



MOX–Report No. 15/2010

## DOMAIN DECOMPOSITION (DD) METHODS

ALFIO QUARTERONI, LUCA FORMAGGIA

MOX, Dipartimento di Matematica “F. Brioschi”  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

[mox@mate.polimi.it](mailto:mox@mate.polimi.it)

<http://mox.polimi.it>



# DOMAIN DECOMPOSITION (DD) METHODS

Alfio Quarteroni and Luca Formaggia

MOX– Modellistica e Calcolo Scientifico  
Dipartimento di Matematica “F. Brioschi”  
Politecnico di Milano  
via Bonardi 9, 20133 Milano, Italy  
email@mate.polimi.it

**Keywords:** Domain Decomposition Methods, Computational Geology

**AMS Subject Classification:** 65M55, 65N55

## Abstract

This short paper gives a review of domain decomposition methods for parallel computations of large scale problems, with a focus on computational geology.

## Definitions

*Computational domain.* A bounded spatial region where a partial differential equation is solved.

*Dirichlet boundary condition.* A condition imposed directly on the unknown of the differential problem, also called essential boundary condition.

*Neumann boundary condition.* A condition imposed on the fluxes associated to the differential problem at hand, i.e. stresses, heat flux, etc. Also called natural boundary condition.

*Parallel computer.* A computer with more than one processing unit capable of computing operations concurrently.

*Parallel computation.* A numerical procedure executed on a parallel computer.

*Preconditioner.* In the context of the solution of a linear system by an iterative method, a preconditioner is an easily invertible matrix spectrally similar to the matrix governing the problem.

## 1 Introduction

Domain Decomposition (DD) method is a technique for the solution of partial differential equations which can be instrumental to the development of *parallel computations*. It can be used in the framework of discretization methods, e.g finite elements, finite volumes, finite differences, or spectral element methods.

It is based on the reformulation of the given boundary-value problem on a partition of the computational domain  $\Omega$  into  $M$  subdomains  $\Omega_i$ , with  $i = 1, \dots, M$ . Typically  $M$  is also the number of processors at disposal, even if it is possible to have more subdomains per processor. The DD method also provides a convenient framework for the solution of heterogeneous or multi-physics problems, i.e. those that are governed by differential equations of different kinds in different subregions of the computational domain. In this case the subdomains conform with the subregions. In this work, however, we will address only homogeneous DD methods, yet some of the concepts presented here may be readily adapted for the non-homogeneous case.

We will refer to a partial differential equation of the form

$$-\operatorname{div}(\mathbf{T}(\mathbf{u})) = \mathbf{f} \quad \text{in } \Omega \quad (1)$$

with suitable boundary conditions on  $\partial\Omega$ .

For instance, in a problem of elastostatic,  $\mathbf{u}$  is the displacement and  $\mathbf{T}(\mathbf{u})$  the stress tensor, in a heat conduction problem  $\mathbf{u}$  is a scalar variable, the temperature, and  $\mathbf{T}$  the Laplace operator. With a rather similar formalism we may also account for the Stokes problem. The method is in fact applicable also to time dependent problems, since in most cases the time advancing scheme required by the numerical simulation leads eventually to a sequence of problems of type (1). For instance if we consider the evolution equation

$$\partial_t \mathbf{u} - \operatorname{div}(\mathbf{T}(\mathbf{u})) = \mathbf{f},$$

a time discretization by the implicit Euler method leads to solve for the unknown  $\mathbf{u}^n$  at each time step  $t^n$  a differential problem of the type

$$\mathbf{u}^n - \delta t \operatorname{div}(\mathbf{T}(\mathbf{u}^n)) = \delta t \mathbf{f}^n + \mathbf{u}^{n-1}.$$

It is still of the form (1) with  $\mathbf{T}$  replaced by  $\delta t \mathbf{T} - \mathbf{I}$ ,  $\delta t$  being the time step and  $\mathbf{I}$  the identity operator. A similar form is obtained also for the elastodynamic equations, discretised in time, for instance, by a Newmark method (Quarteroni and Valli, 1994).

Typically, and for evident practical reasons, the partition into subdomains is made after having triangulated  $\Omega$  into a finite element mesh  $\tau_h$ , each subdomain being in fact formed by a set of elements of the original grid (see figure 1). The partition is often made automatically, using libraries like METIS (Karypis and Kumar, 1998b) or PARMETIS (Karypis and Kumar, 1998a), the latter able to exploit a parallel architecture also for this pre-processing stage.

