Analyse Numérique/ *Numerical Analysis*

Méthodes de Décomposition de Domaine pour la CAO

Jacques-Louis Lions, Olivier Pironneau

Dédié la mémoire de Jean Leray

Résumé En CAO la description des solides par Géométrie Constructive conduit des méthodes de décomposition de domaine basées sur les Òformes primitives", comme il est expliqué brivement dans l'introduction i-dessous. Un rle spécial est joué par les Òtrous" et ils peuvent tre approchés de multiples faons. Nous combinons cette observation avec la méthode des *Contrles Virtuels*, introduite dans une note précédente ([1]), o la décomposition et la formule de Green s'interpretaient comme des contrles virtuels – ici les contrles virtuels sont introduits a priori *sur des supports dans les trous*, ou *en dehors*, ou dans les intersections des domaines. Ils sont choisis pour que les conditions aux limites soient satisfaites modulo les erreurs d'approximations, ce qui est possible grce un *résultat de controlabilité approchée* (ce qu'il faut donc démontrer!). Ces idées sont présentées ici sur un exemple (Paragraphe 2) qui n'est certainement pas le plus général mais qui est suffisant pour comprendre que tout se généralise beaucoup d'autres situations. Ensuite le lemme de Controlabilité Approchée et les algorithmes sont donnés dans le Paragraphe 3 ainsi qu'un exemple numérique au Paragraphe 4.

Domain Decomposition Methods for CAD

Abstract Constructive Solid Geometry (CSG) in CAD leads to Domain Decompositions which are based on primitive shapes, as briefly explained in the introduction below. A special role is played by Òholes", which can be viewed in several different ways. We combine this remark with the method of *Virtual Controls*. In a previous note ([1]) where this method was introduced, the distributions resulting from the decomposition and Green's formula where thought of as virtual controls – here the virtual controls are introduced a priori with *support in the holes*, or *outside* the domain, or in the intersections of the domains. They are then chosen so as to (approximately) satisfy the Boundary Conditions, which is possible by virtue of *approximate controlability results* (which have to be proven!). These ideas are presented here on an example (Section 2) which is certainly not the most general one but which is sufficient to show how everything extends to very many other situations. Algorithms, following the Approximate Controllability Lemma, are given in Section 3 and a numerical example is presented in Section 4.

---

1. Version abrégée franaise. le travail présenté dans cette note est motivé d'une part par la Géométrie Solide Constructive (CSG en Anglais) o tout objet est décrit, autant qu'il est possible, par des domaines obtenus par les opérations topologiques élémentaires partir d'éléments primitifs (cubes, cylindres, spheres, cones) et d'autre part par les situations rencontrées dans la méthode Chimre o l'on utilise plusieurs grilles structurées ne se recollant pas.

On introduit une méthode trs générale basée sur l'usage de contrles virtuels et sur un résultat de contrlabilité approchée. On déduit de cela des algorithmes (CSG algorithmes) simples dont l'efficacité est vérifiée sur des exemples dont l'un est donné dans le texte.

1. Introduction This study is motivated by two applications.

1. In Image Synthesis and Virtual Reality (VR) (see [2] for instance) scenes are entered by Constructive Solid Geometry, i.e. each object of the scene is described by set operations on primitive shapes like cubes, cylinders, spheres and cones. For instance a cubic room with a table inside can be described in VRML (the language of VR) as a cube with four cylinder and a brick inside it, the table's plateau. These elementary objects are positioned at their proper places but never intersected. Rendering on a computer screen is done by triangulation of the surface of each elementary shape and the painter's algorithm with a Z-buffer. Scientific Computing (like the computation of the temperature in the room) in such domains requires a translation from VR data structure to CAD data structure, but this operation is difficult. We propose an alternative by domain decomposition.

2. The Chimera method [3] in CFD uses several non-intersecting structured meshes to avoid the problem of mesh generation on difficult geometries. A typical example is a wing and its engine each having its own mesh. Our algorithm proves the method.

