

UNIVERSITAT POLITÈCNICA DE CATALUNYA

**A FINITE ELEMENT MODEL FOR
INCOMPRESSIBLE FLOW PROBLEMS**

by

RAMON CODINA I ROVIRA

DOCTORAL THESIS

Barcelona, June 1992

CHAPTER 4

TRANSIENT NAVIER-STOKES EQUATIONS: FULLY DISCRETE ALGORITHM AND COMPUTATIONAL ASPECTS

4.1 Introduction

The purpose of this chapter is to present an algorithm for the numerical simulation of the Navier-Stokes equations combining the ideas of Chapters 1 and 3, basically, and part of Chapter 2. The emphasis will be mainly computational, giving in Section 4.6 a fully discrete and linearized numerical scheme adapted to the implementation on a computer.

The basic tools of the numerical model are now briefly described. The temporal derivatives are discretized using the generalized trapezoidal rule as described in Chapter 2 for the convection-diffusion equation. The incompressibility constraint is treated by using div-stable velocity-pressure interpolations. The pressure is eliminated through penalization. Several choices are discussed, in particular the iterative penalty method introduced and analyzed in Chapter 3 (using weak penalization) and a particular version of the artificial compressibility method. Since the stabilization of the pressure is left to the finite element interpolation, only the convection of the velocity has to be stabilized when high Reynolds number flows are considered. This is done by means of a Streamline Diffusion (SD) operator added to the Galerkin variational form and properly linearized.

While the mathematical analysis of the finite element method for the convection-diffusion equation and the stationary Navier-Stokes equations using the Galerkin approach is fairly complete, there are still a lot of open questions for the full Navier-Stokes equations. The most extensive analysis of the transient problem we are aware of is that of Heywood & Rannacher [HR1-4]. Error estimates are given for the Galerkin finite element approximation of the Navier-Stokes equations with homogeneous Dirichlet boundary conditions. The time discretization using the Crank-Nicolson scheme is analysed in the last paper of this series [HR4]. Although we are interested in more general situations, the results obtained by these authors will be often referred to in this chapter. Our approach differs from the one they analyse in the use of penalty methods, the SD operator, the boundary conditions and the way the trapezoidal rule is implemented.

Based on the results of the previous chapter, the penalization of the incompressibility constraint is viewed as an iterative procedure to achieve this restriction rather

than a perturbation of the initial problem. Because of this, it is not considered until Section 4.5, where the way the nonlinear system of equations is solved is treated.

Once the description of the numerical algorithm is complete, Section 4.7 presents the methods used to compute nodal pressure values as well as nodal values of the vorticity and the physical properties when they are variable. This will be used in the next chapter, where the numerical simulation of thermally coupled flows and nonlinear materials is studied. In these cases, the density depends on the temperature and the viscosity depends on the invariants of the strain rate tensor and perhaps also on the temperature. For the particular case of two-dimensional flows, an algorithm to compute the streamfunction is presented.

The numerical examples presented in the previous chapters were mainly intended to verify the theory. However, such theoretical grounds are not available for the general problem considered here and the numerical experimentation is of fundamental importance. The most common benchmark tests for the numerical simulation of incompressible flow problems are presented in Section 4.8, namely, the driven cavity flow at (relatively) high Reynolds numbers, the flow over a backward facing step and the flow past a cylinder. The meshes used in the calculations are somehow coarse, if compared with the results presented in the literature, since one of our purposes is to assess the performance of the SD operator when the Galerkin approach yields oscillatory results. Nevertheless, they have to be fine enough to capture the physical details of the flow. In complicated flow situations (the most common in reality) this is the main challenge that computational fluid dynamics has at present.

The literature on finite element methods for incompressible viscous fluids is vast. We again refer to the well-known text books [CO], [CSS] and the more recent texts [Gu], [Pi] for a general presentation of the problem. A mathematically oriented exposition can be found in the books of Temam [Te] and Girault & Raviart [GR1]. For a comprehensive engineering treatment of the problem the reader is referred to the book of Zienkiewicz & Taylor (vol. 2) [ZT].

4.2 The continuous problem

In this chapter we will attempt the numerical solution of the following initial and boundary value problem:

$$\rho[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}] - 2\mu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p = \rho \mathbf{f} \quad \text{in } \Omega \times (0, T) \quad (4.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \times (0, T) \quad (4.2)$$

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_D \times (0, T) \quad (4.3)$$

$$\mathbf{n} \cdot \boldsymbol{\sigma} = \bar{\mathbf{t}} \quad \text{on } \Gamma_N \times (0, T) \quad (4.4)$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}) \quad \text{on } \Omega \times \{0\} \quad (4.5)$$

Besides the notation introduced in Chapter 3, the meaning of the different symbols appearing in (4.1)–(4.5) is the following. The time interval where the problem is to be solved is $(0, T)$, with $T > 0$. The temporal derivative of the velocity has been denoted by $\partial_t \mathbf{u}$, t being the time variable. The overbars in $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ denote prescribed values (boundary conditions). The former is the velocity given on a part Γ_D of $\Gamma := \partial\Omega$ (Dirichlet-type prescription) and the latter is a given surface force vector on $\Gamma_N \subset \Gamma$

(Neumann-type prescription), satisfying $\Gamma = \overline{\Gamma_D \cup \Gamma_N}$, with $\Gamma_D \cap \Gamma_N = \emptyset$. The unit outward normal to Γ has been indicated by \mathbf{n} . The (Eulerian) stress tensor for a generalized Newtonian fluid is

$$\boldsymbol{\sigma} = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon}(\mathbf{u}) := \frac{1}{2} [(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T] \quad (4.6)$$

where \mathbf{I} is the unit tensor. In this chapter, the dynamical viscosity μ and the density ρ will be considered constant (Newtonian behavior).

There are two main reasons for writing the viscous term as $2\mu\nabla \cdot \boldsymbol{\varepsilon}$. The first is that the boundary condition (4.4) on Γ_N enters naturally the variational form of problem (4.1)–(4.5). The second is that in Chapters 5 and 6 we will consider cases with variable viscosity, and the expression $2\nabla \cdot [\mu\boldsymbol{\varepsilon}(\mathbf{u})]$ will be needed (otherwise, the gradient of μ has to be calculated). Moreover, for obtaining the simplified form $\mu\Delta\mathbf{u}$ used in (3.1) the condition $\nabla \cdot \mathbf{u} = 0$ has to be employed. This condition will not hold exactly when using penalty methods and it seems preferable not to use it in deriving the equations.

Equation (4.5) is the initial condition. A generic point in Ω has been denoted by \mathbf{x} . The given vector function $\mathbf{u}_0(\mathbf{x})$ is assumed to be divergence free and to satisfy the Dirichlet boundary conditions.

A mixed type of boundary prescriptions can also be considered. For that, let $\Gamma_M \subset \Gamma$ and let $\mathbf{g}_1, \mathbf{g}_2$ (in 3D) be the local basis for the tangent space to Γ_M . In practice, it is often useful to consider the following conditions:

$$\mathbf{u} \cdot \mathbf{n} = \bar{u}_n, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_1 = \bar{t}_1, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_2 = \bar{t}_2, \quad \text{on } \Gamma_M \quad (4.7)$$

where \bar{u}_n is a given scalar and \bar{t}_1 and \bar{t}_2 are the components of the force tangent to Γ_M in the local basis $\mathbf{g}_1, \mathbf{g}_2$. In the numerical simulation of turbulent flows, it is common to consider $\bar{u}_n = 0$ (impermeable wall condition) and to express \bar{t}_1 and \bar{t}_2 in terms of the velocity tangent to Γ_M , trying to emulate the frictional effects of turbulent boundary layers. In Chapter 6, a special type of friction law will be introduced. For the moment, and to simplify the exposition, boundary conditions of type (4.7) will not be considered. For the stationary problem and using the Galerkin approach, they have been studied by Verfürth [Ve].

In order to write the weak form of problem (4.1)–(4.5) we need to introduce some functional spaces. The test function spaces for the velocity and the pressure, V_t and Q_t , will be

$$\begin{aligned} V_t &= \{\mathbf{v} \in H^1(\Omega)^{N_{sd}} \mid \mathbf{v}|_{\Gamma_D} = \mathbf{0}\} \\ Q_t &= L^2(\Omega) \end{aligned} \quad (4.8)$$

The spaces of trial solutions will consist of time dependent functions. At least when $\Gamma_N = \emptyset$, it can be shown [Te] that the minimum regularity in time that has to be required is square-integrability. Thus, let us introduce

$$V_s = \{\mathbf{v} \in L^2(0, T; H^1(\Omega)^{N_{sd}}) \mid \mathbf{v}|_{\Gamma_D} = \bar{\mathbf{u}}, \quad t \in (0, T)\} \quad (4.9)$$

$$Q_s = \{q \in L^2(0, T; L^2(\Omega)) \mid \int_{\Omega} q d\Omega = 0, \quad t \in (0, T), \quad \text{if } \Gamma_N = \emptyset\} \quad (4.10)$$

as spaces of trial solutions for the velocity and the pressure. For the latter case, it has to be remarked that when $\Gamma_N \neq \emptyset$ the pressure is *not* underdetermined by a constant, since the boundary condition (4.4) involves the pressure itself, not its derivatives:

$$\mathbf{n} \cdot \boldsymbol{\sigma} = -pn + 2\mu\mathbf{n} \cdot \boldsymbol{\varepsilon} = \bar{\mathbf{t}} \quad (4.11)$$

The data \mathbf{f} , $\bar{\mathbf{u}}$, $\bar{\mathbf{t}}$ and \mathbf{u}_0 is assumed to satisfy the following conditions:

$$\begin{aligned} \mathbf{f} &\in L^2(0, T; L^2(\Omega)^{N_{sd}}) \\ \bar{\mathbf{u}} &\in L^2(0, T; H^{1/2}(\Gamma)^{N_{sd}}) \\ \bar{\mathbf{t}} &\in L^2(0, T; H^{-1/2}(\Gamma)^{N_{sd}}) \\ \mathbf{u}_0 &\in \{\mathbf{v} \in L^2(\Omega)^{N_{sd}} \mid \nabla \cdot \mathbf{v} = 0\} \end{aligned} \quad (4.12)$$

Finally, as for the stationary problem we define:

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= 2\mu \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega, \\ b(q, \mathbf{v}) &= \int_{\Omega} q \nabla \cdot \mathbf{v} d\Omega, \\ c(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \rho \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{v}] \cdot \mathbf{w} d\Omega + \frac{1}{2} \rho \int_{\Omega} (\nabla \cdot \mathbf{u}) \mathbf{v} \cdot \mathbf{w} d\Omega, \\ l(\mathbf{v}) &= \rho \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_N} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma, \end{aligned} \quad (4.13)$$

all these functions taking values in $L^2(0, T)$. The choice of the convective term c has been already discussed in Chapter 3.

The weak form of problem (4.1)–(4.5) reads now as follows: Find $\mathbf{u} \in V_s$ and $p \in Q_s$ such that

$$\begin{aligned} \rho(\partial_t \mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) &= l(\mathbf{v}) & \forall \mathbf{v} \in V_t \\ b(q, \mathbf{u}) &= 0 & \forall q \in Q_t \\ (\mathbf{u}(\mathbf{x}, 0), \mathbf{v}) &= (\mathbf{u}_0(\mathbf{x}), \mathbf{v}) & \forall \mathbf{v} \in V_t \end{aligned} \quad (4.14)$$

For the case $\Gamma_N = \emptyset$, it is proved in [Te] that all the terms in (4.14) make sense. Moreover, the regularity of the solution is higher than *a priori* required. If $N_{sd} = 2$ a unique solution to problem (4.14) exists. One of the most important open questions in the mathematical analysis of the Navier-Stokes problem is the existence and uniqueness for $N_{sd} = 3$. Existence is a well known result (weak solutions), but uniqueness can only be proved in spaces of functions more regular than (4.9)–(4.10) (classical solutions), in which case only local existence can be proved, i.e., for sufficiently small T (see, e.g., [CF], [La], [Li], [Te]).

The coercivity condition (3.9) and the BB condition (3.10) are also needed for the transient equations, both for the continuous and the discrete problems. Condition (3.12) is not assumed.

4.3 Discretization in time

Consider a nonlinear system of ordinary differential equations of the form

$$\dot{\mathbf{x}} = F(\mathbf{x}, t) \quad (4.15)$$

where $\mathbf{x} = \mathbf{x}(t)$ is a vector function. Let us define, for $\theta \in [0, 1]$,

$$\begin{aligned} \mathbf{x}_\theta^n &:= \theta \mathbf{x}^n + (1 - \theta) \mathbf{x}^{n-1}, & t_\theta^n &:= \theta t^n + (1 - \theta) t^{n-1} \\ F^n(\mathbf{x}, t) &:= F(\mathbf{x}^n, t^n) \end{aligned}$$

where $t^n := n\Delta t$, \mathbf{x}^n is an approximation to $\mathbf{x}(t^n)$ and Δt is the time step size of a uniform partition of $[0, T]$. The generalized trapezoidal rule used in Chapter 2 for the linear convection-diffusion equation can be extended in two different ways to *nonlinear* problems. These two forms are:

$$\frac{1}{\Delta t}(\mathbf{x}^n - \mathbf{x}^{n-1}) = \theta F^n(\mathbf{x}, t) + (1 - \theta) F^{n-1}(\mathbf{x}, t) \quad (4.16)$$

$$\frac{1}{\Delta t}(\mathbf{x}^n - \mathbf{x}^{n-1}) = F(\mathbf{x}_\theta^n, t_\theta^n) \quad (4.17)$$

The first choice has a clear geometrical interpretation: the time derivative in the interval (t^{n-1}, t^n) has been approximated by a combination of the derivatives at t^{n-1} and t^n . The interpretation of the second method is not so clear. If $\mathbf{x}(t)$ were linear, then $\mathbf{x}_\theta^n = \mathbf{x}(t_\theta^n)$ and (4.17) would mean that the time derivative has been calculated at a point within the interval (t^{n-1}, t^n) .

Here we will discuss the implementation of both approaches for the Navier-Stokes problem. In the literature, the most common approach is (4.17) [Gu], [HR4], although (4.16) is also used [CSS].

Consider first (4.16). When it is applied to the strong form of the Navier-Stokes equations (4.1)–(4.2) one has to find $\mathbf{u}^n(\mathbf{x})$ and $p^n(\mathbf{x})$, approximations to $\mathbf{u}(\mathbf{x}, t^n)$ and $p(\mathbf{x}, t^n)$, such that

$$\begin{aligned} \rho[(\mathbf{u}^n - \mathbf{u}^{n-1})/\Delta t + \theta(\mathbf{u}^n \cdot \nabla)\mathbf{u}^n + (1 - \theta)(\mathbf{u}^{n-1} \cdot \nabla)\mathbf{u}^{n-1}] \\ - 2\mu\theta\nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}^n) - 2\mu(1 - \theta)\nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}^{n-1}) \\ + \theta\nabla p^n + (1 - \theta)\nabla p^{n-1} = \theta\rho\mathbf{f}^n + (1 - \theta)\rho\mathbf{f}^{n-1} \\ \nabla \cdot \mathbf{u}^n = 0 \end{aligned} \quad (4.18)$$

for $n = 1, 2, \dots$, with $\mathbf{u}^0(\mathbf{x}) = \mathbf{u}_0(\mathbf{x})$. The initial pressure $p^0(\mathbf{x})$ will be the solution of the boundary value problem

$$\begin{aligned} \Delta p^0 &= \nabla \cdot [\rho\mathbf{f}^0 - \rho(\mathbf{u}^0 \cdot \nabla)\mathbf{u}^0] && \text{in } \Omega \\ \frac{\partial p^0}{\partial n} &= \mathbf{n} \cdot [\rho\mathbf{f}^0 - \rho(\mathbf{u}^0 \cdot \nabla)\mathbf{u}^0 - \rho\partial_t \bar{\mathbf{u}} + 2\mu\nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}^0)] && \text{on } \Gamma_D \\ p^0 &= 2\mu\mathbf{n} \cdot \boldsymbol{\varepsilon}(\mathbf{u}^0) \cdot \mathbf{n} - \bar{\mathbf{t}} \cdot \mathbf{n} && \text{on } \Gamma_N \end{aligned} \quad (4.19)$$

In [HR1] it is proved for the case $\Gamma_N = \emptyset$ that a unique solution (modulo constants) exists for this problem.

The velocity \mathbf{u}^n and the pressure p^n solution of problem (4.18) have to satisfy the boundary conditions (4.3) and (4.4).

The weak form of problem (4.18) will be:

$$\begin{aligned} \rho \frac{1}{\Delta t} (\mathbf{u}^n - \mathbf{u}^{n-1}, \mathbf{v}) + \theta c(\mathbf{u}^n, \mathbf{u}^n, \mathbf{v}) + (1 - \theta) c(\mathbf{u}^{n-1}, \mathbf{u}^{n-1}, \mathbf{v}) \\ + \theta a(\mathbf{u}^n, \mathbf{v}) + (1 - \theta) a(\mathbf{u}^{n-1}, \mathbf{v}) \\ - \theta b(p^n, \mathbf{v}) - (1 - \theta) b(p^{n-1}, \mathbf{v}) = \theta l^n(\mathbf{v}) + (1 - \theta) l^{n-1}(\mathbf{v}) \\ b(q, \mathbf{u}^n) = 0 \end{aligned} \quad (4.20)$$

for all $\mathbf{v} \in V_t$ and $q \in Q_t$. Observe that l is a linear function and hence $l^n(\mathbf{v})$ is the value of $l(\mathbf{v})$ evaluated with \mathbf{f}^n and $\bar{\mathbf{t}}^n$.

It is easy to see that (4.20) is also obtained from the time discretization of the continuous variational form (4.14). Symbolically, we have the following commutative diagram:

$$\begin{array}{ccc} \text{Eqns. (4.1) - (4.5)} & \xrightarrow{\text{Time discretization}} & \text{Eqn. (4.18)} \\ \downarrow \text{Weak form} & & \downarrow \text{Weak form} \\ \text{Eqn. (4.14)} & \xrightarrow{\text{Time discretization}} & \text{Eqn. (4.20)} \end{array}$$

This remark might seem obvious in this case. However, the definition of the SD operator will depend on the order of the space and time discretizations.

Let us consider now (4.17) applied to the time discretization of the weak form of the continuous problem (4.14). Instead of (4.20) we will find:

$$\begin{aligned} \rho \frac{1}{\Delta t} (\mathbf{u}^n - \mathbf{u}^{n-1}, \mathbf{v}) + c(\mathbf{u}_\theta^n, \mathbf{u}_\theta^n, \mathbf{v}) + a(\mathbf{u}_\theta^n, \mathbf{v}) - b(p_\theta^n, \mathbf{v}) = l_\theta^n(\mathbf{v}) \\ b(q, \mathbf{u}_\theta^n) = 0 \end{aligned} \quad (4.21)$$

where $\mathbf{u}_\theta^n := \theta \mathbf{u}^n + (1 - \theta) \mathbf{u}^{n-1}$, $p_\theta^n := \theta p^n + (1 - \theta) p^{n-1}$ and $l_\theta^n(\mathbf{v})$ is calculated with $\mathbf{f}_\theta^n := \theta \mathbf{f}^n + (1 - \theta) \mathbf{f}^{n-1}$ and $\bar{\mathbf{t}}_\theta^n := \theta \bar{\mathbf{t}}^n + (1 - \theta) \bar{\mathbf{t}}^{n-1}$. Since c , a , b and l are linear in each argument, the only difference between (4.20) and (4.21) will be the convective term. For (4.21) we will have that

$$\begin{aligned} c(\mathbf{u}_\theta^n, \mathbf{u}_\theta^n, \mathbf{v}) &= c(\theta \mathbf{u}^n + (1 - \theta) \mathbf{u}^{n-1}, \theta \mathbf{u}^n + (1 - \theta) \mathbf{u}^{n-1}, \mathbf{v}) \\ &= \theta^2 c(\mathbf{u}^n, \mathbf{u}^n, \mathbf{v}) + \theta(1 - \theta) c(\mathbf{u}^n, \mathbf{u}^{n-1}, \mathbf{v}) \\ &\quad + \theta(1 - \theta) c(\mathbf{u}^{n-1}, \mathbf{u}^n, \mathbf{v}) + (1 - \theta)^2 c(\mathbf{u}^{n-1}, \mathbf{u}^{n-1}, \mathbf{v}) \end{aligned}$$

If we denote by $c_\theta^n(\mathbf{u}, \mathbf{u}, \mathbf{v}) := \theta c(\mathbf{u}^n, \mathbf{u}^n, \mathbf{v}) + (1 - \theta) c(\mathbf{u}^{n-1}, \mathbf{u}^{n-1}, \mathbf{v})$ the convective term in (4.20), it is easy to show that

$$c(\mathbf{u}_\theta^n, \mathbf{u}_\theta^n, \mathbf{v}) - c_\theta^n(\mathbf{u}, \mathbf{u}, \mathbf{v}) = \theta(\theta - 1) c(\mathbf{u}^n - \mathbf{u}^{n-1}, \mathbf{u}^n - \mathbf{u}^{n-1}, \mathbf{v})$$

and therefore the difference between (4.20) and (4.21) will be a term of order $O(\Delta t^2)$ (in the $H^1(\Omega)^{N,d}$ -norm). Since this is the best consistency error we can hope for (using $\theta = 1/2$, i.e., the Crank-Nicolson scheme), the accuracy will not be affected if (4.20) or (4.21) are used.

Without any further information about the difference between (4.20) and (4.21) concerning their convergence properties (nonlinear stability), the decision for choosing one scheme or another will be based on computational criteria. First observe that

$$\mathbf{u}^n - \mathbf{u}^{n-1} = \frac{1}{\theta} \mathbf{u}_\theta^n - \frac{1}{\theta} \mathbf{u}^{n-1}$$

and thus (4.21) may be rewritten as

$$\rho \frac{1}{\theta \Delta t} (\mathbf{u}_\theta^n, \mathbf{v}) + c(\mathbf{u}_\theta^n, \mathbf{u}_\theta^n, \mathbf{v}) + a(\mathbf{u}_\theta^n, \mathbf{v}) - b(p_\theta^n, \mathbf{v}) = I_\theta^n(\mathbf{v}) + \rho \frac{1}{\theta \Delta t} (\mathbf{u}^{n-1}, \mathbf{v}) \quad (4.22)$$

$$b(q, \mathbf{u}_\theta^n) = 0$$

This expression involves only the unknown \mathbf{u}_θ^n . The computational effort of (4.20) is higher than that of (4.22), since there are more right-hand-side terms to calculate per time step. These additional terms are

$$-(1 - \theta)c(\mathbf{u}^{n-1}, \mathbf{u}^{n-1}, \mathbf{v}) - (1 - \theta)a(\mathbf{u}^{n-1}, \mathbf{v}) + (1 - \theta)b(p^{n-1}, \mathbf{v})$$

In spite of this higher computational effort, we have chosen (4.20) and not (4.22) for the numerical implementation. There are several reasons for this. The first is the definition of the SD operator to be introduced later. It is conceptually simpler if the unknown is the velocity \mathbf{u}^n and not an intermediate value \mathbf{u}_θ^n between \mathbf{u}^{n-1} and \mathbf{u}^n . Also, the penalty methods we will discuss will be based on the fact that the pressure p^n , and not p_θ^n , has to be calculated. The most important reason, however, is the following. Both (4.20) and (4.22) are nonlinear problems and have to be solved iteratively. In Chapters 5 and 6 we will attempt the solution of thermally coupled flows of (possibly) nonlinear materials and perhaps with a free surface. Both for the constitutive laws and for the tracking of the free surfaces the velocity \mathbf{u}^n is needed. Since the iterative procedure due to this new nonlinearity will be coupled to that of the Navier-Stokes equations, the use of (4.22) would require the calculation of \mathbf{u}^n from \mathbf{u}_θ^n and \mathbf{u}^{n-1} for each iteration. We would have to deal with \mathbf{u}^{n-1} , \mathbf{u}_θ^n and \mathbf{u}^n and this means either more computer memory (if \mathbf{u}^n is stored) or more calculations (if \mathbf{u}^n is computed when needed). For all these reasons, scheme (4.20) will be used in what follows.

To conclude this section, let us discuss the choice of the parameter θ . The only interesting cases are $\theta = 0$ (forward Euler), $\theta = 1/2$ (Crank-Nicolson) and $\theta = 1$ (backward Euler). The first value yields a conditionally stable scheme and the other two values an unconditionally stable algorithm [Te]. However, due to the implicit nature of the pressure, the case $\theta = 0$ is unconditionally unstable using the $\mathbf{u}-p$ formulation. If the incompressibility constraint is penalized, ϵ being the penalty parameter, the critical time step Δt_c will behave as ϵ when $\epsilon \rightarrow 0$ (incompressible limit). To see this, one can argue as follows. Once the pressure is eliminated from the penalized incompressibility condition and it is substituted in the momentum equations, the effective viscosity that will multiply some second derivatives of the velocity will be $\mu + 1/\epsilon$ (see Eqn. (3.22)). Since Δt_c will be proportional to the inverse of this viscosity (see Chapter 2), $\Delta t_c \sim \epsilon$ for $\epsilon \rightarrow 0$.

The Crank-Nicolson algorithm will be useful when the accuracy in time be fundamental, since a second order approximation can be expected [HR4]. However, if the transient evolution is not very important, $\theta = 1$ should be preferred, since the resulting scheme is computationally cheaper (several terms in (4.20) vanish). Also, either if $\theta = 1$ or $\theta = 1/2$ is to be employed, the former value is recommended for the first few time

steps (usually one or two). This was also valid for the transient convection-diffusion equation discussed in Chapter 2. The explanation we gave there was the difficulty in reproducing the rapidly oscillating harmonics associated to the series expansion of the solution of parabolic equations. Now there are two more reasons. The first is that if $\theta \neq 1$, the use of (4.20) necessitates the initial pressure p^0 for $n = 1$ and hence problem (4.19) has to be solved. The other reason is the singularity for $t \rightarrow 0$ that will be discussed later.

4.4 Space discretization and Streamline Diffusion operator

The particular version of the SD method we will consider will be based on the idea of stabilizing the convective term of the Navier-Stokes equations, with the same motivation as in Chapter 1 for the convection-diffusion problem. The role of stabilizing the pressure is assigned to the finite element interpolation, that is, the velocity-pressure spaces will have to be div-stable. It is important to emphasize this fact because this is the main difference between the formulation presented here and the least-squares techniques, to which much attention is currently being paid in the literature.

4.4.1 Galerkin approach and finite element spaces

The semidiscrete problem

The finite element approximation we will consider is *conforming*, both for the Galerkin approach and adding the SD operator [Hu1], that is, the discrete spaces of test functions and of trial solutions will be linear subspaces of the corresponding spaces for the continuous problem. We will denote them by $V_{h,t} \subset V_t$ and $V_{h,s} \subset V_s$ for the velocity and $Q_{h,t} \subset Q_t$ and $Q_{h,s} \subset Q_s$ for the pressure. They will be constructed from a finite element partition $\{\Omega^e\}$, $e = 1, \dots, N_{el}$, of the spatial domain Ω .

The Galerkin semidiscrete problem consists in seeking $\mathbf{u}_h \in V_{h,s}$ and $p_h \in Q_{h,s}$ such that

$$\begin{aligned} \rho(\partial_t \mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + a(\mathbf{u}_h, \mathbf{v}_h) - b(p_h, \mathbf{v}_h) &= l(\mathbf{v}_h) & \forall \mathbf{v}_h \in V_{h,t} \\ b(q_h, \mathbf{u}_h) &= 0 & \forall q_h \in Q_{h,t} \\ (\mathbf{u}_h(\mathbf{x}, 0), \mathbf{v}_h) &= (\mathbf{u}_0(\mathbf{x}), \mathbf{v}_h) & \forall \mathbf{v}_h \in V_{h,t} \end{aligned} \quad (4.23)$$






This problem is nothing but the space-discretized version of the continuous variational equations (4.14).

Finite element spaces

As in Chapter 3, we will use penalty methods. It is therefore desirable to employ a discontinuous pressure interpolation, since this allows to eliminate the pressure nodal unknowns at the element level as already explained. Moreover, the velocity-pressure pairs will have to satisfy the Babuška-Brezzi stability condition (div-stability).

Some of the elements we have implemented in the computer code with which the problems of this and the following two chapters have been solved are collected in

Box 4.1. There, N_{nv} is the number of nodes of each element with velocity unknowns (standard C^0 interpolation) and N_{qp} the number of pressure nodes within each element (C^{-1} interpolation). Concerning the schematic for the 2D case, nodes with velocity unknowns have been represented by a circle and nodes with pressure unknowns by a triangle.

Box 4.1 Some finite elements with discontinuous pressure			
Element	$N_{nv}-N_{qp}$ (2D/3D)	Description	Schematic (2D)
Q_1/P_0	4/8 - 1/1	-Continuous bi- or tri-linear velocity. Piecewise constant pressure.	
Q_2^-/P_0	8/20 - 1/1	-Serendipid velocity interpolation. Piecewise constant pressure.	
Q_2/P_1	9/27 - 3/4	-Continuous bi- or tri-quadratic velocity. Piecewise linear pressure.	
P_2/P_0	6/10 - 1/1	-Continuous quadratic velocity. Piecewise constant pressure.	
P_2^+/P_1	7/15 - 3/4	-Continuous quadratic velocity enriched with bubble functions. Piecewise linear pressure.	

Let us comment now some properties of the elements in Box 4.1 concerning their convergence for stationary flows. It will be discussed thereafter what happens for the transient Navier-Stokes equations.

• *Element Q_1/P_0*

This is the bilinear (in 2D) or trilinear (in 3D) velocity-constant pressure element already discussed in Chapter 3. It does not satisfy the BB condition, although there are ways to stabilize it, as it has already been explained in the previous chapter. There is a simple way to see that it may work without any particular stabilization procedure. For simplicity, consider the two-dimensional case. Figure 4.1 shows how a quadratic triangular element enriched with a node placed at the barycenter of the triangle can be splitted into three bilinear elements. If we consider a pressure unknown for each quadrilateral, we see that the velocity and pressure spaces will be isomorphic to those of the P_2^+/P_1 element discussed thereafter. The velocity-pressure interpolation for this element satisfies the BB condition. Therefore, the macroelement depicted in Figure 4.1 composed of Q_1/P_0 elements *will also be div-stable*.

Clearly, the main problem with this approach is the distortion of the triangular patch of three quadrilaterals. This patch has to be regular enough (i.e., the angles sufficiently close to $\pi/3$) to ensure that the isoparametric mapping to the parent domain (usually $[-1, 1] \times [-1, 1]$) be invertible. See Reference [Ci] for the regularity conditions that a finite element partition has to satisfy.

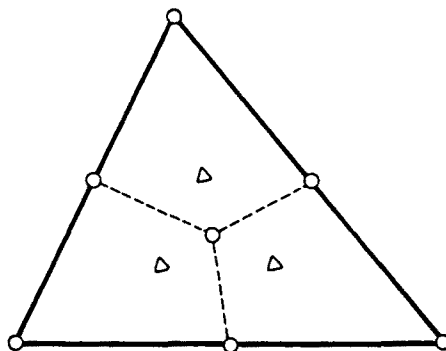


Figure 4.1 A div-stable macroelement composed of Q_1/P_0 elements.

The macroelement of Figure 4.1 is homeomorphic to the macroelement of Le Tallec & Ruas [TR].

In the three-dimensional case, the P_2^+/P_1 element has to be splitted into four Q_1/P_0 subelements. Apparently, this connexion between the P_2^+/P_1 and the Q_1/P_0 elements has never been exploited.

Concerning the convergence properties of the Q_1/P_0 pair, the best we can expect is an error estimate of the form

$$\|\mathbf{u} - \mathbf{u}_h\|_k \leq Ch^{2-k}, \quad \|p - p_h\|_0 \leq Ch \quad (4.24)$$

for $k = 0, 1$ in stationary problems, since this is the interpolation error. In Reference [TR] it is proved that this is true for the Stokes problem using the macroelement introduced in this paper. In (4.24) and below, \mathbf{u} , p denotes the solution of the continuous problem and \mathbf{u}_h , p_h the solution of the problem discretized in space. Also, it is understood that the L^2 estimate for the pressure holds modulo constants if $\Gamma_N = \emptyset$.

- Element Q_2^-/P_0

This and the following elements are div-stable (see, e.g., Reference [GR3] for the proofs). The velocity interpolation uses the serendipid shape functions [Hu2], [ZT] and there is a single pressure unknown for each element. Its convergence will be driven by the pressure interpolation. Again, only an estimate of the form (4.24) can be expected.

Although several researchers actually favor the use of this element (cf. [Hu2]) because of its 'robustness', we have found from numerical experiments that it usually yields overdifusive results, similar to those of the Q_2/P_0 pair emulated via reduced integration of the volumetric term (RIP method) presented in Chapter 3. The slightly higher computational effort needed for the Q_2/P_1 element is certainly worth affording.

- Element Q_2/P_1

For this element, both the velocity and the pressure converge at an optimal rate, i.e.,

$$\|\mathbf{u} - \mathbf{u}_h\|_k \leq Ch^{3-k}, \quad \|p - p_h\|_0 \leq Ch^2 \quad (4.25)$$

for $k = 0, 1$ (see [GR3]).

For the engineering applications, it is not only important to know that the asymptotic estimates (4.25) are optimal, but also to know how accurate the element is for

a given mesh diameter h (loosely speaking, this means how large the constants in (4.25) are). This knowledge is only acquired by numerical experiments. We have found the Q_2/P_1 pair an excellent choice for viscous incompressible flow calculations, in accordance with the results reported in the literature. This element combines several interesting features: it is quadratic in velocities, it is a quadrilateral and pressures are discontinuous. Experience shows that quadratic elements in velocities are an equilibrated compromise between accuracy and complexity (and hence, cost) [Gu]. Moreover, one can hardly expect more regularity for the continuous solution u and p than the one needed for obtaining (4.25), that is $u \in H^3(\Omega)^{N_{sd}}$, $p \in H^2(\Omega)$ for $t \in (0, T)$. On the other hand, quadrilateral elements are known to be more accurate, for a fixed h , than triangular ones, especially for structured meshes. Finally, elements with discontinuous pressures are superior to continuous pressure elements in capturing the details of the flow, especially in recirculation zones and boundary layers. A vast amount of numerical experiments support these facts.

Concerning the implementation of piecewise linear pressures, two options are possible. If $s = (s_1, s_2, s_3)$ (in 3D) are the coordinates of the parent domain of the elements, the first choice is to place $N_{sd} + 1 = 4$ nodes within the elements, with coordinates s_j , $j = 1, 2, 3, 4$, and construct shape functions $N_i(s)$, $i = 1, 2, 3, 4$, such that $N_i(s_j) = \delta_{ij}$ (the Kronecker symbol) for $i, j = 1, 2, 3, 4$. Then, if the pressure is interpolated as $p(s) = \sum_{i=1}^4 N_i(s)p_i$, the coefficients p_i , $i = 1, 2, 3, 4$, have the meaning of being the nodal values of the pressure. A simpler option is to interpolate p as $p(s) = p_0 + s_1 p_1 + s_2 p_2 + s_3 p_3$. Now, p_0 is the value of the pressure for $s_1 = s_2 = s_3 = 0$ and p_1, p_2 and p_3 are its first derivatives. In our computations, we have found no difference in the numerical results using both approaches.

The Q_2/P_1 element for the 3D case is represented in Figure 4.2. The pressure nodes are located at the vertices of a tetrahedron placed in the interior of the brick with the velocity nodes.

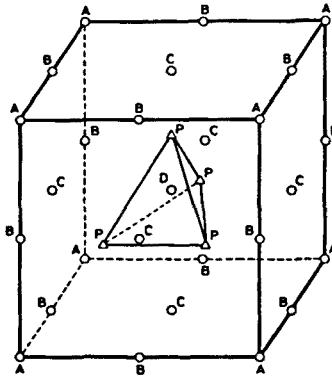


Figure 4.2 Three-dimensional Q_2/P_1 element.

- *Element P_2/P_0*

This element suffers from the same problems that the Q_2^-/P_0 element, perhaps to a lesser extent. Although the velocity is quadratic, the fact that the pressure is piecewise constant controls the error of the approximation. Only the estimate (4.24) can be obtained.

• Element P_2^+/P_1

This element is sometimes referred to as the Crouzieux-Raviart pair [CR]. The triangular quadratic element in velocities is enriched with bubble functions. For the 2D case, a single node is added at the barycenter of the element, whereas for the 3D case nodes have to be added also on the faces of the tetrahedron. The pressure is piecewise linear and discontinuous. The same remarks as for the Q_2/P_1 element regarding the implementation of the pressure interpolation apply. The 3D P_2^+/P_1 element is shown in Figure 4.3.

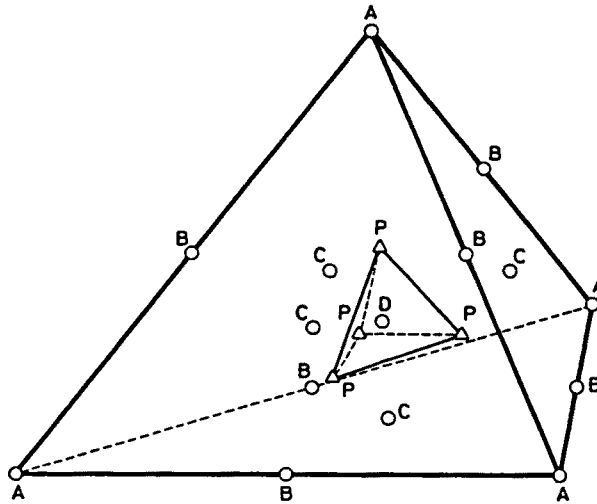


Figure 4.3 Three-dimensional P_2^+/P_1 element.

This element also converges at an optimal rate, i.e., estimates (4.25) hold true.

Although we prefer the Q_2/P_1 element for simple geometries, triangular elements have the important attribute that automatic mesh generation and thus adaptivity are easier to implement using triangles, since they are well suited for designing unstructured meshes. Therefore, the P_2^+/P_1 pair should be considered as a good alternative to the Q_2/P_1 element in complicated geometries or when adaptive procedures have to be used to obtain an error below a prescribed threshold in the computation.

For two-dimensional problems, there is a heuristic index that gives an idea of the accuracy of the element and of how can it reproduce the incompressibility constraint. It is the so called *constraint ratio*, that is the number of velocity unknowns over the number of pressure unknowns in the asymptotic limit $h \rightarrow 0$ [Hu2]. For the P_2^+/P_1 element it is 2, thus reproducing what happens in the continuous problem. In this sense, 2 is the optimal value. For the Q_2/P_1 it is $8/3$, showing that this element is somehow underconstraint.

Fully discrete problem

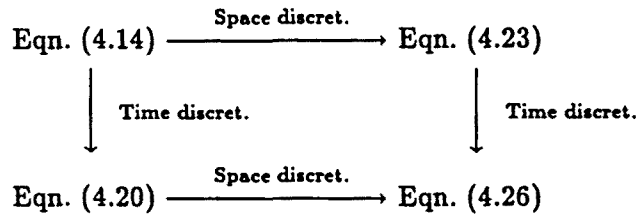
When the generalized trapezoidal rule is applied to problem (4.23) one is led to the following algorithm:

For $n = 1, 2, \dots, N$, given $\mathbf{u}_h^{n-1}(\mathbf{x})$ and $p_h^{n-1}(\mathbf{x})$ find $\mathbf{u}_h^n(\mathbf{x})$ and $p_h^n(\mathbf{x})$ such that

$$\begin{aligned} \rho \frac{1}{\Delta t} (\mathbf{u}_h^n - \mathbf{u}_h^{n-1}, \mathbf{v}_h) + \theta c(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta) c(\mathbf{u}_h^{n-1}, \mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ + \theta a(\mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta) a(\mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ - \theta b(p_h^n, \mathbf{v}_h) - (1 - \theta) b(p_h^{n-1}, \mathbf{v}_h) = \theta l^n(\mathbf{v}_h) + (1 - \theta) l^{n-1}(\mathbf{v}_h) \end{aligned} \quad (4.26)$$

$$b(q_h, \mathbf{u}_h^n) = 0$$

where N is the number of time steps of size Δt in which the interval $[0, T]$ has been divided. Clearly, we could also have started from (4.20) and discretize this problem in space using the Galerkin method. The following diagram shows how the different problems are related:



This is a commutative diagram.

Convergence results

Our purpose now is to quote some of the results obtained by Heywood & Rannacher [HR1-4] for the semidiscrete problem (4.23) and for the fully discrete problem (4.26) in the case $\Gamma_N = \emptyset$, i.e., when the velocity is prescribed on the whole boundary Γ (see also [BR] for a similar analysis). For (4.26), $\theta = 1/2$ is considered. In fact, in the above quoted references the trapezoidal rule is implemented using the discrete version of (4.22). For the reasons explained earlier, we believe that these results will also hold for (4.26).

