Schwarz's Decomposition Method for Incompressible Flow Problems

M. FORTIN* AND R. ABOULAICH*

1. Introduction

Three-dimensional fluid dynamics computation imply the solution of very large linear systems and there seems to be an agreement that direct methods are too expensive to handle them efficiently.

Among the possible alternative paths that can be followed, domain decomposition comes quite naturally to the mind: if many small systems could be properly connected, computing effort could be split into pieces that could even be dealt with in parallel.

There is by no means a unique way to do so. We shall describe in this paper a Schwarz's type (overlapping domains) decomposition algorithm for the numerical solution of the Navier-Stokes equations of incompressible flows. We shall consider its performance in linear Stokes problem, then in the nonlinear case. Finally we shall discuss possible variants, some of which are presently under test.

We shall thus consider in the following a problem of type (cf. Figure 1.1)

$$\alpha \underline{u} - \nu \Delta \underline{u} + \lambda \underline{u} \cdot \underline{\nabla}\, \underline{u} + \underline{\nabla}\, p = \underline{f} \quad \text{on} \quad \Omega \,, \tag{1.1}$$

$$\underline{\nabla} \cdot \underline{u} = 0 \quad \text{on} \quad \Omega \,, \tag{1.2}$$

$$\underline{u}\Big|_{\Gamma_0} = \underline{g}_0 \quad \text{on} \quad \Gamma_0 \,, \tag{1.3}$$

$$p + \frac{\partial \underline{u} \cdot \underline{n}}{\partial n} = g_{1n} \,, \quad \frac{\partial \underline{u} \cdot \underline{n}}{\partial t} + \frac{\partial \underline{u} \cdot \underline{t}}{\partial n} = g_{1t} \quad \text{on} \quad \Gamma_1 \,. \tag{1.4}$$

*Département de Mathématiques, Statistique et Actuariat, Université Laval, Québec G1K 7P4 Canada.

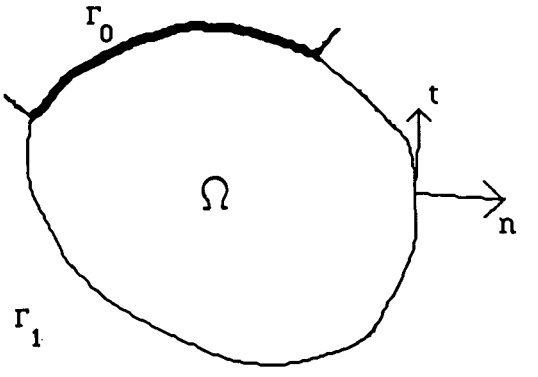


Figure 1.1

Such a problem arises when a time discretization procedure has been applied to an unsteady Navier-Stokes problem. (cf. Dinh-Glowinski-Periaux [6]). For $\alpha = 0$, $\lambda = 1$ we have a standard steady-state problem. Taking $\lambda = 0$ we get a Stokes-like linear problem which we shall first consider.

Some remarks will be needed in the sequel about boundary conditions. Whenever $\Gamma_1 = 0$, (that is if we only have Dirichlet conditions), there is a constraint on the choice of g_0 that is dictated by the divergence-free condition. By Stokes' theorem we must indeed have,

$$(1.5) \qquad \int_\Gamma g_0 \cdot n = 0 \ .$$

This is nothing but balance of mass on Ω. When we shall solve problems on subdomains, such a condition will have to hold on every subdomain. We also recall that pressure is then defined only up to an additive constant. These difficulties however disappear whenever stress is imposed on a part of the boundary.

2. Approximation of a Stokes problem

We have already described in a previous paper a relaxation process for the numerical solution of Stokes problem. We rapidly recall its main features. To make clear why such a procedure can

work and understand some conditions for its use, we need first recall a few parts about the variational formulation of the problem. Defining now,

$$(2.1) \qquad V = \{\underline{v} \mid \underline{v} \in (H^1(\Omega))^n \ , \ \underline{v}\Big|_{\Gamma_0} = 0\} \ ,$$

where $H^1(\Omega)$ is the usual Sobolev space, and also denoting

$$(2.2) \qquad Q = L^2(\Omega) \ ,$$

we consider the variational problem.

$$(2.3) \qquad \begin{cases} \alpha \int_\Omega \underline{u}\cdot\underline{v}\ dx + \nu \int_\Omega \underline{\nabla}\ \underline{u} : \underline{\nabla}\ \underline{v}\ dx - \int_\Omega p\ \underline{\nabla}\cdot\underline{v}\ dx = \int_\Omega \underline{f}\cdot\underline{v}\ dx \quad \forall v \in V, \\ \int_\Omega \nabla\cdot u\ q\ dx = 0 \qquad \forall q \in Q \ . \end{cases}$$

These equations are the optimality conditions of a saddle-point problem,

$$(2.4) \qquad \inf_{v\in V} \sup_{q\in Q} \frac{\alpha}{2} \int_\Omega |v|^2 dx + \frac{\nu}{2} \int_\Omega |\underline{\nabla}\ \underline{v}|^2 dx - \int_\Omega q\ \underline{\nabla}\cdot\underline{v}\ dx - \int_\Omega \underline{f}\ \underline{v}\ dx \ .$$

However, if we restrict ourselves to the divergence-free subspace V_o of V, our problem becomes a standard minimization problem.

$$(2.5) \qquad \inf_{v\in V_o} \frac{\alpha}{2} \int_\Omega |\underline{v}|^2 dx + \frac{\nu}{2} \int_\Omega |\underline{\nabla}\ \underline{v}|^2 dx - \int_\Omega \underline{f}\cdot\underline{v}\ dx \ .$$

This shows that, although minimization techniques such as relaxation methods and conjugate-gradient methods cannot be applied to problem (2.4), they will work if we are able to keep the iterative process in V_o or its discrete analogue. To fix ideas we present a few examples of finite-element approximations of (2.4) and rapidly describe their subspace of discrete divergence-free elements.

