



HAL
open science

Towards a high-resolution numerical strategy based on separated representations

David González, Amine Ammar, Elías Cueto, Francisco Chinesta

► **To cite this version:**

David González, Amine Ammar, Elías Cueto, Francisco Chinesta. Towards a high-resolution numerical strategy based on separated representations. *International Journal of Material Forming*, 2008, 1, pp.1099-1102. 10.1007/s12289-008-0211-0 . hal-00749145

HAL Id: hal-00749145

<https://hal.science/hal-00749145>

Submitted on 6 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Towards a high-resolution numerical strategy based on separated representations

D. González¹, A. Ammar², E. Cueto¹, F. Chinesta³

¹Group of Structural Mechanics and Material Modelling.

Aragón Institute of Engineering Research (*I3A*).

University of Zaragoza. María de Luna, 5.

Campus Rio Ebro. E-50018 Zaragoza, Spain.

e-mail: {gonzal,ecueto}@unizar.es;

²Laboratoire de Rhéologie, UMR 5520, Université J. Fourier

1301 Rue de la piscine. Domaine universitaire,

38041 Grenoble Cedex, France.

e-mail: Amine.Ammar@ujf-grenoble.fr;

³Laboratoire de Mécanique des Systèmes et des Procédés.

UMR 8106 CNRS-ENSAM-ESEM

151 Boulevard de l'Hôpital, F-75013 Paris, France

e-mail: francisco.chinesta@paris.ensam.fr

ABSTRACT: Many models in Science and Engineering are defined in spaces (the so-called conformation spaces) of high dimensionality. In kinetic theory, for instance, the micro scale of a fluid evolves in a space whose number of dimensions is much higher than the usual physical space (two or three). Models defined in such a framework suffer from the curse of dimensionality, since the complexity of the problem grows exponentially with the number of dimensions. This curse of dimensionality makes this class of problems nearly intractable if we perform a standard discretization, say, with finite element methods, for instance. Problems defined in two or three-dimensional spaces, but densely discretized along each spatial dimension are also hardly tractable by finite element methods. In this paper we present some recent results concerning a method based on the method of separation of variables, originally developed in [1]. We focus on an efficient imposition of essential non-homogeneous boundary conditions and the treatment of problems with a very high number of degrees of freedom.

KEYWORDS: Curse of dimensionality, separation of variables, high-dimensional problems, kinetic theory

1 INTRODUCTION

In problems that involve high dimensional spaces, standard discretization techniques, like the finite element method, fail due to an excessive computation time required to perform accurate numerical simulations.

The main aim of this paper is to present a new numerical strategy able to circumvent some of the difficulties due to the multidimensional character of kinetic theory models, among other fields of science. The results will be presented to illustrate its capability for treating highly dimensional spaces by considering different multidimensional Poisson's problems, and an efficient imposition of essential non-homogeneous boundary conditions.

2 THE METHOD

For simplicity, we start by considering the Poisson problem defined in a space of dimension N with homogeneous boundary conditions,

$$-\Delta T = f(x_1, x_2, \dots, x_N) \text{ in } \Omega = (-L, L)^N. \quad (1)$$

The problem solution can be written in the form

$$T(x_1, x_2, \dots, x_N) \approx \sum_{j=1}^Q \alpha_j \prod_{k=1}^N F_{kj}(x_k), \quad (2)$$

where F_{kj} is the j th basis function, with unit norm, and only depends on the k th coordinate. Note that the solution of numerous problems can be accurately approximated using a finite number Q of approximation functions.

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

Step 1 A projection step of the solution in a discrete basis, where the coefficients α_j are computed.

Step 2 Convergence checking.

Step 3 Enrichment of the approximation basis step. From the coefficients α_j just computed the approximation basis can be enriched by adding the new function $\prod_{k=1}^N F_{k(n+1)}(x_k)$

For details of this scheme see [1].

