

## A GREEDY APPROACH TO MULTISCALE METHODS

Antonio Falcó<sup>1\*</sup>, Amine Ammar<sup>2</sup>, Francisco Chinesta<sup>3</sup>

<sup>2</sup> Laboratoire de Rhéologie, UJF-INPG-CNRS (UMR 5520), 1301 Rue de la Piscine, BP 53  
Domaine Universitaire, F-38041 Grenoble cedex 9, France

<sup>3</sup> Institut de Recherche en Génie Civil et Mécanique (GeM), Université de Nantes

**ABSTRACT:** In this paper we study the problem of compute the solution of a linear system in a separable representation form. It arises in the discretized equations appearing in various physical domains, such as kinetic theory, statistical mechanics, quantum mechanics, and in nanoscience and nanotechnology among others. In particular, we use the fact that tensors of order 3 or higher have best rank-1 approximation. This fact allow to us to propose an iterative method based in the so-called by the signal processing community as the Matching Pursuit Algorithm, also known as Projection Pursuit by the statistics community or as a Pure Greedy Algorithm in the approximation theory community. We also give some numerical examples and describe its relationship with the Finite Element Method for High-Dimensional Partial Differential Equations based on the tensorial product of one-dimensional bases. We illustrate this situation taking as a model problem the multidimensional Poisson equation with homogeneous Dirichlet boundary condition.

**KEYWORDS:** Separable representation, Greedy Algorithm, Multiscale methods

### 1 INTRODUCTION

In [1], some of the authors of the present paper propose the use of a separated representation, which allows to define a tensor product approximation basis as well as to decouple the numerical integration of a high dimensional model in each dimension. The purpose of this work is to formalize and analyze the above strategy in the framework of the iterative methods for linear systems. As we will show the approximation given in [1], is closely related with the best low-rank approximation problem for high order tensors. Unfortunately, it has been proved that tensors of order 3 or higher can fail to have best rank- $r$  approximation for  $r \geq 2$ . It is due to the fact that, in our knowledge, only for a given closed and nonempty set contained in  $\mathbb{R}^N$  it is possible to define a (perhaps) multivalued projection of a point into this set. We remark that the unicity for this map can be obtained by adding a convexity condition. Our strategy is to use the fact that tensors of order 3 or higher have best rank-1 approximation. Then we propose an iterative method based in the so-called by the signal processing community as the *Matching Pursuit Algorithm* of Mallat and Zhang, also known as *Projection Pursuit* by the statistics community or as a *Pure Greedy Algorithm* in the Approximation Theory community. This strategy depends strongly on the computation of the best rank-1 approximation of the residual obtained at each step of the proposed algorithm. To solve this we will propose the use of a Coordinate Descend Method Algorithm because it has global convergence. In particular, we will show that

for the class of separable invertible matrices, this problem collapses for each selected direction to an ordinary least-squares problem. Finally, we introduce the relationship between the Finite Element Method, where the shape functions are the tensorial product of a one-dimensional ones, and the class of linear systems considered in this paper. We illustrate this situation taking as a model problem the multidimensional Poisson equation with homogeneous Dirichlet boundary condition.

Before to end this section we describe some of the notation used in this paper. We denote the set of  $N \times M$ -matrices by  $\mathbb{R}^{N \times M}$ , and the transpose of a matrix  $A$  is denoted  $A^T$ . By  $\langle \mathbf{x}, \mathbf{y} \rangle$  we denote the usual Euclidean inner product given by  $\mathbf{x}^T \mathbf{y} = \mathbf{y}^T \mathbf{x}$  and its corresponding 2-norm,  $\|x\|_2 = \langle \mathbf{x}, \mathbf{x} \rangle^{1/2}$ . The matrix  $I_N$  is the  $N \times N$ -identity matrix and when the dimension is clear from the context, we simply denote it by  $I$ . Given a sequence  $\{\mathbf{u}_j\}_{j=0}^{\infty} \subset \mathbb{R}^N$ , we say that a vector  $\mathbf{u} \in \mathbb{R}^N$  can be written as

$$\mathbf{u} = \sum_{j=0}^{\infty} \mathbf{u}_j$$

if and only if

$$\lim_{n \rightarrow \infty} \sum_{j=0}^n \mathbf{u}_j = \mathbf{u}$$

holds in the  $\|\cdot\|_2$ -topology. Now, we recall the definition and some properties of the Kronecker product. The Kronecker product of  $A \in \mathbb{R}^{N_1 \times N_1}$  and  $B \in \mathbb{R}^{N_2 \times N_2}$ , written  $A \otimes_K B$ , is the tensor algebraic operation defined