More in particular, there are two ways of subdividing the computational domain into subdomains. The first uses disjoint subdomains, where  $\Omega_i \cap \Omega_j = \emptyset$  for  $i \neq j$ . Here, the interface between subdomains reduces to surfaces (in 3D), and lines (in 2D). The second strategy adopts *overlapping subdomains*, usually built by letting the subdomains of an initial non-overlapping partition grow of a certain factor  $\delta$ . In most of the cases  $\delta$  is dictated by the number of “layers” of grid elements that are added to the original partition. The minimal overlap is of one element, like that shown in figure 1. For a more complete review of DD methods one may refer to (Smith, Bjorstad, and Gropp, 1996; Quarteroni and Valli, 1999; Wohlmuth, 2001; Toselli and Widlund, 2005; Mathew, 2008); several examples with emphasis on parallel computations are reported, for instance, in (Bruaset and Tveito, 2006).

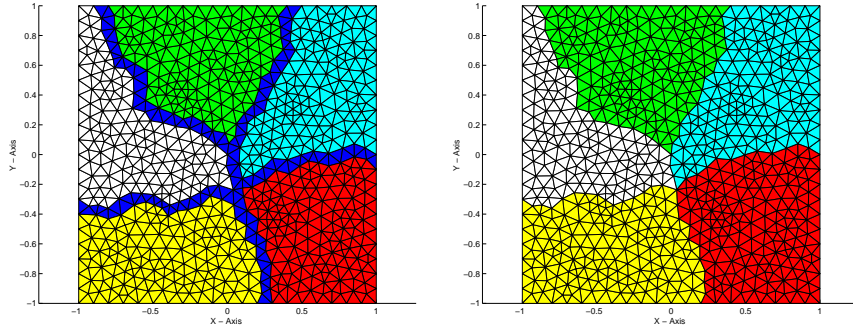


Figure 1: Partition of a computational domain starting from a given triangulation. Left: Partition of a domain with disjoint subdomains. Right: A partition into subdomains with an overlap equal to one layers of elements.

## 2 Algorithms without overlap

We indicate with  $\Gamma_{ij}$  the interface between subdomains  $\Omega_i$  and  $\Omega_j$  and we set  $\Gamma = \cup_{ij} \Gamma_{ij}$ . We exploit the fact that Problem (1) is equivalent to solving the following  $M$  coupled problems for  $i = 1, \dots, M$ ,

$$-\operatorname{div}(\mathbf{T}(\mathbf{u}_i)) = \mathbf{f}_i \quad \text{in } \Omega_i \quad (2)$$

with the same boundary conditions of the original problem applied to  $\partial\Omega_i \cap \partial\Omega$ , while on each  $\Gamma_{ij}$  we set the continuity of the local solutions  $\mathbf{u}_i$  and of the fluxes, that is

$$\mathbf{u}_i = \mathbf{u}_j, \quad \mathbf{T}(\mathbf{u}_i) \cdot \mathbf{n}_{ij} + \mathbf{T}(\mathbf{u}_j) \cdot \mathbf{n}_{ji} = \mathbf{0}, \quad (3)$$

where  $\mathbf{n}_{ij}$  is the normal to  $\Gamma_{ij}$  outward oriented w.r.t.  $\Omega_i$ . It may be proved that  $\mathbf{u}_i = \mathbf{u}_j|_{\Omega_i}$  (Quarteroni and Valli, 1999).

Problems (2) are coupled because of relations (3). A parallel algorithm may be obtained by an iterative procedure where conditions (3) are enforced in such a way to generate at each iteration decoupled problems that can be run on different processes, with only a small amount of communication needed at the beginning of each iteration. For the sake of space we give an example of just one of these procedures, called *Dirichlet-Neumann* for the case of two subdomains  $\Omega_1$  and  $\Omega_2$ . Starting from a guess for  $\mathbf{u}_1^{(0)}$  and  $\mathbf{u}_2^{(0)}$ , the algorithm solves for  $k = 1, 2, \dots$  the following sequence of independent problems,