In both cases the domain Ω is a union and substraction of simple domains. We assume that each domain can be meshed separately and that if a domain is subtracted it is possible to surround it by an admissible mesh. We will compute the solution of a PDE on Ω by an iterative algorithm involving computation of the PDE on each domain only.

Such algorithms already exist ; Schwarz [4] if Ω is a union of sets ; the fictitious domain method [5][6][7] if Ω has holes only. But for the general case the problem seems to be open.

The method is presented in general and it is tested in 2D. There are of course many possibilities and we have been guided by computational efficiency :

1. We assume that we have an efficient general interpolator [8][9] to compute a function irrespectively of the mesh on which it is originally defined.

2. For a problem discretized on a mesh T we avoid computing integrals on curves which are not union of edges of T.

2. Statement of the problem

Let Ω be a bounded open set of $\mathbb{R}^d, d = 2$ or 3, such that

$$\Omega = \Omega_1 \cup \Omega_2, \quad \Omega_i = \text{ open set, } \Omega_1 \cap \Omega_2 \neq \emptyset$$

Moreover, we assume *that* Ω_1 *has a hole* C_1, the position of Ω_i and C_1 being represented on Fig. 1.

Fig. 1

Remark 1 The set C_1 does *not* intersect Ω_2. We can extend what follows to the case where $C_1 \cap \Omega_2 \neq \emptyset$, as we will show elsewhere. □

Remark 2 All what follows readily extend to the cases where there is also a hole C_2 in Ω_2, which does not intersect $\Omega_1 \cap \Omega_2$. □

Remark 3 We shall show elsewhere how all what follows extends to the case of a *collection of domains*: $\Omega = \Omega_1 \cup \ldots \cup \Omega_m$ with *arbitrarily disposed holes.* □

The boundary of Ω consists of $\Gamma_1 \cup \Gamma_2 \cup \partial C_1$, with the notations of Fig. 1 ($\Gamma_1 =$ that part of $\partial\Omega_1$ which is outside Ω_2 and the other way around for Γ_2).

Let $a(u, \hat{u}) = \sum_{i,j} \int_\Omega a_{ij} \frac{\partial u}{\partial x_j} \frac{\partial \hat{u}}{\partial x_i} dx$ be defined on the Sobolev space $H_0^1(\Omega)$ (functions in $H^1(\Omega)$ which are zero on $\partial\Omega$).

We assume that

$$a_{ij} \in L^\infty(\Omega), \sum a_{ij}(x)\xi_i\xi_j \geq \alpha \sum \xi_i^2, \alpha > 0, a_{ij} \text{ not necessarily symmetric} \qquad (1)$$

the a_{ij}'s *are smooth enough such that the unique continuation theorem holds true.* (2)

Let f be given in $L^2(\Omega)$ (to fix ideas). We consider the *unique solution* u *in* $H_0^1(\Omega)$

$$a(u, \hat{u}) = (f, \hat{u}) \quad \forall \hat{u} \in H_0^1(\Omega), \qquad (3)$$

where $(f, \hat{u}) = \int_\Omega f\hat{u}dx$.

We want to give a completely general method of approximation of u based on *boundary value problems in domains without holes.*

3. Virtual controls and approximate controllability

We are going to use (cf. Fig. 2) $\Omega_1 \cup \bar{C}_1$ (domain *without* the hole C_1)

$$\mathcal{D}_1 = \text{ neighborhood of } \bar{C}_1 \text{ such that } \quad \bar{C}_1 \subset \mathcal{D}_1, \quad \mathcal{D}_1 \cap (\Omega_1 \cap \Omega_2) = \emptyset.$$

and we denote by $S_1 =$ that part of $\partial\Omega_1$ which is inside Ω_2 (and the other way around for S_2).

