CHAPTER 26

Domain Decomposition Methods for the Navier-Stokes Equations

M. Fortin†
R. Aboulaich#
C. Kari‡
F. Pitt‡

Abstract : This paper discusses some implementations of domain decomposition methods for the numerical solution of Stokes and Navier-Stokes problems. They are used as preconditioners for conjugate-gradient-like iterative methods such as the **Generalized minimum residuals** (GMRES) or the **Conjugate Gradient Squared** (CGS) methods.

1.The Navier-Stokes problem.

We rapidly recall here the Navier-Stokes equations for a viscous incompressible flow. In a domain Ω of R^2 or R^3 , we are looking for a velocity field $\underline{u}$ and a pressure field p satisfying the system of partial differential equations,

$$(1.1)\quad -\mu\Delta\underline{u} + \underline{u}\cdot\underline{\underline{\text{grad}}}\,\underline{u} + \text{grad}\, p = \underline{f}\,,$$

$$(1.2)\quad \text{div}\,\underline{u} = 0\,.$$

These must be completed by a suitable set of boundary conditions, for instance Dirichlet conditions,

$$(1.3)\quad \underline{u}_{|\Gamma} = \underline{u}_0\,.$$

Choosing a **stable finite element approximation** for this problem requires some care. It is well known that the approximation of $\underline{u}$ and the approximation of p cannot be chosen independently.(see e.g. Brezzi-Fortin[1] ,Fortin[2] or Girault-Raviart[3]). In the present study, computations were done with a full biquadratic approximation for velocity and a discontinuous piecewise approximation for pressure. Once the choice is done, we obtain a finite-dimensional problem of the form,

†Professeur,département de mathématiques et de statistique,Université,Laval,Québec,G1K7P4,Canada.
#Professeur,Ecole Mohamédia d'ingénieurs,Rabat,Maroc
††Research assistants, département de mathématiques et de satatistique,université Laval

$$(1.4)\quad \begin{pmatrix} A(\underline{u}) & B^t \\ B & 0 \end{pmatrix}\begin{pmatrix} \underline{u} \\ p \end{pmatrix} = \begin{pmatrix} \underline{f} \\ g \end{pmatrix},$$

where $\underline{f}$ and g depend on data, $A(\underline{u})$ is a discrete form of $-\mu\Delta\underline{u} + \underline{u}\cdot\mathrm{grad}\,\underline{u}$ and B is the discrete divergence (B^t thus being a discrete gradient). In the case of the linearized Stokes problem, in which non-linear terms are neglected, (1.4) is the optimality condition of the saddle-point problem,

$$(1.5)\quad \inf_{v}\sup_{q}\ \frac{1}{2}\langle Av,v\rangle - \langle q,Bv\rangle - \langle f,v\rangle + \langle g,q\rangle$$

and is therefore indefinite.

A very classical method for the solution of (1.4) is Newton's method :

$$\begin{cases} \text{-Let } \underline{u}_0 \text{ be arbitrarily chosen,} \\ \text{-Given } \underline{u}_n \text{, find } \delta\underline{u} \text{ solution of} \\ (1.6)\quad \begin{pmatrix} A'(\underline{u}_n) & B^t \\ B & 0 \end{pmatrix}\begin{pmatrix} \delta\underline{u} \\ \delta p \end{pmatrix} = \begin{pmatrix} A(\underline{u}) & B^t \\ B & 0 \end{pmatrix}\begin{pmatrix} \underline{u} \\ p \end{pmatrix} - \begin{pmatrix} f \\ g \end{pmatrix} \\ \text{-Update } \underline{u} \text{ and p by} \\ (1.7)\quad \begin{cases} \underline{u}_{n+1} = \underline{u}_n - \delta\underline{u}\,. \\ p_{n+1} = p_n - \delta p \end{cases} \end{cases}$$

Such a method therefore reduces the solution of our non-linear system to a sequence of large scale non-symmetric, ($A'(\underline{u}_n)$ is not in general symmetric), indefinite problems. As we want to attack three-dimensional problems, we are quite naturally led to look for some iterative method able of handling such a system.

2.A domain-decomposition-based iterative method.

