ALGORITHMS FOR A NON CONFORMING DOMAIN DECOMPOSITION METHOD

Y. Achdou [1] & Yu.A. Kuznetsov [2]

Abstract

The Q_1 mortar element method is a non conforming domain decomposition method based on Q_1 finite elements. For the Laplace equation, it yields an ill conditioned linear system. A preconditioner is presented, and two different iterative methods using this preconditioner are implemented and tested numerically.

CONTENT

0 Introduction

This work on preconditioners has been motivated by the development of a solver in 2 dimensions for the Navier Stokes equations in the stream function-vorticity formulation (see [2]) using a non-conforming nonoverlapping domain decomposition method based on Q_1 finite elements, namely the *mortar* element method. This method was first introduced by C.Bernardi, Y.Maday, A.Patera in [6].
A time iteration of the Navier Stokes solver is split into a convection and a diffusion steps. The convection is treated by the characteristic Galerkin method (see [2] and references therein). The diffusion step is the resolution of a fourth order elliptic problem. In fact, this fourth order problem

[1] CMAP,Ecole Polytechnique: achdou@cmapx.polytechnique.fr
[2] Institute of Numerical Mathematics, Russian Academy of Sciences, 32a Leninskij Prospect, Moscow 117334

can be replaced by four second order problems ([8],[2]) two of them involving the Laplace operator Δ and the two other the Helmholtz operator $-\nu\delta t\Delta + I$, where ν is the viscosity and δt is the time step.
The mortar element method will be briefly described in section 1. This method has the advantage to allow nonmatching nonoverlapping grids at the interfaces beetween subdomains and therefore permits to use very simple structured meshes in each subdomain. It leads to linear systems of the following form :

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ \Lambda \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

where A is a block diagonal matrix (one block per subdomain) and B is a jump operator.
In section 1, we shall present numerical evidence that the linear system for the Laplace operator is rather ill conditioned, in any case much worse conditioned than the Helmholtz operator for small values of ν. The aim of this paper is to present a simple preconditioner for the discrete Stecklov-Poincaré, which has been first used in [4] in another context, and to use it with two different iterative solvers. This will be done in section 3.
In section 2, an incomplete Cholesky preconditioner for $BA^{-1}B^T$ is described. It will be used in a preconditioned Uzawa algorithm. The other methods presented in this article will be compared to it. Note that this preconditioner is not really practical since it rather expensive to assemble.
The scope of section 3 is to present a simple preconditioner for the discrete Steklov Poincaré operator. This preconditioner is used in two iterative methods :

- A preconditioned Uzawa method.
- A preconditioned conjugate gradient for a Schur complement method.

The two algorithms are presented along with a method to solve rapidly the linear system involving the preconditioner.
Finally, numerical results are presented in section 4. Estimates on condition numbers will be derived in a forthcoming work ([3]).
In the following, with each finite element function f_h, we associate the vector F of its coordinates in the cannonical basis. Accordingly, with each operator L_h, we associate the matrix L .

NOTATIONS

Ω : domain of $\mathbb{R}^2$, divided into K non overlapping subdomains.
Ω_k : subdomain of Ω.
$\Gamma_{k,l} = \bar{\Omega}_k \cap \bar{\Omega}_l$.
$\Gamma = \cup\Gamma_{k,l}$.
C set of the crosspoints of the domain decomposition.
$\mathcal{T}_{k,h}$: quadrangulation of Ω_k.
X_{kh} : Q_1 finite element space associated to Ω_k and $\mathcal{T}_{k,h}$.
$X_h = \prod_{1\le k\le K} X_{kh}$.
Tr_k : trace operator from Ω_k to $\partial\Omega_k$.
$Z_{kh} = Tr_k X_{kh}$.
$Tr = \bigotimes_{1\le k\le K} Tr_k$.
$Z_h = \prod_{1\le k\le K} Z_{kh} = Tr\, X_h$.
W_h : space for the Lagrange multiplier in the mortar element method.
B_h : $X_h \rightarrow W_h'$: jump operator.

$b_h \ : \ X_h \times W_h \ \rightarrow \mathbb{R}$: bilinear form associated with the jump operator : $b_h(u_h, \mu_h) =< B_h u_h, \mu_h >$
$\tilde{B}_h \ : \ Z_h \rightarrow W_h' \ : \ B_h = \tilde{B}_h \circ Tr$
$\tilde{B}_h^T \ : \ W_h \rightarrow Z_h' \ : \ B_h^T = Tr^T \circ \tilde{B}_h^T$
With each finite element function f_h, we associate the vector F of its coordinates in the cannonical basis. Accordingly, with each operator L_h, we associate the matrix L .

1. Recalls .

In paragraph 1.1, we shall present the algorithm of the Navier Stokes solver, based on splitting each iteration in time into a convection step and a diffusion step. We shall see that the diffusion step requires solving second order elliptic problems with two different kinds of operators, one of which is the Laplace operator. In paragraph 1.2, we shall recall the domain decomposition method used for the elliptic problems solver, *i.e.* the mortar element method . In paragraph 1.3, we shall present the basic solver for the linear systems and numerical evidence that preconditioning is highly suitable for the Laplace operator.

1.1 An algorithm for the 2D Navier Stokes equations .

We consider the incompressible 2D Navier-Stokes equations with stream function and vorticity:

$$\partial_t \omega + u.\nabla\omega - \nu\Delta\omega = 0, \qquad -\Delta\psi = \omega \quad \text{in } \Omega\times]0,T[,$$
$$\psi, \quad \frac{\partial\psi}{\partial n} \text{ given on } \Gamma_0\times]0,T[, \quad \omega_{|t=0} \text{ given in } \Omega.$$

The Characteristic-Galerkin method ([2] and references therein) for time discretization is

$$\frac{1}{\delta t}\left(\omega^{m+1} - \omega^m o X^m\right) - \nu\Delta\omega^{m+1} = 0 \quad - \Delta\psi^{m+1} = \omega^{m+1} \quad \text{in } \Omega,$$
$$\psi,^{m+1} \quad \frac{\partial\psi}{\partial n}^{m+1} \text{ given on } \Gamma_0,$$

where $X^m(x)$ is the solution at time t^m of the ordinary differential equation (intuitively $X^m(x) \approx x - u^m(x)\delta t)$:

$$\frac{dX}{d\tau} = u(X(\tau)) \text{ with end condition } X(t^{m+1}) = x.$$

At each time step one must compute $f = \delta t \ \omega^m o X^m \approx \delta t \ \omega(x - u^m(x)\delta t)$, which is a convection process, and solve the Generalized Stokes problem with $\mu = \nu\delta t$

$$\omega - \mu\Delta\omega = f, \quad -\Delta\psi = \omega \quad \text{in } \Omega, \frac{\partial\psi}{\partial n}|_{\Gamma_0} = g_1, \quad \psi|_{\Gamma_0} = g_2. \tag{1.1}$$

This problem is a fourth order elliptic problem in ψ. It could be discretized in a straightforward way, by using high order finite elements for ψ. Here, we have chosen to discretize both ψ and ω by low order (namely Q_1) finite elements.

The basis of the algorithm dates back to [8] and lies in the following simple remark : The generalized Stokes problem (1.1) would be easy if the trace of the vorticity was known at the boundary, because it would then decouple into two second order problems

$$\begin{aligned} &\omega - \mu\Delta\omega = f, \quad \omega|_{\Gamma_0} = \xi, \\ &-\Delta\psi = \omega \quad \psi|_{\Gamma_0} = g_2. \end{aligned} \tag{1.2}$$

If we denote by $\omega(\xi), \psi(\xi)$ the solution of this problem then equation

$$\frac{\partial\psi}{\partial n}(\xi)|_{\Gamma_0} = g_1,$$

is a linear equation for ξ.

For a Navier Stokes flow around a bounded obstacle, this leads to the following algorithm :
At each time step, we first shift the problem by solving :

$$\begin{aligned} &\omega_0^{n+1} - \nu\delta t\Delta\omega_0^{n+1} = \omega^n(X^n) \quad \text{in } \Omega, \\ &\omega_0^{n+1} = 0 \quad \text{on } \Gamma_0, \end{aligned} \tag{1.3}$$

and

$$\begin{aligned} &-\Delta\psi_0^{n+1} = \omega_0^{n+1}, \quad \text{in } \Omega, \\ &\psi_0^{n+1} = 0 \quad \text{on } \Gamma_0, \end{aligned} \tag{1.4}$$

This is in some sense a prediction step, because ω^{n+1} is certainly not 0 at the boundary. To correct this prediction, one should solve the following fourth order homogeneous elliptic problem in the whole exterior domain

$$\begin{aligned} &\omega_1^{n+1} - \nu\delta t\Delta\omega_1^{n+1} = 0, \\ &-\Delta\psi_1^{n+1} = \omega_1^{n+1}, \\ &\psi_1^{n+1} = 0 \quad , \quad \frac{\partial\psi_1^{n+1}}{\partial n} = -\frac{\partial\psi_0^{n+1}}{\partial n} \quad \text{on } \Gamma_0, \end{aligned} \tag{1.5}$$

We take advantage of the spatial homogeneity of (1.5) to avoid solving it directly : since the problem is a linear spatially homogeneous elliptic problem, its solutions can be totally represented by convolution products on Γ_0 involving two potentials. Therefore, once these potentials are known , the trace $\omega^{n+1}|_{\Gamma_0} = \omega_1^{n+1}|_{\Gamma_0}$ can be readily obtained. In [1],[2], we show that these two potentials are the solutions of a well posed variational integral equation which we do not write here. To summarize, we compute $\omega_1^{n+1}|_{\Gamma_0}$ by solving first a boundary integral equation and then using a representation formula .