\*Departamento de Ciencias Físicas, Matemáticas y de la Computación, Universidad CEU Cardenal Herrera, San Bartolomé 55, 46115 Alfara del Patriarca (Valencia), Spain. afalco@uch.ceu.es

as

$$\begin{bmatrix} A_{1,1}B & A_{1,2}B & \cdots & A_{1,N_2N'_2}B \\ A_{2,1}B & A_{2,2}B & \cdots & A_{2,N_2N'_2}B \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_1N'_1,1}B & A_{N_1N'_1,2}B & \cdots & A_{N_1N'_1,N_2N'_2}B \end{bmatrix}.$$

This paper is organized as follows. In the next section we give a separated representation algorithm for a class of linear systems. It is motivated by the main result of this paper that gives a solution to the problem of constructing an approximate solution of rank- $n$  for a linear system. Moreover, we provide some numerical examples. In 3 we give as a model problem the multidimensional Poisson equation with homogeneous Dirichlet boundary conditions. We conclude with some comments and remarks.

## 2 DEFINITIONS AND STATEMENT OF MAIN RESULT

The concept of separated representation was introduced by Beylkin and Mohlenkamp and it is related with the problem of constructing the approximate solutions of some classes of problems in high-dimensional spaces by means a separable function.

Suppose that for given a linear Partial Differential Equation, and after a discretization by means Finite Elements, we need to solve the linear system

$$A\mathbf{u} = \mathbf{f}, \tag{1}$$

where  $A$  is a  $(N_1 \cdots N_d) \times (N_1 \cdots N_d)$ -dimensional invertible matrix, for some  $N_1, N_2, \dots, N_d \in \mathbb{N}$ . Then from all said above, it seems reasonable to find an approximate solution

$$A^{-1}\mathbf{f} \approx \mathbf{u}_n = \sum_{j=1}^n \mathbf{x}_1^j \otimes_K \cdots \otimes_K \mathbf{x}_d^j$$

for some  $n \geq 1$  and where  $\mathbf{x}_i^j \in \mathbb{R}^{N_i}$  for  $i = 1, 2, \dots, d$  and  $j = 1, 2, \dots, n$ ; satisfying that

$$\lim_{n \rightarrow \infty} \|A^{-1}\mathbf{f} - \mathbf{u}_n\|_2 = 0,$$

that is,

$$A^{-1}\mathbf{f} = \sum_{j=1}^{\infty} \mathbf{x}_1^j \otimes_K \cdots \otimes_K \mathbf{x}_d^j.$$

For each  $n \in \mathbb{N}$ , we define the set

$$\mathcal{S}_n = \{\mathbf{x} \in \mathbb{R}^{N_1 \cdots N_d} : \text{rank}_{\otimes_K} \mathbf{x} \leq n\},$$

in the following way. Given  $\mathbf{x} \in \mathbb{R}^{N_1 \cdots N_d}$  we say that  $\mathbf{x} \in \mathcal{S}_1 = \mathcal{S}_1(N_1, N_2, \dots, N_d)$  if  $\mathbf{x} = \mathbf{x}_1 \otimes_K \mathbf{x}_2 \otimes_K \cdots \otimes_K \mathbf{x}_d$ , where  $\mathbf{x}_i \in \mathbb{R}^{N_i}$ , for  $i = 1, \dots, d$ . For  $n \geq 2$  we define inductively  $\mathcal{S}_n = \mathcal{S}_n(N_1, N_2, \dots, N_d) = \mathcal{S}_{n-1} + \mathcal{S}_1$ , that is,

$$\mathcal{S}_n = \left\{ \mathbf{x} : \mathbf{x} = \sum_{i=1}^k \mathbf{x}^{(i)}, \mathbf{x}^{(i)} \in \mathcal{S}_1 \text{ for } 1 \leq i \leq k \leq n \right\}.$$

Note that  $\mathcal{S}_n \subset \mathcal{S}_{n+1}$  for all  $n \geq 1$ .