$$\begin{cases} -\operatorname{div}(\mathbf{T}(\mathbf{u}_1^{(k+1)})) = \mathbf{f}_1, & \text{in } \Omega_1, \\ \mathbf{u}_1^{(k+1)} = \theta \mathbf{u}_2^{(k)} + (1 - \theta) \mathbf{u}_1^{(k)}, & \text{on } \Gamma_{12} \end{cases} \quad \text{and} \quad \begin{cases} -\operatorname{div}(\mathbf{T}(\mathbf{u}_2^{(k+1)})) = \mathbf{f}_2, & \text{in } \Omega_2 \\ \mathbf{T}(\mathbf{u}_2^{(k+1)}) \cdot \mathbf{n}_{21} = \mathbf{T}(\mathbf{u}_1^{(k)}) \cdot \mathbf{n}_{21} & \text{on } \Gamma_{12} \end{cases}$$

until a measure of the difference  $\mathbf{u}_1^{(k+1)} - \mathbf{u}_2^{(k+1)}$  on  $\Gamma_{12}$  is below a given tolerance;  $\theta$  here is a convenient relaxation factor. The Dirichlet-Neumann technique is not easily extendable to an arbitrary number of subdomains (unless suitable *coloring techniques* are used) and its convergence characteristics strongly depend on the geometry of the subdomains as well as on the possible jump of characteristic coefficients (for instance the viscosity of different rocks forming a sedimentary basin). Other more favourable techniques, like the Neumann-Neumann, Robin-Robin and FETI methods, are described in (Quarteroni and Valli, 1999; Wohlmuth, 2001; Toselli and Widlund, 2005).

The DD method may be also set from an algebraic view point. Indeed when discretized, for instance by a finite element method, (2) and (3) reduce to a system of linear equations which may be written in a block form as

$$\begin{pmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f}_I \\ \mathbf{f}_\Gamma \end{pmatrix},$$

where  $A_{II}$  is a *block diagonal* matrix with  $M$  blocks of dimension equal to the number of unknowns internal to each subdomain, the latter being collected in  $\mathbf{u}_I$ , while  $\mathbf{u}_\Gamma$  is the vector of unknowns on the interface  $\Gamma$ . If  $A_{II}$  is invertible (and normally this is the case) we may obtain a problem for the  $\mathbf{u}_\Gamma$  only (*Schur complement system*),  $\Sigma_\Gamma \mathbf{u}_\Gamma = \boldsymbol{\chi}_\Gamma$ , where  $\Sigma_\Gamma = A_{\Gamma\Gamma} - A_{\Gamma I} A_{II}^{-1} A_{I\Gamma}$  is the so called *Schur complement matrix* w.r.t.  $A_{II}$ . Having solved for  $\mathbf{u}_\Gamma$  the computation of  $\mathbf{u}_I$  can be done in a perfect parallel fashion by solving the block diagonal problem  $A_{II} \mathbf{u}_I = \mathbf{f}_I - A_{I\Gamma} \mathbf{u}_\Gamma$ .

A DD scheme with no overlap may be interpreted as *preconditioned iterative scheme* (Quarteroni and Valli, 1994; Quarteroni, 2009) for the Schur complement system, where the preconditioner can be efficiently applied in a parallel setting. A crucial issue for parallel computing is that of *scalability*. In the DD setting an algorithm is said to be scalable if its convergence properties do not depend on the number of subdomains  $M$ , and in particular does not degrade if we keep the ratio  $M/N$  between number of subdomains and the total number  $N$  of unknowns of our problem fixed. Indeed in this case (if we neglect communication overheads) we may solve in the same time a problem twice as large by doubling the number of processors. A scalable parallel preconditioner cannot be built using only local (i.e. at the subdomain level) approximations of the Schur matrix, we need to add also a *coarse operator* which has the role of transferring information among far away subdomains. The typical form of the preconditioner is (we write directly the inverse operator since it is the one actually required by the iterative procedure(Quarteroni, Sacco, and Saleri, 2007))

$$P_\Sigma^{-1} = \sum_{i=1}^M R_i^T \Sigma_i^* R_i + R_0^T \Sigma_0^* R_0, \quad (4)$$