Example 2.1: $Q_1 - P_1$ approximation (Figure 2.1).

This is the classical (and difficult to analyze) approximation by a bilinear element for velocity and piece-wise constant pressure. The discrete divergence-free condition reduces to element-wise conservation of mass. This element has an obvious 3-D extension which is one of the most widely used 3-D element because of its (apparent?) low cost. This element is well known not to satisfy the inf-sup condition (Brezzi-Fortin [5], Girault-Raviart [10]) and exhibits the famous checkerboard pressure mode. □

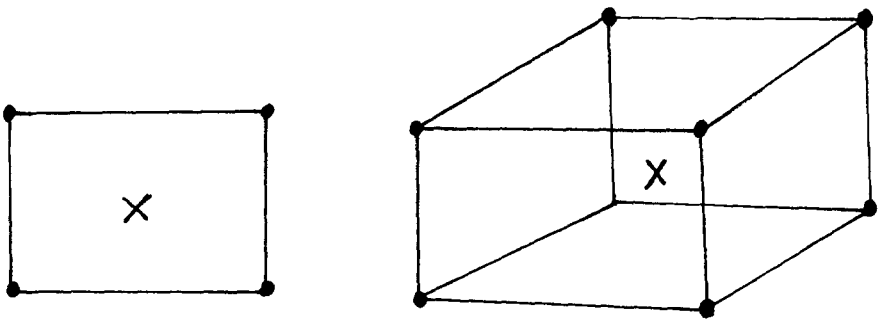


Figure 2.1: Q_1–P_0 · velocity mode
x pressure mode.

Exemple 2.2: The $Q_1^+ - P_0$ elements (Figure 2.2).

The above element can be stabilized (in the sense of the inf–sup condition) by adding normal velocities as degrees of freedom on element interfaces. This extra mode can be added either by making the final element conforming or nonconforming, the latter seeming to yield a slightly better approximation. □

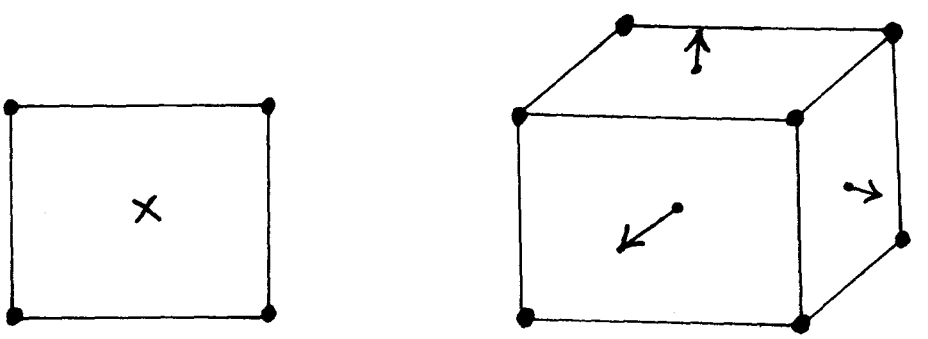


Figure 2.2: $Q_1^+ - P_0$ elements · velocity mode
→ normal velocity mode
x pressure mode.

An important point for our decomposition procedure will be to know, at least qualitatively, a basis of V_{0h} the discrete-divergence free

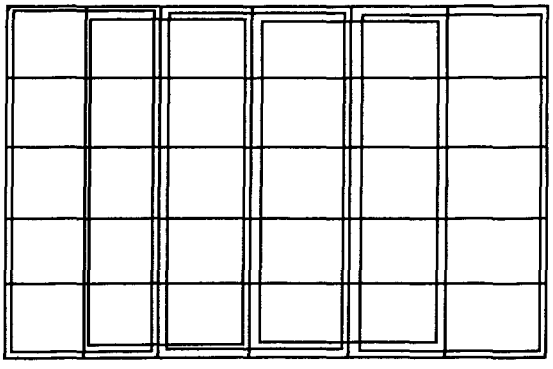
supspace. On a regular mesh this can easily be elucidated and an important difference arises between elements of Example 2.1 and those of Example 2.2.

For $Q_1 - P_0$ element the result is mesh dependent: on a perfectly regular mesh, one basis function can be associated with every 3×3 or 3×3×3 set of elements. For a distorted mesh, 4×4 or 4×4×4 sets have to be used to find a non-zero divergence-free function. (See for instance Fortin [9], Hecht [13] or Griffiths [12])

For the $Q_1^+ - P_0$ element of Example 2.2 a 2×2 or 2×2×2 set is enough to yield the result. We may thus think that $Q_1 - P_0$ approximation are more rigid and this will need special precautions in the following.