It is possible to show that given an invertible matrix  $A \in \mathbb{R}^{N_1 N_2 \cdots N_d \times N_1 N_2 \cdots N_d}$ , then for each fixed  $\mathbf{b} \in \mathbb{R}^{N_1 \cdots N_d}$  we obtain that

$$\text{argmin}_{\mathbf{x} \in \mathcal{S}_1} \|\mathbf{b} - A\mathbf{x}\|_2 \neq \emptyset. \tag{2}$$

This allow to consider the following iterative scheme. Let  $\mathbf{u}_0 = \mathbf{y}_0 = \mathbf{0}$ , and for each  $n \geq 1$  we take

$$\mathbf{r}_{n-1} = \mathbf{f} - A\mathbf{u}_{n-1}, \tag{3}$$

$$\left. \begin{aligned} \mathbf{u}_n &= \mathbf{u}_{n-1} + \mathbf{y}_n \\ \text{where } \mathbf{y}_n &\in \text{argmin}_{\mathbf{y} \in \mathcal{S}_1} \|\mathbf{r}_{n-1} - A\mathbf{y}\|_2. \end{aligned} \right\} \tag{4}$$

Note that  $\mathbf{u}_n \in \mathcal{S}_n$ . Define

$$k(\mathbf{f}, A) = \begin{cases} \infty & \text{if } \{j \geq 1 : \mathbf{y}_j = \mathbf{0}\} = \emptyset, \\ \min\{j \geq 1 : \mathbf{y}_j = \mathbf{0}\} - 1 & \text{otherwise.} \end{cases}$$

The following theorem which provides a constructive approach to represent the solutions of a linear system in a separated form.

**Theorem 1** *Let  $\mathbf{f} \in \mathbb{R}^{N_1 N_2 \cdots N_d}$  and  $A \in \mathbb{R}^{N_1 N_2 \cdots N_d \times N_1 N_2 \cdots N_d}$ , be an invertible matrix. Then, by using the iterative scheme (3)-(4), we obtain that the sequence  $\{\|\mathbf{r}_n\|_2\}_{n=0}^{k(\mathbf{f}, A)}$ , is strictly decreasing and*

$$A^{-1}\mathbf{f} = \lim_{n \rightarrow \infty} \mathbf{u}_n = \sum_{j=0}^{k(\mathbf{f}, A)} \mathbf{y}_j. \tag{5}$$

Moreover, the rate of convergence if given by

$$\frac{\|\mathbf{r}_n\|_2}{\|\mathbf{r}_0\|_2} = \prod_{j=1}^n (1 - \rho_j^2)^{1/2} \tag{6}$$

for  $1 \leq n \leq k(\mathbf{f}, A)$  where

$$\rho_j = \frac{\langle \mathbf{r}_{j-1}, A\mathbf{y}_j \rangle}{\|\mathbf{r}_{j-1}\|_2 \|A\mathbf{y}_j\|_2} \in (0, 1)$$

for  $1 \leq j \leq n$ .

From (5) we obtain that if  $k(\mathbf{f}, A) < \infty$ , then  $\|\mathbf{r}_n\|_2 = 0$  for all  $n > k(\mathbf{f}, A)$ . Thus, the above theorem allow to us to construct a procedure, that we give in the pseudocode form in Algorithm 1, under the assumption that we have a numerical method in order to find a  $\mathbf{y}$  solving (2) (see the step 5 in Algorithm 1) and that we introduce below.