Consider first the semidiscrete problem (4.23) and assume that the finite element spaces satisfy the following interpolation properties:

$$\|(\mathbf{u} - \tilde{\mathbf{u}}_h)(t)\|_0 \leq Ch^m, \quad \|(p - \tilde{p}_h)(t)\|_0 \leq Ch^{m-1} \quad (4.27)$$

where $\tilde{\mathbf{u}}_h$ and \tilde{p}_h are the finite element interpolants for the velocity and the pressure, respectively, $m \in \{2, 3, 4, 5\}$ and C is a constant independent of t . Then, for the solution $\mathbf{u}_h(\mathbf{x}, t)$, $p_h(\mathbf{x}, t)$ of problem (4.23) the following error bounds can be proved [HR2]:

$$\|(\mathbf{u} - \mathbf{u}_h)(t)\|_0 \leq E_1(t)h^m, \quad \|(p - p_h)(t)\|_0 \leq E_2(t)h^{m-1} \quad (4.28)$$

where the error constants $E_1(t)$ and $E_2(t)$ become singular when $t \rightarrow 0^+$. If $\mathbf{u}_0(\mathbf{x}) = \tilde{\mathbf{u}}(\mathbf{x}, 0)$ on Γ and $\nabla \cdot \mathbf{u}_0 = 0$, it can be shown that

$$\|\mathbf{u}(t)\|_m + \|\partial_t \mathbf{u}(t)\|_{m-2} \leq Kt^{1-m/2}, \quad \text{as } t \rightarrow 0^+ \quad (4.29)$$

where K is a constant independent of t . A similar 'smoothing estimate' is found for parabolic equations [LR]. Since the function $E_1(t)$ involves $\|\mathbf{u}\|_m$, it will behave as

$t^{1-m/2}$ for $t \rightarrow 0^+$. A similar bound can be proved for $E_2(t)$. Roughly speaking, $E_1(t)$ and $E_2(t)$ will behave as follows:

$$E_1(t) \sim t^{1-m/2}, \quad E_2(t) \sim t^{1/2-m/2}, \quad \text{as } t \rightarrow 0^+ \quad (4.30)$$

For the particular case $m = 3$, the function $E_1(t)$ can be bounded above by a constant independent of t if the data \mathbf{f} , $\mathbf{u}_0(\mathbf{x})$ and $\bar{\mathbf{u}}(\mathbf{x}, t)$ satisfy the following compatibility condition:

$$\begin{aligned} &\text{There exists } p_0 \in H^1(\Omega)/\mathbb{R} \text{ such that} \\ \Delta p_0 &= \nabla \cdot [\mathbf{f} - (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0] && \text{in } \Omega \\ \nabla p_0 &= \rho[\mathbf{f} - (\mathbf{u}_0 \cdot \nabla) \mathbf{u}_0 - \partial_t \bar{\mathbf{u}}] + \mu \Delta \mathbf{u}_0 && \text{on } \Gamma \end{aligned} \quad (4.31)$$

This is an overdetermined Neumann problem, with boundary conditions on ∇p_0 and not only on $\partial p_0 / \partial n$. If (4.31) holds true, then $\|\mathbf{u}(t)\|_3 < \infty$ and $\|\partial_t \mathbf{u}(t)\|_1 < \infty$ for all $t \in (0, T)$ and therefore $E_1(t) \leq E \forall t \in (0, T)$, with $E < \infty$ independent of t .

The main conclusion of these results for someone interested in computational aspects is that for t small it is not possible in general to achieve the accuracy that the finite element interpolation might provide. For $m = 3$, i.e., when elements quadratic in velocities are used, both $E_1(t) \rightarrow \infty$ and $E_2(t) \rightarrow \infty$ as $t \rightarrow 0^+$. The best one can hope for is one degree of convergence less, i.e., $m = 2$. In this case, $E_1(t)$ remains bounded for $t \rightarrow 0^+$, but still $E_2(t) \rightarrow \infty$. These facts give another convincing argument for choosing a dissipative time stepping algorithm for the first few time steps. Using the generalized trapezoidal rule, $\theta = 1$ is a wise option.

Estimates (4.28) are *local*, i.e., they hold for sufficiently small t . Similarly to what happens for the continuous problem, $E_1(t)$ and $E_2(t)$ grow exponentially in time if $N_{sd} = 3$. Nevertheless, if the solution of the continuous problem is stable, one can hope that upper bounds for $E_1(t)$ and $E_2(t)$ exist as $t \rightarrow \infty$. Results concerning these facts are proved in [HR2] and [HR3].

Let us go now to the fully discrete problem (4.26). In Reference [HR4] the following estimates are derived for the Crank-Nicolson scheme:

$$\|\mathbf{u}_h^n - \mathbf{u}_h(t^n)\|_0 \leq F_1(t^n) \Delta t^2, \quad \|p_h^n - p_h(t^n)\|_0 \leq F_2(t^n) \Delta t \quad (4.32)$$

In general situations, the functions $F_1(t)$ and $F_2(t)$ behave as follows:

$$F_1(t) \sim t^{-1}, \quad F_2(t) \sim t^{-3/2}, \quad \text{as } t \rightarrow 0^+ \quad (4.33)$$

But an upper bound for $F_1(t)$ and $F_2(t)$ is obtained if $\|\partial_{tt}^2 \mathbf{u}(t)\|_0 < \infty$, and this holds if the compatibility condition (4.31) does.

If the continuous solution \mathbf{u} is exponentially stable [Jos], [Ge], then (4.32) are also valid as global estimates in time, that is, the functions $F_1(t)$ and $F_2(t)$ are bounded above as $t \rightarrow \infty$. But this is only true if the time step size is such that

$$\Delta t \leq Ch^{3/2} \quad (4.34)$$

for $\theta = 1/2$. Restriction (4.34) is not needed if the Crank-Nicolson scheme is combined with the implicit Euler method in the way explained in Reference [HR4]. In any case, this analysis shows that the exponential stability of the continuous problem implies exponential stability of its discrete counterpart. A comforting result, actually.

4.4.2 Streamline Diffusion operator

The Galerkin approach discussed so far has a hidden difficulty not apparent in the error estimates (4.24), (4.25) and (4.28): the stability constants are proportional to the Reynolds number of the problem, Re . Therefore, for $Re^{-1} < h$, these estimates are misleading, they are not as optimal as they appear at first glance.

As for the convection-diffusion equation, the suboptimal rate of convergence of the Galerkin approach is not only reflected by a more or less small loss of accuracy, but it is found in practice that important numerical oscillations occur. The parameter that now plays the role of the Péclet number is the *cell (or element) Reynolds number*, already introduced in Chapter 3:

$$(Re)^e := \rho \frac{|\mathbf{u}^e| h^e}{2\mu} \quad (4.35)$$

Linear elements are expected to yield oscillatory results for $(Re)^e > 1$ and quadratic elements for $(Re)^e > 2$.

The use of upwind techniques is absolutely necessary for convection-diffusion problems. However, this is often questioned for the Navier-Stokes equations and in fact these methods (as a family) are blamed to be inaccurate in some well known text books [CSS], [Gu]. We firmly believe that they are also necessary in this case. The problem is that high Reynolds numbers are associated to complex flow features, such as small recirculation zones, boundary layers, flow detachment, periodic oscillating flow patterns, instabilities and, finally, turbulence. There is no way to capture all these flow details but using small element sizes and therefore small cell Reynolds numbers.

Semidiscrete problem

Let us start considering problem (4.23). In order to stabilize the convective term $c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h)$, the same procedure as for the convection-diffusion equation will be applied. The test function \mathbf{v}_h will be perturbed by adding a term only affecting the element interiors. This term will be proportional to *the convection operator* applied to the test function.

The variational problem to be solved is the following: Find $\mathbf{u}_h(\mathbf{x}, t) \in V_{h,s}$ and $p_h(\mathbf{x}, t) \in Q_{h,s}$ such that

$$\begin{aligned} \rho(\partial_t \mathbf{u}_h, \mathbf{v}_h) + c(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}_h) + a(\mathbf{u}_h, \mathbf{v}_h) - b(p_h, \mathbf{v}_h) \\ + \sum_{e=1}^{N_{el}} \mathcal{S}^e(\mathbf{u}_h, p_h; \mathbf{v}_h) = l(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_{h,t} \\ b(q_h, \mathbf{u}_h) = 0 \quad \forall q_h \in Q_{h,t} \\ (\mathbf{u}_h(\mathbf{x}, 0), \mathbf{v}_h) = (\mathbf{u}_0(\mathbf{x}), \mathbf{v}_h) \quad \forall \mathbf{v}_h \in V_{h,t} \end{aligned} \quad (4.36)$$

where $\mathcal{S}^e(\mathbf{u}_h, p_h; \mathbf{v}_h)$ is the nonlinear functional

$$\mathcal{S}^e(\mathbf{u}_h, p_h; \mathbf{v}_h) := \int_{\Omega^e} \zeta(\mathbf{u}_h, \mathbf{v}_h) \cdot [\mathcal{N}(\mathbf{u}_h, p_h) - \rho \mathbf{f}] d\Omega \quad (4.37)$$

The perturbation ζ of the test function is defined as

$$\zeta(\mathbf{u}_h, \mathbf{v}_h) := \tau^e(\mathbf{u}_h \cdot \nabla) \mathbf{v}_h \quad (4.38)$$

where τ^ϵ is the intrinsic time to be specified later. In (4.37), the Navier-Stokes operator \mathcal{N} is

$$\mathcal{N}(\mathbf{u}_h, p_h) := \rho[\partial_t \mathbf{u}_h + (\mathbf{u}_h \cdot \nabla) \mathbf{u}_h] - \mu \Delta \mathbf{u}_h + \nabla p_h \quad (4.39)$$

Remarks 4.1

- (1) We have employed the simplified version $\mu \Delta \mathbf{u}_h$ instead of $2\mu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}_h)$ for the viscous term. Observe that second derivatives of the shape functions will be needed for calculating this term. The main problem with this simplification will be found in the case of nonconstant viscosities. Implicitly, in (4.37) it is assumed that the viscosity is constant for each element. In practice, this situation will be rarely found: fluids with variable viscosity usually flow at a very low Reynolds numbers, in which case adding the SD operator to the Galerkin equations is unnecessary.
- (2) The functional $\mathcal{S}^\epsilon(\mathbf{u}_h, p_h; \mathbf{v}_h)$, defined on $V_{h,s} \times Q_{h,s} \times V_{h,t}$, is linear in the last two arguments, but highly nonlinear in the first. Besides the quadratic dependence on \mathbf{u}_h of $\mathcal{N}(\mathbf{u}_h, p_h)$ and the linear dependence of the term $(\mathbf{u}_h \cdot \nabla) \mathbf{v}_h$, the intrinsic time τ^ϵ will be a function of $|\mathbf{u}_h|$ and the cell Reynolds number $(Re)^\epsilon$ given by (4.35).
- (3) The perturbation $\zeta(\mathbf{u}_h, \mathbf{v}_h)$ will be in practice calculated not with a variable velocity $\mathbf{u}_h(\mathbf{x}, t)$, but with a characteristic value for each element, $\mathbf{u}^\epsilon(t)$, usually taken as the mean velocity in the element. \square

The definition of the SD method (4.36) has the important drawback that it is not clear how to discretize it in time. Once the spatial discretization has been done, a system of nonlinear ordinary differential equations of the form

$$\dot{\mathbf{x}} + F_1(\mathbf{x}, t) + F_2(\mathbf{x}, \dot{\mathbf{x}}, t) = 0$$

is found, with $F_2(\mathbf{x}, \dot{\mathbf{x}}, t)$ coming from the SD term and $F_1(\mathbf{x}, t)$ from the Galerkin terms.

Fully discrete problem

The conceptual problem found above is due to the fact that we are mixing a variational method for the spatial discretization and a finite difference method to discretize in time. In order to have a problem where only the space has to be discretized, we may assume given the time discretization (problem (4.20)) and then to discretize in space. Following this approach, a SD term will have to be added to the Galerkin equations (4.26), and not to (4.23). The residual method we will end up with is the following:

For $n = 1, 2, \dots, N$, given $\mathbf{u}_h^{n-1}(\mathbf{x})$ and $p_h^{n-1}(\mathbf{x})$ find $\mathbf{u}_h^n(\mathbf{x})$ and $p_h^n(\mathbf{x})$ such that

$$\begin{aligned} & \rho \frac{1}{\Delta t} (\mathbf{u}_h^n - \mathbf{u}_h^{n-1}, \mathbf{v}_h) + \theta c(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta) c(\mathbf{u}_h^{n-1}, \mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ & + \theta a(\mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta) a(\mathbf{u}_h^{n-1}, \mathbf{v}_h) - \theta b(p_h^n, \mathbf{v}_h) - (1 - \theta) b(p_h^{n-1}, \mathbf{v}_h) \\ & + \sum_{\epsilon=1}^{N_{el}} \mathcal{S}^{n,\epsilon}(\mathbf{u}_h, p_h; \mathbf{v}_h) = \theta l^n(\mathbf{v}_h) + (1 - \theta) l^{n-1}(\mathbf{v}_h) \end{aligned} \quad (4.40)$$

$$b(q_h, \mathbf{u}_h^n) = 0$$

where $\mathcal{S}^{n,\epsilon}(\mathbf{u}_h, p_h; \mathbf{v}_h)$ is defined as

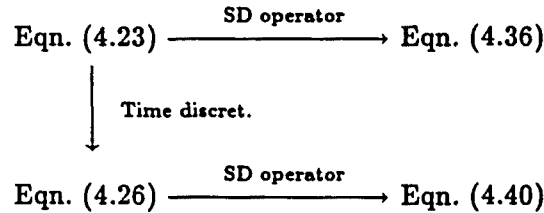
$$\mathcal{S}^{n,\epsilon}(\mathbf{u}_h, p_h; \mathbf{v}_h) := \int_{\Omega^\epsilon} \zeta(\mathbf{u}_h^n, \mathbf{v}_h) \cdot [\mathcal{N}_\theta^n(\mathbf{u}_h, p_h) - \rho \mathbf{f}_\theta^n] d\Omega \quad (4.41)$$

Here, ζ is again given by (4.38) (but evaluated with \mathbf{u}_h^n) and

$$\begin{aligned} \mathcal{N}_\theta^n(\mathbf{u}_h, p_h) := & \rho \frac{1}{\Delta t} (\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) + \rho \theta (\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^n + \rho (1 - \theta) (\mathbf{u}_h^{n-1} \cdot \nabla) \mathbf{u}_h^{n-1} \\ & - \mu \theta \Delta \mathbf{u}_h^n - \mu (1 - \theta) \Delta \mathbf{u}_h^{n-1} + \theta \nabla p_h^n + (1 - \theta) \nabla p_h^{n-1} \end{aligned} \quad (4.42)$$

Remarks 4.2

- (1) Observe that the perturbation ζ given by (4.38) has to be calculated using the velocity \mathbf{u}_h^n , since what is pretended using (4.40) is to balance the convective term $\rho \theta (\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^n$ with the viscous term $2\mu \theta \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}_h^n)$ (written in weak form).
- (2) Clearly, (4.40) is a residual method, i.e., the continuous functions in the space variable $\mathbf{u}^n(\mathbf{x})$ and $p^n(\mathbf{x})$ solution of problem (4.20) satisfy (4.40) for all n .
- (3) The following diagram represents the relation between the semidiscrete and the fully discrete problems using the Galerkin approach and adding the SD operator. Problem (4.40) is *not* obtained from (4.36) using the generalized trapezoidal rule for the time discretization in its standard form. \square



The SD method we will consider in all what follows is (4.40). It only remains to apply the ideas of Chapter 1 to compute the intrinsic time τ^e .

Definition of the intrinsic time

The parameter τ^e will be calculated for each element using the results of Chapter 1, collected in Section 1.6. The only remarkable aspect is that for quadratic elements a single upwind function will be used (see Box 1.1). Moreover, both for linear and quadratic elements, the upwind functions will be approximated by their asymptotic expressions. From the numerical results of Chapter 1 it was concluded that this procedure results in a certain loss of accuracy, but the numerical implementation is much easier and cheaper. In particular, the calculations to be carried out are exactly the same for linear and for quadratic elements.

The steps to be followed to compute τ^e for element e , $e = 1, \dots, N_{el}$, are:

- Compute \mathbf{u}^e as the mean velocity over the element.
- Compute $\mathbf{u}_0^e = \mathbf{J}^{-1} \mathbf{u}^e$, where \mathbf{J} is the Jacobian matrix of the isoparametric mapping to the parent domain evaluated at the center of gravity of the element (assumed to be the point with velocity \mathbf{u}^e).
- Compute the characteristic length as (formula (1.118)):

$$h^e = h_0 \frac{|\mathbf{u}^e|}{|\mathbf{u}_0^e|}$$

The values $h_0 = 2$ and $h_0 = 0.7$ are recommended for the standard parent domains [ZT] of quadrilaterals and triangles, respectively.

- Calculate the cell Reynolds number $(Re)^e$ given by (4.35) using the values just obtained.
- Set the upwind function equal to

$$\alpha^e = \alpha_0 \min\left(\frac{(Re)^e}{3}, 1\right)$$

where $\alpha_0 = 1$ for linear elements and $\alpha_0 = 1/2$ for quadratics.

- Finally, compute

$$\tau^e = \frac{\alpha^e h^e}{2|u^e|} \quad (4.43)$$

Some remarks about least-squares techniques

The SD method described above is close to the original SUPG technique of Brooks & Hughes [BH] and used by many authors. Reference [ADP] is sometimes considered as the first to make a systematic use of this formulation.

Already before the first paper of Hughes *et al.* [HFB] about the Galerkin/least-squares method, Johnson & Saranen proposed in Reference [JS] a velocity-pressure formulation for the Navier-Stokes equations introducing a perturbation of the test functions of the form

$$\tau[(\mathbf{u}_h \cdot \nabla)\mathbf{v}_h + \nabla q_h] \quad (4.44)$$

i.e., including the gradient of the pressure test function (see also [Joh]). Although the method was not analyzed in the quoted reference, the analysis of Hughes *et al.* for the Stokes problem revealed that this was the key for circumventing the BB condition. The least-squares techniques followed as a natural consequence of these results (see the references of Chapter 3), both for the Stokes problem and the Navier-Stokes equations. Several partial results concerning the convergence of this method are already available [HS], [TLu].

A very interesting fact is that sometimes the introduction of ∇q_h in (4.44) is equivalent to a mixed velocity-pressure formulation using div-stable interpolations. In particular, Bank & Welfert [BW] proved that this is indeed the case for the Stokes problem if the minielement of Arnold *et al.* [ABF] is used for the Galerkin approach and the linear simplicial element is used for the least-squares formulation. Of course the finite element interpolation is simpler in the latter case, but the construction and assembly of the element matrices is more complicated. It is not known which method is finally more efficient. Probably, this will depend on the problem. The use of mixed interpolations to stabilize the pressure has the important advantage that discontinuous pressure spaces are easily accommodated and therefore penalty methods fit nicely in this approach.

Let us finally mention other approaches to least-squares methods such as methods based on other variables [HG], [JS] or the interpretation of the least-squares procedure using certain time stepping algorithms [Sa], [Zi]. For another upwind technique different from the SD method and fully analyzed, see [GR2].

4.5 Linearized equations and penalty methods

4.5.1 Linearization of the convective term and the Streamline Diffusion operator

The algorithm we will use for the computations is (4.40). It only remains to describe how it is linearized in order to implement it in a computer code.

There are two sources of nonlinearity: the convective term and the SD operator. The first has a quadratic dependence on the velocity \mathbf{u}_h^n . We will consider the two iterative methods analyzed in Chapter 3 in the context of iterative penalization, namely, the Picard and the Newton-Raphson algorithms. Both methods may be written in the same unified expression. Assume that the velocity at time step n and iteration $i - 1$ ($i \geq 1$) is known. This velocity will be denoted by $\mathbf{u}_h^{n,(i-1)}$. Then, the convective term evaluated at $\mathbf{u}_h^{n,(i)}$ will be approximated by:

$$\begin{aligned} c(\mathbf{u}_h^{n,(i)}, \mathbf{u}_h^{n,(i)}, \mathbf{v}_h) &\approx c(\mathbf{u}_h^{n,(i-1)}, \mathbf{u}_h^{n,(i)}, \mathbf{v}_h) + \beta c(\mathbf{u}_h^{n,(i)}, \mathbf{u}_h^{n,(i-1)}, \mathbf{v}_h) \\ &\quad - \beta c(\mathbf{u}_h^{n,(i-1)}, \mathbf{u}_h^{n,(i-1)}, \mathbf{v}_h) \end{aligned} \quad (4.45)$$

For $\beta = 0$ this is the Picard approximation and for $\beta = 1$ the Newton-Raphson method. If $\delta \mathbf{u} := \mathbf{u}_h^{n,(i)} - \mathbf{u}_h^{n,(i-1)}$, the linearization error in the first case is $O(\|\delta \mathbf{u}\|_1)$ and in the second case it is $O(\|\delta \mathbf{u}\|_1^2)$.

The convective term in the Navier-Stokes operator \mathcal{N}_θ^n given by (4.42) will be linearized in a similar way:

$$\begin{aligned} \rho \theta (\mathbf{u}_h^{n,(i)} \cdot \nabla) \mathbf{u}_h^{n,(i)} &\approx \rho \theta (\mathbf{u}_h^{n,(i-1)} \cdot \nabla) \mathbf{u}_h^{n,(i)} + \rho \theta \beta (\mathbf{u}_h^{n,(i)} \cdot \nabla) \mathbf{u}_h^{n,(i-1)} \\ &\quad - \rho \theta \beta (\mathbf{u}_h^{n,(i-1)} \cdot \nabla) \mathbf{u}_h^{n,(i-1)} \end{aligned} \quad (4.46)$$

Concerning the perturbation of the test function (4.38) that defines the SD method, the velocity with which it is calculated should only affect the accuracy of the numerical method, not the convergence of the iterative procedure. In other words, this source of nonlinearity appears only because the accuracy will be improved evaluating ζ with \mathbf{u}_h^n , one of the unknowns of the problem. In order to simplify the calculations per iteration, ζ has been calculated using the velocity of the previous iteration, $\mathbf{u}_h^{n,(i-1)}$. This leads to the following linearized expression of the SD operator:

$$S^{n,(i),e}(\mathbf{u}_h, p_h, \mathbf{v}_h) \approx \int_{\Omega^e} \zeta(\mathbf{u}_h^{n,(i-1)}, \mathbf{v}_h) \cdot [\mathcal{N}_{\theta,\beta}^{n,(i)}(\mathbf{u}_h, p_h) - \rho \mathbf{f}_\theta^n] d\Omega \quad (4.47)$$

where

$$\begin{aligned} \mathcal{N}_{\theta,\beta}^{n,(i)}(\mathbf{u}_h, \mathbf{v}_h) &:= \rho \frac{1}{\Delta t} (\mathbf{u}_h^{n,(i)} - \mathbf{u}_h^{n-1}) + \rho \theta (\mathbf{u}_h^{n,(i-1)} \cdot \nabla) \mathbf{u}_h^{n,(i)} \\ &\quad + \rho \theta \beta (\mathbf{u}_h^{n,(i)} \cdot \nabla) \mathbf{u}_h^{n,(i-1)} - \rho \theta \beta (\mathbf{u}_h^{n,(i-1)} \cdot \nabla) \mathbf{u}_h^{n,(i-1)} \\ &\quad + \rho (1 - \theta) (\mathbf{u}_h^{n-1} \cdot \nabla) \mathbf{u}_h^{n-1} - \mu \theta \Delta \mathbf{u}_h^{n,(i)} \\ &\quad - \mu (1 - \theta) \Delta \mathbf{u}_h^{n-1} + \theta \nabla p_h^{n,(i)} + (1 - \theta) \nabla p_h^{n-1} \end{aligned} \quad (4.48)$$

is the linearized expression for the Navier-Stokes operator at time step number n and iteration number i .

Remarks 4.3

- (1) The fact that ζ is evaluated with $\mathbf{u}_h^{n,(i-1)}$ could hinder to achieve quadratic convergence of the Newton-Raphson scheme. Numerical experiments indicate that this happens sometimes, but in general the difference in the test functions from one iteration to another keeps the quadratic rate of convergence.
- (2) The values at time step $n - 1$ are considered converged. This is why no iteration superscript has been introduced for them.
- (3) In practice, the initial guess for each time step has been taken as the converged unknown from the previous step, that is,

$$\mathbf{u}_h^{n,(0)} = \mathbf{u}_h^{n-1} \quad (4.49)$$

- (4) In Chapter 3 it has already been said that the Picard scheme converges for the steady-state problem whenever condition (3.12) holds and that the Newton-Raphson algorithm is convergent if the initial guess is close enough to the final solution. For transient problems condition (3.12) does not make sense: the solution is unique for both the continuous and the discrete problems in 2D. For 3D problems, a unique solution also exists for the discrete problem (it is not known whether this is true or not in the continuum). This can be proved using a discrete Gronwall inequality as in References [GR1], [JS] (the stability of this discrete solution is another matter). The only requirement is that the time step size Δt be sufficiently small. This provides a natural way for obtaining stable stationary solutions of the Navier-Stokes equations, whenever they exist: to advance in time until the steady-state is reached. This avoids the need for using continuation techniques for the stationary equations. The situation is completely different in solid mechanics, where the differential equations of motion involve second time derivatives. □

4.5.2 Penalty methods

Similarly to what was done for the stationary equations, the incompressibility constraint in problem (4.40) will be penalized. In view of the results of Chapter 3, the iterative penalization is considered as a way to satisfy this condition. Once again, the iterations due to the nonlinearity of the problem and the penalization will be dealt with in a single iterative loop.

From the algorithmic standpoint, it is possible to place three different penalty methods within the same coding structure. For that, consider that the incompressibility condition in (4.40) is replaced by the penalized equation

$$\epsilon(p_h^{n,\epsilon(i)}, q_h) + b(q_h, \mathbf{u}_h^{n,\epsilon(i)}) = \epsilon(p_h^*, q_h) \quad (4.50)$$

for all $q_h \in Q_{h,t}$. The superscript ϵ has been added to indicate that the solution comes from a penalized problem. According to the pressure p_h^* to be introduced in (4.50) we obtain the following penalty methods:

- $p_h^* \equiv 0$: *Classical penalty method*

This approach should be viewed as a perturbation of the initial problem. The incompressibility constraint will be satisfied up to an error of order ϵ that will not improve as the iterative procedure goes on.

- $p_h^* = p_h^{n-1}$: *Artificial compressibility*

From (4.50) we will have that

$$\epsilon(p_h^{n,\epsilon(i)} - p_h^{n-1}, q_h) + b(q_h, \mathbf{u}_h^{n,\epsilon(i)}) = 0 \quad (4.51)$$

This is the discrete version of the continuous equation

$$\frac{1}{c^2} \partial_t p + \nabla \cdot \mathbf{u} = 0, \quad \text{in } \Omega \times (0, T)$$

provided that the backward Euler scheme is used to discretize in time and ϵ is taken as

$$\epsilon = 1/c^2 \Delta t$$

the constant c being the speed of sound of a slightly compressible fluid. Thus, setting $p_h^* = p_h^{n-1}$ a particular version of the artificial compressibility method of Chorin [Ch] is obtained. Clearly, if the steady-state is reached, $\partial p / \partial t \rightarrow 0$ as $t \rightarrow \infty$ and $p_h^n - p_h^{n-1} \rightarrow 0$ as $n \rightarrow \infty$ for the discrete problem. Therefore, $b(q_h, \mathbf{u}_h^{n,\epsilon(i)}) \rightarrow 0$ as $n \rightarrow \infty$. But when we are far from the steady-state or it simply does not exist, an error of order ϵ will again remain for the incompressibility constraint.

- $p_h^* = p_h^{n,\epsilon(i-1)}$: *Iterative penalization*

This method is the extension of the one analyzed in the previous chapter for the stationary equations to the transient problem. For each time step n , the incompressibility condition is expected to be iteratively approximated. Although the convergence analysis of this method has not been attempted, numerical experiments show that the norm of the discrete divergence of the velocity field in fact decreases similarly to what was observed for the stationary problem. Some of these numerical results will be presented in Section 4.8.

4.5.3 Fully discrete and linearized algorithm

The final problem will be (4.40) with the approximations (4.45), (4.46) and (4.47) for the linearization of the nonlinear terms and (4.50) for the penalty method to be used. In the equations below, we assume that the iterative penalization is employed, that is, the pressure p_h^* is set equal to $p_h^{n,\epsilon(i-1)}$. It is understood that the other possibilities described earlier can be also considered.

The perturbation of the test function for the SD term is computed using the characteristic element velocity \mathbf{u}^e , computed as the mean value of the velocity $\mathbf{u}_h^{n,\epsilon(i-1)}$ over element e . The calculation of the intrinsic time τ^e has already been described in detail.

Concerning the way convergence is checked, we have used the following criterion:

$$\|\mathbf{u}_h^{n,\epsilon(i)} - \mathbf{u}_h^{n,\epsilon(i-1)}\|_{L^q} \leq TOL \|\mathbf{u}_h^{n,\epsilon(i)}\|_{L^q} \quad (4.52)$$

where TOL is a given tolerance and $\|\cdot\|_{L^q}$ denotes the discrete L^q norm. A selected choice for q controls the convergence, although the norms for $q = 1$, $q = 2$ and $q = \infty$ (i.e., maximum norm) are always computed. There is also a check to decide whether

the steady-state has been reached or not. Since the difference between \mathbf{u}_h^n and \mathbf{u}_h^{n-1} will be of order Δt , the stationarity criterion that has been chosen is

$$\|\mathbf{u}_h^n - \mathbf{u}_h^{n-1}\|_{L^e} \leq TOL \Delta t \|\mathbf{u}_h^n\|_{L^e} \quad (4.53)$$

All the terms that are known for a given iteration within a time step have been written in the right-hand-side of the equations. These equations are

• *Momentum equation:*

$$\begin{aligned} & \rho(\mathbf{u}_h^{n,\epsilon(i)}, \mathbf{v}_h) + \theta \Delta t c(\mathbf{u}_h^{n,\epsilon(i-1)}, \mathbf{u}_h^{n,\epsilon(i)}, \mathbf{v}_h) + \theta \Delta t \beta c(\mathbf{u}_h^{n,\epsilon(i)}, \mathbf{u}_h^{n,\epsilon(i-1)}, \mathbf{v}_h) \\ & + \theta \Delta t a(\mathbf{u}_h^{n,\epsilon(i)}, \mathbf{v}_h) - \theta \Delta t b(p_h^{n,\epsilon(i)}, \mathbf{v}_h) \\ & + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [\tau^e(\mathbf{u}^e \cdot \nabla) \mathbf{v}_h] \cdot \left[\rho \mathbf{u}_h^{n,\epsilon(i)} + \rho \theta \Delta t (\mathbf{u}_h^{n,\epsilon(i-1)} \cdot \nabla) \mathbf{u}_h^{n,\epsilon(i)} \right. \\ & \quad \left. + \rho \theta \Delta t \beta (\mathbf{u}_h^{n,\epsilon(i)} \cdot \nabla) \mathbf{u}_h^{n,\epsilon(i-1)} - \mu \theta \Delta t \Delta \mathbf{u}_h^{n,\epsilon(i)} + \theta \Delta t \nabla p_h^{n,\epsilon(i)} \right] \\ & = l_\theta^n(\mathbf{v}_h) + \rho(\mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ & - (1 - \theta) \Delta t c(\mathbf{u}_h^{n-1}, \mathbf{u}_h^{n-1}, \mathbf{v}_h) + \theta \Delta t \beta c(\mathbf{u}_h^{n,\epsilon(i-1)}, \mathbf{u}_h^{n,\epsilon(i-1)}, \mathbf{v}_h) \\ & - (1 - \theta) \Delta t a(\mathbf{u}_h^{n-1}, \mathbf{v}_h) + (1 - \theta) \Delta t b(p_h^{n-1}, \mathbf{v}_h) \\ & + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [\tau^e(\mathbf{u}^e \cdot \nabla) \mathbf{v}_h] \cdot \left[\rho \mathbf{u}_h^{n-1} - \rho(1 - \theta) \Delta t (\mathbf{u}_h^{n-1} \cdot \nabla) \mathbf{u}_h^{n-1} \right. \\ & \quad \left. + \rho \theta \Delta t \beta (\mathbf{u}_h^{n,\epsilon(i-1)} \cdot \nabla) \mathbf{u}_h^{n,\epsilon(i-1)} + \mu(1 - \theta) \Delta t \Delta \mathbf{u}_h^{n-1} - (1 - \theta) \Delta t \nabla p_h^{n-1} \right. \\ & \quad \left. + \theta \Delta t \mathbf{f}^n + (1 - \theta) \mathbf{f}^{n-1} \right] \end{aligned} \quad (4.54)$$

• *Penalized incompressibility equation:*

$$\epsilon(p_h^{n,\epsilon(i)}, q_h) + b(q_h, \mathbf{u}_h^{n,\epsilon(i)}) = \epsilon(p_h^{n,\epsilon(i-1)}, q_h) \quad (4.55)$$

4.6 Matrix formulation

The different problems considered so far will be written now in matrix form. This will allow us to present the basic flow chart of the algorithm implemented in a computer code that collects the numerical techniques that have been discussed in this chapter.

Let us introduce the vector

$$\mathbf{s} := \tau^e \mathbf{u}^e \quad (4.56)$$

defined for each element e with characteristic velocity \mathbf{u}^e and intrinsic time τ^e , computed as described earlier. If \mathbf{s} is calculated using the velocity obtained for the i -th iteration of the n -th time step, we will indicate it by $\mathbf{s}^{n,i}$.

Once the finite element interpolation has been chosen, every element of the spaces of test functions and of trial solutions will be represented by a vector containing the nodal values of this element. This vector will be denoted by the boldface capital letter corresponding to the lower case function. For example, \mathbf{V} will be the vector of

nodal values of a generic velocity test function and \mathbf{U} the vector of nodal values of the unknown velocity. Superscripts will be used to indicate the time step and the iteration counter.

The definitions of the matrices that will be needed are collected in Box 4.2. The L^2 inner product in the pressure space has been denoted by $(\cdot, \cdot)_Q$ and in the velocity space by $(\cdot, \cdot)_V$.

Box 4.2 Matrix form of the discrete equations

<u>Matrix version</u>	<u>Terms from where it comes</u>
$\mathbf{V}^T \cdot \mathbf{M}_{v,s} \cdot \mathbf{U}$	$\rho(\mathbf{u}_h, \mathbf{v}_h)_V + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot (\rho \mathbf{u}_h) d\Omega$
$\mathbf{V}^T \cdot \mathbf{K}_{c,s}(\mathbf{U}_1) \cdot \mathbf{U}_2$	$c(\mathbf{u}_{h,1}, \mathbf{u}_{h,2}, \mathbf{v}_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot [\rho(\mathbf{u}_{h,1} \cdot \nabla) \mathbf{u}_{h,2}] d\Omega$
$\mathbf{V}^T \cdot \mathbf{K}_{c,s}^*(\mathbf{U}_1) \cdot \mathbf{U}_2$	$c(\mathbf{u}_{h,2}, \mathbf{u}_{h,1}, \mathbf{v}_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot [\rho(\mathbf{u}_{h,2} \cdot \nabla) \mathbf{u}_{h,1}] d\Omega$
$\mathbf{V}^T \cdot \mathbf{K}_{d,s} \cdot \mathbf{U}$	$a(\mathbf{u}_h, \mathbf{v}_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot (-\mu \Delta \mathbf{u}_h) d\Omega$
$\mathbf{V}^T \cdot \mathbf{G}_s \cdot \mathbf{P}$	$b(p_h, \mathbf{v}_h) - \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot \nabla p d\Omega$
$\mathbf{V}^T \cdot \mathbf{F}_{v,s}$	$l(\mathbf{v}_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s} \cdot \nabla) \mathbf{v}_h] \cdot (\rho \mathbf{f}) d\Omega$
$\mathbf{Q}^T \cdot \mathbf{M}_p \cdot \mathbf{P}$	$(p_h, q_h)_Q$

Having introduced all these matrices and vectors, some of the problems considered heretofore can be written as follows:

- *Problem (4.23), semidiscrete Galerkin:*

Find $\mathbf{U} = \mathbf{U}(t)$ and $\mathbf{P} = \mathbf{P}(t)$ such that, for $t \in (0, T)$,

$$\begin{aligned}
 \mathbf{M}_{v,0} \cdot \frac{d}{dt} \mathbf{U} + \mathbf{K}_{c,0}(\mathbf{U}) \cdot \mathbf{U} + \mathbf{K}_{d,0} \cdot \mathbf{U} - \mathbf{G}_0 \cdot \mathbf{P} &= \mathbf{F}_{v,0} \\
 \mathbf{G}_0^T \cdot \mathbf{U} &= 0 \\
 \mathbf{M}_{v,0} \cdot \mathbf{U}(0) &= \mathbf{U}_0
 \end{aligned} \tag{4.57}$$

where \mathbf{U}_0 comes from the right-hand-side in the initial condition of problem (4.23).

Neither in (4.57) nor in what follows Dirichlet boundary conditions have been introduced. They will lead to a force term in the discrete continuity equation. Subscript naught in Eqns. (4.57) indicates that the matrices are calculated with $\mathbf{s} = 0$.

- Problem (4.26), fully discrete Galerkin:

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} and \mathbf{P}^{n-1} , find \mathbf{U}^n and \mathbf{P}^n , approximations to $\mathbf{U}(t^n)$ and $\mathbf{P}(t^n)$, such that

$$\begin{aligned} \mathbf{M}_{v,0} \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{c,0}(\mathbf{U}^n) \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{d,0} \cdot \mathbf{U}^n - \theta \Delta t \mathbf{G}_0 \cdot \mathbf{P}^n \\ = \theta \Delta t \mathbf{F}_{v,0}^n + (1 - \theta) \Delta t \mathbf{F}_{v,0}^{n-1} + \mathbf{M}_{v,0} \cdot \mathbf{U}^{n-1} \\ - (1 - \theta) \Delta t \mathbf{K}_{c,0}(\mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} - (1 - \theta) \Delta t \mathbf{K}_{d,0} \cdot \mathbf{U}^{n-1} \\ + (1 - \theta) \Delta t \mathbf{G}_0 \cdot \mathbf{P}^{n-1} \end{aligned} \quad (4.58)$$

$$\mathbf{G}_0^T \cdot \mathbf{U}^n = 0$$

- Problem (4.40), fully discrete SD method:

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} and \mathbf{P}^{n-1} , find \mathbf{U}^n and \mathbf{P}^n , approximations to $\mathbf{U}(t^n)$ and $\mathbf{P}(t^n)$, such that

$$\begin{aligned} \mathbf{M}_{v,s^n} \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{c,s^n}(\mathbf{U}^n) \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{d,s^n} \cdot \mathbf{U}^n - \theta \Delta t \mathbf{G}_{s^n} \cdot \mathbf{P}^n \\ = \theta \Delta t \mathbf{F}_{v,s^n}^n + (1 - \theta) \Delta t \mathbf{F}_{v,s^n}^{n-1} + \mathbf{M}_{v,s^n} \cdot \mathbf{U}^{n-1} \\ - (1 - \theta) \Delta t \mathbf{K}_{c,s^n}(\mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} - (1 - \theta) \Delta t \mathbf{K}_{d,s^n} \cdot \mathbf{U}^{n-1} \\ + (1 - \theta) \Delta t \mathbf{G}_{s^n} \cdot \mathbf{P}^{n-1} \end{aligned} \quad (4.59)$$

$$\mathbf{G}_0^T \cdot \mathbf{U}^n = 0$$

- Problem (4.54)–(4.55), fully discrete and linearized SD method:

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} and \mathbf{P}^{n-1} , find \mathbf{U}^n and \mathbf{P}^n , approximations to $\mathbf{U}(t^n)$ and $\mathbf{P}(t^n)$, as the converged solutions of the following iterative algorithm:

$$\begin{aligned} \mathbf{M}_{v,s^{n,i-1}} \cdot \mathbf{U}^{n,\epsilon(i)} + \theta \Delta t \mathbf{K}_{c,s^{n,i-1}}(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\ + \theta \Delta t \beta \mathbf{K}_{c,s^{n,i-1}}^*(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\ + \theta \Delta t \mathbf{K}_{d,s^{n,i-1}} \cdot \mathbf{U}^{n,\epsilon(i)} - \theta \Delta t \mathbf{G}_{s^{n,i-1}} \cdot \mathbf{P}^{n,\epsilon(i)} \\ = \theta \Delta t \mathbf{F}_{v,s^{n,i-1}}^n + (1 - \theta) \Delta t \mathbf{F}_{v,s^{n,i-1}}^{n-1} + \mathbf{M}_{v,s^{n,i-1}} \cdot \mathbf{U}^{n-1} \\ - (1 - \theta) \Delta t \mathbf{K}_{c,s^{n,i-1}}(\mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} \\ + \theta \Delta t \beta \mathbf{K}_{c,s^{n,i-1}}^*(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i-1)} \\ - (1 - \theta) \Delta t \mathbf{K}_{d,s^{n,i-1}} \cdot \mathbf{U}^{n-1} + (1 - \theta) \Delta t \mathbf{G}_{s^{n,i-1}} \cdot \mathbf{P}^{n-1} \end{aligned} \quad (4.60)$$

$$\epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i)} + \mathbf{G}_0^T \cdot \mathbf{U}^{n,\epsilon(i)} = \epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i-1)}$$

We are now in a position to present the basic flow chart of the algorithm to be implemented on the computer. This has been schematically represented in Box 4.3, where the following integers have been introduced: n_{eu} is the number of time steps in which the Euler scheme ($\theta = 1$) is to be used and i_{pi} is the number of iterations to be carried out using the Picard scheme ($\beta = 0$). For Δt small enough, i_{pi} may be set to zero, since the solution of the previous time step will be a good initial guess for the solution of the current step and the Newton-Raphson method ($\beta = 1$) will converge.

For steady flow calculations, the convergence tolerance within each time step may be larger than the tolerance to check if the steady-state has been reached, in order to perform only one iteration per time step.

In Box 4.3 we have taken into account the fact that the pressure is discontinuous between elements. This allows to eliminate the pressure in the momentum equation as explained in Chapter 3. Expressions (3.48)–(3.50) have been used for the problem now considered to form the element matrices of the final algebraic system.