At this point, we know the exact trace of ω^{n+1}. Then $(\omega^{n+1}, \psi^{n+1})$ are computed by solving

$$\begin{aligned} &\omega^{n+1} - \nu\delta t\Delta\omega^{n+1} = \omega^n(X^n) \quad \text{in } \Omega, \\ &\omega^{n+1} = \omega_1|_{\Gamma_0} \quad \text{on } \Gamma_0, \end{aligned} \tag{1.6}$$

and

$$\begin{aligned} &-\Delta\psi^{n+1} = \omega^{n+1} \quad \text{in } \Omega, \\ &\psi^{n+1} = 0 \quad \text{on } \Gamma_0, \end{aligned} \tag{1.7}$$

Thus at each time step, the algorithm requires solving two second order elliptic problems involving the Helmholtz operator $I - \nu\delta t\Delta$ and two problems with the Laplace operator.

1.2. Recalls on the mortar method

The methodology of the mortar method is the following :

- The domain is first decomposed into non overlapping subdomains.
- The continuous domain decomposition problem is written in a variational form .
- A Galerkin method is used to discretize the problem : for each subdomain, there is a discrete space (possibly of finite elements) and at each intersection of two subdomains, a " mortar function " is introduced to enforce matching conditions.

Notice that in general, such a methodology leads to nonconforming approximations, because the meshes need not match at subdomains'interfaces. We shall see that our implementation is never conforming.

1.2.a The geometry.

Let Ω be a bounded polygonal domain of $\mathbb{R}^2$, and let Ω_k be a partition of Ω into K non-overlapping open quadrilaterals :

$$\overline{\Omega} = \cup_{k=1}^{K} \overline{\Omega}_k \qquad \text{and} \qquad \Omega^k \cap \Omega^l = \emptyset \quad \text{if } k \neq l. \tag{1.8}$$

This restriction to polygonal domain and subdomains is made only for the sake of simplicity. In fact, in our simulations, the domain is not polygonal and neither are the subdomains, (we just require that $\forall k,\ \partial\Omega_k \cap \Omega$ is a made of straight segments).

We also suppose that the domain decomposition is geometrically conforming, which means that the intersection of two subdomains is either empty or a vertex or a whole edge. Note that this is an implementation choice and not a limitation of the mortar method. For any $1 \leq k < l \leq K$, let $\Gamma_{k,l}$ be the closed straight segment, possily degenerate : $\Gamma_{k,l} = \overline{\Omega}_k \cap \overline{\Omega}_l$.

1.2. b The discretization

Let us consider the model elliptic problem in Ω : find u in $H_0^1(\Omega)$ such that

$$\forall v \in H_0^1(\Omega), \qquad \mu \int_\Omega \nabla u \cdot \nabla v \ \mathrm{d}x + \int_\Omega u\, v \ \mathrm{d}x = \int_\Omega f\, v \ \mathrm{d}x. \tag{1.9}$$

Following [6], let us introduce the space Y :

$$Y = \left\{ \mathbf{v} = (v_k)_{1\leq k \leq K},\ v_k \in H^1(\Omega_k),\ \forall l,k,\ v_{k|_{\Gamma_{k,l}}} = v_{l|_{\Gamma_{k,l}}},\ v_{|\partial\Omega} = 0 \right\}. \tag{1.10}$$

The space Y endowed with the norm

$$||\mathbf{v}||_*^2 = \sum_{k=1}^{K} \mu \int_{\Omega_k} |\nabla v^k|^2 \ \mathrm{d}x + \int_{\Omega_k} |v^k|^2 \ \mathrm{d}x, \tag{1.11}$$

is a Hilbert space. One sees easily that problem (1.9) is equivalent to the following continuous variational problem : find $\mathbf{u} \in Y$ such that

$$\forall \mathbf{v} \in Y, \qquad \mu \sum_{k=1}^{K} \int_{\Omega_k} \nabla u_k \cdot \nabla v_k \ \mathrm{d}x + \sum_{k=1}^{K} \int_{\Omega_k} u_k\, v_k \ \mathrm{d}x = \sum_{k=1}^{K} \int_{\Omega_k} f\, v_k \ \mathrm{d}x. \tag{1.12}$$

We discretize this problem by means of a Galerkin method : to do it, with each $1 \leq k \leq K$, we associate a family of quadrangular finite element meshes $\mathcal{T}_{k,h}$ of Ω_k, with the classical regularity assumption for F.E.M., and we denote X_{kh} the related space of isoparametric Q^1 finite elements. We define by X_h the product space :

$$X_h = \prod_{1 \leq k \leq K} X_{kh}$$

Note that we do not impose any matching conditions for the grids at subdomains' interfaces. We now have to define a space of mortar functions : we denote by Tr_k the trace operator from $H^1(\Omega_k)$ onto $H^{1/2}(\partial\Omega_k)$. Let $\tilde{\tilde{W}}_{k,l,h}$ be the space

$$\tilde{\tilde{W}}_{k,l,h} = \left\{ Tr_k v_{|\Gamma_{k,l}}, \quad v \in X_{kh} \right\}, \tag{1.13}$$

of dimension $m_{k,l}$, (here, the number of vertices of $\mathcal{T}_{k,h}$ lying on $\Gamma_{k,l}$) and let $n_{k,l} = max(m_{k,l}, m_{l,k})$.

Let $\tilde{W}_{k,l,h}$ be one of the two spaces $\tilde{\tilde{W}}_{k,l,h}$, achieving the maximum $n_{k,l}$.

For clarity assume that $\tilde{W}_{k,l,h} = \tilde{\tilde{W}}_{k,l,h}$. Let $(\varphi_i), 1 \leq i \leq n_{k,l}$ be the traces of hat functions of $\tilde{\tilde{W}}_{k,l,h}$ associated to vertices of $\mathcal{T}_{k,h}$ lying on $\Gamma_{k,l}$, (φ_1 and $\varphi_{n_{k,l}}$ corresponds to the common corners of Ω_k and Ω^l). We define $W_{k,l,h}$ as the vectorial space spanned by $(\varphi_1 + \varphi_2, \varphi_3, ..., \varphi_i, ..., \varphi_{n_{k,l}-2}, \varphi_{n_{k,l}-1} + \varphi_{n_{k,l}})$. $W_{k,l,h}$ is a subspace of $\tilde{W}_{k,l,h}$ of codimension 2.

Let us call W_h the mortar function space :

$$W_h = \prod_{1 \leq k < l \leq K} W_{k,l,h}. \tag{1.14}$$

We are now able to define the discrete space Y_h :

$$Y_h = \left\{ \begin{array}{l} \mathbf{v}_h = (v_{h,k} \in X_{kh})_k, \quad v_{h,k|\partial\Omega} = 0, \\ \forall 1 \leq k < l \leq K, \forall \mu_h \in W_h, \displaystyle\int_{\Gamma_{k,l}} (Tr_k v_{h,k} - Tr_l v_{h,l})\mu_h = 0 \end{array} \right\}. \tag{1.15}$$

The discretized variational formulation is : find $\mathbf{u}_h \in Y_h$ such that

$$\forall \mathbf{v}_h \in Y_h, \quad \mu \sum_{k=1}^{K} \int_{\Omega_k} \nabla u_{h,k} \cdot \nabla v_{h,k} + \sum_{k=1}^{K} \int_{\Omega_k} u_{h,k}\, v_{h,k} = \sum_{k=1}^{K} \int_{\Omega_k} f\, v_{h,k}. \tag{1.16}$$

Clearly the bilinear form in (1.16) is coercive so the discrete problem (1.16) is well posed. This is not so clear for the Laplace operator with Dirichlet conditions. However, even in this case, one can prove by contradiction the coerciveness of the discrete problem, see([5]).

The error analysis is carried out in [5]. The optimal error estimate is found, inspite of nonconformity.