3. A Schwarz decomposition preconditioner for a conjugate-gradient method

Let us suppose a finite element mesh on domain Ω and let us part this mesh into overlapping blocks like in Figure 3.1.



... Domain 1
... Domain 2

Figure 3.1: OVERLAPPING DOMAINS

The idea is now very simple and natural: we sweep the blocks in some order and we solve our Stokes (or Navier-Stokes) problem on each block using as boundary conditions, the real ones when the boundary of the block meets the true boundary, or on artificial interval boundaries, Dirichlet condition taken from the most recent update of the solution (that is à la Gauss-Seidel). On each block we have a small enough problem and we can use our favorite solution method for a Stokes problem that is an augmented lagrangian method.

For a Stokes problem this means the simple iterations where, p^0 being given, one computes $\underline{u}^n$ and p^{n+1} by

$$(3.1)\qquad \begin{cases} \nu\int_\Omega \underline{\nabla}_h \underline{u}^n \cdot \underline{\nabla}_h \underline{v} \;\; dx + r\int_\Omega (\underline{\nabla}_h \cdot \underline{u}_h^n)(\underline{\nabla}_h \cdot \underline{v}_h) \;\; dx - \int_\Omega p_h^n \; \underline{\nabla}_h \cdot \underline{v}_h \; dx \\ \qquad\qquad = \int_\Omega \underline{f}_h \cdot \underline{v}_h dx \\ -\int_\Omega p_h^{n+1} q_h dx = \int_\Omega p_h^n \; q_h dx + r \int_\Omega u_h^u \; q_h dx \; . \end{cases}$$

It must be noted that the penalty term contains a discrete divergence operator

$$(3.2)\qquad \underline{\nabla}_h \cdot \underline{u}_h = P_{Q_n}(\underline{\nabla} \cdot \underline{u}_h) \; .$$

In order to have (3.1) converging, the mass balance condition must be satisfied on the boundary of the block at least when the boundary conditions of the block are purely Dirichlet. This will pose no problem provided the solution is initialized divergence-free and kept so throughout. In some cases this will mean that a special initialization procedure will have to be employed.

But first let us consider the convergence properties of this method. If we think of it as a relaxation process, it is clear that convergence will take place provided every member of some basis of V_0 is implied at least once, for every cycle, in the sweeping process. This is where knowledge of the basis of V_0 becomes important: blocks should be large enough and their overlopping wide enough to ensure this condition. For a $Q_1 - P_0$ we need, for instance $4\times4\times k$, $k \geq 4$, elements in each block and overlopping should be two element wide. For the $Q_1 - P_0$ element $2\times2\times k$, $k \geq 2$, will be sufficient. Moreover care must be taken of circulation around an immersed body: one block must surround each of them. Periodicity conditions in cascade flows (cf. Fortin-Fortin-Tanguy [8]) also impose a special condition: at least one block should be periodic.

The above conditions are enough for the convergence of velocity. Pressure is determined only up to an additive constant on each block and will not converge unless some further consideration is taken into account. We manage during the iteration to match pressure on one element between overlapping blocks. This can be done by adjusting properly the additive constant. This adjustment should be done starting from blocks which have a natural boundary condition (free outlet for instance) on some part of their boundary and on which pressure is fully determined.

It is now straightfoward to marry the above procedure with a standard conjugate gradient algorithm. To simplify the notations, we introduce the discrete operators A and B defined by,

$$(3.3)\qquad \langle Au_h, v_h \rangle = \alpha \int_\Omega u_h \cdot v_h dx + \nu \int_\Omega \underline{\nabla} \; \underline{u}_h : \underline{\nabla} \; \underline{v}_h dx \; ,$$

$$(3.4)\qquad \langle Bv_h, q_h\rangle = -\int_\Omega q_h \nabla\cdot v_h dx \ .$$

We also define the vector F by

$$(3.5)\qquad \langle F, v_h\rangle = \int_\Omega \underline{f}\cdot\underline{v}_h dx \ .$$

This allows us to write the discrete problem in the more compact form:

$$(3.6)\qquad \begin{cases} A\underline{u}_h + B^t p_h = F \ , \\ B\underline{u}_h = 0 \ . \end{cases}$$

Let us denote S^{-1} the operator associated with one application of a back and forth sweep over all blocks under the above described procedure. S^{-1} is clearly an approximation of A^{-1}. In the following we shall drop subscripts h, being understood that we now consider a discrete problem. We can now describe the algorithm.

Algorithm 1: Preconditioned Conjugate gradient Algorithm.

Let $\underline{u}^0$ be given satisfying the incompressibility condition $\nabla\cdot\underline{u} = 0$.

(3.7) $\underline{u}^n$ being known, $(\nabla\cdot\underline{u}^n = 0)$, compute $\underline{r}^n = A\underline{u}^n - F$,

(3.8) compute $\underline{z}^n = S^{-1}\underline{r}^n$, $(\nabla\cdot\underline{z}^n = 0)$.

(3.9) For $n = 0$, $\underline{\Phi}^n = \underline{z}^n$.

