

**NUMERICAL METHODS FOR THE NAVIER–STOKES EQUATIONS.
APPLICATIONS TO THE SIMULATION OF COMPRESSIBLE
AND INCOMPRESSIBLE VISCOUS FLOWS**

M.O. BRISTEAU

INRIA, 78153 Le Chesnay, France

R. GLOWINSKI

Department of Mathematics, University of Houston, Texas, USA and INRIA

and

J. PERIAUX

Département d'Aérodynamique Théorique, Avions Marcel Dassault / Bréguet Aviation, Saint Cloud, France

Contents

Introduction	75
Part I. Numerical simulation of incompressible viscous flows	76
1. Formulation of the Navier–Stokes equations for incompressible viscous fluids	76
2. Time discretization by operator splitting methods	78
3. Least squares–conjugate gradient solution of the nonlinear subproblems	84
4. Solution of the “quasi” Stokes linear subproblems	91
5. Finite element approximations of the time dependent Navier–Stokes equations	101
6. Numerical solution of the incompressible Navier–Stokes equations founded on the stream function–vorticity formulation	104
7. A backward method of characteristics for the incompressible Navier–Stokes equations. Implementation of downstream boundary conditions for flows in unbounded domains.	116
8. Numerical simulation of some incompressible viscous flows	119
Part II. Numerical simulation of compressible viscous flows	139
9. Generalities	139
10. Formulation of the Navier–Stokes equations for compressible viscous fluids	139
11. Application of operator splitting methods to the solution of the time dependent compressible Navier–Stokes equations	142
12. Solution of the generalized Stokes subproblems	144
13. Least squares–conjugate gradient solution of the nonlinear subproblems	147
14. Finite element approximation	150
15. Self-adaptive mesh refinement techniques for accurate solution of the compressible Navier–Stokes equations	151
16. Numerical experiments	155
Conclusion	183
References	183

Introduction

The main goal of this report is to discuss the numerical solution of the Navier–Stokes equations modelling steady or unsteady flows of incompressible or compressible viscous fluids. In the first part, we shall consider the incompressible case and in the second part, the compressible case. In the incompressible case, we shall particularly emphasize the time discretization by operator splitting methods, since such methods provide an efficient way to decouple the two main difficulties of the problem, namely the incompressibility and the nonlinearity. In the second part, we shall see how some of the techniques used for incompressible fluids still apply to the compressible case.

The results of numerical experiments for two-dimensional and three-dimensional flows will be presented in order to show the possibilities of the methods discussed in this report.

PART I. NUMERICAL SIMULATION OF INCOMPRESSIBLE VISCOUS FLOWS

1. Formulation of the Navier–Stokes equations for incompressible viscous fluids

Let us consider a Newtonian incompressible viscous fluid. If Ω and Γ denote the region of the flow ($\Omega \subset \mathbb{R}^N$, $N = 2, 3$ in practice) and its boundary, respectively, then this flow is governed by the following Navier–Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (1.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \quad (\text{incompressibility condition}). \quad (1.2)$$

In (1.1), (1.2):

$$\nabla = \left\{ \frac{\partial}{\partial x_i} \right\}_{i=1}^N, \quad \Delta = \nabla^2 = \sum_{i=1}^N \frac{\partial^2}{\partial x_i^2}, \quad (a)$$

$$\mathbf{u} = \{u_i\}_{i=1}^N \quad \text{is the flow velocity,} \quad (b)$$

$$p \quad \text{is the pressure,} \quad (c)$$

$$\nu \quad \text{is a viscosity parameter,} \quad (d)$$

$$\mathbf{f} \quad \text{is a density of external forces.} \quad (e)$$

In (1.1), $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is a symbolic notation for the nonlinear vector term

$$\left\{ \sum_{j=1}^N u_j \frac{\partial u_i}{\partial x_j} \right\}_{i=1}^N$$

(more generally we shall denote by $(\mathbf{v} \cdot \nabla) \mathbf{w}$ the vector $\{\sum_{j=1}^N v_j \partial w_i / \partial x_j\}_{i=1}^N$).

Boundary conditions have to be added. For example, in the case of the airfoil A of fig. 1.1, we have (since the fluid is viscous) the following adherence condition

$$\mathbf{u} = \mathbf{0} \quad \text{on } \Gamma_A (= \partial A); \quad (1.3)$$

typical conditions at infinity are

$$\mathbf{u} = \mathbf{u}_\infty, \quad (1.4)$$

where \mathbf{u}_∞ is a constant vector (with respect to the space variables, at least).

If Ω is a bounded region of \mathbb{R}^N , very usual boundary conditions are

$$\mathbf{u} = \mathbf{g} \quad \text{on } \Gamma, \quad (1.5)$$

where (from the incompressibility of the fluid) the given function \mathbf{g} has to satisfy

$$\int_{\Gamma} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0, \quad (1.6)$$

where \mathbf{n} is the outward unit vector normal to Γ .

Finally, for the time dependent problem (1.1), (1.2), an initial condition such as

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x) \quad \text{a.e. on } \Omega, \quad (1.7)$$

with \mathbf{u}_0 given, is usually prescribed.

In practice, for the problem corresponding to fig. 1.1, we should replace Ω by a large bounded domain Ω_c (the computational domain), and on the external boundary Γ_∞ of Ω_c we should prescribe $\mathbf{u} = \mathbf{u}_\infty$, or some more sophisticated boundary conditions.

Remark 1.1: For two-dimensional problems on unbounded domains Ω , exponential stretching methods can be used, allowing very large computational domains (see ref. [1], chap. 7 and ref. [2] for applications of exponential stretching methods to inviscid flow calculations).

Remark 1.2: When using Ω_c (as above) instead of Ω , prescribing $\mathbf{u} = \mathbf{u}_\infty$ on the whole Γ_∞ may not be satisfactory if Ω_c is not sufficiently large. Actually, we should improve the computed solutions using as boundary conditions

$$\mathbf{u} = \mathbf{u}_\infty \quad \text{on} \quad \Gamma_\infty^-, \quad (1.8)$$

and either

$$\nu \frac{\partial \mathbf{u}}{\partial n} - \mathbf{n}_\infty p = 0 \quad \text{on} \quad \Gamma_\infty^+, \quad (1.9a)$$

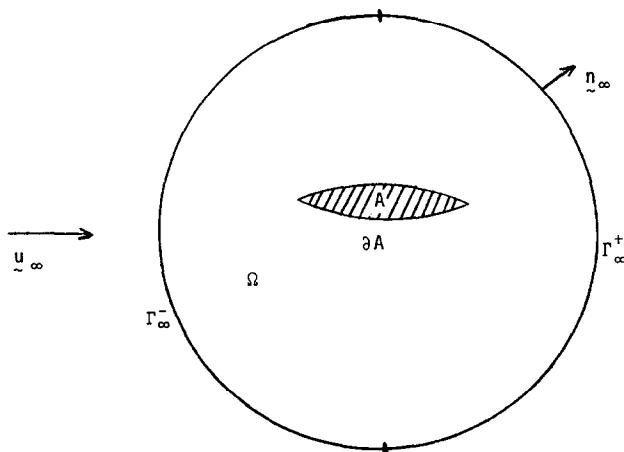


Fig. 1.1.

or

$$\frac{\partial}{\partial t} \omega + c \frac{\partial \omega}{\partial n} = 0 \quad \text{on} \quad \Gamma_{\infty}^+, \quad (1.9b)$$

where

- (i) $\Gamma_{\infty}^+ = \{x \mid x \in \Gamma_{\infty}, \mathbf{u}_{\infty} \cdot \mathbf{n}_{\infty}(x) \geq 0\}$,
- (ii) \mathbf{n}_{∞} is the outward unit vector normal to Γ_{∞} (see fig. 1.1),
- (iii) c is a constant; a natural choice seems to be $c = |\mathbf{u}_{\infty}|$,
- (iv) $\omega = \nabla \times \mathbf{u}$ is the vorticity of the flow.

The main reason for using either (1.8), (1.9a) or (1.8), (1.9b), instead of $\mathbf{u} = \mathbf{u}_{\infty}$ on Γ_{∞} , is that the former boundary conditions are less reflecting (i.e. more absorbing) than the latter.

The numerical implementation of more sophisticated absorbing boundary conditions, taking advantage of the time discretization and of the finite element approximation, will be discussed in section 7. \square

The Navier–Stokes equations for incompressible viscous fluids have motivated a large number of papers, books, reports and symposia; we shall limit our references to:

- (i) refs. [3–8] for the theoretical aspects,
- (ii) refs. [1,6,8–12] for the numerical aspects.

In both cases, the interested reader may consult the references contained in the above books.

The difficulties with the Navier–Stokes equations (even for flows at low Reynold’s numbers, in bounded regions Ω) are

- (i) The nonlinear term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ in (1.1);
- (ii) The incompressibility condition (1.2);
- (iii) The fact that the solutions of the Navier–Stokes equations are vector-valued functions of x , t , whose components are coupled by the nonlinear term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ and by the incompressibility condition $\nabla \cdot \mathbf{u} = 0$.

Using convenient operator splitting methods for the time discretization of the Navier–Stokes equations, we shall be able to decouple those difficulties associated to the nonlinearity and to the incompressibility, respectively.

For simplicity, we suppose from now on that Ω is bounded and that we have (1.5) as boundary conditions (with \mathbf{g} satisfying (1.6) and possibly depending upon t).

2. Time discretization by operator splitting methods

2.1. Generalities. Description of the basic schemes

Let us consider a real Hilbert space H ; we consider then in H the following initial value problem

$$\begin{cases} \frac{du}{dt} + A(u) = f, \\ u(0) = u_0, \end{cases} \quad (2.1)$$

where A is an operator from H to H and where f is a source term and u_0 the initial value, respectively. Let A_1 and A_2 be two operators such that

$$A = A_1 + A_2. \quad (2.2)$$

With $\Delta t (> 0)$ a time discretization step, let us define several schemes for solving (2.1), taking advantage of the decomposition (2.2):

A first scheme (Peaceman–Rachford type): The scheme is defined as follows

$$u^0 = u_0; \quad (2.3)$$

then for $n \geq 0$, with u^n known, compute $u^{n+1/2}$ and then u^{n+1} by

$$\frac{u^{n+1/2} - u^n}{\Delta t/2} + A_1(u^{n+1/2}) + A_2(u^n) = f^{n+1/2}, \quad (2.4)$$

$$\frac{u^{n+1} - u^{n+1/2}}{\Delta t/2} + A_1(u^{n+1/2}) + A_2(u^{n+1}) = f^{n+1}. \quad (2.5)$$

In (2.4), (2.5), $u^{n+\alpha}$ denotes an approximation of $u((n + \alpha)\Delta t)$, and $f^{n+\alpha} = f((n + \alpha)\Delta t)$.

A second scheme (θ -scheme): Let θ belong to the open interval $(0, 1/2)$. The idea behind the scheme is to split the time interval $[n\Delta t, (n + 1)\Delta t]$ into three subintervals, as shown in fig. 2.1, and integrate (approximately) with respect to time (using an implicit scheme for A_1 and an explicit scheme for A_2) on $[n\Delta t, (n + \theta)\Delta t]$, then switch the role of A_1 and A_2 on $[(n + \theta)\Delta t, (n + 1 - \theta)\Delta t]$, and finally, on $[(n + 1 - \theta)\Delta t, (n + 1)\Delta t]$ do like on $[n\Delta t, (n + \theta)\Delta t]$.

Using these principles we obtain the following scheme, some forms of which have been advocated in refs. [13–16] (for $\theta = 1/4$):

$$u^0 = u_0; \quad (2.6)$$

then, for $n \geq 0$, we obtain $u^{n+\theta}$, $u^{n+1-\theta}$, u^{n+1} , from u^n , as follows

$$\frac{u^{n+\theta} - u^n}{\theta \Delta t} + A_1(u^{n+\theta}) + A_2(u^n) = f^{n+\theta}, \quad (2.7)$$

$$\frac{u^{n+1-\theta} - u^{n+\theta}}{(1 - 2\theta) \Delta t} + A_1(u^{n+\theta}) + A_2(u^{n+1-\theta}) = f^{n+1-\theta}, \quad (2.8)$$

$$\frac{u^{n+1} - u^{n+1-\theta}}{\theta \Delta t} + A_1(u^{n+1}) + A_2(u^{n+1-\theta}) = f^{n+1}. \quad \square \quad (2.9)$$

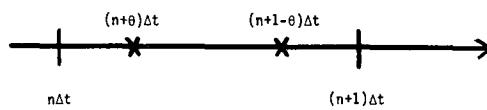


Fig. 2.1.

The convergence of (2.3)–(2.5) has been proved in ref. [17] (see also ref. [18]) under quite general monotonicity assumptions on A_1 and A_2 ; we don't know similar general results for (2.6)–(2.9) (but, very likely, the methods in ref. [17] still apply).

2.2. Convergence and stability properties of the basic schemes

Following the approach used in ref. [19], we consider for simplicity the case where $H = \mathbb{R}^N$, $f = 0$, $u_0 \in \mathbb{R}^N$, A is an $N \times N$ symmetric and positive definite matrix and where

$$A_1 = \alpha A, \quad A_2 = \beta A, \quad \text{with } \alpha + \beta = 1, \quad 0 < \alpha, \beta < 1. \quad (2.10)$$

In that case, the solution of (2.1) is clearly given by

$$u(t) = e^{-tA} u_0. \quad (2.11)$$

Analysis of scheme (2.3)–(2.5) (see also ref. [19]):

We have, from (2.4), (2.5), (2.10)

$$u^{n+1} = (I + \beta(\Delta t/2)A)^{-1}(I - \alpha(\Delta t/2)A)(I + \alpha(\Delta t/2)A)^{-1}(I - \beta(\Delta t/2)A)u^n. \quad (2.12)$$

Using a vector basis consisting of eigenvectors of A , we have, from (2.12), with obvious notation,

$$u_i^{n+1} = \frac{(1 - \alpha(\Delta t/2)\lambda_i)(1 - \beta(\Delta t/2)\lambda_i)}{(1 + \alpha(\Delta t/2)\lambda_i)(1 + \beta(\Delta t/2)\lambda_i)} u_i^n, \quad (2.13)$$

where $\lambda_i (> 0, \forall i = 1, \dots, N)$ is the i th eigenvalue of A ; we suppose that $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. Consider now the rational function R_1 defined by

$$R_1(x) = \frac{(1 - (\alpha/2)x)(1 - (\beta/2)x)}{(1 + (\alpha/2)x)(1 + (\beta/2)x)}; \quad (2.14)$$

we observe that $|R_1(x)| < 1, \forall x > 0$, implying, in that simple case, the unconditional stability of scheme (2.3)–(2.5). Since

$$\lim_{x \rightarrow +\infty} R_1(x) = 1, \quad (2.15)$$

we observe that for stiff problems, i.e. problems such that $\lambda_N/\lambda_1 \gg 1$, scheme (2.3)–(2.5) is not very good to damp simultaneously, the components of u^n associated to the large and to the small eigenvalues of A . From this observation, we can expect that scheme (2.3)–(2.5) is not well-suited to “capture” the steady state solutions of stiff problems (like those obtained from the discretization of partial differential equations); this has been confirmed by numerical experiments. Let us discuss now the accuracy of scheme (2.3)–(2.5). Since

$$e^{-x} = 1 - x + \frac{1}{2}x^2 + x^2\epsilon(x), \quad (2.16)$$

and, from (2.14),

$$R_1(x) = 1 - x + \frac{1}{2}x^2 + x^2\eta(x), \quad (2.17)$$

with $\lim_{x \rightarrow 0}\epsilon(x) = \lim_{x \rightarrow 0}\eta(x) = 0$, we have that scheme (2.3)–(2.5) is second order accurate in the simple case considered here. We observe, from (2.12), that if one takes $\alpha = \beta = \frac{1}{2}$, then the two linear systems which have to be solved, at each full step, are, in fact, associated to the same matrix $I + (\Delta t/4)A$.

Analysis of scheme (2.6)–(2.9):

We have (with $\theta' = 1 - 2\theta$)

$$\begin{cases} u^{n+1} = (I + \alpha\theta \Delta t A)^{-1}(I - \beta\theta \Delta t A)(I + \beta\theta' \Delta t A)^{-1}(I - \alpha\theta' \Delta t A) \\ \quad \times (I + \alpha\theta \Delta t A)^{-1}(I - \beta\theta \Delta t A)u^n, \end{cases} \quad (2.18)$$

which implies

$$u_i^{n+} = \frac{(1 - \beta\theta \Delta t \lambda_i)^2(1 - \alpha\theta' \Delta t \lambda_i)}{(1 + \alpha\theta \Delta t \lambda_i)^2(1 + \beta\theta' \Delta t \lambda_i)} u_i^n. \quad (2.19)$$

Consider now the rational function R_2 defined by

$$R_2(x) = \frac{(1 - \beta\theta x)^2(1 - \alpha\theta' x)}{(1 + \alpha\theta x)^2(1 + \beta\theta' x)}. \quad (2.20)$$

Since

$$\lim_{x \rightarrow +\infty} |R_2(x)| = \beta/\alpha, \quad (2.21)$$

we should prescribe

$$\alpha \geq \beta \quad (2.22)$$

to have, from (2.18), (2.19), the stability of scheme (2.6)–(2.9) for the large eigenvalues of A . We discuss now the accuracy of scheme (2.6)–(2.9); we can show that

$$R_2(x) = 1 - x + \frac{1}{2}x^2\{1 + (\beta^2 - \alpha^2)(2\theta^2 - 4\theta + 1)\} + x^2\eta(x), \quad (2.23)$$

with $\lim_{x \rightarrow 0}\eta(x) = 0$. It follows from (2.23) that scheme (2.6)–(2.9) is second order accurate if, either

$$\alpha = \beta \left(= \frac{1}{2} \text{ from (2.10)}\right), \quad (2.24)$$

or

$$\theta = 1 - \sqrt{2}/2 = 0.29289\dots; \quad (2.25)$$

scheme (2.6)–(2.9) is only first order accurate if neither (2.24) nor (2.25) hold. If one takes $\alpha = \beta = \frac{1}{2}$, it follows from (2.19) that scheme (2.6)–(2.9) is unconditionally stable, $\forall \theta \in]0, \frac{1}{2}[$; however, since (from (2.21)) we have

$$\lim_{x \rightarrow +\infty} |R_2(x)| = 1, \quad (2.26)$$

the remark stated for scheme (2.3)–(2.5) concerning the integration of stiff systems still holds. In general, we shall choose α and β in order to have the same matrix for all the partial steps of the integration procedure; i.e., α, β, θ have to satisfy

$$\alpha\theta = \beta(1 - 2\theta), \quad (2.27)$$

which implies

$$\alpha = (1 - 2\theta)/(1 - \theta), \quad \beta = \theta/(1 - \theta). \quad (2.28)$$

Combining (2.22), (2.28) we obtain

$$0 < \theta \leq \frac{1}{3}. \quad (2.29)$$

For $\theta = \frac{1}{3}$, (2.28) implies $\alpha = \beta = \frac{1}{2}$; the resulting scheme is just a variant of scheme (2.3)–(2.5).

If $0 < \theta < \frac{1}{3}$, and if α and β are given by (2.28), we have then

$$\lim_{x \rightarrow +\infty} |R_2(x)| = \beta/\alpha = \theta/(1 - 2\theta) < 1. \quad (2.30)$$

Actually, we can prove that $\theta \in [\theta^*, \frac{1}{3}]$ (with $\theta^* = 0.087385580\dots$) and α, β given by (2.28), imply the unconditional stability of scheme (2.6)–(2.9). Moreover, if $\theta \in]\theta^*, \frac{1}{3}[$, property (2.30) makes scheme (2.6)–(2.9) have good asymptotic properties as $n \rightarrow +\infty$ and for example, is well suited to compute steady state solutions. If $\theta = 1 - \sqrt{2}/2$ (resp. $\theta = \frac{1}{4}$), we have $\alpha = 2 - \sqrt{2}$, $\beta = \sqrt{2} - 1$, $\beta/\alpha = 1/\sqrt{2}$ (resp. $\alpha = \frac{2}{3}$, $\beta = \frac{1}{3}$, $\beta/\alpha = \frac{1}{2}$).

2.3. Application to the solution of the time dependent Navier–Stokes equations

2.3.1. A first operator splitting method

This method is directly derived from the Peaceman–Rachford scheme (2.3)–(2.5) and is described as follows:

$$\mathbf{u}^0 = \mathbf{u}_0, \quad (2.31)$$

then for $n \geq 0$, compute, from \mathbf{u}^n , $\{\mathbf{u}^{n+1/2}, p^{n+1/2}\}$ and \mathbf{u}^{n+1} by solving

$$\begin{cases} \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\Delta t/2} - \frac{\nu}{2} \Delta \mathbf{u}^{n+1/2} + \nabla p^{n+1/2} = \mathbf{f}^{n+1/2} + \frac{\nu}{2} \Delta \mathbf{u}^n - (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n & \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^{n+1/2} = 0 & \text{in } \Omega, \\ \mathbf{u}^{n+1/2} = \mathbf{g}^{n+1/2} & \text{on } \Gamma, \end{cases} \quad (2.32)$$

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\Delta t/2} - \frac{\nu}{2} \Delta \mathbf{u}^{n+1} + (\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1} = \mathbf{f}^{n+1} + \frac{\nu}{2} \Delta \mathbf{u}^{n+1/2} - \nabla p^{n+1/2} & \text{in } \Omega, \\ \mathbf{u}^{n+1} = \mathbf{g}^{n+1} & \text{on } \Gamma, \end{cases} \quad (2.33)$$

respectively.

2.3.2. A second operator splitting method

This method is derived from scheme (2.6)–(2.9) and is obtained as follows:

$$\mathbf{u}^0 = \mathbf{u}_0, \quad (2.34)$$

then for $n \geq 0$, starting from \mathbf{u}^n we solve

$$\begin{cases} \frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta \Delta t} - \alpha \nu \Delta \mathbf{u}^{n+\theta} + \nabla p^{n+\theta} = \mathbf{f}^{n+\theta} + \beta \nu \Delta \mathbf{u}^n - (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n & \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^{n+\theta} = 0 & \text{in } \Omega, \\ \mathbf{u}^{n+\theta} = \mathbf{g}^{n+\theta} & \text{on } \Gamma, \end{cases} \quad (2.35)$$

$$\begin{cases} \frac{\mathbf{u}^{n+1-\theta} - \mathbf{u}^{n+\theta}}{(1-2\theta) \Delta t} - \beta \nu \Delta \mathbf{u}^{n+1-\theta} + (\mathbf{u}^{n+1-\theta} \cdot \nabla) \mathbf{u}^{n+1-\theta} \\ = \mathbf{f}^{n+1-\theta} + \alpha \nu \Delta \mathbf{u}^{n+\theta} - \nabla p^{n+\theta} & \text{in } \Omega, \\ \mathbf{u}^{n+1-\theta} = \mathbf{g}^{n+1-\theta} & \text{on } \Gamma, \end{cases} \quad (2.36)$$

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1-\theta}}{\theta \Delta t} - \alpha \nu \Delta \mathbf{u}^{n+1} + \nabla p^{n+1} \\ = \mathbf{f}^{n+1} + \beta \nu \Delta \mathbf{u}^{n+1-\theta} - (\mathbf{u}^{n+1-\theta} \cdot \nabla) \mathbf{u}^{n+1-\theta} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^{n+1} = 0 & \text{in } \Omega, \\ \mathbf{u}^{n+1} = \mathbf{g}^{n+1} & \text{on } \Gamma. \end{cases} \quad (2.37)$$

2.3.3. Some comments and remarks concerning schemes (2.31)–(2.33) and (2.34)–(2.37)

Using the two above operator splitting methods, we have been able to decouple nonlinearity and incompressibility in the Navier–Stokes eqs. (1.1), (1.2). We shall describe, in the following

sections, the specific treatment of the subproblems encountered at each step of (2.31)–(2.33) and (2.34)–(2.37). We shall only consider the case where the subproblems are still continuous in space (since the formalism of the continuous problems is much simpler); for the fully discrete case, see refs. [1,20] where finite element approximations of (1.1), (1.2) are discussed.

We observe that $\mathbf{u}^{n+1/2}$ and $\mathbf{u}^{n+\theta}$, \mathbf{u}^{n+1} are obtained from the solutions of linear problems very close to the steady Stokes problem. Despite its greater complexity, scheme (2.30)–(2.37) is almost as economical to use as scheme (2.31)–(2.33); this is mainly due to the fact that the “quasi” steady Stokes problems (2.32) and (2.35), (2.37) (actually convenient finite element approximations of them) can be solved by quite efficient solvers, so that most of the computer time used to solve a full step is, in fact, used to solve the nonlinear subproblem.

If one uses scheme (2.34)–(2.37), the best choice for α and β is given by (2.28). With such a choice, many computer subprograms can be used for both the linear and nonlinear subproblems, resulting, therefore, in quite substantial core memory savings.

We can find, in ref. [9, chap. 4], a variant of scheme (2.31)–(2.33), given by

$$\mathbf{u}^0 = \mathbf{u}_0; \quad (2.38)$$

then for $n \geq 0$, starting from \mathbf{u}^n , compute $\{\mathbf{u}^{n+1/2}, p^{n+1/2}\}$ and \mathbf{u}^{n+1} by

$$\begin{cases} \frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\Delta t/2} - \nu \Delta \mathbf{u}^{n+1/2} + \nabla p^{n+1/2} = \mathbf{f}^{n+1/2} - (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n & \text{in } \Omega, \\ \nabla \cdot \mathbf{u}^{n+1/2} = 0 & \text{in } \Omega, \\ \mathbf{u}^{n+1/2} = \mathbf{g}^{n+1/2} & \text{on } \Gamma, \end{cases} \quad (2.39)$$

and

$$\begin{cases} \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\Delta t/2} - \nu \Delta \mathbf{u}^{n+1} + (\mathbf{u}^{n+1} \cdot \nabla) \mathbf{u}^{n+1} = \mathbf{f}^{n+1} - \nabla p^{n+1/2} & \text{in } \Omega, \\ \mathbf{u}^{n+1} = \mathbf{g}^{n+1} & \text{on } \Gamma. \end{cases} \quad (2.40)$$

3. Least squares–conjugate gradient solution of the nonlinear subproblems

3.1. Classical and variational formulations. Synopsis

At each full step of the operator splitting methods (2.31)–(2.33) and (2.34)–(2.37), we have to solve a nonlinear elliptic system of the following type

$$\begin{cases} \alpha \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \mathbf{f} & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma, \end{cases} \quad (3.1)$$

where α and ν are two positive parameters and where \mathbf{f} and \mathbf{g} are two given functions defined

on Ω and Γ , respectively. We do not discuss here the existence and uniqueness of solutions for problem (3.1).

We introduce now the following functional spaces of Sobolev's type (see e.g. refs. [21–24] for information on Sobolev spaces):

$$H^1(\Omega) = \{ \phi \mid \phi \in L^2(\Omega), \quad \partial\phi/\partial x_i \in L^2(\Omega), \quad \forall i = 1, \dots, N \}, \quad (3.2)$$

$$H_0^1(\Omega) = \{ \phi \mid \phi \in H^1(\Omega), \quad \phi = 0 \quad \text{on} \quad \Gamma \}, \quad (3.3)$$

$$V_0 = (H_0^1(\Omega))^N, \quad (3.4)$$

$$V_g = \{ \mathbf{v} \mid \mathbf{v} \in (H^1(\Omega))^N, \quad \mathbf{v} = \mathbf{g} \quad \text{on} \quad \Gamma \}; \quad (3.5)$$

if \mathbf{g} is sufficiently smooth, then V_g is nonempty.

We shall use the following notation

$$dx = dx_1 \, dx_2 \cdots dx_N,$$

and if

$$\mathbf{u} = \{ u_i \}_{i=1}^N, \quad \mathbf{v} = \{ v_i \}_{i=1}^N, \quad \text{then}$$

$$\mathbf{u} \cdot \mathbf{v} = \sum_{i=1}^N u_i v_i,$$

$$\nabla \mathbf{u} \cdot \nabla \mathbf{v} = \sum_{i=1}^N \nabla u_i \cdot \nabla v_i = \sum_{i=1}^N \sum_{j=1}^N \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j}.$$

Using Green's formula we can prove that for sufficiently smooth functions \mathbf{u} and \mathbf{v} , belonging to $(H^1(\Omega))^N$ and V_0 , respectively, we have

$$-\int_{\Omega} \Delta \mathbf{u} \cdot \mathbf{v} \, dx = \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, dx. \quad (3.6)$$

It can also be shown that \mathbf{u} is a solution of the nonlinear variational problem

$$\begin{cases} \mathbf{u} \in V_g, \\ \alpha \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, dx + \nu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, dx + \int_{\Omega} ((\mathbf{u} \cdot \nabla) \mathbf{u}) \cdot \mathbf{v} \, dx = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \\ \forall \mathbf{v} \in V_0, \end{cases} \quad (3.7)$$

and conversely. We observe that (3.1), (3.7) is not equivalent to a problem of the Calculus of Variations since there is no functional of \mathbf{v} with $(\mathbf{v} \cdot \nabla) \mathbf{v}$ as differential. However, using a

convenient least-squares formulation, we shall be able to solve (3.1), (3.7) by iterative methods originating from Nonlinear Programming, such as Conjugate Gradient for example. Indeed, since problem (3.1), (3.7) is nonlinear, a first natural choice for solving it is Newton's Method; such a method is described in section 3.2, just below.

3.2. On the solution of (3.1), (3.7) by Newton's method

We recall first some general principles concerning Newton's method:
Let X and Y be two real Banach spaces and F a C^1 (possibly nonlinear) operator from X into Y . The Newton's method for solving the following equation

$$F(u) = 0 \quad (3.8)$$

is defined by

$$u^0 \in X, \quad \text{given}; \quad (3.9)$$

then for $m \geq 0$, we define u^{m+1} from u^m by

$$u^{m+1} = u^m - (F'(u^m))^{-1} F(u^m), \quad (3.10)$$

where $F'(u^m)$ ($\in L(X, Y)$) is the derivative of F at u^m .

Applied to the nonlinear problem (3.1), (3.7), the Newton's algorithm (3.9), (3.10) yields

$$\mathbf{u}^0 \in V_g, \quad \text{given}; \quad (3.11)$$

then for $m \geq 0$, compute \mathbf{u}^{m+1} from \mathbf{u}^m , by solving the following linear elliptic problem

$$\begin{cases} \alpha \mathbf{u}^{m+1} - \nu \Delta \mathbf{u}^{m+1} + (\mathbf{u}^{m+1} \cdot \nabla) \mathbf{u}^m + (\mathbf{u}^m \cdot \nabla) \mathbf{u}^{m+1} = (\mathbf{u}^m \cdot \nabla) \mathbf{u}^m + \mathbf{f} \\ \text{in } \Omega, \\ \mathbf{u}^{m+1} = \mathbf{g} \quad \text{on } \Gamma, \end{cases} \quad (3.12)$$

whose variational formulation is given by

$$\begin{cases} \mathbf{u}^{m+1} \in V_g; \quad \forall \mathbf{v} \in V_0 \\ \alpha \int_{\Omega} \mathbf{u}^{m+1} \cdot \mathbf{v} \, dx + \nu \int_{\Omega} \nabla \mathbf{u}^{m+1} \cdot \nabla \mathbf{v} \, dx + \int_{\Omega} (\mathbf{u}^{m+1} \cdot \nabla) \mathbf{u}^m \cdot \mathbf{v} \, dx + \int_{\Omega} (\mathbf{u}^m \cdot \nabla) \mathbf{u}^{m+1} \cdot \mathbf{v} \, dx \\ = \int_{\Omega} (\mathbf{u}^m \cdot \nabla) \mathbf{u}^m \cdot \mathbf{v} \, dx + \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx. \end{cases} \quad (3.13)$$

Problem (3.12), (3.13) is associated to a nonself-adjoint elliptic operator, varying with m . After an appropriate approximation, we should obtain a linear system associated to a nonsymmetric matrix, varying with m . Solving such a linear system may be quite complicated and costly and it

is for that reason that we have developed alternative solution methods, founded on least squares formulations and conjugate gradient algorithms. Such methods are described in sections 3.3, 3.4, below.

3.3. Least squares formulation of the nonlinear problem (3.1), (3.7)

Let $\mathbf{v} \in \mathbf{V}_g$; from \mathbf{v} , we define $\mathbf{y} (= \mathbf{y}(\mathbf{v})) \in \mathbf{V}_0$ as the solution of

$$\begin{cases} \alpha \mathbf{y} - \nu \Delta \mathbf{y} = \alpha \mathbf{v} - \nu \Delta \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} - \mathbf{f} & \text{in } \Omega, \\ \mathbf{y} = \mathbf{0} & \text{on } \Gamma. \end{cases} \quad (3.14)$$

We observe that \mathbf{y} is obtained from \mathbf{v} via the solution of N uncoupled linear Poisson problems (one for each component of \mathbf{y}); using (3.6) it can be shown that problem (3.14) is actually equivalent to the linear variational problem

$$\begin{cases} \text{Find } \mathbf{y} \in \mathbf{V}_0 \text{ such that, } \forall \mathbf{z} \in \mathbf{V}_0 \text{ we have} \\ \alpha \int_{\Omega} \mathbf{y} \cdot \mathbf{z} \, dx + \nu \int_{\Omega} \nabla \mathbf{y} \cdot \nabla \mathbf{z} \, dx \\ = \alpha \int_{\Omega} \mathbf{v} \cdot \mathbf{z} \, dx + \nu \int_{\Omega} \nabla \mathbf{v} \cdot \nabla \mathbf{z} \, dx + \int_{\Omega} (\mathbf{v} \cdot \nabla) \mathbf{v} \cdot \mathbf{z} \, dx - \int_{\Omega} \mathbf{f} \cdot \mathbf{z} \, dx, \end{cases} \quad (3.15)$$

which has a unique solution. Suppose now that \mathbf{v} is a solution of the nonlinear problem (3.1), (3.7); the corresponding \mathbf{y} (obtained from the solution of (3.14), (3.15)) is clearly $\mathbf{y} = \mathbf{0}$. From this observation, it is quite natural to introduce the following (nonlinear) least squares formulation of (3.1), (3.7)

$$\begin{cases} \text{Find } \mathbf{u} \in \mathbf{V}_g \text{ such that} \\ J(\mathbf{u}) \leq J(\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}_g, \end{cases} \quad (3.16)$$

where $J: (\mathbf{H}^1(\Omega))^N \rightarrow \mathbf{R}$ is the function of \mathbf{v} defined by

$$J(\mathbf{v}) = \frac{1}{2} \int_{\Omega} \{ \alpha |\mathbf{y}|^2 + \nu |\nabla \mathbf{y}|^2 \} \, dx, \quad (3.17)$$

and where \mathbf{y} is defined from \mathbf{v} by (3.14), (3.15). We observe that if \mathbf{u} is a solution of (3.1), (3.7), then it is also a solution of (3.16), such that $J(\mathbf{u}) = 0$. Conversely, if \mathbf{u} is a solution of (3.16) such that $J(\mathbf{u}) = 0$, then it is also a solution of (3.1), (3.7).

We conclude this paragraph by the computation of $J'(\mathbf{v})$ where J' is the derivative of J . Let $\delta \mathbf{v}$ be a “perturbation” of \mathbf{v} compatible with the boundary condition $\mathbf{v} = \mathbf{g}$ on Γ . We have then $\delta \mathbf{v} = \mathbf{0}$ on Γ , implying that $\delta \mathbf{v} \in \mathbf{V}_0$. We have, from (3.17),

$$\delta J = \langle J'(\mathbf{v}), \delta \mathbf{v} \rangle = \int_{\Omega} \{ \alpha \mathbf{y} \cdot \delta \mathbf{y} + \nu \nabla \mathbf{y} \cdot \nabla \delta \mathbf{y} \} \, dx. \quad (3.18)$$

We also have, from (3.15),

$$\begin{cases} \delta y \in V_0; \quad \forall z \in V_0, \quad \text{we have} \\ \alpha \int_{\Omega} \delta y \cdot z \, dx + \nu \int_{\Omega} \nabla \delta y \cdot \nabla z \, dx \\ = \alpha \int_{\Omega} \delta v \cdot z \, dx + \nu \int_{\Omega} \nabla \delta v \cdot \nabla z \, dx + \int_{\Omega} (\delta v \cdot \nabla) v \cdot z \, dx + \int_{\Omega} (v \cdot \nabla) \delta v \cdot z \, dx. \end{cases} \quad (3.19)$$

Taking $z = y$ in (3.19), it follows from (3.18) that

$$\langle J'(v), \delta v \rangle = \alpha \int_{\Omega} y \cdot \delta v \, dx + \nu \int_{\Omega} \nabla y \cdot \nabla \delta v \, dx + \int_{\Omega} y \cdot (\delta v \cdot \nabla) v \, dx + \int_{\Omega} y \cdot (v \cdot \nabla) \delta v \, dx,$$

which implies in turn that $J'(v)$ can be identified with the linear functional from V_0 to \mathbb{R} , defined by

$$\begin{cases} \langle J'(v), z \rangle = \alpha \int_{\Omega} y \cdot z \, dx + \nu \int_{\Omega} \nabla y \cdot \nabla z \, dx + \int_{\Omega} y \cdot (v \cdot \nabla) z \, dx + \int_{\Omega} y \cdot (z \cdot \nabla) v \, dx, \\ \forall z \in V_0. \end{cases} \quad (3.20)$$

It has, therefore, a purely integral representation, a property which is of major importance in view of finite element implementations of the conjugate gradient algorithm to be described in the following section.

3.4. Conjugate gradient solution of the least squares problem (3.16)

3.4.1. Description of the algorithm

Step 0: Initialization

$$u^0 \in V_g, \quad \text{given}; \quad (3.21)$$

we define then $g^0, w^0 \in V_0$ by

$$\begin{cases} g^0 \in V_0, \\ \alpha \int_{\Omega} g^0 \cdot z \, dx + \nu \int_{\Omega} \nabla g^0 \cdot \nabla z \, dx = \langle J'(u^0), z \rangle, \quad \forall z \in V_0, \end{cases} \quad (3.22)$$

$$w^0 = g^0, \quad (3.23)$$

respectively. \square

Then for $m \geq 0$, assuming that u^m, g^m, w^m , are known, we obtain $u^{m+1}, g^{m+1}, w^{m+1}$ as follows:

Step 1: Descent

$$\begin{cases} \text{Find } \lambda_m \in \mathbb{R} \text{ such that} \\ J(\mathbf{u}^m - \lambda_m \mathbf{w}^m) \leq J(\mathbf{u}^m - \lambda \mathbf{w}^m), \quad \forall \lambda \in \mathbb{R}, \end{cases} \quad (3.24)$$

$$\mathbf{u}^{m+1} = \mathbf{u}^m - \lambda_m \mathbf{w}^m. \quad (3.25)$$

Step 2: Calculation of the new descent direction

$$\begin{cases} \text{Find } \mathbf{g}^{m+1} \in V_0 \text{ such that} \\ \alpha \int_{\Omega} \mathbf{g}^{m+1} \cdot \mathbf{z} \, dx + \nu \int_{\Omega} \nabla \mathbf{g}^{m+1} \cdot \nabla \mathbf{z} \, dx = \langle J'(\mathbf{u}^{m+1}), \mathbf{z} \rangle, \quad \forall \mathbf{z} \in V_0 \end{cases} \quad (3.26)$$

and compute either

$$\gamma_m = \frac{\alpha \int_{\Omega} |\mathbf{g}^{m+1}|^2 \, dx + \nu \int_{\Omega} |\nabla \mathbf{g}^{m+1}|^2 \, dx}{\alpha \int_{\Omega} |\mathbf{g}^m|^2 \, dx + \nu \int_{\Omega} |\nabla \mathbf{g}^m|^2 \, dx}, \quad (3.27a)$$

or

$$\gamma_m = \frac{\alpha \int_{\Omega} (\mathbf{g}^{m+1} - \mathbf{g}^m) \cdot \mathbf{g}^{m+1} \, dx + \nu \int_{\Omega} \nabla(\mathbf{g}^{m+1} - \mathbf{g}^m) \cdot \nabla \mathbf{g}^{m+1} \, dx}{\alpha \int_{\Omega} |\mathbf{g}^m|^2 \, dx + \nu \int_{\Omega} |\nabla \mathbf{g}^m|^2 \, dx}, \quad (3.27b)$$

or

$$\gamma_m = \frac{\alpha \int_{\Omega} (\mathbf{g}^{m+1} - \mathbf{g}^m) \cdot \mathbf{g}^{m+1} \, dx + \nu \int_{\Omega} \nabla(\mathbf{g}^{m+1} - \mathbf{g}^m) \cdot \nabla \mathbf{g}^{m+1} \, dx}{\alpha \int_{\Omega} \mathbf{w}^m \cdot (\mathbf{g}^m - \mathbf{g}^{m+1}) \, dx + \nu \int_{\Omega} \nabla \mathbf{w}^m \cdot \nabla(\mathbf{g}^m - \mathbf{g}^{m+1}) \, dx}, \quad (3.27c)$$

and set

$$\mathbf{w}^{m+1} = \mathbf{g}^{m+1} + \gamma_m \mathbf{w}^m. \quad \square \quad (3.28)$$

Do $m = m + 1$ and go to (3.24).

Conjugate gradient algorithms can be quite effective as solution methods for very large scale nonlinear problems. Some basic references are refs. [25,26] (see also ref. [1 chap. 7] and refs. [27,28] for further references and applications). Algorithm (3.21)–(3.28) is known as a

Fletcher–Reeves (resp. Polak–Ribiere, resp. Beale) conjugate gradient algorithm if one uses (3.27a) (resp. (3.27b), resp (3.27c)) to compute γ_m . From numerical experiments it appears that Polak–Ribiere and Beale algorithms have, in general, better convergence properties than the Fletcher–Reeves algorithm (there are, however, counter examples).

As we shall see in section 3.4.2, applying algorithm (3.21)–(3.28) to solve the least squares problem (3.16) requires the solution at each iteration of exactly three Dirichlet systems (i.e. $3N$ scalar Dirichlet problems) associated to the elliptic operator $\alpha I - \nu \Delta$.

Remark 3.1: A most important step when making use of algorithm (3.21)–(3.28) to solve the least squares problem (3.16), is the calculation of $\langle J'(\mathbf{u}^{m+1}), \mathbf{z} \rangle$ at each iteration. From the calculation of J' , done in section 3.3 to obtain $\langle J'(\mathbf{u}^{m+1}), \mathbf{z} \rangle$, we should proceed as follows:

- (i) We compute \mathbf{y}^{m+1} , associated to \mathbf{u}^{m+1} by (3.14)–(3.15), as indicated in section 3.4.2., below.
- (ii) We obtain then $\langle J'(\mathbf{u}^{m+1}), \mathbf{z} \rangle$ by taking $\mathbf{v} = \mathbf{u}^{m+1}$ and $\mathbf{y} = \mathbf{y}^{m+1}$ in (3.20).

