Computers Math. Applic. Vol. 25, No. 9, pp. 51–58, 1993
Printed in Great Britain.

0097-4943/93 $6.00 + 0.00

A PARALLEL ALGORITHM FOR SOLVING THE INCOMPRESSIBLE NAVIER-STOKES PROBLEM

J. L. CRUZ SOTO, M. C. CALZADA CANALEJO, M. MARÍN BELTRÁN
Department of Mathematics, University of Córdoba
Avda San Alberto Magno s/n, E-14004 Córdoba, Spain
E. FERNÁNDEZ-CARA
Department of Mathematical Analysis
University of Seville, Tarfia s/n, E-41012 Sevilla, Spain

(Received June 1992; accepted July 1992)

Abstract—We introduce and analyze a parallel algorithm for solving the Navier-Stokes equations based on the splitting of the two main difficulties involved, the presence of nonlinear terms and the zero divergence condition. The numerical results obtained by using the proposed algorithm are quite consistent with those furnished by other known algorithms. Numerical results are discussed, as well as the advantages of this new algorithm.

1. INTRODUCTION

The National Science Foundation has identified a series of problems whose solution would have a great economic or scientific impact worldwide. Supercomputational techniques appear to be the sole alternatives to achieving any significant advances in this context. The aforementioned problems, widely known as the Grand Challenges, include the following:

- The behaviour of a moving object in a bulk fluid, either laminar or turbulent; new techniques for the development of aircraft.
- Meteorological forecasts, overall climatological changes, carbon dioxide dynamics, destruction of the ozone layer.
- Turbulence: structural analysis, transition to turbulence and turbulent flow modelling.

The starting point for all the above problems is the Navier-Stokes equations (NSE), which describe the motion of "real" fluids. These are nonlinear partial differential equations (PDE), so their exact solutions are only known for a fairly small number of simple problems. Consequently, their description relies on numerical techniques. The problem is rather complex; the numerical solution of NSE calls for the joint effort of at least numerical analysts and experts in computation sciences.

The high complexity of these problems is reflected in the fact that, in practice, the numerical treatment of all interesting situations is not feasible by using sequential algorithms [1].

In this work, we introduce a parallel algorithm which is based on the splitting of the original problem in such a way that each subproblem is simultaneously executed by different processors. The advantages of this algorithm in terms of efficiency, speed-up, etc. are discussed. Its performance is compared with that of another (fractional step) algorithm whose conditional stability and theoretical convergence have already been proven [2,3].

2. THE MATHEMATICAL MODEL

The motion of an incompressible, homogeneous and viscous fluid with kinematic viscosity ν can be described on the basis of the Navier-Stokes equations:

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u + (u \cdot \nabla) u + \nabla p &= f, \qquad \text{in } \Omega \times (0,T), \\ \nabla \cdot u &= 0, \qquad \text{in } \Omega \times (0,T). \end{aligned} \tag{1}$$

Here, $u = u(x,t)$ is the fluid velocity in $\Omega \times (0,T)$, with Ω being the region it occupies (Ω is a bounded open set of $\Re^N$, with $N = 2$ or 3 in practice); $[0,T]$ is a time interval; $p = p(x,t)$ is the fluid pressure and $f = f(x,t)$ is an external (and known) density function for a force field. These equations have of course to be complemented with appropriate initial and boundary conditions which depend on the nature of the problem concerned.

3. THE WEAK FORMULATION

For simplicity, we will assume that (1) is to be solved with the initial condition

$$u(x,0) = u_0(x), \quad x \in \Omega \tag{2}$$

and the boundary conditions

$$u(x,t) = 0, \quad (x,t) \in \partial\Omega \times (0,T). \tag{3}$$

Now, $u_0 = u_0(x)$ is a given function and $\partial\Omega$ is the boundary of Ω. In the numerical experiments which we present in Section 6, boundary conditions of other kinds are also considered.

