{Z}(t_0) = \boldsymbol{z}_0\}.$$

The CME describes the time evolution of the probability taking into account each propensity a_j :

$$\begin{aligned} \frac{\partial P(\boldsymbol{z}, t | \boldsymbol{z}_0, t_0)}{\partial t} = {} & \\ & \sum_j \left[a_j(\boldsymbol{z} - \boldsymbol{v}_j) P(\boldsymbol{z} - \boldsymbol{v}_j, t | \boldsymbol{z}_0, t_0) - a_j(\boldsymbol{z}) P(\boldsymbol{z}, t | \boldsymbol{z}_0, t_0) \right]. \end{aligned} \qquad [21]$$

In compact notation, we will write the CME hereafter as

$$\frac{\partial P(\boldsymbol{z},t)}{\partial t} = \mathcal{A}P(\boldsymbol{z},t), \tag{[22]}$$

where operator $\mathcal{A}$ contains the propensities of each reaction :

$$\mathcal{A} = \sum_j \mathcal{A}_j \tag{[23]}$$

Note that the choice of the probability instead of chemical concentrations as essential variable of the problem eliminates the need of determining the rate constants of each reaction, and translates the problem to finding the value of the propensities.

3.2. *A method based on proper generalized decompositions*

The purposed method is constructed by assuming that the essential variable of the problem, the probability of having a particular chemical state, is given by a finite sum of separable functions (separated representation), *i.e.*,

$$P(\boldsymbol{z},t) = \sum_{j=1}^{n_F} \alpha^j F_1^j(z_1) \otimes F_2^j(z_2) \otimes \ldots \otimes F_N^j(z_N) \otimes F_t(t), \tag{[24]}$$

where, as mentioned before, the variables z_i represent the number of molecules of species i present at a given time instant. This particular choice of the form of the basis functions allows for an important reduction in the number of degrees of freedom of the problem, $n_N \times N \times n_F$ instead of $(n_N)^N$, where N is the number of dimensions of the state space and n_N the number of degrees of freedom of each one-dimensional grid established for each spatial dimension. For this to be useful, one has to assume that the probability is negligible outside some interval, and therefore substitute the infinite domain by a subdomain $[0, \ldots, m-1]^N$, m being the chosen limit number of molecules for any species in the simulation. A similar assumption is behind other methods in the literature, such as the Finite State Projection algorithm, for instance (Munsky *et al.*, 2006).

The CME is then written in a similar form, as expressed in Equation [23], by expanding the operator $\mathcal{A}$ in the form

$$\mathcal{A} = \sum_{j=1}^{n_A} \mathbb{A}_1^j \otimes \mathbb{A}_2^j \otimes \cdots \otimes \mathbb{A}_N^j \otimes \mathbb{I}, \tag{[25]}$$

where $\mathbb{A}_i$ represent the matrix form of each operator $\mathcal{A}_i$ involved in the CME and $\mathbb{I}$ represents the identity matrix, acting on the terms depending on time only. Their particular form will be seen readily.

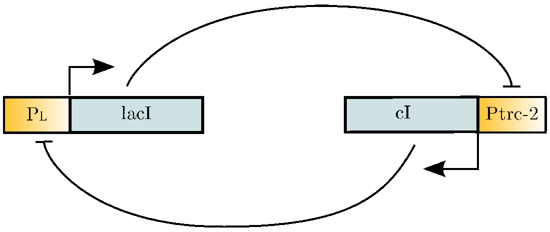

Figure 1. *Schematic mechanism of the toggle switch. The constitutive P_L promoter drives the expression of the $lacI$ gene, which produces the lac repressor tetramer. The lac repressor tetramer binds the lac operator sites adjacent to the $Ptrc - 2$ promoter, thereby blocking transcription of cI. The constitutive $Ptrc - 2$ promoter drives the expression of the cI gene, which produces the λ-repressor dimer. The λ-repressor dimer cooperatively binds to the operator sites native to the P_L promoter, which prevents transcription of $lacI$*

3.3. *Simulation of a toggle switch*

The behaviour of the λ-phage virus is one the most studied and well-known examples in gene regulatory networks. When a bacteriophage λ infects a cell, either stays dormant or it reproduces until the dead of the cell. The resulting behaviour depends crucially on two competing proteins that inhibit mutually each other, see a schematic representation in Figure 1. The so-called toggle switch is composed of a two-gene co-repressive network.

The operator form of the CME for this example is composed by two terms (Hegland *et al.*, 2007) : $\mathcal{A} = \mathcal{A}_1 + \mathcal{A}_2$, given by :

$$\begin{aligned} \mathcal{A}_1 P(z_1, z_2) = \frac{\alpha\beta}{\beta + \gamma z_2} P(z_1 - 1, z_2) + \delta(z_1 + 1) \cdot P(z_1 + 1, z_2) - \\ \left(\frac{\alpha\beta}{\beta + \gamma z_2} + \delta \cdot z_1 \right) P(z_1, z_2). \end{aligned} \qquad [26]$$

and $\mathcal{A}_2$ equivalent with z_1 and z_2 interchanged. We computed the solution for $\delta = 0.05$, $\alpha = 1.0$, $\gamma = 1.0$ and $\beta = 0.4$.