3.4.2. Calculation of λ_m . Further comments on algorithm (3.21)–(3.28)

A problem of practical importance is the calculation of λ_m . Let's denote by $\mathbf{y}^m(\lambda)$ the solution of (3.14), (3.15) associated to $\mathbf{v} = \mathbf{u}^m - \lambda \mathbf{w}^m$. We clearly have

$$\mathbf{y}^m(0) = \mathbf{y}^m, \quad \mathbf{y}^m(\lambda_m) = \mathbf{y}^{m+1}, \quad (3.29)$$

and also

$$\mathbf{y}^m(\lambda) = \mathbf{y}^m - \lambda \mathbf{y}_1^m + \lambda^2 \mathbf{y}_2^m, \quad (3.30)$$

where \mathbf{y}_1^m , \mathbf{y}_2^m are the solutions of

$$\begin{cases} \alpha \mathbf{y}_1^m - \nu \Delta \mathbf{y}_1^m = \alpha \mathbf{w}^m - \nu \Delta \mathbf{w}^m + (\mathbf{u}^m \cdot \nabla) \mathbf{w}^m + (\mathbf{w}^m \cdot \nabla) \mathbf{u}^m & \text{in } \Omega, \\ \mathbf{y}_1^m = \mathbf{0} & \text{on } \Gamma, \end{cases} \quad (3.31)$$

$$\begin{cases} \alpha \mathbf{y}_2^m - \nu \Delta \mathbf{y}_2^m = (\mathbf{w}^m \cdot \nabla) \mathbf{w}^m & \text{in } \Omega, \\ \mathbf{y}_2^m = \mathbf{0} & \text{on } \Gamma, \end{cases} \quad (3.32)$$

respectively. Since

$$J(\mathbf{u}^m - \lambda \mathbf{w}^m) = \frac{1}{2} \int_{\Omega} \{ \alpha |\mathbf{y}^m(\lambda)|^2 + \nu |\nabla \mathbf{y}^m(\lambda)|^2 \} dx, \quad (3.33)$$

the function $\lambda \rightarrow J(\mathbf{u}^m - \lambda \mathbf{w}^m)$ is, from (3.30), a quartic polynomial in λ that we shall denote by $j_m(\lambda)$ and λ_m is, therefore, a solution of the cubic equation

$$j'_m(\lambda) = 0. \quad (3.34)$$

We shall use the standard Newton's method to compute λ_m from (3.34), starting from $\lambda = 0$. The resulting algorithm is given by

$$\lambda^0 = 0. \quad (3.35)$$

Then for $k \geq 0$, we obtain λ^{k+1} from λ^k by

$$\lambda^{k+1} = \lambda^k - j'_m(\lambda^k)/j''_m(\lambda^k). \quad (3.36)$$

In our calculations we always observed a very fast convergence of algorithm (3.35), (3.36). Once λ_m is known, we know y^{m+1} since (from (3.29)) $y^{m+1} = y^m(\lambda_m)$. If we count now the number of Dirichlet systems for $\alpha I - \nu \Delta$ to be solved at each iteration, we observe that we have to solve only three such systems; namely (3.31), (3.32), and then (3.26) (to obtain \mathbf{g}^{m+1}). This number is optimal for a nonlinear problem since the solution of a linear problem, by least squares–preconditioned conjugate gradient algorithms, requires the solution at each iteration of two linear systems associated to the preconditioning operator.

From the above remarks, it appears clearly that the practical implementation of algorithm (3.21)–(3.28) will require an efficient (direct or iterative) elliptic solver, like those discussed in refs. [29,30].

As a final comment, we would like to mention that algorithm (3.21)–(3.28) (in fact, its finite dimensional variants) is quite efficient when used in combination with the operator splitting methods of section 2 to solve the test problems of section 8. Three to five iterations suffice to reduce the value of the cost function by a factor of 10^4 to 10^6 . However, in view of other applications, we have been testing some of these methods, such as the methods discussed in refs. [31,32], which combine the features of conjugate gradient and quasi-Newton algorithms. The results obtained for transonic potential flows have been quite impressive (see ref. [33]), but so far, quite disappointing for the Navier–Stokes equations (at least for the test cases we have been considering).

Remark 3.2: As stopping test for algorithm (3.21)–(3.28), we have been using

$$J(\mathbf{u}^m)/J(\mathbf{u}^0) \leq \epsilon, \quad (3.37)$$

where ϵ is a “small” positive number. Another test could have been

$$\|\mathbf{g}^m\|/\|\mathbf{g}^0\| \leq \epsilon \quad (3.38)$$

with

$$\|\mathbf{g}^m\| = \left(\alpha \int_{\Omega} |\mathbf{g}^m|^2 dx + \nu \int_{\Omega} |\nabla \mathbf{g}^m|^2 dx \right)^{1/2}.$$

We found (3.37) more interesting, however, since (3.38), unlike (3.37), cannot discriminate between global and local minima of J .

4. Solution of the “quasi” Stokes linear subproblems

4.1. Generalities. Synopsis

At each full step of the splitting methods discussed in section 2.3, we have to solve one or two

linear problems of the following type

$$\begin{cases} \alpha \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = \mathbf{g} & \text{on } \Gamma \quad \left(\text{with } \int_{\Omega} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0 \right), \end{cases} \quad (4.1)$$

where α and ν are two positive parameters and where \mathbf{f} and \mathbf{g} are two given functions defined on Ω and Γ , respectively. We recall that if \mathbf{f} and \mathbf{g} are sufficiently smooth, then problem (4.1) has a unique solution in $V_g \times (L^2(\Omega)/R)$ (with V_g still defined by (3.5) and $p \in L^2(\Omega)/R$ means that p is defined only up to an arbitrary constant). We shall describe below several iterative methods for solving (4.1) which are quite easy to implement using finite element methods (more details are given in refs. [1, 9, 10, 20], together with convergence proofs).

4.2. A first iterative method for solving (4.1)

This method is quite classical and is defined as follows

$$p^0 \in L^2(\Omega), \quad \text{given}; \quad (4.2)$$

then for $m \geq 0$, define \mathbf{u}^m and p^{m+1} from p^m by

$$\begin{cases} \alpha \mathbf{u}^m - \nu \Delta \mathbf{u}^m = \mathbf{f} - \nabla p^m & \text{in } \Omega, \\ \mathbf{u}^m = \mathbf{g} & \text{on } \Gamma, \end{cases} \quad (4.3)$$

$$p^{m+1} = p^m - \rho \nabla \cdot \mathbf{u}^m. \quad (4.4)$$

Concerning the convergence of algorithm (4.2)–(4.4), we have the following

Proposition 4.1: Suppose that

$$0 < \rho < 2\nu/N; \quad (4.5)$$

we have then

$$\lim_{m \rightarrow +\infty} \{ \mathbf{u}^m, p^m \} = \{ \mathbf{u}, p_0 \} \quad \text{in } (H^1(\Omega))^N \times L^2(\Omega) \quad (4.6)$$

where $\{ \mathbf{u}, p_0 \}$ is the solution of (4.1) such that

$$\int_{\Omega} p_0 \, dx = \int_{\Omega} p^0 \, dx. \quad (4.7)$$

Moreover, the convergence is linear (i.e. the sequence $\| \mathbf{u}^m - \mathbf{u} \|_{(H^1(\Omega))^N}$ and $\| p^m - p_0 \|_{L^2(\Omega)}$ converge to zero, as fast, at least, as geometric sequences).

See ref. [1] for a proof of proposition 4.1.

Remark 4.1: When using algorithm (4.2)–(4.4) to solve (4.1), we have to solve at each iteration, N uncoupled scalar Dirichlet problems for $\alpha I - \nu \Delta$, to obtain \mathbf{u}^m from p^m . We see, again (as in section 3), the importance of having efficient Dirichlet solvers for $\alpha I - \nu \Delta$.

Remark 4.2: Algorithm (4.2)–(4.4) is related to the so-called method of artificial compressibility of Chorin–Yanenko. Indeed, we can view (4.4) as obtained by a time discretization process from the equation

$$\partial p / \partial t + \nabla \cdot \mathbf{u} = 0$$

(ρ being the size of the time discretization step).

4.3. Conjugate gradient methods for problem (4.2)

4.3.1. Generalities. Synopsis

The above algorithm (4.2)–(4.4) is, in fact, the simplest one of a whole family of descent methods, including among them conjugate gradient algorithms. To justify the use of conjugate gradient methods for solving problem (4.1), we shall show (in section 4.3.2) that the pressure p is a solution of a functional equation associated to a self-adjoint and strongly elliptic operator. Such an equation can be solved, therefore, by conjugate gradient methods like those described in sections 4.3.3, 4.3.4.

4.3.2. A functional equation satisfied by the pressure

We suppose Ω is bounded in \mathbb{R}^N . Let's define H by

$$H = \left\{ q \mid q \in L^2(\Omega), \int_{\Omega} q(x) dx = 0 \right\}, \quad (4.8)$$

and then $A : L^2(\Omega) \rightarrow L^2(\Omega)$ by:

Take $q \in L^2(\Omega)$ and solve

$$\begin{cases} \alpha \mathbf{u}_q - \nu \Delta \mathbf{u}_q = -\nabla q & \text{in } \Omega, \\ \mathbf{u}_q = \mathbf{0} & \text{on } \Gamma \end{cases} \quad (4.9)$$

in $(H_0^1(\Omega))^N$; let then

$$Aq = \nabla \cdot \mathbf{u}_q. \quad (4.10)$$

Since

$$\int_{\Omega} Aq dx = \int_{\Omega} \nabla \cdot \mathbf{u}_q dx = \int_{\Gamma} \mathbf{u}_q \cdot \mathbf{n} d\Gamma = 0,$$

we have $Aq \in H$, $\forall q \in L^2(\Omega)$. We have also

$$\int_{\Omega} (Aq) q' dx = \alpha \int_{\Omega} \mathbf{u}_q \cdot \mathbf{u}_{q'} dx + \nu \int_{\Omega} \nabla \mathbf{u}_q \cdot \nabla \mathbf{u}_{q'} dx, \quad \forall q, q' \in L^2(\Omega), \quad (4.11)$$

from which we can deduce the following theorem.

Theorem 4.1: The operator A is self-adjoint and is a strongly elliptic isomorphism from H onto H .

See ref. [34] for a proof of this result. Define now $\mathbf{u}_0 \in V_g$ by

$$\begin{cases} \alpha\mathbf{u}_0 - \nu \Delta \mathbf{u}_0 = \mathbf{f} & \text{in } \Omega, \\ \mathbf{u}_0 = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (4.12)$$

We clearly have, from the definition of A and from (4.1), (4.12), that

$$Ap = -\nabla \cdot \mathbf{u}_0. \quad (4.13)$$

From the properties of A , it is tempting to solve (4.13) (and therefore (4.1)) by conjugate gradient algorithms operating in the spaces $L^2(\Omega)$ and H . Such algorithms are discussed in sections 4.3.3, 4.3.4.

4.3.3. Conjugate gradient methods for a class of linear variational problems

Let V be a real Hilbert space whose scalar product and associated norm are denoted by

$$(\cdot, \cdot) \quad \text{and} \quad \|\cdot\|,$$

respectively. Consider now the following mathematical objects:

- (i) $a: V \times V \rightarrow \mathbb{R}$, a bilinear, continuous functional which is also V -elliptic (i.e., there exists $\alpha > 0$ such that $a(v, v) \geq \alpha \|v\|^2, \forall v \in V$),
- (ii) $L: V \rightarrow \mathbb{R}$, linear and continuous.

To $a(\cdot, \cdot)$ and $L(\cdot)$, we associate the following (linear variational) problem

$$\begin{cases} \text{Find } \mathbf{u} \in V \text{ such that} \\ a(\mathbf{u}, \mathbf{v}) = L(\mathbf{v}), \quad \forall \mathbf{v} \in V. \end{cases} \quad (4.14)$$

It is a classical result that problems such as (4.14) have a unique solution if the above hypotheses on $a(\cdot, \cdot)$ and $L(\cdot)$ holds (see, for example, [1, appendix 1] for a proof).

Suppose now that $a(\cdot, \cdot)$ is symmetric, i.e.

$$a(\mathbf{v}, \mathbf{w}) = a(\mathbf{w}, \mathbf{v}), \quad \forall \mathbf{v}, \mathbf{w} \in V.$$

In such a case, it is easily shown that problem (4.14) is also equivalent to the following minimization problem

$$\begin{cases} \mathbf{u} \in V, \\ J(\mathbf{u}) \leq J(\mathbf{v}), \quad \forall \mathbf{v} \in V, \end{cases} \quad (4.15a)$$

with

$$J(\mathbf{v}) = \frac{1}{2}a(\mathbf{v}, \mathbf{v}) - L(\mathbf{v}). \quad (4.15b)$$

In the symmetric case, the above equivalence result suggests solving (4.14), via (4.15), by a conjugate gradient method, like, for example, the one described below.

Description of a conjugate gradient method for solving (4.14)

Step 0: Initialization

$$u^0 \in V \text{ is given; } \quad (4.16)$$

solve then

$$\begin{cases} g^0 \in V, \\ (g^0, v) = a(u^0, v) - L(v), \quad \forall v \in V; \end{cases} \quad (4.17)$$

if $g^0 = 0$ (or if $\|g^0\|$ is sufficiently small) take $u = u^0$; if not, set

$$w^0 = g^0. \quad \square \quad (4.18)$$

Then for $m \geq 0$ assuming that u^m , g^m , w^m are known, compute u^{m+1} , g^{m+1} , w^{m+1} as follows:

Step 1: Descent

Compute

$$\rho_m = \frac{(g^m, w^m)}{a(w^m, w^m)} \left(= \frac{\|g^m\|^2}{a(w^m, w^m)} \right), \quad (4.19)$$

and take

$$u^{m+1} = u^m - \rho_m w^m. \quad (4.20)$$

Step 2: Test of the convergence and calculation of the new descent direction

Solve

$$\begin{cases} g^{m+1} \in V, \\ (g^{m+1}, v) = (g^m, v) - \rho_m a(w^m, v), \quad \forall v \in V; \end{cases} \quad (4.21)$$

if $g^{m+1} = 0$, or if $\|g^{m+1}\|$ is sufficiently small, take $u = u^{m+1}$; if not, compute

$$\gamma_m = \|g^{m+1}\|^2 / \|g^m\|^2 \quad (4.22)$$

and define

$$w^{m+1} = g^{m+1} = \gamma_m w^m. \quad \square \quad (4.23)$$

Do $m = m + 1$ and go to (4.19).

Proving the convergence of algorithm (4.16)–(4.23) is not too difficult (see, e.g. ref. [26] for such a proof). What is more interesting, indeed, is that one has the following estimate for the speed of convergence (this is also proved in ref. [26]):

$$\| u^m - u \| \leq c \left(\frac{\sqrt{\nu(a)} - 1}{\sqrt{\nu(a)} + 1} \right)^m, \quad (4.24)$$

where c depends on u^0 . In (4.24), $\nu(a)$ denotes the condition number of $a(\cdot, \cdot)$, namely

$$\nu(a) = \| A \| \| A^{-1} \|, \quad (4.25)$$

where A is the isomorphism from V onto V uniquely defined by

$$a(v, w) = (Av, w), \quad \forall v, w \in V. \quad (4.26)$$

Despite its apparent simplicity, algorithm (4.16)–(4.23) is one of the most useful tools in Scientific Computing. The number of situations to which it can be applied is enormous and would fill a very large book by itself (see, e.g. refs. [1,35] for some applications). In section 4.3.4 just below, we shall apply algorithm (4.16)–(4.23) to the solution of problem (4.1), via the solution of the linear eq. (4.13).

4.3.4. Application of the conjugate gradient algorithm (4.16)–(4.23) to the solution of the Stokes problem (4.1), via the solution of (4.13)

Let us denote by p the pressure in (4.1) belonging to

$$H = \left\{ q \mid q \in L^2(\Omega), \quad \int_{\Omega} q \, dx = 0 \right\}.$$

It follows from section 4.3.2 that p satisfies (4.13), or equivalently:

$$\begin{cases} p \in H, \\ \int_{\Omega} (Ap) q \, dx = - \int_{\Omega} \nabla \cdot \mathbf{u}_0 q \, dx, \quad \forall q \in H. \end{cases} \quad (4.27)$$

From the properties of A discussed in section 4.3.2 and particularly theorem 4.1, the bilinear form

$$\{q, q'\} \rightarrow \int_{\Omega} (Aq) q' \, dx$$

is continuous, symmetric and H -elliptic over $H \times H$. Since the linear functional

$$q \rightarrow \int_{\Omega} \nabla \cdot \mathbf{u}_0 q \, dx$$

is clearly continuous over H , problem (4.27) is a particular case of problem (4.14) and therefore, can be solved by the conjugate gradient algorithm (4.16)–(4.23). Before describing the resulting algorithm, we have to decide on the scalar product to be used over H . In that direction, let's consider first two real parameters a and b , such that

$$a > 0 \quad \text{and} \quad b \geq 0.$$

We define now the operator B from H into H by

$$Bq = aq + b\phi_q, \quad (4.28)$$

where ϕ_q is the unique solution in $H \cap H^1(\Omega)$ of the Neumann problem

$$\begin{cases} -\Delta\phi_q = q & \text{in } \Omega, \\ \partial\phi_q/\partial n = 0 & \text{on } \Gamma; \end{cases} \quad (4.29)$$

(since $\int_{\Omega} q \, dx = 0$, the Neumann problem (4.29) has an infinity of solutions in $H^1(\Omega)$, all defined to within an arbitrary additive constant and therefore, there is a unique solution of (4.29) such that $\int_{\Omega} \phi_q \, dx = 0$). Operator B is clearly continuous from H into H (H being equipped with the classical $L^2(\Omega)$ -norm

$$\|q\|_{L^2(\Omega)} = \left(\int_{\Omega} |q(x)|^2 \, dx \right)^{1/2}.$$

In fact, operator B is a self-adjoint strongly elliptic isomorphism from H onto H . To prove these properties, it suffices to prove that the continuous bilinear form

$$d: H \times H \rightarrow \mathbb{R}$$

defined by

$$d(q, q') = \int_{\Omega} (Bq) q' \, dx \quad (4.30)$$

is symmetric and H -elliptic *. We have, with obvious notation,

$$\begin{aligned} d(q, q') &= a \int_{\Omega} qq' \, dx + b \int_{\Omega} \phi_q q' \, dx = a \int_{\Omega} qq' \, dx - b \int_{\Omega} \phi_q \Delta\phi_{q'} \, dx \\ &= a \int_{\Omega} qq' \, dx + b \int_{\Omega} \nabla\phi_q \cdot \nabla\phi_{q'} \, dx, \quad \forall q, q' \in H. \end{aligned} \quad (4.31)$$

The symmetry of $d(\cdot, \cdot)$ is obvious from (4.31). We also have

$$d(q, q) = a \int_{\Omega} |q|^2 \, dx + b \int_{\Omega} |\nabla\phi_q|^2 \, dx \geq a \int_{\Omega} |q|^2 \, dx, \quad \forall q \in H. \quad (4.32)$$

The H -ellipticity of $d(\cdot, \cdot)$ is clear from (4.32).

* I.e. there exists $\beta > 0$ such that $d(q, q) \geq \beta \|q\|_{L^2(\Omega)}^2$, $\forall q \in H$.

Let's denote by S the inverse of operator B , i.e.

$$S = B^{-1}. \quad (4.33)$$

From the properties of B , operator S is also a strongly elliptic and self-adjoint isomorphism from H onto H , implying that the bilinear form

$$\{q, q'\} \rightarrow \int_{\Omega} (Sq) q' \, dx$$

defines over H , a scalar product such that the associated norm is equivalent to the classical $L^2(\Omega)$ -norm. From now on, we shall use the Hilbert structure over H associated with the scalar product $(\cdot, \cdot)_H$ defined by

$$(q, q')_H = \int_{\Omega} (Sq) q' \, dx (= (Sq, q')_{L^2(\Omega)}). \quad (4.34)$$

Indeed, we shall use $(\cdot, \cdot)_H$ for the solution of problem (4.1) via the conjugate gradient solution of problem (4.13) by algorithm (4.16)–(4.23).

The above metric was introduced, using a different presentation, in refs. [36,37] and is discussed in further detail in ref. [38] (where more general boundary conditions are considered).

Description of the conjugate gradient algorithm

We follow the presentation in ref. [38, chap. 2]. Algorithm (4.16)–(4.23), applied to the solution of the Stokes problem (4.1), yields, if one uses the scalar product defined by (4.34), the following algorithm:

Step 0: Initialization

$$p^0 \in H, \text{ given}; \quad (4.35)$$

solve then the elliptic system

$$\begin{cases} \alpha \mathbf{u}^0 - \nu \Delta \mathbf{u}^0 = \mathbf{f} - \nabla p^0 & \text{in } \Omega, \\ \mathbf{u}^0 = \mathbf{g} & \text{on } \Gamma. \end{cases} \quad (4.36)$$

If $\nabla \cdot \mathbf{u}^0 = 0$ (or, in practice, is sufficiently small), then take $\mathbf{u} = \mathbf{u}^0$ and $p = p^0$; if this is not the case, compute the solution of the Neumann's problem

$$\begin{cases} -\Delta \psi^0 = \nabla \cdot \mathbf{u}^0 & \text{in } \Omega, \\ \partial \psi^0 / \partial n = 0 & \text{on } \Gamma \end{cases} \quad (4.37)$$

which satisfies $\int_{\Omega} \psi^0 dx = 0$, and set

$$g^0 = a \nabla \cdot \mathbf{u}^0 + b \psi^0, \quad (4.38)$$

$$w^0 = g^0. \quad \square \quad (4.39)$$

Then for $m \geq 0$, assuming that $p^m, \mathbf{u}^m, g^m, w^m$ are known, compute $p^{m+1}, \mathbf{u}^{m+1}, g^{m+1}, w^{m+1}$ as follows:

Step 1: Descent

Solve the elliptic system

$$\begin{cases} \alpha \chi^m - \nu \Delta \chi^m = -\nabla w^m & \text{in } \Omega, \\ \chi^m = \mathbf{0} & \text{on } \Gamma, \end{cases} \quad (4.40)$$

and compute

$$\rho_m = \frac{\int_{\Omega} \nabla \cdot \mathbf{u}^m w^m dx}{\int_{\Omega} \nabla \cdot \chi^m w^m dx} \left(= \frac{\int_{\Omega} \nabla \cdot \mathbf{u}^m g^m dx}{\int_{\Omega} \nabla \cdot \chi^m w^m dx} \right), \quad (4.41)$$

$$p^{m+1} = p^m - \rho_m w^m, \quad (4.42)$$

$$\mathbf{u}^{m+1} = \mathbf{u}^m - \rho_m \chi^m. \quad \square \quad (4.43)$$

Step 2: Testing the convergence and calculation of the new descent direction

If $\nabla \cdot \mathbf{u}^{m+1} = 0$ (or, in practice, is sufficiently small), take $\mathbf{u} = \mathbf{u}^{m+1}$, $p = p^{m+1}$; if this is not the case, solve the Neumann's problem

$$\begin{cases} -\Delta \phi^m = \nabla \cdot \chi^m & \text{in } \Omega, \\ \partial \phi^m / \partial n = 0 & \text{on } \Gamma \end{cases} \quad (4.44)$$

with $\int_{\Omega} \phi^m dx = 0$, and take

$$g^{m+1} = g^m - \rho_m (a \nabla \cdot \chi^m + b \phi^m), \quad (4.45)$$

$$\gamma_m = \frac{\int_{\Omega} g^{m+1} \nabla \cdot \mathbf{u}^{m+1} dx}{\int_{\Omega} g^m \nabla \cdot \mathbf{u}^m dx}, \quad (4.46)$$

$$w^{m+1} = g^{m+1} + \gamma_m w^m, \quad \square \quad (4.47)$$

Do $m = m + 1$ and go to (4.41).

If follows from Fourier Analysis (cf. refs. [37,38] for details) that the optimal choice for a and b is provided by

$$a = \nu, \quad b = \alpha. \quad (4.48)$$

With such choices, the numerical experiments reported in refs. [36–38] show that the convergence of algorithm (4.35)–(4.47) is very fast (3 to 4 iterations at most) and remarkably independent of the Reynolds number (as soon as $\alpha \sim 1/\Delta t$ is sufficiently large).

A generalization of algorithm (4.35)–(4.47), which is able to handle boundary conditions more complicated than Dirichlet's, is discussed in ref. [38].

We observe that algorithm (4.35)–(4.47) requires the solution, at each iteration, of only one Dirichlet system, namely (4.40). This algorithm is therefore, almost as economical per iteration as algorithm (4.2)–(4.4), but since it requires fewer iterations, we should use (4.35)–(4.47) instead of (4.2)–(4.4).

4.4. An exact penalty method for solving the Stokes problem (4.1)

This method is, in fact, a generalization of algorithm (4.2)–(4.4) and is related to the augmented Lagrangian methods discussed in the monograph of Glowinski and Le Tallec [39] (see also Fortin and Glowinski [40], Glowinski [1, chap. 7]). It is defined as follows (with r a positive parameter):

$$p^0 \in L^2(\Omega), \quad \text{given}; \quad (4.49)$$

then for $m \geq 0$, define \mathbf{u}^m and p^{m+1} from p^m by

$$\begin{cases} \alpha \mathbf{u}^m - \nu \Delta \mathbf{u}^m - r \nabla (\nabla \cdot \mathbf{u}^m) = \mathbf{f} - \nabla p^m \\ \mathbf{u}^m = \mathbf{g} \quad \text{on} \quad \Gamma, \end{cases} \quad (4.50)$$

$$p^{m+1} = p^m - \rho \nabla \cdot \mathbf{u}^m. \quad (4.51)$$

Concerning the convergence of algorithm (4.49)–(4.51), we should prove (see, e.g. ref. [1, chap. 7] for such a proof) the following:

Proposition 4.2: Suppose that

$$0 < \rho < 2(r + \nu/N); \quad (4.52)$$

then the convergence result (4.6) holds for $\{\mathbf{u}^m, p^m\}$ the convergence being still linear. Moreover, if one uses $\rho = r$ we have

$$\|p^m - p_0\|_{L^2(\Omega)} \leq \|p^0 - p_0\|_{L^2(\Omega)} \left(\frac{\|A^{-1}\|}{r} \right)^m, \quad \forall m \geq 0,$$

where A is the operator defined by (4.9), (4.10); a similar estimate holds for $\|\mathbf{u}^m - \mathbf{u}\|_{(H^1(\Omega))^N}$ if $\rho = r$.

Remark 4.3. (About the choice of ρ and r): We should use $\rho = r$ in practice, since (from the above proposition) the convergence ratio of algorithm (4.49)–(4.51) is $\mathcal{O}(r^{-1})$ for large values of r . In many applications, taking $r = 10^4\nu$, we have a practical convergence of algorithm (4.49)–(4.51) in 3 to 4 iterations. There is, however, a practical upper bound for r . This follows from the fact that for too large values of r , problem (4.50) will be ill-conditioned and its practical solution sensitive to round-off errors.

Remark 4.4: Problem (4.50) is more complicated to solve in practice than problem (4.3) (or (4.40)), since the components of \mathbf{u}^m are coupled by the linear term $\nabla(\nabla \cdot \mathbf{u}^m)$. Actually, the elliptic partial differential operator in the left hand side of (4.50) is very close to the linear elasticity operator and close variants of it occur naturally in compressible (see part II) and/or turbulent viscous flow problems.

Remark 4.5: Other methods for solving the “quasi” Stokes problem (4.1) are discussed in refs. [1,10,20,41,42].

5. Finite element approximation of the time dependent Navier–Stokes equations

We shall describe in this section a specific class of finite element approximations for the time dependent Navier–Stokes equations. Actually, these methods, which lead to continuous approximations for both pressure and velocity, are fairly simple and some of them have been known for years. They have been advocated, for example, by Hood and Taylor (see ref. [43]). Other finite element approximations of the incompressible Navier–Stokes equations can be found in refs. [1,6,9,10,44] (see also the references therein).

The two fundamental references (in our opinion) concerning the convergence of the fully discrete Navier–Stokes equations (finite elements in space, finite differences in time) are refs. [45,46].

5.1. Basic hypotheses. Fundamental discrete spaces

We suppose that Ω is a bounded polygonal domain of R^2 . With T_h a standard finite element triangulation of Ω , and h the maximal length of the edges of the triangles of T_h , we introduce the following discrete spaces (with P_k = space of the polynomials in two variables of degree $\leq k$):

$$H_h^1 = \{ q_h \mid q_h \in C^0(\bar{\Omega}), \quad q_h|_T \in P_1, \quad \forall T \in T_h \}, \quad (5.1)$$

$$V_h = \{ \mathbf{v}_h \mid \mathbf{v}_h \in C^0(\bar{\Omega}) \times C^0(\bar{\Omega}), \quad \mathbf{v}_h|_T \in P_2 \times P_2, \quad \forall T \in T_h \}, \quad (5.2)$$

$$V_{0h} = \{ \mathbf{v}_h \mid \mathbf{v}_h \in V_h, \quad \mathbf{v}_h = \mathbf{0} \quad \text{on} \quad \Gamma \} = V_h \cap (H_0^1(\Omega))^2. \quad (5.3)$$

Two useful variants of V_h (and V_{0h}) are obtained as follows: either

$$V_h = \{ \mathbf{v}_h \mid \mathbf{v}_h \in C^0(\bar{\Omega}) \times C^0(\bar{\Omega}), \quad \mathbf{v}_h|_T \in P_1 \times P_1, \quad \forall T \in \tilde{T}_h \}, \quad (5.4)$$

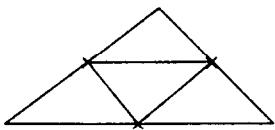


Fig. 5.1.

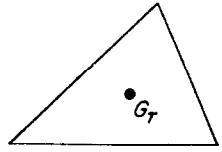


Fig. 5.2.

or (this space has been introduced in ref. [47])

$$V_h = \left\{ \mathbf{v}_h \mid \mathbf{v}_h \in C^0(\bar{\Omega}) \times C^0(\bar{\Omega}), \quad \mathbf{v}_h|_T \in P_{1T}^* \times P_{1T}^*, \quad \forall T \in T_h \right\}. \quad (5.5)$$

In (5.4), \tilde{T}_h is the triangulation of Ω obtained from T_h by joining the midpoints of the edges of $T \in T_h$, as shown in fig. 5.1. We have the same global number of unknowns if we use V_h defined by either (5.2) or (5.4), however, the matrices encountered in the second case are more compact and sparse.

In (5.5), P_{1T}^* is the subspace of P_3 defined as follows

$$\begin{cases} P_{1T}^* = \{ q \mid q = q_1 + \lambda \phi_T, \quad \text{with} \quad q_1 \in P_1, \lambda \in \mathbb{R}, \quad \text{and} \\ \phi_T \in P_3, \quad \phi_T = 0 \quad \text{on} \quad \partial T, \quad \phi_T(G_T) = 1 \}, \end{cases} \quad (5.6)$$

where, in (5.6), G_T is the centroid of T (see fig. 5.2 above). A function like ϕ_T is usually called a bubble-function.

5.2. Approximation of the boundary conditions

If the boundary conditions are defined by

$$\mathbf{u} = \mathbf{g} \quad \text{on} \quad \Gamma, \quad \int_{\Gamma} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0, \quad (5.7)$$

it is of fundamental importance to approximate \mathbf{g} by \mathbf{g}_h such that

$$\int_{\Gamma} \mathbf{g}_h \cdot \mathbf{n} \, d\Gamma = 0. \quad (5.8)$$

Let's discuss the construction of such a \mathbf{g}_h (we follow here ref. [1, appendix 3]). For simplicity, we shall suppose that \mathbf{g} is continuous over Γ . We now define the space γV_h as

$$\gamma V_h = \{ \mu_h \mid \mu_h = \mathbf{v}_h|_{\Gamma}, \mathbf{v}_h \in V_h \}, \quad (5.9)$$

i.e., γV_h is the space of the traces on Γ of those functions \mathbf{v}_h belonging to V_h . Actually, if V_h is defined by (5.2), γV_h is also the space of those functions defined over Γ , taking their values in \mathbb{R}^2 , continuous over Γ and piecewise quadratic over the edges of T_h contained in Γ .

Our problem is to construct an approximation \mathbf{g}_h of \mathbf{g} such that

$$\mathbf{g}_h \in \gamma V_h, \quad \int_{\Gamma} \mathbf{g}_h \cdot \mathbf{n} \, d\Gamma = 0. \quad (5.10)$$

If $\Pi_h \mathbf{g}$ is the unique element of γV_h , obtained by piecewise quadratic interpolation of \mathbf{g} over Γ , i.e., obtained from the values taken by \mathbf{g} at those nodes of T_h belonging to Γ , we usually have $\int_{\Gamma} \Pi_h \mathbf{g} \cdot \mathbf{n} \, d\Gamma \neq 0$. To overcome this difficulty, we may proceed as follows:

(i) We define an approximation \mathbf{n}_h of \mathbf{n} as the solution of the following linear variational problem in γV_h :

$$\mathbf{n}_h \in \gamma V_h, \quad \int_{\Gamma} \mathbf{n}_h \cdot \boldsymbol{\mu}_h \, d\Gamma = \int_{\Gamma} \mathbf{n} \cdot \boldsymbol{\mu}_h \, d\Gamma, \quad \forall \boldsymbol{\mu}_h \in \gamma V_h. \quad (5.11)$$

Problem (5.11) is, in fact, equivalent to a linear system whose matrix is sparse, positive definite, well-conditioned and quite easy to compute.

(ii) Then define \mathbf{g}_h by

$$\mathbf{g}_h = \Pi_h \mathbf{g} - \left(\int_{\Gamma} \Pi_h \mathbf{g} \cdot \mathbf{n} \, d\Gamma / \int_{\Gamma} \mathbf{n} \cdot \mathbf{n}_h \, d\Gamma \right) \mathbf{n}_h. \quad (5.12)$$

It is easy to check that (5.11), (5.12) imply (5.10).

5.3. Space approximation of the time dependent Navier–Stokes equations

Using the spaces H_h^1 , V_h and V_{0h} , we approximate the time dependent Navier–Stokes equations as follows:

Find $\{\mathbf{u}_h(t), p_h(t)\} \in V_h \times H_h^1$, $\forall t \geq 0$, such that

$$\begin{aligned} & \int_{\Omega} \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{v}_h \, dx + \nu \int_{\Omega} \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h \, dx + \int_{\Omega} (\mathbf{u}_h \cdot \nabla) \mathbf{u}_h \cdot \mathbf{v}_h \, dx + \int_{\Omega} \nabla p_h \cdot \mathbf{v}_h \, dx \\ &= \int_{\Omega} \mathbf{f}_h \cdot \mathbf{v}_h \, dx, \quad \forall \mathbf{v}_h \in V_{0h}, \end{aligned} \quad (5.13)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u}_h q_h \, dx = 0, \quad \forall q_h \in H_h^1, \quad (5.14)$$

$$\mathbf{u}_h = \mathbf{g}_h \quad \text{on } \Gamma, \quad (5.15)$$

$$\mathbf{u}_h(x, 0) = \mathbf{u}_{0h} \quad (\text{with } \mathbf{u}_{0h} \in V_h). \quad (5.16)$$

In (5.13)–(5.16), \mathbf{f}_h , \mathbf{u}_{0h} and \mathbf{g}_h are convenient approximations of \mathbf{f} , \mathbf{u}_0 and \mathbf{g} , respectively.

5.4. Time discretization by operator splitting methods

We consider now a fully discrete version of scheme (2.31)–(2.33) discussed in section 2.3.1. It is defined as follows (with Δt as in section 2.3.1):

$$\mathbf{u}_h^0 = \mathbf{u}_{0h}; \quad (5.17)$$

then for $n \geq 0$, compute (from \mathbf{u}_h^n) $\{\mathbf{u}_h^{n+1/2}, p_h^{n+1/2}\} \in \mathbf{V}_h \times \mathbf{H}_h^1$, and then $\mathbf{u}_h^{n+1} \in \mathbf{V}_h$, by solving

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\mathbf{u}_h^{n+1/2} - \mathbf{u}_h^n}{\Delta t/2} \cdot \mathbf{v}_h \, dx + \frac{\nu}{2} \int_{\Omega} \nabla \mathbf{u}_h^{n+1/2} \cdot \nabla \mathbf{v}_h \, dx + \int_{\Omega} \nabla p_h^{n+1/2} \cdot \mathbf{v}_h \, dx \\ = \int_{\Omega} \mathbf{f}_h^{n+1/2} \cdot \mathbf{v}_h \, dx - \frac{\nu}{2} \int_{\Omega} \nabla \mathbf{u}_h^n \cdot \nabla \mathbf{v}_h \, dx - \int_{\Omega} (\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^n \cdot \mathbf{v}_h \, dx, \\ \forall \mathbf{v}_h \in \mathbf{V}_{0h}, \end{array} \right. \quad (5.18a)$$

$$\int_{\Omega} \nabla \cdot \mathbf{u}_h^{n+1/2} q_h \, dx = 0, \quad \forall q_h \in \mathbf{H}_h^1, \quad (5.18b)$$

$$\mathbf{u}_h^{n+1/2} \in \mathbf{V}_h, \quad p_h^{n+1/2} \in \mathbf{H}_h^1, \quad \mathbf{u}_h^{n+1/2} = \mathbf{g}_h^{n+1/2} \quad \text{on } \Gamma, \quad (5.18c)$$

and then

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^{n+1/2}}{\Delta t/2} \cdot \mathbf{v}_h \, dx + \frac{\nu}{2} \int_{\Omega} \nabla \mathbf{u}_h^{n+1} \cdot \nabla \mathbf{v}_h \, dx + \int_{\Omega} (\mathbf{u}_h^{n+1} \cdot \nabla) \mathbf{u}_h^{n+1} \cdot \mathbf{v}_h \, dx \\ = \int_{\Omega} \mathbf{f}_h^{n+1} \cdot \mathbf{v}_h \, dx - \frac{\nu}{2} \int_{\Omega} \nabla \mathbf{u}_h^{n+1/2} \cdot \nabla \mathbf{v}_h \, dx - \int_{\Omega} \nabla p_h^{n+1/2} \cdot \mathbf{v}_h \, dx, \quad \forall \mathbf{v}_h \in \mathbf{V}_{0h}, \end{array} \right. \quad (5.19a)$$

$$\mathbf{u}_h^{n+1} \in \mathbf{V}_h, \quad \mathbf{u}_h^{n+1} = \mathbf{g}_h^{n+1} \quad \text{on } \Gamma, \quad (5.19b)$$

respectively. The same techniques apply to the space discretization of scheme (2.34)–(2.37).

The solution of the subproblems encountered at each step of (5.17)–(5.19) can be obtained by discrete variants of the methods discussed in sections 3 and 4; see ref. [1] (and also ref. [48]) for more details about this part of the solution process.

6. Numerical solution of the incompressible Navier–Stokes equations founded on the stream function–vorticity formulation

6.1. Generalities. Synopsis

These last years have seen a significant decrease in the popularity of the solution methods for the Navier–Stokes equations using the stream function–vorticity formulation. We see two main

reasons for this phenomenon:

- (i) These methods are really convenient for two-dimensional flows. The generalization to three-dimensional flows, although possible, leads to complicated formulations.
- (ii) The treatment of the boundary conditions is more delicate than with the velocity–pressure formulation.

We have decided, however, to dedicate a full section to the stream function–vorticity formulation of the Navier–Stokes equations since an abundant amount of literature exists on this approach and because the difficulties associated with the boundary conditions can be largely overcome nowadays (see, e.g. ref. [49] for the numerical treatment of the boundary conditions in the stream function–vorticity formulation of the Stokes problem).

In section 6.2, we shall derive the stream function–vorticity formulation from the velocity–pressure model. In section 6.3, we shall give a variational formulation of the new equations, most useful in view of the time and space discretizations which are to be discussed in sections 6.4 and 6.5, respectively. Some numerical results obtained by using the stream function–vorticity formulation will be presented in section 8.

6.2. The stream function–vorticity formulation of the Navier–Stokes equations for incompressible viscous flows

We use here the notation of section 1. We denote by Ω the two-dimensional flow region and denote by Γ its boundary. We suppose that Ω is q -connected with q a nonnegative integer. The possible holes (corresponding to obstacles to the flow) are denoted by ω_k , $k = 1, \dots, q$. If we denote by Γ the boundary of Ω , we have (with the notation of fig. 6.1, where $q = 3$)

$$\Gamma = \bigcup_{k=0}^q \Gamma_k.$$

From section 1, unsteady flows in Ω of Newtonian, incompressible viscous fluids are modeled by the following Navier–Stokes equations (written here in the velocity–pressure formulation)

$$\partial \mathbf{u} / \partial t - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \quad (6.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \quad (6.2)$$

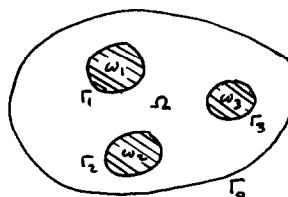


Fig. 6.1.

with the initial condition

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x) \quad (\text{with } \nabla \cdot \mathbf{u}_0 = 0). \quad (6.3)$$

Concerning the boundary conditions, we shall consider only Dirichlet boundary conditions for the velocity, i.e.,

$$\mathbf{u} = \mathbf{g} \quad \text{on} \quad \Gamma. \quad (6.4)$$

From the incompressibility property, \mathbf{g} has to satisfy

$$\int_{\Gamma} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0.$$

In fact, we shall suppose that, for simplicity,

$$\int_{\Gamma_k} \mathbf{g} \cdot \mathbf{n} \, d\Gamma = 0, \quad \forall k = 0, 1, \dots, q. \quad (6.5)$$

In order to derive a stream function–vorticity formulation from (6.1)–(6.4), we observe first that the incompressibility property $\nabla \cdot \mathbf{u} = 0$ implies the existence of a (non-unique) vector $\psi = \{0, 0, \psi\}$ such that

$$\mathbf{u} = \nabla \times \psi,$$

i.e.,

$$u_1 = \partial \psi / \partial x_2, \quad u_2 = -\partial \psi / \partial x_1. \quad (6.6)$$

A function such as ψ is called a stream function since the (obvious) relation

$$\mathbf{u} \cdot \nabla \psi = 0$$

implies that ψ is constant along the streamlines.

If we define now $\omega = \{0, 0, \omega\}$ by

$$\omega = \nabla \times \mathbf{u}, \quad (6.7)$$

we clearly have

$$\omega = \partial u_2 / \partial x_1 - \partial u_1 / \partial x_2, \quad (6.8)$$

which implies in turn (combined with (6.6))

$$-\Delta \psi = \omega. \quad (6.9)$$

The Poisson equation relating the stream function ψ to the vorticity ω will be completed by the time dependent equation describing the evolution of ω . To obtain such an equation, let's take the curl of the momentum equation (6.1). We obtain then (with $\mathbf{f} = \{f_1, f_2\}$):

$$\frac{\partial \omega}{\partial t} - \nu \Delta \omega + \frac{\partial \psi}{\partial x_2} \frac{\partial \omega}{\partial x_1} - \frac{\partial \psi}{\partial x_1} \frac{\partial \omega}{\partial x_2} = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \quad (6.10)$$

to which we associate the initial condition

$$\omega(x, 0) = \omega_0(x) \quad (6.11)$$

where $\{0, 0, \omega_0\} = \nabla \times \mathbf{u}_0$.

Let's derive now the boundary conditions associated with ω and ψ . At this stage, it is convenient to denote by n_1, n_2 the two coordinates of \mathbf{n} and to introduce

$$\tau = \{-n_2, n_1\},$$

which is a vector of unit length, tangent to Γ . We have then

$$\begin{aligned} \frac{\partial \psi}{\partial n} &= \frac{\partial \psi}{\partial x_1} n_1 + \frac{\partial \psi}{\partial x_2} n_2 = u_1 n_2 - u_2 n_1 \\ &= -(u_1 \tau_1 + u_2 \tau_2) = -\mathbf{g} \cdot \tau \quad \text{on } \Gamma. \end{aligned}$$

Similarly, we should prove that

$$\begin{aligned} \frac{\partial \psi}{\partial \tau} &= \frac{\partial \psi}{\partial x_1} \tau_1 + \frac{\partial \psi}{\partial x_2} \tau_2 = u_1 n_1 + u_2 n_2 \\ &= \mathbf{u} \cdot \mathbf{n} = \mathbf{g} \cdot \mathbf{n} \quad \text{on } \Gamma. \end{aligned}$$

Thus ψ satisfies the two following boundary conditions:

$$\frac{\partial \psi}{\partial n} = -\mathbf{g} \cdot \tau \quad \text{on } \Gamma, \quad (6.12)$$

$$\frac{\partial \psi}{\partial \tau} = \mathbf{g} \cdot \mathbf{n} \quad \text{on } \Gamma. \quad (6.13)$$

We observe that there is no simple boundary condition associated with ω .