In order to obtain a weak formulation for the above problem, we shall introduce the following function spaces:

$$D(\Omega) = \left\{ \frac{\varphi}{\varphi} \in C^\infty(\Omega), \quad \text{sop } \varphi \subset \Omega \right\}, \qquad J(\Omega) = \left\{ \frac{\varphi}{\varphi} \in D(\Omega)^N, \quad \nabla\varphi = 0 \quad \text{in } \Omega \right\}.$$

Here, V is the adherence of $J(\Omega)$ in the Sobolev space $H_0^1(\Omega)^N$. Recall that, endowed of its natural scalar product and norm, which we denote by $((.,.))$ and $\|.\|$, respectively, $H_0^1(\Omega)^N$ (and also V) is a Hilbert space. H is the adherence of $J(\Omega)$ in $L^2(\Omega)^N$. With the scalar product and norm in $L^2(\Omega)^N$, which we shall denote by $(.,.)$ and $|.|$, respectively, H is another Hilbert space. V' is the dual of V. The duality pairing between V' and V will be denoted by $\langle .,. \rangle$. Let us also introduce the trilinear form $b(.,.,.)$, defined by

$$b(u,w,v) = \sum_{i,j} \int_\Omega u_i \frac{\partial w_j}{\partial x_i} v_j \, dx, \qquad \forall u,w,v \in H^1(\Omega)^N.$$

By following the usual procedure, i.e., by multiplying the first equation in (1) by an arbitrary function $w \in V$, integrating in Ω and applying well-known integration formulae, we arrive at the weak formulation of the N-S problem, namely: for given $u_0 \in H$ and $f \in L^2(0,T;V')$, find a function $u \in L^2(0,T;V) \cap L^\infty(0,T;V')$ such that

$$\left\langle \frac{du(t)}{dt}, w \right\rangle + \nu((u(t),w)) + b(u(t),u(t),w) = \langle f(t), w \rangle,$$
$$\forall w \in V, t \text{ a.e. in } (0,T), \quad u(0) = u_0. \tag{4}$$

It is well-known that there exists at least one solution to this problem which is furthermore unique if $N = 2$. In addition, if a solution u is regular enough, then the function $(x,t) \to u(t)(x)$ is such that, for some other (scalar) function p, the couple (u,p) is a classical solution of (1–3). For further details, see e.g., [4].

4. THE NUMERICAL APPROXIMATION

The problem described above involves several major difficulties.

- There is a system of PDE and not a single PDE; the unknowns are the N components of the velocity field and the pressure distribution.
- The unknowns are not uncoupled inasmuch as the different u_i are related by the incompressibility condition $\nabla \cdot u = 0$.
- The problem is nonlinear, owing to the presence of the term $(u \cdot \nabla)u$.

As a consequence, we must not expect to obtain exact solutions but for a few special situations. This has fostered the research of numerical algorithms providing approximate solutions.

Our numerical approximation procedure consists of two major steps: first, we carry out a discretization with respect to the time variable and, then, the resulting (sub)problems are discretized in space.

There are lots of classical sequential schemes that can be used to discretize in time [5]. Among the most frequently used at present, we find so the so called *fractional step methods*. It is now widely recognized that these provide very good numerical results when applied to the NSE [6].

In order to better understand the underlying philosophy of fractional step discretization in time, consider the following model initial-value problem:

$$\frac{du}{dt} + Au = f, \quad t > 0, \quad u(0) = u_0. \tag{5}$$

Here, $u = u(t)$ (scalar or vector) is the unknown, a function that describes the state of the system, and $f = f(t)$ is given. We assume that A (linear or nonlinear) is an operator defined in an appropriate vector space. If the time interval is assumed to be divided into subintervals of amplitude Δt, then the "natural" implicit method for the numerical approximation to a solution u, provided it exists and is unique, is given as follows:

$$\frac{u^{n+1} - u^n}{\Delta t} + Au^{n+1} = f((n+1)\Delta t) = f^{n+1}, \qquad n \geq 0. \tag{6}$$

Once (6) is solved, u^{n+1} is, at least formally, an approximation to the solution u at time $(n+1)\Delta t$.