It can be seen that for each fixed  $\mathbf{b} \in \mathbb{R}^{N_1 \cdots N_d}$ , the map  $\Phi(\mathbf{x}) = \|\mathbf{b} - A\mathbf{x}\|_2$  defined over a convex set  $U \subset \mathbb{R}^{N_1 \cdots N_d}$  is a convex function. Since  $\mathcal{S}_1$  is not a convex set we will consider for each  $\alpha \in \{1, 2, \dots, d\}$  and

$$\mathbf{x}_1^0, \dots, \mathbf{x}_{\alpha-1}^0, \mathbf{x}_{\alpha+1}^0, \dots, \mathbf{x}_d^0$$

fixed, the set

$$\begin{aligned} \mathcal{S}_1^{(\alpha)} &= \mathcal{S}_1^{(\alpha)}(\mathbf{x}_1^0, \dots, \mathbf{x}_{\alpha-1}^0, \mathbf{x}_{\alpha+1}^0, \dots, \mathbf{x}_d^0) \\ &= \{\mathbf{x} \in \mathcal{S}_1 : \mathbf{x}_i = \mathbf{x}_i^0, i \neq \alpha\}. \end{aligned}$$

**Algorithm 1** A Separated Representation Algorithm

```

1: procedure ( $\|f - Au\|_2 < \varepsilon : \text{rank}_{\otimes} u \leq$ 
   rank_max)
2:    $r_0 = f$ 
3:    $u = 0$ 
4:   for  $i = 0, 1, 2, \dots, \text{rank\_max}$  do
5:      $y = \text{procedure} (\min_{\text{rank}_{\otimes} x \leq 1} \|r_i - Ay\|_2^2)$ 
6:      $r_{i+1} = r_i - Ay$ 
7:      $u \leftarrow u + y$ 
8:     if  $\|r_{i+1}\|_2 < \varepsilon$  or  $\|\|r_{i+1}\|_2 - \|r_i\|_2\| < \text{tol}$ 
       then goto 13
9:     end if
10:    end for
11:    return  $u$  and  $\|r_{\text{rank\_max}}\|_2$ .
12:    break
13:    return  $u$  and  $\|r_{i+1}\|_2$ 
14: end procedure

```

Since  $\mathcal{S}_1^{(\alpha)} \subset \mathcal{S}_1 \subset \mathbb{R}^{N_1 \cdots N_d}$  is a closed and convex set, this fact allow to us to solve

$$\min_{\mathbf{x} \in \mathcal{S}_1} \|\mathbf{b} - A(\mathbf{x}_1 \otimes_K \cdots \otimes_K \mathbf{x}_d)\|_2, \tag{7}$$

by means a *Cyclic Coordinate Descend* Algorithm. It minimizes  $\Phi$  cyclically with respect to the coordinate variables and it can shown that it has global convergence. These cyclic methods have the advantage of not requiring any information about the gradient  $\nabla_{\mathbf{x}} \Phi$  to determine the descent directions. However, their convergence properties are poorer than steepest descend methods. These coordinate descent methods are attractive because of their easy implementation in some particular cases as we will see below. In particular we minimize  $\Phi$  cyclically with respect to the coordinate variables. We point out that for high-dimensional problems the numerical implementation of solving that equation can be a hardly task. However, if the matrix  $A$  can be represented also in separated representation form, then it can be reduced to a standard least squares problem. Thus, we can solve

$$\min_{(\mathbf{x}_1, \dots, \mathbf{x}_d)} \left\| \mathbf{b} - \sum_{i=1}^{r_A} A_1^i \mathbf{x}_1 \otimes_K \cdots \otimes_K A_d^i \mathbf{x}_d \right\|_2. \tag{8}$$

easily, by means a Cyclic Coordinate Descend Algorithm. Note that given a point  $(\mathbf{x}_1, \dots, \mathbf{x}_d)$ , descend with respect to the coordinate  $\mathbf{x}_\alpha$  means under this conditions that

$$(Z_\alpha^T Z_\alpha)^{-1} Z_\alpha^T \mathbf{b} \in \arg \min_{\mathbf{x}_\alpha} \Phi(\mathbf{x}_1, \dots, \mathbf{x}_d). \tag{9}$$

The pseudocode of the procedure to solve (8) can be seen in Algorithm 2.