To summarize, the pair $\{\omega, \psi\}$ satisfies the following relations:

$$\frac{\partial \omega}{\partial t} - \nu \Delta \omega + \frac{\partial \psi}{\partial x_2} \frac{\partial \omega}{\partial x_1} - \frac{\partial \psi}{\partial x_1} \frac{\partial \omega}{\partial x_2} = \frac{\partial f_2}{\partial x_1} - \frac{\partial f_1}{\partial x_2} \quad \text{in } \Omega, \quad (6.14a)$$

$$-\Delta \psi = \omega, \quad (6.14b)$$

$$\omega(x, 0) = \omega_0(x), \quad (6.14c)$$

$$\frac{\partial \psi}{\partial n} = -\mathbf{g} \cdot \boldsymbol{\tau} \quad \text{on } \Gamma, \quad (6.14d)$$

$$\frac{\partial \psi}{\partial \boldsymbol{\tau}} = \mathbf{g} \cdot \mathbf{n} \quad \text{on } \Gamma. \quad (6.14e)$$

We observe that

$$\frac{\partial \psi}{\partial x_2} \frac{\partial \omega}{\partial x_1} - \frac{\partial \psi}{\partial x_1} \frac{\partial \omega}{\partial x_2} = \mathbf{u} \cdot \nabla \omega.$$

In the next section, we shall discuss a variational formulation of (6.14) which is well suited to time and space discretization.

6.3. Variational formulation of problem (6.14)

Multiplying (6.14a) (respectively (6.14b)) by $\theta \in H_0^1(\Omega)$ (respectively $\phi \in H^1(\Omega)$) and applying Green's formula, we obtain

$$\begin{aligned} & \int_{\Omega} \frac{\partial \omega}{\partial t} \theta \, dx + v \int_{\Omega} \nabla \omega \cdot \nabla \theta \, dx + \int_{\Omega} \left(\frac{\partial \psi}{\partial x_2} \frac{\partial \omega}{\partial x_1} - \frac{\partial \psi}{\partial x_1} \frac{\partial \omega}{\partial x_2} \right) \theta \, dx \\ &= \int_{\Omega} \left(f_1 \frac{\partial \theta}{\partial x_2} - f_2 \frac{\partial \theta}{\partial x_1} \right) \, dx, \quad \forall \theta \in H_0^1(\Omega), \end{aligned} \quad (6.15a)$$

$$\omega(x, 0) = \omega_0(x), \quad (6.15b)$$

$$\int_{\Omega} \nabla \psi \cdot \nabla \phi \, dx = \int_{\Omega} \omega \phi \, dx - \int_{\Gamma} \mathbf{g} \cdot \boldsymbol{\tau} \phi \, d\Gamma, \quad \forall \phi \in H^1(\Omega), \quad (6.15c)$$

$$\frac{\partial \psi}{\partial \boldsymbol{\tau}} = \mathbf{g} \cdot \mathbf{n} \quad \text{on } \Gamma, \quad (6.15d)$$

which is equivalent to (6.14).

6.4. Time discretization of problem (6.14), (6.15)

6.4.1. Generalities. Synopsis

We shall discuss in this section the time discretization of problem (6.14), (6.15) by a semi-implicit scheme, which is quite easy to implement in combination with finite element approximations for the space discretization. Fully implicit schemes are described in ref. [50].

Before describing the time discretization of problem (6.14), (6.15), it is convenient to “localize” ψ . To achieve such a goal, we define γ_k , $k = 0, \dots, q$, by

$$\gamma_k(m) = \int_{\widetilde{M}_m} \mathbf{g} \cdot \mathbf{n} \, d\Gamma, \quad \forall m \in \Gamma_k, \quad (6.16)$$

where M_k is an arbitrary point of Γ_k . The function γ_k vanishes at M_k and is, from (6.5), continuous on Γ_k . Since ψ is defined within an arbitrary constant, we require

$$\psi = \gamma_0 \quad \text{on} \quad \Gamma_0 \quad (6.17)$$

to fix this constant.

We introduce now, the two spaces W_0 and W_g defined as follows:

$$\left\{ \begin{array}{l} W_0 = \{ \phi \mid \phi \in H^1(\Omega), \quad \phi = 0 \quad \text{on} \quad \Gamma_0, \quad \partial\phi/\partial\tau = 0 \quad \text{on} \quad \Gamma_k, \\ \forall k = 1, \dots, q \}, \end{array} \right. \quad (6.18)$$

$$\left\{ \begin{array}{l} W_g = \{ \phi \mid \phi \in H^1(\Omega), \quad \phi = \gamma_0 \quad \text{on} \quad \Gamma_0, \quad \partial\phi/\partial\tau = \mathbf{g} \cdot \mathbf{n} \quad \text{on} \quad \Gamma_k, \\ \forall k = 1, \dots, q \}. \end{array} \right. \quad (6.19)$$

Both spaces W_0 and W_g are closed in $H^1(\Omega)$.

Alternative definitions of W_0 and W_g are given by

$$\left\{ \begin{array}{l} W_0 = \{ \phi \mid \phi \in H^1(\Omega), \quad \phi = \gamma_0 \quad \text{on} \quad \Gamma_0, \quad \phi = \text{constant} \\ \quad \text{on} \quad \Gamma_k, \quad \forall k = 1, \dots, q \}, \end{array} \right. \quad (6.20)$$

$$\left\{ \begin{array}{l} W_g = \{ \phi \mid \phi \in H^1(\Omega), \quad \phi = \gamma_0 \quad \text{on} \quad \Gamma_0, \quad \phi = \gamma_k + \text{constant} \\ \quad \text{on} \quad \Gamma_k, \quad \forall k = 1, \dots, q \}. \end{array} \right. \quad (6.21)$$

It has to be understood that the constants occurring in (6.20), (6.21), are different according to k .

6.4.2. Description of the semi-implicit scheme

Let $\Delta t > 0$ be a time discretization step and let's denote by ω^n and ψ^n approximate values of ω and ψ at time $n\Delta t$. We approximate (6.14), (6.15) by

$$\omega^0 = \omega_0, \quad (6.22)$$

$$\lambda^0 = \omega_0|_{\Gamma}, \quad (6.23)$$

$$\mathbf{u}^0 = \mathbf{u}_0; \quad (6.24)$$

then for $n \geq 0$, assuming that ω^n , λ^n , \mathbf{u}^n are known, we compute ω^{n+1} , λ^{n+1} , \mathbf{u}^{n+1} as follows: solve

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\omega^{n+1} - \omega^n}{\Delta t} \theta \, dx + \nu \int_{\Omega} \nabla \omega^{n+1} \cdot \nabla \theta \, dx + \int_{\Omega} (\mathbf{u}^n \cdot \nabla \omega^{n+1}) \theta \, dx \\ = \int_{\Omega} \left(f_1^{n+1} \frac{\partial \theta}{\partial x_2} - f_2^{n+1} \frac{\partial \theta}{\partial x_1} \right) \, dx, \quad \forall \theta \in H_0^1(\Omega); \\ \omega^{n+1} = \lambda^n \quad \text{on} \quad \Gamma \end{array} \right. \quad (6.25)$$

and then

$$\begin{cases} \psi^{n+1} \in W_{g^{n+1}}, \quad \forall \phi \in W_0, \quad \text{we have} \\ \int_{\Omega} \nabla \psi^{n+1} \cdot \nabla \phi \, dx = \int_{\Omega} \omega^{n+1} \phi \, dx - \int_{\Gamma} \mathbf{g}^{n+1} \cdot \boldsymbol{\tau} \phi \, d\Gamma. \end{cases} \quad (6.26)$$

To update λ^n , solve

$$\begin{cases} \lambda^{n+1} \in M, \quad \forall \mu \in M \quad \text{we have} \\ \int_{\Omega} \lambda^{n+1} \mu \, dx = \int_{\Omega} \nabla \psi^{n+1} \cdot \nabla \mu \, dx + \int_{\Gamma} \mathbf{g}^{n+1} \cdot \boldsymbol{\tau} \mu \, d\Gamma, \end{cases} \quad (6.27)$$

and finally, define \mathbf{u}^{n+1} by

$$\mathbf{u}^{n+1} = \left\{ \partial \psi^{n+1} / \partial x_2, - \partial \psi^{n+1} / \partial x_1 \right\}. \quad \square \quad (6.28)$$

In (6.26), $\mathbf{g}^{n+1} = \mathbf{g}((n+1)\Delta t)$ and $W_{g^{n+1}}$ is the space defined by (6.21) when $\mathbf{g} = \mathbf{g}^{n+1}$. In (6.27), M is a complementary suspace of $H_0^1(\Omega)$ in $H^1(\Omega)$ (i.e. $H^1(\Omega) = H_0^1(\Omega) \oplus M$).

Assuming that \mathbf{g}^{n+1} , \mathbf{f}^{n+1} and λ^n are sufficiently smooth, it is quite easy to show that problems (6.25) and (6.26) are well-posed (problem (6.27) is more delicate to analyze). Concerning problems (6.25), (6.26), we have the following:

Proposition 6.1: Suppose that λ^n , \mathbf{u}^n , \mathbf{f}^{n+1} and \mathbf{g}^{n+1} are sufficiently smooth. Then problems (6.25) and (6.26) have a unique solution in $H^1(\Omega)$ and $W_{g^{n+1}}$, respectively.

Proof: Let's discuss first the solution of the Dirichlet problem (6.25). From the properties of λ^n , \mathbf{u}^n and \mathbf{f}^{n+1} , problem (6.25) will have a unique solution in $H^1(\Omega)$ if we can prove that the bilinear functional

$$\{ \theta, \theta' \} \rightarrow \int_{\Omega} \theta \theta' \, dx + \nu \Delta t \int_{\Omega} \nabla \theta \cdot \nabla \theta' \, dx + \Delta t \int_{\Omega} (\mathbf{u}^n \cdot \nabla \theta) \theta' \, dx \quad (6.29)$$

is $H_0^1(\Omega)$ -elliptic, i.e. there exists an $\alpha > 0$ such that

$$\int_{\Omega} \theta^2 \, dx + \nu \Delta t \int_{\Omega} |\nabla \theta|^2 \, dx + \Delta t \int_{\Omega} (\mathbf{u}^n \cdot \nabla \theta) \theta \, dx \geq \alpha \|\theta\|_{H_0^1(\Omega)}^2, \quad \forall \theta \in H_0^1(\Omega). \quad (6.30)$$

Since $\nabla \cdot \mathbf{u}^n = 0$, we clearly have, $\forall \theta, \theta' \in H_0^1(\Omega)$

$$\int_{\Omega} (\mathbf{u}^n \cdot \nabla \theta) \theta' \, dx = - \int_{\Omega} (\mathbf{u}^n \cdot \nabla \theta') \theta \, dx,$$

which implies in turn that

$$\int_{\Omega} (\mathbf{u}^n \cdot \nabla \theta) \theta \, dx = 0, \quad \forall \theta \in H_0^1(\Omega). \quad (6.31)$$

From (6.31), the bilinear functional defined by (6.29) is clearly $H_0^1(\Omega)$ -elliptic (with, e.g., $\alpha = \nu\Delta t$). Since this functional is also continuous, the existence and uniqueness property follows from a simple variation of the Lax–Milgram Theorem (see, e.g. refs. [51] and [1, appendix 1] for more details on the solution of the nonhomogeneous Dirichlet problem by variational methods).

Let's consider now problem (6.26). Dropping the superscript $n + 1$ in (6.29), we introduce $\psi_g \in W_g$ such that

$$\psi_g = \gamma_k \quad \text{on} \quad \Gamma_k, \quad \forall k = 0, \dots, q. \quad (6.32)$$

If \mathbf{g} is sufficiently smooth, W_g is not empty and therefore, such a ψ_g exists.

Now let ψ be the (hypothetical) solution of (6.26). Define $\bar{\psi}$ by

$$\bar{\psi} = \psi - \psi_g. \quad (6.33)$$

We clearly have that $\bar{\psi}$ satisfies

$$\begin{cases} \bar{\psi} \in W_0, \quad \forall \phi \in W_0, \quad \text{we have} \\ \int_{\Omega} \nabla \bar{\psi} \cdot \nabla \phi \, dx = \int_{\Omega} \omega \phi \, dx - \int_{\Omega} \nabla \psi_g \cdot \nabla \phi \, dx - \int_{\Gamma} \mathbf{g} \cdot \boldsymbol{\tau} \phi \, d\Gamma. \end{cases} \quad (6.34)$$

The linear functional on the right-hand side of (6.34) is clearly continuous over $H^1(\Omega)$ and therefore, over W_0 . On the other hand, since Ω is bounded and the functions of W_0 vanish on Γ_0 , the bilinear functional

$$\{\phi, \phi'\} \rightarrow \int_{\Omega} \nabla \phi \cdot \nabla \phi' \, dx \quad (6.35)$$

is W_0 -elliptic, W_0 being equipped with the $H^1(\Omega)$ -norm. Since the functional (6.35) is obviously continuous over $W_0 \times W_0$, it follows from the Lax–Milgram theorem that problem (6.34) has a unique solution, implying in turn that problem (6.26) also has a solution, possibly dependent on ψ_g . In fact, the W_0 -ellipticity of the functional (6.35) implies the uniqueness of the solution of (6.26).

Remark 2.1: We can prove the existence of solutions of (6.25) (possibly not in $H^1(\Omega)$) with weaker assumptions on λ^n than those in the statement of proposition 6.1, where we implicitly assumed that λ^n was the trace of a function of $H^1(\Omega)$.

Remark 6.2: The above techniques and results can be generalized to cases where ψ and $\partial\psi/\partial n$ (respectively $\partial\psi/\partial n$ and $\partial\omega/\partial n$) are given on a subset Γ_A (respectively Γ_B) of Γ , such that $\Gamma_A \cap \Gamma_B = \emptyset$, $\Gamma_A \cup \Gamma_B = \Gamma$.

6.5. A finite element method for the space discretization of problem (6.14), (6.15)

In this section, we shall assume that Ω is a bounded, polygonal domain of R^2 . As in section 5, we denote by T_h a finite element triangulation of domain Ω . We associate with T_h the following

finite dimensional spaces:

$$\mathbf{H}_h^1 = \left\{ \phi_h / \phi_h \in C^0(\bar{\Omega}), \quad \phi_h|_T \in P_1, \quad \forall T \in \mathbf{T}_h \right\}, \quad (6.36)$$

$$\mathbf{H}_{0h}^1 = \mathbf{H}_h^1 \cap \mathbf{H}_0^1(\Omega) = \left\{ \phi_h \mid \phi_h \in \mathbf{H}_h^1, \quad \phi_h = 0 \quad \text{on} \quad \Gamma \right\}. \quad (6.37)$$

In (6.36), P_l is the space of the polynomials in x_1, x_2 of degree $\leq l$ (in practice, we should use $l = 1$ or 2). We shall also need \mathbf{M}_h defined by

$$\begin{cases} \mathbf{M}_h = \left\{ \mu_h \mid \mu_h \in \mathbf{H}_h^1, \quad \mu_h|_T = 0, \quad \forall T \in \mathbf{T}_h \right. \\ \left. \partial T \cap \Gamma = \emptyset \right\}. \end{cases} \quad (6.38)$$

We clearly have

$$\mathbf{H}_{0h}^1 \oplus \mathbf{M}_h = \mathbf{H}_h^1. \quad (6.39)$$

We approximate then (6.14), (6.15) by the following set of coupled algebraic and differential equations (written in variational form):

$$\begin{aligned} & \int_{\Omega} \frac{\partial \omega_h}{\partial t} \theta_h \, dx + \nu \int_{\Omega} \nabla \omega_h \cdot \nabla \theta_h \, dx + \int_{\Omega} \left(\frac{\partial \psi_h}{\partial x_2} \frac{\partial \omega_h}{\partial x_1} - \frac{\partial \psi_h}{\partial x_1} \frac{\partial \omega_h}{\partial x_2} \right) \theta_h \, dx \\ &= \int_{\Omega} \left(f_{1h} \frac{\partial \theta_h}{\partial x_2} - f_{2h} \frac{\partial \theta_h}{\partial x_1} \right) \, dx, \quad \forall \theta_h \in \mathbf{H}_{0h}^1, \end{aligned} \quad (6.40a)$$

$$\omega_h(x, 0) = \omega_{0h}(x) \quad (\text{with } \omega_{0h} \in \mathbf{H}_h^1), \quad (6.40b)$$

$$\int_{\Omega} \nabla \psi_h \cdot \nabla \phi_h \, dx = \int_{\Omega} \omega_h \phi_h \, dx - \int_{\Gamma} \mathbf{g}_h \cdot \boldsymbol{\tau} \phi_h \, d\Gamma, \quad (6.40c)$$

$$\partial \psi_h / \partial \boldsymbol{\tau} = \mathbf{g}_h \cdot \mathbf{n} \quad \text{on} \quad \Gamma, \quad (6.40d)$$

$$\psi_h(t), \omega_h(t) \in \mathbf{H}_h^1. \quad (6.40e)$$

In (6.40a)–(6.40d), f_{1h} , f_{2h} , ω_{0h} and \mathbf{g}_h are approximations of f_1 , f_2 , ω_0 and \mathbf{g} , respectively, with \mathbf{g}_h satisfying

$$\int_{\Gamma_k} \mathbf{g}_h \cdot \mathbf{n} \, d\tau = 0, \quad \forall k = 0, 1, \dots, q, \quad (6.41)$$

and compatible with the fact that $\psi_h|_T \in P_1$.

Using the approach of Section 6.4, we obtain a fully discrete approximation of problem (6.14), (6.15), defined as follows:

$$\omega_h^0 = \omega_{0h}, \quad (6.42)$$

$$\lambda_h^0 \in \mathbf{M}_h, \quad \lambda_h^0 = \omega_{0h} \quad \text{on} \quad \Gamma, \quad (6.43)$$

$$\mathbf{u}_h^0 = \mathbf{u}_{0h}; \quad (6.44)$$

then for $n \geq 0$, assuming that ω_h^n , λ_h^n , \mathbf{u}_h^n are known, we compute ω_h^{n+1} , ψ_h^{n+1} , λ_h^{n+1} , \mathbf{u}_h^{n+1} as follows:

Solve:

$$\begin{cases} \int_{\Omega} \frac{\omega_h^{n+1} - \omega_h^n}{\Delta t} \theta_h \, dx + \nu \int_{\Omega} \nabla \omega_h^{n+1} \cdot \nabla \theta_h \, dx + \int_{\Omega} (\mathbf{u}_h^n \cdot \nabla \omega_h^{n+1}) \theta_h \, dx \\ = \int_{\Omega} \left(f_{1h}^{n+1} \frac{\partial \theta_h}{\partial x_2} - f_{2h}^{n+1} \frac{\partial \theta_h}{\partial x_1} \right) \, dx, \quad \forall \theta_h \in \mathbf{H}_{0h}^1, \\ \omega_h^{n+1} - \lambda_h^n \in \mathbf{H}_{0h}^1, \end{cases} \quad (6.45)$$

and then

$$\begin{cases} \psi_h^{n+1} \in \mathbf{W}_{g^{n+1}h}; \quad \forall \phi_h \in \mathbf{W}_{0h} \quad \text{we have} \\ \int_{\Omega} \nabla \psi_h^{n+1} \cdot \nabla \phi_h \, dx = \int_{\Omega} \omega_h^{n+1} \phi_h \, dx - \int_{\Gamma} \mathbf{g}_h^{n+1} \cdot \tau \phi_h \, d\Gamma. \end{cases} \quad (6.46)$$

To update λ_h^n , solve

$$\begin{cases} \lambda_h^{n+1} \in \mathbf{M}_h; \quad \forall \mu_h \in \mathbf{M}_h \quad \text{we have} \\ \int_{\Omega} \lambda_h^{n+1} \mu_h \, dx = \int_{\Omega} \nabla \psi_h^{n+1} \cdot \nabla \mu_h \, dx + \int_{\Gamma} \mathbf{g}_h^{n+1} \cdot \tau \mu_h \, d\Gamma. \end{cases} \quad (6.47)$$

Finally, define \mathbf{u}_h^{n+1} by

$$\mathbf{u}_h^{n+1} = \left\{ \partial \psi_h^{n+1} / \partial x_2, - \partial \psi_h^{n+1} / \partial x_1 \right\}. \quad \square \quad (6.48)$$

In (6.44) (respectively (6.46)), \mathbf{u}_{0h} is an approximation of the initial velocity \mathbf{u}_0 (respectively $\mathbf{g}_h^{n+1} = \mathbf{g}_h((n+1)\Delta t)$). Moreover, in (6.46), \mathbf{W}_{0h} and $\mathbf{W}_{g^{n+1}h}$ are defined by

$$\begin{cases} \mathbf{W}_{0h} = \left\{ \phi_h \mid \phi_h \in \mathbf{H}_h^1, \quad \phi_h = 0 \quad \text{on} \quad \Gamma_0, \partial \phi_h / \partial \tau = 0 \quad \text{on} \quad \Gamma_k, \right. \\ \left. \forall k = 1, \dots, q \right\}, \end{cases} \quad (6.49)$$

$$\begin{cases} \mathbf{W}_{g^{n+1}h} = \left\{ \phi_h \mid \phi_h \in \mathbf{H}_h^1, \quad \phi_h = \gamma_{0h}^{n+1} \quad \text{on} \quad \Gamma_0, \partial \phi_h / \partial \tau = \mathbf{g}_h^{n+1} \cdot \mathbf{n} \right. \\ \left. \text{on} \quad \Gamma_k, \quad \forall k = 1, \dots, q \right\}, \end{cases} \quad (6.50)$$

with γ_{0h}^{n+1} defined by (6.16) (with \mathbf{g} replaced by \mathbf{g}_h^{n+1}).

Before discussing the solution of the linear problems (6.45), (6.46), (6.47), it is necessary to make some remarks concerning scheme (6.42)–(6.48).

Remark 6.3: In the particular case where $l = 1$ in (6.36) (piecewise linear approximations for ω

and ψ), it is quite convenient to replace the L^2 -scalar product occurring in (6.45), (6.46), (6.47) by the one (denoted $(\cdot, \cdot)_h$) defined by

$$(\theta_h, \phi_h)_h = \frac{1}{3} \sum_{P \in \Sigma_h} m_P \theta_h(P) \phi_h(P), \quad \forall \theta_h, \phi_h \in H_h^1, \quad (6.51)$$

where m_P is the measure of the polygonal union of those triangles of T_h with P as a common vertex. The scalar product (6.51) is clearly obtained from

$$\int_{\Omega} \theta_h \phi_h \, dx = \sum_{T \in T_h} \int_T \theta_h \phi_h \, dx \quad (6.52)$$

by applying the trapezoidal rule on each triangle on the right-hand side of (6.52). Similarly, we should use the trapezoidal rule to evaluate the boundary integrals in (6.46), (6.47).

For regular triangulations, applying the above numerical integration techniques goes back to classical finite difference schemes for discretizing the stream function–vorticity formulation of the Navier–Stokes equations.

Remark 6.4: Scheme (6.42)–(6.48) is of the semi-implicit type. Actually, it is quite easy to make it “more” implicit (and at the same time more stable and accurate), by using, instead of (6.45)–(6.48), the following iterative method:

$$\lambda_h^{n+1,0} = \lambda_h^n, \quad \mathbf{u}_h^{n+1,0} = \mathbf{u}_h^n; \quad (6.53)$$

then for $m \geq 0$, compute $\omega_h^{n+1,m+1}$, $\psi_h^{n+1,m+1}$, $\lambda_h^{n+1,m+1}$, $\mathbf{u}_h^{n+1,m+1}$ as follows:

Solve:

$$\left\{ \begin{array}{l} \int_{\Omega} \frac{\omega_h^{n+1,m+1} - \omega_h^n}{\Delta t} \theta_h \, dx + \nu \int_{\Omega} \nabla \omega_h^{n+1,m+1} \cdot \nabla \theta_h \, dx \\ \quad + \int_{\Omega} (\mathbf{u}_h^{n+1,m} \cdot \nabla \omega_h^{n+1,m+1}) \theta_h \, dx = \int_{\Omega} \left(f_{1h}^{n+1} \frac{\partial \theta_h}{\partial x_2} - f_{2h}^{n+1} \frac{\partial \theta_h}{\partial x_1} \right) \, dx, \\ \forall \theta_h \in H_{0h}^1, \\ \omega_h^{n+1,m+1} - \lambda_h^{n+1,m} \in H_{0h}^1, \end{array} \right. \quad (6.54)$$

and then

$$\left\{ \begin{array}{l} \psi_h^{n+1,m+1} \in W_{g^{n+1}h}; \quad \forall \phi_h \in W_{0h} \quad \text{we have} \\ \int_{\Omega} \nabla \psi_h^{n+1,m+1} \cdot \nabla \phi_h \, dx = \int_{\Omega} \omega_h^{n+1,m+1} \phi_h \, dx - \int_{\Gamma} \mathbf{g}_h^{n+1} \cdot \tau \mu_h \, d\Gamma. \end{array} \right. \quad (6.55)$$

To update $\lambda_h^{n+1,m}$ solve

$$\begin{cases} \lambda_h^{n+1,m+1} \in M_h; \quad \forall \mu_h \in M_h \quad \text{we have} \\ \int_{\Omega} \lambda_h^{n+1,m+1} \mu_h = \int_{\Omega} \nabla \psi_h^{n+1,m+1} \cdot \nabla \mu_h \, dx + \int_{\Gamma} \mathbf{g}_h^{n+1} \cdot \boldsymbol{\tau} \mu_h \, d\Gamma, \end{cases} \quad (6.56)$$

and finally, define $\mathbf{u}_h^{n+1,m+1}$ by

$$\mathbf{u}_h^{n+1,m+1} = \left\{ \partial \psi_h^{n+1,m+1} / \partial x_2, - \partial \psi_h^{n+1,m+1} / \partial x_1 \right\}. \quad \square \quad (6.57)$$

Do $m = m + 1$ and go to (6.53).

The functions ω_h^{n+1} , ψ_h^{n+1} , λ_h^{n+1} , \mathbf{u}_h^{n+1} will be obtained as the limits of $\omega_h^{n+1,m}$, $\psi_h^{n+1,m}$, $\lambda_h^{n+1,m}$, $\mathbf{u}_h^{n+1,m}$ as $m \rightarrow +\infty$.

Some comments concerning the solution of the finite dimensional problems (6.45), (6.46), (6.47):

(Indeed, the following comments will also apply to the solution of problems (6.54), (6.55), (6.56)), Problems (6.45), (6.47) are classical finite dimensional variational problems. To solve them, we should substitute for θ_h (respectively μ_h) the elements of a vector basis of H_{0h}^1 (respectively M_h) to construct equivalent linear systems. These systems can then be solved by classical direct or iterative methods. Problem (6.46) is less classical. To solve it, let's define first Σ_h (respectively Σ_{0h}) as the set of the vertices of T_h (respectively, the vertices of T_h not located on Γ). To each $P \in \Sigma_h$, associate then w_P such that

$$\begin{cases} w_P \in H_h^1, \\ w_P(P) = 1; \quad w_P(Q) = 0, \quad \forall Q \in \Sigma_h, \quad Q \neq P. \end{cases} \quad (6.58)$$

The following sets

$$\begin{cases} B_h = \{w_P\}_{P \in \Sigma_h}, \quad B_{0h} = \{w_P\}_{P \in \Sigma_{0h}}, \\ B_h \setminus B_{0h} (= \{w_P\}_{P \in \Sigma_h \setminus \Sigma_{0h}}). \end{cases} \quad (6.59)$$

are vector bases of H_h^1 , H_{0h}^1 , M_h , respectively.

Finally, in order to construct a vector basis of W_{0h} , we associate to each Γ_k , $k = 1, \dots, q$, a function w'_k defined as follows:

$$\begin{cases} w'_k \in M_h, \quad w'_k|_{\Gamma_k} = 1, \\ w'_k|_{\Gamma_l} = 0, \quad \forall l \neq k, \quad 0 \leq l \leq q. \end{cases} \quad (6.60)$$

It is quite obvious that

$$\mathbf{B}_{0h} \cup \{w'_k\}_{k=1}^q \quad (6.61)$$

is a vector basis of \mathbf{W}_{0h} . To compute ψ_h^{n+1} from (6.46), we first observe that (6.46) is equivalent to

$$\begin{cases} \int_{\Omega} \nabla \psi_h^{n+1} \cdot \nabla w_P \, dx = \int_{\Omega} \omega_h^{n+1} w_P \, dx, \\ \forall P \in \Sigma_{0h}, \end{cases} \quad (6.62a)$$

$$\begin{cases} \int_{\Omega} \nabla \psi_h^{n+1} \cdot \nabla w'_k \, dx = \int_{\Omega} \omega_h^{n+1} w'_k \, dx - \int_{\Gamma_k} \mathbf{g}_h^{n+1} \cdot \tau w'_k \, d\Gamma, \\ \forall k = 1, \dots, q, \end{cases} \quad (6.62b)$$

$$\psi_h^{n+1} = \sum_{Q \in \Sigma_{0h}} \psi_h(Q) w_Q + \sum_{l=1}^q C_l w'_l + \sum_{l=0}^q \sum_{Q \in \Sigma_h \cap \Gamma_l} \gamma_{lh}(Q) w_Q, \quad (6.62c)$$

where in the right-hand side of (6.62), we have dropped the superscript $n+1$ and where γ_{lh} is defined by (6.16) (with \mathbf{g} replaced by \mathbf{g}_h^{n+1}).

By substituting the expansion (6.62c) of ψ_h^{n+1} into (6.62a) (6.62b) we obtain $\{\psi_h(Q)\}_{Q \in \Sigma_{0h}} \cup \{C_l\}_{l=1}^q$ as the solution of a linear system whose matrix is symmetric and positive definite.

7. A backward method of characteristics for the incompressible Navier–Stokes equations. Implementation of downstream boundary conditions for flows in unbounded domains

7.1. Synopsis

Upwinding techniques to treat the advection term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ have always been quite popular among those solving the Navier–Stokes equations. For their finite difference (respectively, finite element) implementation see, e.g. refs. [11,12] (respectively, refs. [9,10,52]) and the references therein. The Petrov–Galerkin methods advocated by Hugues and his co-authors [53,54] also belong to this class of methods. It is, however, the opinion of the authors of this report that the most elegant way to “upwind” the Navier–Stokes equations is to use those backward methods of characteristics popularized by the investigators of the Laboratoire National d’Hydraulique (L.N.H.) at Electricité de France (see refs. [37,55] for further details and also ref. [56] for related techniques). In the following section 7.2, we shall briefly discuss a backward method of characteristics. Section 7.3 will be devoted to the finite element implementation of downstream boundary conditions which, from our numerical experiments, appear to be more absorbing than those of Dirichlet or Neumann types discussed in section 1.

7.2. Solution of the Navier–Stokes equations by a backward method of characteristics

Let's define the (nonlinear) differential operator D/Dt by

$$\frac{D\phi}{Dt} = \frac{\partial\phi}{\partial t} + \sum_{j=1}^N u_j \frac{\partial\phi}{\partial x_j}, \quad \forall \phi \text{ sufficiently smooth,} \quad (7.1)$$

where $D\phi/Dt$ is the Lagrangian time derivative of ϕ . With this notation, the momentum equation (1.1) takes the following form:

$$Du/Dt - \nu \Delta u + \nabla p = f \quad \text{in } \Omega. \quad (7.2)$$

Now, to $\{x, t\} \in \Omega \times \mathbb{R}_+$, we associate a flow $X(x, t; \tau)$ (the characteristic flow) defined by

$$\begin{cases} dX/d\tau = u(X, \tau), \\ X(x, t; t) = x. \end{cases} \quad (7.3)$$

A simple time discretization of the Navier–Stokes equations using (7.2), (7.3) (i.e. obtained by integrating along the characteristic curves defined by (7.3)) is defined as follows:

$$u^0 = u_0, \quad (7.4a)$$

then for $n \geq 0$, assuming that u^n is known,

$$\frac{u^{n+1}(x) - u^n(X^n(x; n \Delta t))}{\Delta t} - \nu \Delta u^{n+1}(x) + \nabla p^{n+1}(x) = f^{n+1}(x), \quad (7.4b)$$

$$\nabla \cdot u^{n+1} = 0, \quad (7.4c)$$

$$u^{n+1} = g^{n+1} \quad \text{on } \Gamma, \quad (7.4d)$$

where, in (7.4b), the flow X^n is defined as the solution of the (autonomous) differential system

$$\begin{cases} \frac{dX}{d\tau} = u^n(X), \\ X(x; (n+1) \Delta t) = x. \end{cases} \quad (7.5)$$

The Stokes problem (7.4b)–(7.4d) can be solved by the methods described in section 4.

Remark 7.1: Eq. (7.4b) is slightly incorrect. Actually, if the curve described by $X^n(x; \tau)$ for $\tau \in]n\Delta t, (n+1)\Delta t[$, is contained in Ω , we shall use (7.4b) to discretize (1.1) at time $(n+1)\Delta t$. But, if there exists τ_n^* , $n\Delta t < \tau_n^* < (n+1)\Delta t$ such that

$$X^n(x; \tau_n^*) \in \Gamma,$$

we shall discretize (1.1) at $\{x, (n+1)\Delta t\}$ by

$$\frac{\mathbf{u}^{n+1}(x) - \mathbf{u}^n(X^n(x; \tau_n^*), \tau_n^*)}{(n+1) \Delta t - \tau_n^*} - \nu \Delta \mathbf{u}^{n+1}(x) + \nabla p^{n+1}(x) = \mathbf{f}^{n+1}(x). \quad \square \quad (7.6)$$

The finite element implementation of the above method is not difficult, in principle, particularly with the finite element approximation described in section 5 (see refs. [37,55,56] for more details and the results of numerical experiments).

7.3. Finite element implementation of downstream boundary conditions for flows in unbounded domains

One of the major problems associated with the numerical simulation of incompressible viscous flows (at high Reynolds numbers particularly) is to find a convenient set of downstream boundary conditions when the physical unbounded domain is replaced by a bounded computational one.

A partial answer to this question can be found in ref. [57], but the methods discussed in this last reference still have to be further analyzed in view of their computational implementation. In this section, we shall discuss some downstream boundary conditions whose practical implementation is quite easy through the finite element approximation introduced in section 5.

For flows at high Reynolds numbers (which is one of our main concerns), several authors have suggested to take as downstream boundary conditions the following inviscid momentum equation (where for simplicity we have taken $\mathbf{f} = \mathbf{0}$)

$$\partial \mathbf{u} / \partial t + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \mathbf{0}. \quad (7.7)$$

Let's describe the practical implementation of condition (7.7) in the particular case of the flow problem associated with fig. 1.1 of section 1.1. Suppose that a finite element approximation * $\{\mathbf{u}_h^n, p_h^n\}$ of $\{\mathbf{u}, p\}$ is known over $\bar{\Omega}$ at time $n\Delta t$. We should predict $\check{\mathbf{u}}_h^{n+1}$ over Γ_∞ as follows:

(i) We first compute the trace $\check{\mathbf{g}}_h^{n+1}$ of \mathbf{u}_h^{n+1} defined by:

$$\begin{cases} \int_{\Omega} \frac{\check{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^n}{\Delta t} \cdot \mathbf{v}_h \, dx + \int_{\Omega} (\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^n \cdot \mathbf{v}_h \, dx + \int_{\Omega} \nabla p_h^n \cdot \mathbf{v}_h \, dx = 0, \\ \forall \mathbf{v}_h \in \mathbf{V}_h^+; \quad \check{\mathbf{u}}_h^{n+1} \in \mathbf{V}_h, \quad \check{\mathbf{u}}_h^{n+1} = \mathbf{u}_\infty((n+1) \Delta t) \quad \text{on} \quad \Gamma_\infty^-, \end{cases} \quad (7.8)$$

where

$$\mathbf{V}_h^+ = \{ \mathbf{v}_h \mid \mathbf{v}_h \in \mathbf{V}_h, \quad \mathbf{v}_h = \mathbf{0} \quad \text{on} \quad \Gamma_\infty^- \}.$$

In practice, we shall use numerical integration (trapezoidal rule for example) to approximate the first integral in (7.8), namely

$$\int_{\Omega} \frac{\check{\mathbf{u}}_h^{n+1} - \mathbf{u}_h^n}{\Delta t} \cdot \mathbf{v}_h \, dx,$$

* Derived from section 5.

in such a way that the corresponding mass matrix will be diagonal. Then we shall take for \mathbf{v}_h those basis functions of V_h associated with the nodes of T_h (vertices and mid-points if V_h is defined by either (5.2) or (5.4); vertices only if V_h is defined by (5.5)) belonging to $\dot{\Gamma}_\infty^+$ ($\dot{\Gamma}_\infty^+$: interior of Γ_∞^+).

(ii) Once the trace $\check{\mathbf{g}}_h^{n+1}$ of $\check{\mathbf{u}}_h^{n+1}$ is known, we obtain \mathbf{g}_h^{n+1} by correcting $\check{\mathbf{g}}_h^{n+1}$ in such a way that

$$\int_{\Gamma} \mathbf{g}_h^{n+1} \cdot \mathbf{n} \, d\Gamma = 0. \quad (7.9)$$

Such a correction can be obtained by the following variant of (5.12) (see section 5.2):

$$\mathbf{g}_h^{n+1} = \check{\mathbf{g}}_h^{n+1} - \left(\int_{\Gamma} \check{\mathbf{g}}_h^{n+1} \cdot \mathbf{n} \, d\Gamma \Big/ \int_{\Gamma} \mathbf{n} \cdot \mathbf{n}_h \, d\Gamma \right) \mathbf{n}_h, \quad (7.10)$$

with \mathbf{n}_h still defined by (5.11). In order to obtain relation (7.9), we can also use variants of (7.10) in which the correction of $\check{\mathbf{g}}_h^{n+1}$ is only done on Γ_∞^+ .

The above method to predict \mathbf{u}_h^{n+1} on Γ_∞^+ can easily be modified to be combined with the operator splitting methods discussed in sections 2.3 and 5.4.

The above technique has been tested for flows around cylinders and the corresponding numerical results will be reported in ref. [38], together with comparisons to other methods. From these first experiments it seems that these new boundary conditions are more “transparent” than the one obtained by just prescribing $\mathbf{u} = \mathbf{u}_\infty$ on Γ_∞ , or by mixing, as suggested by some authors, Dirichlet et Neumann boundary conditions. Actually, it seems that re-introducing some viscosity in (7.7) may improve the quality of the numerical results. Such modification of the technique described just above will be discussed in ref. [38], where associated numerical results will also be shown. (We can also take advantage of the method of characteristics to implement downstream boundary conditions with good absorbing properties; see, again, ref. [38] for details.)

8. Numerical simulation of some incompressible viscous flows

The main goal of this section is to present the results of numerical experiments concerning the simulation of two and three dimensional incompressible viscous flows. In section 8.1 we shall present the results of a plane jet simulation, and then in sections 8.2 and 8.3 the simulation of flows around and inside air intakes at high angle of attack.

8.1. Plane jet simulation

We consider a plane jet problem in which the flow domain Ω is the right half space

$$(x \mid x = (x_i)_{i=1}^2, \quad x_1 > 0, \quad x_2 \in \mathbb{R}).$$

We assume that the fluid viscosity ν is equal to $1/2000$, that the fluid is at rest at time $t = 0$, i.e.

$$\mathbf{u}(x, 0) = \mathbf{0}, \quad \forall x \in \Omega,$$

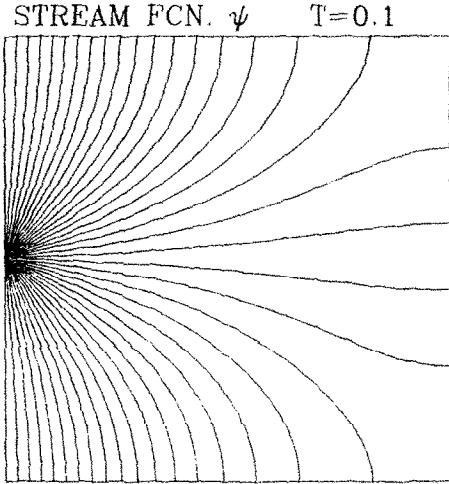


Fig. 8.1.

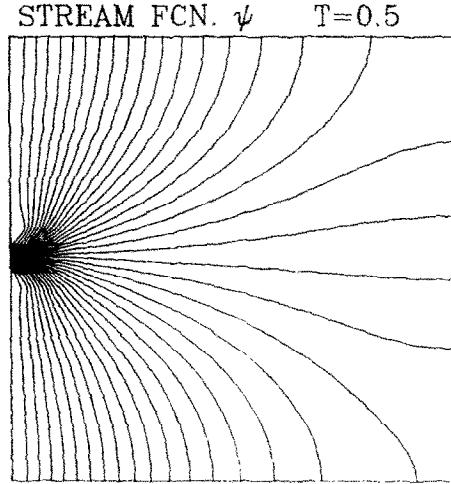


Fig. 8.2.

that the jet aperture is located on the line $x_1 = 0$, is centered at $x = (0, 0)$ and is $1/16$ wide. We suppose also that the jet profile is parabolic, with a maximal velocity equal to 1, the jet being horizontal at $x_1 = 0$ (we have then $\text{Re} = 125$).

We have used several computational domains, all defined by $]-L/2, L/2[\times]0, L[$, with L taking the values 1, 2 and e^2 , successively ($e = 2.718\dots$).

The flow simulation was done using stream function vorticity formulation of section 6, combined to a fully implicit (second order in time) variant of the scheme discussed in sections 6.4 and 6.5; the finite element discretization was achieved using piecewise linear approximations (i.e. $l = 1$ in (6.36)) for ω and ψ . The details of such an approximation may be found in ref. [50].

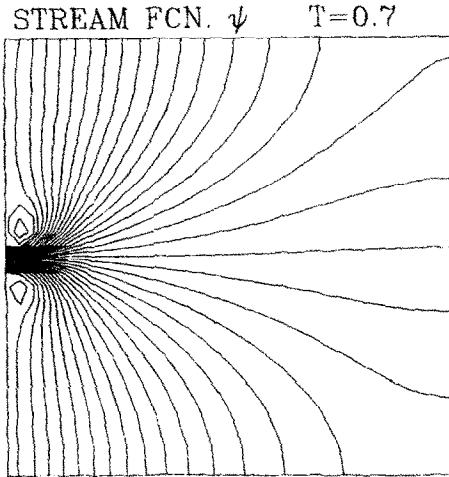


Fig. 8.3.

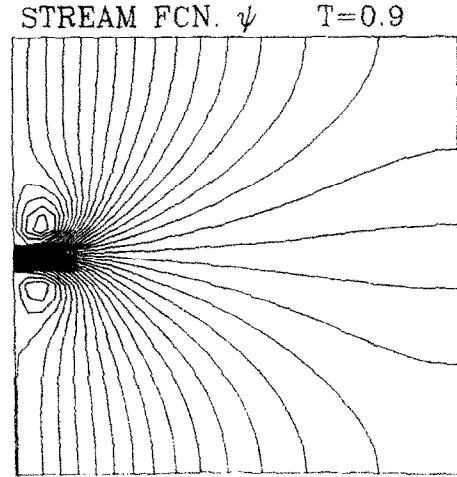


Fig. 8.4.

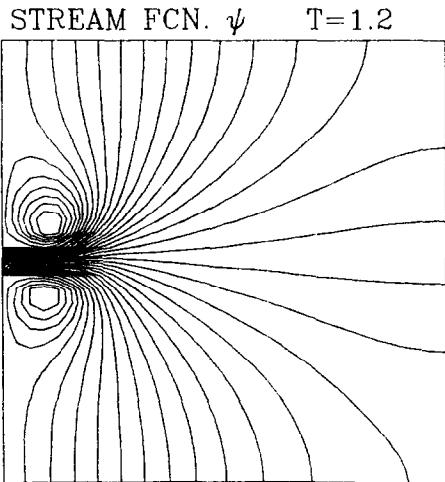


Fig. 8.5.