Let us now assume that A can be splitted in the form

$$A = A_1 + A_2\,,$$

where each $A_i\,(i = 1, 2)$ is a new operator. Also, assume that, for some reason, it is easier to solve (6) if A is replaced by A_i. A fractional step method (of the Peaceman–Rachford type), that can be applied in connection with the above splitting, involves computing u^{n+1} starting from u^n in two steps. Thus, for given $n \geq 0$ and u^n, $u^{n+1/2}$ can be calculated as the solution to

$$\frac{u^{n+1/2} - u^n}{\Delta t/2} + A_1\, u^{n+1/2} + A_2\, u^n = f^{n+1/2}. \tag{7}$$

In a second step, u^{n+1} is computed by solving

$$\frac{u^{n+1} - u^{n+1/2}}{\Delta t/2} + A_1\, u^{n+1/2} + A_2\, u^{n+1} = f^{n+1}. \tag{8}$$

In (7) and (8), we may take (for instance) $f^{n+i/2} = f\left((n + \frac{i}{2})\Delta t\right)$ for $i = 1, 2$.

The proposed algorithm is purely sequential. Starting from a given $u^0 = u_0$, $u^{1/2}$ is calculated from (7). Next, u^1 is obtained from (8) and so on.

Parallelizing the algorithm (7), (8) is fairly easy. For this purpose, we compute u^{n+1} in three rather than two steps. $u^{n+2/3}$ and $u^{n+4/3}$ are calculated simultaneously with the aid of different processors by solving

$$\frac{u^{n+2/3} - u^n}{2\Delta t/3} + A_1\, u^{n+2/3} + A_2\, u^n = f^{n+2/3},$$

$$\frac{u^{n+4/3} - u^n}{4\Delta t/3} + A_1\, u^n + A_2\, u^{n+4/3} = f^{n+4/3}$$

and, then, u^{n+1} is obtained from the formula

$$u^{n+1} = \frac{1}{2}\left(u^{n+2/3} + u^{n+4/3}\right). \tag{9}$$

Obviously, the proposed algorithm can be readily generalized to those cases where A can be written in the form

$$A = A_1 + A_2 + \cdots + A_q.$$

This requires the use of q processors in parallel in a scheme involving $q + 1$ fractional steps.

An obvious major practical question related to the computations is to balance the different subproblems properly, so that each processor takes roughly the same time. This would result in minimal waiting times and maximal efficiency. We will come back to this later.

In this work, we have adapted the above ideas to the numerical solution of the NSE by using a three-step scheme which therefore involves the simultaneous solution of two subproblems. As in [6], the splitting of the "spatial" differential operator (the equivalent of A in (1)) separates the two main difficulties, namely nonlinearity and incompressibility. More precisely, the problems to be solved read as follows:

$$\frac{3}{2\Delta t}(u^{n+2/3} - u^n) - \nu\Delta(\theta u^n + (1-\theta)u^{n+2/3}) + (u^{n+2/3}\cdot\nabla)u^{n+2/3} = f^{n+2/3} + \nabla p^n, \quad (10)$$

(a quasilinear system of PDE (NLP)),

$$\frac{3}{4\Delta t}(u^{n+4/3} - u^n) - \nu\Delta(\theta u^{n+4/3} + (1-\theta)u^n) + \nabla p^{n+4/3} = f^{n+4/3} - (u^n\cdot\nabla)\,u^n, \quad (11)$$

$$\text{div } u^{n+4/3} = 0$$

(a linear problem (LP)). Here, $\theta \in (0,1)$ is a parameter that can be used to increase or decrease the significance of the incompressibility condition and the nonlinear (inertial) terms in each problem. Of course, we search for solutions $u^{n+2/3}$ and $u^{n+4/3}$ of (10) and (11) (respectively) which must satisfy the homogeneous boundary condition (3). In a third step, u^{n+1} is calculated from (9).