For $n \geq 1$, compute $\beta_n = -\dfrac{(A\underline{z}^n, \underline{\Phi}^{n-1})}{(A\underline{\Phi}^{n-1}, \underline{\Phi}^{n-1})}$,

$$\underline{\Phi}^n = \underline{z}^n + \beta_n \underline{\Phi}^{n-1} \ .$$

(3.10) Compute $\alpha_n = \dfrac{(\underline{r}^n, \underline{\Phi}^n)}{(A\underline{\Phi}^n, \underline{\Phi}^n)}$,

$$\underline{u}^{n+1} = \underline{u}^n - \alpha_n \underline{\Phi}^n \ .$$

(3.11) Test convergence and go back to (3.7) if necessary. □

Remark 3.1. It must be noted that in the linear case (Stokes problem), formula for β_n in (3.9) can be simplified in the usual way. Extension to the non-linear case requires this form of the coefficient. □

Remark 3.2: Although r_n can theoretically be computed as in (3.7), it is better for the purpose of numerical stability to compu- $\underline{r}^n = A\underline{u}^n + B\, p^n - \underline{f}$ whenever an estimate of pressure is available as it is the case in the domain decomposition method described above. □

The only non-standard feature of this conjugate-gradient algorithm is the fact of working in a subspace. The difficulty is overcome by the construction of the preconditioning operator. We have the usual orthogonality relations and a direct application of classical results shows that

$$(3.12)\quad \underline{u}^n \to \underline{u}\ ,\quad r^n \to B^t p\ ,$$

where $\underline{u}$ and p is the solution of the problem.

Before presenting numerical results, we shall see how this algorithm can be extended to the non linear case.

4. Extension to the non linear case

Having built a Stokes solver, it is a strong temptation to try using it in the solution of the full Navier-Stokes case. Indeed the domain decomposition method itself can easily be modified by changing the solution method inside each subdomain. We used in practice a variant of Newton-Raphson's method, described in A. Fortin M. Fortin [7], including a modified Uzawa's algorithm. For moderately large values of the Reynolds number $Re = \frac{Ud}{\nu}$, the sweeping process still converges (although a proof is lacking...).

A variant has also been considered, employing a quasi-Newton method by taking (on each subdomain) a fixed value of the tangent operator $A'(\underline{u})$ defined below. Convergence of this procedure require that a good enough estimate of $\underline{u}$ is used and some updates of $A'(\underline{u})$ must be done from time to time.

The next step is extending to the non-linear case the conjugate-gradient method of the previous section. Let us first define

$$(4.1)\quad \langle A'(\underline{u}_h)\underline{v}_h,\ \underline{w}_h\rangle = \alpha\int_\Omega \underline{v}_h\cdot\underline{w}_h dx + \nu\int_\Omega \underline{\nabla}\,\underline{v}_h :\ \underline{\nabla}\, w_h dx$$
$$+ \int_\Omega \underline{u}_h\cdot\underline{\nabla}\,\underline{v}_h\cdot\underline{w}_h dx + \int_\Omega \underline{v}_h\cdot\underline{\nabla}\,\underline{u}_h\cdot\underline{w}_h dx\ .$$

We suppose that the domain decomposition method provides an approximate solution $z^n = S^{-1}(\underline{u})r^n$ of the problem:

$$(4.2)\qquad \begin{cases} A'(\underline{u}^n)\underline{z}^n + B^t\Pi^n = \underline{r}^n \ , \\ B\underline{z}^n = 0 \end{cases}$$

where $\langle \underline{r}^n, \underline{v}\rangle$ is now defined by,

$$(4.3)\qquad \langle \underline{r}^n, \underline{v}\rangle = \alpha \int_\Omega \underline{u}^n \cdot \underline{v}\ dx + \gamma \int_\Omega \underline{\nabla}\, \underline{u}^n : \underline{\nabla}\, \underline{v}\ dx +$$

$$+ \int_\Omega \underline{u}^n \cdot \underline{\nabla}\, \underline{u}^n \cdot v\ dx = \langle A(\underline{u})\underline{u}, v\rangle$$

In fact, problem (4.2) is solved on each subdomain when a Newton-Raphson procedure is used. When the quasi-Newton method is prefered, $A'(u^n)$ is repaced by $A'(\underline{u}^o)$ and one needs that $\underline{u}^o$ be close enough to the solution. The sweeping process then yields an approximate solution of the global problem.

We can now describe the extended conjugate-gradient method. Let us set,

$$(4.4)\qquad \underline{u}^{n+1} = \underline{u}^n - \rho_n \underline{z}^n$$

and let us approximate r^{n+1} by a first order development, writing:

$$(4.5)\qquad \underline{r}^{n+1} = \underline{r}^n - \rho_n A'(\underline{u}\ )^n \underline{z}^{\ n}\ .$$

We can now request $\underline{r}^{n+1}$ to be orthogonal to $\underline{z}^n$. Using the divergence-free condition we have $\langle B^t\Pi^n, \underline{z}^n\rangle = \langle \Pi^n, B\underline{z}^n\rangle = 0$, so that we can express ρ_n in the formula:

$$(4.6)\qquad \rho_n = \frac{(\underline{r}^n, \underline{z}^n)}{(A'(\underline{u}\ )^n_{\underline{z}}\ ,^n_{\underline{z}}\ )^n} = \frac{(S\underline{z}^n, \underline{z}^n)}{(A'(\underline{u}\ )^n_{\underline{z}}\ ,^n_{\underline{z}}\ )^n}.$$