Fig. 2

We define
$$V_1 = \{v_1 | v_1 \in H^1(\Omega_1 \cup \bar{C}_1), v_1 = 0 \quad \text{on} \quad \Gamma_1\}$$
$$V_2 = \{v_2 | v_2 \in H^1(\Omega_2), v_2 = 0 \quad \text{on} \quad \Gamma_2\}$$

$$a_1(u_1,\hat{u}_1)=\sum_{i,j}\int_{\Omega_1\cup\bar{C}_1}a_{ij}(x)\frac{\partial u_1}{\partial x_j}\frac{\partial \hat{u}_1}{\partial x_i}dx, u_1,\hat{u}_1\in V_1$$

where a_{ij} is extended by continuity into $\bar{C}_1$ in such a way that the ellipticity condition (1) and the uniqueness property (2) are valid in C_1. We define $a_2(u_2,\hat{u}_2)$ in the same way (no extension is needed here since Ω_2 has no hole) for $u_2,\hat{u}_2\in V_2$.

In order to avoid any confusion, we define for $w_1,\hat{w}_1\in H^1(\mathcal{D}_1)$.

$$a_{1\mathcal{D}_1}(w_1,\hat{w}_1)=\sum_{i,j}\int_{\mathcal{D}_1}a_{ij}(x)\frac{\partial w_1}{\partial x_j}\frac{\partial \hat{w}_1}{\partial x_i}dx.$$

The algorithms presented here use only the bilinear forms $a_1, a_{1\mathcal{D}_1}, a_2$ and their adjoints.
□

We now introduce the *virtual controls.*

Let $\mathcal{O}$ be an open set contained in $\Omega_1\cap\Omega_2$, and let $\mathcal{G}_1$ be an open set contained in $\mathcal{D}_1, \mathcal{G}_1\cap\mathcal{C}_1\neq\emptyset$.

Fig 3 Fig. 4

($\mathcal{O}$ and $\mathcal{G}_1$ are shaded on the (magnified) Fig. 3 and 4). We set 1_X = characteristic function of a set X. We introduce the *virtual controls* $\lambda_1,\lambda_2\in L^2(\mathcal{O}),\ \mu_1,\nu_1\in L^2(\mathcal{G}_1)$:

$$\lambda_1+\lambda_2=0,\quad \mu_1+\nu_1=0\quad\text{on}\quad \mathcal{G}_1\cap(\mathcal{D}_1\backslash\mathcal{C}_1). \tag{4}$$

Remark 4

We could certainly save the index Ò1” in $\mathcal{C}_1,\mathcal{D}_1,\mathcal{G}_1,\mu_1,\nu_1$ but we keep it here so as to make the extension to the case where there is also a hole in Ω_2 completely straightforward.
□

We still denote by f any extension of f inside the hole C_1, such that $f\in L^2(\Omega\cup\bar{C}_1)$ (we can extend it by 0 inside C_1) and we decompose f in

$$f=f_1+f_{1\mathcal{D}_1}+f_2, f_1\in L^2(\Omega_1\cup\bar{C}_1), f_{1\mathcal{D}_1}\in L^2(\mathcal{D}_1), f_2\in L^2(\Omega_2) \tag{5}$$

(where the functions are extended by 0 where they have to be extended).

We now define

$$u_1\in V_1,\quad w_1\in H^1(\mathcal{D}_1),\quad u_2\in V_2 \tag{6}$$

as the solutions of

$$\begin{aligned}&a_1(u_1,\hat{u}_1)=(f_1,\hat{u}_1)+(\lambda_1 1_{\mathcal{O}},\hat{u}_1)+(\mu_1 1_{\mathcal{G}_1},\hat{u}_1)\quad\forall\hat{u}_1\in V_1,\\&a_{1\mathcal{D}_1}(w_1,\hat{w}_1)=(f_{1\mathcal{D}_1},\hat{w}_1)+(\nu_1 1_{\mathcal{G}_1},\hat{w}_1)\forall\hat{w}_1\in H^1(\mathcal{D}_1),\\&a_2(u_2,\hat{u}_2)=(f_2,\hat{u}_2)+(\lambda_2 1_{\mathcal{O}},\hat{u}_2)\quad\forall\hat{u}_2\in V_2.\end{aligned}\tag{7}$$

For a given choice of the virtual controls λ_i,μ_i,ν_i, the equations (7) (which can be solved in parallel) uniquely define the Òstate” u_1,w_1,u_2.