5. SPACE DISCRETIZATION

Parallelizing the proposed algorithm entails solving two essentially different problems that are described in detail below. Throughout the discussion that follows, superscripts have been omitted for simplicity.

(a) Linear problems (LP) of the Stokes type[1], which can be written as

$$\alpha_1 u - \nu_1\Delta u + \nabla p = g\,, \qquad \text{div } u = 0 \quad \text{in } \Omega. \quad (12)$$

Here, $\alpha_1 = 3/4\Delta t, \nu_1 = \theta\nu$ and $g \in V'$ is known.

We are thus dealing with linear problems that preserve the incompressibility condition. These problems have a unique solution. The corresponding weak formulation is simply

$$\alpha_1(u,w) + \nu_1((u,w)) = \langle g, w\rangle, \qquad \forall w \in V, \quad u \in V. \quad (13)$$

In order to solve numerically these problems, one must discretize in space by using an appropriate method. Here, we have chosen the finite element technique [7] for this purpose. For simplicity, assume that $N = 2$ and Ω is a polygonal domain. Also, let $\{\mathcal{T}_h/h \in (0,1]\}$ be a regular family of Ω triangulations (as in [8]). We shall denote by $\mathcal{P}_k$ the vector space formed by all polynomial functions in the variables x_i whose degree is less than or equal to k. In order to approximate the incompressibility condition, we must introduce nonconformal finite elements (discontinuous functions). Specifically, we shall use the nonconformal P_1 finite element of Crouzeix and Raviart [9]. The corresponding space of approximation is given by

$$V_h = \left\{ w_h : \overline{\Omega} \to \frac{\Re^2}{w_h}\Big|_T \in \mathcal{P}_1^2, \right.$$

$$\left. w_h \text{is continous at all midpoints in } \mathcal{T}_h,\ \textstyle\int_T \nabla\cdot w_h dx = 0,\ \forall T \in \mathcal{T}_h \right\}.$$

Thus, the finite-dimensional problem to be solved is

$$\alpha_1(u_h, w_h) + \nu_1((u_h, w_h)) = \langle g, w_h\rangle, \qquad \forall w_h \in V_h, \quad u_h \in V_h. \quad (14)$$

[1]Recall that the Stokes problem is an approximation to the Navier-Stokes problem that holds at very large viscosity values at which one arrives by neglecting nonlinear terms.

In practice, this is a linear system, the coefficient matrix of which is symmetrical, positively defined and independent of the time iteration considered. Therefore, it can be factorized (e.g., by using the Cholesky method) at the start of the program. Accordingly, at each iteration one only has to solve triangular systems.

As can be seen, the pressure has disappeared from the variational formulation. Once the velocity field is known, a numerical approximation to p can be computed for instance, according to Crouzeix [9], in the space

$$Q_h = \{q_h : \overline{\Omega} \to \Re \,/\, q_h|_T \in \mathcal{P}_0, \qquad \forall T \in \mathcal{T}_h\}.$$

(b) Non-linear problems (NLP), which can be expressed as follows:

$$\alpha_2 u + \nu_2 \Delta u + (u \cdot \nabla)\, u = g^* \qquad \text{in } \Omega. \tag{15}$$

Now, $\alpha_2 = 3/2\Delta t, \nu_2 = (1-\theta)\nu$ and $g^* \in V'$ is known.