Box 4.3 Algorithm for the transient Navier–Stokes equations

- Set the initial condition \mathbf{U}^0 and $\mathbf{P}^0 = 0$
- $n := 0$
- WHILE $n < N$ and (*non-stationary*) DO:
 - $n \leftarrow n + 1$
 - IF $n < n_{cu}$ then $\theta = 1$
ELSE select θ , $\theta \geq 1/2$
 - $i := 0$
 - Set $\mathbf{U}^{n,\epsilon(0)} = \mathbf{U}^{n-1}$ and $\mathbf{P}^{n,\epsilon(0)} = \mathbf{P}^{n-1}$
 - WHILE (*not converged*) DO:
 - $i \leftarrow i + 1$
 - IF $i < i_{pi}$ then $\beta = 0$
ELSE $\beta = 1$
 - IF (*classical penalization*) then $\mathbf{P}^* = 0$
ELSE if (*artificial compressibility*) then $\mathbf{P}^* = \mathbf{P}^{n-1}$
ELSE if (*iterative penalization*) then $\mathbf{P}^* = \mathbf{P}^{n,\epsilon(i-1)}$
 - For each element, compute $\mathbf{s}^{n,i-1}$ and

$$\mathbf{A}^{(\epsilon)} := \mathbf{M}_{v,s^{n,i-1}}^{(\epsilon)} + \theta \Delta t \mathbf{K}_{c,s^{n,i-1}}^{(\epsilon)} (\mathbf{U}^{(\epsilon),n,\epsilon(i-1)})$$

$$+ \theta \Delta t \beta \mathbf{K}_{c,s^{n,i-1}}^{(\epsilon),*} (\mathbf{U}^{(\epsilon),n,\epsilon(i-1)}) + \theta \Delta t \mathbf{K}_{d,s^{n,i-1}}^{(\epsilon)}$$

$$+ \theta \Delta t \frac{1}{\epsilon} \mathbf{G}_{s^{n,i-1}}^{(\epsilon)} \mathbf{M}_p^{(\epsilon)-1} \mathbf{G}_0^{(\epsilon)T}$$

$$\mathbf{B}^{(\epsilon)} := \theta \Delta t \mathbf{F}_{v,s^{n,i-1}}^{(\epsilon),n} + (1 - \theta) \Delta t \mathbf{F}_{v,s^{n,i-1}}^{(\epsilon),n-1} + \mathbf{M}_{v,s^{n,i-1}}^{(\epsilon)} \mathbf{U}^{(\epsilon),n-1}$$

$$- (1 - \theta) \Delta t \mathbf{K}_{c,s^{n,i-1}}^{(\epsilon)} (\mathbf{U}^{(\epsilon),n-1}) \mathbf{U}^{(\epsilon),n-1}$$

$$+ \theta \Delta t \beta \mathbf{K}_{c,s^{n,i-1}}^{(\epsilon),*} (\mathbf{U}^{(\epsilon),n,\epsilon(i-1)}) \mathbf{U}^{(\epsilon),n,\epsilon(i-1)}$$

$$- (1 - \theta) \Delta t \mathbf{K}_{d,s^{n,i-1}}^{(\epsilon)} \mathbf{U}^{(\epsilon),n-1}$$

$$+ (1 - \theta) \Delta t \mathbf{G}_{s^{n,i-1}}^{(\epsilon)} \mathbf{P}^{(\epsilon),n-1} + \theta \Delta t \mathbf{G}_{s^{n,i-1}}^{(\epsilon)} \mathbf{P}^{(\epsilon),*}$$
 - Assemble $\mathbf{A}^{(\epsilon)}$ and $\mathbf{B}^{(\epsilon)}$ and solve $\mathbf{A} \mathbf{U}^{n,\epsilon(i)} = \mathbf{B}$
 - Compute $\mathbf{P}^{n,\epsilon(i)} = \mathbf{P}^* - \frac{1}{\epsilon} \mathbf{M}_p^{-1} \mathbf{G}_0^T \mathbf{U}^{n,\epsilon(i)}$
 - IF $\|\mathbf{U}^{n,\epsilon(i)} - \mathbf{U}^{n,\epsilon(i-1)}\|_{L^q} \leq TOL \|\mathbf{U}^{n,\epsilon(i)}\|_{L^q}$ then (*converged*)
 - END while (*not converged*)
 - $\mathbf{U}^n \leftarrow \mathbf{U}^{n,\epsilon(i)}$
 - $\mathbf{P}^n \leftarrow \mathbf{P}^{n,\epsilon(i)}$
 - IF $\|\mathbf{U}^n - \mathbf{U}^{n-1}\|_{L^q} \leq TOL \Delta t \|\mathbf{U}^n\|_{L^q}$ then (*stationary*)
- END while $n < N$ and (*non-stationary*)
- END

4.7 Computing secondary variables

Once the velocity and pressure are calculated, one may be interested in other physical unknowns of the problem. Moreover, for visualizing the flow it is interesting to obtain a continuous pressure field and, for two dimensional flows, the streamfunction. Here we will describe some numerical procedures to obtain nodal values of the pressure, the vorticity and the physical properties of the fluid whenever they be variable. An algorithm to compute the streamfunction will also be described.

4.7.1 Least-squares smoothing

In the numerical procedure described in the preceeding sections, the pressure nodal values are located within each element. Also, when the physical properties of the fluid are variable (Chapters 5 and 6) they have to be stored at the integration points in order to perform the numerical quadrature. All these scalar fields will be discontinuous across interelement boundaries. For plotting purposes, it is interesting to obtain a continuous function that approximates a discontinuous one. Here, the least-squares technique employed in our calculations will be briefly described.

Let ϕ_c be a computed function, discontinuous across elements. A continuous function ϕ_s is then calculated by minimizing:

$$\|\phi_c - \phi_s\|_{L^2}^2 = \int_{\Omega} (\phi_c - \phi_s)^2 d\Omega \quad (4.61)$$

The function ϕ_s is interpolated like the components of the velocity field. If N_{tp} is the total number of nodal points of the mesh, $N^{(i)}$ denotes the shape function associated to node i and $\phi_s^{(i)}$ is the nodal value of ϕ_s at this point, the minimization of the functional (4.61) leads to the system:

$$\mathbf{M}^c \Phi = \mathbf{R} \quad (4.62)$$

where the components of the matrix \mathbf{M}^c and the vectors Φ and \mathbf{R} are:

$$M_{ij}^c = \int_{\Omega} N^{(i)} N^{(j)} d\Omega, \quad i, j = 1, \dots, N_{tp} \quad (4.63)$$

$$\Phi_j = \phi_s^{(j)}, \quad j = 1, \dots, N_{tp} \quad (4.64)$$

$$R_i = \int_{\Omega} N^{(i)} \phi_c d\Omega, \quad i = 1, \dots, N_{tp} \quad (4.65)$$

This smoothing technique is standard [Hu2], [ZT]. In order to avoid the solution of the system (4.62), it is usual to approximate the matrix \mathbf{M}^c by a diagonal matrix \mathbf{M}^l . This matrix can be obtained either by the row-sum lumping technique or by using a nodal quadrature rule to evaluate the integrals in (4.63) and (4.65). In this case, the quadrature points are placed on the nodes of the element and the shape functions are assumed to be such that

$$N^{(i)}(\mathbf{x}_j) = \delta_{ij} \quad (4.66)$$

when evaluated at the j -th node of the finite element mesh, with coordinates \mathbf{x}_j . An estimate of how well ϕ_s approximates ϕ_c can be easily obtained using standard results from interpolation theory and numerical quadrature theory [SF].

4.7.2 Nodal quadrature rules

We present in Box 4.4 some nodal quadrature rules for the most common finite elements used in practice. Some of these rules are well known (rules 1–3, 6–11, 17 in Box 4.4). Our interest in obtaining the others is not their accuracy but the fact that they allow to approximate the matrix M^c in Eqn. (4.62) by a diagonal matrix, as explained above.

In Box 4.4, R_n indicates the rule number and N_{no} the number of nodes of the element. This is followed by a schematic description of the element that has to be precised. For both 2D and 3D elements, the bubble function is associated with a node placed at the center of the element. It is understood that the original shape functions (without the addition of the new node) have to be modified in order to have zero value at the center. Otherwise, the nodal unknown at this point would not have the meaning of being the value of the interpolated function and the matrix M^c would not be diagonal, since condition (4.66) would not hold. For the element considered in rule number 15, bubble functions are also added in the center of the faces of the element. Elements corresponding to rules number 1, 2, 5 and 6 are triangular, tetrahedral for rules number 9, 10, 13 and 14, quadrilateral for rules number 3, 4, 7 and 8 and hexahedral for rules number 11, 12, 15 and 16.

The quadrature rule is defined by the weights of the nodes. All the nodes placed at the corners of the element have the same weight, as well as the nodes placed in the middle of the edges and in the center of the faces (in 3D elements). The values given have been normalized in such a way that their sum is 1. In the final entry of Box 4.4, the accuracy of the quadrature rule is given by the polynomial that can be exactly integrated. The set of polynomials of degree n is denoted by P_n , whereas Q_n denotes the set of tensor-product polynomials of degree n in each Cartesian direction x, y, z .

All the rules except rule number 5 are the best that can be obtained with the given number of quadrature points. In fact, for the 2D quadratic (simplicial) element, a second order quadrature rule is obtained if the weights are taken as 0 for the corner nodes and $\frac{1}{3}$ for the mid-side nodes. However, this rule yields a matrix M^l , approximation of M^c , with some zero diagonal values (those corresponding to the corner nodes). The weights given for rule number 5 have been obtained splitting the triangle into four subtriangles and applying rule number 1. It is interesting to remark that if the Richardson extrapolation is applied to rules number 1 and 5, the mentioned second order rule is recovered.

In general, these quadrature rules cannot be used for the numerical integration of the matrices of the discrete Navier-Stokes equations, since their accuracy is not enough to preserve the order of convergence of the finite element discretization. Open quadrature rules have to be employed in these cases, i.e., with the nodes placed in the interior of the elements. The product Gauss-Legendre rule is the common option for quadrilateral and hexahedral elements. Open quadrature rules for triangles can be found in Reference [LG] and for tetrahedra in Reference [GeH].

4.7.3 Pressure, vorticity and physical properties smoothing

The least-squares technique combined with the nodal quadrature rules to compute the integrals will be applied now to approximate several discontinuous fields by continuous functions. The number of integration points used for the calculation of the element matrices of the Navier-Stokes equations will be denoted by N_{gp} . A point in the parent

domain Ω_0 will be indicated by ξ .

Box 4.4 Nodal quadrature rules for linear and quadratic elements

Two-dimensional elements

R_n	N_{no}	Description	Weights			Polynomial
			Corners	Edges	Center	
1	3	Linear	1/3			P_1
2	4	Linear + bubble	1/12		3/4	P_2
3	4	Bilinear	1/4			Q_1
4	5	Bilinear + bubble	1/12		2/3	P_2
5	6	Quadratic	1/12	1/4		P_1
6	7	Quad. + bubble	1/20	2/15	9/20	P_3
7	8	Serendipid	-1/12	1/3		P_2
8	9	Biquadratic	1/36	1/9	4/9	Q_3

Three-dimensional elements

R_n	N_{no}	Description	Weights				Polynomial
			Corners	Edges	Faces	Center	
9	4	Linear	1/4				P_1
10	5	Linear + bubble	1/20			4/5	P_2
11	8	Trilinear	1/8				Q_1
12	9	Trilinear + bubble	1/24			2/3	P_2
13	10	Quadratic	-1/120	1/5			P_2
14	11	Quad. + bubble	1/160	1/15		8/15	P_3
15	15	Quad. + bubble + face bubbles	17/840	4/105	27/280	32/105	P_3 and terms x^2yz, xy^2z, xyz^2
16	20	Serendipid	-1/8	1/6			P_2
17	27	Triquadratic	1/216	1/54	2/27	8/27	Q_3

Since all the matrices and vectors are obtained from the assembly of their element contributions, we will only concentrate on these elemental expressions.

The Gramm matrix appearing in Eqn. (4.62) can be approximated in all the cases by the diagonal matrix resulting from the nodal quadrature rule. The right-hand-side term in this equation, R , can be computed either using this nodal rule or the same numerical integration employed for the Navier-Stokes equations. For the smoothing of the pressure and the vorticity, both options are equally easy to implement. However, when the physical properties of the fluid are considered, we will see that the second procedure is much easier than the former.

Let $J(\boldsymbol{\xi})$ be the Jacobian determinant of the isoparametric mapping and w_k , $k = 1, \dots, N_{no}$, the weights given in Box 4.4 multiplied by $\text{meas}(\Omega_0)$. The components of the element contributions to the matrix M^e and the approximated matrix M^l are:

$$\begin{aligned}
 M_{ij}^{(e),c} &= \int_{\Omega^e} N^{(e,i)} N^{(e,j)} d\Omega = \int_{\Omega_0} N^{(e,i)}(\boldsymbol{\xi}) N^{(e,j)}(\boldsymbol{\xi}) |J^{(e)}(\boldsymbol{\xi})| d\Omega_0 \\
 &\approx M_{ij}^{(e),l} = \sum_{k=1}^{N_{no}} w_k |J^{(e)}(\boldsymbol{\xi}_k)| N^{(e,i)}(\boldsymbol{\xi}_k) N^{(e,j)}(\boldsymbol{\xi}_k) \\
 &= \sum_{k=1}^{N_{no}} w_k |J^{(e)}(\boldsymbol{\xi}_k)| \delta_{ik} \delta_{jk} \\
 &= [w_i |J^{(e)}(\boldsymbol{\xi}_i)|] \delta_{ij} \quad (\text{no sum})
 \end{aligned} \tag{4.67}$$

Let $N_p^{(e,j)}$ be the pressure shape function associated to the j -th node of element e and $p^{(e,j)}$ the corresponding pressure nodal value. The components of the force term \mathbf{R} for the pressure smoothing approximated by the nodal quadrature rule will be

$$\begin{aligned}
 R_{p,i}^{(e)} &= \int_{\Omega^e} N^{(e,i)} p d\Omega = \int_{\Omega^e} N^{(e,i)} \left(\sum_{j=1}^{N_{qp}} N_p^{(e,j)} p^{(e,j)} \right) d\Omega \\
 &= \int_{\Omega_0} N^{(e,i)}(\boldsymbol{\xi}) \left(\sum_{j=1}^{N_{qp}} N_p^{(e,j)}(\boldsymbol{\xi}) p^{(e,j)} \right) |J^{(e)}(\boldsymbol{\xi})| d\Omega_0 \\
 &\approx \sum_{k=1}^{N_{no}} w_k |J^{(e)}(\boldsymbol{\xi}_k)| \delta_{ik} \left(\sum_{j=1}^{N_{qp}} N_p^{(e,j)}(\boldsymbol{\xi}_k) p^{(e,j)} \right) \\
 &= w_i |J^{(e)}(\boldsymbol{\xi}_i)| \left(\sum_{j=1}^{N_{qp}} N_p^{(e,j)}(\boldsymbol{\xi}_i) p^{(e,j)} \right)
 \end{aligned} \tag{4.68}$$

It is observed from (4.68) that all the shape functions (those associated to the continuous approximation and the discontinuous pressure interpolation) and their derivatives have to be evaluated at the nodes of the elements.

The smoothing of the vorticity $\boldsymbol{\omega}_h := \nabla \times \mathbf{u}_h$ can be performed in a similar way. For two dimensional flows, this vector has only one non-zero component, $\omega_3 = \partial_1 u_2 - \partial_2 u_1$, the subscripts referring to the Cartesian coordinates now denoted x_i , $i = 1, 2, 3$. For simplicity, assume that $N_{,d} = 2$. The components for the right-hand-side term \mathbf{R} are now

$$\begin{aligned}
 R_{\omega,i}^{(e)} &= \int_{\Omega^e} N^{(e,i)} \omega_3 d\Omega = \int_{\Omega^e} N^{(e,i)} (\partial_1 u_2 - \partial_2 u_1) d\Omega \\
 &= \int_{\Omega^e} N^{(e,i)} \sum_{j=1}^{N_{no}} \left(\partial_1 N^{(e,j)} U_2^{(e,j)} - \partial_2 N^{(e,j)} U_1^{(e,j)} \right) d\Omega \\
 &= \int_{\Omega_0} N^{(e,i)}(\boldsymbol{\xi}) \sum_{j=1}^{N_{no}} \left(\partial_1 N^{(e,j)}(\boldsymbol{\xi}) U_2^{(e,j)} - \partial_2 N^{(e,j)}(\boldsymbol{\xi}) U_1^{(e,j)} \right) |J^{(e)}(\boldsymbol{\xi})| d\Omega_0 \\
 &\approx w_i |J^{(e)}(\boldsymbol{\xi}_i)| \sum_{j=1}^{N_{no}} \left(\partial_1 N^{(e,j)}(\boldsymbol{\xi}_i) U_2^{(e,j)} - \partial_2 N^{(e,j)}(\boldsymbol{\xi}_i) U_1^{(e,j)} \right)
 \end{aligned} \tag{4.69}$$

Consider now a variable physical property $\varphi(\mathbf{x})$. Examples of this situation will be found in the next two chapters. In order to compute the matrices for the Navier-Stokes equations (or for the temperature equation, as it will be seen in the following chapter) the values of φ have to be stored for each element and for each quadrature point within the element. Let us denote them by $\varphi_j^{(e)}$, $j = 1, \dots, N_{gp}$. If φ is interpolated within the element and this interpolation is used to compute the values at the nodes, the resulting function will be discontinuous and the smoothing is again needed. Observe that the interpolation functions are not the standard shape functions of the element. Therefore, it is easier to compute the integrals in Eqn. (4.65) in this case *using the same numerical integration as for the Navier-Stokes equations*. Otherwise, a new set of interpolation functions should be defined. The right-hand-side term of the smoothing equations will be

$$\begin{aligned} R_{\varphi,i}^{(e)} &= \int_{\Omega^e} N^{(e,i)} \varphi^{(e)} d\Omega = \int_{\Omega_0} N^{(e,i)}(\xi) \varphi^{(e)}(\xi) |J^{(e)}(\xi)| d\Omega \\ &\approx \sum_{k=1}^{N_{gp}} w_k^* |J^{(e)}(\xi_k)| N^{(e,i)}(\xi_k) \varphi_k^{(e)} \end{aligned} \quad (4.70)$$

where w_k^* are the weights for the quadrature rule of N_{gp} points, with coordinates in the parent domain ξ_k .

4.7.4 An algorithm for the calculation of the streamfunction

For incompressible bidimensional flows, the streamfunction provides a simple way for plotting streamlines (its contours) and also gives a measure of the quantity of fluid that crosses a segment of a curve per unit of time, i.e., the flux of the velocity field multiplied by the density. Here we will present an algorithm for the calculation of the nodal values of this function (see Reference [Ja] for a different method).

For exactly divergence free velocities ($\nabla \cdot \mathbf{u}_h = 0$) there exists a streamfunction ψ_h such that $\mathbf{u}_h = \nabla \times \psi_h := (\partial_2 \psi_h, -\partial_1 \psi_h)$. Consider a segment of curve defined by the initial point A and the end point B . Let \mathbf{t} be the tangent to this curve and \mathbf{n} normal to it, $\{\mathbf{n}, \mathbf{t}\}$ having the same orientation as the canonical basis. We will have that

$$\mathbf{n} = (n_1, n_2) = (t_2, -t_1)$$

and hence

$$\begin{aligned} \int_A^B \mathbf{u}_h \cdot \mathbf{n} ds &= \int_A^B (\partial_2 \psi_h, -\partial_1 \psi_h) \cdot (t_2, -t_1) ds \\ &= \int_A^B \nabla \psi_h \cdot \mathbf{t} ds = \psi_h(B) - \psi_h(A) \end{aligned}$$

that is,

$$\psi_h(B) = \psi_h(A) + \int_A^B \mathbf{u}_h \cdot \mathbf{n} ds \quad (4.71)$$

If AB is a straight segment of length $|AB|$ and we define

$$\begin{aligned} \vec{AB} &:= (B_1 - A_1, B_2 - A_2) \\ \vec{AB}^\perp &:= (B_2 - A_2, A_1 - B_1) \end{aligned}$$

we will have that $\mathbf{n} = \vec{AB}^\perp / |AB|$. Assuming that the variation of $\mathbf{u}_h \cdot \mathbf{n}$ along AB is linear, Eqn. (4.71) reduces to

$$\begin{aligned}\psi_h(B) &= \psi_h(A) + \frac{1}{2} [(\mathbf{u}_h \cdot \mathbf{n})(A) + (\mathbf{u}_h \cdot \mathbf{n})(B)] |AB| \\ &= \psi_h(A) + \frac{1}{2} [\mathbf{u}_h(A) + \mathbf{u}_h(B)] \cdot \vec{AB}^\perp\end{aligned}\quad (4.72)$$

Equation (4.72) provides a method to calculate the streamfunction values at the nodes of the finite element mesh. Recall that the two approximations inherent to (4.72) are that AB has been considered a straight segment and the variation of $\mathbf{u}_h \cdot \mathbf{n}$ linear along it.

The velocity \mathbf{u}_h that will be obtained from the finite element solution of the Navier-Stokes equations is not pointwise divergence free. All we can expect is that $\int_{\Omega^e} \nabla \cdot \mathbf{u}_h d\Omega = 0$ for all the elements. In fact, this equation will not hold exactly, since the continuity condition has been penalized. Nevertheless, our method will be based on this equality.

The idea is the following. Once ψ_h is known for a certain node of an element, the value of this function can be computed for the next node using (4.72). In this way, we can go through all the nodes of the element placed on its boundary. What is not possible is to compute ψ_h for the interior nodes of the elements, whenever they exist. What we do in these cases is to interpolate ψ_h for this node using the values calculated for the others and the shape functions corresponding to the interpolation without the interior node. For example, for the seven-noded quadratic triangle enriched with a bubble function, ψ_h for the central node is computed from the quadratic interpolation based on the six-noded triangle.

Once we come back following this process to the first node of the element where the streamfunction was known, the new value may be slightly different from the original one. What we do is to compute ψ_h several times for each node (as many as the algorithm presented thereafter yields) and take the final result as the average of the calculated values.

The final algorithm is presented in Box 4.5, where the following variables and arrays have been introduced:

- $\psi(i)$: Value of ψ_h for node i .
- $NT(i)$: Number of times that ψ_h has been calculated for the node i .
- N_{cp} : Number of points where ψ_h is known.
- $N_{cp}^{(e)}$: Number of points of element e where ψ_h is known.
- $I_{cp}^{(e)}$: First node of element e where ψ_h is known.
- \bar{w}_j : Weights coming from the interpolation of ψ_h for the interior nodes.

Since the streamfunction is determined up to a constant, its value for the first node of the mesh has been set equal to zero. Thus, the algorithm starts with one known value of ψ_h .

The fact that node number N_{no} of the element be interior or not has been indicated by the statement (N_{no} interior). No distinction has been made between the global and the local numbering of a node. In Box 4.5, i_{no} stands for the node where ψ_h is to be calculated and i_{pr} for the 'previous' node, where the streamfunction is already known. Finally, recall that N_{tp} is the total number of nodes of the finite element mesh.

This algorithm has proved to work very well for the problems we have considered, even though the velocity is not exactly weakly solenoidal.

Box 4.5 Algorithm for the calculation of the streamfunction

- Set $e = 0$, $NT(1) = 1$, $\psi(1) = 0$, $N_{cp} = 1$
- WHILE $N_{cp} < N_{tp}$ DO:
 - $e \leftarrow \text{mod}(e + 1, N_{el})$
 - IF $e = 0$ then $e \leftarrow N_{el}$
 - Compute $N_{cp}^{(e)}$
 - IF $0 < N_{cp}^{(e)} < N_{no}$ then
 - Determine $I_{cp}^{(e)}$
 - FOR $i = 1, N_{no}$ DO:
 - $i_{no} = I_{cp}^{(e)} + i$
 - IF $i_{no} = N_{no}$ and (N_{no} interior) then $i_{no} \leftarrow i_{no} + 1$
 - IF $i_{no} > N_{no}$ then $i_{no} \leftarrow \text{mod}(i_{no}, N_{no})$
 - IF $i_{no} > 1$ then $i_{pr} = i_{no} - 1$
 ELSE if (N_{no} interior) then $i_{pr} = N_{no} - 1$
 ELSE $i_{pr} = N_{no}$
 - Compute

$$\psi(i_{no}) \leftarrow \psi(i_{no}) + \psi(i_{pr})$$

$$+ 0.5 [u_1(i_{no}) + u_1(i_{pr})][Y(i_{no}) - Y(i_{pr})]$$

$$+ 0.5 [u_2(i_{no}) + u_2(i_{pr})][X(i_{pr}) - X(i_{no})]$$
 - IF $NT(i_{no}) = 0$ then $N_{cp} \leftarrow N_{cp} + 1$
 - $NT(i_{no}) \leftarrow NT(i_{no}) + 1$
 - END
 - IF (N_{no} interior) then
 - $\psi(N_{no}) = \sum_{j=1}^{N_{no}-1} \bar{w}_j \psi(j)$
 - $N_{cp} \leftarrow N_{cp} + 1$
 - $NT(i_{no}) \leftarrow NT(i_{no}) + 1$
 - END
 - END
- FOR $i = 1, N_{tp}$ DO:
 - $\psi(i) \leftarrow \psi(i)/NT(i)$
- END
- END

4.8 Numerical examples

In this section, some classical benchmark problems for incompressible viscous flows will be solved. The first is the driven cavity flow, for which detailed numerical results can be found, e.g., in References [GGS], [GC2], [GuH], [Ki], [KM], [Sh], [SK], [So], among many others. The second problem is the flow over a backward facing step. Numerical experiments for this problem are reported in References [ADS], [Ga], [GC2], [HRS], [Ki], [KM], [So]. Both for this problem and for the first one the stationary solution is sought. The next example is the flow past a cylinder, for which the steady-state solution is not stable and a periodic flow pattern develops behind the cylinder if a uniform initial condition is slightly perturbed. Numerical experiments for this problem can be found in References [EJ], [GC2], [TGL], [TLi], [TMS].

All the above quoted references have been selected because of the details they give about how the numerical simulation has been carried out, but many other works dealing with numerical models for the Navier-Stokes equations present similar experiments.

Our calculations have been performed on a CONVEX-C120 computer using double arithmetic precision.

Example 4.1 *Flow inside a wall-driven cavity*

The Stokes solution for this problem has been considered in detail in the previous chapter. The essential feature of this benchmark test case when the Navier-Stokes equations are solved is the prediction of various vortices inside the cavity. The notation we will use for them is shown in Figure 4.4.

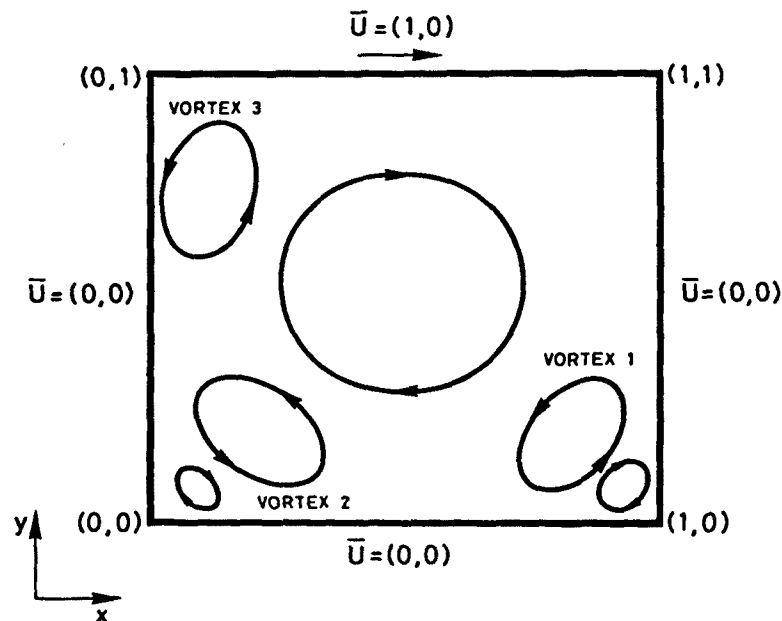


Figure 4.4 Geometry, boundary conditions and nomenclature of cavity flow

Numerical results will be presented for values of the Reynolds number $Re = 1000$, 4000 and 8000, computed using the length of the cavity and the velocity prescribed on the top edge. In References [GGS], [GC2], [Ki] and [So] results are presented for

other values of Re , so our results are complementary to theirs. In particular, the case $Re = 10000$ is solved in these references. This value can be considered as a limit for steady calculations, since Shen has shown through detailed numerical experiments that above this bound the stationary solution ceases to be stable and a Hopf bifurcation occurs [Sh].

The computational domain has been discretized using a mesh composed of 676 Q_2/P_1 elements and 2809 nodal points for all the Reynolds numbers. This mesh has been designed to capture the details of the flow in the corners and boundary layers (see Figure 4.5). The smallest element size is $h_{min} = 0.01$ (twice the distance between nodes).

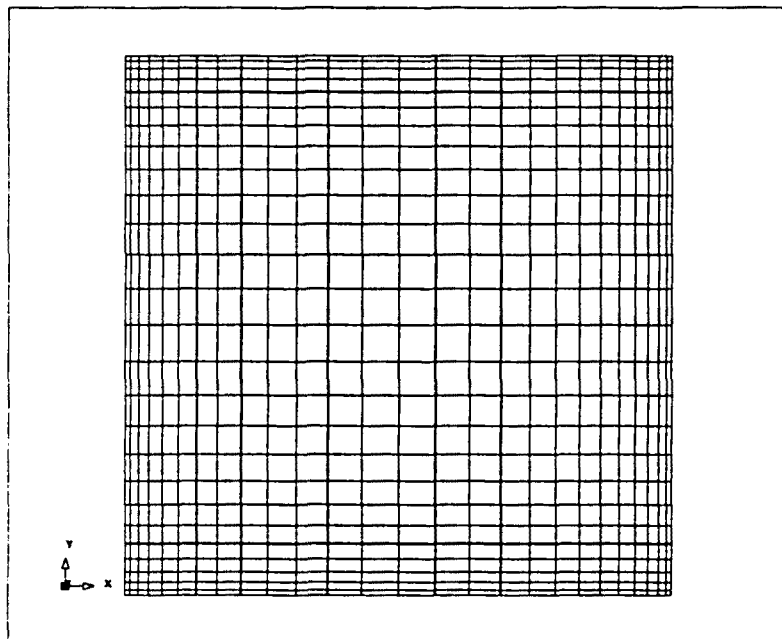


Figure 4.5 Finite element mesh for the cavity flow problem (676 Q_2/P_1 elements, 2809 nodal points).

This mesh is similar to the one used by Gresho *et al.* [GC2], which consists of 50×50 Q_1/P_0 elements and $51 \times 51 = 2601$ nodal points. They also used an upwind technique [GC1]. For $Re > 4000$, the Galerkin method yields oscillatory results and it is only possible to use this method on much finer meshes, as those used by Kim [Ki] (1024 Q_2/P_1 elements, 4225 nodal points), Sohn [So] (1600 Q_2/P_1 elements, 6561 nodal points, $h_{min} = 0.00326$) or Ghia *et al.* [GGS] (uniform mesh of $257 \times 257 = 66049$ nodal points, with $h = 0.0039$). In this last reference, a finite-difference multigrid method based on the streamfunction-vorticity formulation is used.

In all our computations we have taken $\epsilon = 10^{-3}$ (penalty parameter). The iterative penalty method has been employed. Concerning the parameters of the SD method, $h_0 = 2$ has been chosen (element length for the parent domain) and $\alpha_0 = 0.5$ (upwind factor).

For $Re = 1000$, the Galerkin solution yields very good answers, without any oscillations. Results are shown in Figures 4.6 to 4.9.

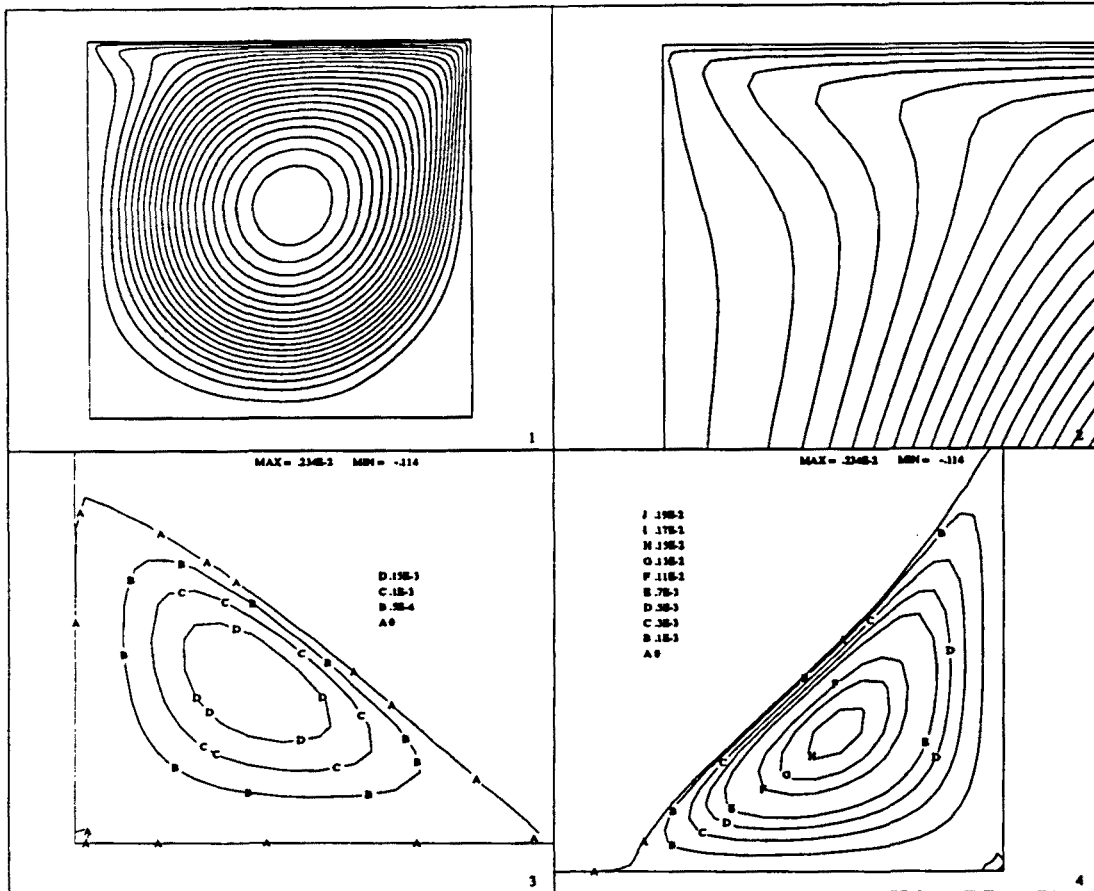


Figure 4.6 Numerical solution of the cavity flow problem at $Re = 1000$ (Galerkin method), streamlines. (1): General pattern; (2): Detail of the top left corner; (3): Detail of bottom left corner; (4): Detail of bottom right corner.

Since the computation has started with zero velocities everywhere, the first effective initial guess is the Stokes solution. If the Newton-Raphson method is then used, the algorithm does not converge. The strategy we have followed is to use the Picard method ($\beta = 0$) for the first three iterations and then move to the Newton-Raphson scheme ($\beta = 1$). For a convergence tolerance in the relative L^2 norm of 0.1 %, i.e., $TOL = 10^{-3}$ in (4.52), eight iterations have been required. The final value of the norm of the discrete divergence of the velocity has been approximately 10^{-11} , starting from an initial value of order 10^{-4} .

The streamlines are shown in Figure 4.6. For this value of the Reynolds number, vortex 3 in Figure 4.4 does not appear. The extreme values of the streamfunction in vortices 1 and 2 will be compared with the results presented in References [GGS], [GC2], [So] and [Ki]. Recall that our results have been obtained using the Galerkin formulation. Results of Reference [So] have been obtained using the FIDAP code, which allows to use a version of the streamline upwind (STU) technique employed also in [GC2] and described in [GC1], but now applied to quadratic elements (in particular, to the Q_2/P_1 pair). This consists basically in adding an anisotropic viscosity following

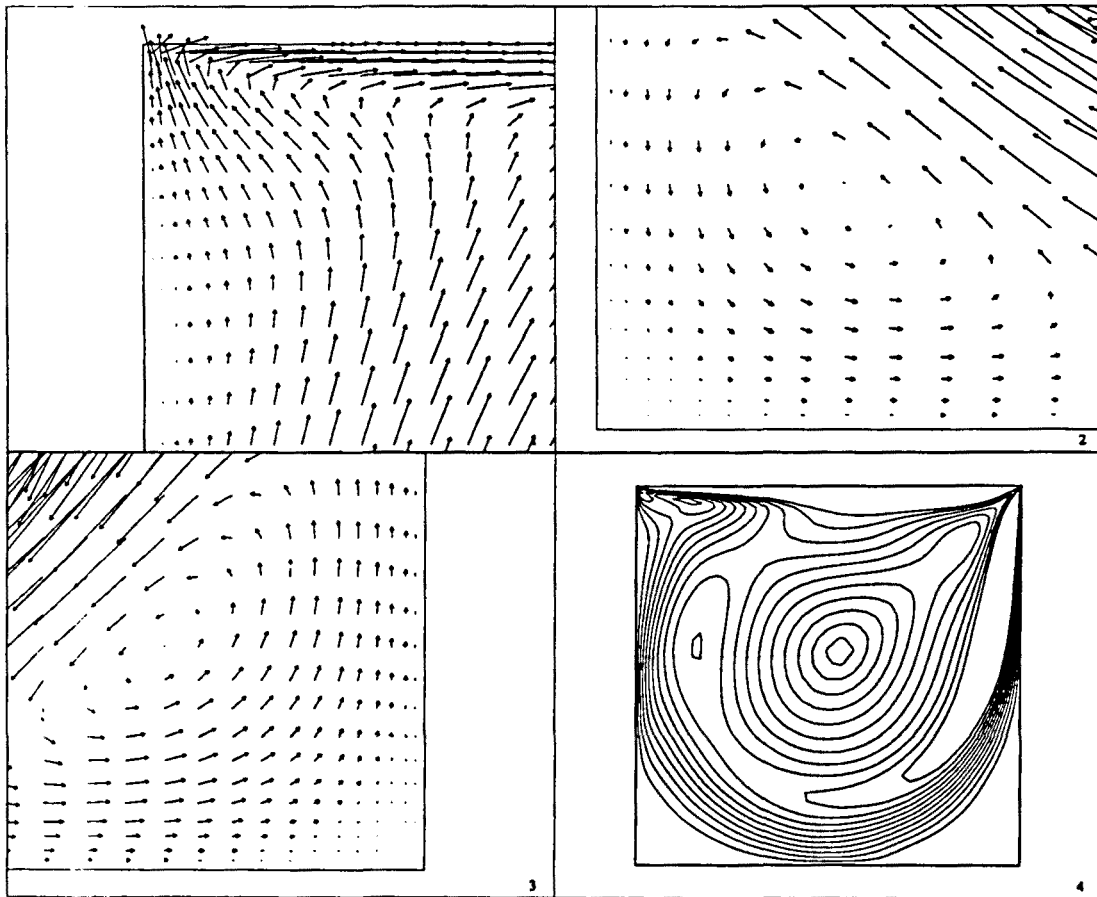


Figure 4.7 Numerical solution of the cavity flow problem at $Re = 1000$, velocities. (1): Detail of the top left corner; (2): Detail of bottom left corner; (3): Detail of bottom right corner; (4): Contours of the velocity norm.

the streamlines, something very similar to what the Taylor-Galerkin method yields. Therefore, the final scheme is not a consistent weighted residual method, in the sense that the exact solution does not satisfy exactly the discrete variational equations.

The extreme values of the streamfunction to be compared are the following:

<u>Reference</u>	<u>Vortex 1</u>	<u>Vortex 2</u>
[GGS]	1.75×10^{-3}	2.31×10^{-4}
[GC2]	1.76×10^{-3}	2.00×10^{-4}
[Ki]	1.66×10^{-3}	2.20×10^{-4}
[So], without STU	1.63×10^{-3}	2.17×10^{-4}
[So], with STU	1.10×10^{-3}	9.40×10^{-5}
Present study	1.61×10^{-3}	1.99×10^{-4}

It is observed that the STU technique used by Sohn yields overdiffusive results

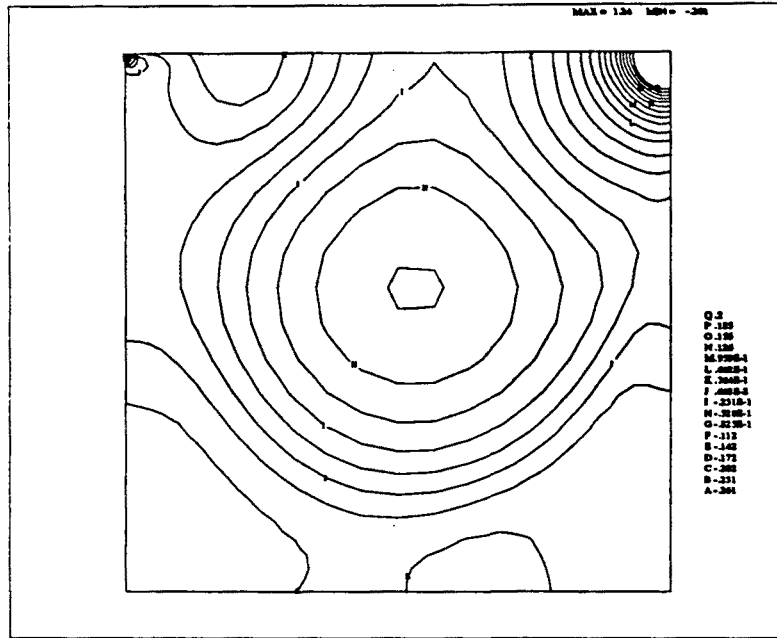


Figure 4.8 Numerical solution of the cavity flow problem at $Re = 1000$. Pressure contours.

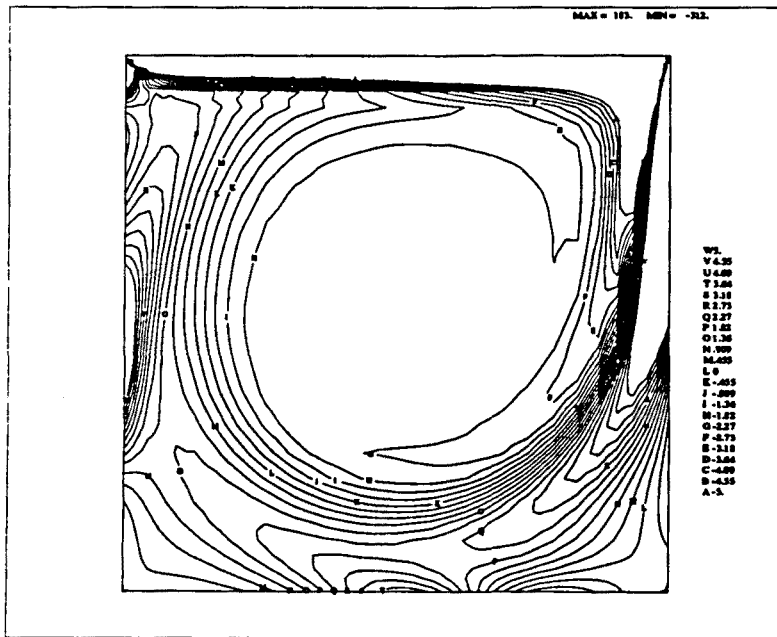


Figure 4.9 Numerical solution of the cavity flow problem at $Re = 1000$. Vorticity contours.

and that the extreme values of ψ_h are higher using linear elements (Q_1/P_0 in [GC2], a difference scheme in [GGS]) than the Q_2/P_1 pair (results of Kim, Sohn and ours). It is

also observed that the extreme values found in the present work are slightly smaller than those in [Ki] and [So]. It is very important to keep this fact in mind because the same behavior will be observed for higher Reynolds numbers using the SD method described in this chapter. Since, apart from the iterative penalty method, our formulation is very close to that employed in [Ki] and [So], we believe that these differences are due to the fact that the mesh employed here is much coarser than theirs. Moreover, Kim has also compared the results for a coarser mesh (25×25 nodal points) concluding that this yields smaller absolute values for the peaks of the streamfunction.