Let us admit for a moment the

Approximate Controllability Lemma. *When the virtual controls λ_i,μ_i,ν_i span $L^2(\mathcal{O}),L^2(\mathcal{G}_1)$ subject to conditions (4), the functions $(u_1+w_1)|_{\partial C_1},w_1|_{\partial\mathcal{D}_1},u_1|_{S_1},u_2|_{S_2}$ span a dense subset of $L^2(\partial C_1)\times L^2(\partial\mathcal{D}_1)\times L^2(S_1)\times L^2(S_2)$.*

Then, one can find $\lambda_1,\lambda_2,\mu_1,\nu_1$ such that

$$\begin{aligned}&u_1+w_1\simeq 0\quad\text{on}\quad\partial C_1(\simeq 0\text{ means of }L^2(\partial C_1)\text{ norm arbitrarily small})\\&w_1\simeq 0\quad\text{on}\quad\partial D_1,\quad u_1\simeq 0\quad\text{on}\quad S_1,\quad u_2\simeq 0\quad\text{on}\quad S_2.\end{aligned}\tag{8}$$

But by virtue of (7) one has

$$\frac{\partial u_1}{\partial\nu}=0\quad\text{on}\quad S_1,\frac{\partial w_1}{\partial\nu}=0\quad\text{on}\quad\partial\mathcal{D}_1,\frac{\partial u_2}{\partial\nu}=0\quad\text{on}\quad S_2.$$

(where $\frac{\partial}{\partial\nu}$ = co-normal derivative attached to $A=-\sum\frac{\partial}{\partial x_i}(a_{ij}\frac{\partial}{\partial x_j}))$ so that if we extend u_1 (resp. u_2, resp. w_1) by 0 outside Ω_1 (resp. Ω_2, resp. $\mathcal{D}_1$) we have ”approximately”, outside S_i and ∂D_1 that $u_1\in H^2(\Omega),u_2\in H^2(\Omega),w_1\in H^2(\Omega)$. According to (7), we have

$$Au_1=f_1+\lambda_1 1_{\mathcal{O}}+\mu_1 1_{\mathcal{G}_1},\quad Aw_1=f_{1\mathcal{D}_1}+\nu_1 1_{\mathcal{G}_1},\quad Au_2=f_2+\lambda_2 1_{\mathcal{O}}.$$

Adding up and using (4), we obtain $A(u_1+w_1+w_2)=f,u_1+w_1+u_2\simeq 0\quad\text{on}\quad\partial C_1,u_1+w_1+u_2=0\quad\text{on}\quad\Gamma_1\cup\Gamma_2$. *so that $u_1+w_1+u_2$ is an approximation of the solution u.* □

Before showing in Section 3 below how all this can be applied to find *an iterative algorithm to compute the virtual controls leading to (8)*, let us give a hint of the proof of the approximate controllability lemma.

By translation, we can assume that $f_1,f_{1\mathcal{D}_1}f_2$ are zero. Let g_1,h_1,l_1,l_2 be in $L^2(\partial C_1)\times L^2(\partial\mathcal{D}_1)\times L^2(S_1)\times L^2(S_2)$ such that

$$\int_{\partial C_1}g_1(u_1+w_1)+\int_{\partial\mathcal{D}_1}h_1w_1+\int_{S_1}l_1u_1+\int_{S_2}l_2u_2=0\quad\forall\lambda_i,\mu_1,\nu_1\tag{9}$$