1.3. Implementation: the saddle point problem.

The aim of this section is to reformulate problem (1.16) into a saddle point problem. Let a_h be the symmetric bilinear form on $X_h \times X_h$:

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \mu \sum_{k=1}^{K} \int_{\Omega_k} \nabla u_{h,k} \cdot \nabla v_{h,k} \, \mathrm{d}x + \sum_{k=1}^{K} \int_{\Omega_k} u_{h,k}\, v_{h,k} \, \mathrm{d}x. \tag{1.17}$$

We associate to a_h the linear operator A_h such that $a_h(\mathbf{u}_h, \mathbf{v}_h) = (A_h \mathbf{u}_h, \mathbf{v}_h)$. Let b_h be the bilinear form on $X_h \times W_h$

$$b_h(\mathbf{v}_h, \mu_h) = \sum_{1 \le k < l \le K} \int_{\Gamma_{k,l}} (Tr_k v_{h,k} - Tr_l v_{h,l}) \mu_h. \tag{1.18}$$

We associate to b_h the linear operator B_h from X_h to W_h' such that $b_h(\mathbf{v}_h, \mu_h) = (B_h \mathbf{v}_h, \mu_h)$. Then the space Y_h has the new description :

$$Y_h = \{\mathbf{v}_h \in X_h \text{ such that } \quad B_h \mathbf{v}_h = 0\}. \tag{1.19}$$

Problem (1.16) admits a following saddle point formulation of the form : find $(\mathbf{u}_h, \lambda_h) \in X_h \times W_h$ such that

$$\begin{aligned} A_h \mathbf{u}_h + B_h^T \lambda_h &= f_h, \\ B_h \mathbf{u}_h &= 0. \end{aligned} \tag{1.20}$$

We supply X_h with its natural basis of hat functions and W_h with the basis introduced above. Let A and B be the matrices of the operators A_h and B_h in these basis . Let us also call U, Λ and F the vectors corresponding to $\mathbf{u}_h$, λ_h and f_h. Problem (1.20) is equivalent to the following linear system :

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ \Lambda \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix} \tag{1.21}$$

The matrix A is the block diagonal matrix

$$A = \begin{pmatrix} A_1 & & & & & \\ & \cdot & & & & \\ & & \cdot & & & \\ & & & A_k & & \\ & & & & \cdot & \\ & & & & & \cdot \\ & & & & & & A_K \end{pmatrix}$$

each block corresponding to a discrete Neumann problem in subdomain Ω_k, except if $\Omega_k \cap \partial\Omega \neq \emptyset$. In the following except in paragraph 3.2, we shall assume that for each subdomain, the Neumann problem is well posed. Hence, all the methods described below (except in paragraph 3.2) cannot be applied directly to the Laplace operator. In this case, one can either use penalization or the method proposed in [12].

Eliminating U in system (1.21) leads to the following linear system in Λ :

$$BA^{-1}B^T\Lambda = \Phi = BA^{-1}F, \tag{1.22}$$

and U is given by :

$$U = A^{-1}(F - B^T\Lambda). \tag{1.23}$$

The Uzawa algorithm is in fact the conjugate gradient method applied to system (1.22), whose matrix is positive definite. Such an algorithm allows not to construct effectively the matrices B and B^T, which saves memory.
We have tested this method without preconditioning, for both linear systems corresponding to problems (1.3) with $\nu\delta t = 0.02$ and (1.4). The mesh is described in paragraph 4.5. It leads to 600 unknowns for Λ and 4800 unknowns for U. For both linear systems, the stopping criteria was to

decrease the residual by a factor 10^{-6}. The histories of convergence are displayed on the following figure.

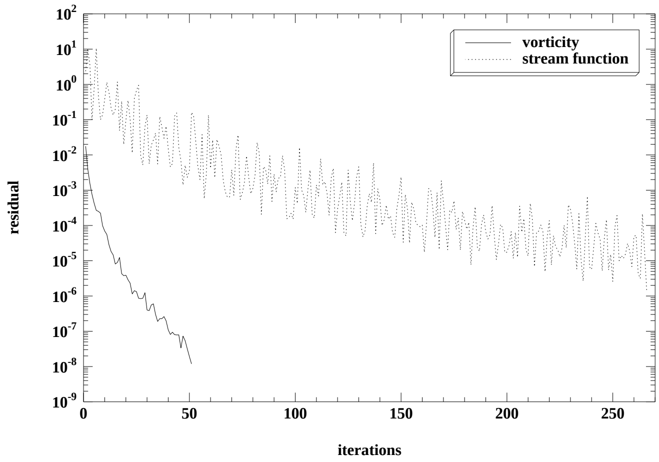

Figure 1.1 : performances of the unpreconditioned Uzawa method

As shown on this figure, this method proves satisfactory for the linear system for the vorticity (the operator is close enough to the identity) , but is clearly too slow for Laplace's equation. To make the Navier Stokes solver efficient, preconditioning is absolutely necessary, especially for problems (1.4), (1.7).

2. An incomplete Cholesky preconditioner .

In this very short section, we describe some kind of incomplete preconditioner for system (1.22). Again, we assume that the discrete local Neumann problems are well posed.

Let us introduce a few notations : assume that the inner edges of the domain decomposition are numbered from 1 to J. For all $j; 1 \le j \le J$, call W_{jh} the space of mortar functions supported by edge j, and pr_j the projector from W_h onto W_{jh}. Call Pr_j the matrix of pr_j in the cannonical basis of W_h.

Let us call M the proposed preconditioner : M is a block diagonal matrix, with J blocks, each block corresponding to an inner edge of the domain decomposition. For all $j; 1 \le j \le J$, the block M_j is given by :

$$M_j = Pr_j B A^{-1} B^T Pr_j. \tag{2.1}$$

The matrix M_j is assembled by a loop on the degrees of freedom located on edge j. For each degree of freedom, one needs to solve a discrete Neumann problem in the two neighboring subdomains to edge j. Therefore, assembling M may be really expensive for decompositions with many unknowns per edges.
It can be checked easily that M_j is a positive definite symmetric matrix. Therefore, once M is assembled, a block Cholesky factorization can easily be performed in parallel.

Remark
The preconditioner described above can be used only in the case of decompositions with many subdomains and relatively few unknowns per subdomains. In realistic cases, assembling this preconditioner would be too expensive. However, since this preconditioner is very close to the matrix

$BA^{-1}B^T$ itself, it can appear as a point of reference for the cheaper methods described in section 3.

3. Two iterative methods based on the use of a simple preconditioner .

3.1. Preconditioned Uzawa method.

For the Uzawa method described in section 1, each iteration involves the product of the matrix $BA^{-1}B^T$ by a vector Ψ, hence the resolution of a discrete Neumann problem per subdomain :

$$a_k(v_{kh}, w_{kh}) = b_h(w_{kh}, \psi_h), \qquad \forall w_{kh} \in X_{kh}. \tag{3.1}$$

Let us first introduce Tr_k the trace operator from Ω_k to $\partial\Omega_k$, and Tr the operator:

$$Tr_h = \bigotimes_{1\le k\le K} Tr_k.$$

For each subdomain Ω_k, we call Z_{kh} the trace of X_{kh} on $\partial\Omega_k$, namely

$$Z_{kh} = Tr_k X_{kh}.$$

Finally, Z_h is the product space :

$$Z_h = \prod_{1\le k\le K} Z_{kh} = Tr\, X_h.$$

The diagramm of the operator $B_h \circ A_h^{-1} \circ B_h^T = \tilde{B}_h \circ Tr \circ A_h^{-1} \circ Tr^T \circ \tilde{B}_h^T$ is the following :

$$W_h' \xrightarrow[\tilde{B}_h^T]{} Z_h' \xrightarrow[Tr^T]{} X_h' \xrightarrow[A_h^{-1}]{} X_h \xrightarrow[Tr]{} Z_h \xrightarrow[\tilde{B}_h]{} W_h$$

We would like to find a preconditioner for $BA^{-1}B^T$, which would avoid solving discrete problems in subdomains (what is done by preconditioners proposed by [7], [11]), by acting only on degrees of freedom located on interfaces. To achieve this, we need to find an operator $\tilde{A}_h \;:\; Z_h \to Z_h'$, such that

- $\tilde{A}_h^{-1}$ should approximate $Tr \circ A_h^{-1} \circ Tr^T$,
- the linear systems $\tilde{A}Y = F$ and $\tilde{B}\tilde{A}^{-1}\tilde{B}^T\Lambda = \Phi$ should be easy to solve.