Details of the velocity vectors in the corners of the cavity are shown in Figures 4.7.(1)–(3). The contours of the Euclidian norm of these vectors have been plotted in Figure 4.7.(4). Observe that no oscillations appear. Figures 4.8 and 4.9 show the pressure and vorticity contours, respectively. In general trends, these results are very similar to those presented in the above quoted references.

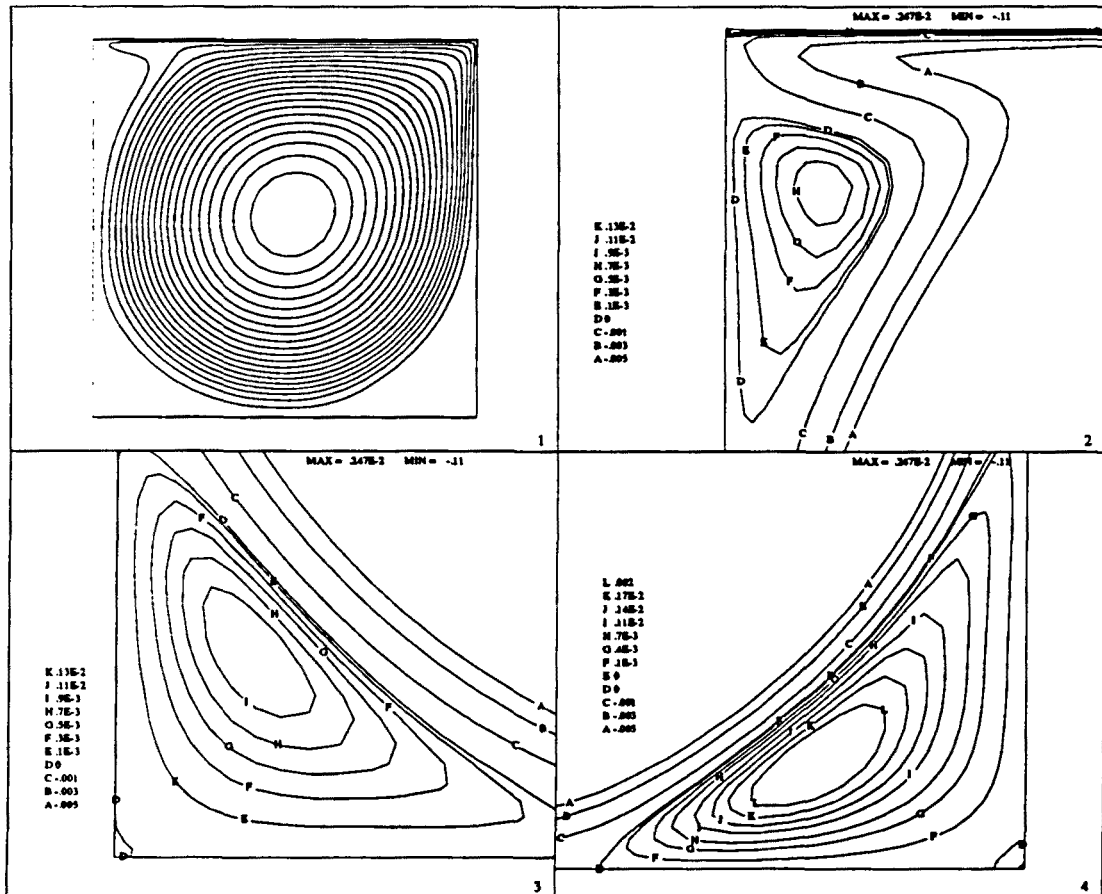


Figure 4.10 Numerical solution of the cavity flow problem at $Re = 4000$ (Galerkin method), streamlines. (1): General pattern; (2): Detail of the top left corner; (3): Detail of bottom left corner; (4): Detail of bottom right corner.

Consider now the case $Re = 4000$. The iterative strategy followed consists in using the Picard scheme for the first three iterations and the Newton-Raphson algorithm from

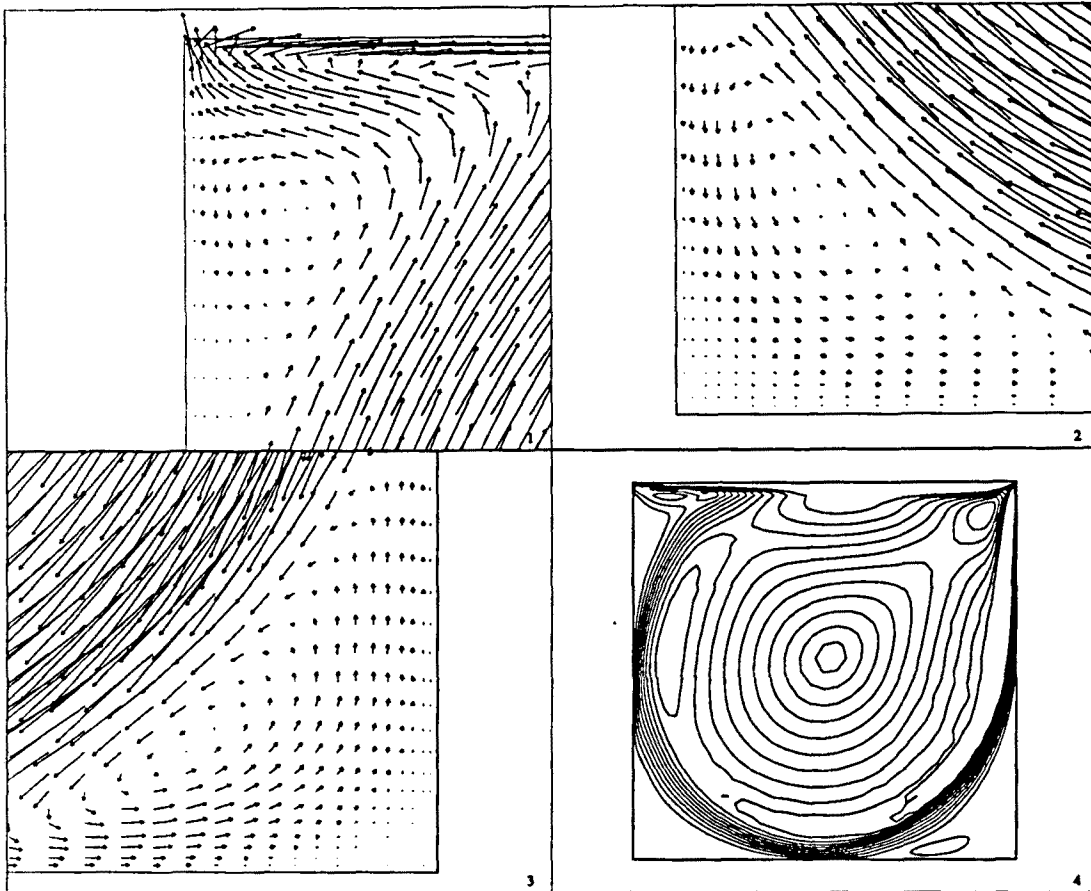


Figure 4.11 Numerical solution of the cavity flow problem at $Re = 4000$, velocities. (1): Detail of the top left corner; (2): Detail of bottom left corner; (3): Detail of bottom right corner; (4): Contours of the velocity norm.

there on. Now, twelve iterations have been needed to converge up to a convergence tolerance of 0.1%, but again the Stokes solution has been found to be a good enough initial guess for the iterative process.

Numerical results are shown in Figures 4.10 to 4.12, corresponding to the same plots as for the $Re = 1000$ case. Once again, the *Galerkin* formulation has been employed.

From Figure 4.10.(2) it is observed that now the top left vortex has appeared, with an extreme value of 0.9×10^{-3} for the streamfunction. The other two vortices have an increased strength with respect to the results for $Re = 1000$.

The velocity vectors are shown in Figure 4.11. From the contours of their norm plotted in Figure 4.11.(4) it is seen that *small numerical oscillations begin to appear* in zones with a high velocity and a relatively large element size, that is, with a large cell Reynolds number. These oscillations appear near the bottom right corner. Nevertheless, they do not affect the quality of the solution in the rest of the domain and, in particular, the vortices are well reproduced. The effect of the small velocity oscillations on the pressure is very weak, as it may be observed from Figure 4.12.

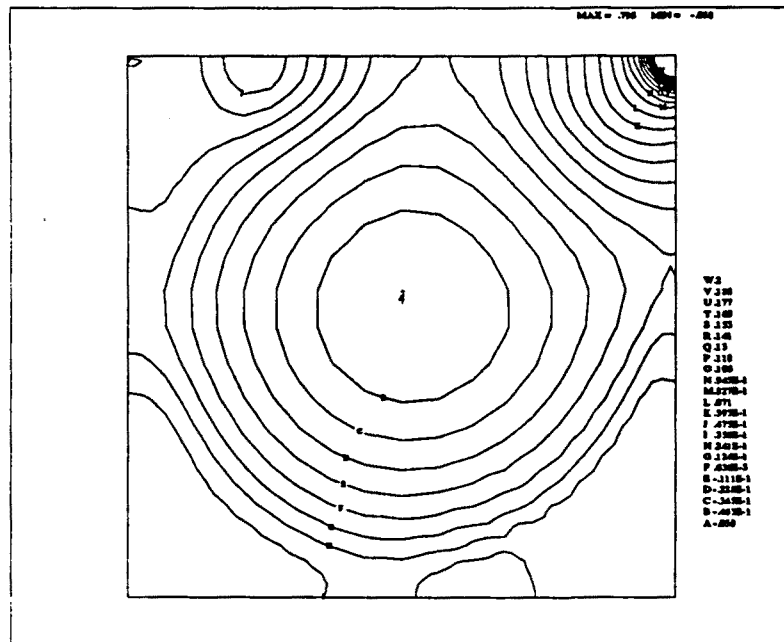


Figure 4.12 Numerical solution of the cavity flow problem at $Re = 4000$. Pressure contours.

The situation is completely different for $Re = 8000$. Concerning the iterative procedure, we have failed to obtain a converged solution by solving directly the stationary Navier-Stokes equations. The alternative to use continuation techniques as in [So] is to advance in time. We have chosen the solution obtained for $Re = 4000$ as the initial condition. In order to decrease the computational effort, the following strategies have been used:

- $\theta = 1$, i.e., the Euler scheme has been used. The steady-state is reached faster than using $\theta = 1/2$ (Crank-Nicolson) and the computational effort is smaller.
- Artificial compressibility method. Only for the steady state a good approximation to the incompressibility constraint is needed. For $\epsilon = 10^{-3}$, the final value of the norm of the discrete velocity divergence has been found to be of order 10^{-10} .
- High convergence tolerance. In order to perform only one iteration per time step, we have taken $TOL = 0.1$ (10%) in (4.52).
- Small tolerance to check the steady-state. $TOL = 10^{-3}$ (0.1%) has been taken in (4.53).
- The time step size has been chosen as $\Delta t = 0.1$. For higher values, the solution oscillates from one time step to another.

Using all these numerical parameters, the steady-state has been found using the SD method for $t = 5.1$, i.e., after 51 time steps. The steady-state has not been found using the Galerkin method. The solution obtained at every time step is oscillatory.

Numerical results using the SD method are shown in Figures 4.13 to 4.15. In general, the flow features encountered for $Re = 4000$ are now accentuated, although nothing new appears. For $Re = 10000$ it is known that new secondary vortices develop in the left and right bottom corners.

Let us compare now the extreme values for the streamfunction at $Re = 5000$ that

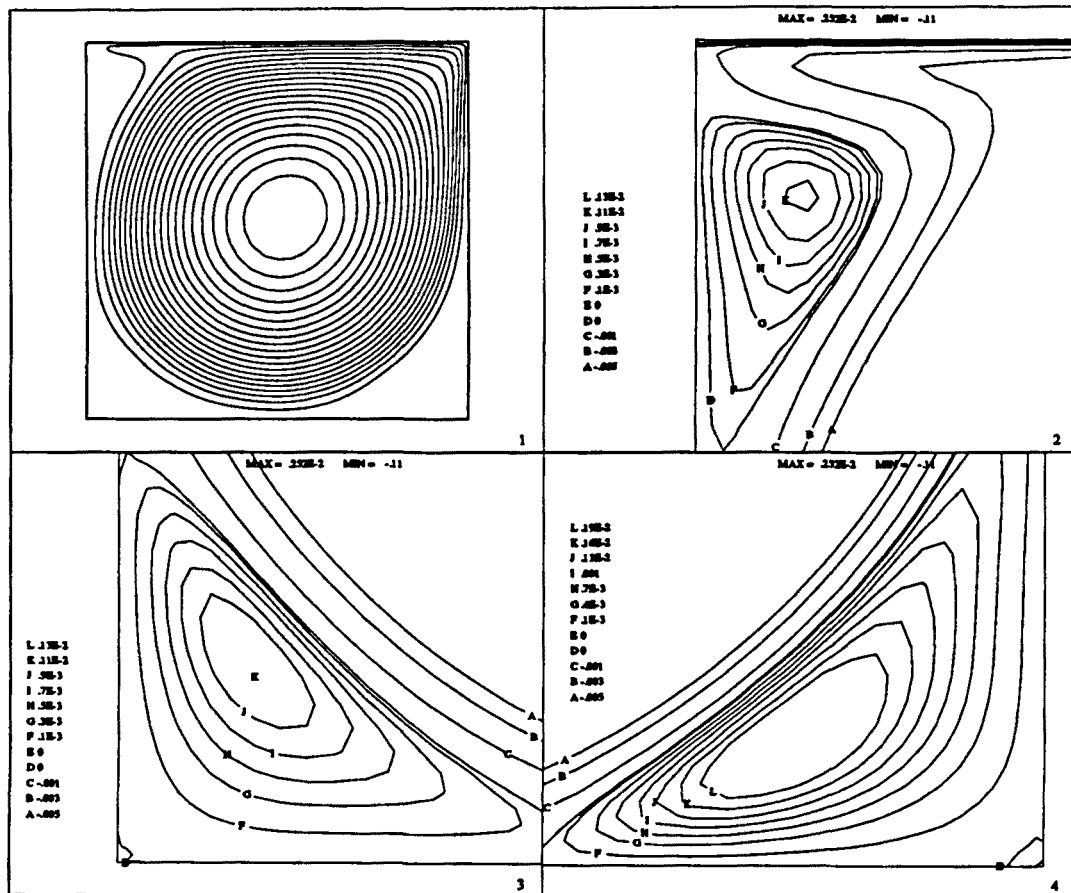


Figure 4.13 Numerical solution of the cavity flow problem at $Re = 8000$ (SD method), streamlines. (1): General pattern; (2): Detail of the top left corner; (3): Detail of bottom left corner; (4): Detail of bottom right corner.

we have obtained (plots not shown) with those given in the previous references. The results are the following:

<u>Reference</u>	<u>Vortex 1</u>	<u>Vortex 3</u>
[GGS]	3.08×10^{-3}	1.46×10^{-3}
[GC2]	3.87×10^{-3}	1.23×10^{-3}
[Ki]	2.79×10^{-3}	1.30×10^{-3}
[So], without STU	2.80×10^{-3}	1.28×10^{-3}
[So], with STU	1.71×10^{-3}	1.94×10^{-4}
Present study	2.49×10^{-3}	1.21×10^{-3}

The conclusions that may be drawn from these values is that the SD method we have employed is much less overdifusive than the STU technique used by Sohn, but peaks are still smaller than in [Ki] and [So] using the Galerkin formulation. Recall

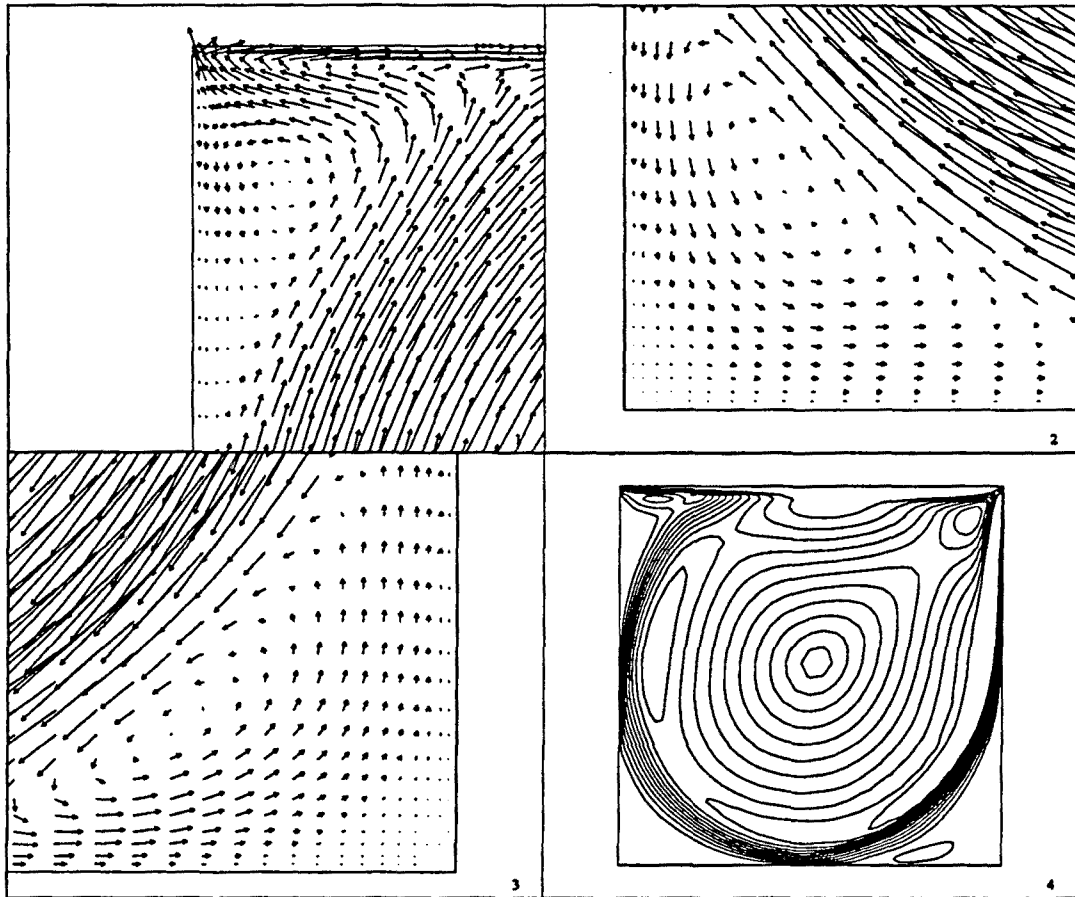


Figure 4.14 Numerical solution of the cavity flow problem at $Re = 8000$, velocities. (1): Detail of the top left corner; (2): Detail of bottom left corner; (3): Detail of bottom right corner; (4): Contours of the velocity norm.

that this also happened for $Re = 1000$ when the Galerkin method was used in our computation. The fact that our mesh is coarser than the one used in [Ki] and [So] may be responsible in part for these results. Anyway, if the SD method contributes to damp peaks out, it is clear that this effect is not very important and the numerical answers are very accurate.

We consider now the convergence of the SD method. When the Picard scheme is used, the rate of convergence is only linear and the way the SD has been linearized does not affect it. The situation is different when the Newton-Raphson algorithm is employed. The linearization of the SD operator described earlier is only linear, and therefore the rate of convergence of the scheme may be driven by this linearization, regardless of the fact that the convective terms of the equations have been linearized up to second order.

Let us see what happens when the P_2^+/P_1 element is used. The mesh used is shown in Figure 4.16. It is an unstructured mesh composed of 338 P_2^+/P_1 elements and 1063 nodal points. The iterative penalization has been used, with $\epsilon = 10^{-4}$. For the SD method we have taken $\alpha_0 = 0.5$ (upwind factor) and $h_0 = 0.7$ (element

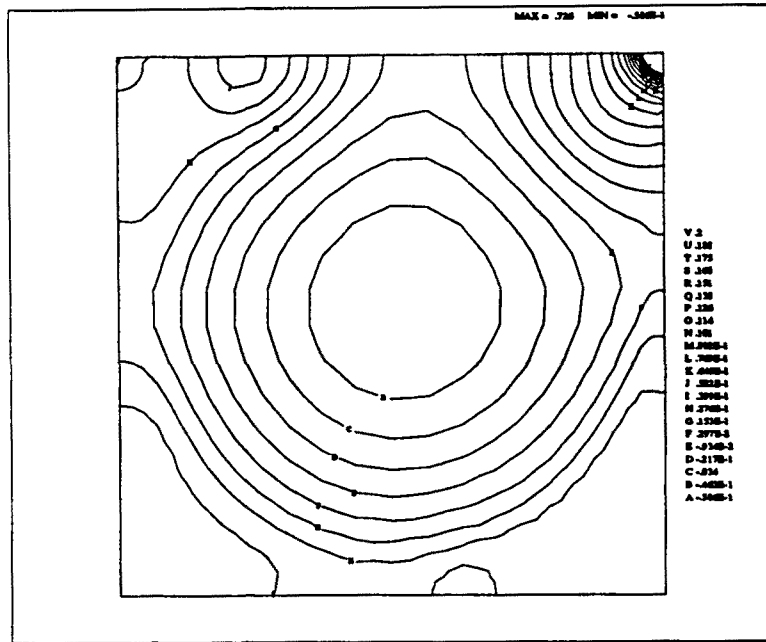


Figure 4.15 Numerical solution of the cavity flow problem at $Re = 8000$. Pressure contours.

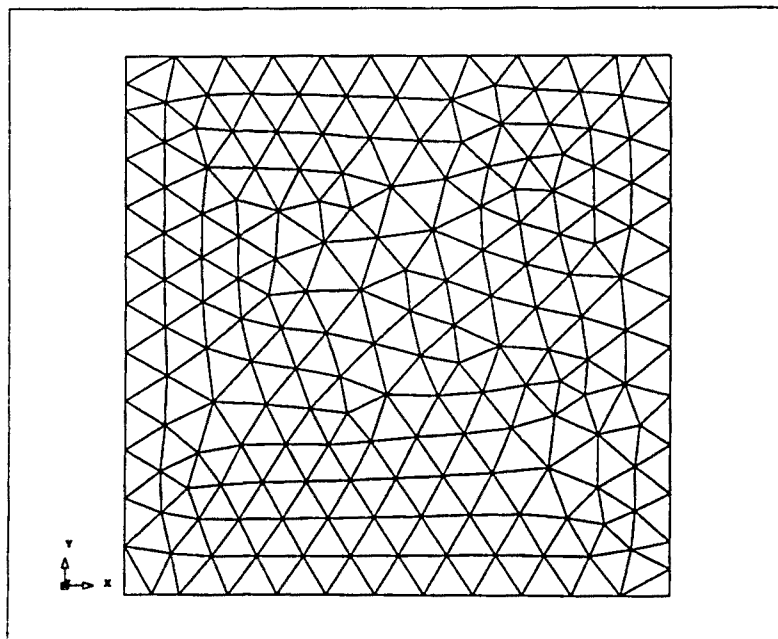


Figure 4.16 Finite element mesh for the cavity flow problem (338 P_2^+/P_1 elements, 1063 nodal points).

length in the parent domain). For $Re = 1000$, the velocity does not oscillate using the Galerkin method. Results are shown in Figure 4.17, both using the Galerkin and the

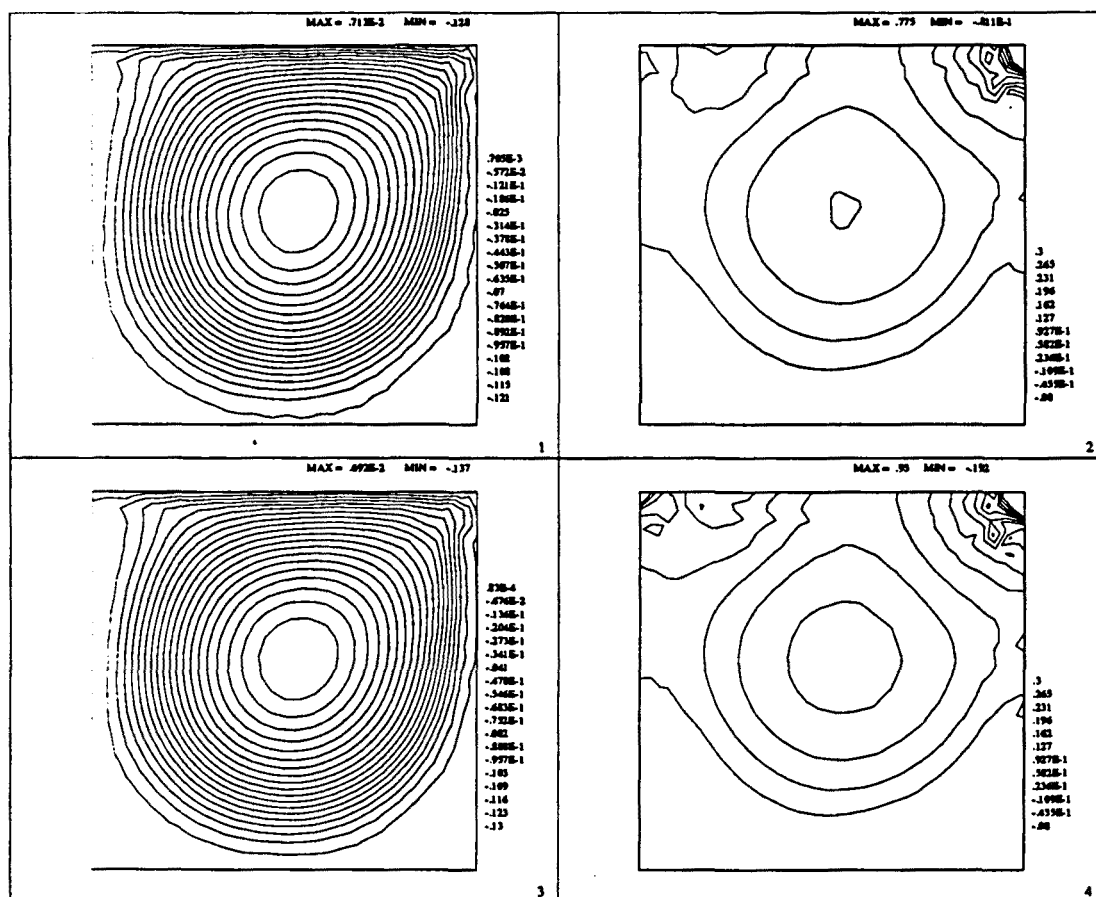


Figure 4.17 Numerical solution of the cavity flow problem at $Re = 1000$ using the P_2^+/P_1 element. (1): Streamlines, Galerkin formulation; (2): Pressure contours, Galerkin formulation; (3): Streamlines, SD method; (4): Pressure contours, SD method.

SD formulations.

The convergence history has been plotted in Figure 4.18.(a). The Picard method has been used for the first two iterations, after which the Newton-Raphson scheme has been employed. It is observed that the Galerkin method yields a quadratic rate of convergence. However, this rate turns from quadratic to linear at iteration number eight if the SD method is used. The evolution of the norm of the discrete divergence is quite peculiar (the same notation as in Chapter 3 has been adopted). From Figure 4.18.(b) it is seen that this norm increases during the first three iterations and then decreases with a rate similar to that of the convergence history. Of course this does not contradict the results of Chapter 3, since what we obtained there was only an error bound for the difference between the penalized solution and the solenoidal one. In Figure 4.18.(b), $\|\mathbf{BU}\|$ has been normalized by dividing it by $N_{tp}^{1/2}$.

The same numerical experiments have been carried out using the Q_2/P_1 pair for $Re = 1000$ and a uniform mesh composed of 12×12 elements (625 nodal points), with $\epsilon = 10^{-4}$. Results are again very similar using the Galerkin and the SD methods (not shown). The convergence history and the evolution of the discrete norm of the velocity

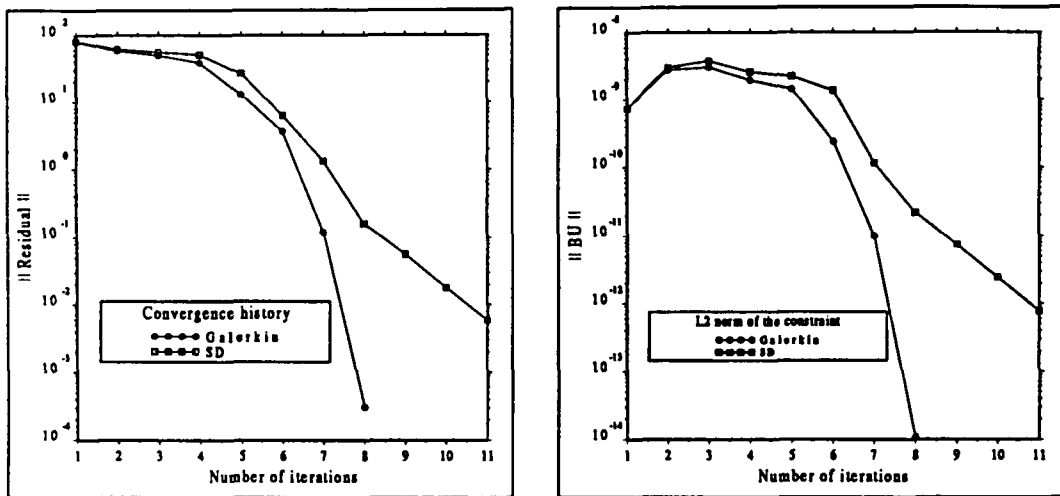


Figure 4.18 Comparison of the convergence of the Galerkin and the SD methods for $Re = 1000$ using the P_2^+/P_1 element, $\epsilon = 10^{-4}$. (1): Convergence history; (2): Norm of the constraint.

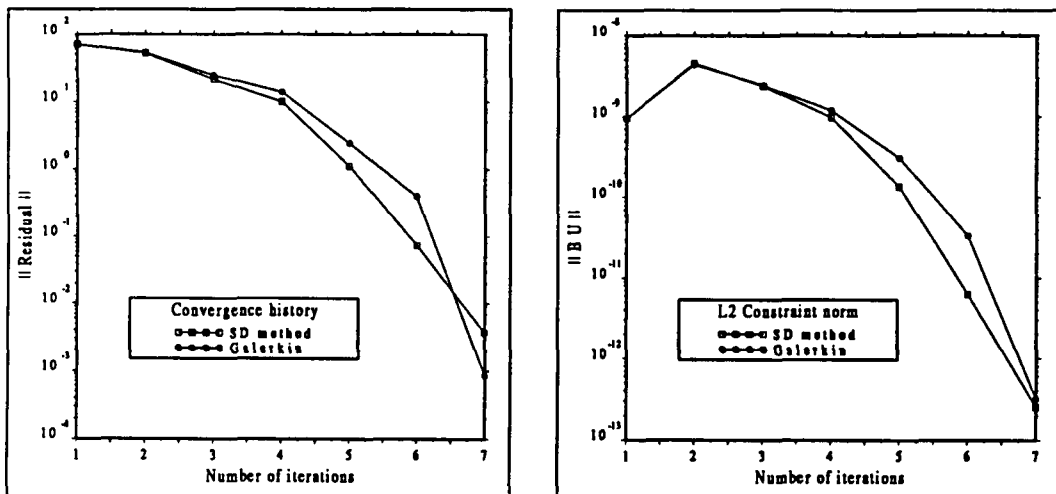


Figure 4.19 Comparison of the convergence of the Galerkin and the SD methods for $Re = 1000$ using the Q_2/P_1 element, $\epsilon = 10^{-4}$. (1): Convergence history; (2): Norm of the constraint.

divergence have been plotted in Figure 4.19. The same general trends as for the P_2^+/P_1 element are observed, although now convergence is faster and the residual using the SD method is smaller than using the Galerkin approach during the first six iterations.

From the all the results obtained for this example, it may be concluded that the SD method fulfils the requirement for which it has been designed: it produces numerical answers without oscillations at high cell Reynolds numbers. Nevertheless, there is a price for it. First, care must be taken in the computation of the intrinsic time in order to avoid overdifusive results, which anyway will be somehow overdamped. Second, the SD operator introduces a high nonlinearity in the problem that may deteriorate the

convergence rate of the Newton-Raphson algorithm.

Example 4.2 Flow over a backward-facing step

The laminar backward-facing step flow is now considered. The problem description is shown in Figure 4.20. Aspect ratio of the backward-facing step (H) to the overall sectional width is 1:2 and the total length in the horizontal direction is $40H$. A fully developed velocity parabolic velocity profile is prescribed at the inlet boundary. Experimental data can be found in Reference [ADS]. A detail of the mesh used in the calculation is shown in Figure 4.21. This mesh is composed of 495 Q_2/P_1 elements and 2077 nodal points.

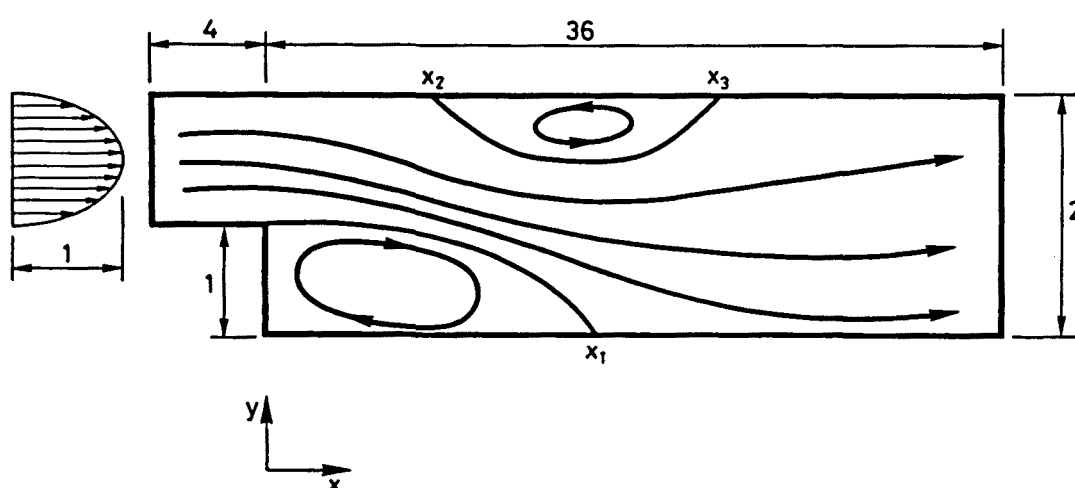


Figure 4.20 Geometry, boundary conditions and nomenclature of backward-facing step problem.

According to Arnali *et al.* [ADS], the Reynolds number will be based on the average value of the inlet velocity profile and the cross-sectional width of the whole domain. For $Re < 500$ there exists only one recirculation zone behind the step. For higher values of Re , another recirculation zone appears at the top wall of the channel. Experimental results indicate that a third recirculation zone appears at the bottom wall for values of Re higher than approximately 1000.

The main feature of this test for the stationary Navier-Stokes equations is the prediction of the vortices as well as the position of the separation and reattachment points (coordinates x_1 , x_2 and x_3 in Figure 4.20). For low values of Re , approximately up to 500, a fairly good agreement exists among the numerical and experimental results that can be found in the literature [ADS], [KM], [Ki], [So]. For $Re > 600$, three-dimensional effects in the experiments are the argued reason for the discrepancies between computational predictions and experimental results [ADS].

Our numerical results agree very well for $Re \leq 600$ with those that can be found in the above mentioned references. For brevity, they have not been included here. We will concentrate only on high values of the Reynolds number. In particular, results will be shown for $Re = 800$ and $Re = 1000$.

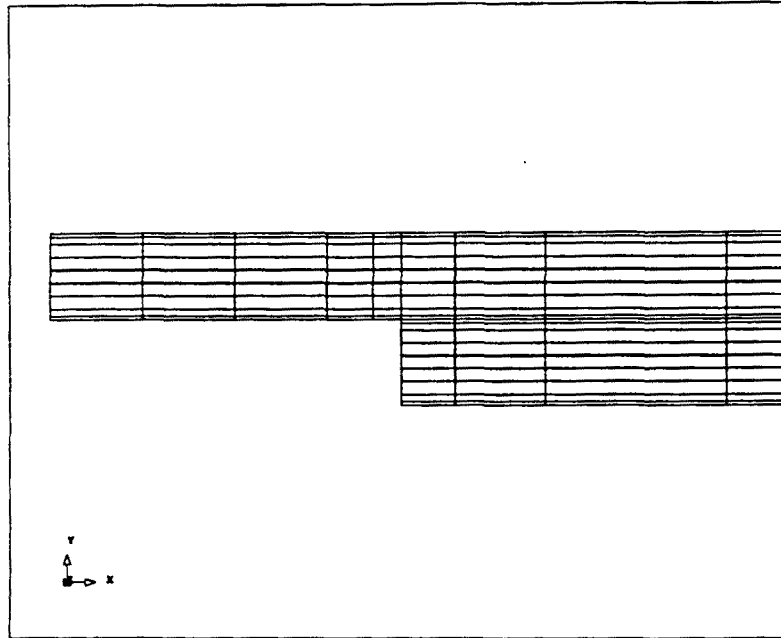


Figure 4.21 Detail of the finite element mesh for the backward-facing step problem (495 Q_2/P_1 elements, 2077 nodal points).

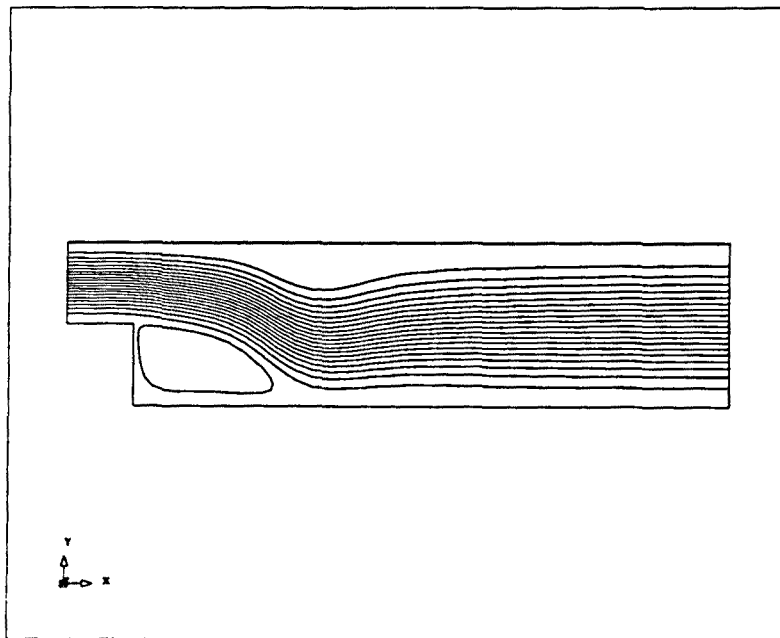


Figure 4.22 General pattern of the streamlines for the backward-facing step problem at $Re = 800$.

The SD method has been used in the calculations, with $\alpha_0 = 0.5$ and $h_0 = 2$. The penalty parameter has been taken as $\epsilon = 10^{-4}$, using the iterative penalization,

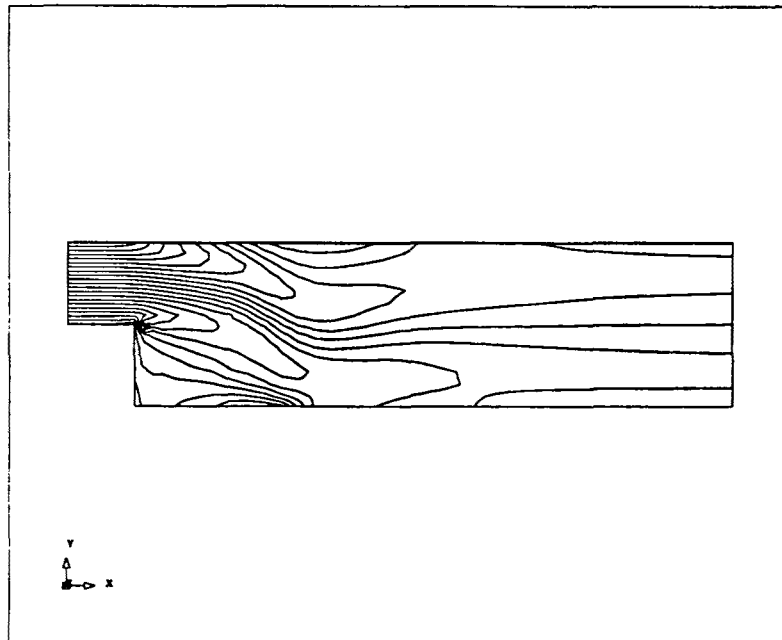


Figure 4.23 Vorticity contours for the backward-facing step problem at $Re = 800$.