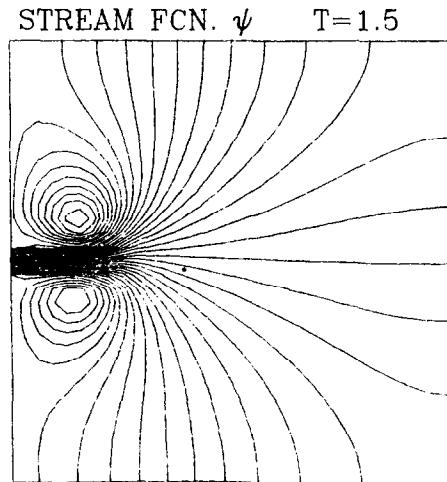


Fig. 8.6.

Taking $\Delta t = 0.1$, the following results have been obtained:

- (i) Figs. 8.1 to 8.8 ($t = 0.1, 0.5, 0.7, 0.9, 1.2, 1.5, 2.5, 4$) show the flow pattern (particularly the vortex creation) close to the jet aperture at different times. For these results we have been using $L = 1$ and a uniform finite element triangulation of 2048 triangles.
- (ii) Since at $t \approx 4$ the computational domain appears to be too small, we have been using $L = 2$ and a uniform finite element triangulation of 8192 triangles.

The corresponding results are reported in figs. 8.9 to 8.12 ($t = 4.5, 8.5, 8.9, 9.5$). These figures show some interesting phenomena such as a symmetry breaking occurring close to $t = 8.5$ and the creation of small eddies in the lower shear layer between $t = 8.5$ and 8.9. These secondary

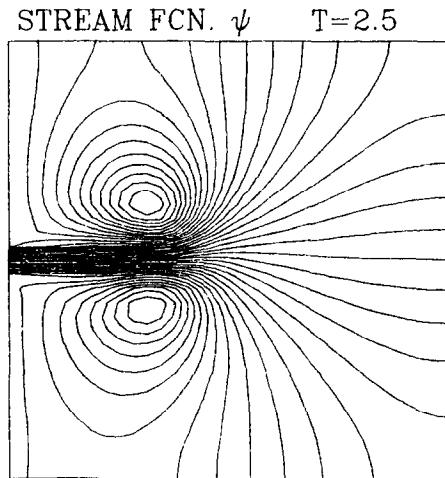


Fig. 8.7.

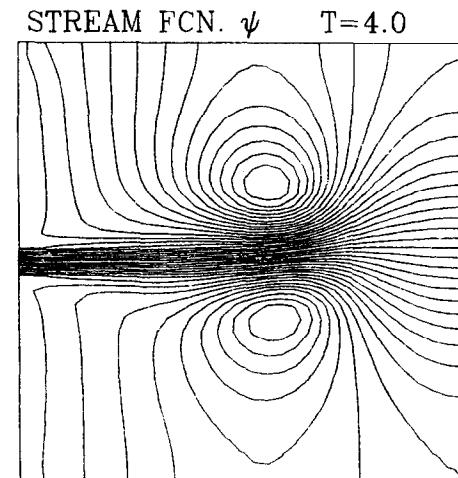


Fig. 8.8.

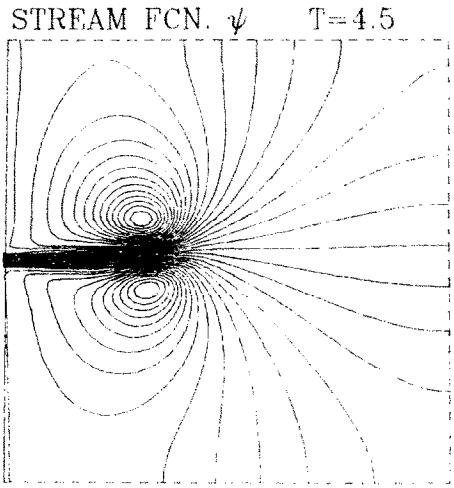


Fig. 8.9.

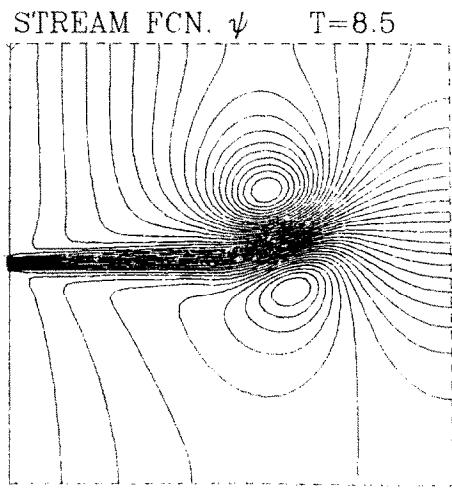


Fig. 8.10.

eddies remain if we decrease Δt and the space discretization step, and if we increase the size of the computational domain.

(iii) In order to check the influence of the mesh and of the size of the computational domain we have been running a numerical simulation with $L = e^2$ and the nonuniform finite element grid of fig. 8.13 (18432 triangles). The results confirm those of (i) and (ii); in Fig. 8.14, we have shown the flow pattern corresponding to $t = 13$.

8.2. Flow in a nozzle at high incidence

The second test problem that we consider is much more complicated than the first one. It concerns the simulation of an incompressible viscous flow inside and around a (two-dimensional)

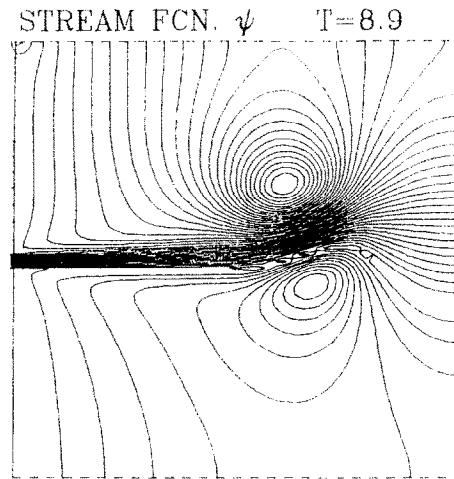


Fig. 8.11.

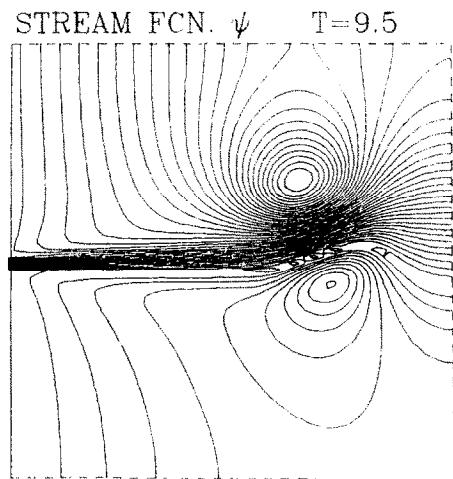


Fig. 8.12.

EXPONENTIAL GRID N=M=95

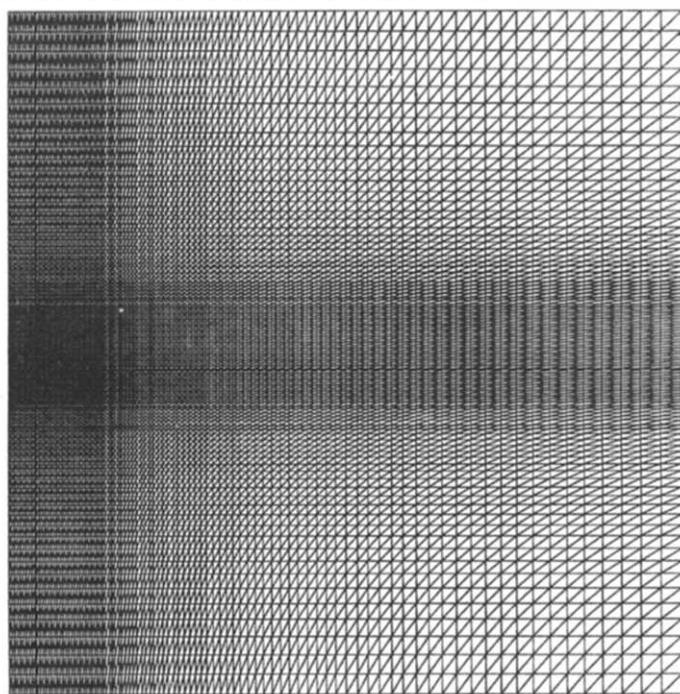


Fig. 8.13.

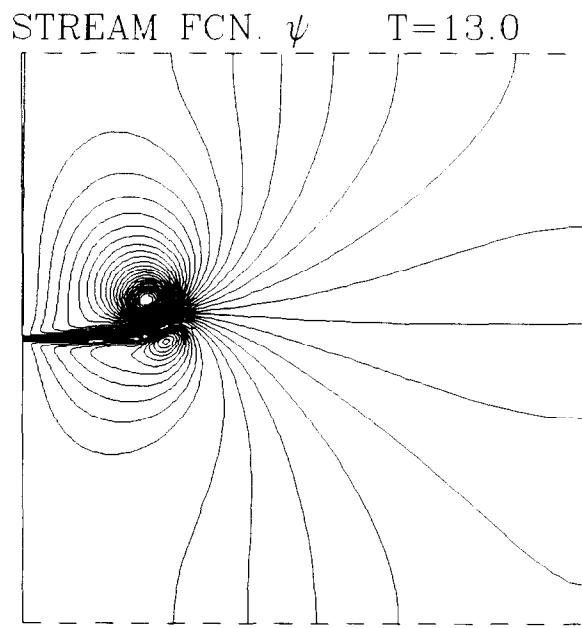


Fig. 8.14.

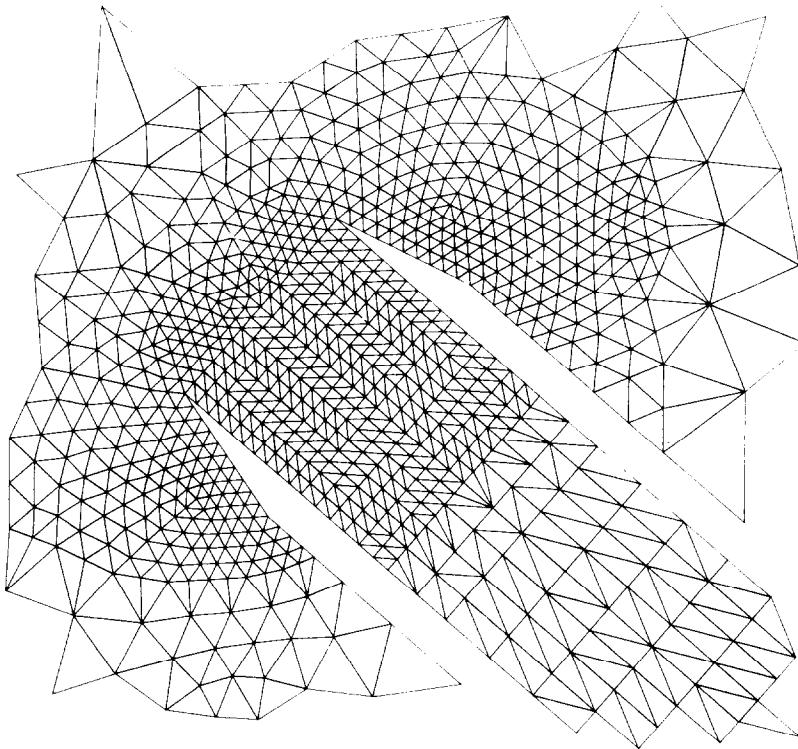


Fig. 8.15. Pressure grid.

nozzle at high incidence (40 degrees) and at $\text{Re} = 750$ (the characteristic length is taken as the distance between the walls of the nozzle). We used the same type of finite element approximations as in section 5, with V_h defined by (5.4). Figs. 8.15 and 8.16 show the details of the triangulations T_h and \tilde{T}_h , respectively, close to the air intake. Using $\Delta t = 0.05$, we obtained the streamlines and vortex pattern shown in figs. 8.17 to 8.21 ($t = 0., 2., 4., 6., 8.$). The initial velocity distribution is associated to the corresponding steady Stokes flow and a suction phenomenon is simulated inside the nozzle. We have shown in fig. 8.22 the visualization of experiments done at ONERA by Werle [89] for a Reynolds number of same order of magnitude and for the same angle of attack. We observe the good agreement between the computed and experimental results. In figs. 8.23 to 8.25 we have shown a color visualisation (with the flow coming from the left hand side) of the vorticity at $t = 4., 6., 8..$

A similar simulation has been done for $\text{Re} = 1500$. A finer grid was required for these calculations as shown in figs. 8.26 (pressure grid) and 8.27 (velocity grid). Using $\Delta t = 0.025$ we obtained the velocity configuration of figs. 8.28 ($t = 3.5$) and 8.29 ($t = 4.5$); a visualization of the vorticity at the same time cycles is shown in figs. 8.30 and 8.31.

8.3. Flow in a three-dimensional nozzle at high incidence

We consider here a flow inside and around a three-dimensional nozzle at $\text{Re} = 200$, for an angle of attack of 40 degrees. The corresponding simulations have been done using the θ -scheme

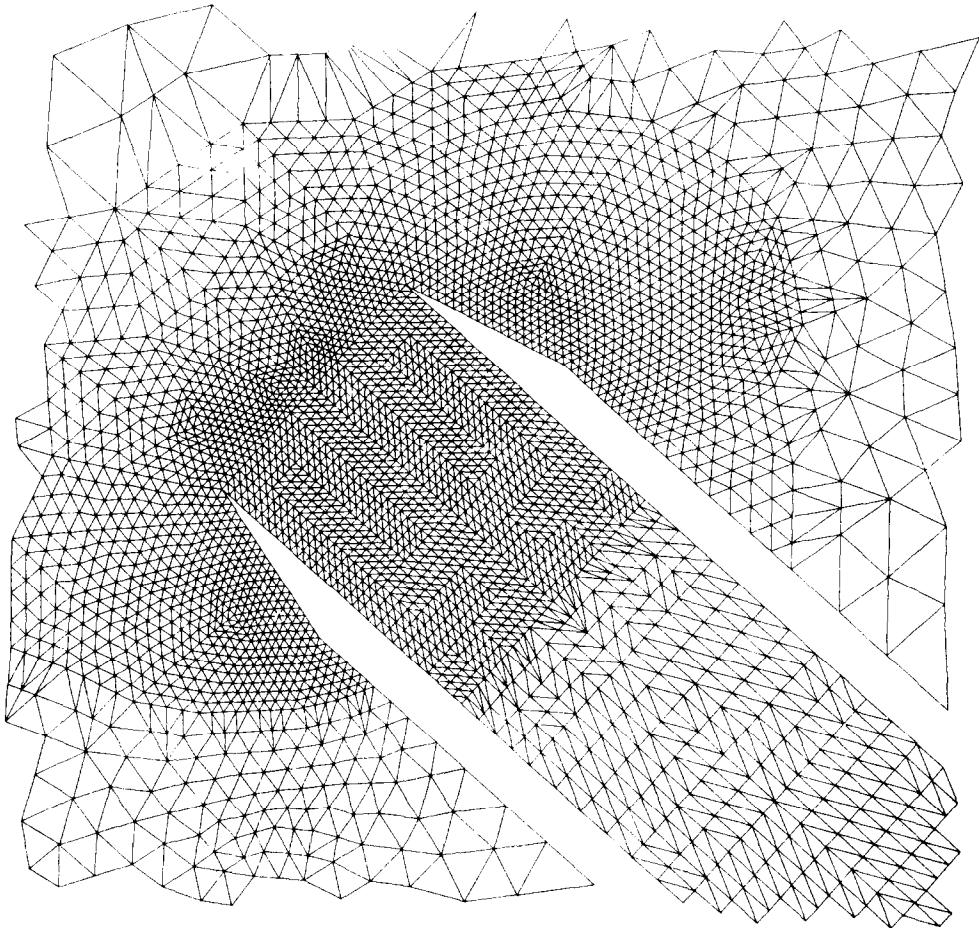


Fig. 8.16. Velocity grid.

(2.34)–(2.37) of section 2.3.2, with $\theta = 1 - \sqrt{2}/2$, combined to the three-dimensional variant of the finite element approximation of section 5 defined by (5.5) (bubble-function for the velocity, piecewise linear pressure). Indeed, a first calculation has been done during 30 time cycles (with $\Delta t = 0.2$), using the grid shown in fig. 8.32 (3096 nodes, 16632 tetrahedra). From these results the grid is locally refined (see fig. 8.33), using the techniques of section 15 (we have now 4632 nodes and 24457 tetrahedra), and the integration is performed with this new grid during 20 time cycles with the same Δt . The results obtained at $t = 10.$, without and with refinement, are shown in figs. 8.34 and 8.35. (Figs. 8.32, 8.33 on page 138, figs. 8.34, 8.35 on page 137.)

We observe an improved accuracy for the pressure and vorticity with the refined grid.

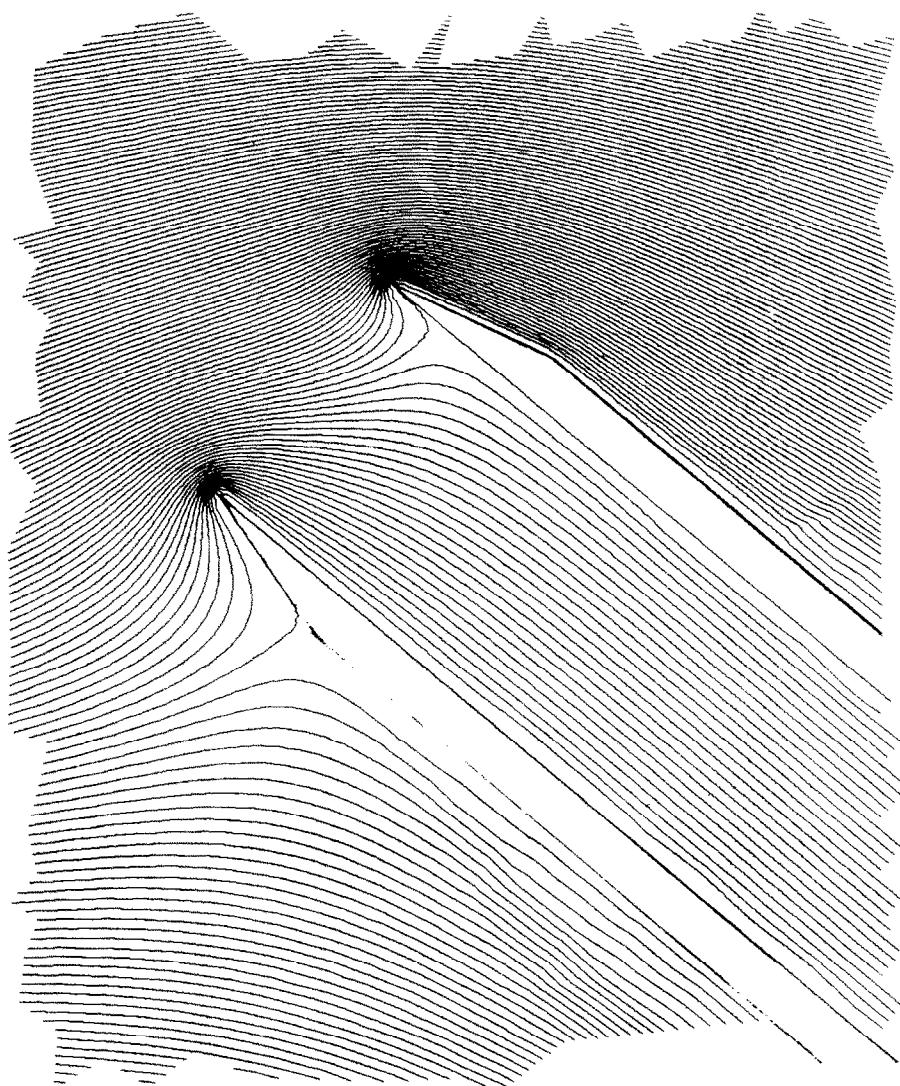


Fig. 8.17. $\text{Re} = 750$; $t = 0$.

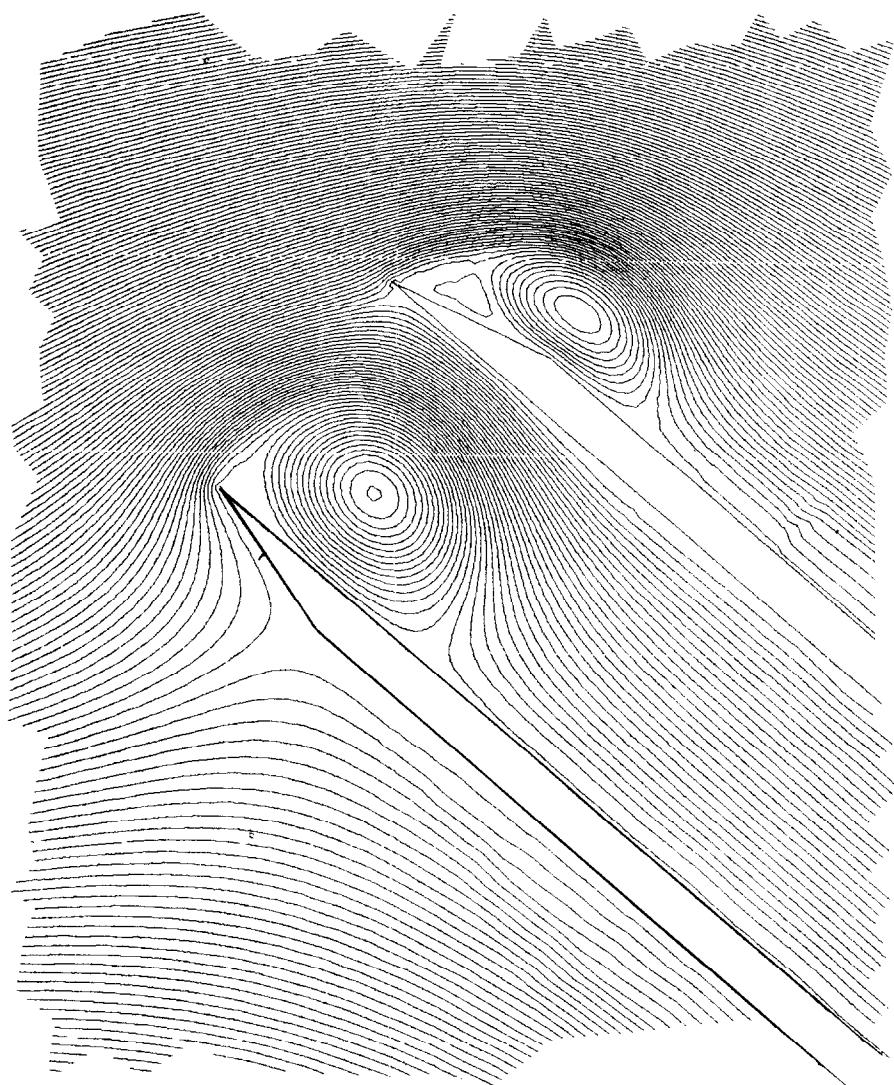
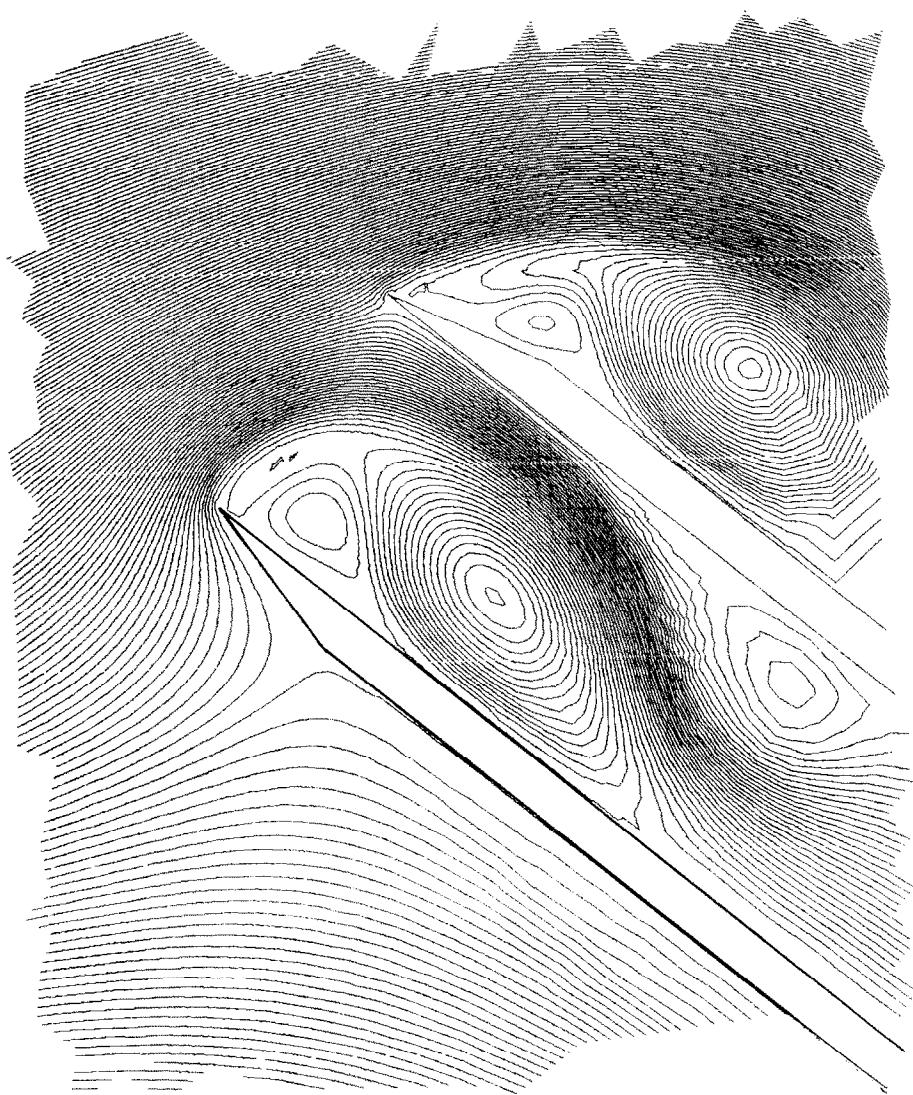


Fig. 8.18. $\text{Re} = 750$; $t = 2$.

Fig. 8.19. $\text{Re} = 750$; $t = 4$.

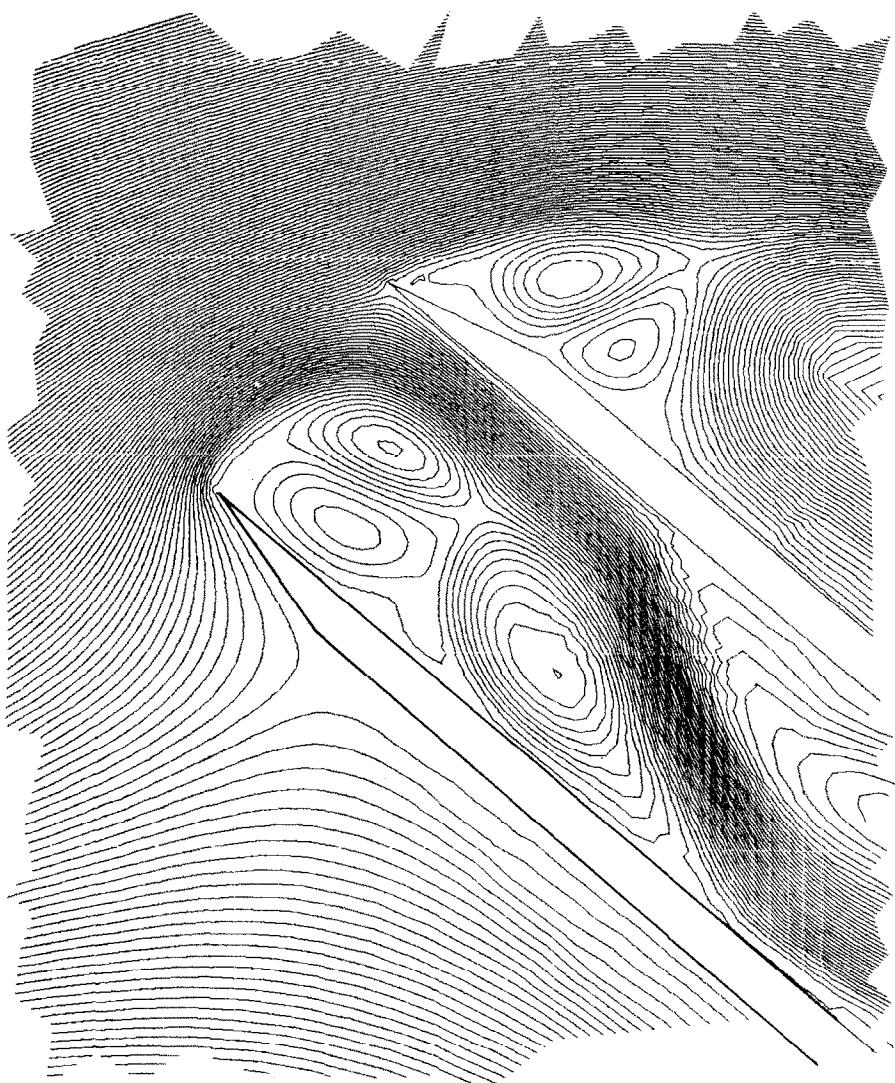


Fig. 8.20. $\text{Re} = 750$; $t = 6$.

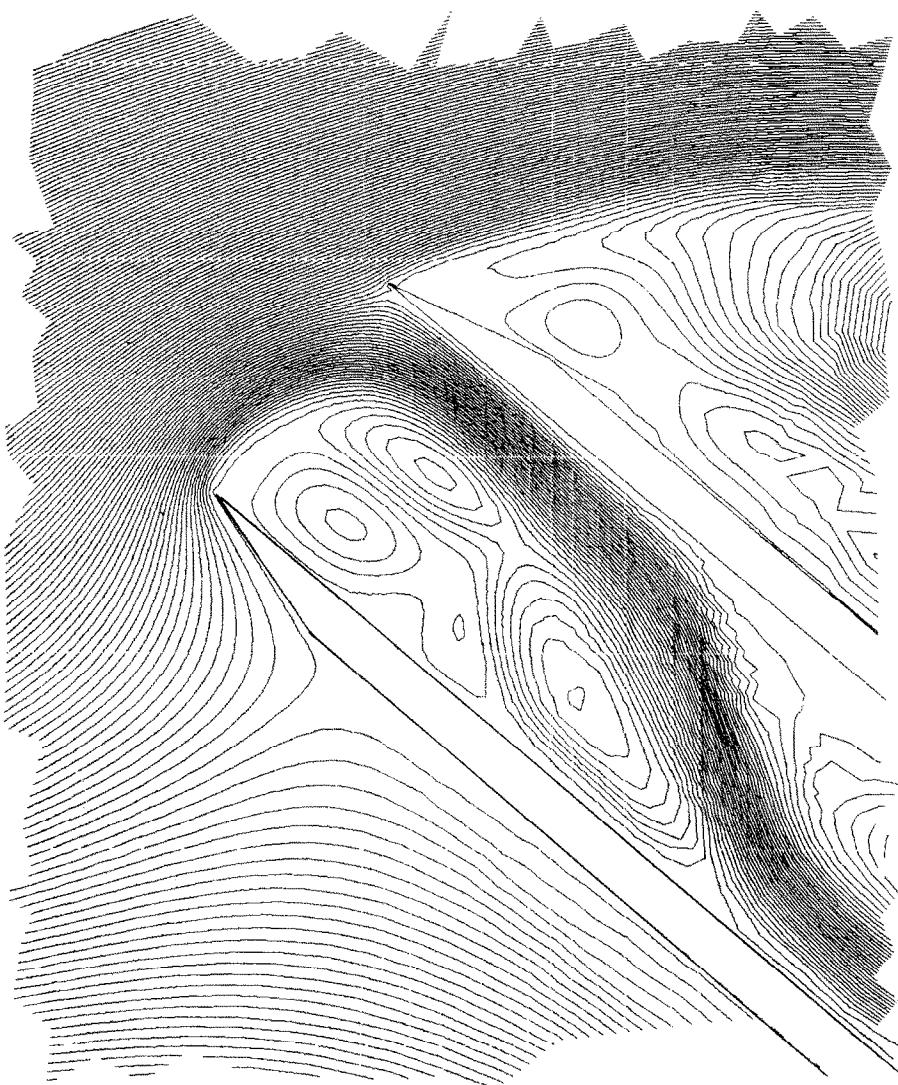


Fig. 8.21. $\text{Re} = 750$; $t = 8$.

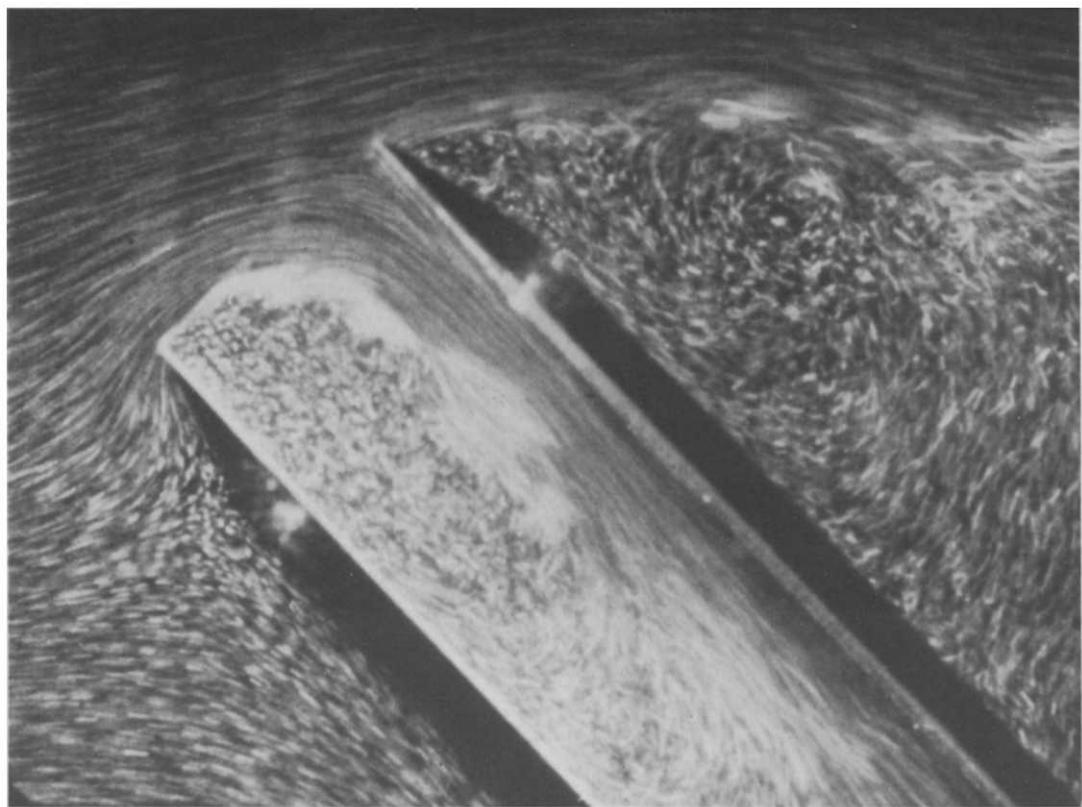


Fig. 8.22. ONERA experimental results: courtesy of DRET and H. Werle.

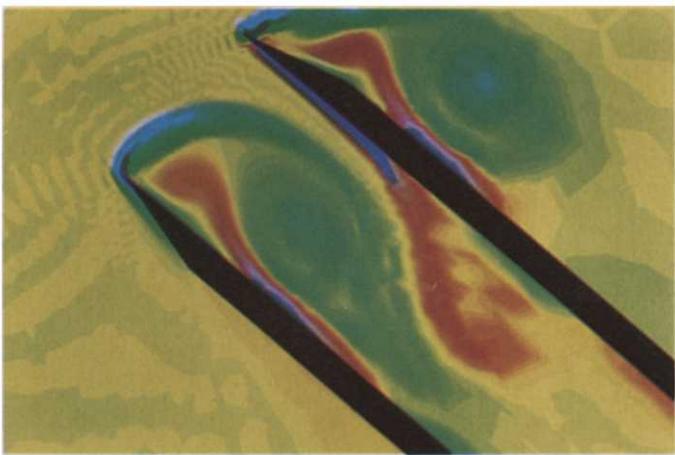


Fig. 8.23. Vorticity visualization, $\text{Re} = 750$; $t = 4$.

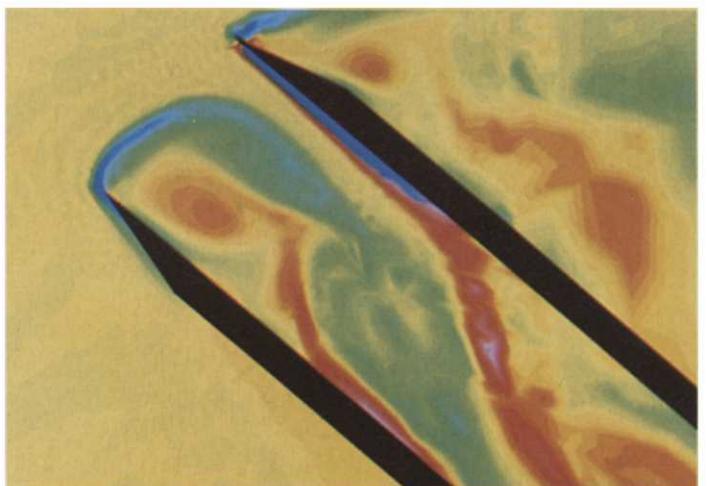


Fig. 8.24. Vorticity visualization, $\text{Re} = 750$; $t = 6$.

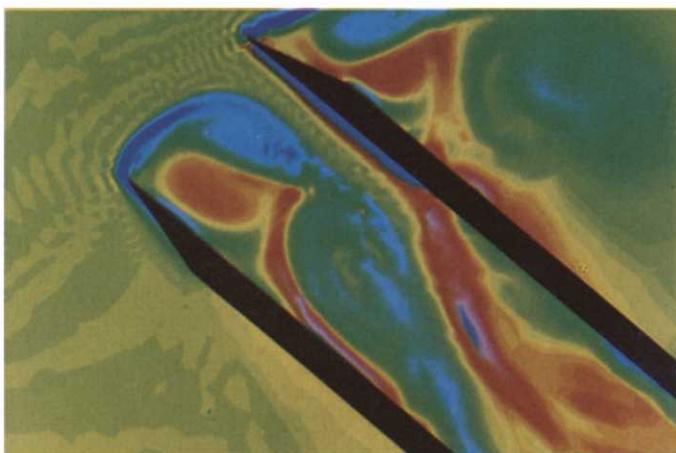


Fig. 8.25. Vorticity visualization, $\text{Re} = 750$; $t = 8$.

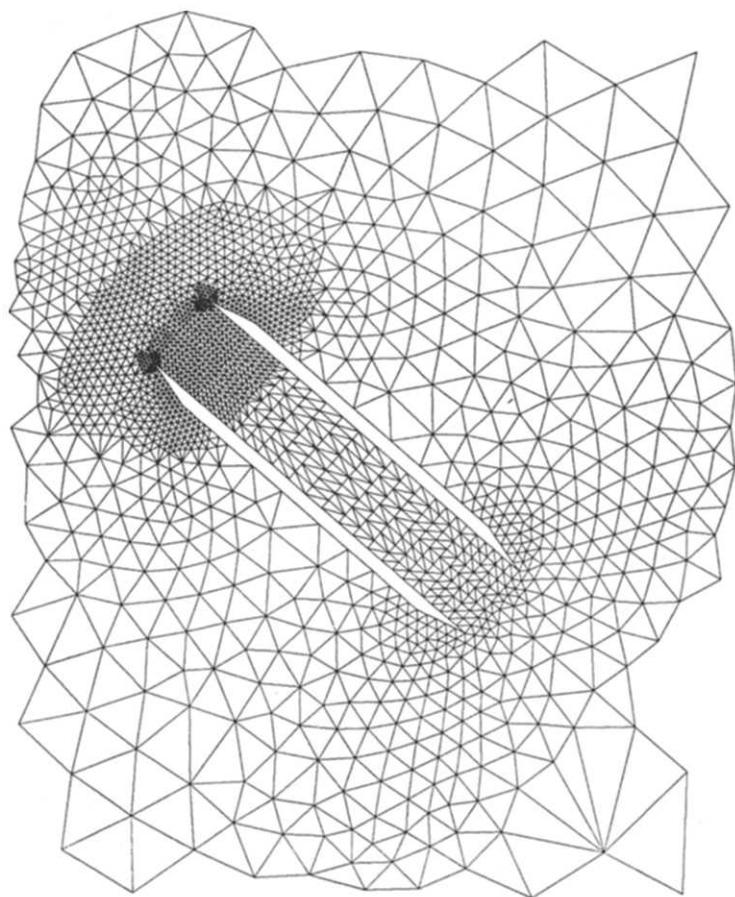


Fig. 8.26. Pressure grid.

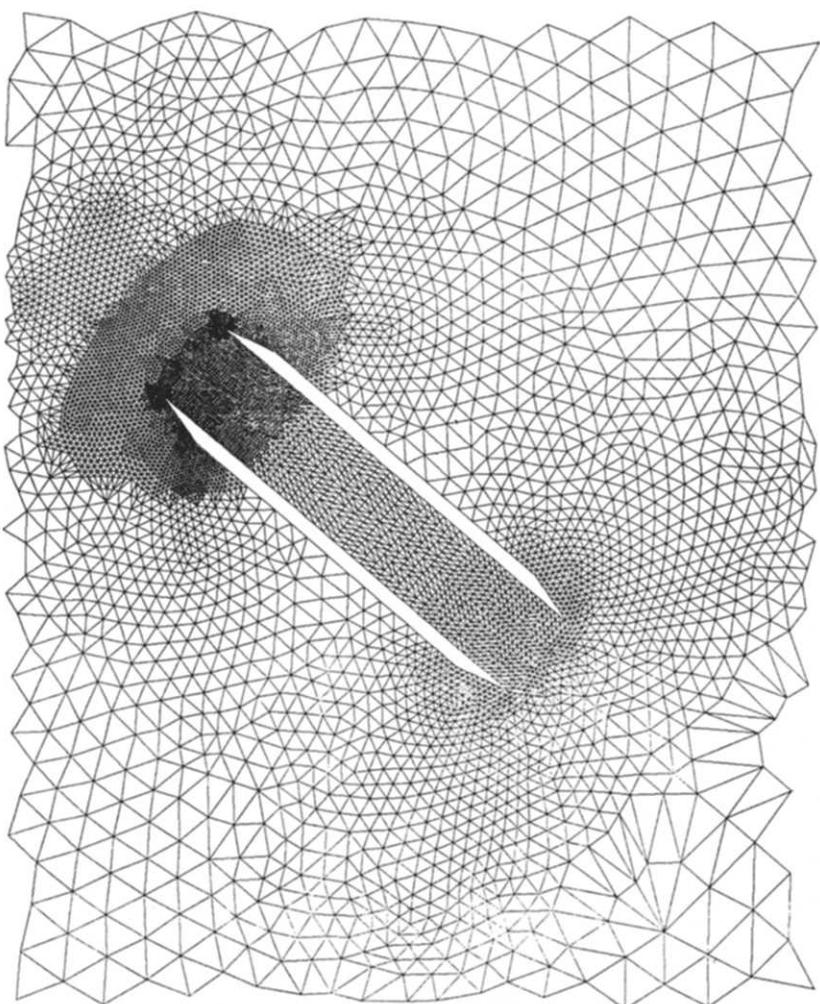


Fig. 8.27. Velocity grid.