**3 A MODEL PROBLEM: THE POISSON EQUATION IN  $(0, 1)^D$**

Firstly, we consider the following problem in 3D: Solve for

$$(x_1, x_2, x_3) \in \Omega = (0, 1)^3 :$$

**Algorithm 2** Solving the minimization problem (8)

```

1: procedure ( $\min_{\mathbf{x} \in \mathcal{S}_1} \|\mathbf{b} - A\mathbf{x}\|_2^2$ )
2:   Initialize  $\mathbf{x}_i^0$  for  $i = 1, 2, \dots, d$ .
3:   iter = 1
4:   while iter < iter_max do
5:      $\hat{\mathbf{x}}_\alpha \leftarrow \mathbf{x}_\alpha^0, \alpha = 1, \dots, d$ 
6:     for  $\alpha = 1, 2, \dots, d$  do
7:        $Z_\alpha = \sum_{j=1}^{r_A} Z_\alpha^j$ 
8:        $Z_\alpha^j = A_1^{(j)} \hat{\mathbf{x}}_1 \otimes_K \cdots \otimes_K A_{\alpha-1}^{(j)} \hat{\mathbf{x}}_{\alpha-1} \otimes_K A_\alpha^{(j)} \otimes_K$ 
          $A_{\alpha+1}^{(j)} \mathbf{x}_{\alpha+1}^0 \cdots \otimes_K A_d^{(j)} \mathbf{x}_d^0$ 
9:        $\hat{\mathbf{x}}_\alpha = (Z_\alpha^T Z_\alpha)^{-1} Z_\alpha^T \mathbf{b}$ 
10:    end for
11:    if  $\prod_{\alpha=1}^d \|\mathbf{x}_\alpha^0 - \hat{\mathbf{x}}_\alpha\|_2 < \text{tol}$  then goto 14
12:    end if
13:    iter = iter + 1
14:  end while
15:  return  $\mathbf{x}^0 = (\mathbf{x}_1^0, \dots, \mathbf{x}_d^0)$ 
16: end procedure

```

$$\begin{aligned}
 -\Delta u &= (2\pi)^2 \cdot 3 \prod_{i=1}^3 \sin(2\pi x_i - \pi), \\
 u|_{\partial\Omega} &= 0,
 \end{aligned}$$

which has as closed form solution

$$u(x_1, x_2, x_3) = \prod_{i=1}^3 \sin(2\pi x_i - \pi).$$

We used the separable representation algorithm given in Section 2 with parameter values  $\text{iter\_max} = 5$ ,  $\text{rank\_max} = 1000$  and  $\varepsilon = 0.001$ . The algorithm give us an approximated solution  $\mathbf{u}_1 \in \mathcal{S}_1$ . In Figure 1 we represent the relative error of the solution computed using the separable representation algorithm, using logarithmic scale, as a function of the number of nodes used in the discretization of the Poisson equation. All the computations were performed using the GNU software OCTAVE in a AMD 64 Athlon K8 with 2Gib of RAM.

In Figure 2 we represent the CPU time, in logarithmic scale, used in solving the linear system against the separable representation algorithm. In both cases all the linear systems involved were solved using the standard linear system solver (A\b) of OCTAVE.

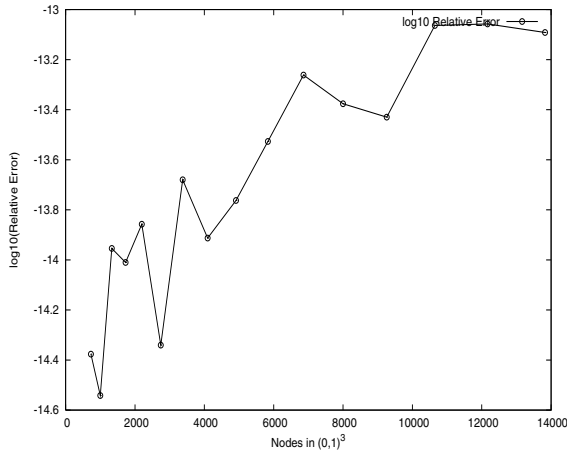
Finally we are addressing some highly multidimensional models. To this end we solve numerically the Poisson equation for  $(x_1, \dots, x_d) \in \Omega = (0, \pi)^d$  where