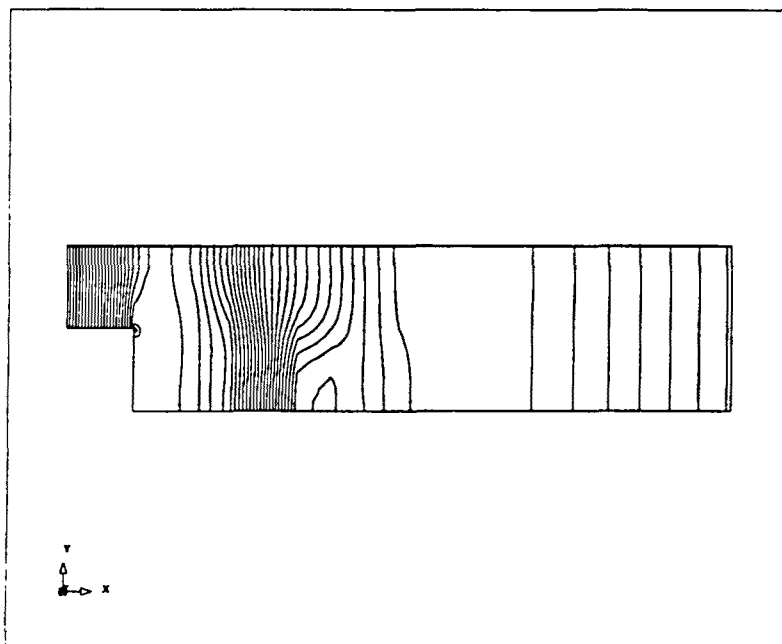


Figure 4.24 Pressure contours for the backward-facing step problem at $Re = 800$.

yielding a final value of order 10^{-14} for the norm of the discrete velocity divergence. For $Re = 100$, the computation has started with zero velocities everywhere. The numerical

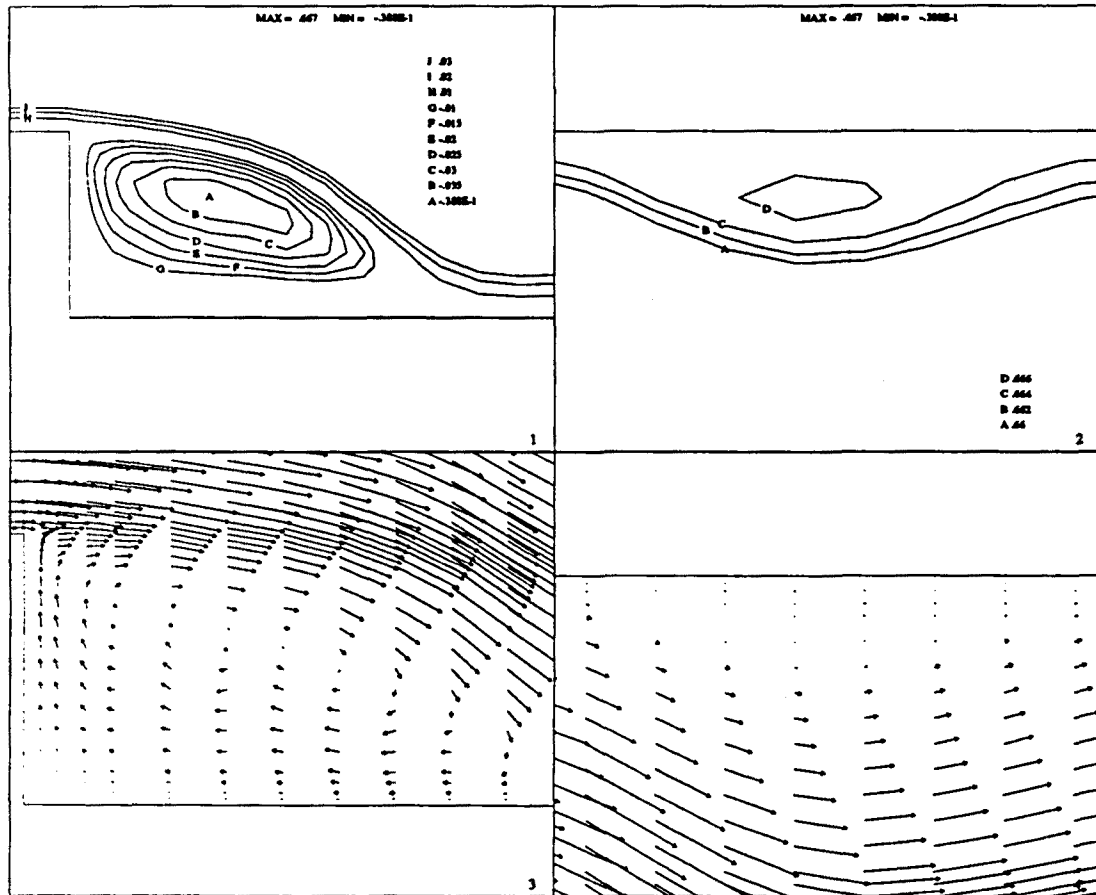


Figure 4.25 Details of the velocity and the streamlines in the recirculation zones for the backward-facing step problem at $Re = 800$. (1): Streamlines behind the step; (2): Streamlines in the recirculation zone at the top wall; (3): Velocity vectors behind the step; (4): Velocity vectors in the recirculation zone at the top wall.

results obtained for this case have been used as the initial guess for $Re = 200$, and the procedure has been repeated until $Re = 1000$. This type of continuation technique has been adopted only because the whole range of Reynolds numbers were to be solved. For $Re = 1000$ we have also tried to reach the stationary solution via the evolution in time, starting from the Stokes flow solution. The convergence towards the steady-state has been found to be extremely slow, and only after a time $t = 207$ the steady-state has been reached. We have used $\Delta t = 0.1$ and the backward Euler scheme ($\theta = 1$), with a single iteration per time step ($TOL = 0.1$ in (4.53)) and a tolerance of 0.1% to check if the steady-state has been reached. This slow evolution towards the stationary solution is due to the pressure waves that are reflected at the outflow boundary, for which the numerical boundary condition chosen is zero traction. Concerning the steady calculations, two Picard iterations and three or four Newton-Raphson iterations have been performed for each Reynolds number increment to reach a convergence tolerance of 0.1%.

Consider first the case $Re = 800$. Figure 4.22 shows the general streamline pattern

(the y -direction has been scaled by a factor of 5 in all the plots). Vorticity and pressure contours are shown in Figures 4.23 and 4.24, respectively. From the last picture, it is observed that the zero traction outflow condition, which must be viewed as an artificial boundary condition to simulate a long channel, does not produce pressure reflexion. The pressure gradient is parallel to the x -direction.

A detail of the streamlines and the velocity vectors in the recirculation zones is shown in Figure 4.25. The vortex behind the step is much stronger than the one in the top wall of the channel. The extreme values of the streamfunction are -3.88×10^{-2} for the first vortex and 6.67×10^{-1} for the second. The values of the coordinates x_i , $i = 1, 2, 3$ in Figure 4.20 are the following:

<u>Coordinate</u>	<u>Experimental</u>	<u>Computed</u>
x_1	14.3	10.3
x_2	10.6	10.8
x_3	19.8	17.2

The given computed values have been obtained from the plots and therefore should be considered only as an approximation. The experimental values correspond to those given in [ADS]. As it has been already said, discrepancies should be expected due to the three-dimensionality of the experimental flow at this Reynolds number.

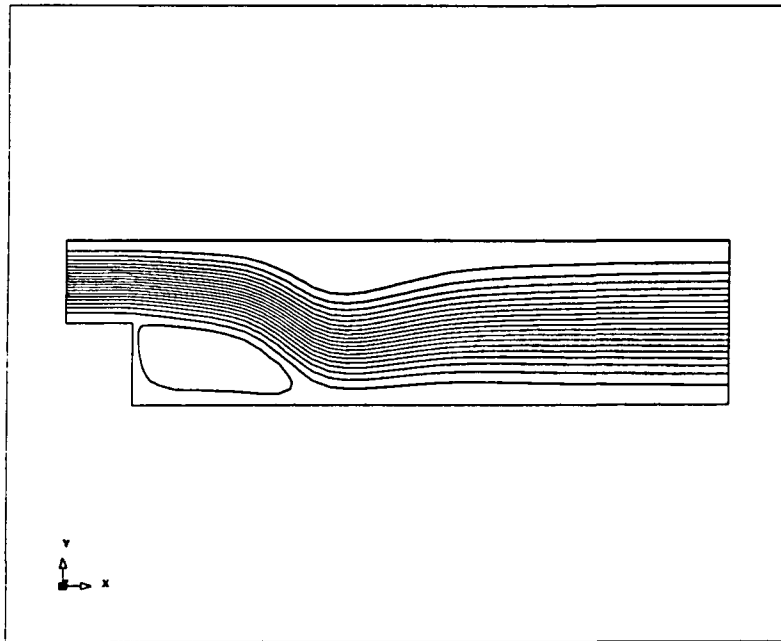


Figure 4.26 General pattern of the streamlines for the backward-facing step problem at $Re = 1000$.

Results for $Re = 1000$ are shown in Figures 4.26 to 4.29. The essential features of the flow are the same as for $Re = 800$, although now accentuated. In Figure 4.28 it is observed that higher pressure gradients develop at the reattachment point behind the cylinder. The detail of the vortices depicted in Figure 4.29 indicate that they are

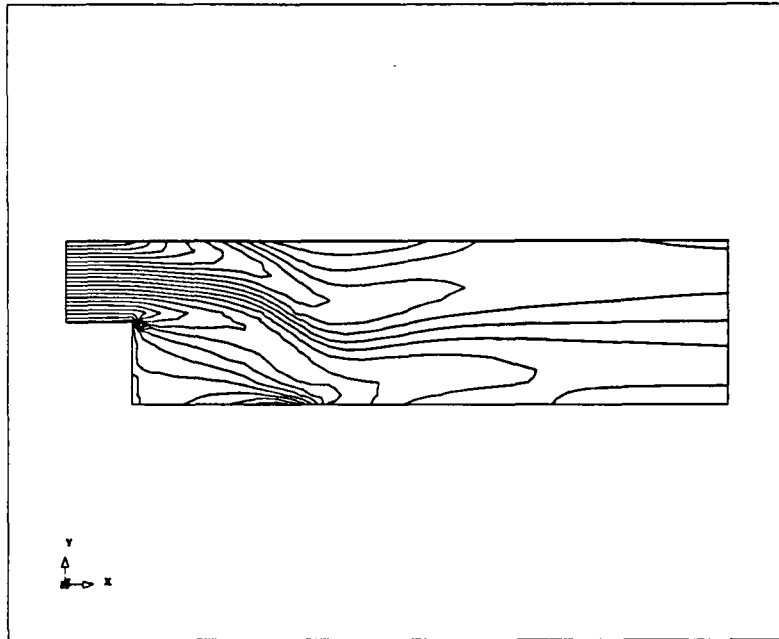


Figure 4.27 Vorticity contours for the backward-facing step problem at $Re = 1000$.

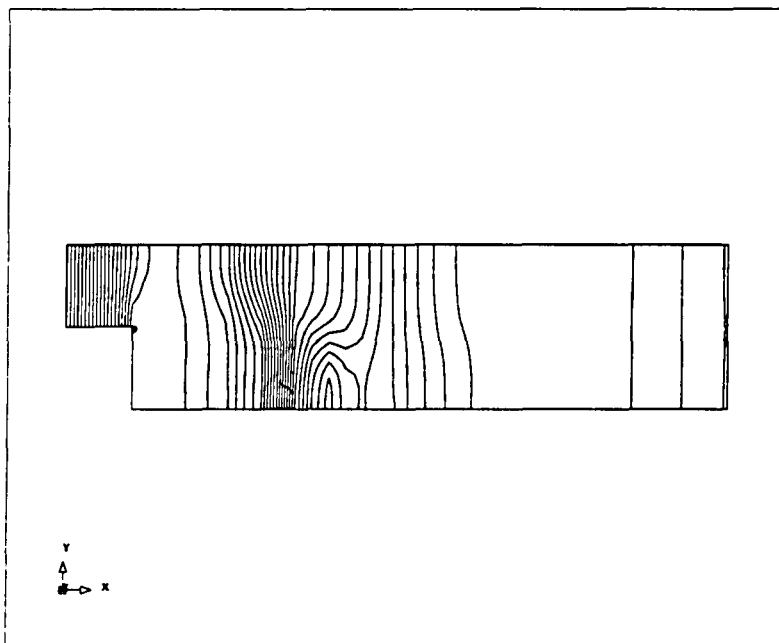


Figure 4.28 Pressure contours for the backward-facing step problem at $Re = 1000$.

stronger now than for $Re = 800$. It is also observed that a third vortex begins to appear at the top wall (Figure 4.29.(2)).

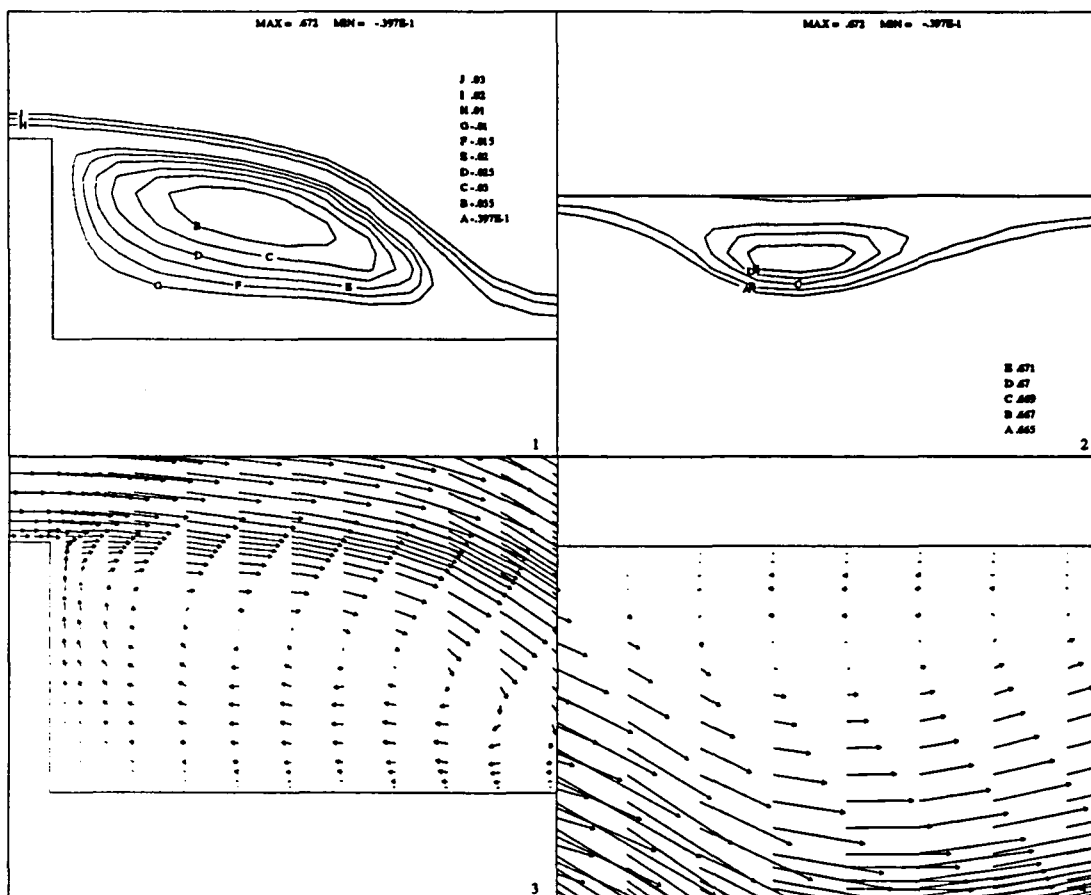


Figure 4.29 Details of the velocity and the streamlines in the recirculation zones for the backward-facing step problem at $Re = 1000$. (1): Streamlines behind the step; (2): Streamlines in the recirculation zone at the top wall; (3): Velocity vectors behind the step; (4): Velocity vectors in the recirculation zone at the top wall.

The approximate values of the coordinates x_i , $i = 1, 2, 3$ are found to be $x_1 = 12.8$, $x_2 = 13.0$ and $x_3 = 22.1$. Although the recirculation zones are now longer than that for the $Re = 800$ case, they are still shorter than the experimental values for this Reynolds number.

Example 4.3 Vortex shedding behind a cylinder

This last example involves the flow past a cylinder, another widely solved benchmark problem. A circular cylinder is immersed in a viscous fluid. The Reynolds number is based on the cylinder diameter and the prescribed uniform inflow velocity. The geometry and boundary conditions are shown in Figure 4.30.

For Re approximately less than 40, two symmetrical eddies develop behind the cylinder. These eddies become unstable at higher Reynolds numbers and periodic vortex shedding occurs, leading to the so called von Karman vortex street. The case $Re = 100$ to be solved here is usually considered as the standard test.

Consider first the stationary (unstable) solution. To show the behavior of the

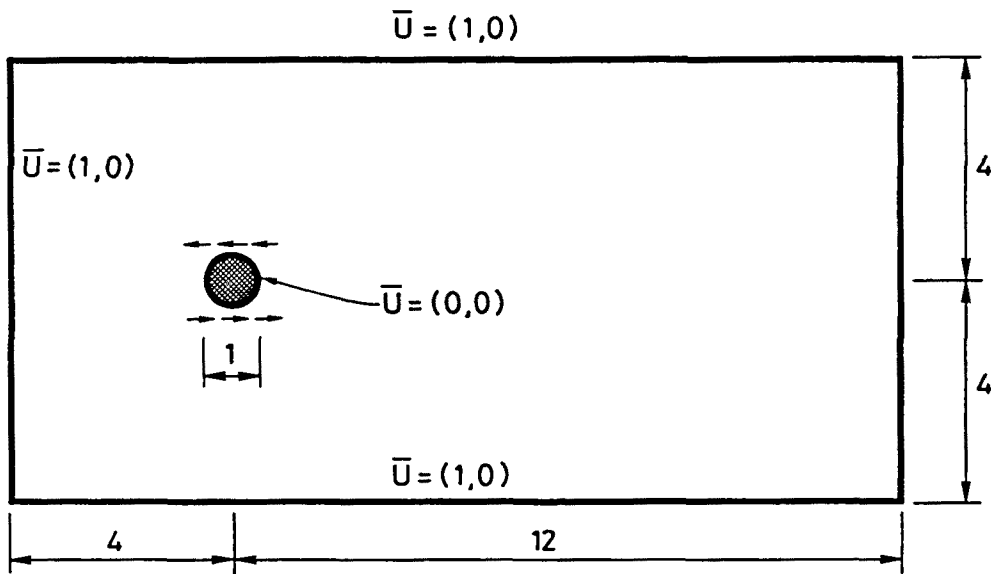


Figure 4.30 Geometry, boundary conditions and initial perturbation for the flow past a cylinder.

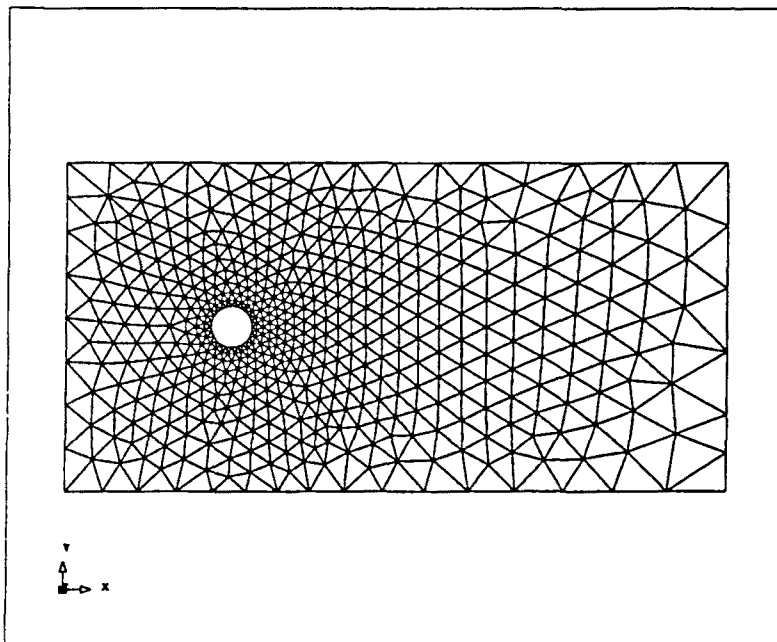


Figure 4.31 Finite element mesh for the flow past a cylinder using the P_2^+/P_1 element (1014 elements, 3112 nodal points).

P_2^+/P_1 pair, we have solved this problem using this element. The finite element mesh shown in Figure 4.31 consists of 1014 elements and 3112 nodal points. The steady calculation has started with zero velocities everywhere. First, two Picard iterations have been performed, after which four more Newton-Raphson iterations have been

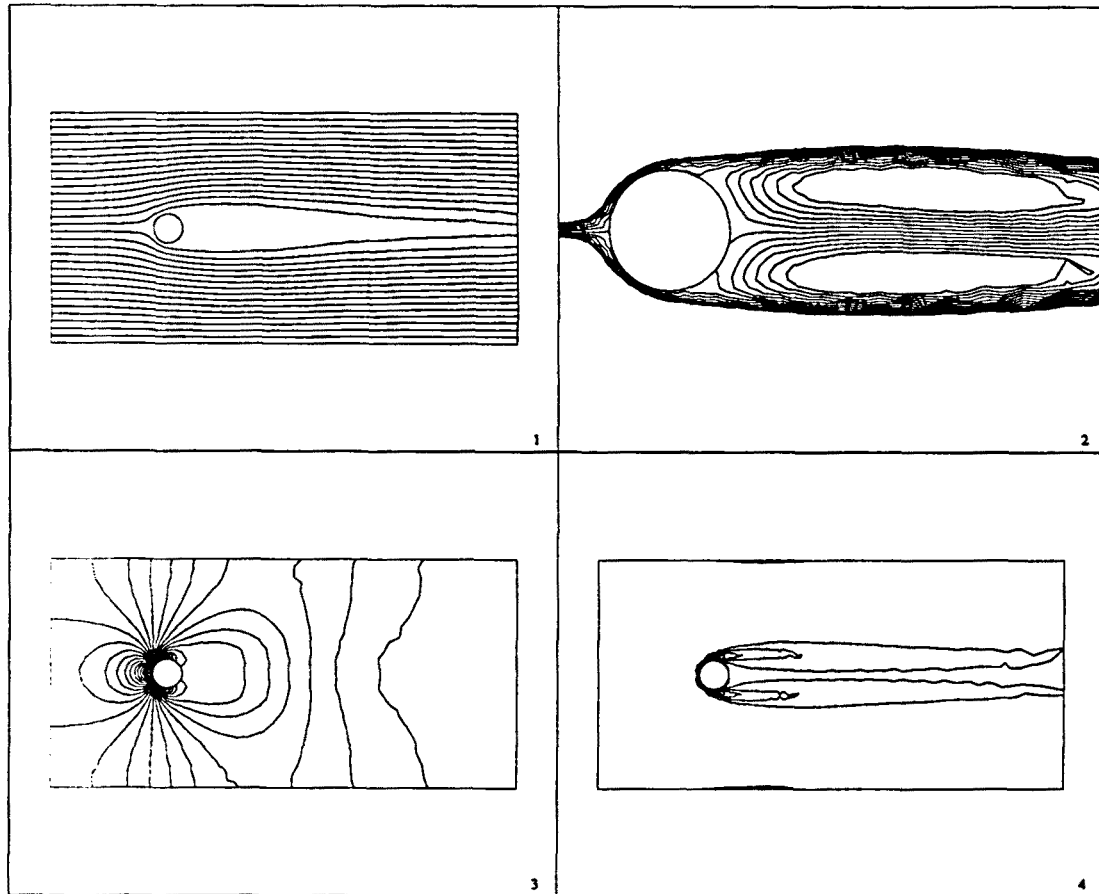


Figure 4.32 Stationary (unstable) solution using the P_2^+/P_1 element. (1): Streamlines; (2): Detail of the symmetrical eddies at the downstream side of the cylinder; (3): Pressure contours; (4): Vorticity contours.

needed to reach a convergence tolerance of 0.1%. The iterative penalty method has been employed, with a penalty parameter $\epsilon = 10^{-3}$. The upwind factor to calculate the intrinsic time has been chosen as $\alpha_0 = 0.5$ (quadratic elements) and the length of the parent domain $h_0 = 0.7$. Results are shown in Figure 4.32.

If the stationary solution is slightly perturbed, the two symmetric eddies disappear and vortex shedding occurs. The numerical simulation of this phenomenon has been carried out using the Q_2/P_1 element and the finite element mesh depicted in Figure 4.33 (500 elements, 2100 nodal points). First, the stationary solution has been obtained (results not shown), with a strategy similar to the previous case. This solution has been perturbed by introducing a small rotating flow field around the cylinder, as shown in Figure 4.30, and taking this as the initial condition for the transient computation.

In order to obtain a fully developed vortex shedding, 90 time steps have been performed with $\Delta t = 1$ (time step size) and $\theta = 0.5$ (Crank-Nicolson scheme), although $\theta = 1$ has been chosen for the first time step. The convergence tolerance within each time step has been taken as 1%. A single Picard iteration has been needed using the classical penalty method with $\epsilon = 10^{-3}$. The parameters of the SD method are

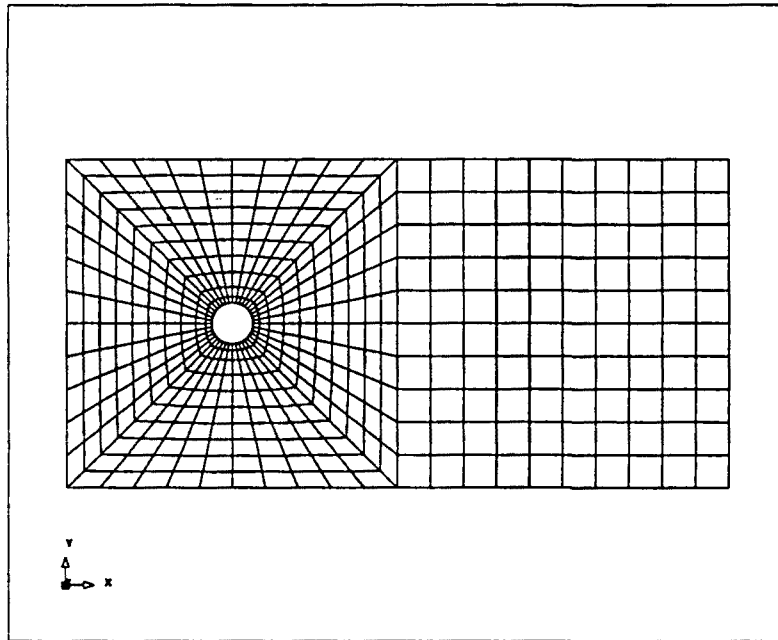


Figure 4.33 Finite element mesh for the flow past a cylinder using the Q_2/P_1 element (500 elements, 2100 nodal points).

$\alpha_0 = 0.5$ and $h_0 = 2$. The solution thus obtained is only a crude approximation, but the computational effort has been relatively low (56 CPU seconds per time step) and the periodic flow pattern obtained is fully developed.

The results obtained using this procedure have been taken as the initial condition for a more accurate calculation. Now, $\Delta t = 0.1$ has been chosen. Two Newton-Raphson iterations coupled with the iterative penalization have been performed for each time step. The initial guess for the first one has been the solution of the previous step. The relative L^2 -norm of the velocity residuals found has been approximately the 2% and the normalized norm of the discrete velocity divergence of order 10^{-6} . After the second iteration, the relative norm of the velocity residuals decreases to the 0.02% and the norm of the discrete velocity divergence to a value of order 10^{-8} . The total CPU time required per time step has been 139 seconds.

Numerical results are shown in Figures 4.34 to 4.40. The period of the oscillations has been found to be 5.7 time units. The values given in references [BH] and [GC2] are 6.0 and 5.6, respectively. In Reference [EJ], the period obtained with a very fine mesh (3426 Q_2/P_1 elements, 14000 nodal points) is 5.8 time units.

The streamline snapshots shown in Figure 4.34 correspond to the times $t = 10, 11, 12$ and 13 , that is, approximately half a period ($t = 0$ corresponds to the periodic solution computed as described earlier with a higher tolerance and a higher time step size). Details of the streamlines and the velocity vectors at the downstream side of the cylinder are plotted in Figures 4.35 and 4.36, respectively. The pressure and vorticity are shown in Figures 4.37–4.38 and 4.39–4.40. In general, all these results agree very well with those that can be found in the literature. Perhaps the only point to be remarked is that the smoothing of the pressure and the vorticity we have employed does not yield very smooth contours, since both fields are highly variable in space due to the transportation of the eddies downstream.

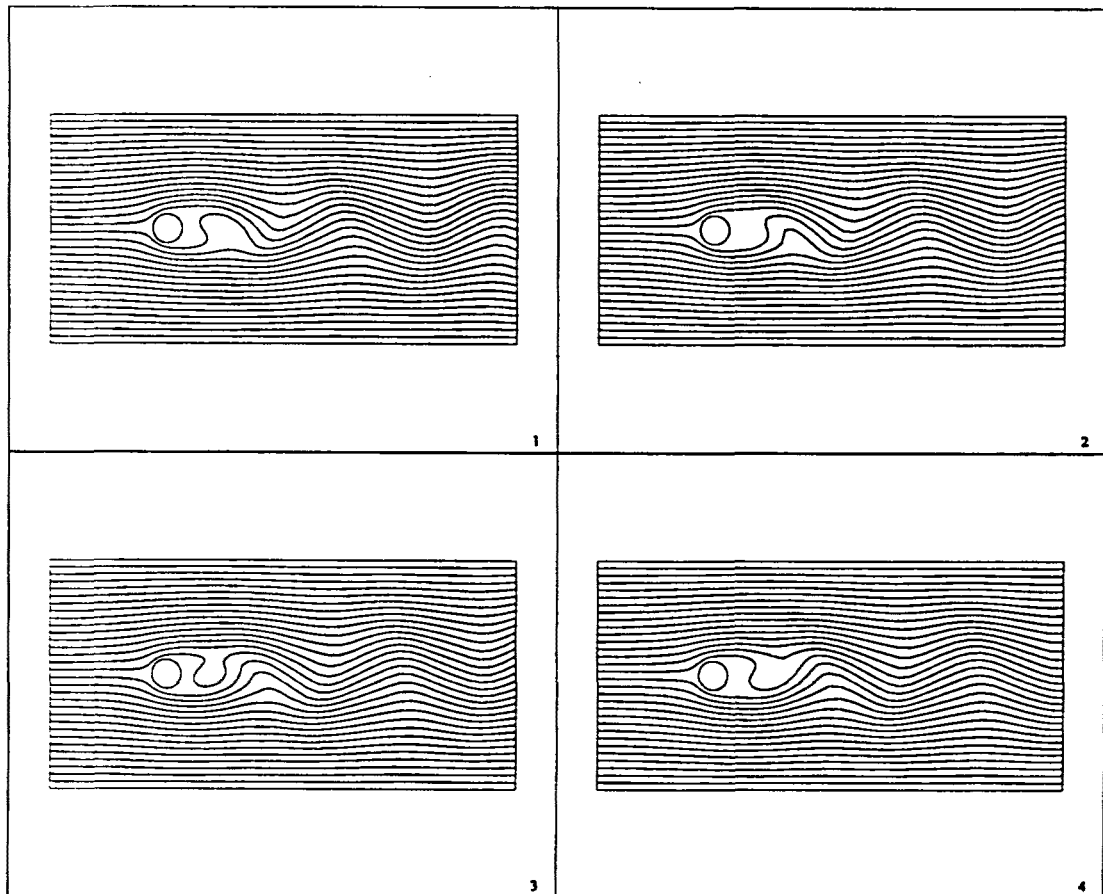


Figure 4.34 Development of vortex shedding: Streamlines. (1): $t = 10.$; (2): $t = 11.$; (3): $t = 12.$; (4): $t = 13.$.

4.9 Summary and conclusions

The finite element method to solve the Navier-Stokes equations proposed in this work has been fully described in this chapter. Most of the ideas developed in the previous chapters have been applied here, although now the purpose has been to present a *methodology* rather than to introduce new developments. In particular, the following items have been treated:

- *Time discretization.* The trapezoidal rule applied to the transient Navier-Stokes equations has been described in detail. Special emphasis has been given to justify, both using theoretical and computational arguments, the choice of the parameter θ of the trapezoidal rule.
- *Streamline Diffusion method.* As for the convection-diffusion equation studied in Chapter 1, a SD term is added to the Galerkin formulation of the Navier-Stokes equations. This term has been designed to avoid the numerical oscillations of the Galerkin approach, but not to stabilize the pressure interpolation. Therefore, the velocity-pressure spaces to be used have to be div-stable. The calculation of the intrinsic time is of fundamental importance, since overdifusive answers are

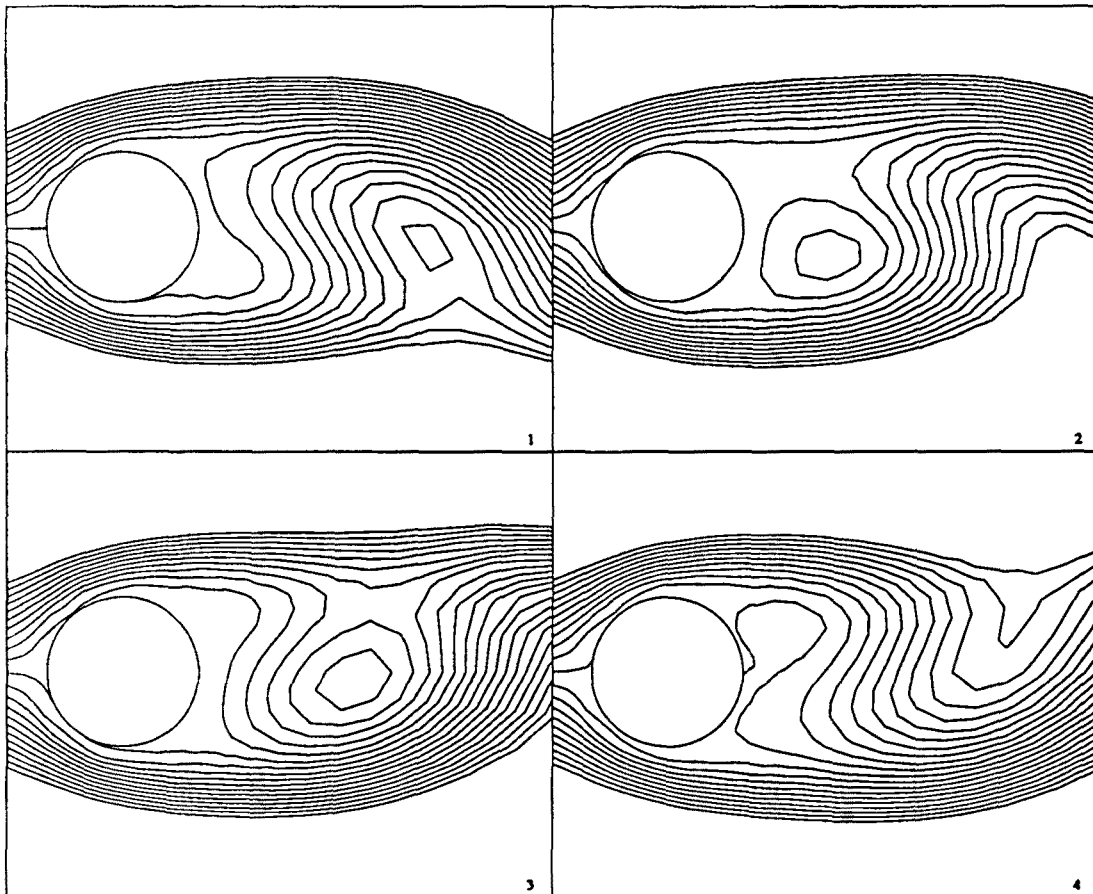


Figure 4.35 Development of vortex shedding: Detail of streamlines. (1): $t = 10.$; (2): $t = 11.$; (3): $t = 12.$; (4): $t = 13.$.

obtained if this parameter is overestimated. The simplest method of those proposed in Chapter 1 for computing the upwind function has proved to be effective. Whenever a converged solution has been obtained, no oscillations have been found and the results compare very well with reference numerical solutions selected from the available literature.

- *Linearization procedures.* The way the final nonlinear system of equations is linearized has been treated in detail. In order to avoid a high computational effort due to the SD method, terms coming from the SD operator have been linearized only up to first order. When the convective term is linearized up to second order, the quadratic rate of convergence that one finds using the Galerkin approach is in general deteriorated, although convergence is still much faster than using the Picard scheme, i.e., first order linearization for the convective term. This is a price to be paid for using the SD method.
- *Iterative penalization.* The iterative penalty method analyzed in Chapter 3 has been extended to the transient equations and used in conjunction with the SD method. We have found that this is certainly worth doing in all the cases. Although it has not been our purpose here to check its behavior, for which the numerical experiments of Chapter 3 were intended, in all the numerical examples

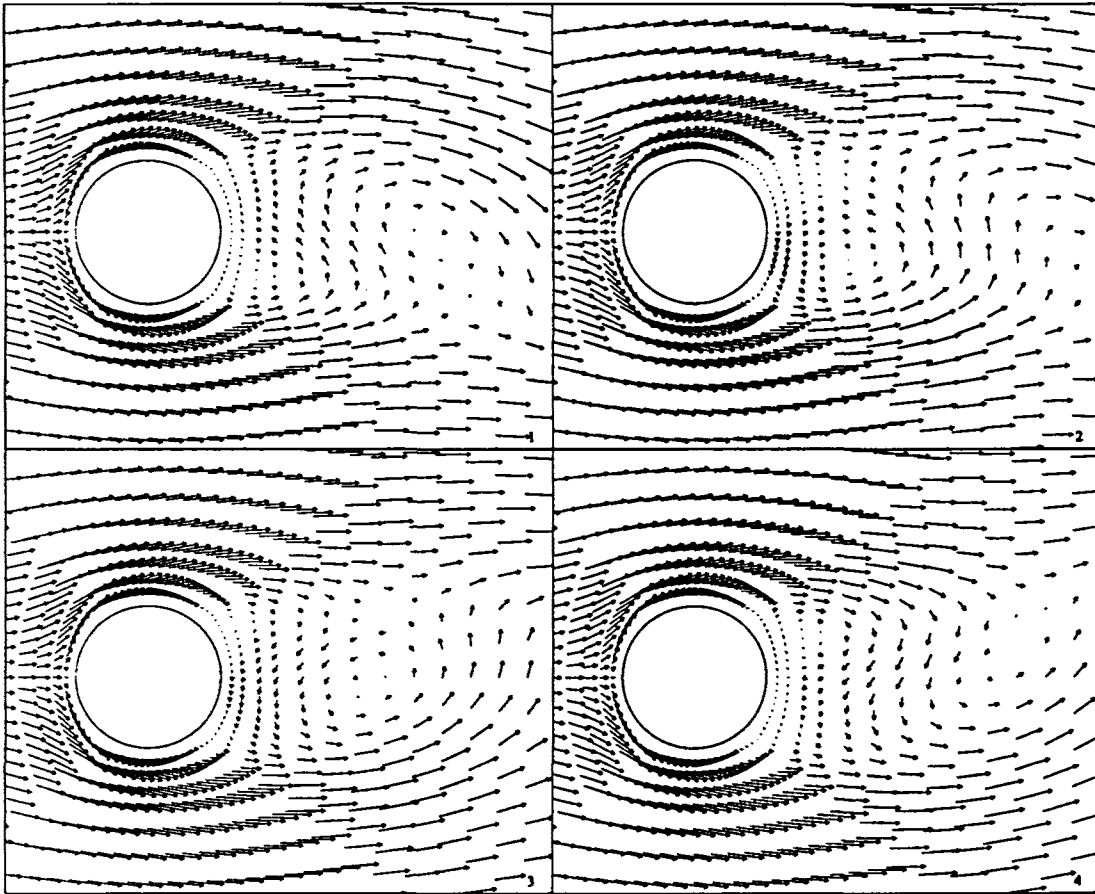


Figure 4.36 Development of vortex shedding: Detail of velocity vectors. (1): $t = 10.$; (2): $t = 11.$; (3): $t = 12.$; (4): $t = 13..$

we have given the penalty parameter and the final value of the norm of the velocity divergence. Results have always been very good, with an approximation of the incompressibility constraint much better than what could be expected using the classical penalty method.

Some specific contributions have also been introduced here. After describing the smoothing technique employed in the calculation of the pressure and the vorticity, nodal quadrature rules have been given for the most common finite elements used in practice, not only those that have been employed here. Finally, an algorithmic procedure to calculate the streamfunction has been presented using the genuine structure of finite element programming.

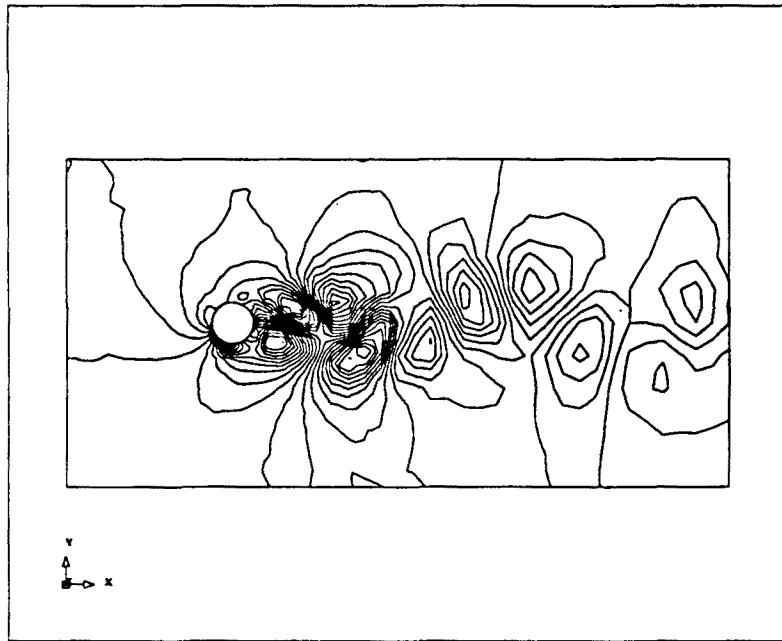


Figure 4.37 Pressure contours at $t = 15$. time units.