In order to obtain a positive value of ρ_n, one sees that a sufficient condition is the coercivity of both $A'(\underline{u})$ and $S(\underline{u})$ that is,

$$(4.7)\qquad \langle A'(\underline{u})v, v\rangle \geq \alpha ||v||_1^2\ ,$$

$$(4.8)\qquad \langle S(\underline{u}), v, v\rangle \geq \tilde{\alpha} ||v||_1^2\ .$$

Condition (4.7) holds for small enough Reynolds numbers. (cf. Temam [15] for instance). Condition (4.8) will depend on how well $S(\underline{u})$ is built. In practice the domain decomposition method satisfied (4.8) in almost the same range of Reynolds numbers as that required for (4.7).

Up to now, we have a "gradient" method. If we want to introduce some conjugation we set for $n \geq 1$.

$$\underline{\Phi}^n = \underline{z}^n + \beta_n \underline{\Phi}^{n-1} \tag{4.9}$$

and require that $\underline{r}^{n+1}$ be orthogonal to $\underline{\Phi}^{n-1}$ as well as to $\underline{\Phi}^n$.

This yields the formulas

$$\beta_n = \frac{(A'(\underline{u}^n)\underline{z}^n, \underline{\Phi}^{n-1})}{(A'(\underline{u}^n)\underline{\Phi}^{n-1}, \underline{\Phi}^{n-1})} \quad , \quad \rho_n = \frac{(\underline{r}^n, \underline{\Phi}^n)}{(A'(\underline{u}^n)\underline{\Phi}^n, \underline{\Phi}^n)} \, . \tag{4.10}$$

Formally, our algorithm is therefore exactly the same as the one we described in the previous section. In fact the same program can be used. An important difference is that now no recurrence relations hold. We cannot expect now the superlinear convergence of the conjugate-gradient method as in the case of a symmetric linear problem. Convergence results have been obtained by R. Aboulaich [1]. It would be too long and technical to present them here. Basically convergence will take place whenever (4.7) and (4.8) hold, $S(\underline{u}^0)$ is "close enough" to $A'(\underline{u}^0)$, $\underline{u}^0$ close enough to $\underline{u}$ and $\underline{u}$ is a regular solution of the equations. In fact this result needs as a prerequisite convergence of the Newton-Raphson algorithm to $\underline{u}$.

5. Numerical results

We have tested the decomposition domain technique in both 2-D and 3-D situations. It is only in this last case that we expect an advantage with respect to a direct factorization method. To make this advantage apparent let us recall a few facts about storage requirements and number of operations requested by both methods. To make things simple let us consider a N×N×N square cavity solved either a Cholesky's factorization or a conjugate-gradient method. A direct count shows that for N large, a factorization method will required (N^7) operations with $O(N^5)$ words of storage. Figure 5.1 and 5.2 compare actual results of a Cholesky's factorization with those of our domain decomposition method applied with $Q_1^1 - P_0$ elements and 2×2×N blocks (tubes). Storage results are clear; domain decomposition is $O(N^4)$ and permits to use $N = 10$ on our SUN workstation while Cholesky fails for $N \geq 7$. As to operations the domain decomposition method is $O(N^5)$ while the direct method starts $O(N^5)$ and becomes $O(N^7)$ for larger N . Extrapolation shows that for $N \geq 10$ domain decomposition becomes more economical. It must be noted that the stop test for the iterative method was, a variation of velocity in maximum norm smaller than 10^{-7} , which is much smaller than practical requirements. Figure 5.3 compares the C.P.U. time of a decomposition into 2×2×N tubes with a decomposition into 3×3×N. There is a clear gain although this is paid by larger storage requirements (Figure 5.1). Even if this result is encouraging, it opens a few questions. Following Axelsson [3] or Golum-Van Loan [11] for instance, one should expect $O(N^4)$ operations for a properly preconditioned conjugate-gradient method. In the context in which we work