Iterative methods for the solution of elliptic problems are now known to be very effective in three-dimensional problems. For symmetric problems, in particular, preconditioned conjugate-gradient methods are widely used. For non-symmetric problems minimum residuals methods, like the GMRES method of Saad and Schultz[4], have been very popular in the last few years. They have been, in particular quite successful in aerodynamics problems. (Boivin[5], Dutto[6]). A very important feature of conjugate gradient or minimum residuals methods such as GMRES is that, when used to solve subproblems of the form (1.6) in Newton's method, they do not require the computation of the matrix $A'(\underline{u})$. Indeed they only need the computation of expressions such as $A'(\underline{u})w$, where w is some "descent direction". This can be approximated by,

$$(2.1)\quad \frac{A(\underline{u}+\lambda w)(\underline{u}+\lambda w) - A(\underline{u})\underline{u}}{\lambda}$$

for λ small enough. Such an approximation permits to use quasi-Newton methods in situations where the jacobian matrix is difficult to compute, This is hardly the case for Navier-Stokes equations but this remark becomes essential in more complex situations such as viscoelastic flows.(Fortin-Fortin[7]).

Most iterative methods are however suited to the solution of **positive definite problems.** Incompressible Stokes and Navier-Stokes problems,which we shall consider

in this paper, lead to indefinite systems which are moreover not symmetric . Even for a minimum residuals method such as GMRES, positivity is important. In the indefinite case, our experience has shown that the **generic situation** is a stagnation of the residual at a non-zero value, while the descent direction becomes singular that is, denoting $\boldsymbol{K}$ the matrix of the system,

$$(2.1)\quad \langle \boldsymbol{K} w_n, w_n \rangle = 0.$$

This leads the algorithm to fail but this failure can be corrected by a better preconditioning: convergence depends on the positivity of eigenvalues of $S^{-1}\boldsymbol{A}$.

We shall describe in this paper a preconditioner based on domain decomposition and adapted to the solution of Navier-Stokes equations.The starting point for our method will be Arrow-Hurwicz's method for saddle-point problems, (see e.g. Fortin-glowinski[8]). Applied to a linearised problem like (1.6) the continuous version of the method would be, $\underline{u}_0$ being given,

$$(2.2)\quad \begin{cases} \dfrac{\partial S\underline{u}}{\partial t} + A'(\underline{u}_0)\underline{u} + B^t p - \underline{f} = 0, \\ \dfrac{\partial p}{\partial t} - kB\underline{u} = 0. \end{cases}$$

for some well-chosen operator S and some parameter k. Although other points of views could be adopted, an interesting way of describing our iterative process is to apply a domain decomposition procedure to a discrete version of (2.2). To consider the simplest possible case, suppose that domain Ω is split into two overlapping subdomains Ω_1 and Ω_2.

We shall denote $\{\underline{u}_1^n, p_1^n\}$ and $\{\underline{u}_2^n, p_2^n\}$ the solution at time t_n on Ω_1 and Ω_2 respectively. One possible algorithm is then to solve in sequence, $\underline{u}_n$ and p_n being known,

$$(2.3)\quad \begin{cases} \underline{u}_1^n = \underline{u}^n \big|_{\Omega_1},\ p_1^n = p^n \big|_{\Omega_1} \\ S_1\underline{u}_1^{n+1/2} = S_1\underline{u}_1^n - \rho(A'(\underline{u}_0)\underline{u}_1^n + B^t p_1^n - \underline{f}) \\ p_1^{n+1/2} = p_1^n + \rho k\, B\underline{u}_1^{n+1/2} \end{cases}$$

in Ω_1, with if boundary conditions are needed $\underline{u}_1^{n+1/2} = \underline{u}^n$ on $\partial\Omega_1$. Then we set,

$$(2.4)\quad \begin{cases} p^{n+1/2} = \begin{cases} = p^n \text{ on } \Omega_2 \setminus \Omega_1 \\ = p_1^{n+1/2} \text{ on } \Omega_1 \end{cases} \\ \underline{u}^{n+1/2} = \begin{cases} = \underline{u}^n \text{ on } \Omega_2 \setminus \Omega_1 \\ = \underline{u}_1^{n+1/2} \text{ on } \Omega_1 \end{cases} \end{cases}$$