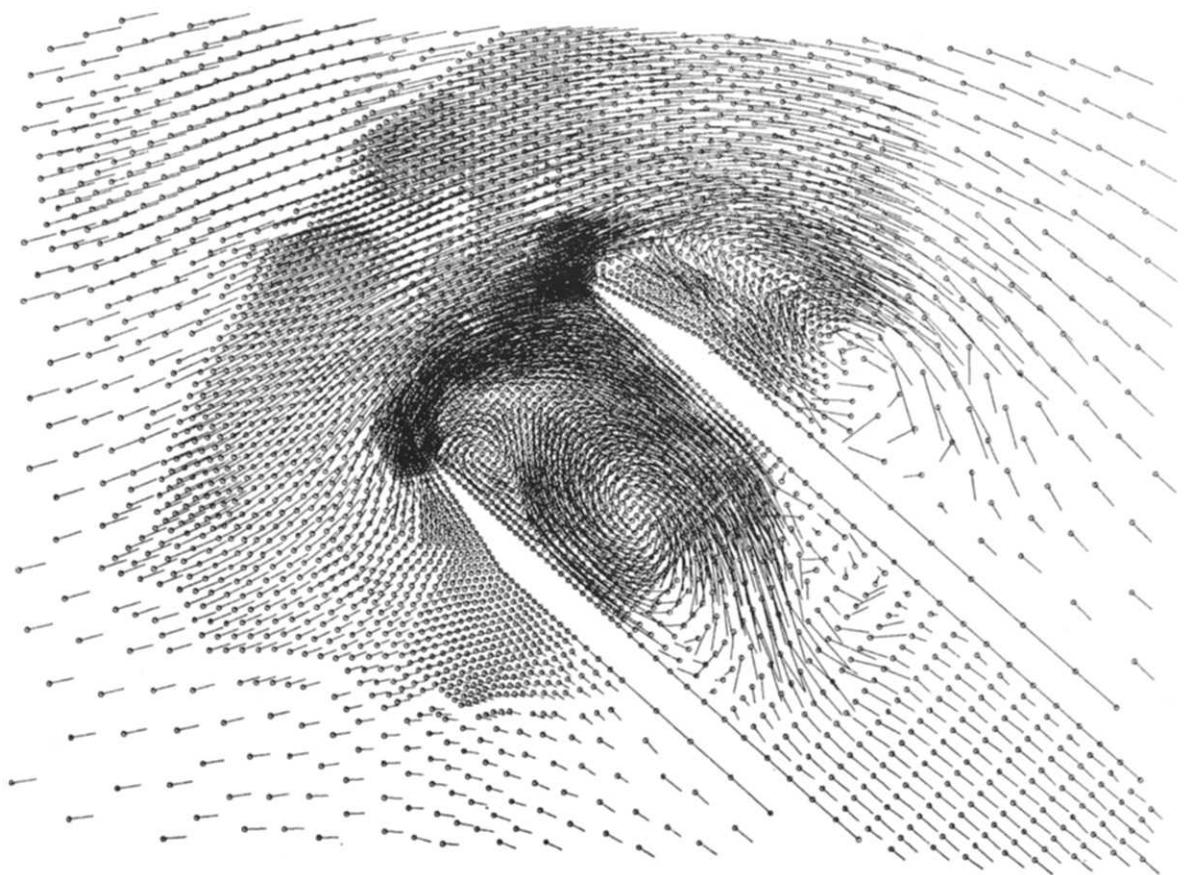


Fig. 8.28. Velocity visualization, $\text{Re} = 1500$; $t = 3.5$.

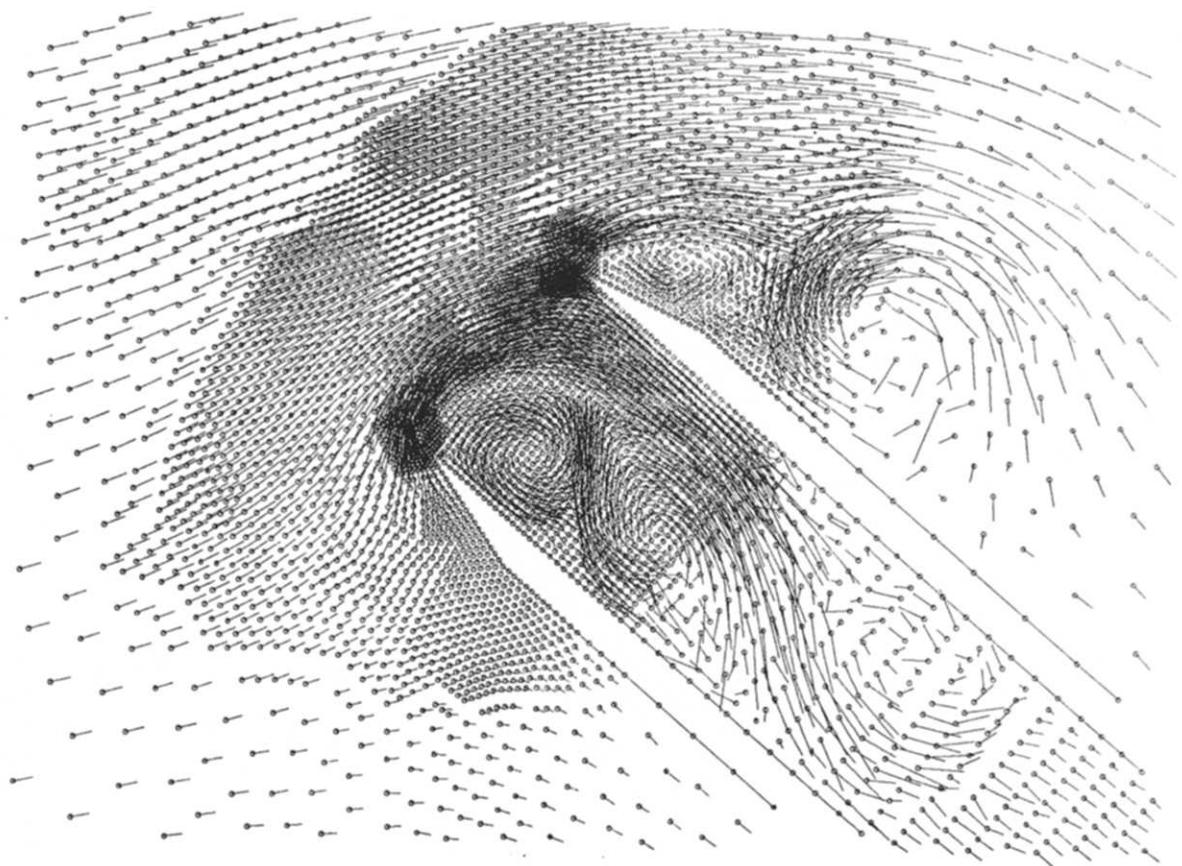


Fig. 8.29. Velocity visualization, $\text{Re} = 1500$; $t = 4.5$.



Fig. 8.30. Vorticity contours, $\text{Re} = 1500$; $t = 3.5$.



Fig. 8.31. Vorticity contours, $\text{Re} = 1500$; $t = 4.5$.



Fig. 8.34. Vorticity distribution for the non-refined grid.

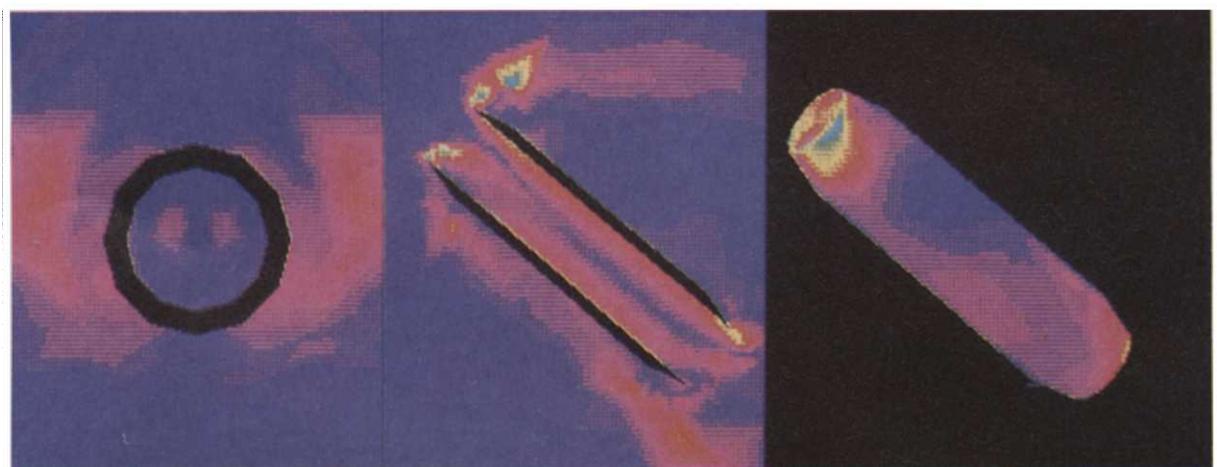


Fig. 8.35. Vorticity distribution for the refined grid.

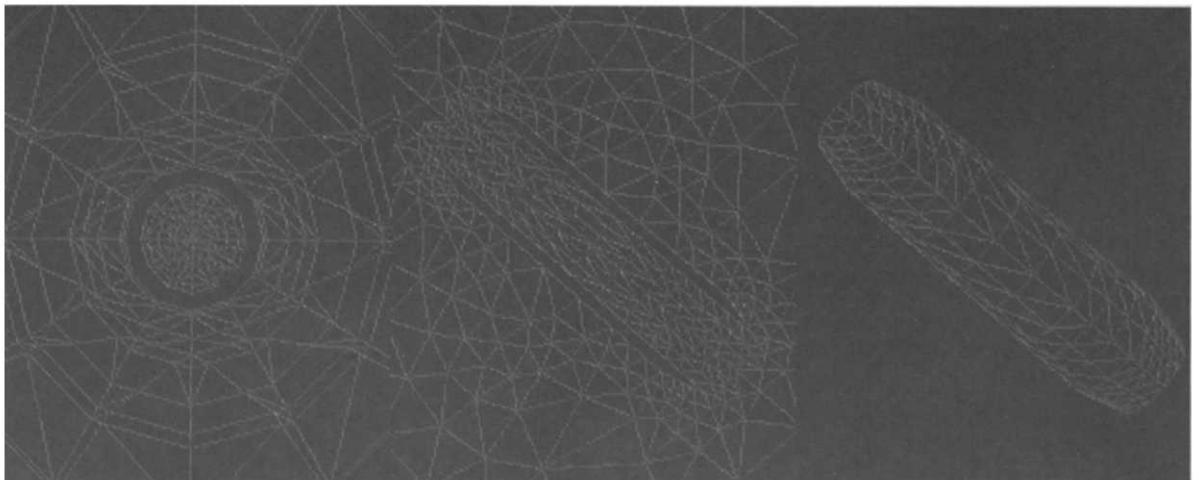


Fig. 8.32. Non-refined grid.

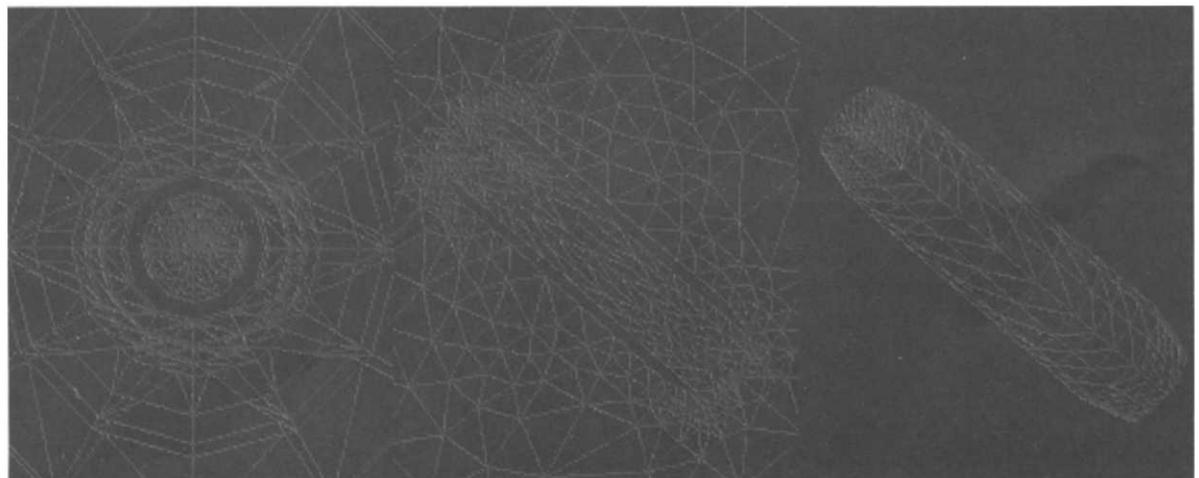


Fig. 8.33. Refined grid by self-adaptive techniques.

PART II: NUMERICAL SIMULATION OF COMPRESSIBLE VISCOUS FLOWS

9. Generalities. Synopsis

The numerical simulation of compressible viscous flows has been the subject of quite intensive efforts these last years. One of the main motivations of this fact is the computer (and more recently supercomputer) design of advanced aircrafts and space vehicles.

The main goal of this part II is to discuss finite element methods and associated algorithms, that we have used these last years to solve various compressible viscous flow problems, many of them being of industrial interest.

The numerical techniques to be discussed below are indeed natural extensions of those discussed in part I for incompressible viscous flows. However we have included in this part II a brief discussion of a self-adaptive grid refinement method that we found most useful in order to increase the local accuracy in those regions where the solution exhibits very strong gradients.

A large part of the material discussed in the following sections is contained in refs. [58,59].

10. Formulation of the Navier–Stokes equations for compressible viscous fluids

Let $\Omega \subset \mathbf{R}^N$ ($N = 2, 3$ in practice) be the flow domain and Γ be its boundary. The non-dimensional conservative form of the equations is given below by:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \mathbf{u} = 0, \quad (10.1a)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \frac{1}{Re} [\Delta \mathbf{u} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{u})], \quad (10.1b)$$

$$\frac{\partial e}{\partial t} + \nabla \cdot (e + p) \mathbf{u} = \frac{1}{Re} \left[\nabla \cdot [\mathbf{u} (-\frac{2}{3} \nabla \cdot \mathbf{u} \mathbf{I} + \nabla \mathbf{u} + \nabla \mathbf{u}^\dagger)] + \frac{\gamma}{Pr} \Delta \epsilon \right]; \quad (10.1c)$$

in (10.1), we have normalized by the subscript r

- (i) the density ρ by ρ_r ,
- (ii) the velocity \mathbf{u} by $|\mathbf{u}_r|$,
- (iii) the specific internal energy ϵ by $|\mathbf{u}_r|^2$,
- (iv) the pressure p by $\rho_r |\mathbf{u}_r|^2$.

The pressure obeys the ideal gas law:

$$p = (\gamma - 1) \rho \epsilon \quad (10.2)$$

Finally, we have also normalized the total energy e by $\rho_r |\mathbf{u}_r|^2$, and we have

$$e = \rho \epsilon + \rho |\mathbf{u}|^2 / 2. \quad (10.3)$$

The constants Re , Pr and γ are the Reynolds number, the Prandtl number and the ratio of specific heats, respectively ($\gamma = 1.4$ in air).

The above equations express the conservation of mass, momentum and energy. We normalize the temperature T by $|\mathbf{u}_r|^2/c_v$, implying that

$$T = \epsilon. \quad (10.4)$$

From (10.1)–(10.4), we can deduce the following non-conservative form of the Navier–Stokes equations:

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho + \rho \nabla \cdot \mathbf{u} = 0, \quad (10.5)$$

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho (\mathbf{u} \cdot \nabla) \mathbf{u} + (\gamma - 1) \nabla(\rho T) = \frac{1}{Re} [\Delta \mathbf{u} + \frac{1}{3} \nabla(\nabla \cdot \mathbf{u})], \quad (10.6)$$

$$\rho \frac{\partial T}{\partial t} + \rho \mathbf{u} \cdot \nabla T + (\gamma - 1) \rho T \nabla \cdot \mathbf{u} = \frac{1}{Re} \left[\frac{\gamma}{Pr} \Delta T + F(\nabla \mathbf{u}) \right], \quad (10.7)$$

where (10.2)–(10.4) still hold.

For three-dimensional flows, we have $\mathbf{u} = \{u, v, w\}$ and $F(\cdot)$ in (10.7) has the following expression:

$$\begin{aligned} F(\nabla \mathbf{u}) &= \frac{4}{3} \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2 \right] + \left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right)^2 \\ &\quad - \frac{4}{3} \left(\frac{\partial u}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial u}{\partial x} \frac{\partial w}{\partial z} + \frac{\partial v}{\partial y} \frac{\partial w}{\partial z} \right). \end{aligned} \quad (10.8)$$

We consider external flows around airfoils or inlets for instance; the domain of computation is described in fig. 10.1. Let Γ_∞ be the far-field boundary of the domain; we introduce then

$$\Gamma_\infty^- = \{x \mid x \in \Gamma_\infty, \mathbf{u}_\infty \cdot \mathbf{n} < 0\}, \quad (10.9)$$

$$\Gamma_\infty^+ = \Gamma_\infty \setminus \Gamma_\infty^-, \quad (10.10)$$

where \mathbf{u}_∞ denotes the free stream velocity and \mathbf{n} the unit vector of the outward normal to Γ .

We assume the flow to be uniform at infinity, and the corresponding variables to be normalized by the free stream values; then for example, we prescribe at infinity

$$\mathbf{u} = \mathbf{u}_\infty = \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix}, \quad \alpha \text{ angle of attack}, \quad (10.11)$$

$$\rho = 1, \quad (10.12)$$

$$T = T_\infty = 1/\gamma(\gamma - 1) M_\infty^2, \quad (10.13)$$

where M_∞ denotes the free stream Mach number.

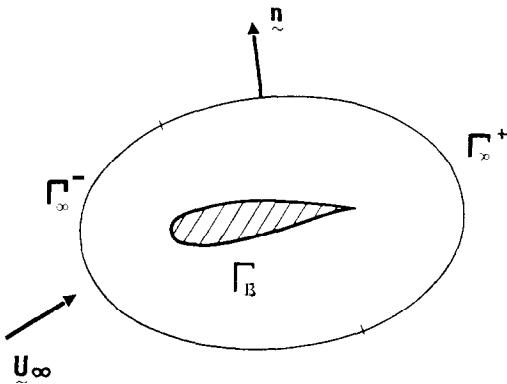


Fig. 10.1.

The problem of the boundary conditions on the computational boundary Γ_∞ is more delicate and will be discussed later.

On the rigid boundary Γ_B , we shall use the following conditions:

$$\mathbf{u} = \mathbf{0} \quad (\text{no-slip condition}), \quad (10.14)$$

$$T = T_B = T_\infty(1 + (\gamma - 1)/2 M_\infty^2) \quad (\text{free stream total temperature}). \quad (10.15)$$

The value of the pressure (and density) at a rigid boundary is deduced from the momentum equation. This operation is done implicitly through a generalized Stokes solver (to be discussed hereafter). An alternative is to combine (10.14) and (10.1); we obtain

$$\frac{\partial p}{\partial n} = \frac{1}{Re} [\Delta \mathbf{u} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{u})] \cdot \mathbf{n}, \quad (10.16)$$

which can be used through an explicit procedure.

Finally, since we consider time dependent equations (even if we are looking for steady solutions), initial conditions have to be added; we shall take

$$\rho(x, 0) = \rho_0(x), \quad (10.17)$$

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x), \quad (10.18)$$

$$T(x, 0) = T_0(x). \quad (10.19)$$

Solving the compressible Navier–Stokes equations is a most difficult task. Most of the existing numerical solution methods are based on Finite Differences techniques, for both space and time discretizations (see refs. [11,60–63] and the references therein); Spectral Methods have also been considered (see refs. [64,65]).

More recently, Finite Element Methods have been investigated to solve the above problem (see refs. [66,67]). As a final comment, we would like to mention ref. [68] where some theoretical results concerning the compressible Navier–Stokes equations have been proved, assuming that the data are “small”. The above reference list is far from being complete.

11. Application of operator splitting methods to the solution of the time-dependent compressible Navier–Stokes equations

11.1. Transformation of the equations

Using convenient operator splitting methods described in part I, section 2, for the time discretization of the compressible Navier–Stokes equations, we are able to decouple the difficulties due to the nonlinearity and to the pressure computation at rigid boundaries, respectively.

In order to extend the methodology previously used for solving the incompressible Navier–Stokes equations (see part I), we introduce the following new variable:

$$\sigma = \ln \rho. \quad (11.1)$$

With this new variable, the system (10.5)–(10.7) becomes:

$$\frac{\partial \sigma}{\partial t} + \nabla \cdot \mathbf{u} + \mathbf{u} \cdot \nabla \sigma = 0, \quad (11.2)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + (\gamma - 1)(T \nabla \sigma + \nabla T) = \frac{e^{-\sigma}}{Re} (\Delta \mathbf{u} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{u})), \quad (11.3)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T + (\gamma - 1)T \nabla \cdot \mathbf{u} = \frac{e^{-\sigma}}{Re} \left(\frac{\gamma}{Pr} \Delta T + F(\nabla \mathbf{u}) \right); \quad (11.4)$$

the eqs. (11.3), (11.4) can also be written as:

$$\frac{\partial \mathbf{u}}{\partial t} - \mu \Delta \mathbf{u} + \beta \nabla \sigma - \Psi(\sigma, \mathbf{u}, T) = \mathbf{0}, \quad (11.5)$$

$$\frac{\partial T}{\partial t} - \Pi \Delta T - \chi(\sigma, \mathbf{u}, T) = 0, \quad (11.6)$$

with δ : a mean value of the inverse of the density, $\delta = 1$ is a possible value,

$$\nu = 1/Re, \quad \mu = \nu \delta, \quad \Pi = \gamma \nu \delta / Pr,$$

$$\beta = (\gamma - 1)T_B = \frac{1}{\gamma} \left(\frac{\gamma - 1}{2} + \frac{1}{M_\infty^2} \right),$$

$$\Psi(\sigma, \mathbf{u}, T) = -(\gamma - 1)[\nabla T + (T - T_B)\nabla \sigma] - (\mathbf{u} \cdot \nabla) \mathbf{u} + \nu [e^{-\sigma} (\Delta \mathbf{u} + \frac{1}{3} \nabla (\nabla \cdot \mathbf{u})) - \delta \Delta \mathbf{u}],$$

$$\chi(\sigma, \mathbf{u}, T) = -(\gamma - 1)T \nabla \cdot \mathbf{u} - \mathbf{u} \cdot \nabla T + \frac{\gamma \nu}{Pr} (e^{-\sigma} \Delta T - \delta \Delta T) + \nu e^{-\sigma} F(\nabla \mathbf{u}).$$

11.2. An operator splitting method

Let $\Delta t (> 0)$ be a time discretization step and θ belongs to $]0, 1/2[$. This method is derived from scheme (2.6)–(2.9) and is obtained as follows:

$$\begin{cases} \sigma^0 = \sigma_0 = \ln \rho_0, \\ \mathbf{u}^0 = \mathbf{u}_0, \\ T^0 = T_0, \end{cases} \quad (11.7)$$

then for $n \geq 0$, starting from $\{\sigma^n, \mathbf{u}^n, T^n\}$, we solve

$$\begin{cases} \frac{\sigma^{n+\theta} - \sigma^n}{\theta \Delta t} + \nabla \cdot \mathbf{u}^{n+\theta} = -\mathbf{u}^n \cdot \nabla \sigma^n, \\ \frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta \Delta t} - a\mu \Delta \mathbf{u}^{n+\theta} + \beta \nabla \sigma^{n+\theta} = \Psi(\sigma^n, \mathbf{u}^n, T^n) + b\mu \Delta \mathbf{u}^n, \\ \frac{T^{n+\theta} - T^n}{\theta \Delta t} - a\Pi \Delta T^{n+\theta} = \chi(\sigma^n, \mathbf{u}^n, T^n) + b\Pi \Delta T^n, \end{cases} \quad (11.8)$$

$$\begin{cases} \frac{\sigma^{n+1-\theta} - \sigma^{n+\theta}}{(1-2\theta) \Delta t} + \mathbf{u}^{n+1-\theta} \cdot \nabla \sigma^{n+1-\theta} = -\nabla \cdot \mathbf{u}^{n+\theta}, \\ \frac{\mathbf{u}^{n+1-\theta} - \mathbf{u}^{n+\theta}}{(1-2\theta) \Delta t} - b\mu \Delta \mathbf{u}^{n+1-\theta} - \Psi(\sigma^{n+1-\theta}, \mathbf{u}^{n+1-\theta}, T^{n+1-\theta}) = a\mu \Delta \mathbf{u}^{n+\theta} - \beta \nabla \sigma^{n+\theta}, \\ \frac{T^{n+1-\theta} - T^{n+\theta}}{(1-2\theta) \Delta t} - b\Pi \Delta T^{n+1-\theta} - \chi(\sigma^{n+1-\theta}, \mathbf{u}^{n+1-\theta}, T^{n+1-\theta}) = a\Pi \Delta T^{n+\theta}, \end{cases} \quad (11.9)$$

$$\begin{cases} \frac{\sigma^{n+1} - \sigma^{n+1-\theta}}{\theta \Delta t} + \nabla \cdot \mathbf{u}^{n+1} = -\mathbf{u}^{n+1-\theta} \cdot \nabla \sigma^{n+1-\theta}, \\ \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1-\theta}}{\theta \Delta t} - a\mu \Delta \mathbf{u}^{n+1} + \beta \nabla \sigma^{n+1} = \Psi(\sigma^{n+1-\theta}, \mathbf{u}^{n+1-\theta}, T^{n+1-\theta}) + b\mu \Delta \mathbf{u}^{n+1-\theta}, \\ \frac{T^{n+1} - T^{n+1-\theta}}{\theta \Delta t} - a\mu \Delta T^{n+1} = \chi(\sigma^{n+1-\theta}, \mathbf{u}^{n+1-\theta}, T^{n+1-\theta}) + b\Pi \Delta T^{n+1-\theta}, \end{cases} \quad (11.10)$$

with $0 < a, b < 1$, $a + b = 1$, satisfying (2.28) (a, b play the role of α, β in the previous part). The boundary conditions will be specified in the following.

11.3. Some comments and remarks concerning scheme (11.8)–(11.10)

Using the above operator splitting method, we have been able to decouple nonlinearity and pressure (and density) computation at rigid boundaries. We shall describe, in the following sections, the specific treatment of the subproblems encountered at each step.

The linear systems (11.8), (11.10) are generalized Stokes problems which are also found in the shallow-water equations [69].

We use a direct Stokes solver (section 12) introduced by Glowinski–Pironneau [70], which reduces the problem to a cascade of Poisson problems.

For the nonlinear system (11.9), we introduce either a rational Runge–Kutta technique using dynamic relaxation [71] or a least squares formulation and a preconditioned conjugate gradient algorithm (section 13).

12. Solution of the generalized Stokes subproblems

12.1. Generalities. Synopsis

At each full step of the splitting method (11.8)–(11.10), we have to solve two linear problems of the following type

$$\alpha\sigma + \nabla \cdot \mathbf{u} = g, \quad (12.1)$$

$$\alpha\mathbf{u} - a\mu \Delta \mathbf{u} + \beta \nabla \sigma = \mathbf{f}, \quad (12.2)$$

$$\alpha T - a\mu \Delta T = h, \quad (12.3)$$

$$\begin{cases} \sigma = 0 & \text{on } \Gamma_{\infty}^-, \\ \partial\sigma/\partial n = 0 & \text{on } \Gamma_{\infty}^+, \end{cases} \quad (12.4)$$

$$\begin{cases} \mathbf{u} = \mathbf{0} & \text{on } \Gamma_B, \\ \mathbf{u} = \mathbf{u}_{\infty} & \text{on } \Gamma_{\infty}^-, \\ \partial\mathbf{u}/\partial n = \mathbf{0} & \text{on } \Gamma_{\infty}^+, \end{cases} \quad (12.5)$$

$$\begin{cases} T = T_B & \text{on } \Gamma_B, \\ T = T_{\infty} & \text{on } \Gamma_{\infty}^-, \\ \partial T/\partial n = 0 & \text{on } \Gamma_{\infty}^+, \end{cases} \quad (12.6)$$

where α is a positive parameter and where g , \mathbf{f} , h are given functions.

The temperature T is the solution of the decoupled Poisson equation (12.3), (12.6). We shall describe below a direct method for solving the generalized Stokes problem satisfied by $\{\sigma, \mathbf{u}\}$.

A direct method for solving Stokes problem has been introduced by Glowinski and Pironneau [70] and has been extensively used as a component of an operator splitting method for the

solution of the time dependent incompressible Navier–Stokes equations [1], [38] in place of the iterative methods described in section 4. This Stokes solver has been generalized by Benque et al. [69,72] in order to solve the shallow water equations; problem (12.1), (12.2), (12.4), (12.5) is of the same type and therefore can be solved by techniques very close to those used in refs. [69,72].

The proposed method reduces the solution of the problem (12.1), (12.2), (12.4), (12.5) to the solution of a finite number of Poisson problems plus the solution of a boundary integral equation.

Let us introduce the motivation of the algorithm; for more precisions and theoretical results, see refs. [1,70]. First we apply the operator $\nabla \cdot$ to eq. (12.2), we obtain:

$$\alpha \nabla \cdot \mathbf{u} - a\mu \Delta (\nabla \cdot \mathbf{u}) + \beta \Delta \sigma = \nabla \cdot \mathbf{f}; \quad (12.7)$$

on the other hand, eq. (12.1) yields

$$\nabla \cdot \mathbf{u} = g - \alpha \sigma. \quad (12.8)$$

Then, eq. (12.7) can be written with only σ as unknown variable:

$$\alpha^2 \sigma - (\beta + \alpha a\mu) \Delta \sigma = \alpha g - \nabla \cdot \mathbf{f} - a\mu \Delta g, \quad (12.9)$$

with the boundary conditions (12.4).

In order to have a well-posed problem in σ , it is necessary to have an additional boundary condition of type:

$$\sigma = \lambda \quad \text{on} \quad \Gamma_B.$$

If σ is evaluated, then the velocity \mathbf{u} is solution of

$$\alpha \mathbf{u} - a\mu \Delta \mathbf{u} = -\beta \nabla \sigma + \mathbf{f}, \quad (12.10)$$

with the boundary conditions (12.5).

The value of λ has to be calculated in order that eq. (12.1) is satisfied. Therefore we introduce the function ψ solution of:

$$\alpha^2 \psi - \zeta \Delta \psi = \alpha \sigma + \nabla \cdot \mathbf{u} - g, \quad (12.11)$$

$$\begin{cases} \psi = 0 & \text{on} \quad \Gamma_B \cup \Gamma_\infty^-, \\ \partial \psi / \partial n = 0 & \text{on} \quad \Gamma_\infty^+ \end{cases} \quad (12.12)$$

with $\zeta = \beta + \alpha a\mu$.

It can easily be proved using (12.1), (12.2) that the function ψ satisfies an equation of the type:

$$\mathcal{A}\psi - \mathcal{B}\Delta\psi + \mathcal{C}\Delta^2\psi = 0, \quad \mathcal{A}, \mathcal{B}, \mathcal{C} > 0.$$

If we can prove that $\psi \equiv 0$, then eq. (12.1) will be satisfied. The function ψ depends linearly of λ , and the following boundary conditions have to be satisfied

$$\begin{cases} \partial \psi / \partial n = 0 & \text{on} \quad \Gamma_B \cup \Gamma_\infty^-, \\ \psi = 0 & \text{on} \quad \Gamma_\infty^+; \end{cases} \quad (12.13)$$

therefore we introduce the symmetric linear operator

$$A: \lambda \rightarrow \left\{ \partial\psi/\partial n \mid_{\Gamma_B \cup \Gamma_\infty^-}, \psi \mid_{\Gamma_\infty^+} \right\}.$$

Then the solution method can be written as follows:

Construction of the implicit integral operator A

Let us define σ_λ , \mathbf{u}_λ , ψ_λ solutions of (12.14), (12.15), (12.16) respectively

$$\begin{cases} \alpha^2\sigma_\lambda - \xi \Delta\sigma_\lambda = 0, \\ \sigma_\lambda = \lambda \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^-, \\ \partial\sigma_\lambda/\partial n = \lambda \quad \text{on} \quad \Gamma_\infty^+, \end{cases} \quad (12.14)$$

$$\begin{cases} \alpha\mathbf{u}_\lambda - a\mu \Delta\mathbf{u}_\lambda = -\beta\nabla\sigma_\lambda, \\ \mathbf{u}_\lambda = \mathbf{0} \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^-, \\ \partial\mathbf{u}_\lambda/\partial n = \mathbf{0} \quad \text{on} \quad \Gamma_\infty^+, \end{cases} \quad (12.15)$$

$$\begin{cases} \alpha^2\psi_\lambda - \xi \Delta\psi_\lambda = \alpha\sigma_\lambda + \nabla \cdot \mathbf{u}_\lambda, \\ \psi_\lambda \quad \text{satisfying (12.12)}. \end{cases} \quad (12.16)$$

Then we define A by:

$$A\lambda = \left\{ -\partial\psi_\lambda/\partial n \mid_{\Gamma_B \cup \Gamma_\infty^-}, \psi_\lambda \mid_{\Gamma_\infty^+} \right\}. \quad (12.17)$$

Construction of the explicit part of the boundary integral equation

Let us define σ_0 , \mathbf{u}_0 , ψ_0 solutions of (12.18), (12.19), (12.20) respectively

$$\begin{cases} \alpha^2\sigma_0 - \xi \Delta\sigma_0 = \alpha g - \nabla \cdot \mathbf{f} - a\mu \Delta g, \\ \sigma_0 = 0 \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^-, \\ \partial\sigma_0/\partial n = 0 \quad \text{on} \quad \Gamma_\infty^+, \end{cases} \quad (12.18)$$

$$\begin{cases} \alpha\mathbf{u}_0 - a\mu \Delta\mathbf{u}_0 = \mathbf{f} - \beta\nabla\sigma_0, \\ \mathbf{u}_0 \quad \text{satisfying (12.5)}, \end{cases} \quad (12.19)$$

$$\begin{cases} \alpha^2\psi_0 - \xi \Delta\psi_0 = \alpha\sigma_0 + \nabla \cdot \mathbf{u}_0 - g, \\ \psi_0 \quad \text{satisfying (12.12)}, \end{cases} \quad (12.20)$$

and we obtain

$$b_0 = \left\{ \partial\psi_0/\partial n \mid_{\Gamma_B \cup \Gamma_\infty^-}, \psi_0 \mid_{\Gamma_\infty^+} \right\}. \quad (12.21)$$

Then the value of λ is given by the linear integral equation:

$$A\lambda = b_0, \quad (12.22)$$

with

$$\lambda = 0 \quad \text{on} \quad \Gamma_\infty,$$

in order to satisfy the boundary conditions (12.4).

It follows from (12.14)–(12.22) that the solution of the generalized Stokes problem is reduced to $2N + 3$ Poisson problems ($N + 2$ to obtain ψ_0 , $N + 1$ to obtain $\{\sigma, \mathbf{u}\}$ once λ is known via the boundary integral solution of (12.22)).

13. Least squares-conjugate gradient solution of the nonlinear subproblems (11.9)

13.1. Variational formulations

At each full step of the operator splitting method (11.8)–(11.10), we have to solve a nonlinear system of the following type

$$\begin{cases} \alpha\sigma + \mathbf{u} \cdot \nabla\sigma = g, \\ \alpha\mathbf{u} - b\mu \Delta\mathbf{u} - \psi(\sigma, \mathbf{u}, T) = \mathbf{f}, \\ \alpha T - b\Pi \Delta T - \chi(\sigma, \mathbf{u}, T) = h, \end{cases} \quad (13.1a)$$

$$\begin{cases} \sigma = \sigma_B & \text{on } \Gamma_B, \\ \sigma = 0 & \text{on } \Gamma_\infty^-, \end{cases} \quad (13.1b)$$

$$\begin{cases} \mathbf{u} = \mathbf{0} & \text{on } \Gamma_B, \\ \mathbf{u} = \mathbf{u}_\infty & \text{on } \Gamma_\infty^-, \\ \partial\mathbf{u}/\partial n = \mathbf{0} & \text{on } \Gamma_\infty^+, \end{cases} \quad (13.1c)$$

$$\begin{cases} T = T_B & \text{on } \Gamma_B, \\ T = T_\infty & \text{on } \Gamma_\infty^-, \\ \partial T/\partial n = 0 & \text{on } \Gamma_\infty^+, \end{cases} \quad (13.1d)$$

where α is a positive parameter and where g , \mathbf{f} , h are given functions. We denote by σ_B the rigid boundary value of σ calculated in the previous Stokes step.

Using the notations of fig. 10.1, we introduce the following functional spaces:

$$R_r = \{ \phi \in H^1(\Omega) \mid \phi = r \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \}, \quad (13.2)$$

$$W_z = \{ \mathbf{v} \in (H^1(\Omega))^n \mid \mathbf{v} = \mathbf{z} \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \}, \quad (13.3)$$

$$V_s = \{ \theta \mid \theta \in H^1(\Omega), \theta = s \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \}, \quad (13.4)$$

where $H^1(\Omega)$ is defined by (3.2). If r (resp. z , s) is sufficiently smooth, then R_r (resp. W_z , V_s) is non-empty (the above choice for the space of the densities R_r is motivated by the fact that ρ will be approximated by continuous functions and that the restriction of ρ to Γ_∞^- makes sense; of course, this supposes implicitly that ρ has some regularity).

Then an equivalent variational formulation of eqs. (13.1) is

$$\begin{cases} \alpha \int_{\Omega} \sigma \phi \, dx + \int_{\Omega} \mathbf{u} \cdot \nabla \sigma \phi \, dx = \int_{\Omega} g \phi \, dx, \\ \alpha \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, dx + b\mu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, dx - \int_{\Omega} \psi(\sigma, \mathbf{u}, T) \cdot \mathbf{v} \, dx = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \\ \alpha \int_{\Omega} T \theta \, dx + b\Pi \int_{\Omega} \nabla T \cdot \nabla \theta \, dx - \int_{\Omega} \chi(\sigma, \mathbf{u}, T) \theta \, dx = \int_{\Omega} h \theta \, dx, \\ \forall \{\phi, \mathbf{v}, \theta\} \in R_0 \times W_0 \times V_0, \quad \{\sigma, \mathbf{u}, T\} \in R_r \times W_z \times V_s, \end{cases} \quad (13.5)$$

where the value of r , z , s are precised by the boundary conditions of (13.1).

13.2. Least squares formulation

A natural least squares formulation of (13.5) is

$$\begin{cases} \text{Find } \{\sigma, \mathbf{u}, T\} \in R_z \times W_z \times V_s \text{ such that} \\ J(\sigma, \mathbf{u}, T) \leq J(\eta, \mathbf{w}, \tau), \forall \{\eta, \mathbf{w}, \tau\} \in R_r \times W_z \times V_s, \end{cases} \quad (13.6)$$

with

$$\begin{aligned} J(\eta, \mathbf{w}, \tau) &= \frac{\alpha}{2} \int_{\Omega} |\sigma - \eta|^2 \, dx + \frac{A}{2} \left\{ \alpha \int_{\Omega} |\mathbf{u} - \mathbf{w}|^2 \, dx + b\mu \int_{\Omega} |\nabla(\mathbf{u} - \mathbf{w})|^2 \, dx \right\} \\ &\quad + \frac{B}{2} \left\{ \alpha \int_{\Omega} |T - \tau|^2 \, dx + b\Pi \int_{\Omega} |\nabla(T - \tau)|^2 \, dx \right\}, \end{aligned} \quad (13.7)$$

where, in (13.7), we have $A > 0$, $B > 0$, and where σ , \mathbf{u} , T are functions of $\{\eta, \mathbf{w}, \tau\}$ through the state equations:

$$\begin{cases} \alpha \int_{\Omega} \sigma \phi \, dx = \int_{\Omega} g \phi \, dx - \int_{\Omega} \mathbf{w} \cdot \nabla \eta \phi \, dx, \quad \forall \phi \in R_0, \\ \alpha \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \, dx + b\mu \int_{\Omega} \nabla \mathbf{u} \cdot \nabla \mathbf{v} \, dx = \int_{\Omega} \psi(\eta, \mathbf{w}, \tau) \cdot \mathbf{v} \, dx + \int_{\Omega} \mathbf{f} \cdot \mathbf{v} \, dx, \quad \forall \mathbf{v} \in W_0, \\ \alpha \int_{\Omega} T \theta \, dx + b\Pi \int_{\Omega} \nabla T \cdot \nabla \theta \, dx = \int_{\Omega} \chi(\eta, \mathbf{w}, \tau) \theta \, dx + \int_{\Omega} h \theta \, dx, \quad \forall \theta \in V_0. \end{cases} \quad (13.8)$$

In order to solve the above minimization problem (13.6)–(13.8), we use a conjugate gradient algorithm analogous to the one described in section 3.4 for the incompressible case.