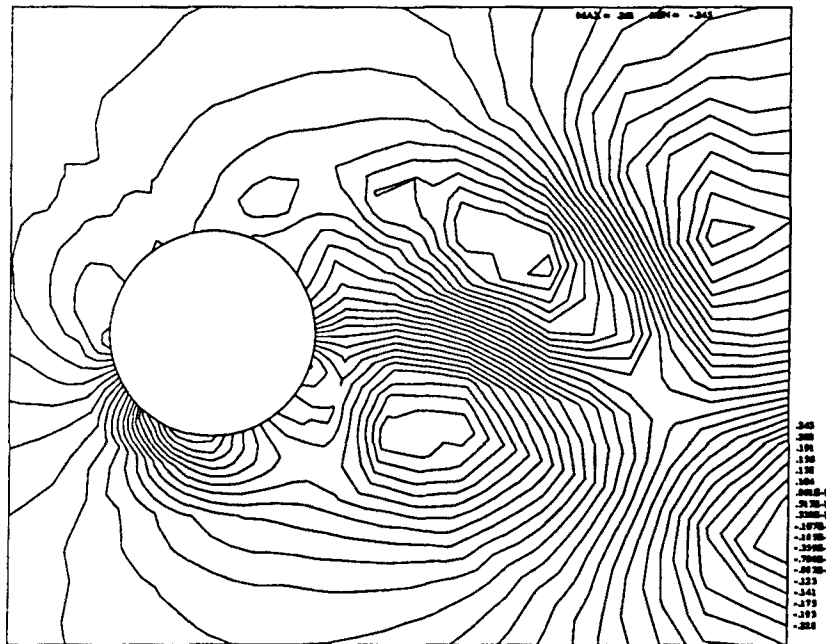


Figure 4.38 Detail of pressure contours at $t = 15$. time units.

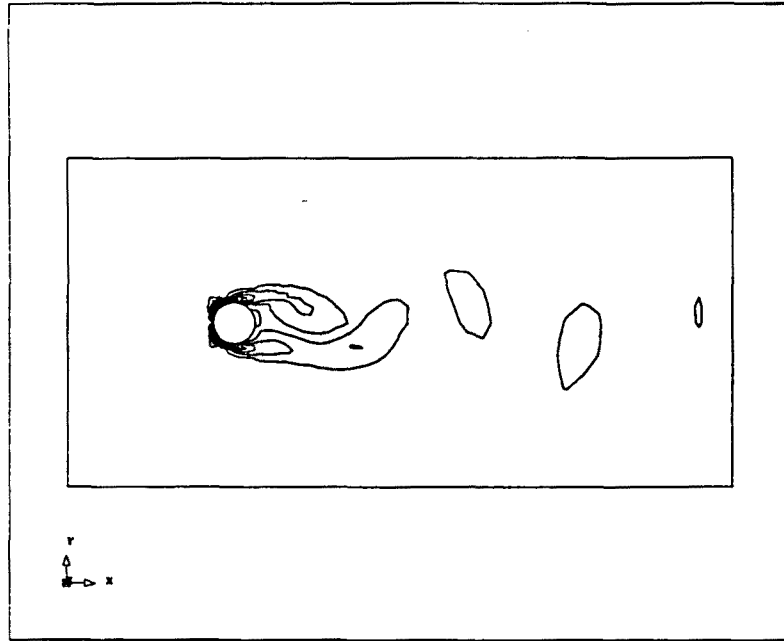


Figure 4.39 Vorticity contours at $t = 15$. time units.

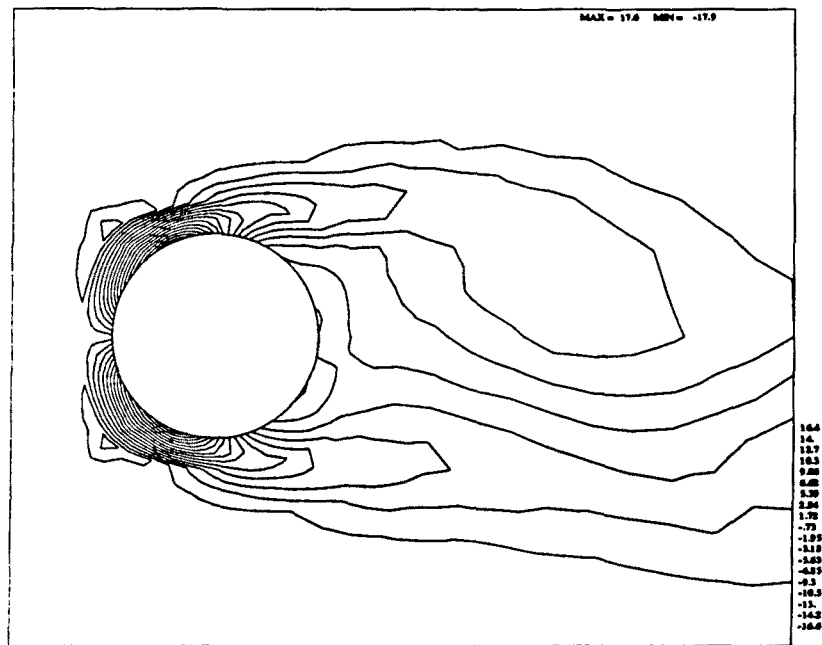


Figure 4.40 Detail of vorticity contours at $t = 15$. time units.

References

- [ADP] J. Argyris, J. St. Doltsinis, P.M. Pimenta and H. Wustenberg. Natural finite element techniques for viscous fluid motion. *Comput. Meth. Appl. Mech. Engrg.*, vol. 45 (1984), 3–55
- [ADS] B.F. Armaly, F. Durst, J.C.F. Pereira and B. Schonung. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.*, vol. 127 (1983), 473–476
- [ABF] D.N. Arnold, F. Brezzi and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, vol. 21 (1984), 337–344
- [BW] R.E. Bank and B.D. Welfert. A comparison between the mini-element and the Petrov-Galerkin formulations for the generalized Stokes problem. *Comput. Meths. Appl. Mech. Engrg.*, vol. 83 (1990), 61–68
- [BR] C. Bernardi and G. Raugel. A conforming finite element method for the time dependent Navier-Stokes equations. *SIAM J. Numer. Anal.*, vol. 22 (1985), 455–473
- [BH] A.N. Brooks and T.J.R. Hughes. Streamline Upwind/Petrov-Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Meths. Appl. Mech. Engrg.*, vol. 32 (1982), 199–259
- [CO] G.F. Carey and J.T. Oden. *Finite Elements: Fluid Mechanics*. The Texas Finite Element Series, vol. VI (Prentice Hall, 1986)
- [Ch] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, vol. 2 (1967), 12–26
- [Ci] P.G. Ciarlet. *The finite element method for elliptic problems* (North-Holland, 1978).
- [CF] P. Constantin and C. Foias. *Navier-Stokes equations* (Chicago Press, 1989).
- [CR] M. Crouzeix and P.A. Raviart. Conforming and non-conforming finite element methods for the stationary Stokes equations. *RAIRO Anal. Numer.*, vol. 7 (1973), 33–76
- [CSS] C. Cuvelier, A. Segal and A. van Steenhoven. *Finite element methods and Navier-Stokes equations* (Reidel, 1986).
- [EJ] M.S. Engelman and M.A. Jamnia. Transient flow past a circular cylinder: A benchmark solution. *Int. J. Numer. Meth. Fluids*, vol. 11 (1990), 985–1000
- [Ga] D.K. Gartling. A test problem for outflow boundary conditions—Flow over a backward-facing step. *Int. J. Numer. Meth. Fluids*, vol. 11 (1990), 953–967
- [GeH] M. Gellert and R. Harbord. Moderate degree cubature formulas for 3D tetrahedral finite-element approximations. *Comm. Appl. Numer. Meth.*, vol. 7 (1991), 487–495
- [Ge] A. Georgescu. *Hydrodynamic stability theory* (Nijhoff, 1985).
- [GGS] U. Ghia, K.N. Ghia and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multi-grid. *J. Comput. Phys.*, vol. 48 (1982), 387–411
- [GR1] V. Girault and P.A. Raviart. *Finite element approximation of the Navier-Stokes equations* (Springer-Verlag, 1979).
- [GR2] V. Girault and P.A. Raviart. An analysis of upwind schemes for the Navier-Stokes equations. *SIAM J. Numer. Anal.*, vol. 19 (1982), 312–333

- [GR3] V. Girault and P.A. Raviart. *Finite element methods for Navier-Stokes equations* (Springer-Verlag, 1986).
- [GC1] P.M. Gresho, S.T. Chan, R.L. Lee and C.D. Upson. A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 1: Theory. *Int. J. Numer. Meth. Fluids*, vol. 4 (1984), 557–598
- [GC2] P.M. Gresho, S.T. Chan, R.L. Lee and C.D. Upson. A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. Part 2: Applications. *Int. J. Numer. Meth. Fluids*, vol. 4 (1984), 619–640
- [Gu] M. Gunzburger. *Finite element methods for viscous incompressible flows* (Academic Press, 1989).
- [GuH] K. Gustafson and K. Halasi. Vortex dynamics of cavity flows. *J. Comput. Phys.*, vol. 64 (1986), 279–319
- [HRS] L.P. Hackman, G.D. Raithby and A.B. Strong. Numerical predictions of flows over backward-facing steps. *Int. J. Numer. Meth. Fluids*, vol. 4 (1984), 711–724
- [HS] P. Hansbo and A. Szepessy. A velocity-pressure streamline diffusion finite element method for the incompressible Navier-Stokes equations. *Comput. Meths. Appl. Mech. Engrg.*, vol. 84 (1990), 175–192
- [HG] R. Harbord and M. Gellert. A simple least-squares method for FE analysis of the Navier-Stokes problem. *Comput. Mech.*, vol. 8 (1991), 19–24
- [HR1] J.G. Heywood and R. Rannacher. Finite element approximation of the non-stationary Navier-Stokes problem. I: Regularity of solutions and second order error estimates for spatial discretization. *SIAM J. Numer. Anal.*, vol. 19 (1982), 275–311
- [HR2] J.G. Heywood and R. Rannacher. Finite element approximation of the non-stationary Navier-Stokes problem. II: Stability of solutions and error estimates uniform in time. *SIAM J. Numer. Anal.*, vol. 23 (1986), 750–777
- [HR3] J.G. Heywood and R. Rannacher. Finite element approximation of the non-stationary Navier-Stokes problem. III: Smoothing property and higher order estimates for spatial discretization. *SIAM J. Numer. Anal.*, vol. 25 (1988), 489–512
- [HR4] J.G. Heywood and R. Rannacher. Finite element approximation of the non-stationary Navier-Stokes problem. IV: Error analysis for second-order time discretization. *SIAM J. Numer. Anal.*, vol. 27 (1990), 353–384
- [Hu1] T.J.R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. In: *Finite elements in Fluids*, vol. 7, R.H. Gallagher, R. Glowinski, P.M. Gresho, J.T. Oden and O.C. Zienkiewicz (eds.) (John Wiley & Sons Ltd., 1987)
- [Hu2] T.J.R. Hughes. *The finite element method. Linear static and dynamic analysis* (Prentice-Hall, 1987).
- [HFB] T.J.R. Hughes, L.P. Franca and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska-Brezzi condition: a stable Petrov-Galerkin formulation for the Stokes problem accommodating equal-order interpolations. *Comput. Meths. Appl. Mech. Engrg.*, vol. 59 (1986), 85–99
- [Ja] Y.J. Jan. A simple program for plotting streamlines and calculating residence

- times. *Comm. Appl. Numer. Meth.*, vol. 4 (1988), 699–707
- [Joh] C. Johnson. The Streamline Diffusion finite element method for compressible and incompressible fluid flow. Von Karman Lecture Series, IV: Computational Fluid Dynamics (March 1990)
- [JS] C. Johnson and J. Saranen. Streamline Diffusion methods for the incompressible Euler and Navier-Stokes equations. *Math. Comput.*, vol. 47 (1986), 1–18
- [Jos] D.D. Joseph. *Stability of fluid motions* (Springer-Verlag, 1976).
- [Ki] S.W. Kim. A finite element computational method for high Reynolds number laminar flows. NASA CR-179135 (1987).
- [KM] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.*, vol. 59 (1985), 308–323
- [LG] M.E. Laursen and M. Gellert. Some criteria for numerically integrated matrices and quadrature formulas for triangles. *Int. J. Numer. Meth. Engrg.*, vol. 12 (1978), 67–76
- [La] O. Ladyzhenskaya. *The mathematical theory of viscous incompressible flow* (Gordon-Breach, 1963).
- [Li] J.L. Lions. *Quelques méthodes de résolution des problèmes aux limites non lineaires* (Dunod, 1968).
- [LR] M. Luskin and R. Rannacher. On the smoothing property of the Galerkin method for parabolic equations. *SIAM J. Numer. Anal.*, vol. 19 (1981), 93–113
- [Pi] O. Pironneau. *Finite element methods for fluid flow* (John Wiley & Sons, 1989).
- [Sa] P.A.B. de Sampaio. A Petrov-Galerkin formulation for the incompressible Navier-Stokes equations using equal order interpolations for velocity and pressure. *Int. J. Numer. Meth. Engrg.*, vol. 31 (1991), 1135–1150
- [SK] R. Schreiber and H.B. Keller. Driven cavity flows by efficient numerical techniques. *J. Comput. Phys.*, vol. 49 (1983), 310–333
- [Sh] J. Shen. Hopf bifurcation of the unsteady regularized driven cavity-flow. *J. Comput. Phys.*, vol. 95 (1991), 228–245
- [So] J.L. Sohn. Evaluation of FIDAP on some classical laminar and turbulent benchmarks. NASA CR NAS8-35918 (1987).
- [SF] G. Strang and G. Fix. *An analysis of the finite element method* (Prentice-Hall, 1973).
- [TR] Ph. Le Tallec and V. Ruas. On the convergence of the bilinear-velocity constant-pressure finite element method in viscous flow. *Comput. Meths. Appl. Mech. Engrg.*, vol. 54 (1986), 235–243
- [Te] R. Temam. *Navier-Stokes equations* (North-Holland, 1984).
- [TGL] T.E. Tezduyar, R. Glowinski and J. Liou. Petrov-Galerkin methods on multiply connected domains for the vorticity-stream function formulation of the incompressible Navier-Stokes equations. *Int. J. Numer. Meth. Fluids*, vol. 8 (1988), 1269–1290
- [TLi] T.E. Tezduyar and J. Liou. Computation of spatially periodic flows based on the vorticity-stream function formulation. *Comput. Meths. Appl. Mech. Engrg.*, vol. 83 (1990), 121–142
- [TMS] T.E. Tezduyar, S. Mittal and R. Shih. Time-accurate incompressible flow computations with quadrilateral velocity-pressure elements. *Comput. Meths. Appl. Mech. Engrg.*, vol. 87 (1991), 363–384

- [TLu] L. Tobiska and G. Lube. A modified streamline-diffusion method for solving the stationary Navier-Stokes equations. *Numer. Math.*, vol. 59 (1991), 13–29
- [Ve] R. Verfürth. Finite element approximation of incompressible Navier-Stokes equations with slip boundary conditions. *Numer. Math.*, vol. 50 (1987), 697–721
- [Zi] O.C. Zienkiewicz. Explicit (or semiexplicit) general algorithm for compressible and incompressible flows with equal finite element interpolation. Department of Structural Mechanics, Chalmers University of Technology, Goteborg, Report 90:5 (1990)
- [ZT] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Fourth Edition, vols. 1 and 2 (McGraw-Hill, 1989)

PART III:

APPLICATIONS

CHAPTER 5

THERMALLY COUPLED FLOWS AND NONLINEAR MATERIALS

5.1 Introduction

The numerical model developed in the previous chapters will be now applied to several problems of physical and engineering interest. In particular, this chapter will be devoted to the numerical simulation of thermally coupled flows and nonlinear materials and the following to the mould filling simulation. The technological interest of these problems will be discussed in detail in the numerical examples that will be presented.

Thermally coupled flows involve the numerical solution of the energy balance equation, together with the momentum and incompressibility equations, and a coupling algorithm between these two problems. Besides the description of the numerical solution procedure for each problem independently, a block iterative technique used to couple them will be discussed. This will be the only new ingredient of the numerical model and will be treated in some detail.

The coupling between the mechanical and the thermal behavior of a fluid may be due basically to two physical effects. First, temperature variations may lead to density gradients whose presence means that gravitational potential energy can be converted into motion through the action of buoyant forces. These density gradients may be also due to concentration differences in mixtures of one or more components, like salt water. Both effects are coupled in some practical physical situations (see [He] and references therein). Here, only density variations due to temperature will be considered. When the fluid is assumed to have a uniform density except for the body force term, one is led to the so called Boussinesq approximation (see, e.g., [LL]), a model of wide applicability in practical problems. The Boussinesq problem will be the subject of Section 5.2.

Thermally coupled flows may also arise because of the variation of some physical properties of the fluid with temperature [IOS], [ZMS], such as the viscosity, the diffusion or the specific heat. The temperature in turn changes with the velocity field due to convection and the dissipation of mechanical work into heat (Joule effect). Usually, this source term in the energy equation is negligible, although it has to be taken into account when highly viscous flows are considered. In particular, this term is fundamental when the flow of viscoplastic materials is studied.

Section 5.3 is concerned with the numerical simulation of generalized Newtonian fluids. This is a particular case of non-Newtonian behavior in which the constitutive law

takes the same expression as for Newtonian materials although the viscosity is allowed to depend on the invariants of the strain rate tensor. The flow of many fluids can be accurately represented by some well known constitutive laws (power-law, Carreau model, etc.) derived from experimental results (see [Ta] for a comprehensive description of this type of fluids). Another important family of constitutive laws of this kind is the one represented by viscoplastic materials when the elastic effects are neglected (flow approach). This rheological behavior is widely used in metal forming processes (see, e.g., papers in [CO], [TWZ]).

The general problem, including material nonlinearity and thermal coupling, is considered in Section 5.4. The basic algorithm of Box 4.3 is completed with the numerical solution of the energy balance equation and the block iterative algorithm.

Some results concerning the analysis of the problems to be considered here will be referred to during the exposition. These analyses are restricted to some simplified problems, but they help to get insight into the numerical problems that may be encountered when dealing with more complicated situations.

The last part of this chapter contains the numerical results obtained for three different model problems, representative of the type of applications that may be treated with the numerical tools described here. The first problem is the simulation of the thermoconvective instability of plane Poiseuille flow heated from below, using the Boussinesq approximation. This model is also used to solve the natural convection of low-Prandtl-number fluids, such as liquid metals. A periodic oscillating flow pattern is encountered when the Grashof number exceeds a critical value. The last example is the 4:1 plane extrusion of a power-law fluid with an exponential-type thermal dependence. Numerical results will help to understand the physics of this problem. Although the simulation of all these flow problems has an inherent interest, emphasis will be placed on the numerical behavior of the finite element model proposed here, trying to demonstrate its potential applications.

5.2 The Boussinesq model

5.2.1 The continuous problem

The Boussinesq approximation is based on several thermodynamical assumptions and an analysis of the relative importance of the thermal effects (see, e.g., [Jo], [LL]). The main hypothesis, based on thermodynamical grounds, is that the density satisfies the following equation of state:

$$\rho = \rho_0[1 - \beta(\vartheta - \vartheta_0)] \quad (5.1)$$

Here and below, the temperature will be denoted by ϑ (not to be confused with the parameter θ of the generalized trapezoidal rule). The parameter β in Eqn. (5.1) is the volume expansion coefficient and subscript naught refers to a reference state.

Once (5.1) is assumed, a dimensional analysis reveals that the density may be taken as constant and equal to ρ_0 for all the terms of the Navier-Stokes and temperature equations except for the body force term of the former. Neglecting the rate of dissipation of mechanical energy in the temperature equation and assuming the physical properties to be constant, the system of partial differential equations we are led to

is the following:

$$\rho_0[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}] - 2\mu \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p = \rho_0 \mathbf{g}[1 - \beta(\vartheta - \vartheta_0)] \quad (5.2)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (5.3)$$

$$\rho_0 c_p [\partial_t \vartheta + (\mathbf{u} \cdot \nabla) \vartheta] - k \Delta \vartheta = 0 \quad (5.4)$$

to be solved in an open bounded domain Ω of $\mathbb{R}^{N,d}$ with some certain initial and boundary conditions. In (5.2)–(5.4), \mathbf{g} is the gravitational acceleration, c_p is the specific heat at constant pressure and k is the thermal conduction coefficient. The rest of the notation has been introduced in the previous chapters. For simplicity, source terms in (5.4) have been omitted, as well as body forces in the Navier-Stokes equations other than gravitational.

Assume now that there is a length scale L and a temperature scale $\delta\vartheta$ inherent to the problem. For example, L may be taken as the diameter of Ω and $\delta\vartheta$ as the temperature difference between two walls of $\partial\Omega$. Given a velocity scale U , the following dimensionless numbers are defined:

$$\begin{aligned} Re &:= \frac{\rho_0 L U}{\mu}, & \text{Reynolds number} \\ Pe &:= \frac{\rho_0 c_p L U}{k}, & \text{Péclet number} \\ Gr &:= \frac{\beta |\mathbf{g}| \rho_0^2 L^3 \delta\vartheta}{\mu^2}, & \text{Grashof number} \\ Pr &:= \frac{c_p \mu}{k}, & \text{Prandtl number} \\ Ra &:= \frac{\beta |\mathbf{g}| c_p \rho_0^2 L^3 \delta\vartheta}{\mu k}, & \text{Rayleigh number} \\ Fr &:= \frac{U^2}{\rho_0 |\mathbf{g}| \delta\vartheta L}, & \text{Froude number} \end{aligned}$$

These numbers are related by

$$Ra = Gr Pr, \quad Fr = Re^2 Gr^{-1}, \quad Re = Pe Pr^{-1} \quad (5.5)$$

If the thermal diffusivity $\kappa := k/c_p \rho_0$ and the kinematic viscosity $\nu := \mu/\rho_0$ are introduced, Pr may be written as $Pr = \nu/\kappa$. Therefore, the Prandtl number is a measure for the similarity of the transport of heat and momentum. The Grashof number is a measure of the relative importance of the buoyancy forces to the viscous forces.

For the definition of the velocity scale U , two cases will be distinguished. First, if the velocity \mathbf{u} is prescribed to a nonzero value on a part of $\partial\Omega$, a characteristic value of the boundary condition may be chosen as U (e.g., the average or maximum prescribed values). This is the so called *forced convection* problem. If time is nondimensionalized using L/U as time scale, equations (5.2)–(5.4) may be written in dimensionless form as follows:

$$\begin{aligned} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} - 2 \frac{1}{Re} \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p &= -\frac{1}{Fr} \vartheta \hat{\mathbf{g}} \\ \nabla \cdot \mathbf{u} &= 0 \\ \partial_t \vartheta + (\mathbf{u} \cdot \nabla) \vartheta - \frac{1}{Pe} \Delta \vartheta &= 0 \end{aligned} \quad (5.6)$$

Here, body force terms of the form $C\mathbf{g}$, with C a constant, have been introduced in ∇p and the resulting pressure nondimensionalized by $\rho_0 U^2$. The vector $\hat{\mathbf{g}}$ in the momentum equation denotes the normalized gravity acceleration vector. No distinction has been made between dimensional and dimensionless variables.

The second case of interest is found when \mathbf{u} is prescribed to zero on $\partial\Omega$, part of which may be left free. In this case, there are two possibilities for choosing the velocity scale U . If we take $U = \mu/L\rho_0$, then $Re = 1$. On the other hand, if $U = \kappa/L$ is chosen, then $Pe = 1$ automatically. Using this last choice, the dimensionless form of equations (5.2)–(5.4) may be written as

$$\begin{aligned} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} - 2Pr \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}) + \nabla p &= -Pr Ra \vartheta \hat{\mathbf{g}} \\ \nabla \cdot \mathbf{u} &= 0 \\ \partial_t \vartheta + (\mathbf{u} \cdot \nabla) \vartheta - \Delta \vartheta &= 0 \end{aligned} \quad (5.7)$$

This case is known as *natural convection*.

Both for the forced convection and the natural convection cases, a stationary (or motionless) solution may exist, whenever it does exist for the uncoupled Navier-Stokes equations. If the temperature field is such that $\nabla \vartheta$ is parallel to $\hat{\mathbf{g}}$ and normal to \mathbf{u} , the velocity and pressure solutions are independent of the temperature. Otherwise, motion is induced by the bouyancy forces. However, the stability of the stationary solution can only be ensured for low values of the Reynolds and Rayleigh numbers [Jo]. As they are increased, bifurcation phenomena occur and stable solutions are no more stationary.

Let Γ be the boundary of the domain Ω , splitted into two sets of disjoint components $\Gamma = \overline{\Gamma_{du}} \cup \overline{\Gamma_{nu}}$ and $\Gamma = \overline{\Gamma_{dt}} \cup \overline{\Gamma_{nt}}$. The type of boundary conditions that will be considered is the same as in the previous chapters for the Navier-Stokes equations and the convection-diffusion equation, that is, (4.3),(4.4) and (2.2),(2.3), respectively. Let \mathbf{n} be the unit vector normal to Γ , $\bar{\mathbf{u}}$ the velocity prescribed on Γ_{du} , $\bar{\mathbf{t}}$ the prescribed traction on Γ_{nu} , $\bar{\vartheta}$ the given temperature on Γ_{dt} and $\bar{\varphi}$ the prescribed heat flux on Γ_{nt} . The boundary conditions to be considered are

$$\begin{aligned} \mathbf{u} &= \bar{\mathbf{u}} & \text{on } \Gamma_{du} \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} & \text{on } \Gamma_{nu} \\ \vartheta &= \bar{\vartheta} & \text{on } \Gamma_{dt} \\ -k\mathbf{n} \cdot \nabla \vartheta &= \bar{\varphi} & \text{on } \Gamma_{nt} \end{aligned} \quad (5.8)$$

where the expression of the stress tensor $\boldsymbol{\sigma}$ is given by (4.6).

The notation used heretofore to indicate the spaces of test functions and of trial solutions will be slightly modified. Subscripts u , p and t will be used to refer to velocity, pressure and temperature, respectively.

If $(0, T)$ denotes the time interval where the problem is to be solved and $\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})$, $\vartheta(\mathbf{x}, 0) = \vartheta_0(\mathbf{x})$ are the initial conditions ($\mathbf{x} \in \Omega$), the spaces of trial solutions that will be needed are

$$\begin{aligned} V_u &= \{\mathbf{v} \in L^2(0, T; H^1(\Omega)^{N \times d}) \mid \mathbf{v}|_{\Gamma_{du}} = \bar{\mathbf{u}}, \quad t \in (0, T)\} \\ V_p &= \{q \in L^2(0, T; L^2(\Omega)) \mid \int_{\Omega} q d\Omega = 0, \quad t \in (0, T), \text{ if } \Gamma_{nu} = \emptyset\} \\ V_t &= \{\eta \in L^2(0, T; H^1(\Omega)) \mid \eta|_{\Gamma_{dt}} = \bar{\vartheta}, \quad t \in (0, T)\} \end{aligned} \quad (5.9)$$

The corresponding spaces of test functions are

$$\begin{aligned} W_u &= \{\mathbf{v} \in H^1(\Omega)^{N,d} \mid \mathbf{v}|_{\Gamma_{ds}} = \mathbf{0}\} \\ W_p &= L^2(\Omega) \\ W_t &= \{\eta \in H^1(\Omega) \mid \eta|_{\Gamma_{ds}} = 0\} \end{aligned} \quad (5.10)$$

The reason for choosing these spaces has already been explained in Chapters 2 and 4.

In order to write the weak form of problem (5.6) with the dimensionless form of the boundary conditions (5.8), let us introduce the multilinear forms

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &= \frac{2}{Re} \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega, \\ b(q, \mathbf{v}) &= \int_{\Omega} q \nabla \cdot \mathbf{v} d\Omega, \\ c(\mathbf{u}, \mathbf{v}, \mathbf{w}) &= \int_{\Omega} [(\mathbf{u} \cdot \nabla) \mathbf{v}] \cdot \mathbf{w} d\Omega + \frac{1}{2} \int_{\Omega} (\nabla \cdot \mathbf{u}) \mathbf{v} \cdot \mathbf{w} d\Omega, \\ d(\eta, \mathbf{v}) &= \frac{1}{Fr} \int_{\Omega} \eta \hat{\mathbf{g}} \cdot \mathbf{v} d\Omega, \\ e(\vartheta, \eta) &= \frac{1}{Pe} \int_{\Omega} \nabla \vartheta \cdot \nabla \eta d\Omega, \\ f(\mathbf{u}, \vartheta, \eta) &= \int_{\Omega} \eta \mathbf{u} \cdot \nabla \vartheta d\Omega, \\ l_u(\mathbf{v}) &= \int_{\Gamma_{ns}} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma, \\ l_t(\eta) &= \int_{\Gamma_{ns}} \bar{\varphi} \eta d\Gamma, \end{aligned} \quad (5.11)$$

where $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V_u$ or W_u , $q \in V_p$ or W_p and $\vartheta, \eta \in V_t$ or W_t . The weak formulation of the problem is now given as follows: Find $\mathbf{u} \in V_u$, $p \in V_p$ and $\vartheta \in V_t$ such that

$$(\partial_t \mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) + d(\vartheta, \mathbf{v}) = l_u(\mathbf{v}) \quad (5.12)$$

$$b(q, \mathbf{u}) = 0 \quad (5.13)$$

$$(\partial_t \vartheta, \eta) + f(\mathbf{u}, \vartheta, \eta) + e(\vartheta, \eta) = l_t(\eta) \quad (5.14)$$

$$(\mathbf{u}(\mathbf{x}, 0), \mathbf{v}) = (\mathbf{u}_0(\mathbf{x}), \mathbf{v}) \quad (5.15)$$

$$(\vartheta(\mathbf{x}, 0), \eta) = (\vartheta_0(\mathbf{x}), \eta) \quad (5.16)$$

for all $\mathbf{v} \in W_u$, $q \in W_p$ and $\eta \in W_t$ and for $t \in (0, T)$.

Some partial results concerning the existence, uniqueness and regularity of solutions for problem (5.12)–(5.16) are known. For the stationary problem and natural convection, uniqueness of solution can only be proved for sufficiently small values of the Rayleigh number [Li], [Cu] (the changes to be introduced in the dimensionless parameters of (5.11) to consider the natural convection problem (5.7) are obvious).

5.2.2 Discretization in space and time

The numerical solution of problem (5.12)–(5.16) will be carried out using the same techniques as in Chapter 4 for the transient Navier-Stokes equations. The space discretization will be performed using div-stable velocity-pressure finite element interpolations

and the Streamline Diffusion (SD) method to stabilize high-Reynolds-number flows. The finite element space for the temperature will consist of piecewise polynomials of the same degree and with respect to the same finite element partition $\{\Omega^e\}$ as for the velocity components. In doing so, the same convergence rate for the temperature as for the velocity can be expected, at least for the Galerkin approach (cf. [Gu]). The SD method will also be used for the energy equation.

Once space has been discretized, the resulting initial-value problem will be solved using the generalized trapezoidal rule.

As usual, subscript h will be used to denote the discrete finite element spaces and the functions belonging to them.

Using the same notation and arguments that led us to problem (4.40), the fully discrete version of problem (5.12)–(5.16) that will be used reads as follows:

For $n = 1, 2, \dots, N$, given $\mathbf{u}_h^{n-1}(\mathbf{x})$, $p_h^{n-1}(\mathbf{x})$ and $\vartheta_h^{n-1}(\mathbf{x})$, find $\mathbf{u}_h^n(\mathbf{x})$, $p_h^n(\mathbf{x})$ and $\vartheta_h^n(\mathbf{x})$ such that

$$\begin{aligned} \frac{1}{\Delta t}(\mathbf{u}_h^n - \mathbf{u}_h^{n-1}, \mathbf{v}_h) + \theta c(\mathbf{u}_h^n, \mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta)c(\mathbf{u}_h^{n-1}, \mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ + \theta a(\mathbf{u}_h^n, \mathbf{v}_h) + (1 - \theta)a(\mathbf{u}_h^{n-1}, \mathbf{v}_h) \\ - \theta b(p_h^n, \mathbf{v}_h) - (1 - \theta)b(p_h^{n-1}, \mathbf{v}_h) \\ + \theta d(\vartheta_h^n, \mathbf{v}_h) + (1 - \theta)d(\vartheta_h^{n-1}, \mathbf{v}_h) \\ + \sum_{e=1}^{N_{el}} \mathcal{S}_u^{n,e}(\mathbf{u}_h, p_h, \vartheta_h; \mathbf{v}_h) \\ = \theta l_u^n(\mathbf{v}_h) + (1 - \theta)l_u^{n-1}(\mathbf{v}_h) \end{aligned} \quad (5.17)$$

$$b(q_h, \mathbf{u}_h^n) = 0 \quad (5.18)$$

$$\begin{aligned} \frac{1}{\Delta t}(\vartheta_h^n - \vartheta_h^{n-1}, \eta_h) + \theta f(\mathbf{u}_h^n, \vartheta_h^n, \eta_h) + (1 - \theta)f(\mathbf{u}_h^{n-1}, \vartheta_h^{n-1}, \eta_h) \\ + \theta e(\vartheta_h^n, \eta_h) + (1 - \theta)e(\vartheta_h^{n-1}, \eta_h) \\ + \sum_{e=1}^{N_{el}} \mathcal{S}_t^{n,e}(\mathbf{u}_h, \vartheta_h; \eta_h) \\ = \theta l_t^n(\eta_h) + (1 - \theta)l_t^{n-1}(\eta_h) \end{aligned} \quad (5.19)$$

for all $\mathbf{v}_h \in W_{u,h}$, $q_h \in W_{p,h}$ and $\eta_h \in W_{t,h}$.

The Streamline Diffusion term $\mathcal{S}_u^{n,e}(\mathbf{u}_h, p_h, \vartheta_h; \mathbf{v}_h)$ for the momentum equations is defined as

$$\begin{aligned} \mathcal{S}_u^{n,e}(\mathbf{u}_h, p_h, \vartheta_h; \mathbf{v}_h) := \int_{\Omega^e} \zeta_u(\mathbf{u}_h^n, \mathbf{v}_h) \cdot [\mathcal{N}_\theta^n(\mathbf{u}_h, p_h) \\ + \frac{1}{Fr} \hat{\mathbf{g}} (\theta \vartheta_h^n + (1 - \theta) \vartheta_h^{n-1})] d\Omega \end{aligned} \quad (5.20)$$

where

$$\begin{aligned} \mathcal{N}_\theta^n(\mathbf{u}_h, p_h) := \frac{1}{\Delta t}(\mathbf{u}_h^n - \mathbf{u}_h^{n-1}) + \rho \theta (\mathbf{u}_h^n \cdot \nabla) \mathbf{u}_h^n + \rho (1 - \theta) (\mathbf{u}_h^{n-1} \cdot \nabla) \mathbf{u}_h^{n-1} \\ - \frac{1}{Re} \theta \Delta \mathbf{u}_h^n - \frac{1}{Re} (1 - \theta) \Delta \mathbf{u}_h^{n-1} + \theta \nabla p_h^n + (1 - \theta) \nabla p_h^{n-1} \end{aligned} \quad (5.21)$$

$$\zeta_{\mathbf{u}}(\mathbf{u}_h, \mathbf{v}_h) := \tau_{\mathbf{u}}^e(\mathbf{u}_h \cdot \nabla) \mathbf{v}_h \quad (5.22)$$

The intrinsic time $\tau_{\mathbf{u}}^e$ is computed as in Chapter 4 (Eqn. (4.43)) using the cell Reynolds number, now given by $(Re)^e = |\mathbf{u}^e| h^e Re/2$ (recall that \mathbf{u}^e and h^e are assumed to be dimensionless).

For the energy equation, the SD term is

$$\mathcal{S}_t^{n,e}(\mathbf{u}_h, \vartheta_h; \eta_h) := \int_{\Omega^e} \zeta_t(\mathbf{u}_h^n, \eta_h) \mathcal{E}_\theta^n(\mathbf{u}_h, \vartheta_h) d\Omega \quad (5.23)$$

where

$$\begin{aligned} \mathcal{E}_\theta^n(\mathbf{u}_h, \vartheta_h) := & \frac{1}{\Delta t}(\vartheta_h^n - \vartheta_h^{n-1}) + \theta(\mathbf{u}_h^n \cdot \nabla) \vartheta_h^n + (1 - \theta)(\mathbf{u}_h^{n-1} \cdot \nabla) \vartheta_h^{n-1} \\ & - \frac{1}{Pe} \theta \Delta \vartheta_h^n - \frac{1}{Pe} (1 - \theta) \Delta \vartheta_h^{n-1} \end{aligned} \quad (5.24)$$

$$\zeta_t(\mathbf{u}_h, \eta_h) := \tau_t^e(\mathbf{u}_h \cdot \nabla) \eta_h \quad (5.25)$$

and the intrinsic time τ_t^e is computed as explained in Chapter 1 for the convection-diffusion equation using the cell Péclet number $\gamma := |\mathbf{u}^e| h^e Pe/2$.

Remarks 5.1

- (1) The observations pointed out in Remarks 4.1 and 4.2 also apply to the problem now considered.
- (2) In Reference [He], it is concluded that the consistent Streamline Diffusion method does not work for a problem very similar to the present one using the Q_1/P_0 element. The misbehavior found was overcome dropping the buoyancy forces in (5.20) and the discretized version of the velocity time derivative in (5.21) (the viscous and pressure terms vanish for the Q_1/P_0 element). As explained in Chapter 1, this is equivalent to introduce an artificial diffusion along the streamlines. We have not encountered these problems. The answer we give is that the Q_1/P_0 element is not div-stable. Pressure gradients in (5.21) do have an important role and for the Q_1/P_0 element they do not approximate the gradients of the continuous pressure field.
- (3) An analysis of the Galerkin finite element solution of the natural convection problem can be found in Reference [BL], where a slightly different formulation of the physical problem is considered. The fluid-filled domain is linked through an interface with heat conduction in the solid enclosing the fluid. Optimal rates of convergence are proved when the velocity-pressure interpolation consists of non-conforming P_1/P_0 elements and $V_{t,h}$ is built up using P_1 elements with respect to the same triangulation as for the velocity. \square

Before going any further, let us introduce the matrix version of problem (5.17)–(5.19). The matrices defined in Box 4.2 for the Navier-Stokes equations will also be used now (taking $\rho = 1$, $\mu = 1/Re$, $l = l_{\mathbf{u}}$, $\mathbf{f} = \mathbf{0}$ and replacing the vector \mathbf{s} by $\mathbf{s}_{\mathbf{u}} := \tau_{\mathbf{u}}^e \mathbf{u}^e$, \mathbf{u}^e being the characteristic velocity for element e). The new matrices that will be needed to account for the thermal coupling are defined in Box 5.1, where $\mathbf{s}_t := \tau_t^e \mathbf{u}^e$, Θ denotes the vector of nodal values of a generic function in V_t and Θ the vector of nodal values of an element in W_t . The vector of nodal values of an element in the velocity test function space $W_{\mathbf{u}}$ has been represented by \mathbf{V} . The L^2 inner product in the temperature space has been indicated by $(\cdot, \cdot)_t$. In order to avoid the introduction of more subscripts in the matrices, the convection and diffusion matrices for the temperature have been denoted by \mathbf{H} , instead of the traditional notation \mathbf{K} , already used for the matrices of the Navier-Stokes equations.

Box 5.1 Matrix form of the discrete equations

<u>Matrix version</u>	<u>Terms from where it comes</u>
$\mathbf{V}^T \cdot \mathbf{C}_{s_u} \cdot \Theta$	$d(\vartheta_h, \mathbf{v}_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s}_u \cdot \nabla) \mathbf{v}_h] \cdot \hat{\mathbf{g}} \frac{1}{F_r} \vartheta_h d\Omega$
$\bar{\Theta}^T \cdot \mathbf{M}_{t,s_t} \cdot \Theta$	$(\vartheta_h, \eta_h)_t + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s}_t \cdot \nabla) \eta_h] \vartheta_h d\Omega$
$\bar{\Theta}^T \cdot \mathbf{H}_{c,s_t}(\mathbf{U}) \cdot \Theta$	$f(\mathbf{u}_h, \vartheta_h, \eta_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s}_t \cdot \nabla) \eta_h] (\mathbf{u}_h \cdot \nabla) \vartheta_h d\Omega$
$\bar{\Theta}^T \cdot \mathbf{H}_{d,s_t} \cdot \Theta$	$e(\vartheta_h, \eta_h) + \sum_{e=1}^{N_{el}} \int_{\Omega^e} [(\mathbf{s}_t \cdot \nabla) \eta_h] \left(-\frac{1}{P_e} \Delta \vartheta_h \right) d\Omega$
$\bar{\Theta}^T \cdot \mathbf{F}_t$	$l_t(\eta_h)$

Having introduced these matrices and vectors, problem (5.17)–(5.19) may be written as follows:

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} , \mathbf{P}^{n-1} and Θ^{n-1} , find \mathbf{U}^n , \mathbf{P}^n and Θ^n , approximations to $\mathbf{U}(t^n)$, $\mathbf{P}(t^n)$ and $\Theta(t^n)$, such that

$$\begin{aligned}
 & \mathbf{M}_{v,s_u} \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{c,s_u}(\mathbf{U}^n) \cdot \mathbf{U}^n + \theta \Delta t \mathbf{K}_{d,s_u} \cdot \mathbf{U}^n \\
 & \quad - \theta \Delta t \mathbf{G}_{s_u} \cdot \mathbf{P}^n + \theta \Delta t \mathbf{C}_{s_u} \cdot \Theta^n \\
 & = \theta \Delta t \mathbf{F}_{v,s_u}^n + (1 - \theta) \Delta t \mathbf{F}_{v,s_u}^{n-1} + \mathbf{M}_{v,s_u} \cdot \mathbf{U}^{n-1} \\
 & \quad - (1 - \theta) \Delta t \mathbf{K}_{c,s_u}(\mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} - (1 - \theta) \Delta t \mathbf{K}_{d,s_u} \cdot \mathbf{U}^{n-1} \\
 & \quad + (1 - \theta) \Delta t \mathbf{G}_{s_u} \cdot \mathbf{P}^{n-1} - (1 - \theta) \Delta t \mathbf{C}_{s_u} \cdot \Theta^{n-1}
 \end{aligned} \tag{5.26}$$

$$\mathbf{G}_0^T \cdot \mathbf{U}^n = 0 \tag{5.27}$$

$$\begin{aligned}
 & \mathbf{M}_{t,s_t} \cdot \Theta^n + \theta \Delta t \mathbf{H}_{c,s_t}(\mathbf{U}^n) \cdot \Theta^n + \theta \Delta t \mathbf{H}_{d,s_t} \cdot \Theta^n \\
 & = \theta \Delta t \mathbf{F}_t^n + (1 - \theta) \Delta t \mathbf{F}_t^{n-1} + \mathbf{M}_{t,s_t} \cdot \Theta^{n-1} \\
 & \quad - (1 - \theta) \Delta t \mathbf{H}_{c,s_t}(\mathbf{U}^{n-1}) \cdot \Theta^{n-1} - (1 - \theta) \Delta t \mathbf{H}_{d,s_t} \cdot \Theta^{n-1}
 \end{aligned} \tag{5.28}$$

5.2.3 Block iterative algorithm

Now we will consider an iterative solution procedure for problem (5.26)–(5.28). In particular, the block iterative technique used to uncouple the calculation of the temperature and the velocity and pressure will be discussed in detail.