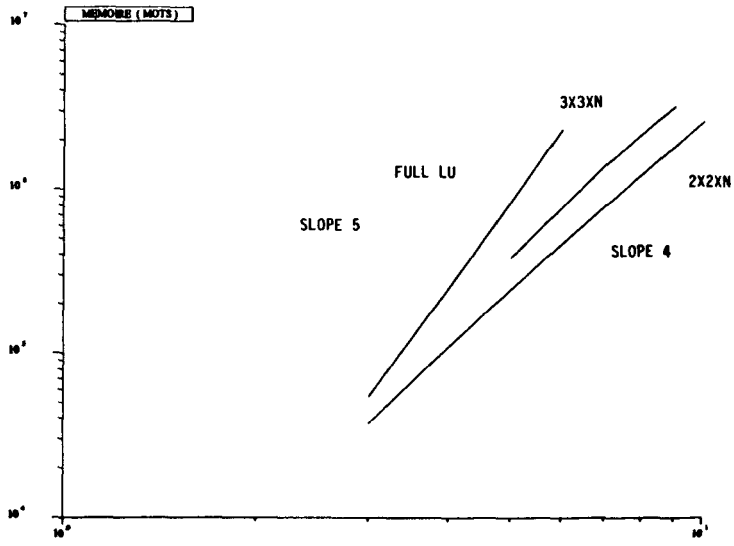


Figure 5.1

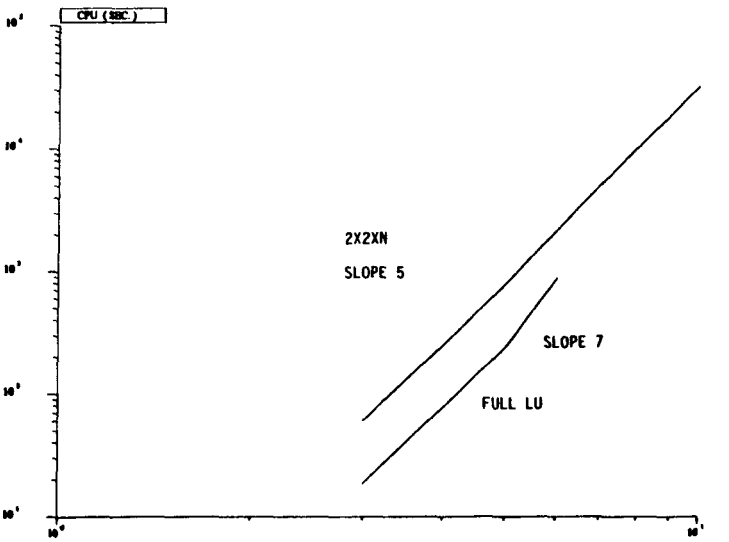


Figure 5.2

that would mean a second order problem with a condition number $K = O(N^2)$ and a number of iterations proportional to $\sqrt{K}$.

The trouble is that what we solve is a Stokes problem which is in reality a fourth order problem: we can see our method as a domain decomposition method for a biharmonic problem. The condition number

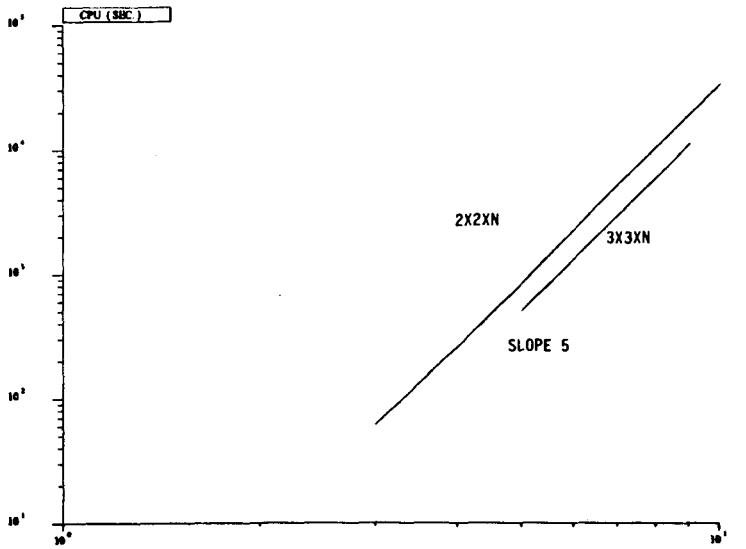


Figure 5.3

should be the $O(N^4)$. Experimentally, we have observed $O(N^3)$ for 2-D problem. The conjugate-gradiant method effectively brings this down to $O(N^{1.3})$.

This is illustrated by Figure 5.4 and 5.5 which count the number of iterations required to solve a given problem to a fixed precision as a fonction of N . Figure 5.4 shows the 2-D result, comparing computation with of 3 and the second one of approximately 1.3 which is quite acceptable. Figure 5.5 compares the result of the conjugate-gradient method based on 2×2×N and 3×3×N tubes. In both case, the slope is approximately 2.

Introducing an overrelaxation factor in the sweeping process has been of no use: the optimal value is 1 or very near to 1. This is again an in compressibility effect. When we attempted to solve a laplacian operator by the same method using an overrelaxation factor provided a strong improvement. Slackning the incompressibility constraint would therefore be a possible source of improvement. However it has a distasteful side effect of distroying positivity of the problem thus making conjugate-gradient much more difficult to handle.

Let us now consider some other points. As expected size of subdomains and their degree of overlapping are important for convergence. Using N×N×2 slices instead of tubes improves convergence but increases storage requirement (but leaving them $O(N^4)$) . As expected larger blocks converge faster. Overlapping increases the rapidity of convergence but also the number of operations by iterations.