(where the measure on the surfaces are not written). We define the functions p_1,q_1,p_2 $p_1\in V_1,q_1\in H^1(\mathcal{D}_1),p_2\in V_2$, as the solutions of

$$\begin{aligned}&a_1^*(p_1,\hat{p}_1)=\int_{\partial C_1}g_1\hat{p}_1+\int_{S_1}l_1\hat{p}_1\quad\forall\hat{p}_1\in V_1,\\&a_{1\mathcal{D}_1}^*(q_1,\hat{q}_1)=\int_{\partial C_1}g_1\hat{q}_1+\int_{\partial\mathcal{D}_1}h_1\hat{q}_1\quad\forall\hat{q}_1\in H^1(\mathcal{D}_1),\\&a_2^*(p_2,\hat{p}_2)=\int_{S_2}h_2\hat{p}_2\quad\forall\hat{p}_2\in V_2,\end{aligned}\tag{10}$$

where a_i^* denotes the adjoint of a_i.

By taking $\hat{p}_1 = u_1, \hat{q}_1 = w_1, \hat{p}_2 = u_2$ in (10), and using (7) (where the f's are zero), and also using (4), we obtain :

$$\begin{aligned} &p_1 = 0 \quad \text{in} \quad \mathcal{G}_1 \cap \mathcal{C}_1, \quad q_1 = 0 \quad \text{in} \quad \mathcal{G}_1 \cap \mathcal{C}_1, \\ &p_1 = q_1 \quad \text{in} \quad \mathcal{G}_1 \cap (\mathcal{D}_1 \backslash \mathcal{C}_1), \quad p_1 = p_2 \quad \text{in} \quad \mathcal{O}. \end{aligned} \tag{11}$$

But $A^* p_1 = 0$ inside $\mathcal{C}_1$ and using the *uniqueness property* (we use (2) here), it follows that

$$p_1 = 0 \quad \text{in} \quad \mathcal{C}_1, \text{ and (analogously) } q_1 = 0 \quad \text{in} \quad \mathcal{C}_1. \tag{12}$$

By using the same uniqueness theorem (in various domains) we obtain that

$$p_1 = q_1 \quad \text{in} \quad \mathcal{D}_1 \backslash \mathcal{C}_1, p_1 = p_2 \quad \text{in} \quad \Omega_1 \cap \Omega_2. \tag{13}$$

But p_1 is regular enough such that, by (12),

$$p_1|_{\partial \mathcal{C}_1} = 0 \tag{14}$$

If we then set $\pi = p_1$ in Ω_1, p_2 in Ω_2 (they adjust on $\Omega_1 \cap \Omega_2$ by (13)), one has

$$A^* \pi = 0 \quad \text{in} \quad \Omega, \pi = 0 \quad \text{on} \quad \Gamma_1 \cup \Gamma_2, \pi = 0 \quad \text{on} \quad \partial C_1,$$

by (14), so that $\pi = 0$ in Ω. Therefore $p_1 = 0$ in $\Omega_1 \cup C_1$, $p_2 = 0$ in Ω_2 and therefore $g_1 = 0, l_1 = 0, l_2 = 0$. But by (13) $q_1 = p_1$ in $\mathcal{D}_1 \backslash \mathcal{C}_1$ so that $q_1 = 0$ in $\mathcal{D}_1 \backslash \mathcal{C}_1$ and in $\mathcal{C}_1$, hence $h_1 = 0$. □

Remark Properties analogous to (2), needed here to establish (12), for finite element approximations are known to be an open question in general.

3. The CSG Algorithms (Constructive Solid Geometry Algorithms)

We penalize conditions (8). It leads to the introduction of the cost function

$$\begin{aligned} \mathcal{J}(\lambda, \mu_1, \nu_1) = \frac{\varepsilon}{2}[\int_{\mathcal{O}} (\lambda_1^2 + \lambda_2^2) dx + \int_{\mathcal{G}_1} (\mu_1^2 + \nu_1^2) dx] + \\ + \frac{1}{2} \int_{\partial C_1} (u_1 + w_1)^2 + \frac{1}{2} \int_{\partial \mathcal{D}_1} w_1^2 + \frac{1}{2} \int_{S_1} u_1^2 + \frac{1}{2} \int_{S_2} u_2^2, \end{aligned} \tag{15}$$