The simulation started from a non-physiological state in which both proteins showed a very high probability around $z_1 = z_2 = 15$. Despite this initial state, after $t = 100s$ (Figure 2) one has a case where both average values of both proteins and small levels of the one protein combined with higher level of the other protein are quite likely, and this remains the case for the stationary distribution as well (Hegland *et al.*, 2007), Figure 3.

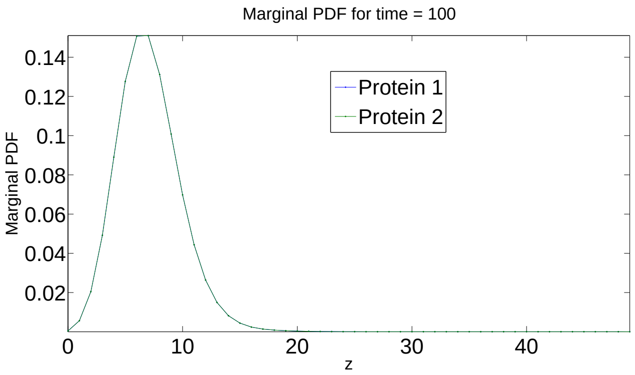

Figure 2. *Marginal probability distribution function at* $t = 100s$. *Axes denote the number of protein 1 and 2*

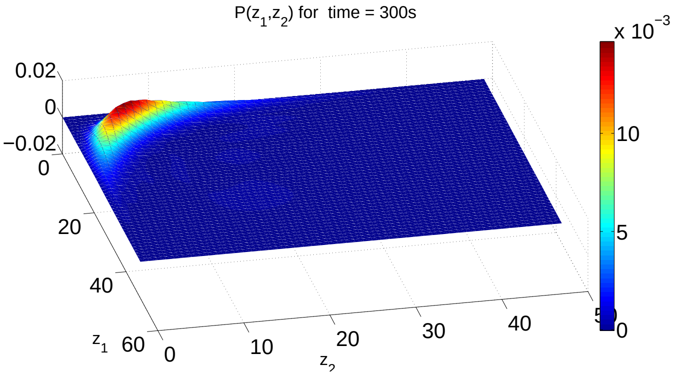

Figure 3. *Solution at steady state (*$t \approx 300s$*) by separation of variables. Axes denote the number of protein 1 (abscissa) and protein 2 (ordinate)*

4. Conclusions

We have reviewed here the essential features of the Proper Generalized Decomposition technique. This technique is particularly useful for the numerical solution of models defined in highly-dimensional spaces. As a particularly challenging example we have presented the simulation of gene regulatory networks, in particular that behaviour of the virus bacteriophage λ. This virus, although very simple, since its behaviour is governed by only two competing proteins, is very well known, and clearly shows the potential of the technique in the field of Computational Biology.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation, through grant number CICYT-DPI2008-00918, whose support is gratefuly acknowledged.

5. References

Ammar A., Mokdad B., Chinesta F., , Keunings. R., « 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.*, vol. 144, p. 98-121, 2007.

Ammar A., Mokdad B., Chinesta F., Keunings R., « 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.*, vol. 139, p. 153-176, 2006.

Bungartz H.-J., Griebel M., « Sparse grids », *Acta Numerica*, vol. 13, p. 1-123, 2004.

Cancès E., Defranceschi M., Kutzelnigg W., Bris C. L., Maday Y., « Computational Quantum Chemistry : a primer », *Handbook of Numerical Analysis*, vol. X, p. 3-270, 2003.

Chinesta F., Ammar A., Joyot P., « 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*, vol. 6, p. 191-213, 2008.

Hasty J., McMillen D., Isaacs F., Collins J. J., « Computational studies of gene regulatory networks : in numero molecular Biology », *Nature Reviews Genetics*, vol. 2, p. 268-279, 2001.

Hegland M., Burden C., Santoso L., MacNamara S., Boothm H., « A solver for the stochastic master equation applied to gene regulatory networks », *Journal of Computational and Applied Mathematics*, vol. 205, p. 708-724, 2007.

Ladeveze P., *Nonlinear Computational Structural Mechanics*, Springer, N.Y., 1999.

Laughlin R. B., Pines D., « The Theory of Everything », *Proceedings of the National Academy of Sciences*, vol. 97(1), p. 28-31, 2000.

Mokdad B., Pruliere E., Ammar A., Chinesta F., « On the Simulation of Kinetic Theory Models of Complex Fluids Using the Fokker-Planck Approach », *Applied Rheology*, vol. 17, p. 1-14, 2007.

Munsky B., Khammash M., « The finite state projection algorithm for the solution of the chemical master equation », *The Journal of Chemical Physics*, vol. 124, n˚ 4, p. 044104, 2006.

Sreenath S. N., Cho K.-H., Wellstead P., « Modelling the dynamics of signalling pathways », *Essays in biochemistry*, vol. 45, p. 1-28, 2008.