Equations (5.26)–(5.28) may be written together in a unified matrix expression. Let us denote by \mathbf{R}_u and \mathbf{R}_t the right-hand-side terms in Eqns. (5.26) and (5.28), respectively, and define the following matrices:

$$\begin{aligned}
\mathbf{A}_{11}(\mathbf{U}^n) &:= \mathbf{M}_{\mathbf{u},s_2^n} + \theta\Delta t \mathbf{K}_{c,s_2^n}(\mathbf{U}^n) + \theta\Delta t \mathbf{K}_{d,s_2^n} \\
\mathbf{A}_{12} &:= -\theta\Delta t \mathbf{G}_{s_2^n} \\
\mathbf{A}_{13} &:= \theta\Delta t \mathbf{C}_{s_2^n} \\
\mathbf{A}_{21} &:= \mathbf{G}_0^T \\
\mathbf{A}_{33}(\mathbf{U}^n) &:= \mathbf{M}_{t,s_1^n} + \theta\Delta t \mathbf{H}_{c,s_1^n}(\mathbf{U}^n) + \theta\Delta t \mathbf{H}_{d,s_1^n}
\end{aligned} \tag{5.29}$$

Having introduced this notation, Eqns. (4.26)–(4.28) are rewritten as:

$$\begin{pmatrix} \mathbf{A}_{11}(\mathbf{U}^n) & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{33}(\mathbf{U}^n) \end{pmatrix} \begin{pmatrix} \mathbf{U}^n \\ \mathbf{P}^n \\ \Theta^n \end{pmatrix} = \begin{pmatrix} \mathbf{R}_u \\ \mathbf{0} \\ \mathbf{R}_t \end{pmatrix} \tag{5.30}$$

In fact, \mathbf{A}_{12} and \mathbf{A}_{13} also depend on \mathbf{U}^n through the SD term, although such dependence has not been explicitly indicated. Assume for a moment that the Galerkin formulation is used and therefore \mathbf{A}_{12} and \mathbf{A}_{13} are constant. Suppose that the convective terms in the Navier-Stokes and the energy equations are linearized as follows:

$$\begin{aligned}
c(\mathbf{u}_h^{n,i}, \mathbf{u}_h^{n,i}, \mathbf{v}_h) &\approx c(\mathbf{u}_h^{n,i-1}, \mathbf{u}_h^{n,i}, \mathbf{v}_h) + \beta_u c(\mathbf{u}_h^{n,i}, \mathbf{u}_h^{n,i-1}, \mathbf{v}_h) - \beta_u c(\mathbf{u}_h^{n,i-1}, \mathbf{u}_h^{n,i-1}, \mathbf{v}_h) \\
f(\mathbf{u}_h^{n,i}, \vartheta_h^{n,i}, \eta_h) &\approx f(\mathbf{u}_h^{n,i-1}, \vartheta_h^{n,i}, \eta_h) + \beta_t f(\mathbf{u}_h^{n,i}, \vartheta_h^{n,i-1}, \eta_h) - \beta_t f(\mathbf{u}_h^{n,i-1}, \vartheta_h^{n,i-1}, \eta_h)
\end{aligned} \tag{5.31}$$

where superscript i denotes the iteration counter. For $\beta_u = \beta_t = 1$, (5.31) is the Newton-Raphson linearization and for $\beta_u = \beta_t = 0$ the Picard scheme. The resulting matrix version of the linearized equations will have the following aspect:

$$\begin{pmatrix} \mathbf{A}_{11}(\mathbf{U}^{n,i-1}) + \beta_u \mathbf{A}_{11}^*(\mathbf{U}^{n,i-1}) & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{0} & \mathbf{0} \\ \beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1}) & \mathbf{0} & \mathbf{A}_{33}(\mathbf{U}^{n,i-1}) \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n,i} \\ \mathbf{P}^{n,i} \\ \Theta^{n,i} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_u^* \\ \mathbf{0} \\ \mathbf{R}_t^* \end{pmatrix} \tag{5.32}$$

where

$$\begin{aligned}
\mathbf{R}_u^* &:= \mathbf{R}_u + \beta_u \mathbf{A}_{11}^*(\mathbf{U}^{n,i-1}) \cdot \mathbf{U}^{n,i-1} \\
\mathbf{R}_t^* &:= \mathbf{R}_t + \beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1}) \cdot \mathbf{U}^{n,i-1}
\end{aligned} \tag{5.33}$$

and \mathbf{A}_{11}^* , \mathbf{A}_{31}^* are the matrices coming respectively from the terms $c(\mathbf{u}_h^{n,i}, \mathbf{u}_h^{n,i-1}, \mathbf{v}_h)$ and $f(\mathbf{u}_h^{n,i}, \vartheta_h^{n,i-1}, \eta_h)$. Let us define now

$$\begin{aligned}
\mathbf{B}_{11} &:= \begin{pmatrix} \mathbf{A}_{11}(\mathbf{U}^{n,i-1}) + \beta_u \mathbf{A}_{11}^*(\mathbf{U}^{n,i-1}) & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{0} \end{pmatrix} \\
\mathbf{B}_{12} &:= \begin{pmatrix} \mathbf{A}_{13} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{B}_{21} := (\beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1}) \quad \mathbf{0}) \\
\mathbf{B}_{22} &:= \mathbf{A}_{33}(\mathbf{U}^{n,i-1}) \\
\mathbf{X} &:= \begin{pmatrix} \mathbf{U}^{n,i} \\ \mathbf{P}^{n,i} \end{pmatrix}, \quad \mathbf{F}_x := \begin{pmatrix} \mathbf{R}_u^* \\ \mathbf{0} \end{pmatrix} \\
\mathbf{Y} &:= \Theta^{n,i}, \quad \mathbf{F}_y := \mathbf{R}_t^*
\end{aligned} \tag{5.34}$$

The linear system (5.32) may be written as

$$\begin{pmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_x \\ \mathbf{F}_y \end{pmatrix} \tag{5.35}$$

A block iterative algorithm may be employed to solve (5.35). The main advantage of this method is that smaller linear systems will have to be solved, although iterations will be required. For the particular case of (5.35), two options are equally easy to implement [SB]:

- *Block Jacobi method (or block total-step method):*

$$\begin{aligned} \mathbf{B}_{11}\mathbf{X}^j &= \mathbf{F}_x - \mathbf{B}_{12}\mathbf{Y}^{j-1} \\ \mathbf{B}_{22}\mathbf{Y}^j &= \mathbf{F}_y - \mathbf{B}_{21}\mathbf{X}^{j-1} \end{aligned} \quad (5.36)$$

- *Block Gauss-Seidel method (or block single-step method):*

$$\begin{aligned} \mathbf{B}_{11}\mathbf{X}^j &= \mathbf{F}_x - \mathbf{B}_{12}\mathbf{Y}^{j-1} \\ \mathbf{B}_{22}\mathbf{Y}^j &= \mathbf{F}_y - \mathbf{B}_{21}\mathbf{X}^j \end{aligned} \quad (5.37)$$

or

$$\begin{aligned} \mathbf{B}_{22}\mathbf{Y}^j &= \mathbf{F}_y - \mathbf{B}_{21}\mathbf{X}^{j-1} \\ \mathbf{B}_{11}\mathbf{X}^j &= \mathbf{F}_x - \mathbf{B}_{12}\mathbf{Y}^j \end{aligned} \quad (5.38)$$

Remarks 5.2

- (1) It is understood that a convergence criterion has to be chosen to stop the iterative algorithms (5.36)–(5.38).
- (2) Physically, the distinction between (5.37) and (5.38) relies on which equation (mechanical or thermal) is solved first. Depending on the physics of the problem, one option may be more efficient than the other, although the improvement will be in no more than one iteration. In what follows, we will assume that the Navier-Stokes equations are solved first, being clear that the following discussion carries out verbatim if the order of block iterations is swapped. \square

The convergence of any of the algorithms (5.36)–(5.38) depends on the spectral radius of the square matrices contained in the off-diagonal matrices \mathbf{B}_{12} and \mathbf{B}_{21} [SB]. Matrix \mathbf{B}_{21} may be set to zero by selecting $\beta_t = 0$ (Picard method for the energy equation). However, from (5.34) and (5.29) it is seen that \mathbf{B}_{12} contains the coupling matrix $\mathbf{C}_{\theta,u}$. For a given time step size and a mesh diameter h , this matrix is proportional to $1/Gr$ (see Box 5.1) or, equivalently, to Gr/Re^2 (cf. Eqns. (5.5)). Therefore, *algorithms (5.36)–(5.38) will only converge for sufficiently small values of the Grashof number*. Although this fact might seem an important drawback for using a block iterative algorithm, this is not the case: when Gr (or Ra) are very high, even the linearization of the initial problem (5.30) leads to diverging schemes. Relaxation procedures are needed for these extreme cases to compute converged solutions.

The computational efficiency of a block iterative scheme is not clear for the linear problem (5.35). However, this linear system arises from the linearization of (5.30), i.e., from (5.32). The natural idea is to deal with the iterations due to the problem non-linearity and the block iterations in a single iterative loop. This leads to the following scheme:

$$\begin{pmatrix} \mathbf{A}_{11}(\mathbf{U}^{n,i-1}) + \beta_u \mathbf{A}_{11}^*(\mathbf{U}^{n,i-1}) & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{U}^{n,i} \\ \mathbf{P}^{n,i} \end{pmatrix} = \begin{pmatrix} \mathbf{R}_u^* - \mathbf{A}_{13}\Theta^{n,i-1} \\ \mathbf{0} \end{pmatrix} \quad (5.39)$$

$$\mathbf{A}_{33}(\mathbf{U}^{n,i-1})\Theta^{n,i} = \mathbf{R}_t^* - \beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1})\mathbf{U}^{n,i} \quad (5.40)$$

where for $k = i - 1$ this is a Jacobi-type method and for $k = i$ a Gauss-Seidel-type algorithm. Using the expression for \mathbf{R}_t^* given by (5.33), from Eqn. (5.40) we have that

$$\mathbf{A}_{33}(\mathbf{U}^{n,i-1}) \cdot \Theta^{n,i} + \beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1}) \cdot \mathbf{U}^{n,k} - \beta_t \mathbf{A}_{31}^*(\Theta^{n,i-1}) \cdot \mathbf{U}^{n,i-1} = \mathbf{R}_t \quad (5.41)$$

For $\beta_t = 0$ (Picard linearization for the energy equation) or $k = i - 1$ (block Jacobi algorithm), this last expression reduces to

$$\mathbf{A}_{33}(\mathbf{U}^{n,i-1}) \cdot \Theta^{n,i} = \mathbf{R}_t$$

Assume that $\beta_t = 1$ and $k = i$. Since

$$\mathbf{A}_{33}(\mathbf{U}^{n,i-1}) \cdot \Theta^{n,i} + \mathbf{A}_{31}^*(\Theta^{n,i-1}) \cdot \mathbf{U}^{n,i} - \mathbf{A}_{31}^*(\Theta^{n,i-1}) \cdot \mathbf{U}^{n,i-1}$$

is precisely the linearized expression of $\mathbf{A}_{33}(\mathbf{U}^{n,i}) \cdot \Theta^{n,i}$ and $\mathbf{U}^{n,i}$ is already known from (5.39), Eqn. (5.41) reduces to

$$\mathbf{A}_{33}(\mathbf{U}^{n,i}) \cdot \Theta^{n,i} = \mathbf{R}_t$$

Summarizing, Eqn. (5.40) may be replaced by

$$\mathbf{A}_{33}(\mathbf{U}^{n,k}) \cdot \Theta^{n,i} = \mathbf{R}_t \quad (5.42)$$

where

- a) $k = i - 1$ if the convective term in the energy equation is linearized up to first order (Picard method) or the block Jacobi method is used to couple the mechanical and thermal problems,
- b) $k = i$ otherwise, that is, second order linearization is used for the convective term in the energy equation and the block Gauss-Seidel method is employed as block iterative scheme.

Let us go back now to the original matrix notation for (5.39) and (5.40). There are two sources of nonlinearity reflected in these equations, the first coming from the nonlinear character of the physical problem and the second due to the block iterative method. As in Chapter 4, we will add two more sources: the SD method and the iterative penalization. Altogether, there are four reasons to iterate (nonlinear terms, block iterative coupling, SD method and iterative penalization) and they will be dealt with in a single iterative loop.

The final algorithm is the following (compare with (4.60)):

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} , \mathbf{P}^{n-1} and Θ^{n-1} , find \mathbf{U}^n , \mathbf{P}^n and Θ^n , approximations to $\mathbf{U}(t^n)$, $\mathbf{P}(t^n)$ and $\Theta(t^n)$, as the converged solutions of the following iterative algorithm:

$$\begin{aligned} & \mathbf{M}_{v,s_u^{n,i-1}} \cdot \mathbf{U}^{n,\epsilon(i)} + \theta \Delta t \mathbf{K}_{c,s_u^{n,i-1}}(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\ & \quad + \theta \Delta t \beta_u \mathbf{K}_{c,s_u^{n,i-1}}^*(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\ & \quad + \theta \Delta t \mathbf{K}_{d,s_u^{n,i-1}} \cdot \mathbf{U}^{n,\epsilon(i)} - \theta \Delta t \mathbf{G}_{s_u^{n,i-1}} \cdot \mathbf{P}^{n,\epsilon(i)} \\ & = \theta \Delta t \mathbf{F}_{v,s_u^{n,i-1}}^n + (1 - \theta) \Delta t \mathbf{F}_{v,s_u^{n,i-1}}^{n-1} + \mathbf{M}_{v,s_u^{n,i-1}} \cdot \mathbf{U}^{n-1} \\ & \quad - (1 - \theta) \Delta t \mathbf{K}_{c,s_u^{n,i-1}}(\mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} \\ & \quad + \theta \Delta t \beta_u \mathbf{K}_{c,s_u^{n,i-1}}^*(\mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i-1)} \\ & \quad - (1 - \theta) \Delta t \mathbf{K}_{d,s_u^{n,i-1}} \cdot \mathbf{U}^{n-1} + (1 - \theta) \Delta t \mathbf{G}_{s_u^{n,i-1}} \cdot \mathbf{P}^{n-1} \\ & \quad - \Delta t \mathbf{C}_{s_u^{n,i-1}} \cdot (\theta \Theta^{n,\epsilon(i-1)} + (1 - \theta) \Theta^{n-1}) \end{aligned} \quad (5.43)$$

$$\epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i)} + \mathbf{G}_0^T \cdot \mathbf{U}^{n,\epsilon(i)} = \epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i-1)} \quad (5.44)$$

$$\begin{aligned} & \mathbf{M}_{t,s_i^{n,k}} \cdot \Theta^{n,\epsilon(i)} + \theta \Delta t \mathbf{H}_{c,s_i^{n,k}}(\mathbf{U}^{n,\epsilon(k)}) \cdot \Theta^{n,\epsilon(i)} + \theta \Delta t \mathbf{H}_{d,s_i^{n,k}} \cdot \Theta^{n,\epsilon(i)} \\ & = \theta \Delta t \mathbf{F}_i^n + (1 - \theta) \Delta t \mathbf{F}_i^{n-1} + \mathbf{M}_{t,s_i^{n,k}} \cdot \Theta^{n-1} \\ & \quad - (1 - \theta) \Delta t \mathbf{H}_{c,s_i^{n,k}}(\mathbf{U}^{n-1}) \cdot \Theta^{n-1} \\ & \quad - (1 - \theta) \Delta t \mathbf{H}_{d,s_i^{n,k}} \cdot \Theta^{n-1} \end{aligned} \quad (5.45)$$

where $k = i - 1$ or $k = i$, according to the options a) and b) indicated above.

It is assumed in (5.44) that the iterative penalty method is used.

Remarks 5.3

- (1) The linearization of the SD term, the iterative penalization and the block Jacobi or block Gauss-Seidel methods can only yield a linear convergence rate, with a more or less steep slope in a plot iterations vs logarithm of the residual. Sooner or later, convergence will be driven by the slowest of these rates as the iterative procedure goes on, even though $\beta_u = 1$ be selected to linearize the Navier-Stokes equations. We have found from numerical experiments that the Newton-Raphson method is only useful when the Reynolds and the Rayleigh numbers are small. Otherwise, it only contributes to increase the computational cost, without reducing the number of iterations needed to reach a prescribed convergence tolerance.
- (2) If instead of using $\Theta^{n,\epsilon(i-1)}$ in (5.44) and $\mathbf{U}^{n,\epsilon(k)}$ in (5.45) ($k = i - 1$ or $k = i$) they are replaced by the temperature and velocity nodal values of the previous time step, Θ^{n-1} and \mathbf{U}^{n-1} , one is led to the so called 'staggered algorithms', in which the coupling between the Navier-Stokes and the energy equations is accomplished by means of the time stepping. The algorithm in time in this case is block explicit, regardless of the value of the parameter θ . Therefore, a critical time step exists above which the algorithm becomes unstable. See, e.g., References [PF], [WTS], [Zi] for related methods.
- (3) Referring again to the stability in time, if a fully converged solution is obtained for (5.43)–(5.45) then stability should be ensured provided that $\theta \geq 1/2$. Obviously, the block iterative method will not give exactly the same solution as the full nonlinear system. An error will remain that may affect the stability of the algorithm in time. Numerical experiments indicate that this in fact happens. We have found that $\theta = 1/2$ (Crank-Nicolson) is very sensitive to the convergence tolerance adopted for each time step. The higher it is, the sooner instabilities begin to appear, leading to the numerical blow-up after a few time steps. In this sense, the backward Euler scheme ($\theta = 1$) has been found to be much more robust. We have never found instability problems using this method. \square

5.3 Creeping flow of nonlinear materials

5.3.1 Generalized Newtonian fluids

The constitutive equation for a Newtonian fluid relates the stress tensor σ with the strain rate tensor ϵ through a linear equation, viz.,

$$\sigma = -p\mathbf{I} + 2\mu\epsilon(\mathbf{u}) \quad (5.46)$$

For a number of important materials, it is not possible to describe their rheological behavior with Eqn. (5.46) and using a constant value for the dynamical viscosity μ .

A very simple extension of the Newtonian constitutive law is to consider a variable viscosity μ in Eqn. (5.46) (this is still a particular case of the Reiner-Rivlin constitutive model). This allows to model several non-Newtonian flow phenomena observed in practice.

According to Tanner [Ta], the non-Newtonian fluid behavior in shear may be classified into three different types: time independent fluids, time dependent fluids and viscoelastic materials. The constitutive law for the first two types can be written as (5.46), with μ variable. Elastic effects have to be taken into account for viscoelastic materials.

Here, only time independent non-Newtonian fluids will be considered. Sometimes they are just called non-Newtonian viscous fluids or generalized Newtonian fluids. Time dependent materials could also be easily accommodated within the following formulation. For these materials, the viscosity increases in time (rheopectic fluids) or decreases (thixotropic fluids) for a constant shear rate. For simplicity, we shall assume that μ is time independent.

Generalized Newtonian fluids may be classified in turn into Bingham, pseudo-plastic and dilatant materials. Bingham materials only flow after the stress exceeds a certain threshold (yield stress). For pseudo-plastic fluids the viscosity falls progressively as the shear rate increases. Only for very high rates of shear, it ceases to decrease and remains constant. Dilatant materials exhibit the opposite response.

Pseudo-plastic and dilatant fluids have very important technological applications. High polymers, polymer solutions and many suspensions exhibit a pseudo-plastic behavior. Dilatant fluids are much less common in industrial applications. Some concentrated solutions of solids are an example of this type of materials.

One of the most extensively used constitutive laws for generalized Newtonian fluids is the so called *power law*. For a simple shear flow, its expression is

$$\mu = K_0|\dot{\gamma}|^{n-1}, \quad n > 0, \quad K_0 > 0 \quad (5.47)$$

where K_0 is the *material consistency* and n the *rate sensitivity*. Both K_0 and n are physical parameters to be determined from experimental data. In Eqn. (5.47), $\dot{\gamma}$ is the shear rate. For $0 < n < 1$, this equation represents a pseudo-plastic fluid (presenting the shear thinning effect near the walls) and for $n > 1$ a dilatant material (showing shear thickening near the walls). Observe that when $|\dot{\gamma}| = 0$ Eqn. (5.47) is meaningless. Another constitutive law that has a wider range of applicability is the Carreau model, whose expression for a simple shear flow is

$$\mu = \mu_0 [1 + (\lambda\dot{\gamma})^2]^{(n-1)/2} \quad (5.48)$$

where $\mu_0 > 0$ and $\lambda > 0$ are physical parameters and now $0 < n < 1$.

In general, μ can be considered as a function of the principle invariants of the strain rate tensor $\boldsymbol{\varepsilon}$, defined as

$$\begin{aligned} I_1(\boldsymbol{\varepsilon}) &:= \text{Tr}(\boldsymbol{\varepsilon}) = \nabla \cdot \mathbf{u}, \\ I_2(\boldsymbol{\varepsilon}) &:= \frac{1}{2} \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}, \\ I_3(\boldsymbol{\varepsilon}) &:= \det(\boldsymbol{\varepsilon}) \end{aligned} \quad (5.49)$$

For incompressible fluids, $\nabla \cdot \mathbf{u} = 0$ and thus $I_1(\boldsymbol{\varepsilon}) = 0$. In most situations of physical interest, μ is independent of $I_3(\boldsymbol{\varepsilon})$.

Let us see how Eqns. (5.47) and (5.48) can be generalized. For a simple plane shear flow with $\mathbf{u} = (u_1(x_2), 0)$ (in Cartesian coordinates x_1 and x_2) we have that

$$|\dot{\gamma}| = \left| \frac{\partial u_1}{\partial x_2} \right| = |2\boldsymbol{\varepsilon} : \boldsymbol{\varepsilon}|^{1/2} = |4I_2(\boldsymbol{\varepsilon})|^{1/2}$$

and therefore the generalized expressions for (5.47) and (5.48) are

$$\mu = K_0 [4I_2(\boldsymbol{\varepsilon})]^{(n-1)/2} \quad (\text{Power law}) \quad (5.50)$$

$$\mu = \mu_0 [1 + 4\lambda^2 I_2(\boldsymbol{\varepsilon})]^{(n-1)/2} \quad (\text{Carreau}) \quad (5.51)$$

Another very important type of constitutive law is the one representing viscoplastic materials. This rheological behavior is particularly well suited to model the flow of metals in metal forming processes. When plastic deformations are much more important than elastic deformations, elastic effects may be simply neglected. This is the so called *flow approach*.

Here we will briefly describe a particular type of viscoplastic model, namely, Perzyna's model (see, e.g., References [Oñ], [OH], [ZG], [ZJO], [ZOH] for more information). The basic assumption is that the viscoplastic strain rate is related to the stress through the following equation:

$$\varepsilon_{ij} = \gamma \langle \phi(F) \rangle \frac{\partial Q}{\partial \sigma_{ij}} \quad (5.52)$$

where F is the yield function for the material, Q the plastic potential, ϕ a certain function that defines the model and $\langle \cdot \rangle$ is the Macauley bracket, defined by $\langle f \rangle = f$ if $f \geq 0$ and $\langle f \rangle = 0$ if $f < 0$. The constant γ in Eqn. (5.52) has the physical meaning of being the fluidity parameter.

Assume now associate plasticity ($F = Q$) and take for F the von Mises yield surface,

$$F = Q = \sqrt{2I_2(\boldsymbol{\sigma}') - \sigma_y} \quad (5.53)$$

where σ_y is the uniaxial yield stress of the material and $I_2(\boldsymbol{\sigma}')$ is the second principle invariant of the tensor $\boldsymbol{\sigma}' := \boldsymbol{\sigma} + p\mathbf{I}$. For the function ϕ , the following power law is adopted:

$$\phi(F) = F^m \quad (5.54)$$

After some calculations, from (5.53) it is found that

$$\frac{\partial Q}{\partial \sigma_{ij}} = \frac{\sqrt{3}}{2\sqrt{I_2(\boldsymbol{\sigma}')}} \sigma'_{ij} \quad (5.55)$$

Using Eqns. (5.53)–(5.55) in (5.52) we obtain that

$$\varepsilon_{ij} = \gamma < \left(\sqrt{3I_2(\boldsymbol{\sigma}')} - \sigma_y \right)^m > \frac{\sqrt{3}}{2\sqrt{I_2(\boldsymbol{\sigma}')}} \sigma'_{ij} \quad (5.56)$$

This equation (5.56) represents a generalized Newtonian material, the viscosity μ being one half of the inverse of the coefficient multiplying σ'_{ij}

Assume that $\sqrt{3I_2(\boldsymbol{\sigma}')} - \sigma_y > 0$ and define

$$\hat{\varepsilon} := \left(\frac{2}{3} \varepsilon_{ij} \varepsilon_{ij} \right)^{1/2} = \left(\frac{4}{3} I_2(\boldsymbol{\varepsilon}) \right)^{1/2} \quad (5.57)$$

We will have that

$$\begin{aligned} I_2(\boldsymbol{\sigma}') &= \frac{1}{2} \boldsymbol{\sigma}' : \boldsymbol{\sigma}' = \frac{1}{2} (2\mu)^2 \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon} = 3\mu^2 \hat{\varepsilon}^2, \\ \hat{\varepsilon} &= \frac{1}{\sqrt{3\mu}} \sqrt{I_2(\boldsymbol{\sigma}')} = \gamma \left(\sqrt{3I_2(\boldsymbol{\sigma}')} - \sigma_y \right)^m \\ &= \gamma (3\mu \hat{\varepsilon} - \sigma_y)^m \end{aligned}$$

and hence

$$\mu = \frac{\sigma_y + (\hat{\varepsilon}/\gamma)^{1/m}}{3\hat{\varepsilon}} \quad (5.58)$$

This expression will only be valid for high values of $\hat{\varepsilon}$. Observe that for $\sigma_y = 0$ it reduces to the power-law model given by (5.50) and with a certain identification of the physical parameters.

In Section 5.5.3 we will present a numerical simulation of a fluid whose viscosity obeys the power-law (5.50) and in next chapter the problem of lamination of a metal flat plate using the constitutive equation (5.58).

Besides the nonlinear dependence of the viscosity on the invariant $I_2(\boldsymbol{\varepsilon})$ expressed by Eqns. (5.50), (5.51) and (5.58), it may also depend on the temperature and the pressure. The physical parameters in these equations depend on the temperature [ZJO], [ZOH]. But even for Newtonian fluids, the viscosity depends on the temperature and the pressure. Using reaction-rate concepts [Ta], the viscosity may be expressed in terms of the (absolute) temperature ϑ as

$$\mu = \mu_0 \exp \left(\frac{E}{R\vartheta} \right) \quad (5.59)$$

where E is the activation energy, R the gas constant ($8.314 \text{ J K}^{-1} \text{ mol}^{-1}$) and μ_0 is the viscosity for $E = 0$. For small temperature changes around a reference value ϑ_0 , Eqn. (5.59) may be replaced by

$$\mu = \mu_0 \exp [-\alpha(\vartheta - \vartheta_0)]$$

where now μ_0 is the viscosity for $\vartheta = \vartheta_0$.

Concerning the dependence of μ on a pressure variation p , an expression of the form

$$\mu = \mu_0 \exp \left(\frac{p}{B} \right) \quad (5.60)$$

is often adopted. This equation can be derived from thermodynamical bases using the free volume concept [Ta]. In general, the physical parameter B is very large and pressure variations do not affect much the value of the viscosity.

In what follows, we will assume that an expression of μ in terms of $I_2(\boldsymbol{\varepsilon})$ and ϑ is given. Since $I_2(\boldsymbol{\varepsilon})$ is really a function of the velocity field \mathbf{u} , we will write, symbolically,

$$\mu = \mu(\vartheta, \mathbf{u}) \quad (5.61)$$

The only way to solve fluid flow problems involving nonlinear viscosities is numerically. For Newtonian flows, analytical solutions in some simple cases allow to understand which could be the flow behavior in more general situations. However, for non-Newtonian flows even for simple problems numerical techniques are needed. For the numerical simulation of some simple flow cases of non-Newtonian fluids, see, e.g., References [BLL], [BP], [CC], [DK], [DR], [SY], [TTK], among many others.

5.3.2 Stationary problem and finite element discretization

The rheological behavior described above is usually valid for highly viscous materials. Therefore, inertial terms in the Navier-Stokes equations will have a very little influence, i.e., the Reynolds number will be very small. In order to simplify the exposition, the convective term in the momentum equations will be dropped, that is, only creeping flows will be considered. Moreover, since the transient evolution will not introduce anything new, the stationary problem will be treated.

Under the assumptions just stated, the problem to be solved is to find a velocity field \mathbf{u} , a pressure p and a temperature ϑ such that

$$\begin{aligned} -2\nabla \cdot [\mu\boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p &= \rho\mathbf{f} & \text{in } \Omega \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega \\ \rho c_p \mathbf{u} \cdot \nabla \vartheta - k\Delta \vartheta &= Q & \text{in } \Omega \end{aligned} \quad (5.62)$$

In the energy equation, Q is the source term. Only the source coming from the mechanical dissipation into heat will be taken into account:

$$\begin{aligned} Q &= \boldsymbol{\sigma} : \boldsymbol{\varepsilon} = -p\mathbf{I} : \boldsymbol{\varepsilon}(\mathbf{u}) + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) \\ &= 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{u}) \end{aligned} \quad (5.63)$$

where the fact that $\mathbf{I} : \boldsymbol{\varepsilon}(\mathbf{u}) = \nabla \cdot \mathbf{u} = 0$ has been used.

The same boundary conditions as in Section 5.2 will be considered (Eqns. (5.8)). Sometimes, the Neumann-type prescription for the temperature has to be generalized to a Robbins boundary condition to include the surface heat convection, although this is immaterial for what follows.

The spaces of trial solutions needed for the stationary problem are:

$$\begin{aligned} V_{\mathbf{u}} &= \{\mathbf{v} \in H^1(\Omega)^{N_{sd}} \mid \mathbf{v}|_{\Gamma_{du}} = \bar{\mathbf{u}}\} \\ V_p &= \{q \in L^2(\Omega) \mid \int_{\Omega} q d\Omega = 0 \text{ if } \Gamma_{nu} = \emptyset\} \\ V_t &= \{\eta \in H^1(\Omega) \mid \eta|_{\Gamma_{dt}} = \bar{\vartheta}\} \end{aligned} \quad (5.64)$$

The spaces of test functions are again given by (5.10). Introducing the forms

$$\begin{aligned}
a(\mu; \mathbf{u}, \mathbf{v}) &= 2 \int_{\Omega} \mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega, \\
b(q, \mathbf{v}) &= \int_{\Omega} q \nabla \cdot \mathbf{v} d\Omega, \\
l_u(\mathbf{v}) &= \int_{\Omega} \rho \mathbf{f} \cdot \mathbf{v} d\Omega + \int_{\Gamma_{nu}} \bar{\mathbf{t}} \cdot \mathbf{v} d\Gamma, \\
e(\vartheta, \eta) &= \int_{\Omega} k \nabla \vartheta \cdot \nabla \eta d\Omega, \\
f(\mathbf{u}, \vartheta, \eta) &= \int_{\Omega} \rho c_p \eta \mathbf{u} \cdot \nabla \vartheta d\Omega, \\
l_t(\mu, \mathbf{u}; \eta) &= 2 \int_{\Omega} \mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\eta) d\Omega + \int_{\Gamma_{nt}} \bar{\varphi} \eta d\Gamma,
\end{aligned} \tag{5.65}$$

the weak form of problem (5.62) with the boundary conditions (5.8) is: Find $\mathbf{u} \in V_u$, $p \in V_p$ and $\vartheta \in V_t$ such that

$$\begin{aligned}
a(\mu; \mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) &= l_u(\mathbf{v}) & \forall \mathbf{v} \in W_u \\
b(q, \mathbf{u}) &= 0 & \forall q \in W_p \\
f(\mathbf{u}, \vartheta, \eta) + e(\vartheta, \eta) &= l_t(\mu, \mathbf{u}; \eta) & \forall \eta \in W_t
\end{aligned} \tag{5.66}$$

We now consider the finite element discretization of problem (5.66). For simplicity, the Galerkin approach will be used, although the SD formulation might be needed to stabilize the convective term in the energy equation.

The discrete version of problem (5.66) leads to the following algebraic system:

$$\mathbf{K}(\mu) \cdot \mathbf{U} - \mathbf{G} \cdot \mathbf{P} = \mathbf{F}_u \tag{5.67}$$

$$\mathbf{G}^T \cdot \mathbf{U} = \mathbf{0} \tag{5.68}$$

$$\mathbf{H}(\mathbf{u}) \cdot \boldsymbol{\Theta} = \mathbf{F}_t(\mu, \mathbf{u}) \tag{5.69}$$

The notation used for the matrices and the vectors is the same as before, although subscripts have been omitted. Matrix $\mathbf{H}(\mathbf{u})$ for the temperature equation (5.69) accounts for both the diffusive and convective terms. We have explicitly indicated the dependence of the matrices and vectors in the above equations on the viscosity and the velocity.

Let us discuss now the construction of the finite element spaces $V_{u,h}$, $V_{p,h}$ and $V_{t,h}$. Consider first the case in which the viscosity μ does not depend on the temperature ϑ . Under this assumption, Eqns. (5.67) and (5.68) are uncoupled with Eqn. (5.69), that can be solved once \mathbf{U} is known. If the viscosity μ is constant (Newtonian fluid), we know that the discrete velocity space $V_{u,h}$ and pressure space $V_{p,h}$ must satisfy the discrete Babuška-Brezzi (BB) stability condition. When the viscosity depends on the invariants of the strain-rate tensor $\boldsymbol{\varepsilon}(\mathbf{u})$, the question is whether this condition will be sufficient for assessing stability and convergence of the finite element scheme. In Reference [BN], it is proved that for the case in which the viscosity obeys the power law or the Carreau model, stable and convergent velocity-pressure pairs for the Stokes problem with μ constant are also stable for the nonlinear case. Concerning the convergence of the method, let h be the diameter of $\{\Omega^e\}$ and suppose that the rate of convergence for

the velocity is of order h^m for Newtonian flows. Assume now that μ satisfies the power law with rate of sensitivity n , with $0 < n < 1$. Then, the rate of convergence for the velocity will be of order h^{mn} . For the Carreau model, the same rate of convergence as for the constant viscosity case can be obtained. See Reference [BN] for details.

Based on these results, finite element interpolations for the velocity and the pressure that are known to satisfy the discrete Babuška-Brezzi condition have been employed also for this problem. As in Section 5.2, the temperature will be interpolated like the velocity components.

5.3.3 Iterative techniques

Iterative penalty method

Let us consider first the case in which μ is constant. The iterative penalty method applied to problem (5.67)–(5.69) is:

Given $\mathbf{P}^{\epsilon(0)}$, for $i = 1, 2, \dots$ find $\mathbf{U}^{\epsilon(i)}$ and $\mathbf{P}^{\epsilon(i)}$ such that

$$\begin{aligned} \mathbf{K}(\mu) \cdot \mathbf{U}^{\epsilon(i)} - \mathbf{G} \cdot \mathbf{P}^{\epsilon(i)} &= \mathbf{F}_u \\ \mathbf{G}^T \cdot \mathbf{U}^{\epsilon(i)} + \epsilon \mathbf{M}_p \cdot \mathbf{P}^{\epsilon(i)} &= \epsilon \mathbf{M}_p \cdot \mathbf{P}^{\epsilon(i-1)} \end{aligned} \quad (5.70)$$

This is the discrete version of problem (3.73). It should be remarked that the initial guess $\mathbf{P}^{\epsilon(0)}$ must be such that the associated pressure (interpolated from these nodal values) have zero mean value.

The analysis of Section 3.4.1 revealed that the convergence of (5.70) relies on the value of the parameter

$$\bar{\epsilon} := \epsilon \frac{N_a^2}{K_a K_b^2}$$

where N_a is the norm of $a(\mu; \cdot, \cdot)$, K_a its coercivity constant and K_b the constant in the Babuška-Brezzi condition. Since now both N_a and K_a will be proportional to μ , we will have that

$$\bar{\epsilon} = \epsilon \mu C \quad (5.71)$$

for a certain constant C . In fact, using the same arguments as in Section 3.4.1 one obtains (see Reference [CCO] for details):

$$\begin{aligned} \|\mathbf{U} - \mathbf{U}^{\epsilon(i)}\| &\leq (\epsilon \mu C)^i \frac{C'}{\mu} \|\mathbf{P} - \mathbf{P}^{\epsilon(0)}\| \\ \|\mathbf{P} - \mathbf{P}^{\epsilon(i)}\| &\leq (\epsilon \mu C)^i \|\mathbf{P} - \mathbf{P}^{\epsilon(0)}\| \end{aligned} \quad (5.72)$$

where C' is a constant and $\|\cdot\|$ denotes the discrete L^2 norm.

It is important to observe that convergence is governed by the parameter $\bar{\epsilon}$, which is proportional to the viscosity μ . This explains why ϵ must be taken proportional to μ^{-1} , since what provides an idea of how well the incompressibility constraint will be approximated is $\bar{\epsilon}$, and not ϵ itself. Of course, this comment can also be applied to the classical penalty method (observe that the first pass in (5.70) is nothing but the standard penalty method), and must be kept in mind when one deals with non-constant viscosities.

In practical problems, we have encountered two cases in which the standard penalty method cannot be applied and the iterative penalization is mandatory. The first is the one discussed now, concerning non-Newtonian flows with variable viscosity. In the numerical examples presented below, it will be seen that the viscosity varies several orders of magnitude in the fluid domain. Recall that the practical rule for choosing the penalty parameter for the classical penalty method is to take it in the range $10^{-6}\mu^{-1}$ to $10^{-9}\mu^{-1}$. If a reference viscosity μ_0 is chosen *a priori* for determining a suitable value of ϵ , it is not known whether this penalty parameter will yield a sufficiently accurate satisfaction of the incompressibility constraint or to ill-conditioning of the final stiffness matrix. We will insist on this point later. Let us just mention that this ill-conditioning for non-Newtonian flows precludes the use of iterative solvers for the resulting algebraic system, in which case the behavior of the standard penalty method is certainly disappointing [CWJ]. For an application of the Augmented Lagrangian method to non-Newtonian fluids, see Reference [HTB].

Perhaps another case in which the importance of the variable viscosity is more clear is when the pseudo-concentration method is used to follow free surfaces. This will be the subject of Chapter 6.

Iterative algorithm for thermally coupled non-Newtonian flows

In order to solve the coupled nonlinear system of equations (5.67)–(5.69) we will use a block iterative algorithm, as in Section 5.2. Once again, the nonlinearity of the problem and the iterative penalization will be dealt with within the same iterative loop.

Let $\mu^{(k,l)}$ denote the viscosity function when the temperature is known at iteration k and the velocity at iteration l . Let TOL be a given convergence tolerance. As in the previous chapter, we check convergence using the criterion $\|\mathbf{U}^{\epsilon(i)} - \mathbf{U}^{\epsilon(i-1)}\| < TOL\|\mathbf{U}^{\epsilon(i)}\|$. The iterative scheme used is the following:

Box 5.1 Algorithm for thermally coupled non-Newtonian flows

- Initialise $\mu^{(0,0)}$, $\mathbf{P}^{(0)}$, $\Theta^{(0)}$
- $i := 0$
- WHILE (*not converged*) DO:
 - $i \leftarrow i + 1$
 - Solve:

$$\mathbf{K}(\mu^{(i-1,i-1)}) \cdot \mathbf{U}^{\epsilon(i)} - \mathbf{G} \cdot \mathbf{P}^{\epsilon(i)} = \mathbf{F}_u$$

$$\mathbf{G}^T \cdot \mathbf{U}^{\epsilon(i)} + \epsilon \mathbf{M}_p \cdot \mathbf{P}^{\epsilon(i)} = \epsilon \mathbf{M}_p \mathbf{P}^{\epsilon(i-1)}$$
 - Update:

$$\mu^{(i-1,i)} = \mu(\vartheta^{\epsilon(i-1)}, \mathbf{u}^{\epsilon(i)})$$
 - Solve:

$$\mathbf{H}(\mathbf{u}^{\epsilon(i)}) \cdot \Theta^{\epsilon(i)} = \mathbf{F}_t(\mu^{(i-1,i)}, \mathbf{u}^{\epsilon(i)})$$
 - Update:

$$\mu^{(i,i)} = \mu(\vartheta^{\epsilon(i)}, \mathbf{u}^{\epsilon(i)})$$
 - Check convergence:

$$\text{If } \|\mathbf{U}^{\epsilon(i)} - \mathbf{U}^{\epsilon(i-1)}\| < TOL\|\mathbf{U}^{\epsilon(i)}\| \text{ then (converged)}$$
- END while
- END

Remarks 5.4

- (1) Observe that the thermal problem is solved once the mechanical variables \mathbf{U} and \mathbf{P} are known for a certain iteration. There is also the possibility of swapping the order of block iterations. However, for the problems we have considered so far we have found the described option (slightly) more efficient.
- (2) The iterative penalization in the above algorithm is coupled with the iterative loop used to deal with the nonlinearity of the problem. It will be seen in the numerical experiments presented below that this does not deteriorate the convergence rate of the scheme.
- (3) If the viscosity does not depend on the temperature, the algorithm presented is a Picard (or successive substitution) type scheme. This is the most common option in practice [CWJ], [HTB], [LLH], [ZJO], [ZOH]. In fact, convergence problems have been observed when a Newton-Raphson scheme has been employed in the type of problems we consider (see Reference [CSS] for further discussion and references therein). The Picard method has been found to be faster than the Newton-Raphson algorithm. Anyway, convergence is slow for small values of the rate sensitivity n when the Power-law model is adopted for the viscosity. In Reference [TNB], it is proposed to redefine $\mu^{(k,i)}$ as

$$\mu^{(k,i)} \leftarrow \mu^{(k,i)} + \omega(1-n) (\mu^{(k,i)} - \mu^{(k,i-1)})$$

The value $\omega = 0.4$ was found to be a good choice.