and we introduce a *gradient algorithm* for the approximation of

$$\inf_{\lambda_1, \lambda_2, \mu_1, \nu_1} \mathcal{J}(\lambda, \mu_1, \nu_1)$$

(subject to (4)). We introduce p_1, q_1, p_2 by

$$\begin{aligned} &a_1^*(p_1, \hat{p}_1) = \int_{\partial C_1} (u_1 + w_1) \hat{p}_1 + \int_{S_1} u_1 \hat{p}_1 \quad \forall \hat{p}_1 \in V_1, p_1 \in V_1, \\ &a_{1\mathcal{D}_1}^*(q_1, \hat{q}_1) = \int_{\partial C_1} (u_1 + w_1) \hat{q}_1 + \int_{\partial \mathcal{D}_1} w_1 \hat{q}_1 \quad \forall \hat{q}_1 \in H^1(\mathcal{D}_1), q_1 \in H^1(\mathcal{D}_1), \\ &a_2^*(p_2, \hat{p}_2) = \int_{S_2} u_2 \hat{p}_2 \quad \forall \hat{p}_2 \in V_2, p_2 \in V_2. \end{aligned} \tag{16}$$

Then, for $\rho > 0$ sufficiently small, one defines

$$\begin{aligned}
\mu_1^{n+1} &= \mu_1^n - \rho(\varepsilon\mu_1^n + \begin{cases} p_1^n & \text{in} \quad \mathcal{G}_1 \cap \mathcal{C}_1 \\ p_1^n - q_1^n & \text{in} \quad \mathcal{G}_1 \cap (\mathcal{D}_1 \backslash \mathcal{C}_1) \end{cases}) \,, \\
\nu_1^{n+1} &= \nu_1^n - \rho(\varepsilon\nu_1^n + \begin{cases} q_1^n & \text{in} \quad \mathcal{G}_1 \cap \mathcal{C}_1 \\ p_1^n - q_1^n & \text{in} \quad \mathcal{G}_1 \cap (\mathcal{D}_1 \backslash \mathcal{C}_1) \end{cases}) \,, \\
\lambda_i^{n+1} &= \lambda_i^n - \rho(\varepsilon\lambda_i^n + p_i^n) \quad \text{in} \quad \mathcal{O}, \; i = 1, 2
\end{aligned} \tag{17}$$

One computes $u_1^{n+1}, w_1^{n+1}, u_2^{n+1}$ (in parallel) by (7) where $\lambda_1 = \lambda_1^{n+1}$ etc, and then one computes (in parallel) $p_1^{n+1}, q_1^{n+1}, p_2^{n+1}$ by (15), and one proceeds. □

Remark 5

Everything applies with trivial changes to the cases where other Boundary Conditions are used on $\Gamma_1 \cup \Gamma_2$ and on ∂C_1. We shall return on that. □

Remark 6

All what has been presented extends to *higher order boundary value problems* and to *systems* of equations. □

Remark 7

Problems are (of course) much more difficult in the *nonlinear cases*.

We *conjecture* that the approximate controllability Lemma is still (in general) valid for nonlinear problems, but this has not been proven. □

Remark 8

For *control problems* (i.e. problems where there is a *Òreal" control* say v) one can still apply the above remarks. We deal then with virtual *and* real controls. The cost function (15) becomes now a function of the virtual controls and of the real ones and one can apply conjugate gradients algorithms.

With different techniques (based on *Domain Decomposition Methods*) this idea has been used in the note (^{1}) of the authors. □

Remark 9

Of course the conditions (8) could be treated by other methods than penalty, such as Lagrange multipliers, augmented Lagrangians etc... □

3. A numerical example

We wish to compute the temperature $u(x)$ in a spanner. The geometry Ω is entered by set operations on a rectangle Ω_1, a cercle Ω_2 a hole C_1 and the mouth C_2 (see Fig. 5): $\Omega = \Omega_1 \cup \Omega_2 \backslash (C_1 \cup C_2)$. The partial Differential Equation is

$$-\Delta u = 0, \quad \text{in } \hat{\text{E}}\hat{E}\Omega, \quad u|_{\partial C_1 \cup \partial C_2} = 100, \quad u|_{\partial\Omega - \partial C_1 \cup \partial C_2} = 0$$

where C_1 is the mouth of the spanner and C_2 the hole.