and then,

$$(2.5)\quad \begin{cases} \underline{u}_2^{n+1/2} = \underline{u}^{n+1/2} \big|_{\Omega_2},\ p_2^{n+1/2} = p^{n+1/2} \big|_{\Omega_2} \\ S_2\underline{u}_2^{n+1} = S_2\underline{u}_2^{n+1/2} - \rho(A'(\underline{u}_0)\underline{u}_2^{n+1/2} + B^t p_2^{n+1/2} - \underline{f}) \\ p_2^{n+1} = p_2^{n+1/2} + \rho k\, B\underline{u}_2^{n+1} \end{cases}$$

in Ω_1, with if boundary conditions are needed $\underline{u}_2^{n+1} = \underline{u}^{n+1/2}$ on $\partial\Omega_1$. Finally we set,

$$(2.6)\begin{cases} p^{n+1} = \begin{cases} p^{n+1/2} \text{ on } \Omega_1 \setminus \Omega_2 \\ p_2^{n+1} \text{ on } \Omega_2 \end{cases} \\ \underline{u}^{n+1} = \begin{cases} \underline{u}^{n+1/2} \text{ on } \Omega_1 \setminus \Omega_2 \\ \underline{u}_2^{n+1} \text{ on } \Omega_2 \end{cases} \end{cases}$$

As to the choice of S_1 and S_2, it is quite natural in the context of domain decomposition methods to take,

$$(2.5)\quad S_i = A'(\underline{u}_0)\big|_{\Omega_i},$$

the restriction of the jacobian matrix to Ω_i. In that case, it is also natural to take $\rho = 1$. When one works inside a global Newton's procedure, it is often convenient not to update matrices S_i at every step but to keep them fixed for a few or all iterations. We shall discuss this later when presenting numerical results. In the case of a linear Stokes problem, the choice (2.5) with $\rho = 1$ reduces the computation of $\underline{u}_i^{n+1}$ to

$$(2.6)\quad A_i\underline{u}_i^{n+1} + B^t p_2^n - \underline{f} = 0$$

and we fall back on a method which we already discussed in previous papers (Aboulaich-Fortin[9]). This method reduces to the classical Uzawa's algorithm in the case of a single (sub)-domain.

It is also easy to generalise (2.3)-(2.4) to an arbitrary number of subdomains. The rule might be described rapidly as: **on each subdomain, make one step of Arrow-Hurwicz's method, starting from the last known values.** In our actual implementation, we prefered to use many small subdomains. This was done in view of storage considerations for large-scale problems but is obviously not the only possibility. The idea can also be adapted to other global procedures: parallel computations on subdomains, additive Schwarz's method etc.

As we shall see in numerical results, the method is fairly robust and converges even for rather small values of the viscosity parameter in the Navier-Stokes Equations. No proof is presently known, even in the linear case. We hope that recent progress in domain decomposition methods applied to time-dependent problems might lead to new insights in building such a proof.

3.Conjugate-gradient methods.

Although the domain decomposition procedure just described can work as "standalone" method, our aim was to employ it as a preconditioner for conjugate-gradient-like methods, defining a search direction by

$$(3.1)\begin{cases} w_{\underline{u}} = \underline{u}^{n+1} - \underline{u}^n, \\ w_p = p^{n+1} - p^n. \end{cases}$$

Doing so, we hoped that the combination would yield a more efficient algorithm as, for instance when over-relaxatioon is used as a preconditioner for a conjugate -gradient method in the solution of a discrete elliptic problem (Axelsson and Barker[10]). From our previous

experience, we chose to try two methods, namely GMRES and the conjugate gradient squared (CGS) method of Sonneveld[11] both being, a priori, suited to the solution of non-symmetric problems.

In the standard conjugate-gradient method, the search direction is made orthogonal to all the previous ones through recurrence relations. This is also the case in CGS which is based on the bi-conjugate-gradient method where a symmetric system is handled, associated with the matrix,

$$(3.2)\quad \begin{pmatrix} 0 & K^t \\ K & 0 \end{pmatrix}.$$

The method requires the storage of six vectors.

In GMRES on the contrary, one keeps orthogonality to a given number of previous directions by a Gram-Schmidt procedure. For storage reasons this number of orthogonal directions cannot be taken very large. In a typical three-dimensional problem, ten is already prohibitive but is very small with respect to the number of variables. It is therefore very important, when using such a procedure that the iterative process should not rely on orthogonality properties but should converge for any number of orthogonal directions (including one).