These problems can be solved, the solution being unique provided the time step Δt is small enough (cf. [2]). The weak formulation of (15) is the following:

$$\alpha_2(u,w) + \nu_2((u,w)) + b(u,u,w) = \langle g^*, w\rangle, \qquad \forall w \in H_0^1(\Omega)^2, \quad u \in H_0^1(\Omega)^2, \tag{16}$$

We can now choose the following approximation space:

$$W_h = \left\{w_h : \overline{\Omega} \to \Re^2, w_h \text{ is continous at all midpoints in } \mathcal{T}_h, w_h|_T \in \mathcal{P}_1^2, \qquad \forall T \in \mathcal{T}_h\right\}.$$

Accordingly, the approximated problem is as follows:

$$\alpha_2(u_h,w_h) + \nu_2((u_h,w_h)) + b(u_h,u_h,w_h) = \langle g^*, w_h\rangle, \qquad \forall w_h \in W_h, \quad u_h \in W_h. \tag{17}$$

In practice, since (17) is still nonlinear, it is reformulated in the sense of least squares. For simplicity, we shall deal only with continuous problems (i.e., those which are not discretized in space), even though the procedure can be readily adapted to the corresponding discretized problems. Let us introduce the function J, given as follows:

$$J(w) = \frac{1}{2}\int_\Omega \left[\alpha_2|y|^2 + \nu_2|\nabla y|^2\right] dx, \qquad \forall w \in H_0^1(\Omega)^2.$$

Here, y is the unique solution in $H_0^1(\Omega)^2$ of the linear system

$$\alpha_2 y - \nu_2 \Delta y = \alpha_2 w - \nu_2 \Delta w + (w \cdot \nabla) w - g^*, \qquad \text{in } \Omega. \tag{18}$$

Obviously, a solution u to problem (15) is characterized by the fact that it minimizes J. Hence, a possible strategy is to search for a function $u \in H_0^1(\Omega)^2$ such that $J(u) \leq J(w) \quad \forall w \in H_0^1(\Omega)^2$.

The search of such a minimizer can be achieved, for instance, by using a conjugate gradient algorithm [10]. Again, the task is reduced in practice to solve linear problems involving symmetric, positively defined matrices that are independent of the time iteration and can be factorized at the beginning of the program.

6. SOME NUMERICAL EXPERIMENTS

The above ideas have been implemented in accordance with the basic guidelines of the MODULEF Library [11] by using several computers in a local area network. The various machines were synchronized by using the binary semaphore technique, taking advantage of the convenience of the NFS. We thus used an MIMD computer where the different processors were geographically scattered and only shared the disk memory (MIMD multicomputers) [12,13]. This is worthy of special emphasis because a multiprocessor MIMD computer with shared memory would have provided faster data transfers, which took quite a long time, thereby increasing the overall throughput.

By way of example, we chose a two-dimensional fluid subject to a sudden expansion, as is the case in a squared cavity after a channel. This situation is closely related to the mechanical

behaviour of eddies in turbomachines. Domain Ω is depicted in Figure 1. The boundary of Ω consists of three parts that are associated to the boundary conditions:

$$\partial\Omega = \Gamma_B \cap \Gamma_\infty^- \cap \Gamma_\infty^+,$$

where Γ_B is a wall, and Γ_∞^- and Γ_∞^+ are, respectively, the entering and exit (artificial) boundaries for the fluid.

We have used a triangulation composed of 508 triangles and 1211 nodes (Figure 1), which was created by employing the MODULEF library.

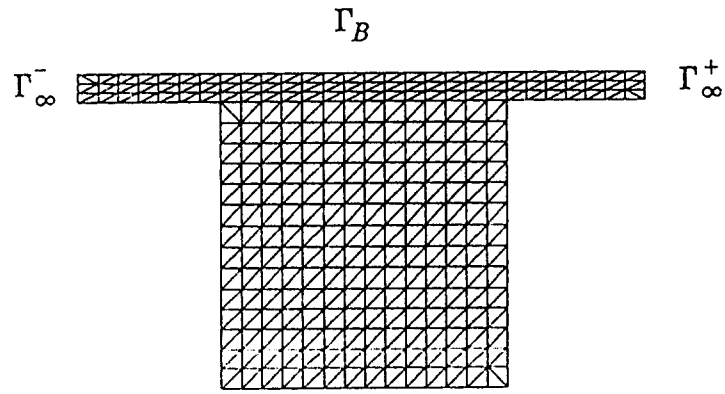


Figure 1.