In order to account for the holes we construct two arbitrary domains Ω_3 and Ω_4 around C_1 and C_2 and included respectively in Ω_1 and Ω_2. So we use the Virtual Control algorithm with four subdomains.

Discretization is done with the Finite Element Method of order one on triangles; we have used freefem+[9] to handle the interpolations on different meshes in an efficient way.

Figure 5 shows the 4 subdomains, the solution of the Laplace equation obtained, a convergence history and a one dimensional plot of $\lambda \to u(x(\lambda), y(\lambda))$ where $\lambda \to \{x(\lambda), y(\lambda)\}$ is the line joining the upper right corner of the spanner mouth with the center of the hole.

Conclusion

This example shows that the method is numerically efficient. However more work needs to be done, one to justify convergence at the discrete level (an inf-sup condition is likely to be necessary for compatibility between meshes) and two to treat extrusions like the mouth of the spanner here, because in practice extrusion of a set D from Ω is done by $\Omega \backslash (\Omega \cap D)$ while here we require that $D \subset \Omega$.

Références bibliographiques

[**1**] J.L. LIONS, O. PIRONNEAU, Algorithmes parallles pour la solution de problmes aux limites. C.R.A.S., Nov. 1998.

[2] G.BURDEN, Ph. COIFFE *Virtual Reality Technology*, New York, Wiley 1994.

[3] J.L. STEGER, The Chimera method of flow simulation,Workshop on applied CFD, Univ of Tennessee Space Institute, August 1991.

[4] P.L. LIONS, On the Schwarz alternating method I., in *First International Symposium on Domain Decomposition Methods for Partial Differential Equations* (Paris, 1987), SIAM, Philadelphia, PA, 1-42. 1988.

[5] Y. KUZNETSOV, D. TRUFANOV. Two-stage fictitious components method for solving the wave Helmoltz equation, Sov. J. Numer. Anal. Math. Modelling, III ,5, pp371-391, 1988.

[6] R. GLOWINSKI, T. PAN, J. PERIAUX A fictitious domain method for external incompressible flows modelled by Navier-Stokes equations. Univ. Houston internal report, 1992.

[7] O. PIRONNEAU, Fictitious Domain Methods versus Boundary Fitted Meshes, *Proc Regional Conference in applied math and engineering*, E. Oate ed. La Corua, Spain, 1993.

[8] F. HECHT, J.L. LIONS, O. PIRONNEAU , Algorithmes de points fixes et de contrle pour la CAO par CSG. (to appear).

[9] F. HECHT, O. PIRONNEAU , Handling multiple triangulations in freefem+, INRIA report (to appear).

- *Jacques-Louis Lions, Collge de France, 3, rue d'Ulm, 75231 Paris Cedex 05*

- *Olivier Pironneau, Institut Universitaire de France et Laboratoire d'Analyse Numérique, Université Pierre et Marie Curie, 4, place Jussieu, 75231 Paris Cedex 05.*

.

Fig. 5

The spanner is the union of a circle Ω_2 (upper left) with a rectangle Ω_1 (upper right) minus two wholes C_1, C_2 which, for computational purpose, are surrounded by triangulated artifical domains (lower left triangulation for the mouth and lower right for the spanner hole). The solution of the PDE after 11 iterations is displaid in the center; no jump discontinuities are visible. A convergence history of the control cost function J and of the L^2 error with the standard FEM solution is given on the lower left graph. On the lower right graph a one dimensional plot of the solution is shown on a line joining the upper corner of the spanner mouth with the center of the left hole; iteration 1,2 and the standard FEM solution are shown , the difference with the solution at iteration 11 is not visible.