3 IMPOSITION OF NON-HOMOGENEOUS ESSENTIAL BOUNDARY CONDITIONS

Consider, for simplicity and without loss of generality, the Poisson problem described before, see Eq. (1) subjected to the boundary conditions:

$$T = T_D \neq 0 \text{ on } \Gamma = \partial\Omega. \quad (3)$$

Let us assume that we are able to find a function ψ , continuous in $\bar{\Omega}$, such that $-\Delta\psi \in L_2(\Omega)$ verifying Eq. (3). Then, the solution of the problem given by Eqs. (1)-(3) can be obtained straightforwardly by

$$T = \psi + z, \quad (4)$$

where we thus face a problem in the z variable

$$-\Delta z = f + \Delta\psi \text{ in } \Omega \quad (5)$$

$$z = 0 \text{ on } \Gamma = \partial\Omega, \quad (6)$$

solvable by the method of separation of variables before presented.

Following this procedure, the weak form of the problem will be: *Find $z \in H^1(\Omega)$ such that for every $z^* \in H_0^1(\Omega)$*

$$\int_{\Omega} \nabla z \nabla z^* d\Omega = \int_{\Omega} z^* f d\Omega - \int_{\Omega} \nabla \psi \nabla z^* d\Omega \quad (7)$$

holds.

Several choices for the function ψ exist. In this work we employ R -functions [4], which constitute an efficient means of constructing ψ functions.

The obtained function ψ , which is generally not product or sum separable, is separated approximately by employing the Singular Value Decomposition (SVD) technique. To this end, we employ singular values up to a given precision, which is normally set to 10^{-5} .

4 NUMERICAL EXAMPLES

In this section we describe some simple numerical experiments that show the potential of the method.

4.1 Example: 2D problem

Let us consider now the problem

$$\Delta T = 0 \text{ in } [-1, 1]^2, \quad (8)$$

subjected to the boundary conditions

$$T = x^2 + y^2 \text{ on } \partial\Omega,$$

that in this case do not verify the partial differential equation (8).

In this case, using the before mentioned theory of R -functions, we obtain the following function verifying essential boundary conditions:

$$\psi = \frac{-2 + x^4 + y^4}{-2 + x^2 + y^2} \quad (9)$$

which is represented in Fig. 1. This function is again not sum or product separable, according to [3]. Applying SVD, we arrive to the one-dimensional functions depicted in Fig. 2, that approximate the function ψ to the chosen level of precision, 10^{-5} in this case.

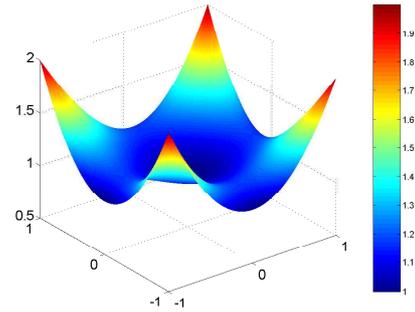


Figure 1: Function ψ for the second problem, Eq. (8).

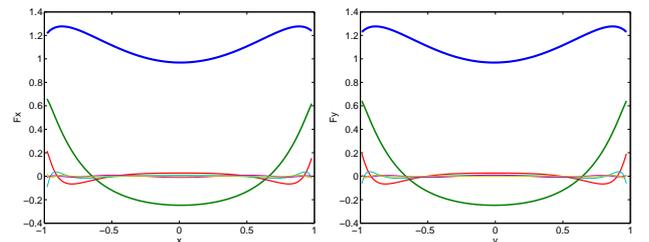


Figure 2: Singular Value Decomposition of ψ up to the precision 10^{-5} .

The method before described was applied in meshes composed by 15, 20, 30, 50, 80 and 130 nodes along each spatial direction. The solution was compared to one approximate solution obtained by standard Finite Elements in a mesh composed by 40000 nodes, thus taken as exact. The order of convergence is remarkably high, and has been found to be of the order of 4 (i.e., superconvergent) for all the cases tested.