These are the most significant features of the test example:

- The Reynolds number Re = 100. Here, Re = UL/ν was calculated from the maximum value U of $|u|$ on Γ_∞^-, the channel width (L) and the kinematic viscosity (ν).
- The behaviour of the velocity field on $\partial\Omega$, which can be summarized as follows. There is a Poiseuille profile on Γ_∞^-, the no-slip condition on Γ_B and a natural condition on Γ_∞^+.

The sequencing and coordination of the different tasks on two computers of the local area network are illustrated in Figure 2 as Gantt charts. Non-shaded areas represent waiting times while vertical, dark bars separate mutually exclusive tasks and hence points where the two machines were to be synchronized.

Figures 3 and 4 show the streamlines and pressure isovalues which are obtained after 100 time iterations ($\Delta t = 10^{-2}$) by using the two-step sequential algorithm described in Section 4.

The next two figures, Figures 5 and 6, correspond to the results obtained using the equivalent parallel algorithm. As can be seen, there is good agreement between the two.

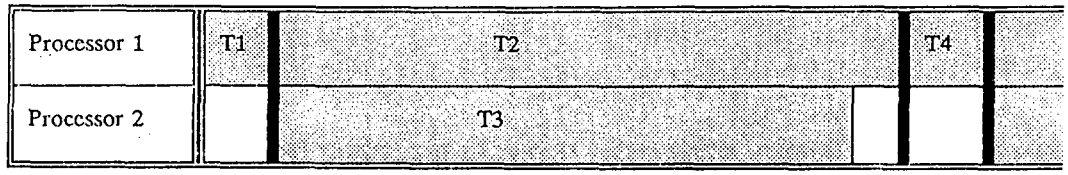
In order to assess the performance and advantages of the proposed algorithm, the same numerical experiment was repeated with different values of Re and the following parameters were calculated:

- Speed-up ratio, given by $S_p = T_s/T_p$, where T_s is the resolution time for the sequential algorithm and T_p is that of the parallel algorithm.
- Efficiency, defined as EFF = T_s/qT_p, with q being the number of processors used.

The results are listed in the following table, where times are given in 1/100 of a second.

Table 1.

Re	T_s	T_p	EFF	S_p
1	117600	78814	0.746	1.492
10	83150	44308	0.938	1.876
100	82791	43994	0.940	1.880
1000	83270	44341	0.939	1.878



T1: data acquisition/recording
T2: linear problems + pressure calculation
T3: nonlinear problems
T4: velocity interpolation (4.5) + transfer

Figure 2.

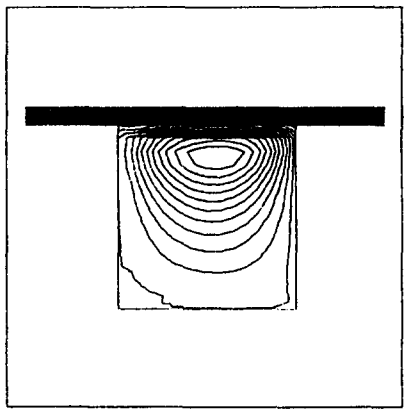


Figure 3.

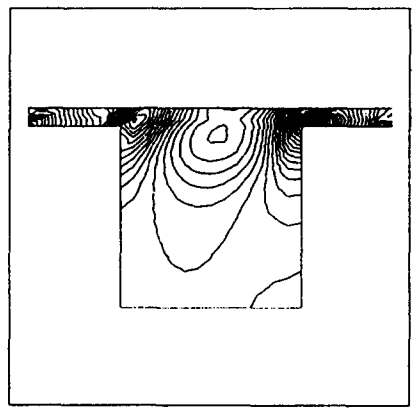


Figure 4.