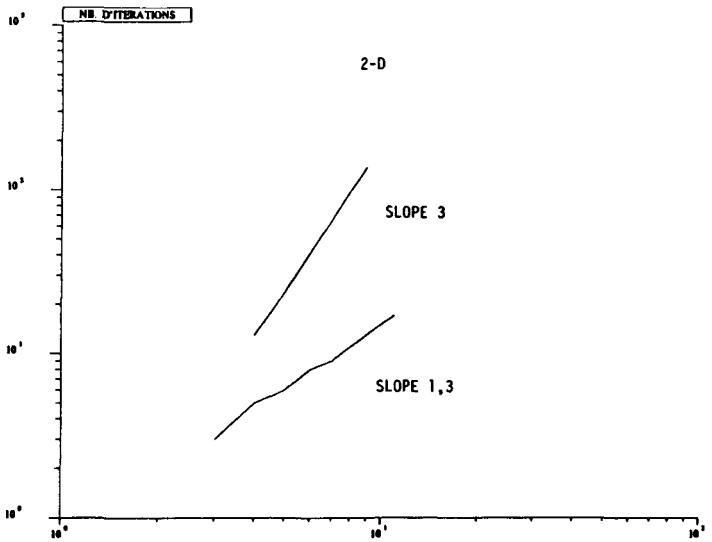


Figure 5.4

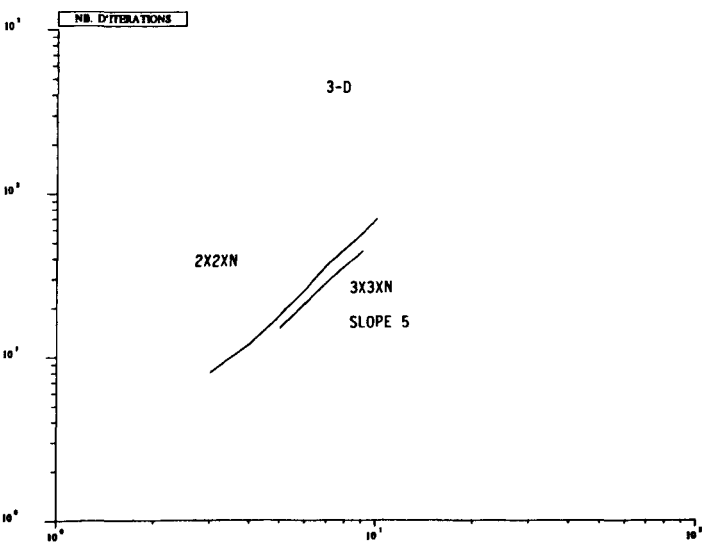
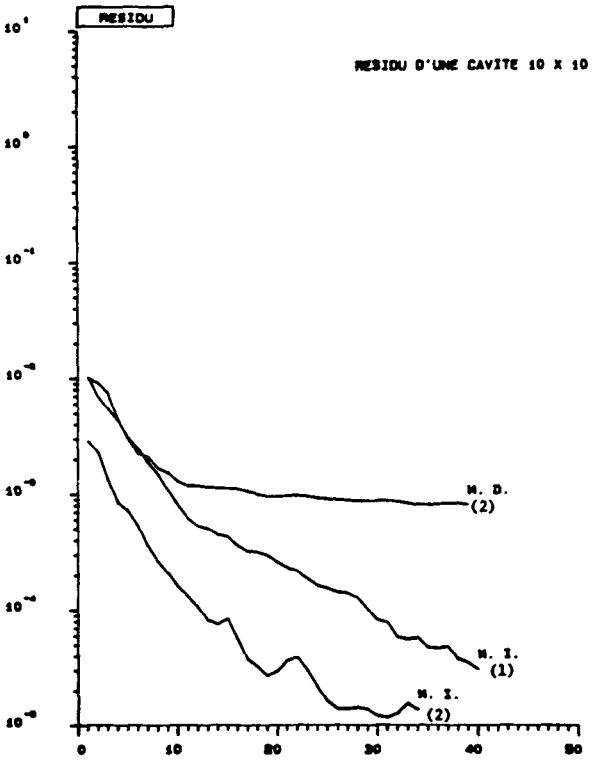


Figure 5.5

Using a symmetric preconditioner by sweeping back and forth is important. A simple sweep destroys the superlinear behavior of the conjugate-gradiant method for orthogonality is lost.

Finally, Figure 5.6 presents a non-linear case illustrating the importance of using a good preconditioner. A direct approach to Reynolds=500 could not converge, while building $S(\underline{u})^0$ from the result for Re=200 converged pretty well.



(1) Re = 200

(2) Re = 500

M.D.: Direct Method

M.I.: Incremental Method

Figure 5.6

6. Possible extensions

As the previous numerical results show, the method is still not as efficient as one would like. To treat large problem we would like to bring down the number of operations to $O(N^4)$. Let us see some possibilities of getting a more efficient procedure.

Non overlapping blocks: An iterative method alternating between Dirichlet and Neumann conditions at the boundary of subdomains has been introduced by A. Quarteroni [14]. An analogue method has been described in Bramble-Pasciak-Schatz [4] and as a preconditioner.

It is not clear how this procedure must be used to a obtain symmetric preconditioning operator. The idea is however appealing for it solves implicitly the problem of initialization in the case of a divergence-free iteration.

Another difficulty lies here in the bad convergence properties of the preconditioner alone. We are considering the possibility of

using it with some degree of overlapping. More classical methods with a multiplies at interfaces can also be worth looking at.