where  $R_i$  is a *restriction operator* that selects from all the  $\mathbf{u}_\Gamma$  those local to the  $i$ -th subdomain, and  $\Sigma_i^*$  is a local approximation of the inverse Schur matrix, which typically can be built using just data related to the  $i$ -th subdomain (and thus in parallel). Finally,  $\Sigma_0^*$  is the coarse operator, of small dimension (typically of the order of  $M$ ), whose role is to guarantee the coupling among all subdomains and it is necessary for scalability. The application of the preconditioner (that is the computation of  $P_\Sigma^{-1} \mathbf{x}$ , where  $\mathbf{x}$  is any vector of the right length), can be done in parallel a part from the coarse operator, which however, being small in size, is irrelevant in the computational cost balance. Several scalable preconditioners are available, see (Toselli and Widlund, 2005) and (Canuto, Hussaini, Quarteroni, and Zang, 2007).

### 3 Methods with overlap

A typical DD scheme with overlapping subdomains is the Schwarz method. In its basic form it is an iterative algorithm for the solution of (1) that for  $k = 1, 2, \dots$  solves a series of local

problems on each subdomain  $\Omega_i$ , where on the non-empty interfaces  $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$  we apply Dirichlet boundary conditions using the latest values available from  $\Omega_j$ , that is

$$\begin{cases} -\operatorname{div}(\mathbf{T}(\mathbf{u}_i^{(k+1)})) = \mathbf{f}_i, & \text{in } \Omega_i, \\ \mathbf{u}_i^{(k+1)} = \mathbf{u}_j^{(k)}, & \text{on } \Gamma_{ij}. \end{cases}$$

The iteration continues until the difference between two successive iteration is sufficiently small. The convergence analysis of the Schwarz method may be found in (Smith, Bjorstad, and Gropp, 1996; Quarteroni and Valli, 1999; Toselli and Widlund, 2005), and in (Canuto, Hussaini, Quarteroni, and Zang, 2007) in the context of spectral element discretizations.

The method is seldom used in this form, however. Yet, it is probably the most adopted method to build parallel preconditioners for an iterative solver of the global problem, giving rise the so-called *Krylov-Schwarz* (for linear problems) and *Newton-Krylov-Schwarz* methods. A Schwarz parallel preconditioner  $P_S$  may be written similarly to (4), that is

$$P_S^{-1} = \sum_{i=1}^M R_i^T A_i^{-1} R_i + R_0^T A_0^{-1} R_0,$$

where here the restriction operator  $R_i$  extracts from a vector of length  $N$  the elements corresponding to the unknowns internal to the extended subdomain  $\Omega_i$ ,  $A_i = R_i A R_i^T$  is the local matrix extracted from the matrix  $A$  of our problem, and finally  $A_0$  is again a global coarse operator of size of the order  $M$  needed for scalability,  $R_0$  being the corresponding restriction matrix.

Again, a part from the coarse operator, the computation of  $P_S^{-1}\mathbf{x}$  can be done in parallel since the matrices  $A_i$  are local, and can be handle by each processor.

## 4 Application to geophysical and geological problems

DD methods have been applied successfully in the context of geophysical problems, like acoustic and elastodynamic wave propagation (Faccioli, Maggio, Paolucci, and Quarteroni, 1997), or full-waveform tomography (Soubier, Operto, Virieux, Amestoy, and L'Excellent, 2009).

It has been also applied to speed up the simulation of the evolution of sedimentary basins. Here different type of sediments such as gravel, sand, rocks and biological remains that have been transported by the wind, the rivers and, sometimes, by the sea, accumulate are buried deeper and deeper and transform to rocks by a complex processes of compaction and diagenesis.

On geological scales, the evolution of a sedimentary basin may be tackled by a fluid approach where each rock layer is modelled as a viscous fluid. This is particularly convenient in the presence of salt tectonics (Massimi, Quarteroni, Saleri, and Scrofani, 2007).