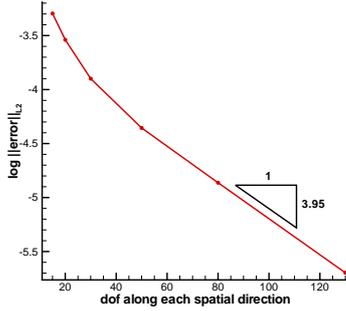


Figure 3: Solution convergence obtained by the proposed technique.

4.2 Extension of the proposed technique to higher-dimensional problems

The problem with the straightforward extension of the proposed technique to higher dimensional problems lies in obtaining the appropriate counterpart of the SVD decomposition in high dimensions.

To this end, we have employed the PARAFAC (Parallel Factors) decomposition, also known as CANDECOMP or Canonical Decomposition, provided by the MATLAB tensor toolbox [2]. This technique provides a diagonal tensor Σ , suitable for the purpose of this work. The r most relevant entries of this tensor are employed in the decomposition, in order to accomplish with a user-predefined tolerance in the representation of the function ψ . In general, we seek for decompositions of our function ψ , whose nodal values are stored in an order n ($n = 3$ in the example here considered, for simplicity of graphical representation) tensor \mathbf{A} , in the form:

$$\mathbf{A} = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W} \times_4 \dots \quad (10)$$

where Σ is a diagonal tensor and \mathbf{U} , \mathbf{V} , \mathbf{W} , ... are the components or factors of the decomposition. Columns of the factors are orthonormal in standard, two-dimensional SVD, but this is not possible, in general, in tensors of order three or higher. The products

involved in the previous decomposition are defined as follows:

$$\mathbf{B} = \mathbf{C} \times_1 \mathbf{U} \quad (11)$$

is such that

$$(\mathbf{B})_{ijk} = \sum_{l=1}^m \mathbf{C}_{ijk} \mathbf{U}_{li} \quad (12)$$

To analyze the behavior of the technique, we have extended the previous problem to three-dimensional settings:

$$\Delta T = 0 \text{ in } [-1, 1]^3, \quad (13)$$

subjected to the boundary conditions

$$T = x^2 + y^2 + z^2 \text{ on } \partial\Omega.$$

The higher-order SVD decomposition of the function ψ up to the tolerance 10^{-6} provides the functions depicted in Fig. 4. Only four functions along each spatial direction were necessary.

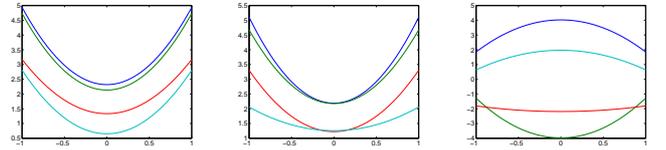


Figure 4: Functions obtained along x (a), y (b) and z (c) spatial directions for the boundary conditions of the problem stated in Eq. (13).

With these functions thus calculated, we let the method run to obtain the solution to the problem as depicted along each direction in Fig. 5. The global, three-dimensional, solution to the problem is depicted in Fig. 6. In this case, 18 functions were necessary along each spatial direction to obtain the before mentioned accuracy. The figure depicts the intersection of the solution with the plane $x = 0$.

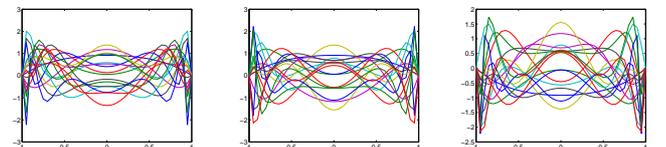


Figure 5: Functions obtained along x , y and z spatial directions for the problem stated in Eq. (13).