Multilevel methods

One very simple method to build overlapping blocks is to start from a crude mesh then subdividing it into a finer mesh. At the refined level a sub-domain can be defined as the reunion of all elements surrounding a vertex in the coarse qid. (Figure 6.1). The procedure could eventually repeated many times. One could then think of a preconditioning operator by solving the coarse problem then sweeping the subdomains to correct the solution at the level of the refined mesh. Such a preconditioner would be likely to be much more efficient for it would take care of long waves as well as of short waves which are

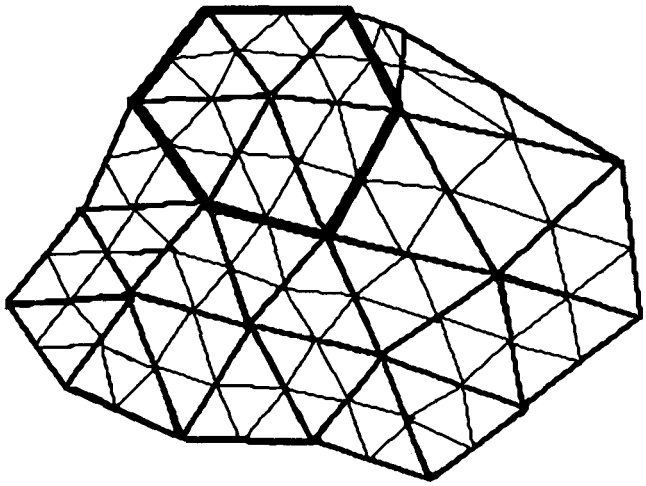


Figure 6.1: Subdivided mesh

advantages by the standard sweeping procedure. The method is under development and results should be available soon.

Finally a last possibility would be to iterate without imposing the divergence-free condition by preconditioning only the "laplacian part" of the Stokes operator. We suggest for this purpose a variant of the Arrow-Hurwicz algorithm.

Algorithm 2: Let u^0 and p^0 be arbitrarily chosen,

(6.1) $\underline{u}^n$ and p^n being known, compute

$$\underline{u}^{n+\frac{1}{2}} = \underline{u}^u - \rho_n S^{-1}(A\underline{u}^n + B^t p - \underline{F})$$

(6.2) $n^{n+\frac{1}{2}}$ being known, compute

$$u^{n+1} = u^{n+\frac{1}{2}} - \alpha_n S^{-1} B^t B u^{n+\frac{1}{2}}$$

$$p^{n+1} = p^n + \alpha_n B u^{n+\frac{1}{2}} .$$

Test convergence and go back to (6.1) if needed. □

ρ_n can be computed by minimizing the lagrangien of (2.4) with respect to $\underline{v}$ while α_n can be computed by minimizing $B\underline{u}^{n+1}$. Convergence of this algorithm will be analyzed in a forthcoming paper. (R. Aboulaich-M. Fortin [2]). It is possible to introduce a conjugation process with respect to the non-positive symmetrie matrix

$$(6.3) \qquad A = \begin{pmatrix} A & B^t \\ B & 0 \end{pmatrix} .$$

Conclusion:

Domain decomposition methods provide an interesting approach for the solution of 3-D flow problems. Work remains to be done to obtain a more efficient preconditioner when the divergence-free condition is present. However, even in the present state of the art, we have obtained a method that is competitive with standard methods and that enable 3-D computations even on relatively small machines.

References

[1] ABOULAICH, R., "Méthodes du gradient conjugué pour la résolution numérique des équations de Navier-Stokes". Thèse, Université Laval, 1986.

[2] ABOULAICH, R., FORTIN, M., "Iterative methods for the solution of Stokes problem", (to appear).

[3] AXELSSON, O.A., BARKER, V.A., "Finite Element Solution of Boundary Value Problem", Academic Press, (1984).

[4] BRAMBLE, J.H., PASCIAK, J.E., SCHATZ, A.H., "An iterative method for elliptic problem on regions partitioned into substructure", Math. of Comp., Vol. 46 (174), pp. 361-369, 1986.

[5] BREZZI, F., FORTIN, M., "Mixed and Hybrid Finite Element Method" (to appear).

[6] DINH, R.V., GLOWINSKI, R., PERIAUX, J., Application of domain decomposition techniques to the numerical solution of the Navier-Stokes equations", Num. Meth. for Eng.1, (1980), pp. 383-404.

[7] FORTIN, A., FORTIN, M., "A generalization of Uzawa's algorithm for the solution of the Navier-Stokes equations", Comm. in Appl. Num. Meth., Vol. 1, 5, pp. 205-208 (1985).

[8] FORTIN, A., FORTIN, M., TANGUY, P., "Numerical Simulation of Viscous flows in Hydraulic Turbomachinery by the finite element Method", Comp. Meth. in Mech. Eng. 58 (1986), pp. 337-358.

[9] FORTIN, M., "Old and new finite elements for incompressible flows", *Int. Jour. Num. Meth. in Fluids*, 1 (1981), pp. 347-364.

[10] GIRAULT, Y., RAVIART, P.A., "Finite Element Methods for Navier-Stokes Equations", Springer-Verlag (1986).

[11] GOLUB, G.E., VAN LOAN, C.F., "Matrix Computations John Hopkins, University Press, Baltimore, Maryland (1983).

[12] GRIFFITHS, D.F., "The construction of approximately divergence-free finite elements', in Mathematics of Finite Elements and Applications, (ed. J.R. Whiteman) Academic Press, New York, 1979.

[13] HECHT, F., "Construction d'une base d'un élément fini non conforme à divergence $\mathbb{R}^3$, thèse de 3ème cycle, Université Pierre et Marie Curie, Paris, 1980.

[14] QUARTERONI, A. (Paper in this book.)

[15] TEMAM, R., Navier-Stokes Equations, North-Holland, Amsterdam, 1978.