The computational domain  $\Omega$  is split in several subdomains  $\Omega_i$  which usually correspond to the rock layers, as shown in figure 2. At each time step we have to solve in the domain  $\Omega$  a Stokes problem, possibly with non-Newtonian rheology, to compute the instantaneous velocity field,

$$\begin{cases} -\operatorname{div}(\mathbf{T}(\mathbf{v})) + \nabla p = \mathbf{f}, \\ \operatorname{div}(\mathbf{v}) = \phi, \\ \frac{\partial \rho}{\partial t} + (\mathbf{v} \nabla) \rho = 0. \end{cases} \quad (5)$$

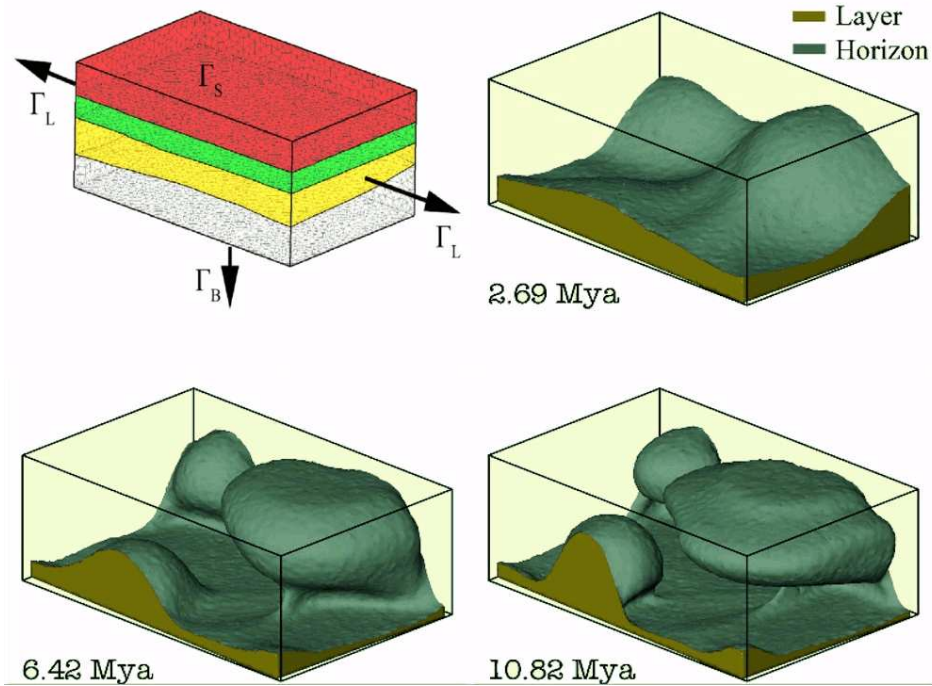


Figure 2: The simulated evolution of a salt diapir. In the first graphic the computational domain showing the subdivision of the layers. The other pictures show snapshots of the evolution of the salt layer with the formation of a diapir.

Here,  $\mathbf{v}$  is the velocity,  $p$  the pressure,  $\mathbf{f}$  the external gravitational field and  $\rho$  the density. The stress tensor  $\mathbf{T}$  depends on characteristics of the rocks, such as viscosity and density, which may be discontinuous across layers. The function  $\phi$  may account for compaction processes, or may be simply set to zero when the hypothesis of isochoric flow is acceptable. The last equation describes the evolution of the density, which is simply advected by the velocity field. The presence of faults is accounted for by appropriately reducing the rock viscosity in the vicinity of the faults.

The movement of the layers has been tracked using a level set technique. A finite element scheme has been adopted for the discretization of the Stokes problem, while a conservative finite volume scheme has been used for the tracking of the layer interfaces. The parallel implementation has been carried out with the help of the TRILINOS library (Heroux and et al., 2005), using a Schwarz algorithm with a coarse operator built by aggregation (Sala, 2004).

Figure 2 shows the evolution of a salt dome. Salt is less compressible than the surrounding rock, so during the sedimentation process it ends having a smaller density than the overburden. We are then facing a Rayleigh-Taylor instability and any small perturbation will cause the salt to rise, which large movements that causes the so-called salt diapirism.