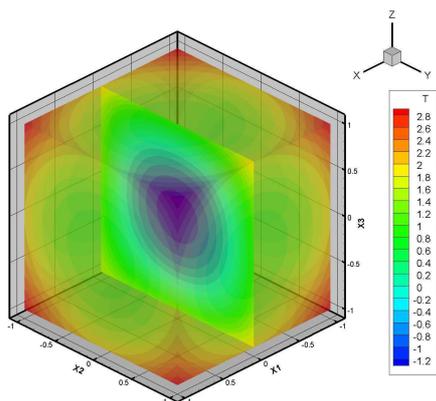


Figure 6: Solution obtained by the proposed technique for the problem with non-homogeneous boundary conditions in three dimensions. Mesh of 40 elements along each direction.

4.3 A problem with 10^9 degrees of freedom

In order to show the behavior of the proposed method we have analyzed the problem

$$\Delta T = 2(1 - y^4)(1 - z^6) + 12(1 - x^2)y^2(1 - z^6) + 30(1 - x^2)(1 - y^4)z^4 \text{ in } [-1, 1]^3, \quad (14)$$

(the same analyzed in [1]), subjected to the boundary conditions

$$T = 0 \text{ on } \partial\Omega,$$

that has been discretized in a regular lattice of 1000^3 elements, thus giving 1001^3 degrees of freedom. This value is on the limit of current capabilities of nowadays computers, and is analyzed here only for the purpose of showing the performance of the method. The analytical solution to this problem is given by the function $T = (1 - x^2)(1 - y^4)(1 - z^6)$.

The functions obtained by the method in each spatial direction are depicted in Fig. 7. As can be noticed, the method obtains a very accurate solution in only one iteration, as was the case in [1], employing fewer degrees of freedom.

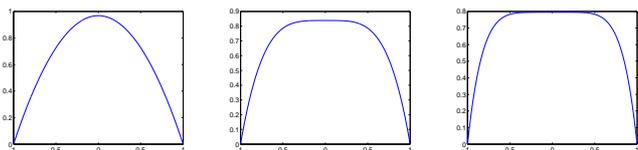


Figure 7: Functions obtained along x , y and z spatial directions for the problem stated in Eq. (14).

The solution obtained is represented in Fig. 8(top), where a downsampling has been applied to the results by employing a coarser mesh. For obvious reasons, it is difficult to depict the results on a mesh of 10^9 nodes. The error obtained is depicted in

Fig. 8(bottom), where a slice on an arbitrary section of the domain has been performed for clarity.

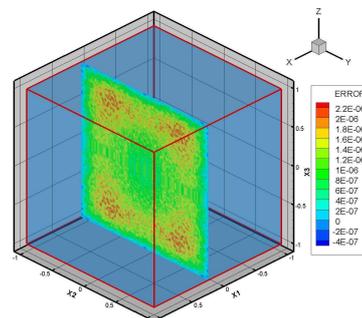
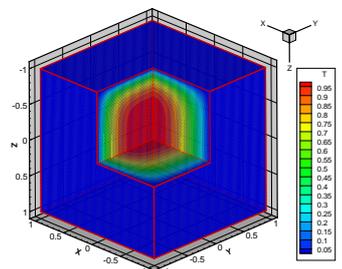


Figure 8: Solution of the problem stated in Eq. (7) and a slice on the error map.

In this example we do not pursue accuracy, but testing the possibility of employing a huge number of degrees of freedom. What is noticeable about this solution, easily reachable with traditional Finite Elements, is that, despite the extremely high number of degrees of freedom involved, the Matlab code ran in around 1 hour on a PC with four processors (only one was employed, no parallel calculation was allowed) with 1.96 Gb of RAM memory each, running under Linux. This time comprised data file writing, generation of figures, etc.

REFERENCES

- [1] 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.
- [2] Brett W. Bader and Tamara G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, July 2007. Accepted.
- [3] M. Messiter and Y. Shamash. Product and sum separable functions. *IEEE Transactions on Automatic Control*, (7):694–697, 1985.
- [4] V. L. Rvachev and T. I. Sheiko. R-functions in boundary value problems in mechanics. *Applied Mechanics reviews*, 48:151–188, 1995.