The diagramm of $\tilde{B}_h\tilde{A}_h^{-1}\tilde{B}_h^T$ would then be :

$$W_h' \xrightarrow[\tilde{B}_h^T]{} Z_h' \xrightarrow[\tilde{A}_h^{-1}]{} Z_h \xrightarrow[\tilde{B}_h]{} W_h$$

Following [4], it is possible to choose $\tilde{A}$ as a block diagonal matrix, each block $\tilde{A}_k$ being a non singular matrix acting on Z_{kh}, of the form :

$$\tilde{A}_k = \begin{pmatrix} \times & \times & & & & & & \times \\ \times & \times & \times & & & & & \\ & \cdot & \cdot & \cdot & & & & \\ & & \cdot & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \\ & & & & & \cdot & \cdot & \cdot \\ & & & & & \times & \times & \times \\ \times & & & & & & \times & \times \end{pmatrix}$$

The choice of the matrix $\tilde{A}$ depends on the mesh and on the operator A. We just give an example : let subdomain Ω_k be a rectangle, and the mesh $\mathcal{T}_{k,h}$ be a rectangular finite difference grid. Assume that the nodes on $\partial\Omega_k$ are numbered from 1 to N_k. Let ϵ be a small positive parameter used to make $\tilde{A}_k$ positive definite. For simplicity, assume also that $1 < i < N_k$. The i^{th} row of $\tilde{A}_k$ is

$$(\ldots \quad -1/2 \quad 1+\epsilon \quad -1/2 \quad \ldots).$$

Here ϵ is a small positive parameter chosen in order to make $\tilde{A}_k$ positive definite.
This matrix is obtained heuristically by making the assumption that the mesh is fine enough so that v_{hk} solution of (3.1) does not vary too much when moving normally from the boundary $\partial\Omega_k$ to the first rows of vertices boarding $\partial\Omega_k$. Identifying formally the values of v_{hk} on $\partial\Omega_k$ and the values on the first row of vertices boarding $\partial\Omega_k$ yields a new linear system on $Z_{k,h}$ whose matrix is $\tilde{A}_k$. The rows of $\tilde{A}_k$ corresponding to crosspoints are chosen such that 0 is almost an eigenvalue of $\tilde{A}_k$ (because 0 is an eigenvalue of A_k). Hence, $\tilde{A}$ appears as a condensation of A on the edges of the domain decomposition.

Remark As we shall see in numerical experiments, this choice of $\tilde{A}_k$ proves good for quasi-uniform meshes.

In subsection 3.3, we shall explain how to solve rapidly the system :

$$\tilde{B}\tilde{A}^{-1}\tilde{B}^T\Lambda = \Psi. \tag{3.2}$$

We conclude the subsection by recalling the preconditioned Uzawa method used for system (1.22) :

Initialization

Choose Λ_0;

Compute $\Psi_0 = \Phi - BA^{-1}B^T\Lambda_0$. (Use a block Cholesky factorisation of A);

Solve linear system : $\tilde{B}\tilde{A}^{-1}\tilde{B}^T\Sigma_0 = \Psi_0$;

Set $\Pi_0 = \Sigma_0$;

Compute $\alpha_0 = (\Sigma_0, \Psi_0)$;

Set i=0;

Loop

while $(i < imax)$ **and** $(\frac{\alpha_i}{\alpha_0} > \epsilon)$

{

Compute $\Theta_i = BA^{-1}B^T\Pi_i$. (Use a block Cholesky factorisation of A);

Compute $\rho_i = \frac{(\Psi_i,\Pi_i)}{(\Theta_i,\Pi_i)}$;

Set $\Lambda_{i+1} = \Lambda_i + \rho_i\Pi_i$;

Set $\Psi_{i+1} = \Psi_i - \rho_i\Theta_i$;

Solve $\tilde{B}\tilde{A}^{-1}\tilde{B}^T\Sigma_{i+1} = \Psi_{i+1}$;

Compute $\alpha_{i+1} = (\Sigma_{i+1}, \Psi_{i+1})$ and $\gamma_{i+1} = \frac{\alpha_{i+1}}{\alpha_i}$;

Set $\Pi_{i+1} = \Sigma_{i+1} + \gamma_{i+1}\Pi_i$;

i=i+1;

}

3.2 Preconditioned conjugate gradient for a Schur complement method.

As opposed to the methods described in paragraphs 2 and 3.1, the method described below can be applied in a straightforward manner to Laplace operator because it is based on solving Dirichlet problems (instead of Neumann) in each subdomain.

For all k, $1 \leq k \leq K$, the finite element space X_{kh} has the following decomposition :

$$X_{kh} = \overset{\circ}{X}_{kh} \oplus Z_{kh},$$

where

$$\overset{\circ}{X}_{kh} = \{f_h \in X_{kh}\ ; \quad f_h \text{ has value } 0 \text{ on } \partial\Omega_k\}\,.$$

For each subdomain Ω_k, we renumber the unknowns as follows : We number first the degrees of freedom of functions of $\overset{\circ}{X}_{kh}$, then those corresponding to Z_{kh}. With this numbering, the matrix A_k is mapped to :

$$\begin{pmatrix} A_{k11} & A_{k12} \\ A_{k12}^T & A_{k22} \end{pmatrix}. \tag{3.3}$$

The matrix A_{k11} corresponds to a Dirichlet problem in Ω_k. Similarly, the vectors U and F in (1.21) are transformed into

$$\begin{pmatrix} U_{11} \\ U_{12} \\ . \\ . \\ U_{k1} \\ U_{k2} \\ . \\ . \\ U_{K1} \\ U_{K2} \end{pmatrix}, \quad \begin{pmatrix} F_{11} \\ F_{12} \\ . \\ . \\ F_{k1} \\ F_{k2} \\ . \\ . \\ F_{K1} \\ F_{K2} \end{pmatrix}$$

Let U_{k1}^0 be the solution of linear system $A_{k11}\ U_{k1}^0 = F_{k1}$, which corresponds to a local Dirichlet problem. Define the block diagonal matrix S, whose k^{th} block is the Schur complement

$$S_k = A_{k22} - A_{k12}^T A_{k11}^{-1} A_{k12}, \tag{3.4}$$

Then calling V and P the vectors

$$V = \begin{pmatrix} U_{12} \\ \cdot \\ \cdot \\ U_{k2} \\ \cdot \\ \cdot \\ U_{K2} \end{pmatrix}, \quad P = \begin{pmatrix} F_{12} - A_{112}^T U_{11}^0 \\ \cdot \\ \cdot \\ F_{k2} - A_{k12}^T U_{k1}^0 \\ \cdot \\ \cdot \\ F_{K2} - A_{K12}^T U_{K1}^0 \end{pmatrix},$$

one checks easily that (V, Λ) solve the linear system :

$$\begin{pmatrix} S & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} V \\ \Lambda \end{pmatrix} = \begin{pmatrix} P \\ 0 \end{pmatrix} \qquad (3.5).$$

Once this system is solved, U_{k1} is given by :

$$U_{k1} = U_{k1}^0 - A_{k11}^{-1} A_{k12} U_{k2}. \qquad (3.6)$$

Following [10], the idea is to solve system (3.5) by the preconditioned conjugate gradient method (PCG) with preconditioner

$$\begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}. \qquad (3.7)$$

The main feature of this method is the following : if the method is initialized with V_0 such that

$$\tilde{B} V_0 = 0, \qquad (3.8)$$

then, at iteration i, V_i has the same property :

$$\forall i \geq 0, \quad \tilde{B} V_i = 0. \qquad (3.9)$$

In other words, with the special choice (3.7) of preconditioner, the PCG algorithm remains in the kernel of $\tilde{B}$. The preconditioner is not only used to improve the system's condition number , but also to keep invariant the kernel of $\tilde{B}$.

It can be shown that the matrices

$$\begin{pmatrix} S & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}$$

and

$$\begin{pmatrix} S & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}^{-1} \begin{pmatrix} S & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}$$

are positive definite in the kernel of $\tilde{B}$. According to [4] the corresponding Preconditioned Conjugate method in this subspace can be applied to solve (3.5). Moreover, in a work in preparation, we shall show the following estimate on the condition number , by adapting results of [4], [9] :

$$cond \begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix}^{-1} \begin{pmatrix} S & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \leq C max(\frac{d_k}{h_k}),$$

where d_k is the diameter of Ω_k and h_k is the average mesh step size in Ω_k. Of course, this estimate holds for the choice of $\tilde{A}$ made in paragraph (3.1), and for quasi-uniform meshes.

An iteration of the proposed PCG requires solving the following linear system :

$$\begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} V \\ \Lambda \end{pmatrix} = \begin{pmatrix} Q \\ 0 \end{pmatrix} \qquad (3.10).$$

An algorithm for this system will be described in section 3.3.
We conclude the subsection by writing the preconditioned conjugate gradient algorithm for (3.5) in subspace $Ker\tilde{B}$.