13.3. Conjugate gradient solution of the least squares problem (13.6)

The space $\mathbf{R}_r \times \mathbf{W}_z \times \mathbf{V}_s$ is equipped with the product norm below:

$$\begin{aligned} \{\phi, \mathbf{v}, \theta\} \rightarrow & \left\{ \alpha \int_{\Omega} \phi^2 dx + \alpha A \int_{\Omega} \mathbf{v}^2 dx + b\mu A \int_{\Omega} |\nabla \mathbf{v}|^2 dx \right. \\ & \left. + \alpha B \int_{\Omega} \theta^2 dx + b\Pi B \int_{\Omega} |\nabla \theta|^2 dx \right\}^{1/2}; \end{aligned}$$

then, the algorithm can be written as follows:

Step 0: Initialization

$$\{\eta^0, \mathbf{w}^0, \tau^0\} \in \mathbf{R}_r \times \mathbf{W}_z \times \mathbf{V}_s \quad \text{given}; \quad (13.9)$$

solve then the following problems:

$$\alpha \int_{\Omega} g_{\eta}^0 \phi dx = \left\langle \frac{\partial J}{\partial \eta}(\eta^0, \mathbf{w}^0, \tau^0), \phi \right\rangle \quad \forall \phi \in \mathbf{R}_0; \quad g_{\eta}^0 \in \mathbf{R}_0, \quad (13.10)$$

$$\alpha \int_{\Omega} \mathbf{g}_w^0 \cdot \mathbf{v} dx + b\mu \int_{\Omega} \nabla \mathbf{g}_w^0 \cdot \nabla \mathbf{v} dx = \left\langle \frac{\partial J}{\partial \mathbf{w}}(\eta^0, \mathbf{w}^0, \tau^0), \mathbf{v} \right\rangle \quad \forall \mathbf{v} \in \mathbf{W}_0; \quad \mathbf{g}_w^0 \in \mathbf{W}_0, \quad (13.11)$$

$$\alpha \int_{\Omega} g_{\tau}^0 \theta dx + b\Pi \int_{\Omega} \nabla g_{\tau}^0 \nabla \theta dx = \left\langle \frac{\partial J}{\partial \tau}(\eta^0, \mathbf{w}^0, \tau^0), \theta \right\rangle \quad \forall \theta \in \mathbf{V}_0; \quad g_{\tau}^0 \in \mathbf{V}_0, \quad (13.12)$$

where $\langle \cdot, \cdot \rangle$ denotes duality pairings, and set

$$z_{\eta}^0 = g_{\eta}^0, \quad z_w^0 = \mathbf{g}_w^0, \quad z_{\tau}^0 = g_{\tau}^0. \quad (13.13)$$

Then for $n \geq 0$, assuming $\{\eta^n, \mathbf{w}^n, \tau^n\}$, $\mathbf{g}^n = \{g_{\eta}^n, \mathbf{g}_w^n, g_{\tau}^n\}$, $\mathbf{z}^n = \{z_{\eta}^n, z_w^n, z_{\tau}^n\}$ known, we obtain $\{\eta^{n+1}, \mathbf{w}^{n+1}, \tau^{n+1}\}$, $\mathbf{g}^{n+1} = \{g_{\eta}^{n+1}, \mathbf{g}_w^{n+1}, g_{\tau}^{n+1}\}$, $\mathbf{z}^{n+1} = \{z_{\eta}^{n+1}, z_w^{n+1}, z_{\tau}^{n+1}\}$ as follows:

Step 1: Descent

$$\begin{cases} \text{Find } \lambda_n \in \mathbf{R} \text{ such that} \\ J(\{\eta^n, \mathbf{w}^n, \tau^n\} - \lambda_n \mathbf{z}^n) \leq J(\{\eta^n, \mathbf{w}^n, \tau^n\}) \quad \forall \lambda \in \mathbf{R}, \end{cases} \quad (13.14)$$

and set

$$\{\eta^{n+1}, \mathbf{w}^{n+1}, \tau^{n+1}\} = \{\eta^n, \mathbf{w}^n, \tau^n\} - \lambda_n \mathbf{z}^n. \quad (13.15)$$

Step 2: Calculation of the new descent direction

Solve

$$\alpha \int_{\Omega} g_{\eta}^{n+1} \phi \, dx = \left\langle \frac{\partial J}{\partial \eta}(\eta^{n+1}, w^{n+1}, \tau^{n+1}), \phi \right\rangle \quad \forall \phi \in R_0; g_{\eta}^{n+1} \in R_0, \quad (13.16)$$

$$\begin{aligned} \alpha \int_{\Omega} \mathbf{g}_w^{n+1} \cdot \mathbf{v} \, dx + b\mu \int_{\Omega} \nabla \mathbf{g}_w^{n+1} \cdot \nabla \mathbf{v} \, dx &= \left\langle \frac{\partial J}{\partial w}(\eta^{n+1}, w^{n+1}, \tau^{n+1}), \mathbf{v} \right\rangle \\ \forall \mathbf{v} \in W_0; \mathbf{g}_w^{n+1} \in W_0, \end{aligned} \quad (13.17)$$

$$\begin{aligned} \alpha \int_{\Omega} g_{\tau}^{n+1} \theta \, dx + b\Pi \int_{\Omega} \nabla g_{\tau}^{n+1} \cdot \nabla \theta \, dx &= \left\langle \frac{\partial J}{\partial \tau}(\eta^{n+1}, w^{n+1}, \tau^{n+1}), \theta \right\rangle \quad \forall \theta \in V_0, g_{\tau}^0 \in V_0. \\ (13.18) \end{aligned}$$

Compute then

$$\gamma_n = \frac{\left(\{g_{\eta}^{n+1}, \mathbf{g}_w^{n+1}, g_{\tau}^{n+1}\}, \{g_{\eta}^{n+1} - g_{\eta}^n, \mathbf{g}_w^{n+1} - \mathbf{g}_w^n, g_{\tau}^{n+1} - g_{\tau}^n\} \right)_1}{\left(\{g_{\eta}^n, \mathbf{g}_w^n, g_{\tau}^n\}, \{g_{\eta}^n, \mathbf{g}_w^n, g_{\tau}^n\} \right)_1}, \quad (13.19)$$

where $\forall \{\phi, \mathbf{v}, \theta\}, \{\phi', \mathbf{v}', \theta'\}$, we have

$$\begin{aligned} (\{\phi, \mathbf{v}, \theta\}, \{\phi', \mathbf{v}', \theta'\})_1 &= \alpha \int_{\Omega} \phi \phi' \, dx + \alpha A \int_{\Omega} \mathbf{v} \cdot \mathbf{v}' \, dx + b\mu A \int_{\Omega} \nabla \mathbf{v} \cdot \nabla \mathbf{v}' \, dx \\ &\quad + \alpha B \int_{\Omega} \theta \theta' \, dx + b\Pi B \int_{\Omega} \nabla \theta \cdot \nabla \theta' \, dx. \end{aligned}$$

Finally define \mathbf{z}^{n+1} by

$$\mathbf{z}^{n+1} = \mathbf{g}^{n+1} + \gamma_n \mathbf{z}^n, \quad (13.20)$$

do $n = n + 1$ and go to (13.14).

Applying algorithm (13.9)–(13.20) to solve the least squares problem (13.6) requires the solution at each iteration of several decoupled Poisson problems associated to an elliptic operator of the type $(\alpha I - \mu \Delta)$.

We have also tested some of these methods combining the features of conjugate gradient with updated variable preconditioning and quasi-Newton algorithm (see refs. [31,32]) which have proved performing (20% faster) and robust for the solution of (13.6).

14. Finite element approximation

For simplicity, we suppose that Ω is a bounded polygonal domain of R^2 but the method is easily extended to domain of R^3 . The space discretization is achieved by a finite element

approximation. With T_h a standard finite element triangulation of Ω , and h the maximal length of the edges of the triangles of T_h , we introduce the following discrete spaces (with P_k = space of polynomials in two variables of degree $\leq k$):

$$R_{rh} = \left\{ \phi_h \mid \phi_h \in C^0(\bar{\Omega}), \quad \phi_{h|T} \in P_1, \forall T \in T_h, \quad \phi_h = r_h \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \right\}, \quad (14.1)$$

$$W_{zh} = \left\{ w_h \mid w_h \in (C^0(\bar{\Omega}))^2, \quad w_{h|T} \in P_1 \times P_1, \forall T \in T_h, \quad w_h = z_h \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \right\}, \quad (14.2)$$

$$V_{sh} = \left\{ \theta_h \mid \theta_h \in C^0(\bar{\Omega}), \quad \theta_{h|T} \in P_1, \forall T \in T_h, \quad \theta_h = s_h \quad \text{on} \quad \Gamma_B \cup \Gamma_\infty^- \right\}. \quad (14.3)$$

Then, we can write for the discrete problem, the algorithm previously defined for the continuous one; concerning the construction of the approximate integral operator A_h , we refer to Glowinski [1], Péraux [73].

We observe that unlike in the approximation of the incompressible Navier–Stokes equations, we use discrete spaces of the same type to approximate σ , \mathbf{u} and T .

15. Self-adaptive mesh refinement techniques for accurate solution of the compressible Navier–Stokes equations

The study presented in this section is based on the recent works of Palmerio [74] and Pouletty [75].

The geometric flexibility of triangular elements is well suited for refinement processes. The basic idea is, starting from a solution on an initial mesh, to improve locally the accuracy of this solution by refining the mesh according to criteria which are functions of the known solution. Then the mesh is depending not only on the geometry or on an a priori knowledge of the critical areas of the computational domain, but is also influenced a posteriori by the solution itself.

Presently, we are considering adaptive mesh refinement only for steady state problems, the generalization to transient problems is a most interesting question, but it still requires further investigations.

The refinement techniques are based on two procedures:

- i) *moving nodes*, their number remaining constant [76,77],
- ii) *adding nodes*, the location of the old ones being fixed [74,78].

Concerning the criterion, it can be of two types, either numerical based on a posteriori error estimates [79,80] or physical [78,81] (areas of large gradients, for instance).

The technique presented here is based principally on (ii) and physical criteria.

15.1. Global algorithm

We proceed via the following algorithm:

- (i) Let T_h^0 be the initial triangulation (or tetrahedrization in 3D) and \mathbf{u}^0 a quasi-converged solution of the considered problem.

Then, for $k \geq 0$, knowing the solution \mathbf{u}^k on the triangulation T_h^k , we define T_h^{k+1} , \mathbf{u}^{k+1} by:

(ii) Construction of E^k , set of the elements to be divided.

Let $C^k(T)$ be the value of the criterion, $T \in T_h^k$,

if

$$C^k(T) > \epsilon \max_{T \in T_h} C^k(T)$$

then

$$T \in E^k$$

with

$$0 < \epsilon < 1, \quad \text{to be specified,}$$

(iii) construction of the new triangulation T_h^{k+1} ,

(iv) computation of \mathbf{u}^{k+1} on T_h^{k+1} ;

the initialization of \mathbf{u}^{k+1} is given by linear interpolation of \mathbf{u}^k on T_h^{k+1} .

Let us now describe in details the important steps (ii) in section 15.3 and (iii) in section 15.2, of the above algorithm.

15.2. Local refinement technique, construction of T_h^{k+1} knowing E^k

For the areas containing the elements of E^k , the aim is to replace the elements of size h by elements of size $h/2$, ensuring moreover a good quality of the triangulation.

We give thereafter details for the 2D case (the 3D refinement is based on the same methodology, but introduces some extra geometrical difficulties (cf. remark (15.2))):

(i) Each triangle T of E^k is divided into four sub-triangles by adding a node at each mid-side; then a triangle T of the previous triangulation T_h^k has j nodes added ($j = 0, \dots, 3$) (cf. fig. 15.1).

(ii) If $j = 1, \dots, 3$, the triangles have to be sub-divided in order to obtain a conforming finite element triangulation, and, in the matching area ($j = 1, 2$), special care is required in order to ensure a “good quality” of the resulting triangulation. Therefore, we apply the following rules:

– $j = 0$, triangle T unchanged,

– $j = 1$, one node added, triangle T divided into two sub-triangles,

– $j = 2$, if T is divided into 3 sub-triangles, their shape is considered as unacceptable, then a third node is added and T is divided into four elements.

– $j = 3$, T is divided into 4 sub-triangles,

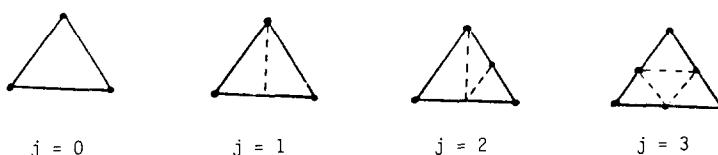


Fig. 15.1.

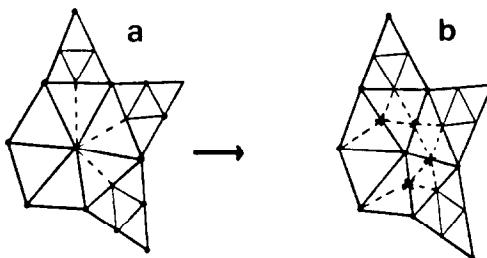


Fig. 15.2. Rejected splitting (a); allowed splitting: added nodes (b).

Iterative use of the above procedure generates a new mesh.

In order to maintain a good level of quality of the final triangulation, the following remarks are taken into account:

Remark 15.1. If a triangle with 2 vertices on the body has one node added, then two extra nodes are added and the triangle is divided into four sub-triangles.

Remark 15.2. If a node is added on a side adjacent to the body, this point is projected on the body (either by a parabolic interpolation or, if possible, by using the analytical function defining the body). For 3D industrial application, this technique requires rather important geometrical tools (interpolation, splines) in order to guarantee that new nodes are smoothly projected on the body.

Remark 15.3. In the areas where triangles have been sub-divided, a local elasticity-like equation [75] is applied in order to move locally the nodes and to obtain a smoother mesh. This technique is particularly efficient in the matching region where triangles have been sub-divided into two sub-triangles.

Among other moving nodes techniques to improve the smoothing of the mesh, we want to mention the recent works of Cua [82] using generalized gradient methods.

Remark 15.4. The quality of the triangulation is also related to the number of neighbours surrounding a node which has to be upper limited. This restriction applies to the triangles divided into two and to the node from which the splitting lines are issued. If more than two of them are issued from the same node, this situation is automatically rejected and in this case, the peculiar triangles are subdivided into four (see fig. 15.2).

In practice, the matching region connecting refined elements of E_k to non-refined elements of T_h^k is restricted to a limited size strip.

15.3. Choice of criteria for the compressible Navier–Stokes equations

For a viscous flow simulation, it is important to refine the mesh in the vicinity of the shocks and in the boundary layer and wake regions. For this reason, we introduce different criteria which can be combined:

$$C_1(T) = |\mathbf{u} \cdot \nabla M|_T, \quad (15.1)$$

$$C_2(T) = |\mathbf{u} \cdot \nabla M|_T / \|\mathbf{u}\|_T, \quad (15.2)$$

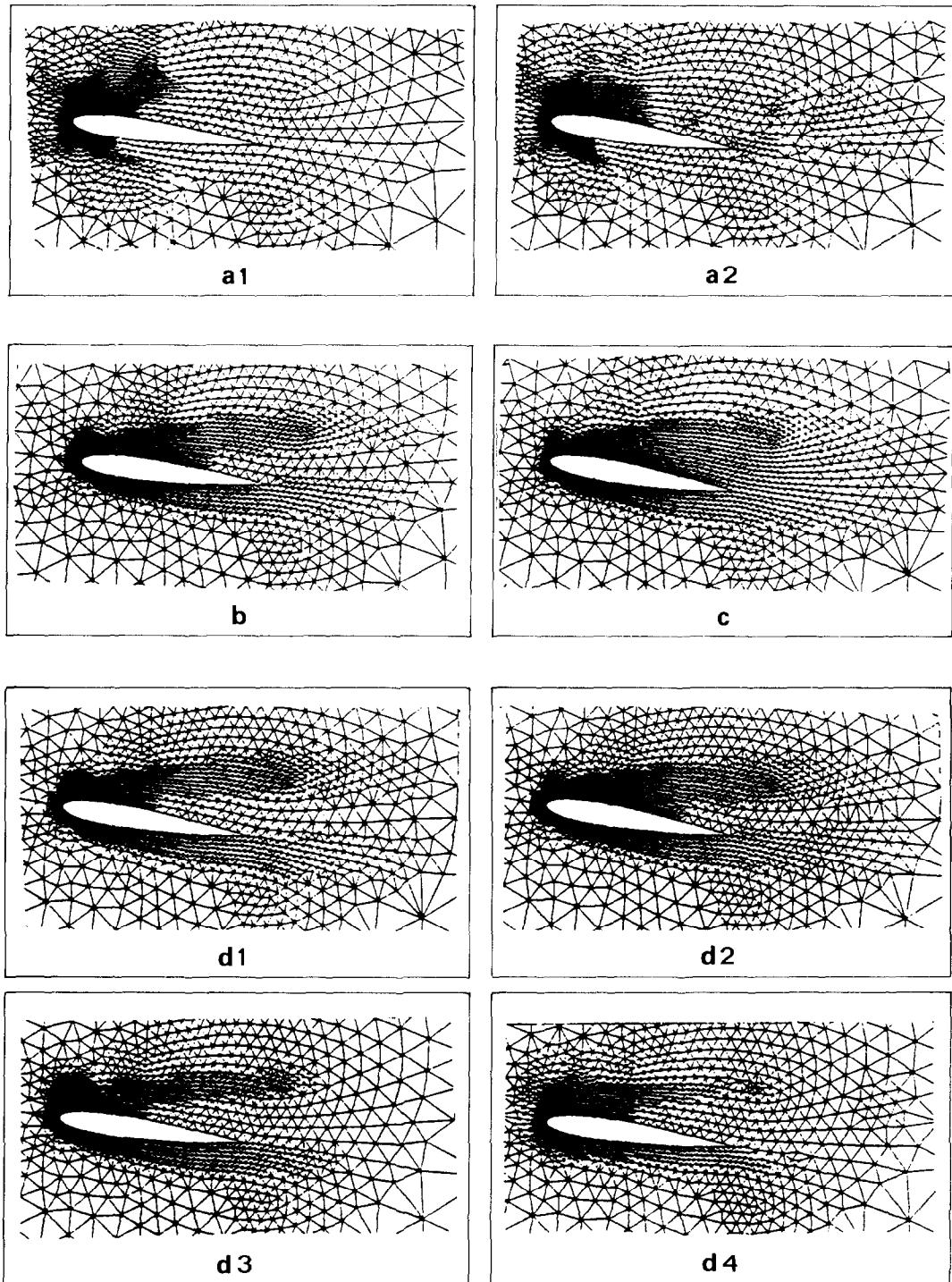


Fig. 15.3. Adapted meshes obtained with different criteria for the simulation of the flow around a NACA 0012, $M_\infty = 0.8$; $\text{Re} = 500$; $\alpha = 10^\circ$. (a1) $C_1(T)$; 2689 nodes, 5221 elements; (a2) $C_2(T)$; 2624 nodes, 5085 elements; (b) $C_3(T)$; 2652 nodes, 5122 elements; (c) $C_4(T)$; 2687 nodes, 5186 elements; (d1) $C_6(T)$; 2664 nodes, 5150 elements; (d2) $C_7(T)$; 2657 nodes, 5131 elements; (d3) $C_1 + C_6$; 2630 nodes, 5083 elements; (d4) $C_2 + C_7$; 2623 nodes, 5069 elements.

$$C_3(T) = \|\nabla M\|_T, \quad (15.3)$$

$$C_4(T) = \|\nabla \times u\|_T, \quad (15.4)$$

$$C_5(T) = \|\nabla \times u\|_T / \|u\|_T, \quad (15.5)$$

$$C_6(T) = \|u \times \nabla M\|_T, \quad (15.6)$$

$$C_7(T) = \|u \times \nabla M\|_T / \|u\|_T, \quad (15.7)$$

$$C_8(T) = C_7(T) \times L_T, \quad \forall T \in T_h, \quad (15.8)$$

using averaging formulae if necessary, these different criteria are defined as piecewise constant on each triangle (or tetrahedron) T of T_h .

In (15.1)–(15.8):

- (a) M is the local Mach number, $M = \|u\|/c$, with $c = \sqrt{\gamma p/\rho}$,
- (b) L_T is the distance between the triangle under consideration and the leading edge.

Several combinations of these criteria acting in preferred directions have been tested to generate adapted refined grids for the computation of a viscous flow around a NACA0012 airfoil as depicted in fig. 15.3.

Self-adaptive mesh refinement techniques provide for a given number of nodes, more accurate results (cf. ref. [81]). In order to improve efficiency of the algorithm, the step under investigation is to develop a full multigrid method on the different adapted meshes [83,86].

16. Numerical experiments

We illustrate the methods described in the above section by the presentation of the results of numerical experiments, where these methods have been applied to simulate some compressible viscous 2D and 3D flows of practical interest. All the calculations which follow have been performed either on IBM 3090 of AMD-BA industries or on CRAY 1S of CCVR using the finite element method associated to the discrete spaces defined by (14.1)–(14.3).

16.1. A first class of test problems

We consider the solution of the Navier–Stokes equations for the flow of compressible viscous fluids around a NACA0012 airfoil (fig. 16.1). We have selected this problem since it is a quite classical and significant test problem for Navier–Stokes solvers [87]. The method described in the above sections 9–14 has been applied to compute the steady solution of (10.5)–(10.7) for the two following tests.

The first test is a 2D transonic calculation at $M_\infty = 0.85$, $Re = 500$, at an angle of attack of $\alpha = 0^\circ$. Fig. 16.1 shows an enlargement of the triangulation near the profile and fig. 16.2 shows the computed pressure coefficient. Results of fig. 16.3, Mach distribution and density distribution have been obtained with scheme (11.8)–(11.10) after 50 time cycles (time step $\Delta t = 0.2$).

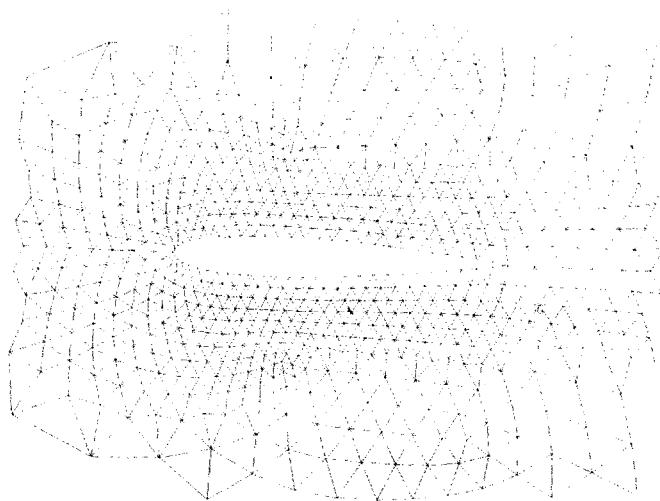


Fig. 16.1. An enlargement of the mesh near the airfoil, nodes = 2354, elements = 4586.

The second test is a 2D supersonic flow around a NACA0012 at $M_\infty = 2$, $Re = 106$, at an angle of attack of 10° . Fig. 16.4 shows an enlargement of the locally refined triangulation generated by self-adaptive algorithms (see section 16.3.1). In fig. 16.5 is represented the pressure distribution on the airfoil obtained with scheme (11.8)–(11.10). The colored pictures of fig. 16.6 (see page 159) show the pressure, Mach and density distributions around the airfoil. These results are in good agreement with those obtained by other Navier–Stokes solvers [87].

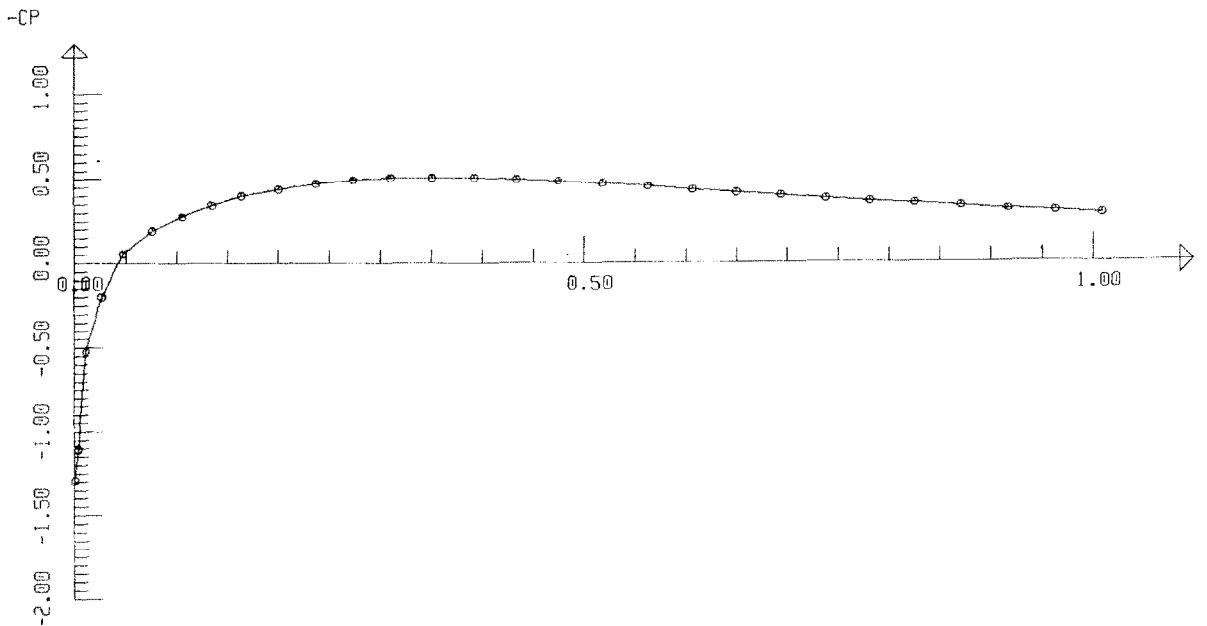


Fig. 16.2. Pressure coefficient on the NACA 0012 airfoil, $M_\infty = 0.85$; $Re = 500$.

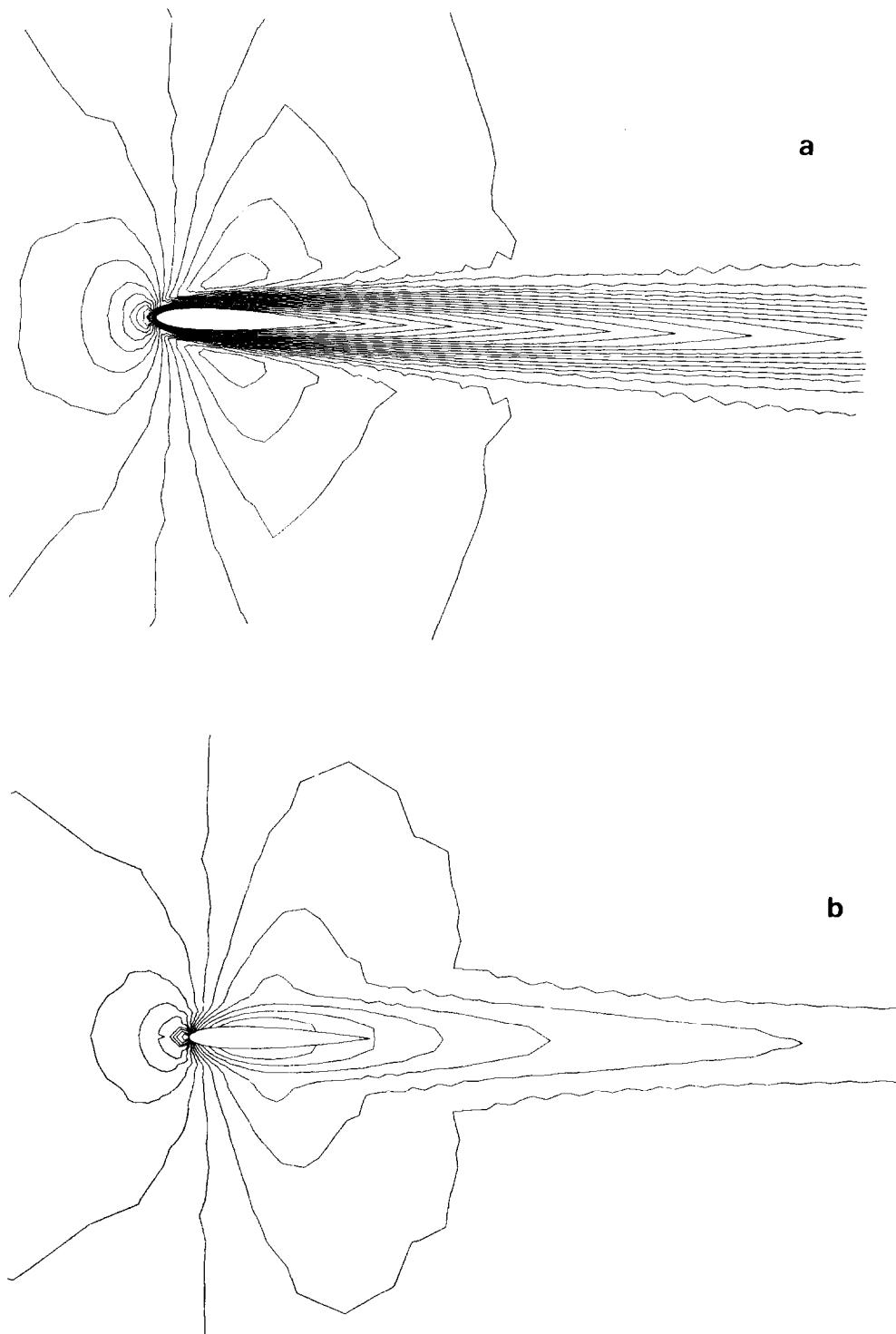


Fig. 16.3. Flow around a NACA 0012 airfoil, $M_\infty = 0.85$, $Re = 500$. (a) Mach contours; (b) density contours.

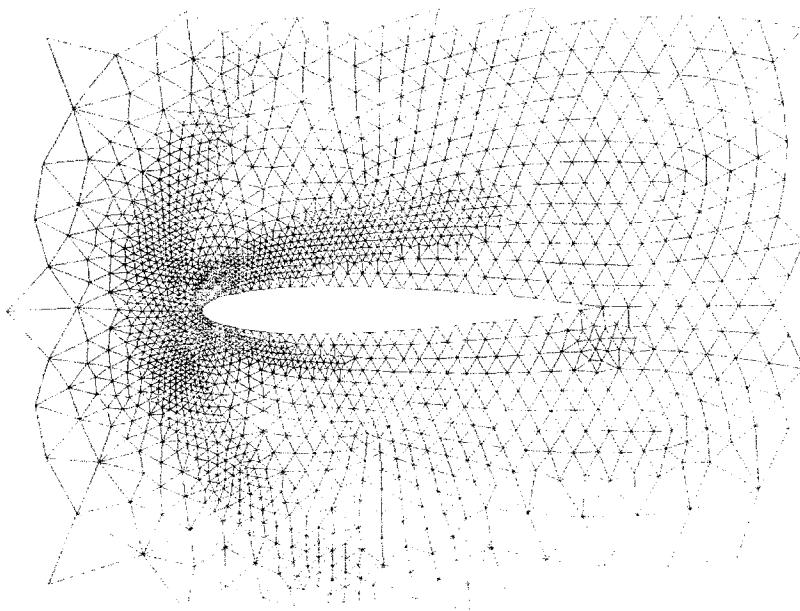


Fig. 16.4. Adapted mesh, $M_\infty = 2$; $\text{Re} = 106$; $\alpha = 10^\circ$.

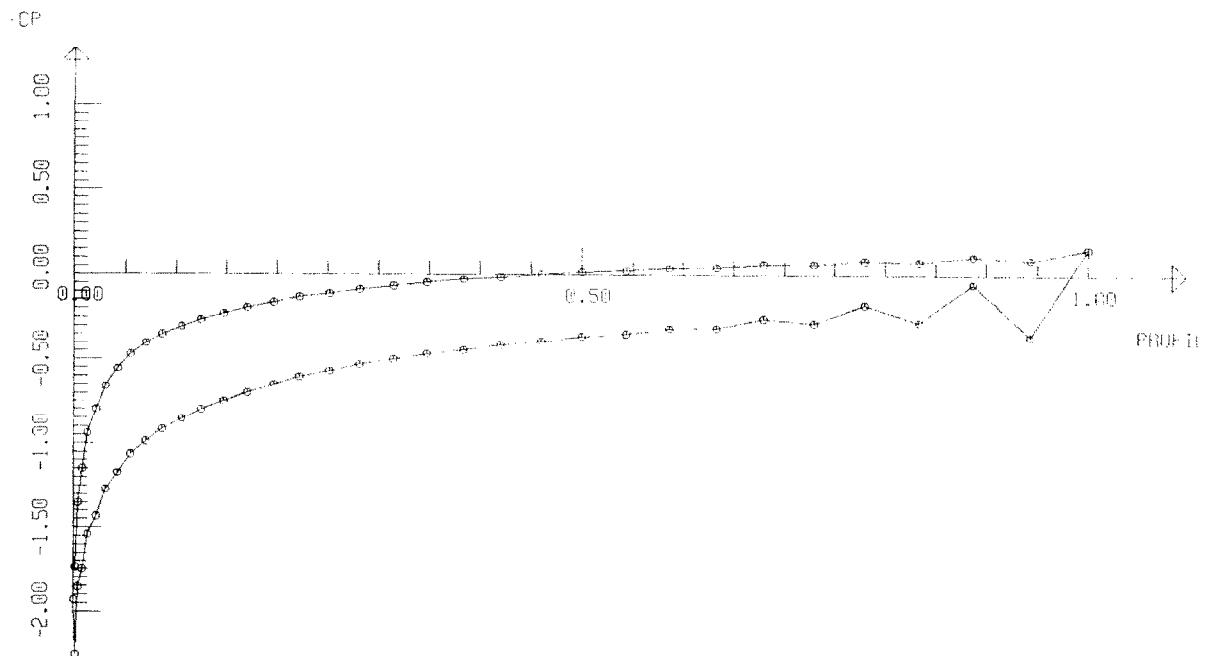


Fig. 16.5. Pressure coefficient on the NACA 0012 airfoil, $M_\infty = 2$; $\text{Re} = 106$; $\alpha = 10^\circ$.

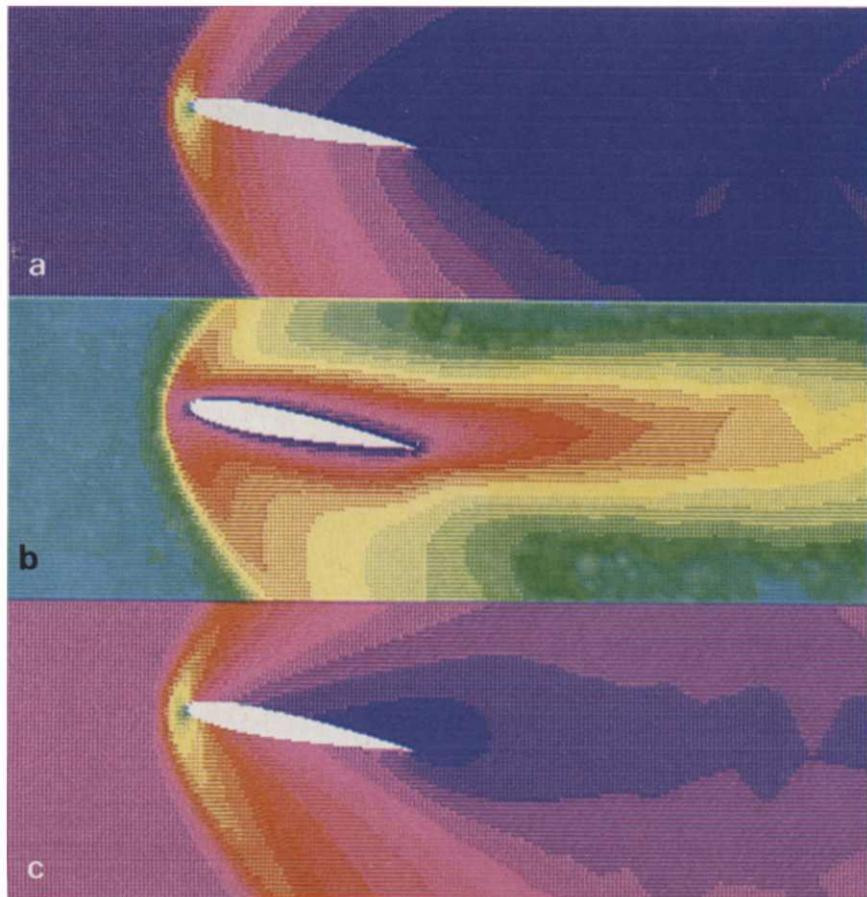


Fig. 16.6. Flow around a NACA 0012 airfoil, $M_\infty = 2$; $\text{Re} = 106$; $\alpha = 10^\circ$. (a) Pressure map; (b) Mach number map; (c) density map.

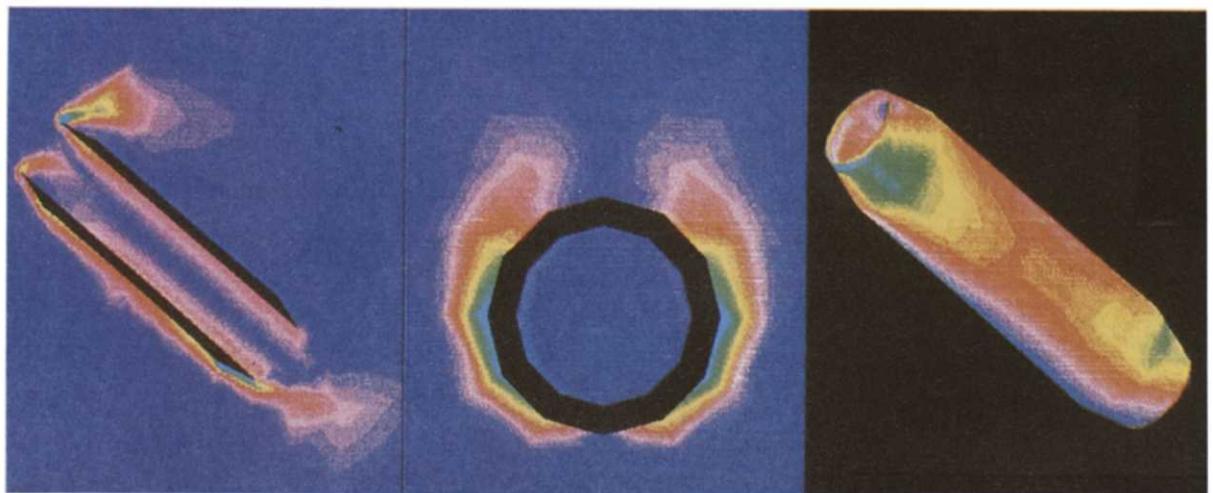


Fig. 16.21. Vorticity maps. $M_\infty = 0.9$, $\text{Re} = 100$, $\alpha = 40^\circ$. (a) Vertical section; (b) cross section; (c) on the inlet.

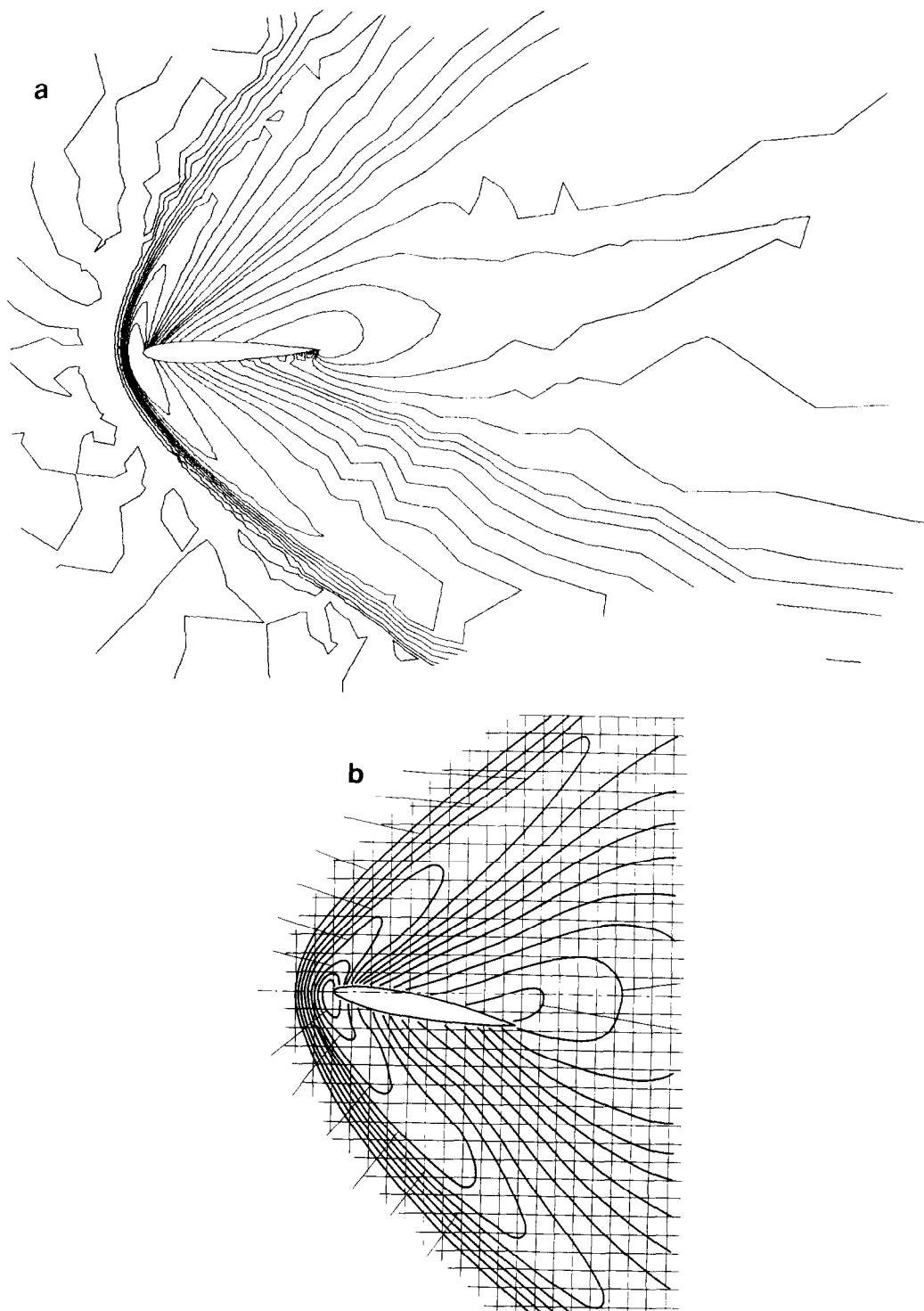


Fig. 16.7. Comparison of density contours. NACA 0012 airfoil, $M_\infty = 2$; $Re = 106$; $\alpha = 10$. (a) Computational results; (b) experimental results.

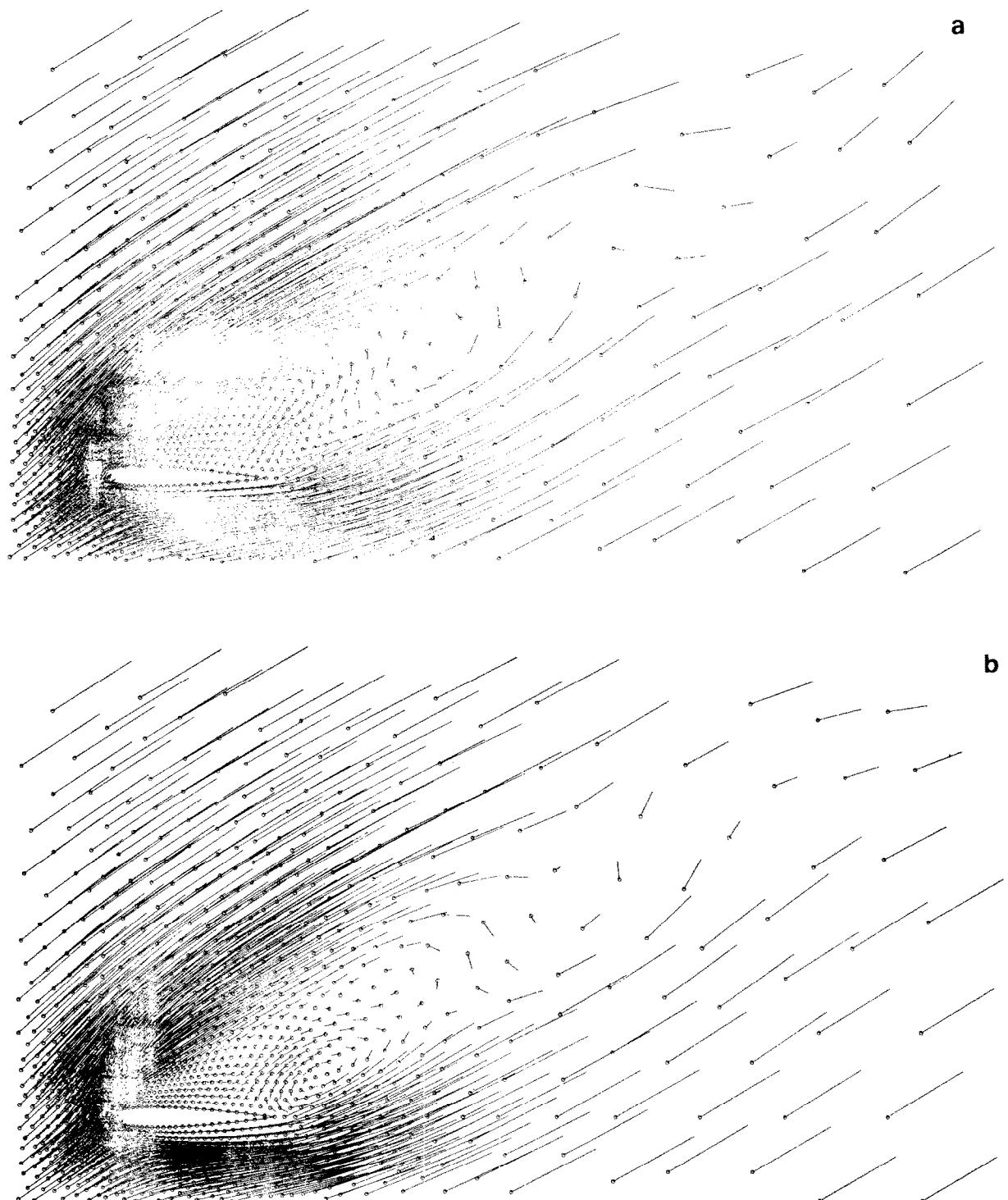


Fig. 16.8. Velocity distribution, $M_\infty = 0.7$; $\text{Re} = 100$; $\alpha = 30^\circ$; $\Delta t = 0.2$. (a) $t = 8$; (b) $t = 12$; (c) $t = 16$; (d) $t = 20$; (e) $t = 24$; (f) $t = 28$; (g) $t = 32$; (h) $t = 36$.