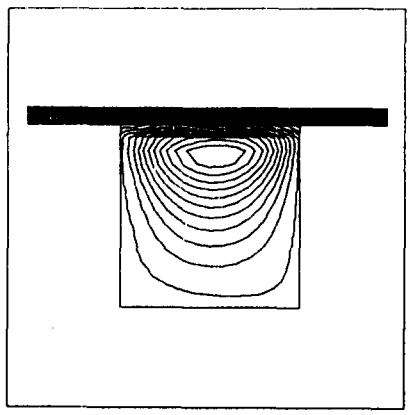


Figure 5.

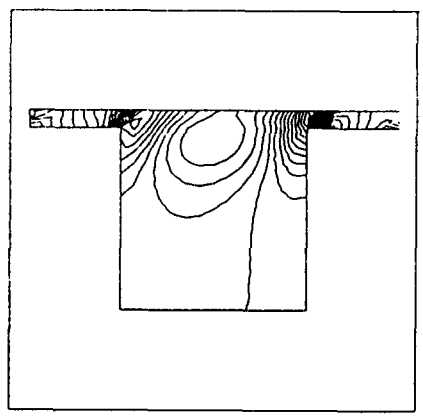


Figure 6.

It can be seen that the efficiency and speed-up values obtained at Re = 1 are significantly different from the others. This can be ascribed to the fact that, while linear problems take up the same time with independence of the physical conditions, nonlinear problems are solved by an iterative method, the rate of convergence of which does depend on the values of the physical parameters involved.

REMARKS.

1. It has been observed in this, and also in other numerical examples, that linear and nonlinear subproblems are properly balanced (T2 and T3 times are quite similar in Figure 2).
2. We see in Figure 2 that the proposed algorithm consists of a sequential part (corresponding to task T4) that cannot be avoided by using two processors in the same local area network. According to Amdahl's law, this fraction results in an efficiency below the expected value. By using MIMD multiprocessor, the fraction can be substantially reduced because:
   (a) data transfer between processors in both directions is much faster, and
   (b) the "coordination" operation, i.e., (9), can be readily parallelized.

REFERENCES

1. V. Peterson, *Future Generation Computer System* **5** (2/3), 243–258 (1989).
2. M. Marín, Análisis numérico del problema incompresible de Navier-Stokes, *Métodos de tipo direcciones alternadas*, Ph.D. Thesis (1986).
3. E. Fernández Cara and M. Marín, The convergence of two numerical schemes for the Navier-Stokes equations, *Numerische Mathematik* **55**, 33–60 (1989).
4. J.L. Lions, *Quelques Méthodes de Résolution des Problèmes aux Limites non Linéaires*, Dunod, Gauthiers-Villars, Paris, (1969).
5. R. Témam, *Navier-Stokes Equations*, 2nd ed., North-Holland, Amsterdam, (1977).
6. R. Glowinski, B. Mantel, J. Periaux, and O. Pironneau, A finite element approximation of Navier-Stokes equations for incompressible viscous fluids. Functional least-squares methods of solution, In *Computer Methods in Fluids*, (Edited by K. Morgan, C. Taylor, and A. Brebbia), pp. 84–133, Pentech Press, London, (1980).
7. P.A. Raviart, J.M. Thomas, *Introduction à l'Analyse Numérique des Equations aux Derivees Partielles*, Masson, Paris, (1983).
8. O. Pironneau, *Méthodes des Eléments Finis pour les Fluides*, Masson, Paris, (1983).
9. C. Cuvelier, A. Segal, and A. van Steenhoven, *Finite Element Methods and Navier-Stokes Equations*, D. Reidel Pub. Co., Dordrecht, Holland, (1986).
10. P.G. Ciarlet, *Introduction à l'Analyse Numérique Matricielle et à l'Optimisation*, Masson, Paris, (1982).
11. M. Bernadou *et al.*, *MODULEF, Une Bibliothèque Modulaire d'Eléments Finis*, INRIA, Roquencourt, (1989).
12. G.A. Selim, *The Design and Analysis of Parallel Algorithms*, Prentice-Hall Inc., (1989).
13. M.J. Quinn, *Designing Efficient Algorithms for Parallel Computers*, McGraw-Hill, (1987).