It must be noted that for the linear Stokes problem, which is symmetric, a standard conjugate-gradient method will keep orthogonality through recurrence relations, and that convergence is much easier to achieve, despite the non-definiteness of the matrix.

We shall not discuss here the precise details of the GMRES algorithm which is now classical and for which we refer to Saad and Schulz[4] or subsequent papers of these authors.We would just like to make here a few remarks about the actual implementation that was done in our numerical tests .

The first remark concerns the management of orthogonal vectors. Two possible ways (at least) can be imagined to keep the last direction orthogonal to some number (say k) of previous vectors. The first is to reinitialise the process every time the number of permitted vectors is attained. The second is to keep orthogonality to the last k vectors, discarding the oldest ones from the stack. A comparison will be made in the section about numerical results.

The second remark concerns preconditioning and the various ways in which it can be done. In the present study, preconditioning was done by changing the system at hand $\mathbf{Kx} = \mathbf{b}$, into $\mathbf{S}^{-1}\mathbf{Kx} = \mathbf{S}^{-1}\mathbf{b}$, where $\mathbf{S}$ is some approximation of $\mathbf{K}$, defined implicitly in our case by the domain decomposition procedure. This is known as left preconditioning and there is obviously a right counterpart. If an approximate factorisation of $\mathbf{K}$ were known, that is $\mathbf{S}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}$, one could use a "balanced" preconditioning, changing the system into $\mathbf{L}^{-1}\mathbf{K}\mathbf{U}^{-1}\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}$. Apparently, these various ways can change rather strongly the convergence behaviour of preconditioned conjugate-gradient method. This is understandable as the scalar product in which orthogonality is kept is changed for instance from $\mathbf{S}^{-t}\mathbf{S}^{-1}$ to $\mathbf{L}^{-t}\mathbf{L}^{-1}$.

4.Numerical results.

We can only present in this short paper a few results illustrating the performance of the algorithm that we propose. The figures show two or three curves describing the convergence of the method. They respectively correspond to a norm of the residual of the non-linear problem, a norm of the residual of the linearized problem and to a norm of the divergence. All the tests have been done on the square cavity problem using regular meshes. This is a relatively easy problem and we were able to reach fairly high Reynolds numbers.

Figure 1 presents the convergence of algorithm (2.3)-(2.6) without any conjugate gradient acceleration. The test consisted of computing a solution at Re=2000 from a solution at Re=1000 on a 10x10 mesh. The preconditioning operator was defined as in (2.7) and was updated at every tenth iteration. One sees the linear convergence which is to be expected. The update of the preconditioner is essential for convergence.

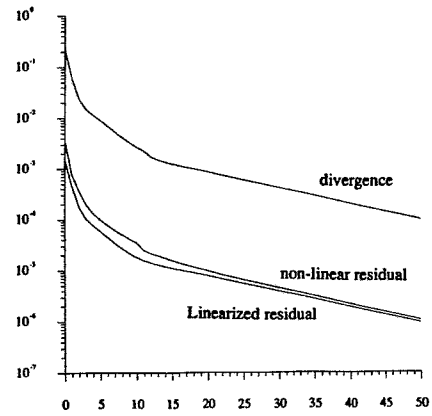

Figure 1: Convergence of algorithm (2.3)-(2.6)

The next two Figures present the behaviour of GMRES using the domain decomposition method as a preconditioner. The problem was a 20x20 cavity at Re=200 starting from a null solution. The subdomains were simple 2x20 overlapping stripes. The algorithm is in fact a quasi-Newton method in which the Jacobian matrix is approximated from (2.1). One therefore has an inner GMRES iteration on the linearized problem imbedded into an outer Newton iteration. The inner iteration was done in order to reduce the residual by a fixed factor of 200. The number of descent directions retained was 10 and GMRES was restarted till the criterion on the residual was met. The preconditioner was updated at every Newton step and was therefore a good approximation of the linearized problem being solved at this step. The difference between Figures 2 and 3 lies in the application of the domain decomposition preconditioning. In figure 2, this was done by one sweep through the subdomains while for Figure 3 two sweeps were done, thus yielding a better preconditioning. This is a general method to get a better convergence with such a method. One can see a direct improvement of convergence with respect to the number of iterations although not with respect to computer time.