Initialization

Choose Λ_0 and V_0 such that $\tilde{B}V_0 = 0$;

Compute $R_0 = F - \tilde{B}^T\Lambda_0 - SV_0$. (requires solving local discrete Dirichlet problems);

Solve linear system

$$\begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} S_0 \\ \Sigma_0 \end{pmatrix} = \begin{pmatrix} R_0 \\ 0 \end{pmatrix},$$

as explained above ;

Set $P_0 = S_0$, $\Pi_0 = \Sigma_0$;

Compute $\alpha_0 = (S_0, R_0)$;

Set i=0;

Loop

while $(i < imax)$ **and** $(\frac{\alpha_i}{\alpha_0} > \epsilon)$

{

Compute $Q_i = SP_i + \tilde{B}^T\Pi_i$. (requires solving local discrete Dirichlet problems);

Compute $\rho_i = \frac{(R_i,P_i)}{(Q_i,P_i)}$;

Set $\Lambda_{i+1} = \Lambda_i + \rho_i\Pi_i$; $V_{i+1} = V_i + \rho_i P_i$;

Set $R_{i+1} = R_i - \rho_i Q_i$;

Solve linear system

$$\begin{pmatrix} \tilde{A} & \tilde{B}^T \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} S_{i+1} \\ \Sigma_{i+1} \end{pmatrix} = \begin{pmatrix} Q_i \\ 0 \end{pmatrix};$$

Compute $\alpha_{i+1} = (S_{i+1}, R_{i+1})$ and $\gamma_{i+1} = \frac{\alpha_{i+1}}{\alpha_i}$;

Set $\Pi_{i+1} = \Sigma_{i+1} + \gamma_{i+1}\Pi_i$; $P_{i+1} = S_{i+1} + \gamma_{i+1}P_i$;

i=i+1;

}

3.3 Algorithm for solving the preconditioner linear system.

The algorithm to solve (3.2) is based on renumbering the unknowns : indeed, Z_{kh} has the following decomposition

$$Z_{kh} = \overset{\circ}{Z}_{kh} \oplus Z^c_{kh},$$

where

$$\overset{\circ}{Z}_{kh} = \{f_h \in Z_{kh}\ ;\quad f_h \text{ has value 0 on } C\},$$

where C is the set of the crosspoints of the domain decomposition, and

$$Z^c_{kh} = \{f_h \in Z_{kh}\ ;\quad f_h \text{ has value 0 at any node but crosspoints}\}.$$

We can also define

$$\overset{\circ}{Z}_h = \prod_{1\le k\le K} \overset{\circ}{Z}_{kh},$$

and

$$Z^c_h = \prod_{1\le k\le K} Z^c_{kh}.$$

Of course

$$Z_h = \overset{\circ}{Z}_h \oplus Z^c_h.$$

The renumbering is the following :
we number first the degrees of freedom of functions of $\overset{\circ}{Z}_h$, then those corresponding to Z^c_h. With this numbering, the matrices $\tilde{B}$, $\tilde{A}$ are mapped into :

$$(\tilde{B}_1 \quad \tilde{B}_2), \text{ and } \begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}^T_{12} & \tilde{A}_{22} \end{pmatrix} \tag{3.11}$$

The matrix $\tilde{A}_{11}$ is a block diagonal matrix with $4K$ blocks $\tilde{A}_{i,j,1,1}$, $1 \le i \le K$, $1 \le j \le 4$: block $\tilde{A}_{i,j,1,1}$ is a tridiagonal symmetric matrix corresponding to edge j of subdomain i: for each block $\tilde{A}_{i,j,1,1}$ a Cholesky factorization is performed. The matrix $\tilde{A}_{22}$ is a diagonal matrix.

The first step in deriving the algorithm for solving (3.2) is to notice that the solution of

$$\begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}^T_{12} & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{3.12}$$

reads

$$T_2 = (\tilde{A}_{22} - \tilde{A}^T_{12}\tilde{A}^{-1}_{11}\tilde{A}_{12})^{-1}(F_2 - \tilde{A}^T_{12}\tilde{A}^{-1}_{11}F_1) \tag{3.13}$$

$$T_1 = \tilde{A}_{11}^{-1}F_1 + \tilde{A}_{11}^{-1}\tilde{A}_{12}(\tilde{A}_{22} - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12})^{-1}\tilde{A}_{12}^T\tilde{A}_{11}^{-1}F_1 - \tilde{A}_{11}^{-1}\tilde{A}_{12}(\tilde{A}_{22} - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12})^{-1}F_2 \quad (3.14)$$

Replacing vector $\begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$ with $\begin{pmatrix} \tilde{B}_1^T\Phi \\ \tilde{B}_2^T\Phi \end{pmatrix}$, and multiplying by $(\tilde{B}_1, \tilde{B}_2)$, we obtain that with the new numbering the matrix $\tilde{B}\tilde{A}^{-1}\tilde{B}^T$ becomes :

$$\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T + (\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{B}_2)(\tilde{A}_{22} - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12})^{-1}(\tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T - \tilde{B}_2^T), \quad (3.15)$$

which can be viewed as a Schur complement of matrix :

$$\mathcal{M} = \begin{pmatrix} \tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T & \tilde{B}_1\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{B}_2 \\ \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T - \tilde{B}_2^T & -\tilde{A}_{22} + \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12} \end{pmatrix}. \quad (3.16)$$

Hence, solving (3.2) is equivalent to solving :

$$\mathcal{M}\begin{pmatrix} \Lambda \\ \Xi \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}. \quad (3.17)$$

Let Λ^0 be the solution of system :

$$\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T\Lambda^0 = F. \quad (3.18)$$

Note that the matrix $\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T$ is a block diagonal matrix, one block for each edge of the decomposition. It can be easily checked that each block is positive definite. Hence, such a linear sytem can be solved in parallel, by using for instance a block Cholesky factorization of $\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T$. Calling

$$\Lambda^1 = \Lambda - \Lambda^0,$$

we are led to considering the following system :

$$\mathcal{M}\begin{pmatrix} \Lambda^1 \\ \Xi \end{pmatrix} = \begin{pmatrix} 0 \\ (\tilde{B}_2^T - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T)\Lambda^0 \end{pmatrix}. \quad (3.19)$$

Eliminating Λ^1 yields the system :

$$\begin{aligned} &\Big(\tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{A}_{22} - (\tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T - \tilde{B}_2^T)(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T)^{-1}(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{B}_2)\Big)\,\Xi \\ &= (\tilde{B}_2^T - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T)\Lambda^0 \end{aligned}, \quad (3.20)$$

Let us call $\mathcal{C}$ the matrix in (3.20) :

$$\mathcal{C} = \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{A}_{22} - (\tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T - \tilde{B}_2^T)(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T)^{-1}(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{B}_2).$$

Notice that $\mathcal{C}$ and the matrix in (3.15) are 2 " conjugate " Schur complements of $\mathcal{M}$.
This matrix is symmetric. By using a discrete maximum principle, it is possible to show that $\tilde{A}_{22} - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{A}_{12}$ is positive definite, and therefore $-\mathcal{C}$ is also positive definite.

Note that system (3.20) is a small system, with as many unknowns as degrees of freedom located on crosspoints, whose size is less than $4\times$ the number of crosspoints. The following figure shows the locations of the degrees of freedom j such that $\mathcal{C}_{i,j} \neq 0$:

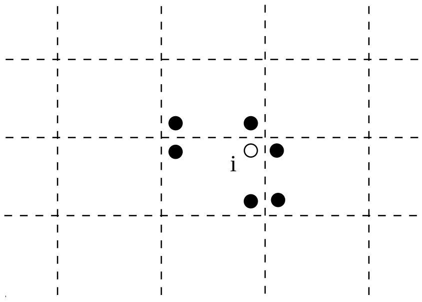

Figure 3.1.

This figure also indicates that solving the very small system (3.20) may be difficult to parallelize. To solve system (3.20), one can either use the conjugate gradient method (which avoids assembling the matrix) or a Cholesky factorization. We have taken the second option.
Once Ξ is found, Λ^1 is computed as follows :

$$\Lambda^1 = -(\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T)^{-1} (\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{A}_{12} - \tilde{B}_2)\Xi, \tag{3.21}$$

which can be done in parallel.

To summarize the method, let us list the operations to :
A Assemble and factorize the matrices $\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T$ and $\mathcal{C}$.
B Solve (3.2).

A Matrices Assembling

1 Block Cholesky factorization of $\tilde{A}_{11}$.

2 Assembly of $\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T$:
For each basis function φ_h of W_h (correponding vector of coordinates : Φ):
3.1 Compute $\tilde{B}_1^T \Phi$.
3.2 Solve $\tilde{A}_{11} T_1 = \tilde{B}_1^T \Phi$, (use factorization of $\tilde{A}_{11}$).
3.3 Compute $\tilde{B}_1^T T_1$

3 Block Cholesky factorization of $\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T$.

4 Assembly of $\mathcal{C}$.
For each basis function ξ_h of Z_h^c : (correponding vector of coordinates : Ξ)
4.1 Compute $\Phi = (\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{A}_{12} - \tilde{B}_2)\Xi$, (use factorization of $\tilde{A}_{11}$).
4.2 Solve $\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T \Psi = \Phi$, (use factorization of $\tilde{B}_1 \tilde{A}_{11}^{-1} \tilde{B}_1^T$).
4.3 Compute $(\tilde{B}_2^T - \tilde{A}_{12}^T \tilde{A}_{11}^{-1} \tilde{B}_1^T)\Psi$, (use factorization of $\tilde{A}_{11}$).
4.4 Add $(\tilde{A}_{12}^T \tilde{A}_{11}^{-1} \tilde{A}_{12} - \tilde{A}_{22})\Xi$, (use factorization of $\tilde{A}_{11}$).