## References

- Bruaset A, Tveito A (eds), 2006. Numerical Solution of Partial Differential Equations on Parallel Computers, Lecture Notes in Computational Science and Engineering, vol 51. Springer, New York
- Canuto C, Hussaini MY, Quarteroni A, Zang TA, 2007. Spectral Methods. Evolution to Complex Geometries and Application to Fluid Dynamics. Springer-Verlag, Berlin Heidelberg
- Faccioli E, Maggio F, Paolucci R, Quarteroni A, 1997. 2D and 3D elastic wave propagation by a pseudo-spectral domain decomposition method. Journal of seismology 1(3):237–251
- Heroux M et al, 2005. An overview of the TRILINOS project. ACM Transactions on Mathematical Software (TOMS) 31(3):397–423
- Karypis G, Kumar V, 1998.a) A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing 20:359–392
- Karypis G, Kumar V, 1998.b) METIS: Unstructured graph partitioning and sparse matrix ordering system. Tech. Rep. 98-036, University of Minnesota, Department of Computer Science
- Massimi P, Quarteroni A, Saleri F, Scrofani G, 2007. Modeling of salt tectonics. Comput Methods Appl Mech Engrg 197(1-4):281–293
- Mathew T, 2008. Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations. Lecture Notes in Computational Science and Engineering, Springer, New York
- Quarteroni A, 2009. Numerical Models for Differential Problems, MS&A, vol 2. Springer, Berlin
- Quarteroni A, Valli A, 1994. Numerical Approximation of Partial Differential Equations. Springer, Berlin Heidelberg
- Quarteroni A, Valli A, 1999. Domain Decomposition Methods for Partial Differential Equations. Oxford Science Publications, Oxford
- Quarteroni A, Sacco R, Saleri F, 2007. Numerical Mathematics, II edn. Springer, Berlin and Heidelberg
- Sala M, 2004. Analysis of two-level domain decomposition preconditioners based on aggregation. Mathematical Modelling and Numerical Analysis 38(5):765–780
- Smith B, Bjorstad P, Gropp W, 1996. Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, New York
- Soubrier F, Operto S, Virieux J, Amestoy P, L'Excellent JY, 2009. FWT2D: A massively parallel program for frequency-domain full-waveform tomography of wide-aperture seismic data—part 2: Numerical examples and scalability analysis. Computers & Geosciences 35(3):496 – 514

Toselli A, Widlund O, 2005. Domain Decomposition Methods - Algorithms and Theory, Springer Series in Computational Mathematics, vol 34. Springer, New York

Wohlmuth B, 2001. Discretization methods and iterative solvers based on domain decomposition. Lecture Notes in Computational Science and Engineering, Springer, Berlin

**Submitted as contributed Chapter to:** *Encyclopedia of Solid Earth Geophysics, Springer.*

## MOX Technical Reports, last issues

Dipartimento di Matematica “F. Brioschi”,  
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 15/2010** ALFIO QUARTERONI, LUCA FORMAGGIA:  
*Domain Decomposition (DD) methods*
- 14/2010** PAOLA F. ANTONIETTI, LOURENCO BEIRÃO DA VEIGA,  
MARCO VERANI:  
*A Mimetic Discretization of Elliptic Obstacle Problems*
- 13/2010** G.M. PORTA, SIMONA PEROTTO, F. BALLIO:  
*A Space-Time Adaptation Scheme for Unsteady Shallow Water Problems*
- 12/2010** RICCARDO SACCO, PAOLA CAUSIN, PAOLO ZUNINO,  
MANUELA T. RAIMONDI:  
*A multiphysics/multiscale numerical simulation of scaffold-based cartilage regeneration under interstitial perfusion in a bioreactor*
- 11/2010** PAOLO BISCARI, SARA MINISINI, DARIO PIEROTTI,  
GIANMARIA VERZINI, PAOLO ZUNINO:  
*Controlled release with finite dissolution rate*
- 10/2010** ALFIO QUARTERONI, RICARDO RUIZ BAIER:  
*Analysis of a finite volume element method for the Stokes problem*
- 09/2010** LAURA M. SANGALLI, PIERCESARE SECCHI, SIMONE VANTINI,  
VALERIA VITELLI:  
*Joint Clustering and Alignment of Functional Data: an Application to Vascular Geometries*
- 08/2010** FRANCESCA IEVA, ANNA MARIA PAGANONI:  
*Multilevel models for clinical registers concerning STEMI patients in a complex urban reality: a statistical analysis of MOM<sup>2</sup> survey*
- 07/2010** LAURA M. SANGALLI, PIERCESARE SECCHI, SIMONE VANTINI,  
VALERIA VITELLI:  
*Functional clustering and alignment methods with applications*
- 06/2010** JORDI ALASTRUEY, TIZIANO PASSERINI, LUCA FORMAGGIA,  
JOAQUIM PEIRÓ:  
*The effect of visco-elasticity and other physical properties on aortic and cerebral pulse waveforms: an analytical and numerical study*