Fig. 16.8 (continued).

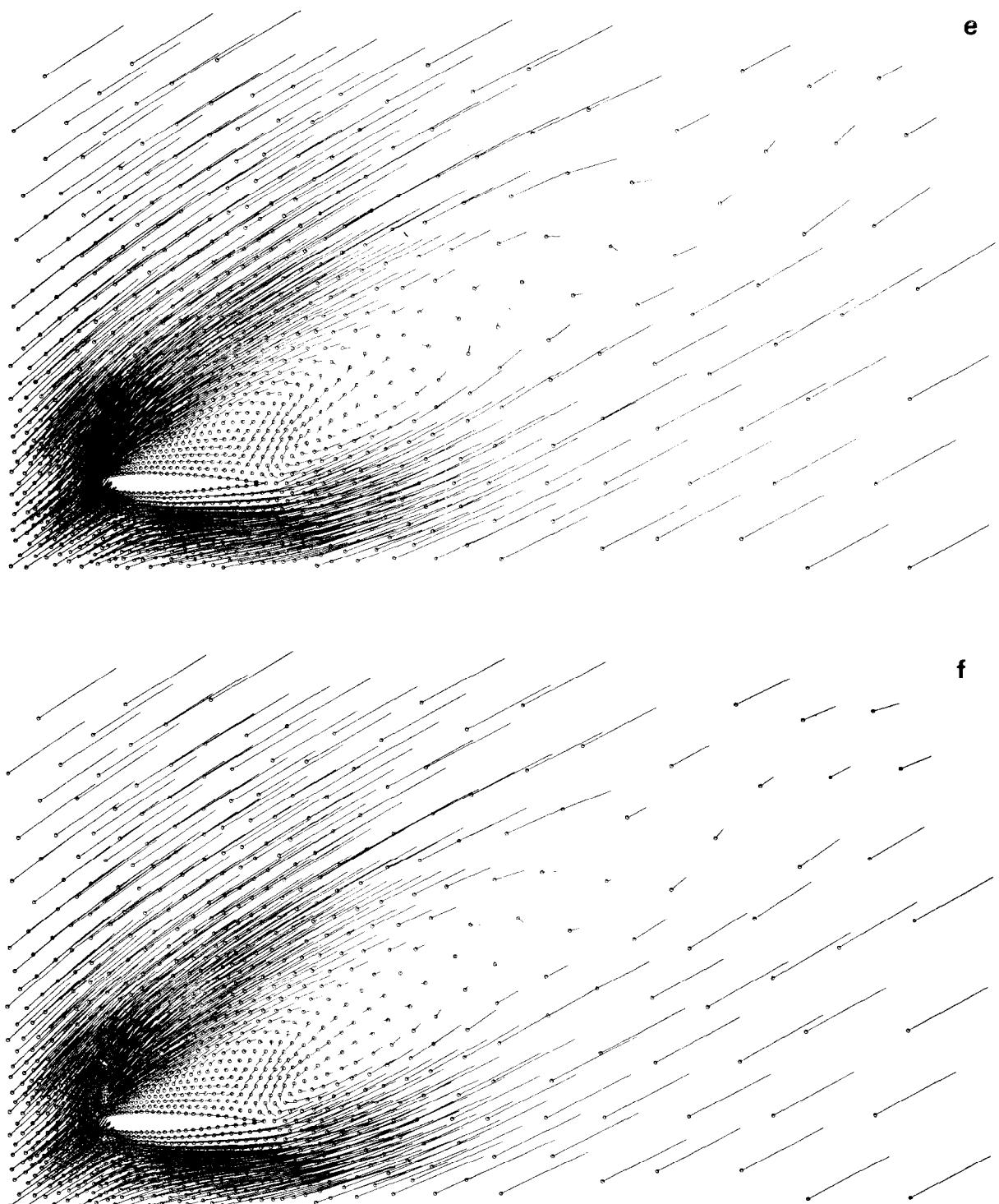


Fig. 16.8 (continued).

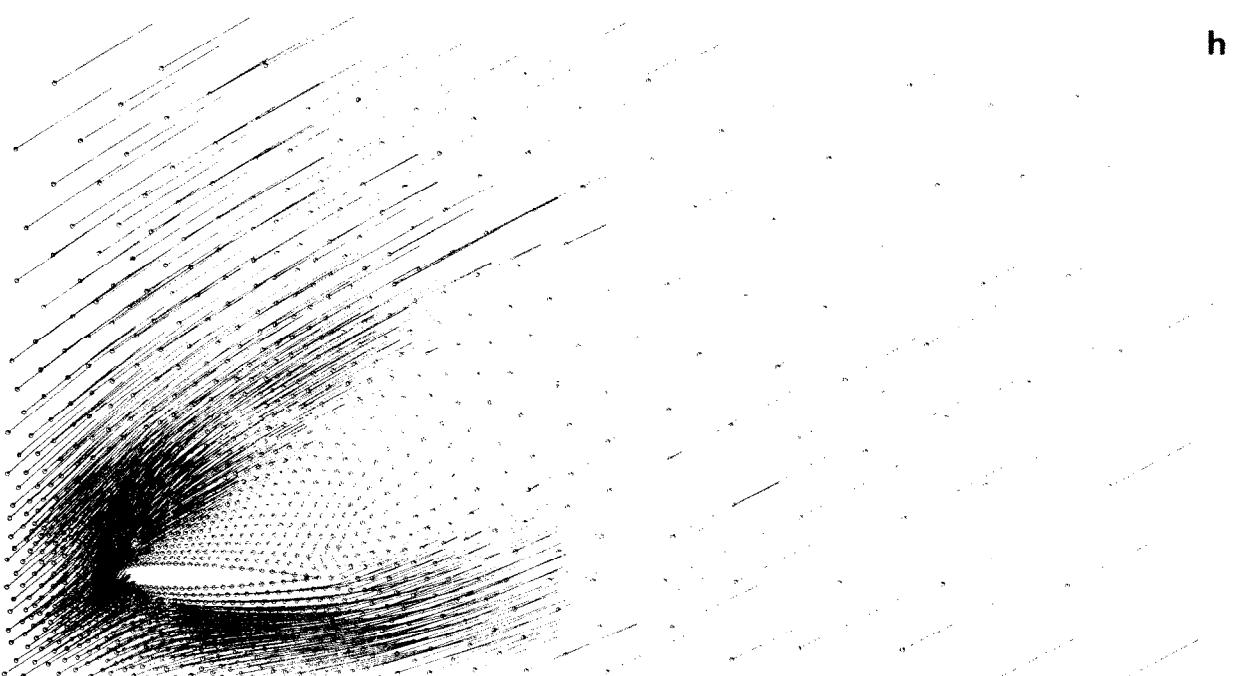
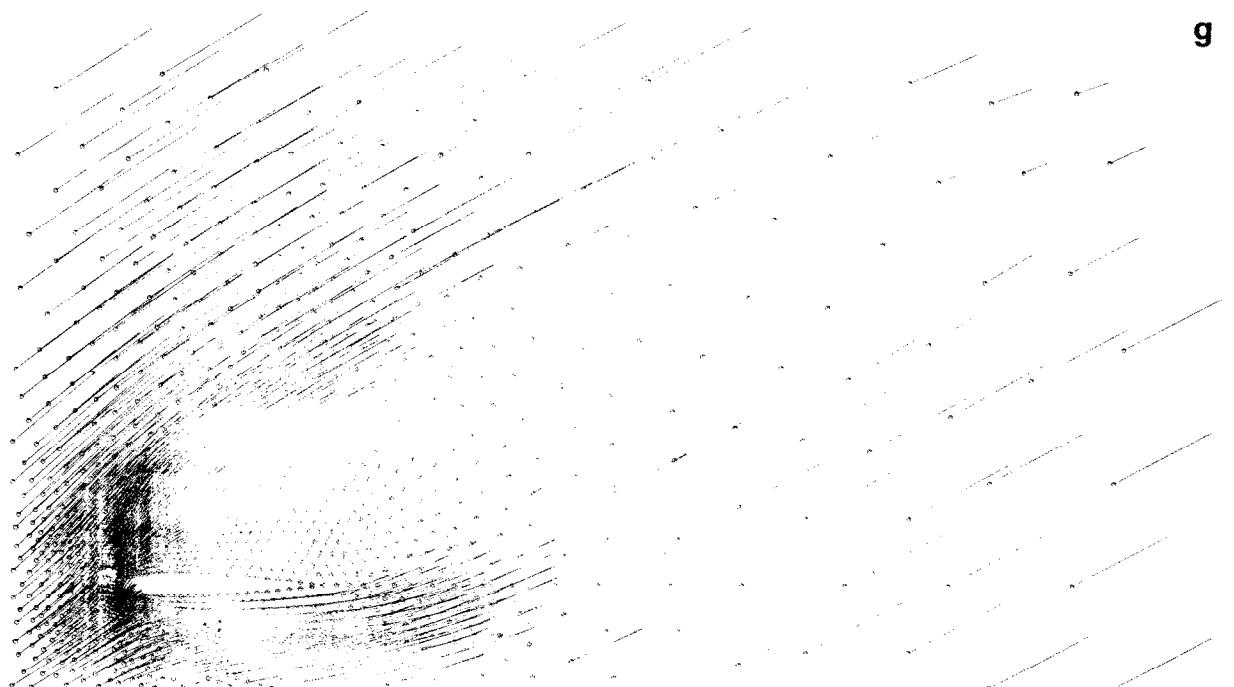


Fig. 16.8 (continued).

For this case, a comparison between numerical results and measurements due to Allegre et al. [88] has been done. In fig. 16.7, we have plotted and compared the density lines obtained with the numerical method and the experiments.

It can be assumed that the computational results would be still in better agreement with measurements if the adherence condition on velocity prescribed at the rigid boundaries, was replaced by some velocity distribution taking more closely into account the boundary layer phenomenon.

Finally, we have also considered an unsteady problem around the same NACA0012 airfoil with the following data: $M_\infty = 0.7$, $Re = 100$, $\alpha = 30^\circ$. A time dependent solution has been obtained starting from a free-stream initial solution. Fig. 16.8 describes the vortex pattern of the flow at different time steps ($t = 8, 12, 16, \dots, 36$, selected time step $\Delta t = 0.2$) on which we can observe the creation and propagation of vortices in the deviated wake of the airfoil.

16.2. A second class of problems

A second class of problems that we consider is much more complicated than the first one, since it concerns the unsteady simulation of a transonic viscous flow around and inside a 2D nozzle at high incidence $\alpha = 30^\circ$, mach number $M_\infty = 0.9$ and Reynolds number $Re = 100$ (the characteristic length being the distance between the walls of the nozzle). Fig. 16.9 shows the details of an adapted grid close to the air intake obtained with adaptive mesh refinement techniques (section 15.). Figs. 16.10, 16.11 show the Mach lines and the vortex pattern at different time cycles ($t = 16., 20., \dots, 44.$). We can observe the creation of vorticity at the

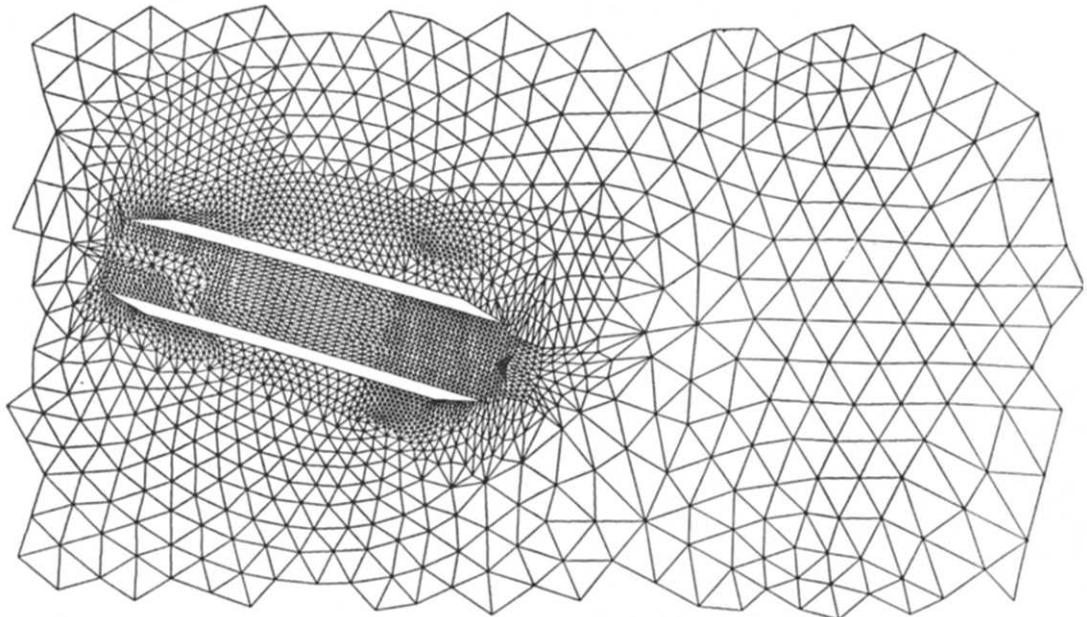


Fig. 16.9. Adapted mesh around and inside an inlet, 3811 nodes, 7303 triangles in the global mesh.

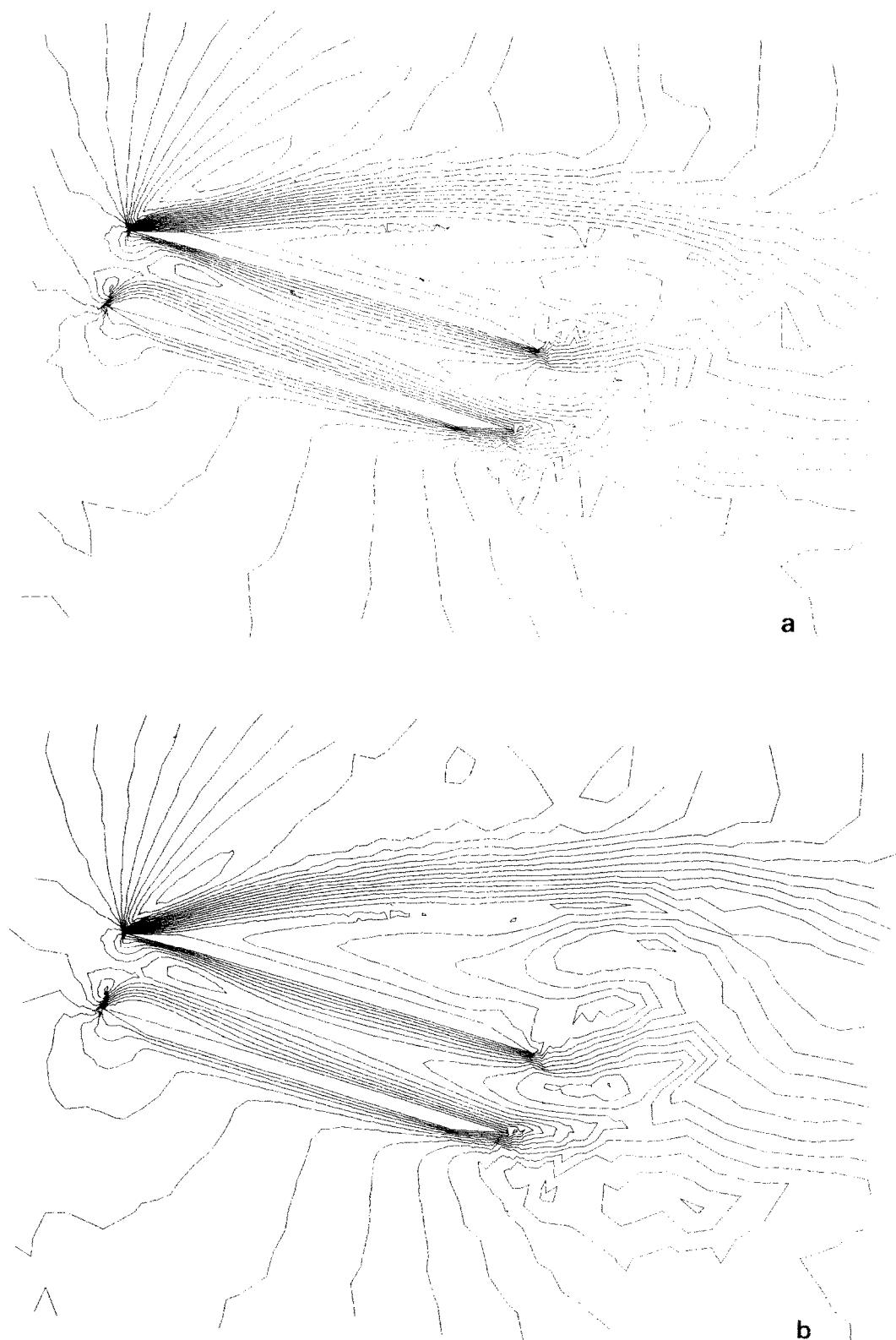


Fig. 16.10. Flow around and inside an inlet. Mach lines, $M_\infty = 0.9$, $Re = 100$; $\alpha = 30^\circ$; $\Delta t = 0.2$. (a) $t = 16$; (b) $t = 20$; (c) $t = 24$; (d) $t = 28$; (e) $t = 32$; (f) $t = 36$; (g) $t = 40$; (h) $t = 44$.

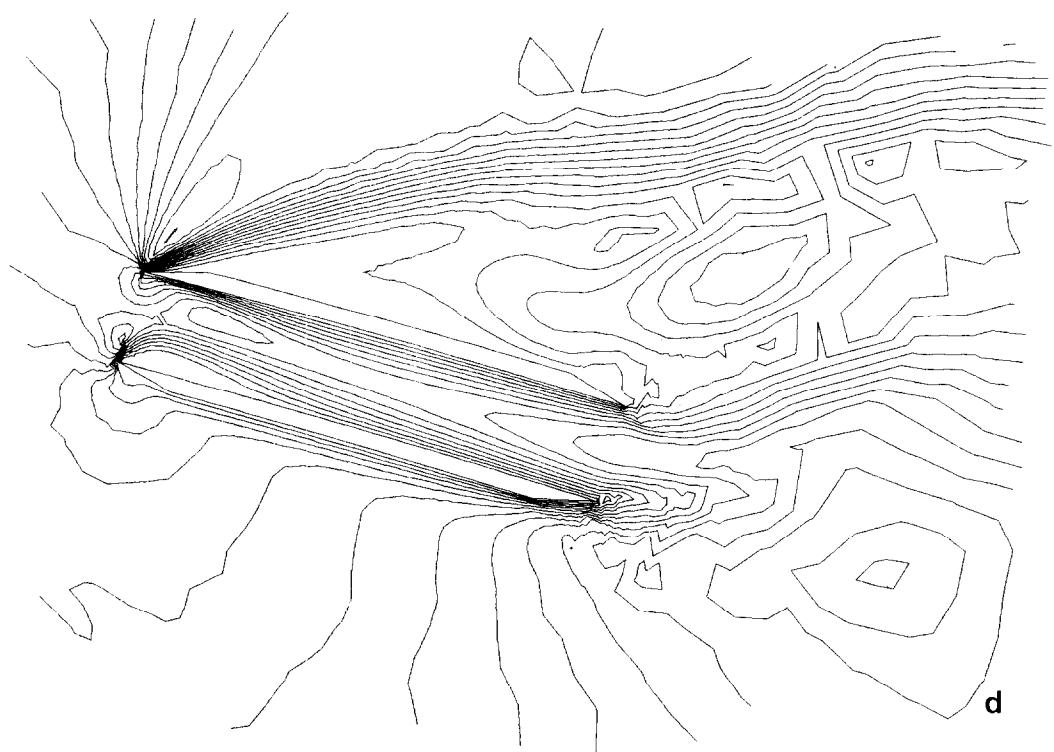
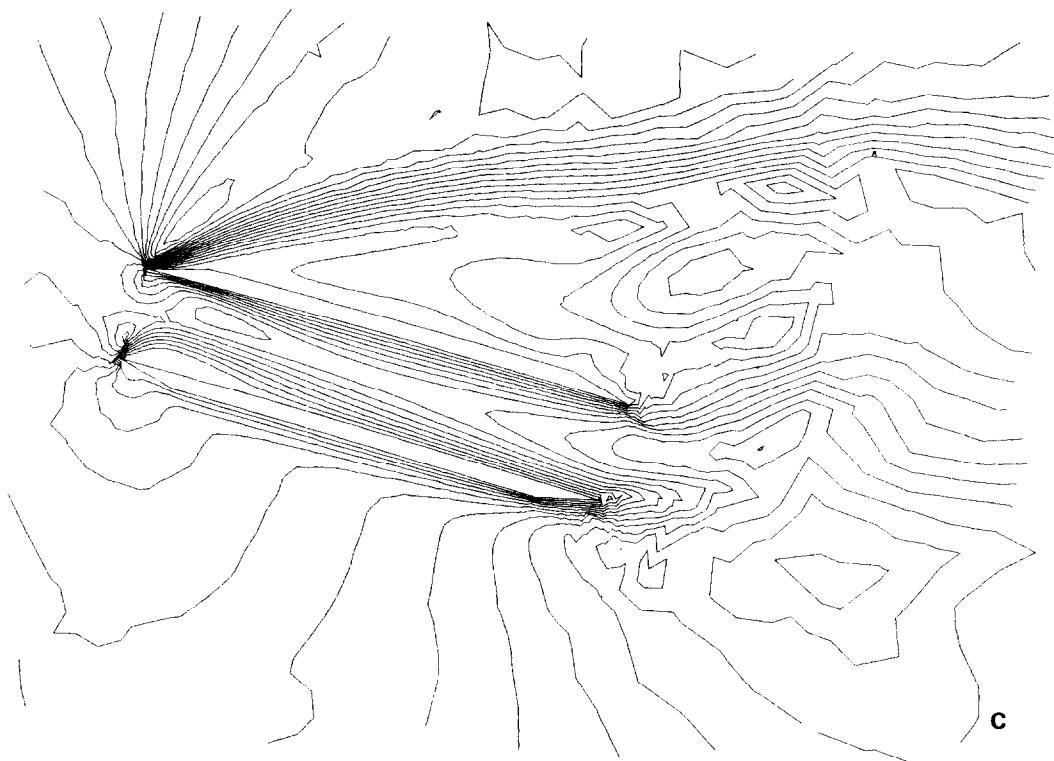


Fig. 16.10 (continued).

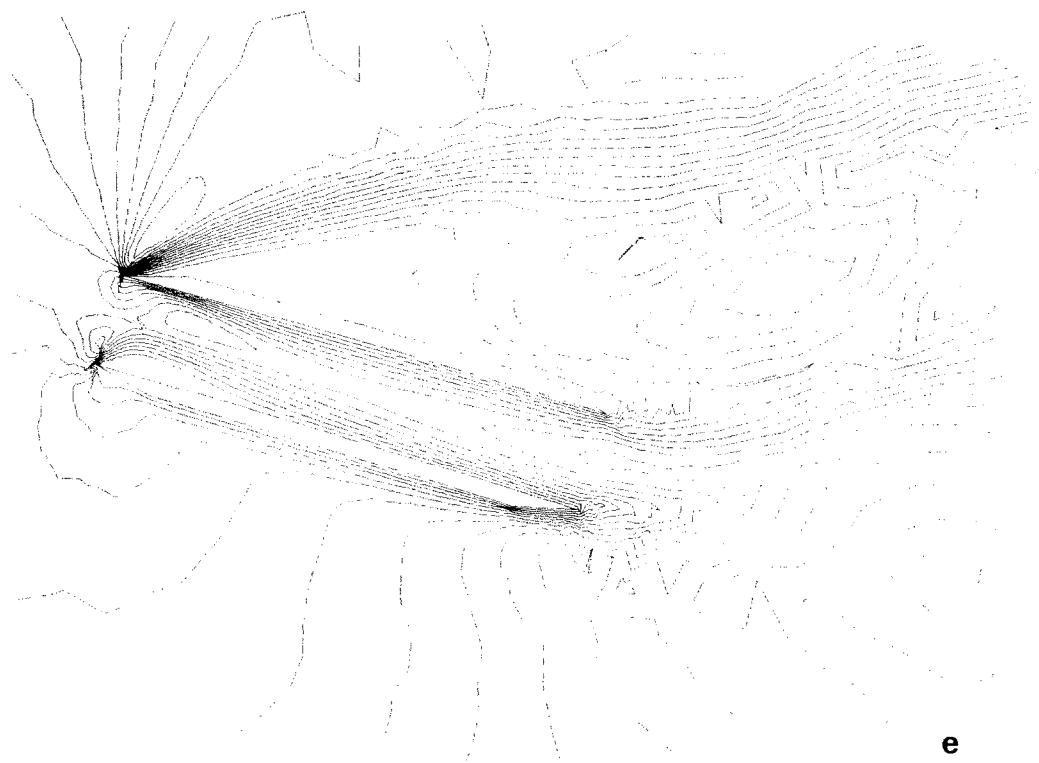
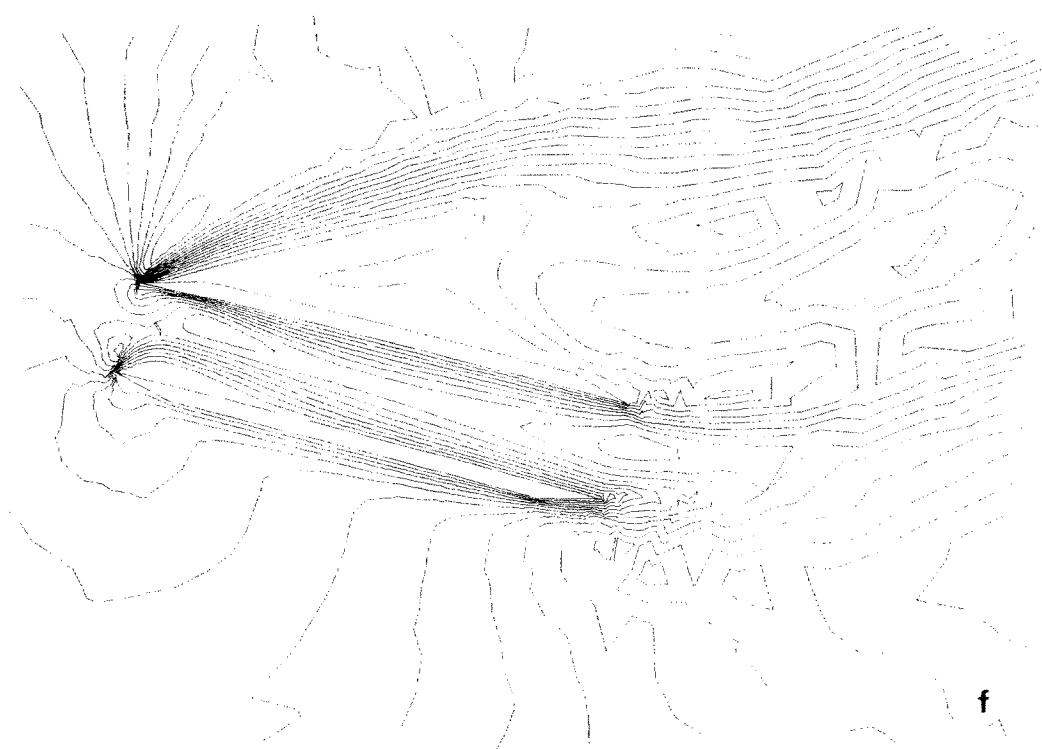
**e****f**

Fig. 16.10 (continued).

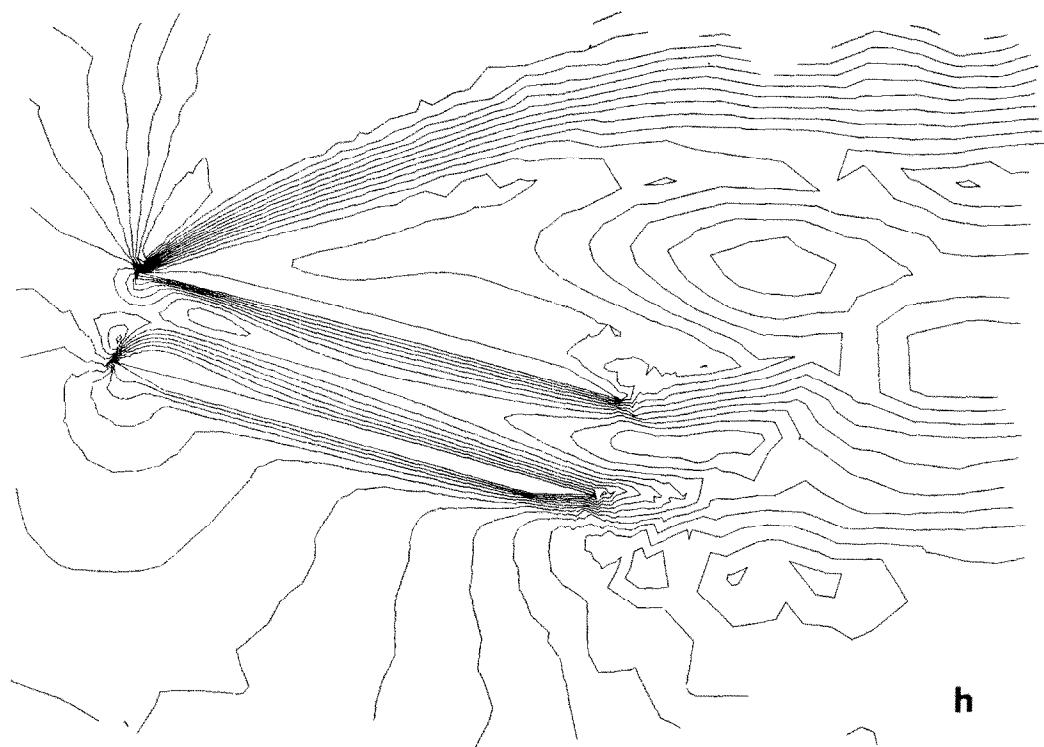
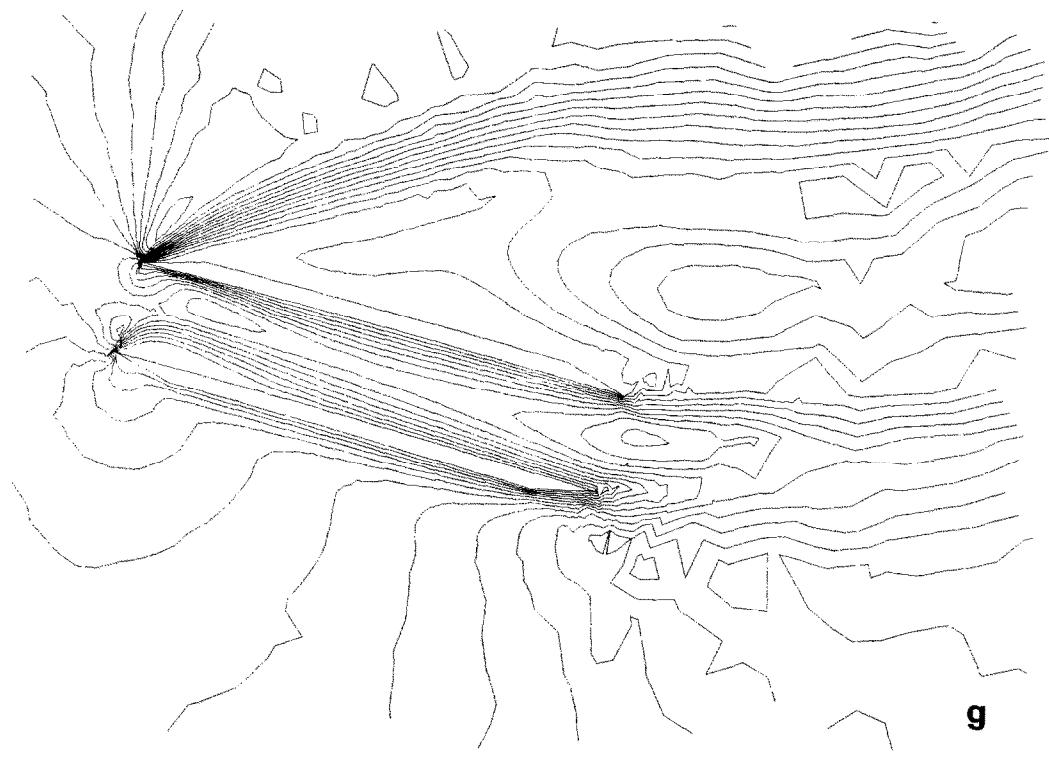


Fig. 16.10 (continued).

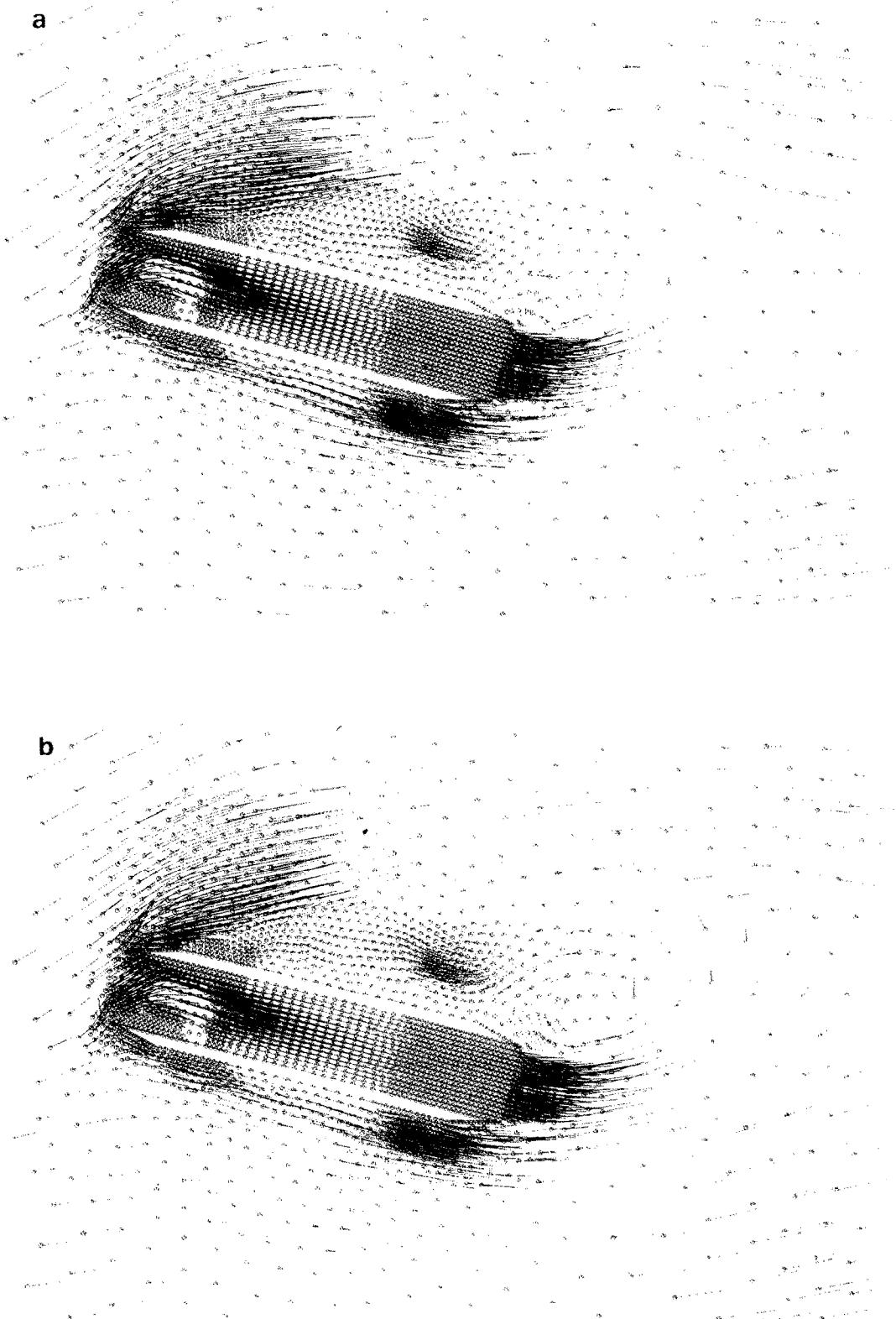


Fig. 16.11. Flow around and inside an inlet, velocity distribution. $M_\infty = 0.9$; $\text{Re} = 100$; $\alpha = 30^\circ$; $\Delta t = 0.2$ (a) $t = 16$; (b) $t = 20$; (c) $t = 24$; (d) $t = 28$; (e) $t = 32$. (f) $t = 36$; (g) $t = 40$; (h) $t = 44$.

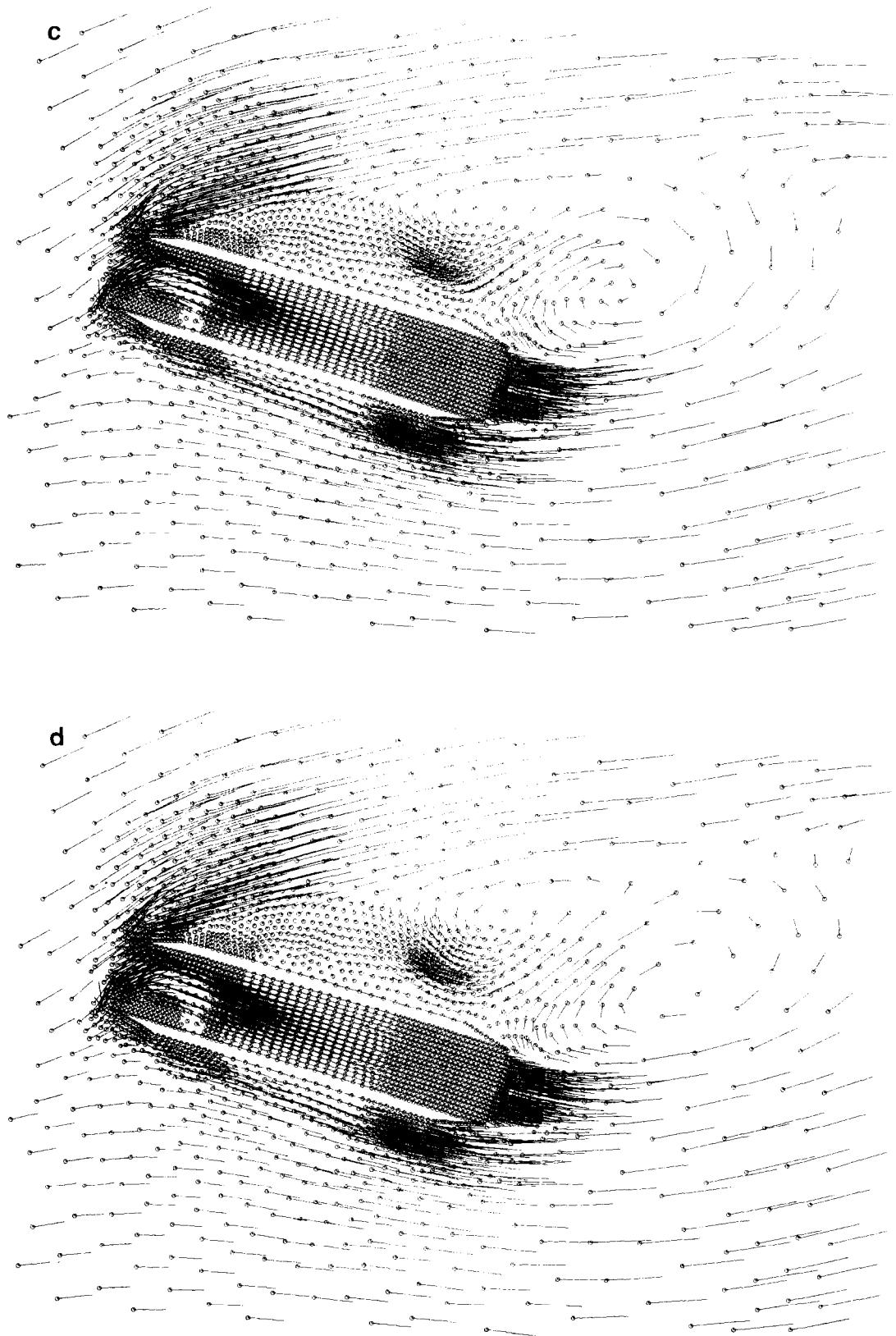


Fig. 16.11 (continued).

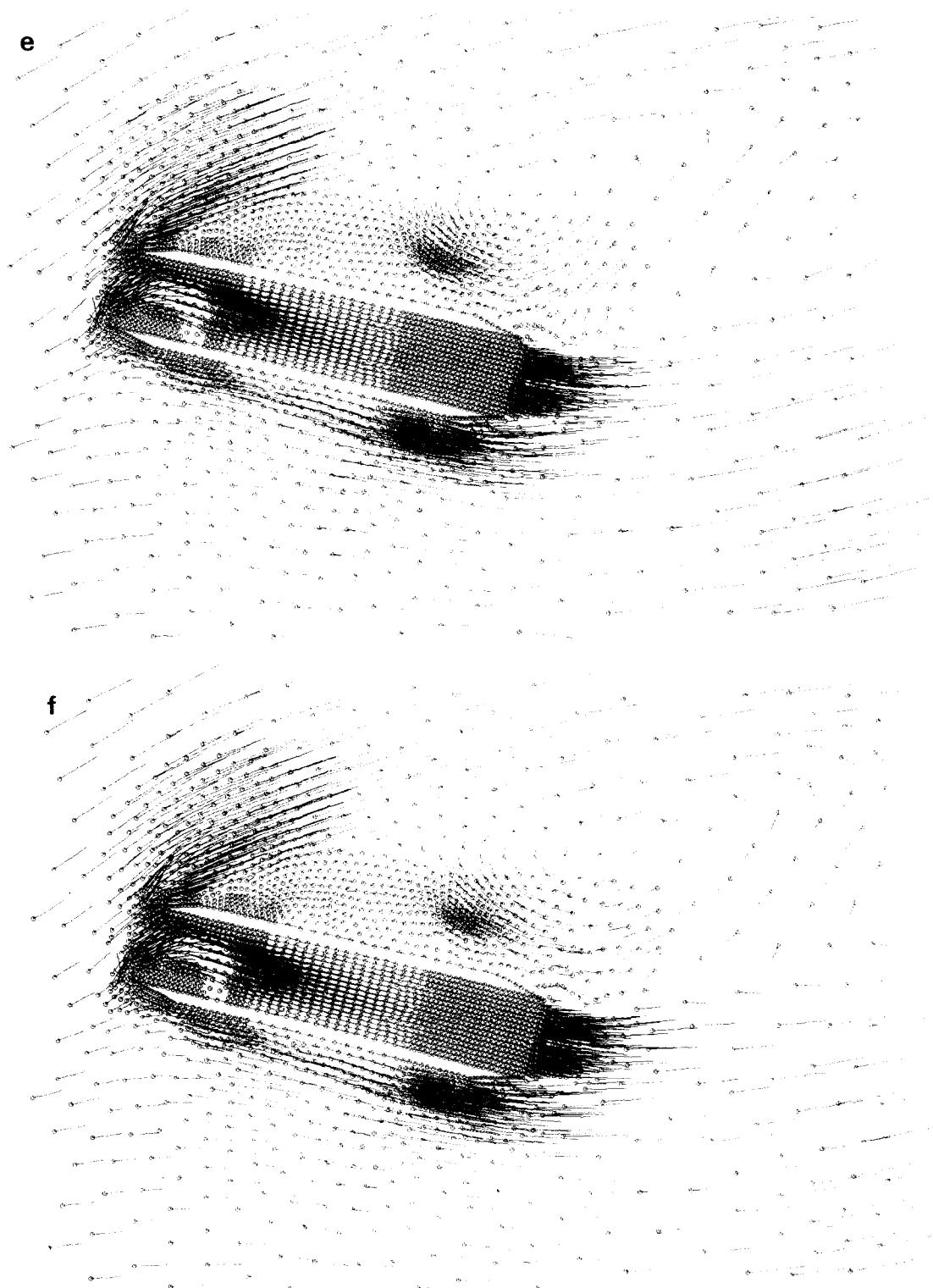


Fig. 16.11 (continued).