5 Cholesky factorization of $\mathcal{C}$.

B Solving 3.2

1 Solve $\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T\Lambda^0 = F$.(use factorization of $\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T$).

2 Compute $(\tilde{B}_2^T - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T)\Lambda^0$, (use factorization of $\tilde{A}_{11}$).

3 Solve $\mathcal{C}\Xi = (\tilde{B}_2^T - \tilde{A}_{12}^T\tilde{A}_{11}^{-1}\tilde{B}_1^T)\Lambda^0$, (use factorization of $\mathcal{C}$).

4 Compute $\Lambda^1 = -(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T)^{-1}(\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{A}_{12} - \tilde{B}_2)\Xi$, (use factorizations of $\tilde{A}_{11}$ and of $\tilde{B}_1\tilde{A}_{11}^{-1}\tilde{B}_1^T$).

5 Compute $\Lambda = \Lambda^0 + \Lambda^1$

Observe that only the algorithm described in subsection 3.2 requires solving

$$\tilde{A}V = Q. \tag{3.22}$$

Recalling that $\tilde{A}$ is positive definite (thanks to the perturbation of order ϵ), system (3.22) is solved exactly in the same way as (3.2) : after renumbering the unknonws, the system becomes :

$$\begin{pmatrix} \tilde{A}_{11} & \tilde{A}_{12} \\ \tilde{A}_{12}^T & \tilde{A}_{22} \end{pmatrix} \begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}.$$

After having solved

$$\tilde{A}_{11}V_1^0 = Q_1,$$

by using the block Cholesky factorization $\tilde{A}_{11}$, Q_2 is found by solving the following small linear system (unknowns located on crosspoints) :

$$\left(\tilde{A}_{22} - \tilde{A}_{12}\tilde{A}_{11}^{-1}\tilde{A}_{12}^T\right)V_2 = Q_2 - \tilde{A}_{12}V_1^0, \tag{3.23}$$

then $V_1 = V_1^0 + V_1^1$, where

$$\tilde{A}_{11}V_1^1 = -\tilde{A}_{12}^T V_2.$$

The following figure shows the locations of the degrees of freedom j such that $\left(\tilde{A}_{22} - \tilde{A}_{12}\tilde{A}_{11}^{-1}\tilde{A}_{12}^T\right)_{i,j} \neq 0$:

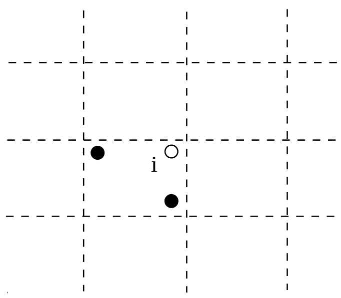

Figure 3.2.

As opposed to system (3.20), solving system (3.23) is very easy to parallelize.

4. Numerical Results

In this section, we study numerically the performances of the above mentioned preconditioners. In subsection 4.1, the test problem is described. In subsection 4.2, numerical tests are presented for a fixed domain decomposition but different mesh sizes. In subsection 4.3, the dependance on the number of subdomains is investigated. Subsection 4.4 is devoted to non uniform meshes, and subsection 4.5 to nonmatching grids.

4.1 The test problem.

The domain Ω is the approximation of the exterior of a circular body . Its boundary $\partial\Omega$ can be divided into two connected parts : the body's wall Γ_0 and the outer boundary Γ_∞.
Since exactly polygonal subdomains are much easier to treat than curved ones (in the first case isoparametric finite elements are needed whereas the second case requires overparametric finite elements), the subdomains should be preferably polygonal. For this reason, we set Γ_∞ to be piecewise polygonal. Therefore, the domain decomposition is designed so that every subdomain Ω_k is a polygonal, unless $\partial\Omega_k \cap \Gamma_0 \neq \emptyset$. In that case, $\partial\Omega_k$ is made of three straight segments and a curved line. The following figure represents an exemple of domain decomposition :

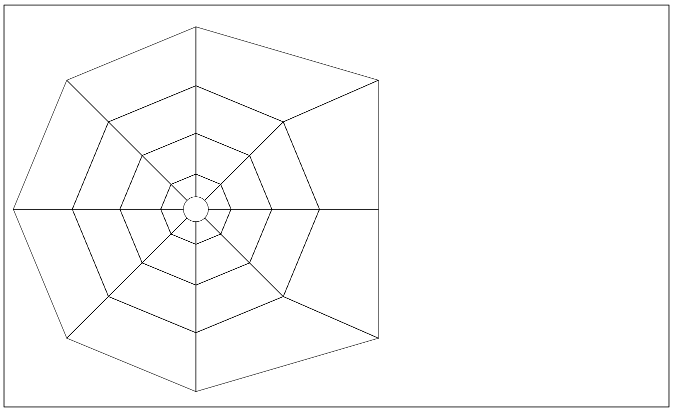

Figure 4.1 : an example of domain decomposition :
every subdomain Ω_k is polygonal, except if $\partial\Omega_k \cap \Gamma_0 \neq \emptyset$.

Let T_k map the unit square into Ω_k. The construction of T_k is rather simple when Ω_k is polygonal, (isoparametric transformation given by 6 parameters per subdomain). When Ω_k is curved, and $\partial\Omega_k$ has 3 straight segments, T_k is the composition of an affine transformation on a curved line and of the previously described isoparametric transformation (see fig 4.2).

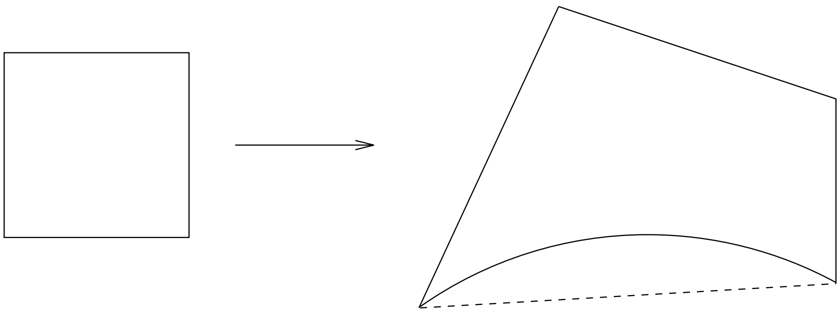

Figure 4.2 : the transformation T^k if $\partial\Omega_k \cap \Gamma_0 \neq \emptyset$.

For each subdomain Ω_k, a fine grid $\mathcal{T}_{k,h}$ is built in the following way : in the unit square, we construct a structured mesh, fully described by a few parameters : here, we use finite difference grids. Each quadrangle is divided into $nx \times ny$ smaller quadrangles. These divisions are made with parallel lines in the frame of reference where the quadrangle is a unit square (isoparametric mapping unless one border is curved). However the division is not uniform : in each subdomain, two aspect ratio are given so as to refine near one boundary in each direction: ρ_x, ρ_y . The mesh size increases geometrically from dx to $dx\rho_x$ from one cell to the next going West to Est and similarly in y northbound; To summarize, for each subdomain Ω_k, the mesh in the reference frame is described by the four parameters : nx, ny, ρ_x, ρ_y.

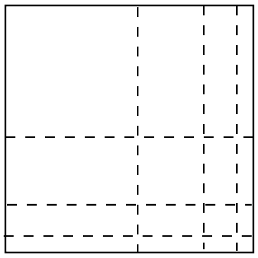

Figure 4.3 : Exemple of a mesh in the reference frame

Then this mesh is mapped into Ω_k by means of T_k. The meshes need not match at subdomains' interfaces. Therefore, this yields a nonconforming mesh for Ω with a few parameters only.
Isoparametric Q_1 finite elements are used in subdomains Ω_k such that $\partial\Omega_k \cap \Gamma_0 = \emptyset$. On the contrary, if an edge of $\partial\Omega_k$ is supported by Γ_0, then we use overparametric Q_1 finite elements.

We have chosen the following problem

$$\begin{aligned} &-\Delta u = 0 \quad \text{ in } \Omega, \\ &u = y \text{ on } \Gamma_0, \\ &\frac{\partial u}{\partial n} = 0 \text{ on } \Gamma_\infty, \end{aligned} \tag{4.1}$$

as a test problem, because it yields a linear system close to those encountered by the Navier Stokes solver.