However the possibility of controling the quality of the preconditioner is important as can be seen in Figure 4 . This is the same kind of test as the previous ones, this time on a 10x10 problem. The intersting feature is here the stagnation of the inner GMRES iteration at the first Newton step. This iteration was in fact stopped by reaching the maximum

number of allowed steps. This behaviour is typical of GMRES on an indefinite problem when the preconditioner is not good enough. Indeed updating the preconditioner for the second Newton step restored the convergence of the inner loop and made the whole process to converge rapidly and superlinearly. There are of course many ways of improving the quality of a domain decomposition preconditioner through the construction of subdomains and their degree of overlap. An actual implementation must keep a balance between storage and computer time requirements.

As to the CGS algorithm, we do not present curves but we can state that its performance was very disappointing on our problem. Convergence could be obtained only with very good preconditioning (e.g. 5 sweeps per iteration). It was also much more sensitive to the Reynolds number.

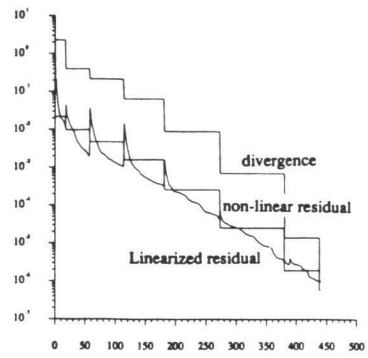

Figure 2: Convergence of GMRES (one sweep preconditioner)

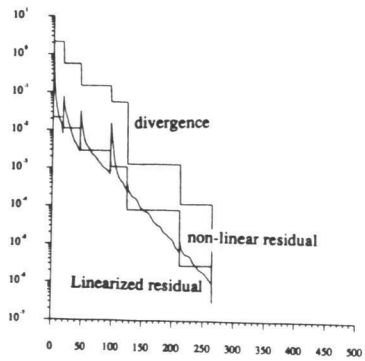

Figure 3: Convergence of GMRES (two sweeps preconditioner)

5. Conclusions

The combination of GMRES with a domain decomposition preconditioner was shown to be effective for the solution of incompressible Navier-Stokes problems. The

domain decomposition method can also be used as a standalone iterative process and is quite robust with respect to Reynolds number .

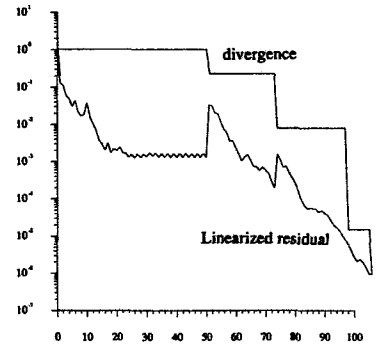

Figure 4: Effect of insufficient preconditioning on GMRES .

6.References

1. F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer Verlag, New-York, 1991.
2. M. Fortin , *Old and New Finite Elements for Incompressible Flows*, Int.J.Num. Meth.Fluids , 1, 347-364, 1981.
3. V.Girault, P.A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, 1986.
4. Y.Saad M.H.Schultz, *GMRES: a generalized minimum residual algorithm for solving non symmetric linear systems*, SIAM J. Sc. Stat. Comput. , 17, 856-869, 1986.
5. S.Boivin , *Simulations d'écoulements compressibles à nombre de Reynolds élevé*, Thèse, Université Laval, Québec, Canada, 1990.
6. L. Dutto , *Etude de préconditionnements pour la résolution par la méthode des éléments finis des équations de Navier-Stokes pour un fluide compressible*, Thèse, Université Pierre et Marie Curie, Paris, 1990.
7. A.Fortin, M.Fortin , *A preconditioned generalized minimum residual algorithm for the numerical solution of viscoelastic fluid flows*,J. Non Newt. Fl. Mech., 36, 277-288,1990.
8. M. Fortin, R. Glowinski, *Augmented Lagrangian Methods* , North-Holland, 1983.
9. M. Fortin, R. Aboulaich, *Schwarz's Decomposition Method For Incompressible Flow Problems*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, (Glowinski, Golub, Meurant, Périaux , eds) , SIAM, Philadelphia, 1988.
10. O.A. Axelsson, V.A. Barker, *Finite Element Solution of Boundary Value Problems* , Academic Press, 1984.
11. P. Sonneveld, *CGS: a fast Lanczos-type solver for nonsymmetric linear systems*, SIAM J. Sc. Stat. Comput., 10, 36-52, 1989.