$$f = \sum_{k=1}^d -(1+k) \sin^{(-1+k)}(x_k) \times$$

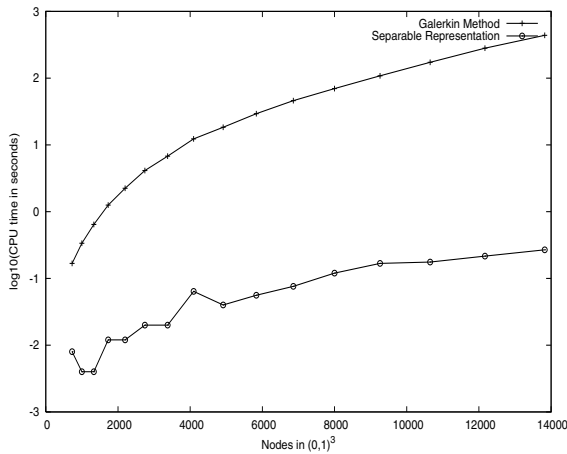
$$(-k \cos^2(x_k) + \sin^2(x_k)) \prod_{k'=1, k' \neq k}^d \sin^{(1+k')}(x_{k'}),$$

which has as closed form solution

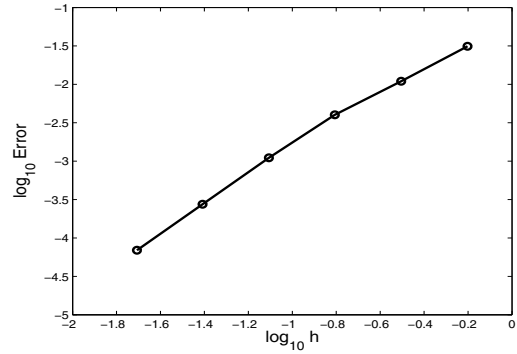
$$u(x_1, \dots, x_d) = \prod_{k=1}^d \sin^{(k+1)}(x_k).$$



**Figure 1:** The relative error  $\|\mathbf{u}_1 - A^{-1}\mathbf{f}\|_2 / \|A^{-1}\mathbf{f}\|_2$  in logarithmic scale.



**Figure 2:** The CPU time, in seconds, used in solving the linear system as a function of the number of nodes employed in the discretization of the Poisson Equation.



**Figure 3:** The absolute error  $\|\hat{\mathbf{u}} - \mathbf{u}\|_2$  as a function of  $h = \pi/N$  for  $N = 5, 10, 20, \dots, 160$  in  $\log_{10}$ -scale.

Here we consider the true solution  $\mathbf{u}$  given by  $U_{i_1, \dots, i_d} = u(\hat{x}_{i_1+1}, \dots, \hat{x}_{i_d+1})$ . For  $d = 10$  we use the parameter values  $\text{iter\_max} = 2$ ,  $\text{rank\_max} = 10$  and  $\varepsilon = 0.001$ . In a similar way as above the algorithm give us an approximated solution  $\hat{\mathbf{u}} \in \mathcal{S}_1$ . In Figure 3 we represent the absolute error  $\|\hat{\mathbf{u}} - \mathbf{u}\|_2$  as a function of  $h = \pi/N$  for  $N = 5, 10, 20, \dots, 160$  in  $\log_{10}$ -scale. By using similar parameters values the problem has been solved for  $d = 100$  in about 20 minutes.

#### 4 CONCLUDING REMARKS

In this paper we analyze, at analytical level, the iterative method proposed in [1] in order to compute the numerical solutions of high dimensional PDE's. As we can show the method runs under very weak conditions, recall that we only use the condition that the linear system has a decomposable and invertible matrix. However, its efficiency depends strongly on the matrix form (symmetric, tridiagonal, full, sparse, ...).

#### 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 modelling Complex Fluids*, Journal of Non-Newtonian Fluid Mechanics, **139**, (2006), pp. 153-176.