The Uzawa method described in section 1 willl be refered to as U, the preconditioned Uzawa method of section 2 as P.U.1, the preconditioned Uzawa method of paragraph 3.1. as P.U.2. Finally, the preconditioned Schur complement method proposed in paragraph 3.2 will be denoted P.C.G. In the following, we shall always present the performances of U, P.U.1, P.U.2 on the same tables or graphics, and those of P.C.G apart. This is to emphasize the fact that the P.C.G method is different in nature from the other methods.

In order to use the algorithms U, P.U.1, P.U.2 in a straight forward manner, we must add a penalization term to equation (4.1). Indeed, we replace the equation in (4.1) with

$$-\Delta u + \varepsilon u = 0 \quad \text{ in } \Omega, \tag{4.2}$$

where ε is a small parameter (10^{-3} in the following experiments).

For all the experiments below, the stopping criteria of the iterative methods is :

$$\frac{\alpha_i}{min(\alpha_0, 1)} < 10^{-13}, \tag{4.3}$$

where α_i is the residual at iteration i (see the description of the different algorithm).

4.2 Influence of the mesh size.

In this series of experiments, the domain and the domain decomposition are fixed, but the mesh refinement is varied. Here, we consider meshes $\mathcal{T}_h$ such that for every subdomain Ω_k, the mesh $\mathcal{T}_{kh}$ is mapped (by T_k) to a uniform mesh in the frame of reference. We also restrict ourselves to the case of matching grids at subdomains'interfaces.
As shown on the next figure, there are 24 subdomains , disposed in 3 concentric rings of 8 subdomains each.

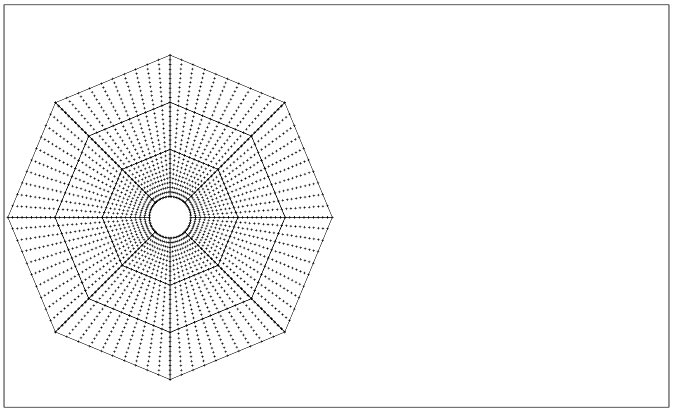

Figure 4.4 : Domain decomposition and mesh (mesh 1)

The four above mentioned methods have been tested on five different meshes. We have chosen the meshes such that the numbers nx and ny are the same for all subdomains. The next table shows the characteristic numbers of these meshes :

mesh	1	2	3	4	5
nb of subdomains	24	24	24	24	24
nx	10	15	25	30	38
ny	10	15	25	30	38
d.o.f. of U	2400	5400	15000	21600	34656
d.o.f. of U on Γ	800	1200	2000	2400	3040
d.o.f. of Λ	400	600	1000	1200	1520

Table 4.1. Meshes' characteristics

The next table presents the iteration numbers of the three methods based on the Uzawa algorithm

mesh	1	2	3	4	5
U	368	471	542	563	599
P.U.1	36	46	43	57	59
P.U.2	51	61	80	85	86

Table 4.2. Performances of U, P.U.1, P.U.2 .

This table shows that the two preconditioners are quite efficient. Algorithm P.U.2 is almost as fast as P.U.1 , with a much shorter assembly time.
The next table shows the number of iterations of P.C.G.

mesh	1	2	3	4	5
P.C.G	33	35	37	36	36

Table 4.3. Performances of P.C.G .

This table clearly shows that the convergence of P.C.G is almost unaffected by the size of the mesh. The number of iterations is the lowest among all the presented methods. Allthough one iteration of P.C.G is longer than one of P.U.2, the gain in iteration numbers makes P.C.G faster. To summarize, its straightforward application to the Laplace operator, short assembly time and fast convergence make the P.C.G method the most attractive.
It may be interesting to compare the histories of convergence . The two next figures show the convergence of the three Uzawa based algorithms and of the method P.C.G for mesh 5 :

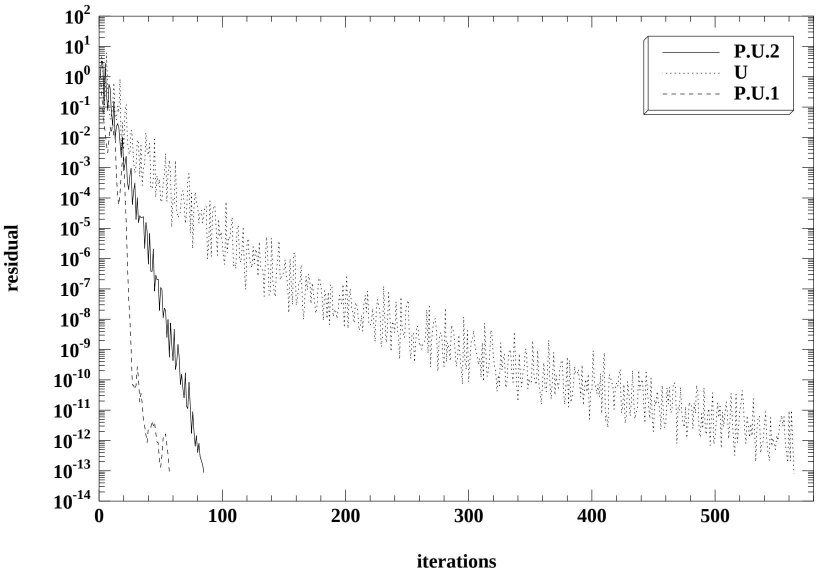

Figure 4.5. Convergence of U, P.U.1, P.U.2 for mesh 5 .

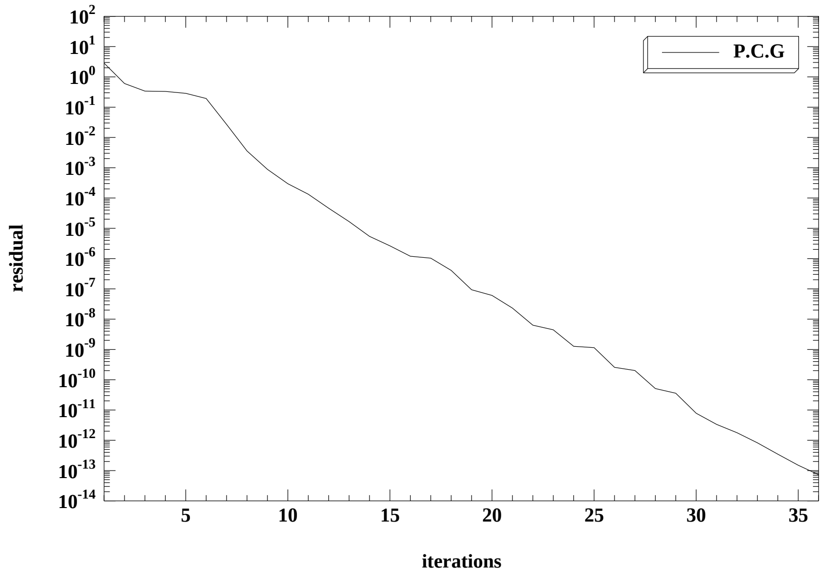

Figure 4.6. Convergence of P.C.G for mesh 5 .

4.3 Influence of the number of subdomains.

In this paragraph, we test the behaviour of the four algorithms when varying the number of subdomains. All the meshes used in the following series of experiments match at subdomains interfaces. Again, the parameters nx and ny are the same for all subdomains. As before, the domain is first split into n_{rings} concentring rings, then the rings are all divided into n_r subdomains. The meshes are designed in order to conserve the number of degrees of freedom for Λ : 600. The next table shows the characteristics of the different meshes :

mesh	1	2	3	4
nb of subdomains	24	18	12	8
n_{rings}	3	3	3	2
n_r	8	6	4	4
nx	15	28	52	50
ny	15	15	15	50
d.o.f. of U	5400	7560	9630	20000
d.o.f. of U on Γ	1200	1212	1192	1200
d.o.f. of Λ	600	606	596	600

Table 4.4. Meshes' characteristics

The iteration numbers of the three Uzawa based methods are listed below :

mesh	1	2	3	4
U	471	372	433	269
P.U.1	46	45	44	31
P.U.2	61	62	63	45

Table 4.5. Performances of U, P.U.1, P.U.2 .

The next table shows the number of iterations of P.C.G.

mesh	1	2	3	4
P.C.G	35	37	43	41

Table 4.6. Performances of P.C.G .