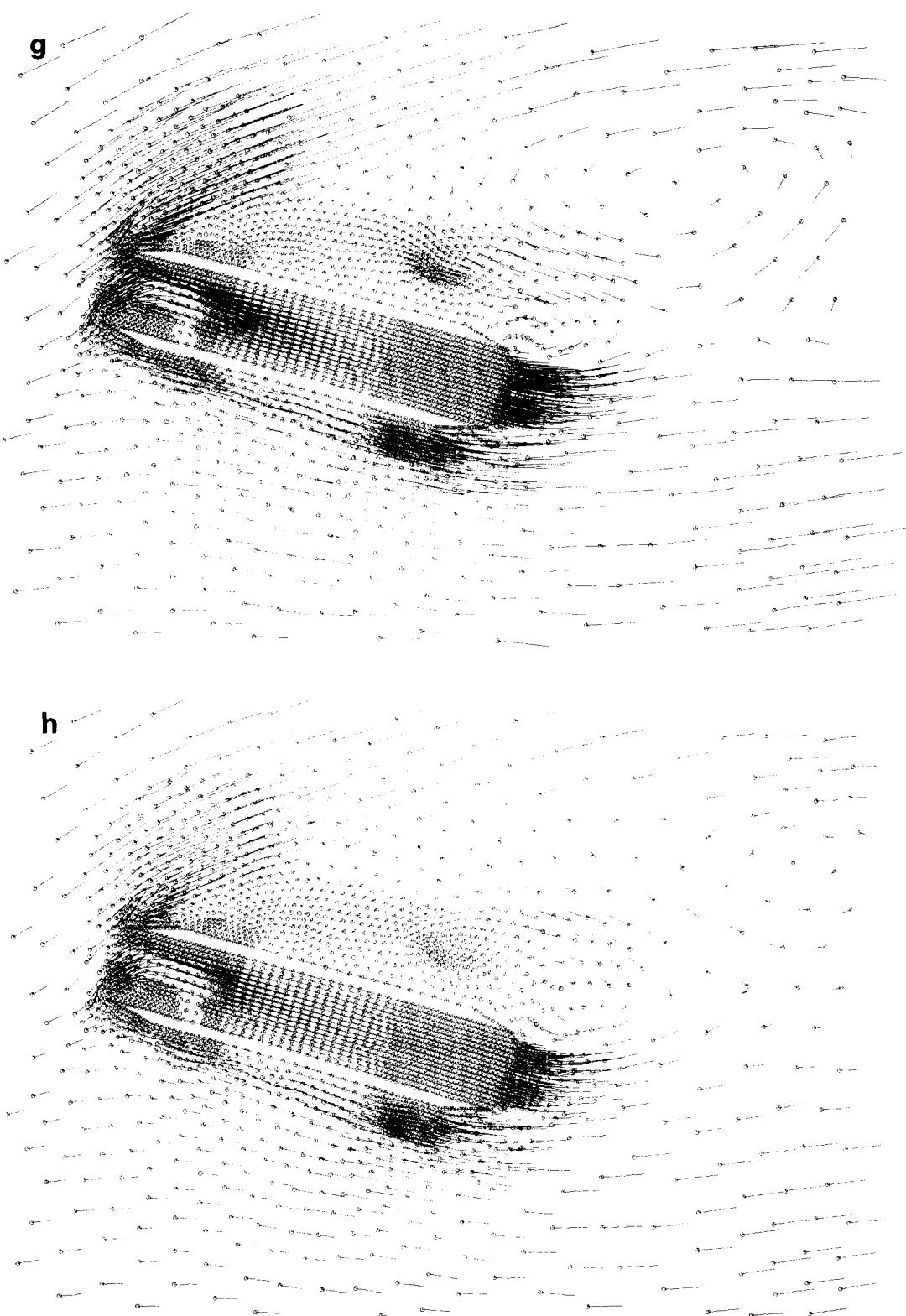


Fig. 16.11 (continued).

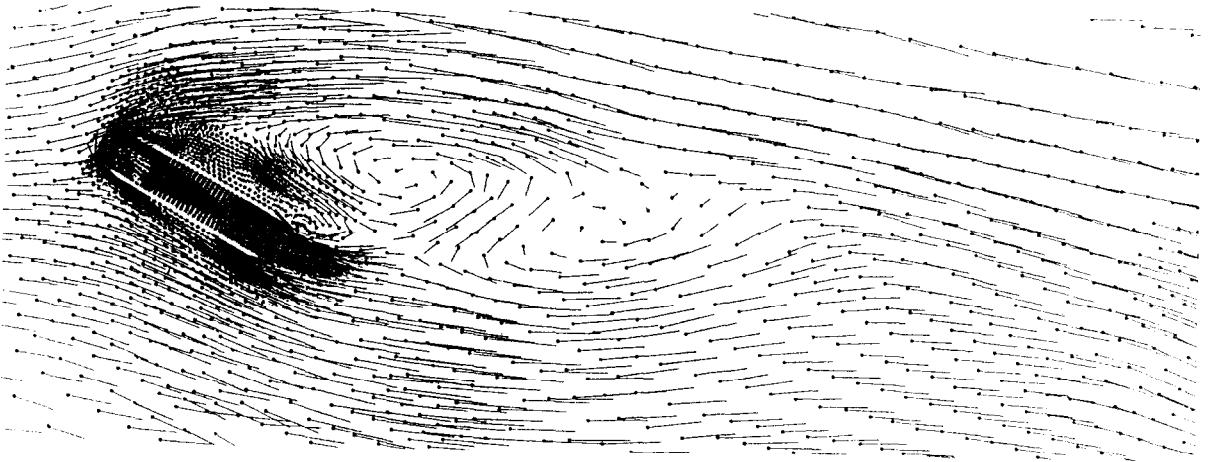


Fig. 16.12. Global view of the deviated wake behind the inlet. $M_\infty = 0.9$; $Re = 100$; $\alpha = 30^\circ$.

leading edge and the growing structures which propagate around the inlet. A global view of the deviated wake behind the inlet is shown in fig. 16.12.

16.3. 2D and 3D Navier–Stokes solutions using self-adaptive mesh refinements

The methodology described in section 15 has been applied to various numerical 2D and 3D simulations around geometries of industrial interest in order to improve unexpensively local accuracy of the Navier–Stokes solutions.

16.3.1. Local enrichment around a NACA0012

The main features of a viscous flow are difficult to identify due to its complexity: physical criteria in preferred directions have to be found to localize not only discontinuities but also boundary layers and wakes and a very important work remains still open in the broad area of unsteady flow with moving features. In the following, we will concentrate only on steady solutions. We consider the numerical simulation of a supersonic flow at $M_\infty = 2$, $Re = 106$, at an angle of attack of $\alpha = 10^\circ$ (section 16.1). Starting from a computation on a coarse mesh (800 nodes, 1 514 triangles), two successive refined meshes are generated as shown in fig. 16.13 using criterion $C_1 + C_6$ (see section 15.3). Comparison of the corresponding Mach contours (fig. 16.14) shows a sensitive improvement of the accuracy with an increasing stability of the lines and also a better capture of the shock layer.

16.3.2. 3D Navier–Stokes solution around an inlet

Finally a 3D computation of industrial interest around and inside an idealized air intake at $M_\infty = 0.9$, $Re = 100$, and $\alpha = 40^\circ$, using local enrichment by self adaptive procedure has been also performed. The initial mesh is depicted in figs. 16.15a–16.17a by a view of the inlet and by vertical sections in the flow direction and perpendicular to the flow direction. The mesh is locally

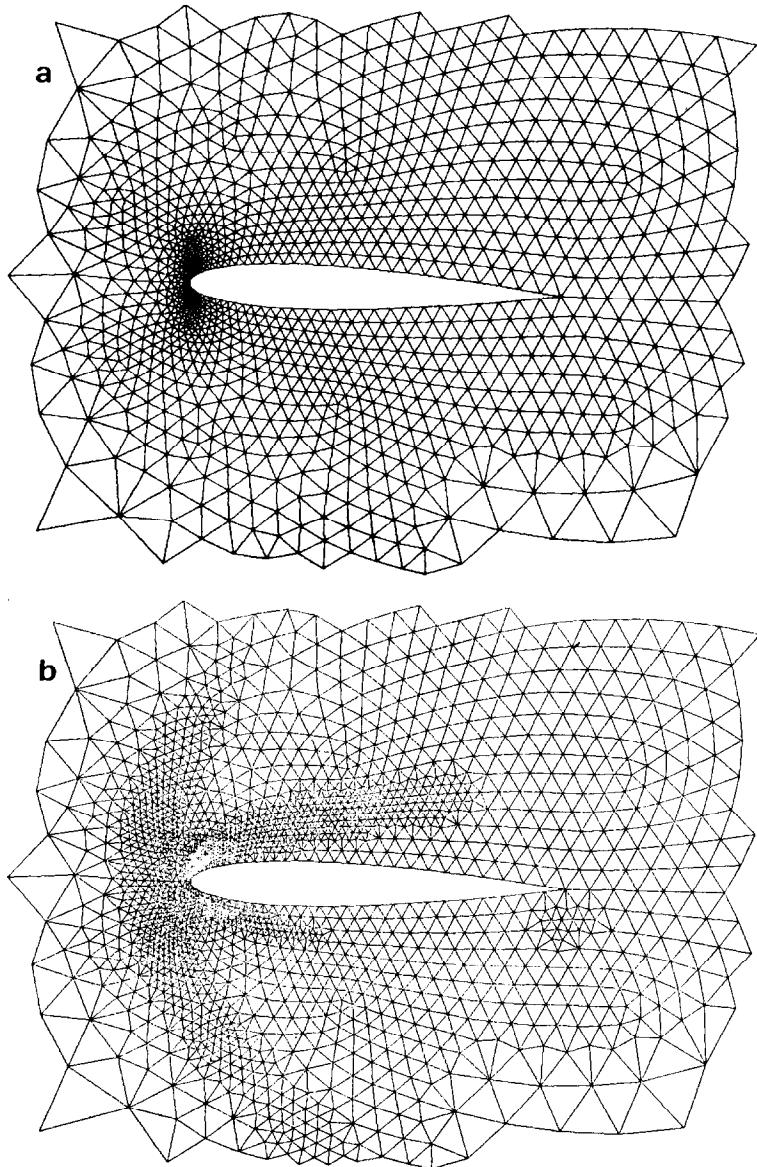


Fig. 16.13. Adapted meshes. Criterion $C_1 + C_6$. $M_\infty = 2$; $Re = 106$; $\alpha = 10^\circ$. (a) Nodes: 1797, elements: 3474; (b) nodes: 2646, elements: 5172.

enriched using criterion C_4 , an exceeded threshold of the vorticity generates finer elements in the vicinity of the leading and trailing edges of the air intake. After two successive enrichments, the adapted mesh depicted by figs. 16.15b–16.17b is obtained. The pressure and Mach contours of the solution computed on the initial coarse mesh are shown in fig. 16.18a, 16.19a, while the same iso lines of the solution on the refined mesh are presented in figs. 16.18b, 16.19b. A significant improvement of the quality of pressure and Mach lines can be observed specially in the vicinity



Fig. 16.14. Mach contours on adapted meshes, $M_\infty = 2$; $Re = 106$, $\alpha = 10^\circ$. (a) Nodes: 1797, elements: 3474; (b) nodes: 2646, elements: 5172.

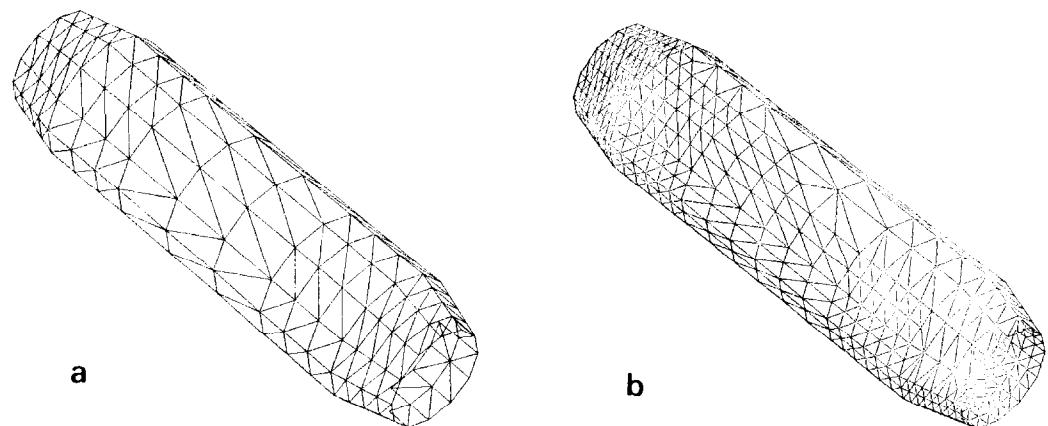


Fig. 16.15. Details of the initial and adapted meshes on a 3D inlet.

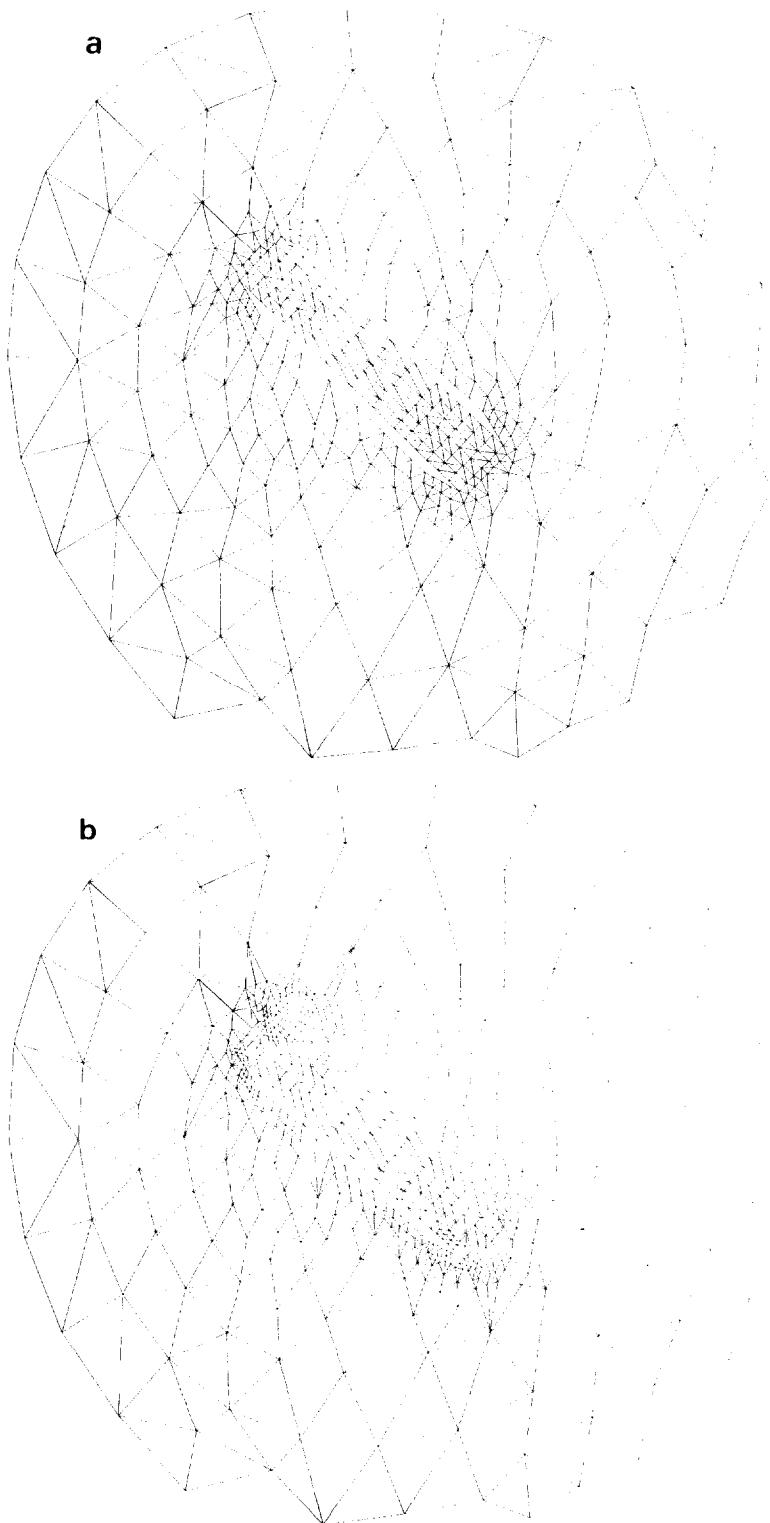


Fig. 16.16. Initial and adapted meshes around and inside a 3D inlet. Vertical section in flow direction. (a) Initial mesh, nodes: 3096, elements: 16632; (b) adapted mesh: nodes: 6958, elements: 34760.

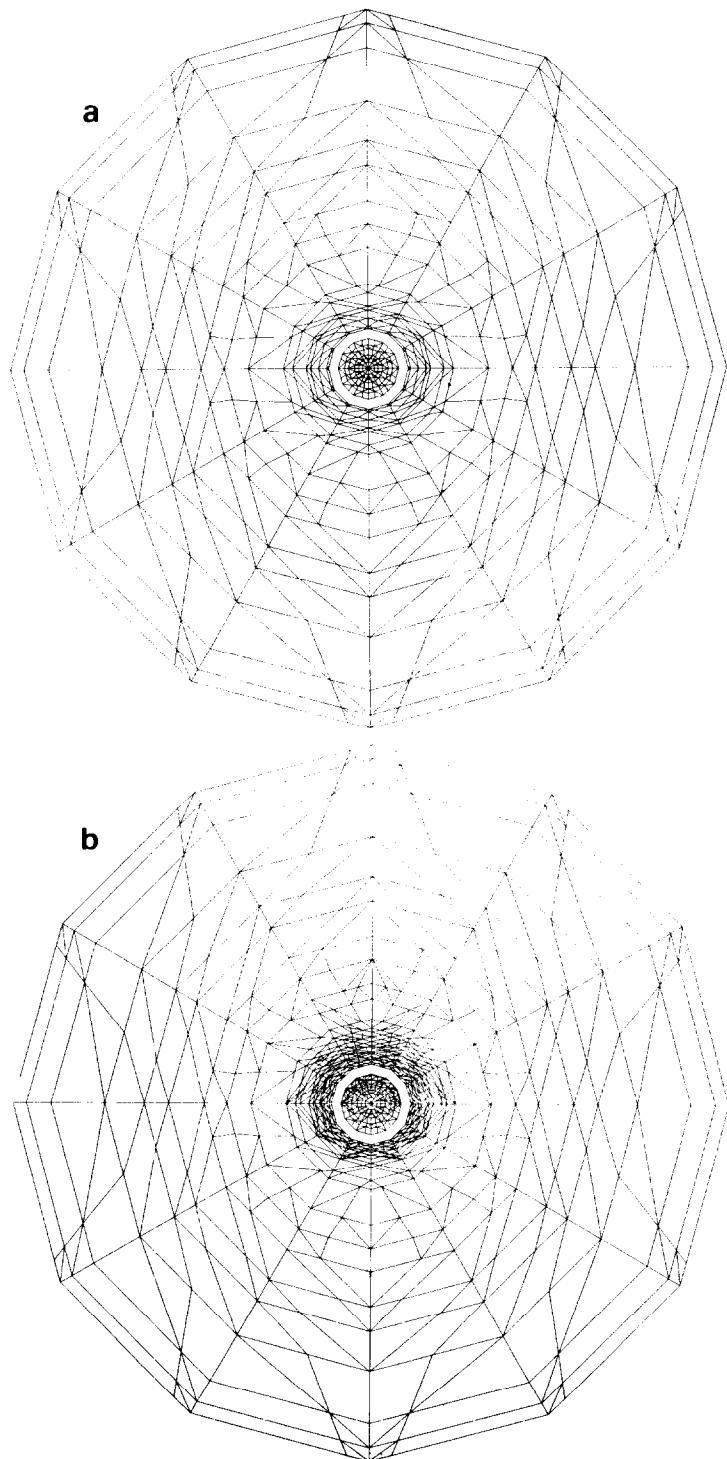


Fig. 16.17. Vertical section perpendicular to the flow direction.



Fig. 16.18. Pressure lines around and inside a 3D inlet. Vertical section in the flow direction. $M_{\infty} = 0.9$, $Re = 100$, $\alpha = 40^\circ$.



Fig. 16.19. Mach lines around and inside a 3D inlet. Vertical section in the flow direction. $M_\infty = 0.9$, $Re = 100$, $\alpha = 40^\circ$.

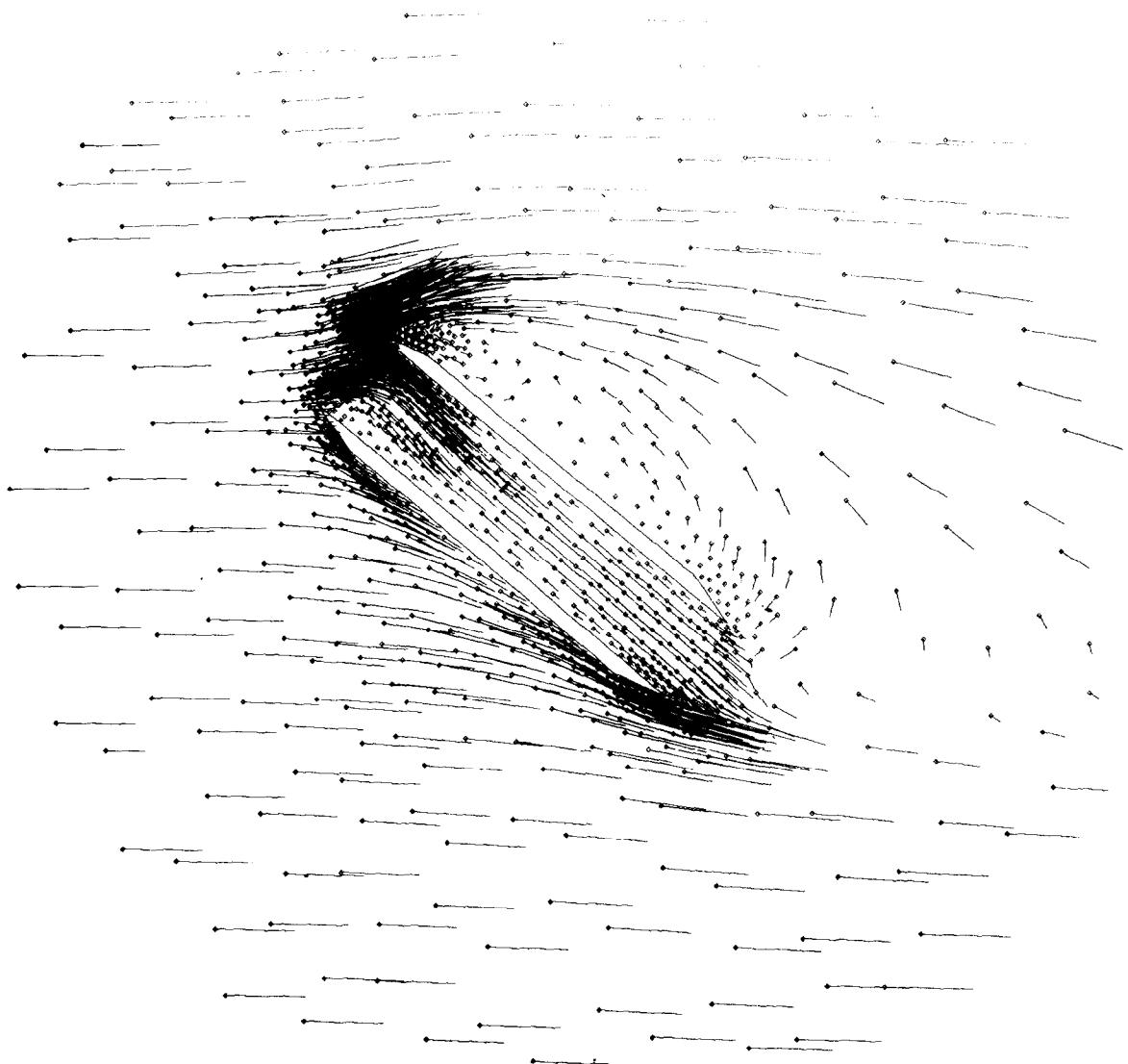


Fig. 16.20. Velocity distribution. Vertical section in the flow direction. $M_\infty = 0.9$, $Re = 100$, $\alpha = 40^\circ$.

of the inlet entrance and also in the boundary layer regions. Fig. 16.20 shows the velocity distribution in a vertical section, obtained on the finest grid and in fig. 16.21 (see page 159) are presented different views of the vorticity distribution. As a solution has been now obtained with the above enrichment procedure, the next step will be to accelerate its convergence via adapted full multigrid techniques.

Conclusion

We have presented in this paper a family of methods for solving viscous flow problems modeled by the Navier–Stokes equations. Efficient and accurate methods exist now for incompressible flow simulations on practically any geometry, at Reynolds numbers of the order of 10^3 to 10^4 , in two and three dimensions. Recent progresses suggest that a similar situation may hold very soon concerning compressible flow problems.

However a lot of theoretical and computational work has still to be done in order to extend the above methods to turbulent and/or reacting flows.

Acknowledgements

The results and methods described in this report have been obtained in cooperation with many colleagues in France and abroad through fruitful discussions. We would like to mention most particularly, J. Cahouet, J. Goussebaile, A. Hauguel, A.K.M.F. Hussain, G. Labadie, B. Palmerio, P. Perrier, O. Pironneau, C. Pouletty, G. Singh, T. Tezduyar, M.F. Wheeler, and emphasize on the important role played by C. Begue, E.J. Dean, A. Fischler and most particularly B. Mantel, who have implemented most of the methods described in this paper.

We would like to thank also Mrs. C. Barny, M. Desnous, C. Dubois and Y. Fisk for the typing of the manuscript, and last, but not least, R. Gruber for his quasi infinite patience with us. The support of DRET under grants 83/403, 86/34, and of CRAY Research is also acknowledged.

References

- [1] R. Glowinski, Numerical Methods for Nonlinear Variational Problems (Springer-Verlag, New York, 1984).
- [2] H. Berestycki, E. Fernandez-Cara and R. Glowinski, A numerical study of some questions in vortex rings theory, R.A.I.R.O. Numer. Anal. 18 (1984) 1.
- [3] J.L. Lions, Quelques Méthodes de Résolution des Problèmes aux Limites Non Linéaires (Dunod, Paris, 1969).
- [4] O.A. Ladyzenskaya, The Mathematical Theory of Viscous Incompressible Flows (Gordon and Breach, New York, 1969).
- [5] L. Tartar, Topics in Nonlinear Analysis (Publications Mathématiques d'Orsay, Université Paris-Sud, Département de Mathématiques, Paris, 1978).
- [6] R. Temam, Navier–Stokes Equations (North-Holland, Amsterdam, 1977).
- [7] R. Temam, Navier–Stokes Equations and Nonlinear Functional Analysis (CBMS 41, SIAM, Philadelphia, PA, 1983).
- [8] R. Rautmann, ed., Approximation Methods for Navier-Stokes problems, Lecture Notes in Mathematics, 771 (Springer-Verlag, Berlin, 1980).

- [9] V. Girault and P.A. Raviart, Finite Element Methods for Navier–Stokes Equations (Springer-Verlag, Berlin, 1986).
- [10] F. Thomasset, Implementation of Finite Element Methods for Navier–Stokes Equations (Springer-Verlag, New York, 1981).
- [11] R. Peyret and T.D. Taylor, Computational Methods for Fluid Flow (Springer-Verlag, New York, 1982).
- [12] P.J. Roache, Computational Fluid Dynamics (Hermosa, Albuquerque, NM 1972).
- [13] G. Strang, On the construction and comparison of difference schemes, SIAM J. Numer. Anal. 5 (1968) 506.
- [14] J.T. Beale and A. Majda, Rates of convergence for viscous splitting of the Navier–Stokes equations, Math. Comput. 37 (1981) 243.
- [15] R. Levêque, Time-Split Methods for Partial Differential Equations, Ph.D. Thesis, Computer Science Department, Stanford University, Stanford, CA (1982).
- [16] R. Levêque and J. Oliger, Numerical methods based on additive splittings for hyperbolic partial differential equations, Math. Comput. 40 (1983) 469.
- [17] P.L. Lions and B. Mercier, Splitting algorithms for the sum of two operators. SIAM J. Numer. Anal. 16 (1979) 964.
- [18] E. Godlewski, Méthodes à pas multiples et de directions alternées pour la discréétisation d'équations d'évolution, Thèse de 3ème cycle, Université P. et M. Curie, Paris (1980).
- [19] G.I. Marchuk, Methods of Numerical Mathematics (Springer-Verlag, New York, 1975).
- [20] R. Glowinski, B. Mantel and J. Periaux, Numerical solution of the time dependent Navier–Stokes equations for incompressible viscous fluids by finite element and alternating direction methods, in: Numerical Methods in Aeronautical Fluid Dynamics, ed. P.L. Roe (Academic Press, London, 1982) p. 309–336.
- [21] J.L. Lions, Problèmes aux Limites dans les Equations aux Dérivées Partielles, Presses de l'Université de Montréal, Montréal, P.Q., Canada.
- [22] J. Nečas, Les Méthodes Directes en Théorie des Equations Elliptiques (Masson, Paris, 1967).
- [23] R.A. Adams, Sobolev Spaces (Academic Press, New York, 1971).
- [24] J.T. Oden and J.N. Reddy, An Introduction to the Mathematical Theory of Finite Elements (Wiley, New York, 1976).
- [25] E. Polak, Computational Methods in Optimization (Academic Press, New York, 1971).
- [26] J. Daniel, The Approximate Minimization of Functionals (Prentice Hall, Englewood Cliffs, NJ, 1970).
- [27] R. Glowinski, H.B. Keller and L. Reinhart, Continuation-conjugate gradient methods for the least squares solution of nonlinear boundary value problems, SIAM J. Sci. Stat. Comput. 6 (1985) 793.
- [28] L. Reinhart, On the numerical analysis of the Von Karman equations: Mixed Finite element approximation and continuation techniques, Numer. Math. 39 (1982) 371.
- [29] R. Glowinski, B. Mantel, J. Periaux, P. Perrier and O. Pironneau, On an efficient new preconditioned conjugate gradient method. Application to the in core solution of the Navier–Stokes equations, in: Finite Element in Fluids, vol. 4, eds. R.H. Gallagher, D.H. Norrie, J.T. Oden, O.C. Zienkiewicz (Wiley, London, 1982) p. 365–401.
- [30] G. Birkhoff and A. Schoenstadt, Elliptic Problem Solvers, vol. 2 (Academic Press, New York, 1984).
- [31] D.F. Shanno, Conjugate gradient method with inexact line search, Math. of Oper. Research 13 (1978) 155.
- [32] A. Buckley and A. Lenir, QN-like variable storage conjugate gradient methods, Math. Programming 27 (1983) 155.
- [33] M.O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, O. Pironneau and G. Poirier, On the numerical solution of nonlinear problems in fluid dynamics by least squares and finite element methods (II). Application to transonic flow simulations, Comput. Meth. Appl. Mech. Eng. 51 (1985) 363.
- [34] M. Crouzeix, Etude d'une méthode de linéarisation. Réésolution numérique des équations de Stokes stationnaires. Application aux équations de Navier-Stokes stationnaires, in: Approximation et Méthodes Itératives de Résolution d'Inéquations Variationnelles et de Problèmes Non Linéaires, Cahier de l'INRIA 12 (1974) 139.
- [35] O. Axelsson and V.A. Barker, Finite Element Solution of Boundary Value Problems (Academic Press, Orlando, 1984).
- [36] J. Cahouet and J.P. Chabard, Multi-Domains and Multi-Solvers Finite Element Approach for the Stokes problem, in: Innovative Numerical Methods in Engineering, eds., R.P. Shaw, J. Periaux, A. Chaudouet, J. Wu, C. Marino, C.A. Brebbia (Springer-Verlag, Berlin, 1986) p. 317–322.
- [37] A. Hauguel and J. Cahouet, Finite Element Methods for Incompressible Navier–Stokes and for Shallow Water

- Equations, Lecture Notes at the Von Karman Institute, March 1986, Report HE41/86.03, Electricité de France, Chatou.
- [38] R. Glowinski, J. Goussebaile and G. Labadie, eds., Numerical Methods for the Stokes Problem, Application to Compressible and Incompressible Viscous Flow Simulation (to appear).
 - [39] R. Glowinski and P. Le Tallec, SIAM Monograph on the Applications of Augmented Lagrangian Methods to Nonlinear Mechanics (to appear).
 - [40] M. Fortin and R. Glowinski, Augmented Lagrangian Methods (North-Holland, Amsterdam, 1983).
 - [41] M.O. Bristeau, R. Glowinski, J. Periaux, P. Perrier, O. Pironneau and G. Poirier, Application of Optimal Control and Finite Element Methods to the calculation of transonic flows and incompressible viscous flows, in: Numerical Methods in Applied Fluid Dynamics, ed. B. Hunt (Academic Press, London, 1980) p. 203–312.
 - [42] M.O. Bristeau, R. Glowinski, B. Mantel, J. Periaux, P. Perrier and L. Pironneau, A finite element approximation of Navier–Stokes equations for incompressible viscous fluids. Iterative methods of solution, in: Approximation Methods for Navier–Stokes problems, ed. R. Rautmann, Lecture Notes in Mathematics, vol. 771 (Springer-Verlag, Berlin, 1980) p. 78–128.
 - [43] P. Hood and C. Taylor, A numerical solution of the Navier–Stokes equations using the finite element technique, Computer and Fluids 1 (1973) 73.
 - [44] K. Morgan, J. Periaux and F. Thomasset, eds., Numerical Analysis of Laminar Flow over a Step, GAMM Workshop, Bièvres, France, January 1983 (Vieweg-Verlag, Braunschweig, Wiesbaden, 1984).
 - [45] J. Heywood and R. Rannacher, Finite Element Approximation of the Nonstationary Navier-Stokes Problem (I): Regularity of solutions and second-order error estimates for spatial discretization, SIAM J. Numer. Anal. 19 (1982) 275.
 - [46] J. Heywood and D. Rannacher, Finite element approximation of the nonstationary Navier–Stokes problem (II): Stability of solutions and error estimates uniform in time, SIAM J. Numer. Anal. 23 (1986) 750.
 - [47] D.N. Arnold, F. Brezzi and M. Fortin, A stable finite element for the Stokes equations, Calcolo 21 (1984) 337.
 - [48] M. Fortin and F. Thomasset, Application of augmented Lagrangian methods to the Stokes and Navier–Stokes equations, in: Augmented Lagrangian Methods, eds. M. Fortin and R. Glowinski (North-Holland, Amsterdam, 1983) p. 47–95.
 - [49] R. Glowinski and O. Pironneau, Numerical methods for the first biharmonic equation and for the two-dimensional Stokes problem, SIAM Review 21 (1979) 167.
 - [50] E.J. Dean and R. Glowinski, Finite element implementation of mixed boundary conditions for the biharmonic equations (II): Application to the Navier-Stokes equations for incompressible viscous fluids (to appear).
 - [51] P.A. Raviart and J.M. Thomas, Introduction à l’Analyse Numérique des Equations aux Dérivées Partielles (Masson, Paris, 1983).
 - [52] M. Fortin and F. Thomasset, Mixed finite element methods for incompressible flow problems, Journal of Computational Physics, 31, (1979), pp. 173–215.
 - [53] A.N. Brooks and T.J.R. Hughes, Stream line upwind / Petrov–Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Comput. Meth. Appl. Mech. Eng. 32 (1982) 199.
 - [54] T.J.R. Hughes and A.N. Brooks, A theoretical framework for Petrov–Galerkin methods with discontinuous weighting functions: application to the streamline upwind procedure, in: R.H. Gallagher, D.H. Norrie, J.T. Oden, O.C. Zienkiewicz eds., Finite Elements in Fluids, vol. IV (Wiley, London, 1982) p. 46–65.
 - [55] J.P. Benque, B. Ibler, A. Keramsi and G. Labadie, A finite element method for Navier–Stokes equations, in: Proc. Third Intern. Conf. on Finite Element Methods in Flow Problems, Banff, Alberta, Canada, 10–13 June 1980, ed. D.H. Norrie vol. 1, p. 110–120.
 - [56] O. Pironneau, On the transport diffusion algorithm and its application to the Navier–Stokes equations, Numer. Math. 38 (1982) 309.
 - [57] L. Halpern, Approximations paraxiales et conditions aux limites absorbantes. Thèse d’Etat, Université Pierre et Marie Curie, Paris (1986).
 - [58] M.O. Bristeau and J. Periaux, Finite Element Methods for the calculation of compressible viscous flows using self-adaptive mesh refinements. Lecture Notes in Computational Fluid Dynamics, Von Karman Institute for Fluid Dynamics, Rhode-St-Genèse, Belgium (March 1986).

- [59] R. Glowinski and J. Periaux, Numerical Methods for Nonlinear problems in Fluid Dynamics, Proc. Intern Seminar on Scientific Supercomputers, Paris, Feb. 1987 (North-Holland, Amsterdam) to appear.
- [60] R.M. Beam and R.F. Warming, An implicit factored scheme for the compressible Navier-Stokes equations. AIAA J. 16 (1978) 393.
- [61] R.W. MacCormack, Current status of numerical solutions of the Navier–Stokes equations. AIAA Paper 85-0032.
- [62] J.L. Thomas and R.W. Walters, Upwind relaxation algorithms for the Navier–Stokes Equations, AIAA Paper 85-1501.
- [63] R. Peyret and H. Viviand, Computation of viscous compressible flows based on the Navier–Stokes equations, AGARD-AG-212 (1975).
- [64] S. Bokhari, M.Y. Hussaini, J.J. Lambiotte and S.A. Orzag, A Navier–Stokes solution on the CYBER-203 by a pseudo-spectral technique, ICASE Report no. 81-82, (1981).
- [65] N. Satofuka, Modified differential quadrature method for numerical solution of multi-dimensional flow problems, Proc. Intern. Symp. on Applied Mathematics and Information Science, Kyoto, March 1982, pp. 5.7–5.14.
- [66] R. Lohner, K. Morgan, J. Peraire and O.C. Zienkiewicz, Finite Element Methods for high speed flows. AIAA Paper 85. 1531.
- [67] T.J.R. Hughes, M. Mallet and L.P. Franca, New Finite Element Methods for the Compressible Euler and Navier–Stokes Equations. Proc. Seventh Intern. Conf. on Comp. Meth. in Applied Sciences and Engineering, Versailles. Dec. 1985.
- [68] A. Matsumura and T. Nishida, Initial boundary value problems for the equations of motion in general fluids, in: Computing methods in Applied Sciences and Engineering V, eds. R. Glowinski and J.L. Lions (North-Holland, Amsterdam, 1982) p. 389–406.
- [69] J.P. Benque, G. Labadie and B. Latteux, Une méthode d’éléments finis pour les équations de Saint-Venant, 2nd Intern. Conf. on Numerical Methods in Laminar and Turbulent Flows, Venise, 1981.
- [70] R. Glowinski and O. Pironneau, On a mixed finite element approximation of the Stokes problem. Convergence of the approximate solution, Numerische Mathematik, 33 (1979) 397.
- [71] G.S. Singh, Résolution des équations de Navier–Stokes en éléments finis par des méthodes de décomposition d’opérateurs et des schémas de Runge-Kutta stables. Thèse de 3ème cycle, Paris VI, Juin 1985.
- [72] J. Goussebaile, F. Hecht, G. Labadie and L. Reinhart, Finite Element Solution of the Shallow Water Equations by a Quasi-direct Decompositon Procedure, Intern. J. Numer. Meth. in Fluids, vol. 4 (1984) 1117.
- [73] J. Periaux, Résolution de quelques problèmes non linéaires en aérodynamique par des méthodes d’éléments finis et de moindres carrés. Thèse 3ème cycle, Université Pierre et Marie Curie, June 1979.
- [74] B. Palmerio, V. Billey, A. Dervieux and J. Periaux, Self-adaptive mesh refinements and finite Element Methods for solving the Euler equations, Proc. of the ICFD Conf. on Numerical Methods for Fluid Dynamics, Reading (1985), to be published by Pineridge.
- [75] C. Pouletty, Génération et Optimisation de maillages éléments finis, Application à la résolution de quelques équations en Mécanique des Fluides, Thèse de Docteur-Ingénieur, Ecole Centrale, Paris, Déc. 85.
- [76] R. Lohner, K. Morgan and O.C. Zienkiewicz, Adaptive grid refinement for the compressible Euler equations, Accuracy Estimates and Adaptivity for Finite Elements, Wiley, Chichester (to appear).
- [77] B. Palmerio and A. Dervieux, Application of a FEM moving node adaptive method to accurate shock capturing, Proceedings of the 1st Int. Conf. on Numerical Grid Generation in Comp. Fluid Dynamics, Lanshut, Germany, July 1986, to be published by Pineridge.
- [78] R. Lohner, K. Morgan, J. Peraire, O.C. Zienkiewicz and L. Kong, Finite Element Methods for Compressible Flow, Proc. of the Conf. Numerical Methods for Fluid Dynamics, Reading, Numer. Meth. for Fluid Dyn. II (Clarendon, Oxford, 1986).
- [79] I. Babuska and W.C. Rheinbold, A posteriori error estimates for the Finite Element method. Intern. J. Numer. Meth. Eng. 112 (1978) 1597.
- [80] I. Babuska and W.C. Rheinbold, Error estimates for adaptive finite element computations. SIAM J. Numer. Anal. 15 (1978) 736.
- [81] J.F. Dannenhofer and J.R. Baron, Grid adaptation for the 2D Euler equations. AIAA Papers, 85. 0484, 1985.
- [82] C. Cua, Amélioration de maillages par des méthodes de sous-gradient. Thèse de 3ème Cycle, Université de Grenoble, Nov. 1985.

- [83] R.E. Bank and A.H. Sherman, An adaptive multi-level method for elliptic boundary value problems. Computing 26 (1981) 91.
- [84] E. Perez, J. Periaux, J.P. Rosenblum, B. Stoufflet, A. Dervieux and M.H. Lallemand, Adaptive Full-Multigrid FEM for solving the two-dimensional Euler equations, IC 10NMFD, Pekin, 1986, Lecture Notes in Physics (Springer-Verlag, Berlin).
- [85] F. Angrand and P. Leyland, Une méthode multigrille pour la résolution des équations de Navier-Stokes compressible en deux dimensions par des éléments finis, Rapport de Recherche INRIA, No 593, Dec. 1986.
- [86] D. Mavriplis and A. Jameson, Multigrid Solution of the Two-Dimensional Euler Equations on Unstructured Triangular Meshes, AIAA Paper 87-0353.
- [87] GAMM Workshop, Numerical Simulation of Compressible Navier–Stokes Flows, Nice, Déc. 1985, Notes on Numerical Fluid Mechanics, Vieweg (to appear).
- [88] J. Allegre, J.C. Lengrand and M. Raffin, Champs d'Écoulements raréfiés compressibles autour d'un profil NACA0012, Contrat DRET 80/636, Rapport SESSIA 894/82-380, 1982.
- [89] H. Werle, Décollement instationnaire dans une prise d'air bidimensionnelle avec incidence, Rapport Technique ONERA 62/2149 AN, 1976.