These tables seem to indicate that the preconditioners' efficiency is not much affected by the domain decomposition. The experiments for meshes 1, 2, 3 show that the preconditioner $\tilde{A}$ (used by P.U.2 and P.C.G) looses a little efficiency as the number of subdomains is decreased. This can be explained by noticing that the fewer the subdomains are, the less they look like rectangles (for which the approximation of S by $\tilde{A}$ is accurate.)

4.4 Influence of mesh refinement

The domain decomposition considered in this experiment is the same as in paragraph 4.2 (3 concentric rings, 8 subdomains per ring). However the mesh is refined near the obtacle. This kind of mesh is encountered when solving Navier Stokes problems at high Reynolds numbers, for treating correctly the boundary layer of the vorticity (or the velocity). In the first ring (the closest to the body), the subdomains' grids have the same characteristics :

$$nx = 15, \quad ny = 25, \quad \rho_x = 1, \quad \rho_y = 1.1.$$

In the other two rings, the subdomains' grids correspond to uniform meshes in the reference frame; they are given by :

$$nx = 15, \quad ny = 15, \quad \rho_x = 1, \quad \rho_y = 1.$$

The iteration numbers of the different solvers are displayed on the following table :

	U	P.U.1	P.U.2	P.C.G
P.C.G	-	46	67	36

Table 4.7. Performances for a non uniform mesh

These results should be compared to those of paragraph 4.2, especially for meshes 2 and 3. The unpreconditioned Uzawa method is the only method whose behaviour is much affected by the mesh refinement. In this particular case, preconditioning is crucial , as shown by the two next figures.

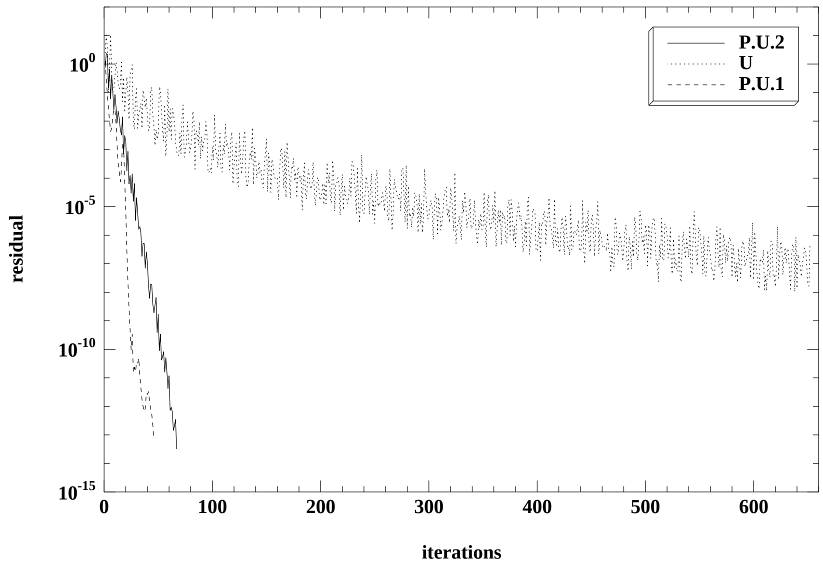

Figure 4.7. Convergence of U, P.U.1, P.U.2 .

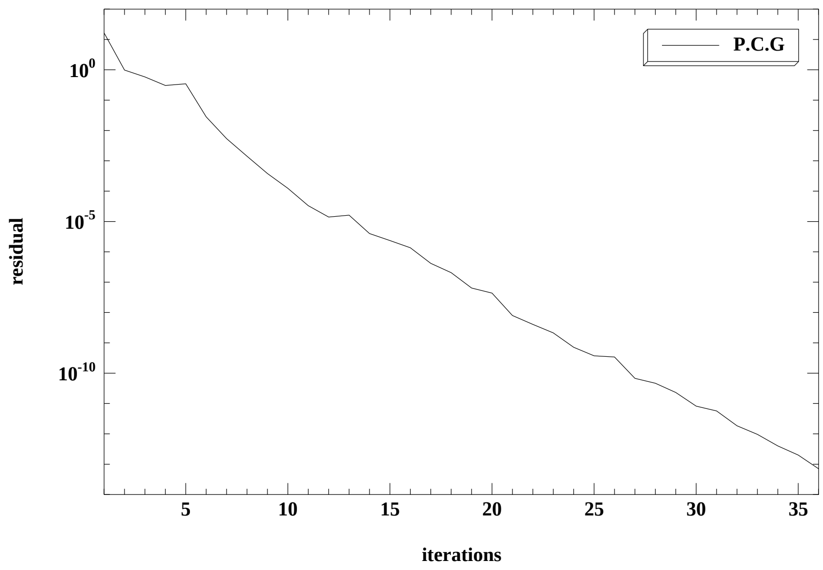

Figure 4.8. Convergence of P.C.G .

4.5 Influence of non matching grids

The domain decomposition considered in this experiment is the same as in paragraph 4.2 (3 concentric rings, 8 subdomains per ring). The subdomains'grids of the inner ring are described by the following parameters

$$nx = 20, \quad ny = 15, \quad \rho_x = 1, \quad \rho_y = 1,$$

whereas in the two other rings, the parameters are :

$$nx = 10, \quad ny = 15, \quad \rho_x = 1, \quad \rho_y = 1.$$

Therefore, the grids do not match at the interface of the two first rings of subdomains.
Note that the number of degrees of freedom of Λ, hence the results below should be compared to the results for meshes 2 and 3 of paragraph 4.2. The results are the following :

	U	P.U.1	P.U.2	P.C.G
P.C.G	-	59	56	36

Table 4.8. Performances for a case of non matching grids

Here again, the unpreconditioned Uzawa method does not converge in less than 650 iterations, whereas the other methods'behaviours are not affected .

Conclusion.

The mortar element method for the Laplace operator yields a quite ill conditioned linear system. A simple preconditioner has been presented, and two iterative methods using this preconditioner have been implemented and tested. The first one is a preconditioned Uzawa method, and the second one is a preconditioned conjugate gradient in the kernel of the jump operator. The second method is the best one, although more difficult to implement. Both methods yield a dramatic improvement with respect to the unpreconditioned Uzawa method. For the Uzawa method, the preconditioner has been succesfully compared with an incomplete Cholesky preconditioner which is much more expensive to assemble. Estimates on the condition number will be presented in a forthcoming work ([3]).

Bibliography

[1] Y. Achdou, O.Pironneau, *Integral Equation for the Generalized Stokes Operator with Applications to Boundary layer Matching*, Comptes Rendus Acad. Sci. Paris 315 Série I (1992), pp 91-96.

[2] Y. Achdou, O.Pironneau, *A fast solver for Navier-Stokes equations in the laminar regime using mortar finite element and boundary element method.*, Submitted.

[3] Y. Achdou, Yu.A. Kuznetsov,O.Pironneau, in preparation.

[4] V.K.Agapov, Yu.A. Kuznetsov. *On some versions of the domain decomposition methods.* Soviet J. of Num. Anal. and Math. modelling ,1988-3, pp 245-266.

[5] F.Ben Belgacem, Y. Maday, *A non conforming finite element method with application to parallel implementations* , submitted to M^3AS, (1993).

[6] C. Bernardi, Y. Maday, A. Patera, *A new nonconforming approach to domain decomposition : the mortar element method* , Nonlinear partial differential equations and their applications, Pitman, H. Brezis, J.L. Lions eds. (1989).

[7] J.F. Bourgat, R. Glowinski, P. Le Tallec, M. Vidrascu,*Variational Formulation and Algorithm for Trace Operator in Domain Decomposition Calculations.* in T.F Chan et al. eds., Second

International Symposium on Domain Decomposition Methods for Partial Differential Equations, pp3-17, Philadelphia, 1989.

[8] R. Glowinski, O.Pironneau, *Numerical methods for the first biharmonic equation and for the two dimensional Stokes problem.*, SIAM Review 21,No2, (1979) pp.167-212.

[9] Yu.R. Hakopian,Yu.A. Kuznetsov. *Algebraic Multigrid/Substructuring Preconditioners on Triangular Grids.*Soviet J. of Num. Anal. and Math. modelling ,1991, 6, pp 453-484

[10] Yu.A. Kuznetsov, G.I.Marchuk. *Méthodes itératives et fonctionelles quadratiques*, Méthodes Mathématiques de l'Informatique-4 : sur les méthodes numériques en Sciences physiques et économiques. (J.L. Lions ,G.I.Marchuk eds) pp3-132, Dunod, Paris, 1974.

[11] P.Le Tallec, T. Sassi. *Domain decomposition with non matching grids*, INRIA report (1991) .

[12] F.X Roux, *Méthodes de décomposition de domaine à l'aide de multiplicateurs de Lagrange, et applications à la résolution en parallèle des èquations de l'élasticité linéaire*, P.H.D thesis University of Paris,(1989).

Aknowledgements

The authors would like to thank O.Pironneau for making this work possible and for helpful discussions.