- (4) From the results of the previous section, it is clear that the algorithm of Box 5.1 can be thought of as a Gauss-Seidel iterative scheme with a Newton-Raphson linearization of the energy equation, since the velocity used in this equation is the actual iterate, both for the convective term and the source term. \square

5.4 General problem—Iterative procedure

5.4.1 Motivation

In Sections 5.2 and 5.3 we have described the numerical techniques used for two particular problems of practical interest. Both problems can be placed in the general setting to be considered now, defined by the following system of partial differential equations:

$$\begin{aligned} \rho[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}] - 2\nabla \cdot [\mu \boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p &= \rho \mathbf{f} & \text{in } \Omega, t \in (0, T) \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega, t \in (0, T) \\ \rho c_p [\partial_t \vartheta + (\mathbf{u} \cdot \nabla) \vartheta] - \nabla \cdot (k \nabla \vartheta) &= Q & \text{in } \Omega, t \in (0, T) \end{aligned} \quad (5.73)$$

where the physical properties and the forcing terms may be variable. Particular cases of interest are:

- ρ and c_p depend on the temperature. This in fact is observed experimentally [IOS], [ZMS]. However, the Eulerian derivatives of ρ and c_p have to be small enough to ensure that the simplifications that lead from the general conservation equations of continuum mechanics (momentum, mass and energy) to (5.73) are still valid.
- μ is a function of ϑ and the invariants of $\boldsymbol{\varepsilon}(\mathbf{u})$. This is the problem considered in Section 5.3.

- k is a function of ϑ . No assumption on the magnitude of the spatial and temporal derivatives of the diffusion is now required. This situation is found when the Fourier law of heat conduction, $\mathbf{q} = -k\nabla\vartheta$, \mathbf{q} being the flux of heat, has to be generalized to $\mathbf{q} = -\nabla g(\vartheta)$, where g is a nonlinear function of the temperature. The effective conduction coefficient is now $g'(\vartheta)$. Nonlinear diffusion problems are often found in practice.
- \mathbf{f} depends on the temperature. The Boussinesq approximation is an example of this situation.
- Q depends on the velocity and the temperature. This happens when the Joule effect is not neglected (Section 5.3). Internal heat sources may be also introduced, due for example to chemical reactions or electromagnetic effects.
- In Chapter 6, ρ , μ , c_p and k will be considered variable in space due to the presence of two different fluids in the domain Ω .

Other nonlinearities in the problem may arise because of the boundary conditions. For example, typical surface radiation models lead to the boundary condition

$$-k\mathbf{n} \cdot \nabla\vartheta = \bar{\varphi} + \alpha(\vartheta^r - \vartheta_\infty^r)$$

for the temperature, α and r being physical parameters and ϑ_∞ the ambient temperature outside the domain Ω . Surface convection and surface conduction laws have similar expressions.

5.4.2 Time discretization

The fact that the density ρ and the specific heat c_p be variable introduce an additional difficulty in the time discretization of Eqns. (5.73). To see this, let us neglect the convective term in the momentum equation and let us write it as

$$\rho\partial_t\mathbf{u} + \mathcal{G}(\mathbf{u}, p) = \rho\mathbf{f} \quad (5.74)$$

where $\mathcal{G}(\mathbf{u}, p) = -2\nabla \cdot [\mu\boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p$. Dividing Eqn. (5.74) by ρ and using the generalized trapezoidal rule to discretize in time leads to

$$\begin{aligned} \frac{1}{\Delta t} (\mathbf{u}^n - \mathbf{u}^{n-1}) + \frac{\theta}{\rho^n} \mathcal{G}(\mathbf{u}^n, p^n) + \frac{1-\theta}{\rho^{n-1}} \mathcal{G}(\mathbf{u}^{n-1}, p^{n-1}) \\ = \theta\mathbf{f}^n + (1-\theta)\mathbf{f}^{n-1} \end{aligned} \quad (5.75)$$

If μ is constant, the terms in $\mathcal{G}(\mathbf{u}, p)$ will lead to constant matrices once the spatial discretization has been performed, and one needs to compute them only once. However, if $\mathcal{G}(\mathbf{u}, p)$ is multiplied by $1/\rho$, these matrices have to be computed for each time step. In order to avoid this additional computational cost, let us multiply Eqn. (5.75) by ρ^n :

$$\begin{aligned} \frac{\rho^n}{\Delta t} (\mathbf{u}^n - \mathbf{u}^{n-1}) + \theta\mathcal{G}(\mathbf{u}^n, p^n) + (1-\theta)\frac{\rho^n}{\rho^{n-1}}\mathcal{G}(\mathbf{u}^{n-1}, p^{n-1}) \\ = \rho^n\theta\mathbf{f}^n + \rho^n(1-\theta)\mathbf{f}^{n-1} \end{aligned} \quad (5.76)$$

Since the temporal derivative of ρ must be small, we can approximate

$$\frac{\rho^n}{\rho^{n-1}} \approx 1 \quad (5.77)$$

Using the approximation (5.77), the variation of ρ only affects the terms where it appears explicitly in Eqn. (5.74), that is, the approximation of the velocity time derivative and the body force term.

Observe from Eqn. (5.76) that the approximation given by (5.76) is unnecessary when $\theta = 1$. A similar situation is found when the specific heat varies in time.

The approximation just described will be used in Chapter 6, where the temporal variation of ρ and c_p will be due to the advance of a fluid in an air-filled domain.

5.4.3 Fully discrete and linearized problem

We proceed now to present the algorithm that combines all the ideas developed up to now. The basic scheme for the numerical solution of the Navier-Stokes equations was presented in Box 4.3. This scheme will be completed now with the inclusion of the temperature equation and the block iterative method to couple the mechanical and thermal problems.

The notation used before will be kept in what follows. In particular, the forms that define the problem will be those given by (4.13) and (5.65), now with all the physical properties within the integral symbol, since they may be variable. The dependence on these physical properties of the matrices and vectors resulting after the finite element discretization has been performed will be explicitly indicated. The source term Q in the energy equation will be considered in the force vector \mathbf{F}_t .

The final transient and iterative algorithm using the generalized trapezoidal rule to discretize in time, the SD formulation for the space discretization, the iterative penalty method and the block iterative coupling is the following:

For $n = 1, 2, \dots, N$, given \mathbf{U}^{n-1} , \mathbf{P}^{n-1} and Θ^{n-1} , find \mathbf{U}^n , \mathbf{P}^n and Θ^n , approximations to $\mathbf{U}(t^n)$, $\mathbf{P}(t^n)$ and $\Theta(t^n)$, as the converged solutions of the following iterative algorithm:

$$\begin{aligned}
& \mathbf{M}_{v,s_u^{n,i-1}}(\rho^n) \cdot \mathbf{U}^{n,\epsilon(i)} + \theta \Delta t \mathbf{K}_{c,s_u^{n,i-1}}(\rho^n; \mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\
& \quad + \theta \Delta t \beta_u \mathbf{K}_{c,s_u^{n,i-1}}^*(\rho^n; \mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i)} \\
& \quad + \theta \Delta t \mathbf{K}_{d,s_u^{n,i-1}}(\mu^n) \cdot \mathbf{U}^{n,\epsilon(i)} - \theta \Delta t \mathbf{G}_{s_u^{n,i-1}} \cdot \mathbf{P}^{n,\epsilon(i)} \\
& = \theta \Delta t \mathbf{F}_{v,s_u^{n,i-1}}^n + (1 - \theta) \Delta t \mathbf{F}_{v,s_u^{n,i-1}}^{n-1} + \mathbf{M}_{v,s_u^{n,i-1}}(\rho^n) \cdot \mathbf{U}^{n-1} \\
& \quad - (1 - \theta) \Delta t \mathbf{K}_{c,s_u^{n,i-1}}(\rho^n; \mathbf{U}^{n-1}) \cdot \mathbf{U}^{n-1} \\
& \quad + \theta \Delta t \beta_u \mathbf{K}_{c,s_u^{n,i-1}}^*(\rho^n; \mathbf{U}^{n,\epsilon(i-1)}) \cdot \mathbf{U}^{n,\epsilon(i-1)} \\
& \quad - (1 - \theta) \Delta t \mathbf{K}_{d,s_u^{n,i-1}}(\mu^{n-1}) \cdot \mathbf{U}^{n-1} + (1 - \theta) \Delta t \mathbf{G}_{s_u^{n,i-1}} \cdot \mathbf{P}^{n-1}
\end{aligned} \tag{5.78}$$

$$\epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i)} + \mathbf{G}_0^T \cdot \mathbf{U}^{n,\epsilon(i)} = \epsilon \mathbf{M}_p \mathbf{P}^{n,\epsilon(i-1)} \tag{5.79}$$

$$\begin{aligned}
& \mathbf{M}_{t,s_t^{n,i}}(\rho^n, c_p^n) \cdot \Theta^{n,\epsilon(i)} + \theta \Delta t \mathbf{H}_{c,s_t^{n,i}}(\rho^n, c_p^n; \mathbf{U}^{n,\epsilon(i)}) \cdot \Theta^{n,\epsilon(i)} \\
& \quad + \theta \Delta t \mathbf{H}_{d,s_t^{n,i}}(k^n) \cdot \Theta^{n,\epsilon(i)} \\
& = \theta \Delta t \mathbf{F}_{t,s_t^{n,i}}^n + (1 - \theta) \Delta t \mathbf{F}_{t,s_t^{n,i}}^{n-1} + \mathbf{M}_{t,s_t^{n,i}}(\rho^n, c_p^n) \cdot \Theta^{n-1} \\
& \quad - (1 - \theta) \Delta t \mathbf{H}_{c,s_t^{n,i}}(\rho^n, c_p^n; \mathbf{U}^{n-1}) \cdot \Theta^{n-1} \\
& \quad - (1 - \theta) \Delta t \mathbf{H}_{d,s_t^{n,i}}(k^{n-1}) \cdot \Theta^{n-1}
\end{aligned} \tag{5.80}$$

Observe that the approximation given by (5.77) has been employed for the terms involving the density (and also for the specific heat).

Box 5.2 General algorithm for thermally coupled flows

- Set the initial condition \mathbf{U}^0 , Θ^0 and $\mathbf{P}^0 = 0$
 - $n := 0$
 - WHILE $n < N$ and (*non-stationary*) DO:
 - $n \leftarrow n + 1$
 - IF $n < n_{eu}$ then $\theta = 1$
ELSE select θ , $\theta \geq 1/2$
 - $i := 0$
 - Set $\mathbf{U}^{n,\epsilon(0)} = \mathbf{U}^{n-1}$, $\mathbf{P}^{n,\epsilon(0)} = \mathbf{P}^{n-1}$ and $\Theta^{n,\epsilon(0)} = \Theta^{n-1}$
 - WHILE (*not converged*) DO:
 - $i \leftarrow i + 1$
 - Solve the Navier-Stokes equations (5.78)–(5.79)
 - Update:

$$\begin{aligned} \mu^n &\leftarrow \mu(\vartheta^{n,\epsilon(i-1)}, \mathbf{u}^{n,\epsilon(i)}) \\ \mathbf{F}_t^n &\leftarrow \mathbf{F}_t(\vartheta^{n,\epsilon(i-1)}, \mathbf{u}^{n,\epsilon(i)}) \end{aligned}$$
 - Solve the temperature equation (5.80)
 - Update:

$$\begin{aligned} \rho^n &\leftarrow \rho(\vartheta^{n,\epsilon(i)}) \\ \mu^n &\leftarrow \mu(\vartheta^{n,\epsilon(i)}, \mathbf{u}^{n,\epsilon(i)}) \\ \mathbf{F}_v^n &\leftarrow \mathbf{F}_v(\vartheta^{n,\epsilon(i)}) \\ c_p^n &\leftarrow c_p(\vartheta^{n,\epsilon(i)}) \\ k^n &\leftarrow k(\vartheta^{n,\epsilon(i)}) \\ \mathbf{F}_t^n &\leftarrow \mathbf{F}_t(\vartheta^{n,\epsilon(i)}, \mathbf{u}^{n,\epsilon(i)}) \end{aligned}$$
 - Check convergence:

$$\begin{aligned} \text{IF } \|\mathbf{U}^{n,\epsilon(i)} - \mathbf{U}^{n,\epsilon(i-1)}\|_{L^q} < TOL \|\mathbf{U}^{n,\epsilon(i)}\|_{L^q} \\ \text{and } \|\Theta^{n,\epsilon(i)} - \Theta^{n,\epsilon(i-1)}\|_{L^q} < TOL \|\Theta^{n,\epsilon(i)}\|_{L^q} \\ \text{then (converged)} \end{aligned}$$
 - END while (*not converged*)
 - $\mathbf{U}^n \leftarrow \mathbf{U}^{n,\epsilon(i)}$
 - $\mathbf{P}^n \leftarrow \mathbf{P}^{n,\epsilon(i)}$
 - $\Theta^n \leftarrow \Theta^{n,\epsilon(i)}$
 - Check if the steady-state has been reached:

$$\begin{aligned} \text{IF } \|\mathbf{U}^n - \mathbf{U}^{n-1}\|_{L^q} < TOL \Delta t \|\mathbf{U}^n\|_{L^q} \\ \text{and } \|\Theta^n - \Theta^{n-1}\|_{L^q} < TOL \Delta t \|\Theta^n\|_{L^q} \\ \text{then (stationary)} \end{aligned}$$
 - END while $n < N$ and (*non-stationary*)
- END

We shall assume that the physical properties ρ , c_p and k and the body force term in Eqn. (5.78) are functions of the temperature ϑ and that the viscosity and the forcing term \mathbf{F}_t depend on the temperature and the velocity, the latter through the source term Q . The basic flow chart to solve Eqns. (5.78)–(5.79) is given in Box 5.2, where the same notation as in Box 4.3 has been employed. It is assumed that all the terms depending

on the temperature and the velocity are updated as soon as possible. For the particular case of the Boussinesq problem, it has been shown in Section 5.2 that this is equivalent to use the Gauss-Seidel block iterative method and the Newton-Raphson linearization of the energy equation.

5.5 Some applications of the numerical method

We present thereafter the numerical simulation of three different problems involving thermally coupled flows. The Boussinesq approximation is the mathematical model for the first two examples. The last problem is the 4:1 plane extrusion of a nonlinear material, with the viscosity depending on the temperature.

The numerical calculations have been carried out on a CONVEX-C320 computer using double arithmetic precision.

5.5.1 Thermoconvective instability of plane Poiseuille flow

The problem definition is sketched in Figure 5.1. It consists of a two-dimensional laminar flow in a horizontal channel suddenly heated from below. A parabolic inlet velocity profile is prescribed, whereas the outlet is left free, i.e., the associated natural boundary condition is zero traction.

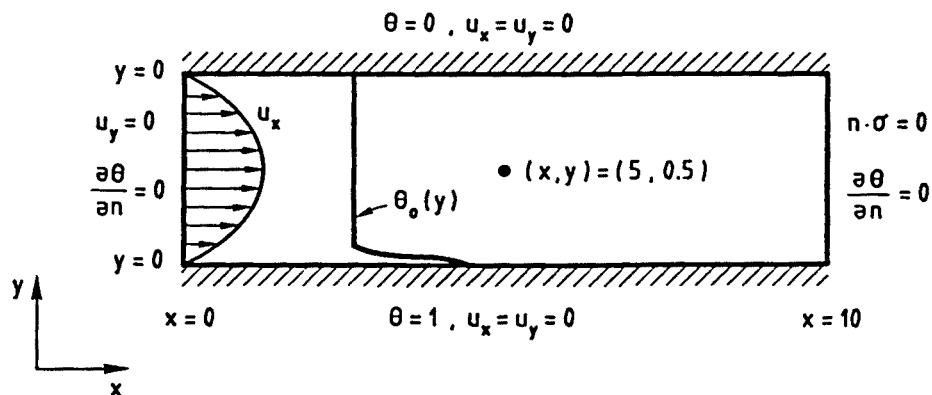


Figure 5.1 Geometry, initial and boundary conditions for the problem of thermoconvective instability of plane Poiseuille flow. Coordinates, velocity and temperature are assumed to be dimensionless.

This problem is solved in Reference [EP] as a benchmark for open boundary flows using a finite difference method and a fine grid.

This numerical test can be considered as a model for several relevant engineering problems, such as the fabrication of microelectronic circuits using the chemical vapour deposition process (cf. [EP], see references therein).

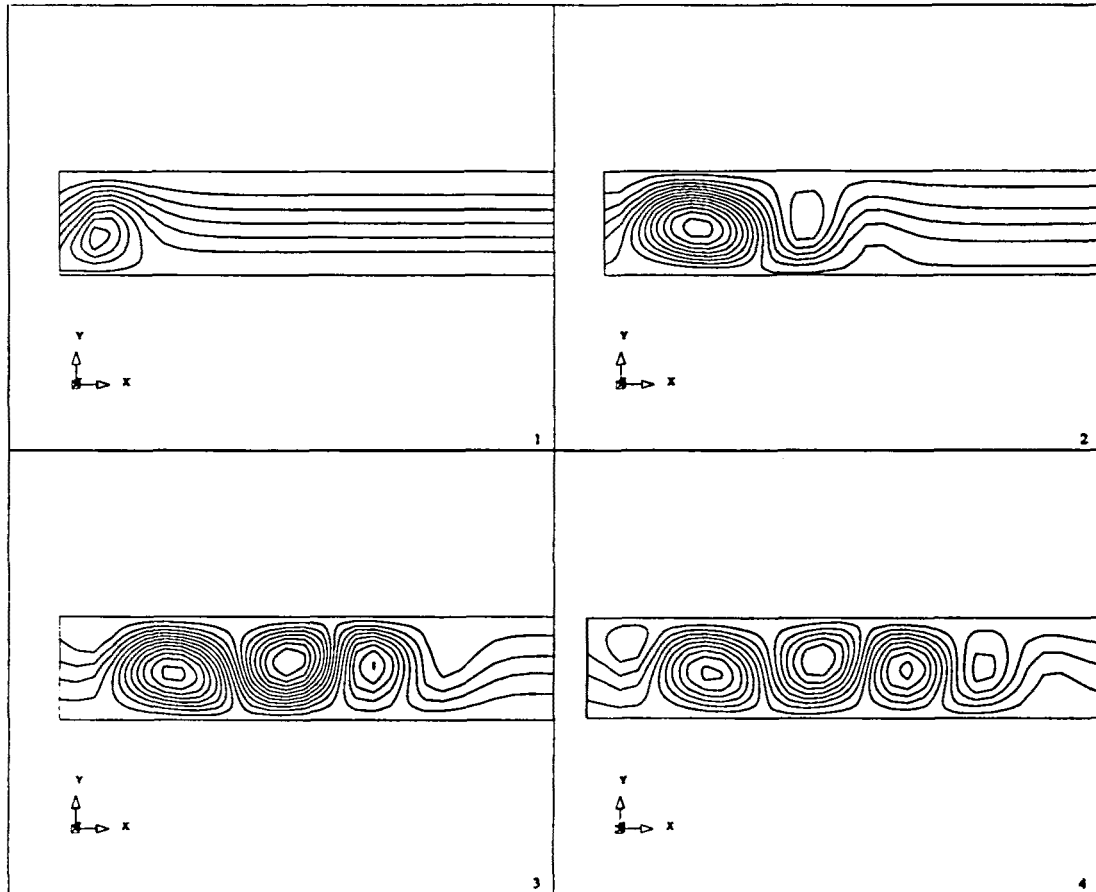


Figure 5.2 Transient evolution of the streamlines for the plane Poiseuille flow (PPF) heated from below at times: (1): $t = 0.2$; (2): $t = 0.8$; (3): $t = 1.2$; (4): $t = 1.4$.

Referring to Eqns. (5.6), the dimensionless parameters of the problem have been taken as $Re = 10$, $Fr = 1/150$ and $Pe = 40/9$ (the average inlet velocity, the height of the channel and the temperature difference between the top and bottom walls have been chosen as reference values for velocity, length and temperature, respectively). These parameters are the same as in Reference [EP] except for the Péclet number, which is slightly higher in that work ($Pe = 20/3$). In both cases, these values result in a thermoconvective instability of the basic Poiseuille flow. The linear stability analysis of unstable stratified plane Poiseuille flow in an infinite horizontal channel can be found in Reference [GR]. It is shown there that the form of the instability could vary from travelling transverse waves to longitudinal rolls, with axes parallel to the main flow direction and thus leading to a three-dimensional flow pattern. Travelling transverse waves are found for small values of the Rayleigh number. This is the situation for the dimensionless parameters used here and therefore a two-dimensional calculation is possible. It should be remarked, however, that three-dimensional effects are in general very important for thermally coupled flows [Ke].

Let us describe now the numerical strategy followed to solve this problem. The domain $[0, 10] \times [0, 1]$ has been discretized using a uniform mesh of $30 \times 15 = 450$ Q_2/P_1

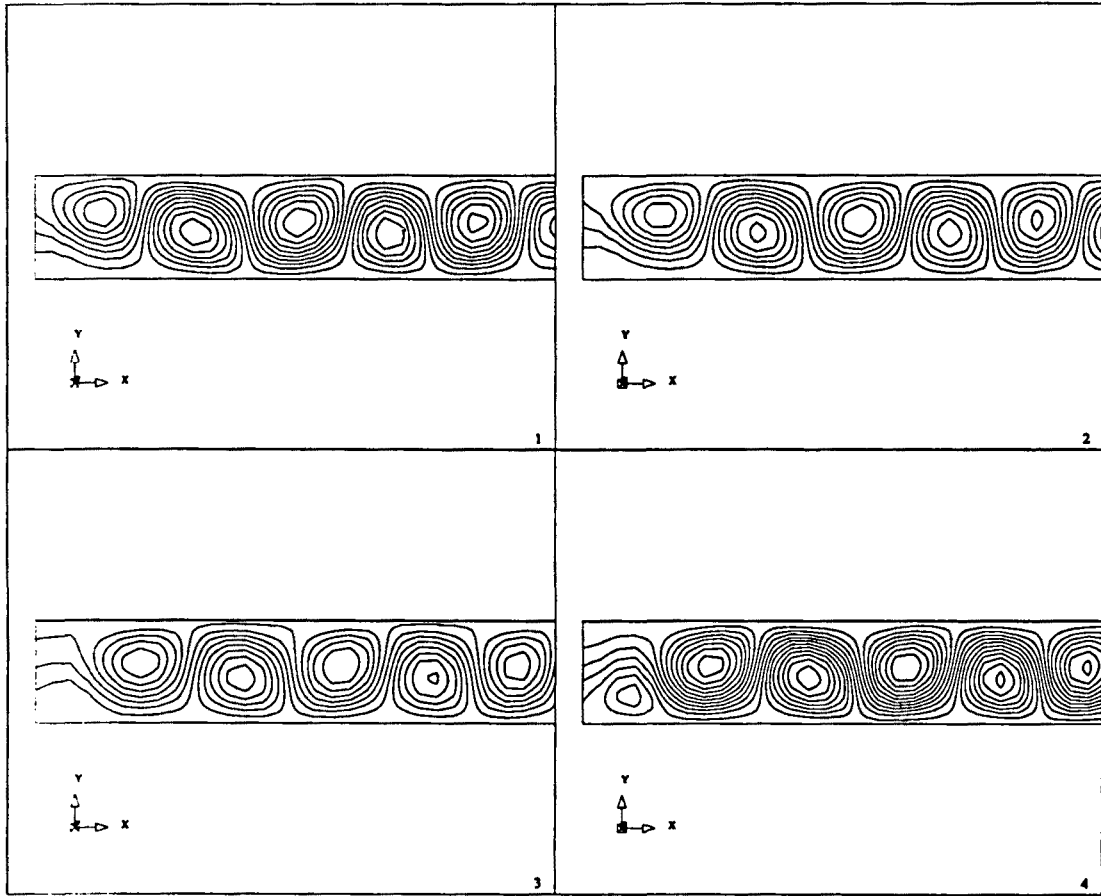


Figure 5.3 Transient evolution of the streamlines for the plane Poiseuille flow (PPF) heated from below at times: (1): $t = 1.6$; (2): $t = 1.8$; (3): $t = 2.2$; (4): $t = 2.5$.

elements, yielding 1891 nodal points. For this longitudinal length, it is concluded in Reference [EP] that the numerical solution is not affected by the artificial boundary conditions for $2 \leq x \leq 8$.

We have tested both the iterative penalty method with a parameter $\epsilon = 10^{-4}$ and the classical penalization, now with $\epsilon = 10^{-7}$. We will show later that both approaches yield a similar approximation for the incompressibility constraint and convergence history.

The SD formulation has been used for the space discretization, with an upwind parameter $\alpha_0 = 0.5$ (quadratic elements) and a natural length $h_0 = 2$ (corresponding to quadrilateral elements). The use of this method is needed to stabilize the convective terms of both the Navier-Stokes and the energy equations, since the cell Reynolds number and the cell Péclet number are higher than two. It is found that the maximum velocity norm is about 14.6 (cf. Figure 5.15) and therefore the maximum values of these parameters are $(Re)_{max}^\epsilon = |\mathbf{u}^\epsilon|_{max} h^\epsilon Re/2 \approx 14.6 \times 0.33 \times 10/2 = 24.33$ and $\gamma_{max} = |\mathbf{u}^\epsilon|_{max} h^\epsilon Pe/2 \approx 14.6 \times 0.33 \times 4.44/2 = 10.81$.

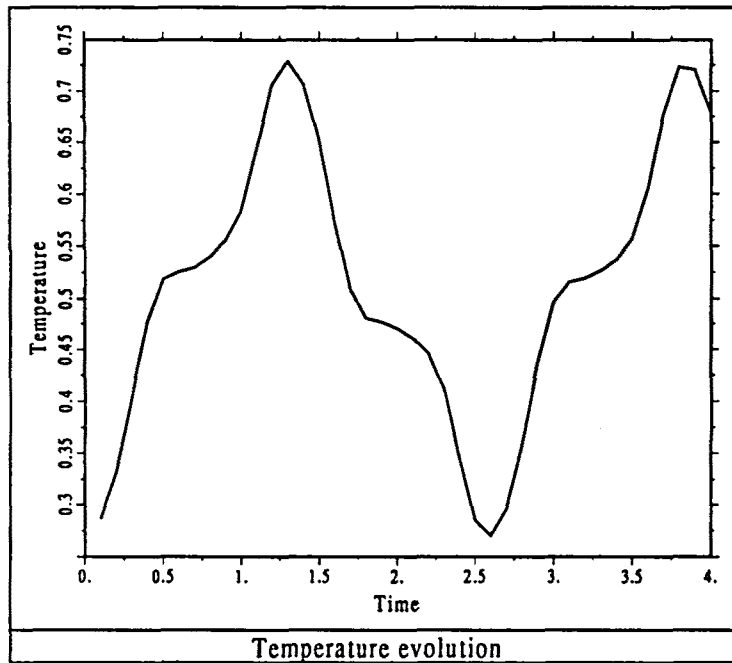


Figure 5.4 Transient evolution of the temperature at the central point ($x = 5$, $y = 0.5$). The initial time corresponds to $t = 3.3$ of the initial calculation shown in Figures 5.2 and 5.3 (PPF).

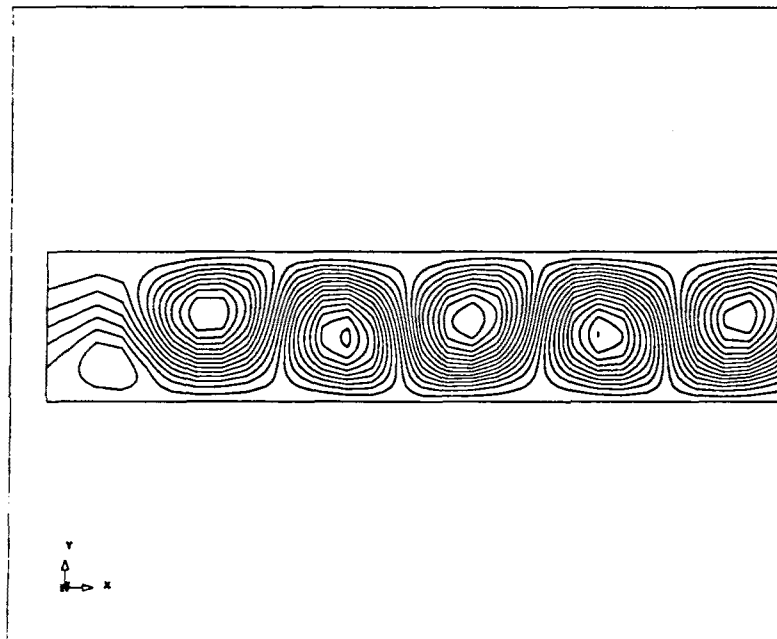


Figure 5.5 Streamlines for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

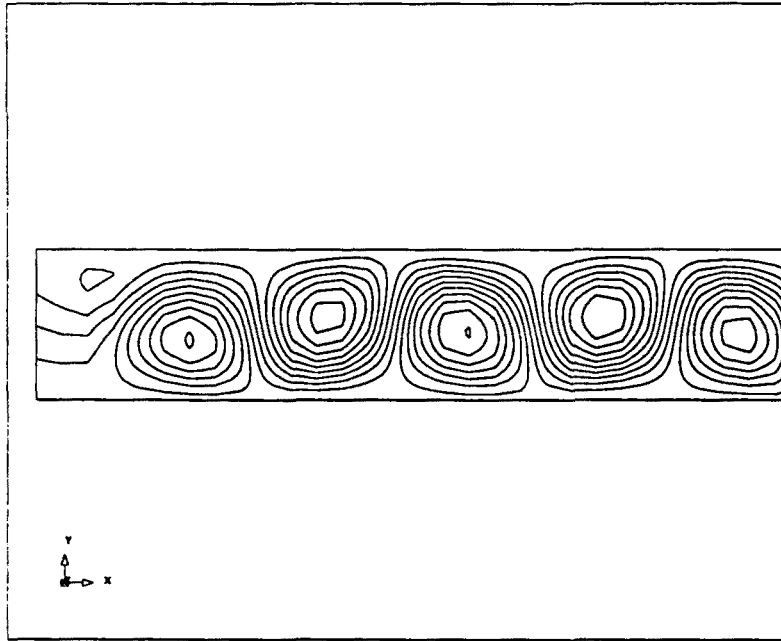


Figure 5.6 Streamlines for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

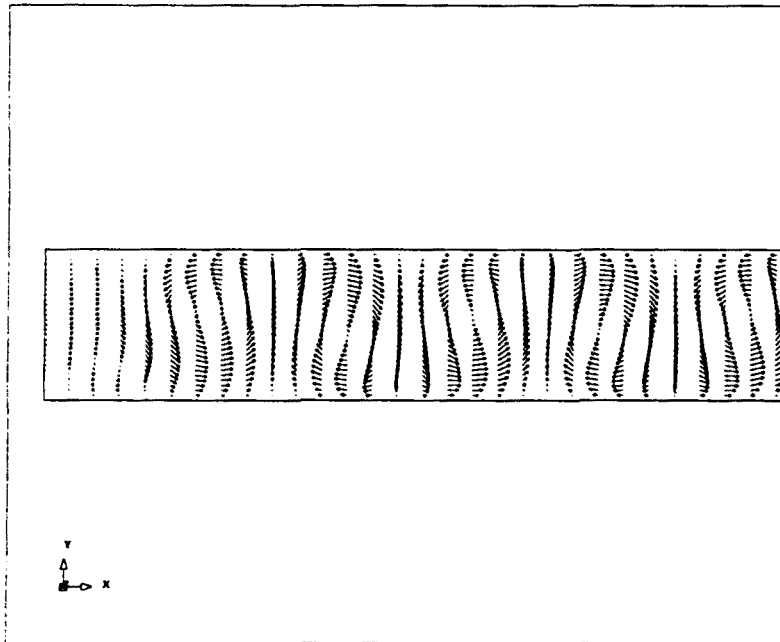


Figure 5.7 Velocity vectors for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

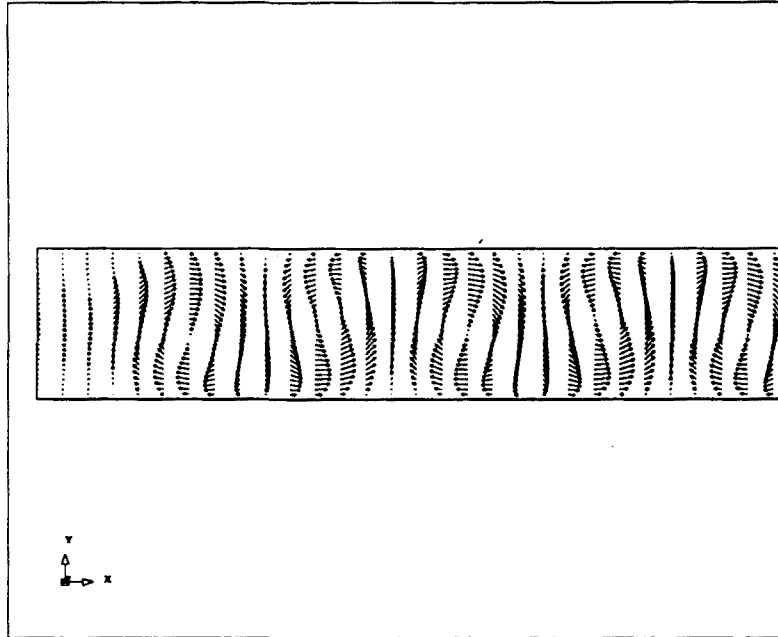


Figure 5.8 Velocity vectors for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

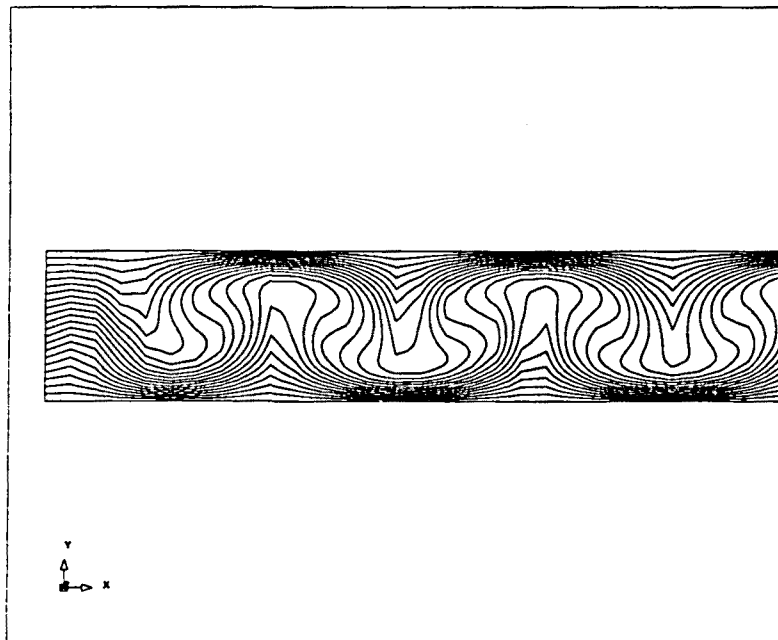


Figure 5.9 Temperature contours for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

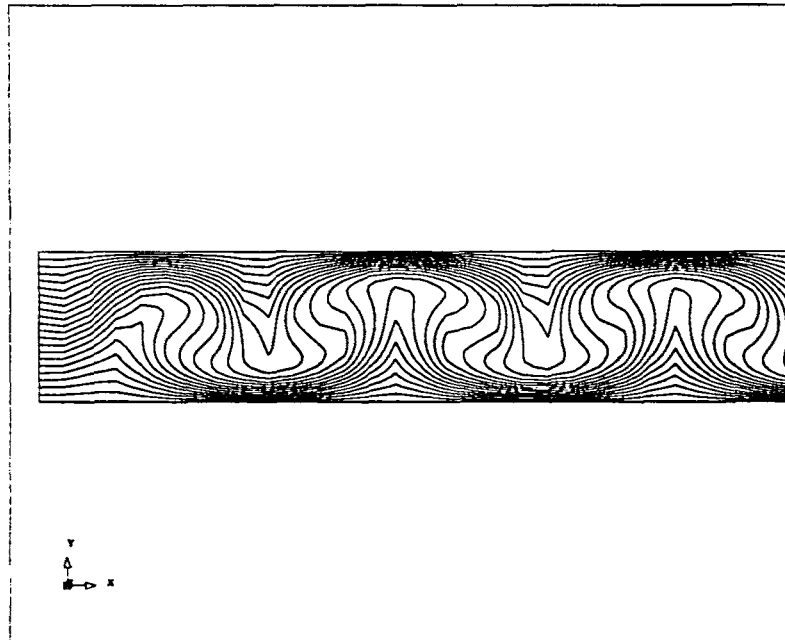


Figure 5.10 Temperature contours for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

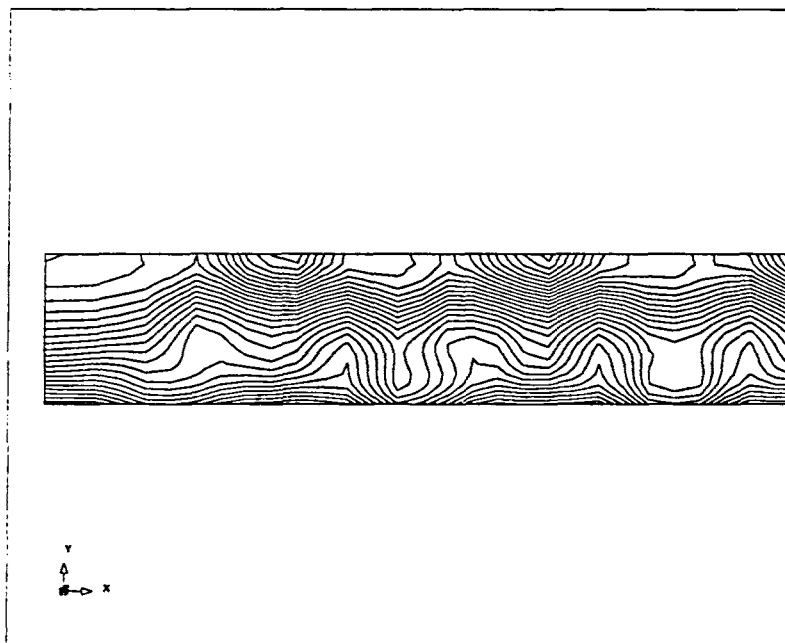


Figure 5.11 Pressure contours for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

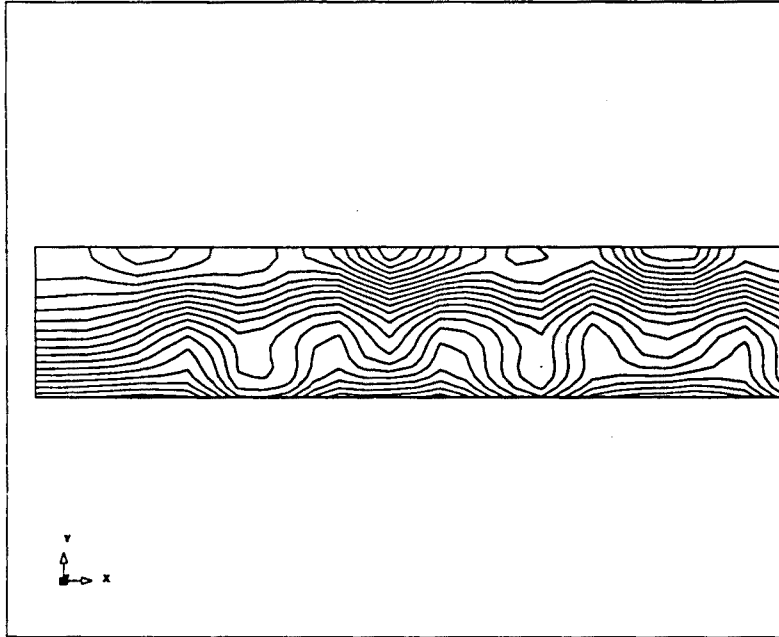


Figure 5.12 Pressure contours for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

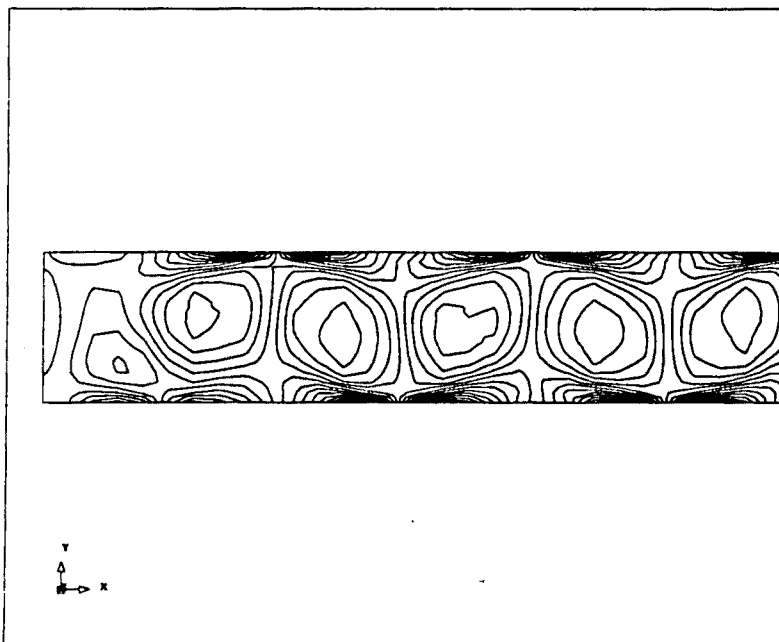


Figure 5.13 Vorticity contours for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

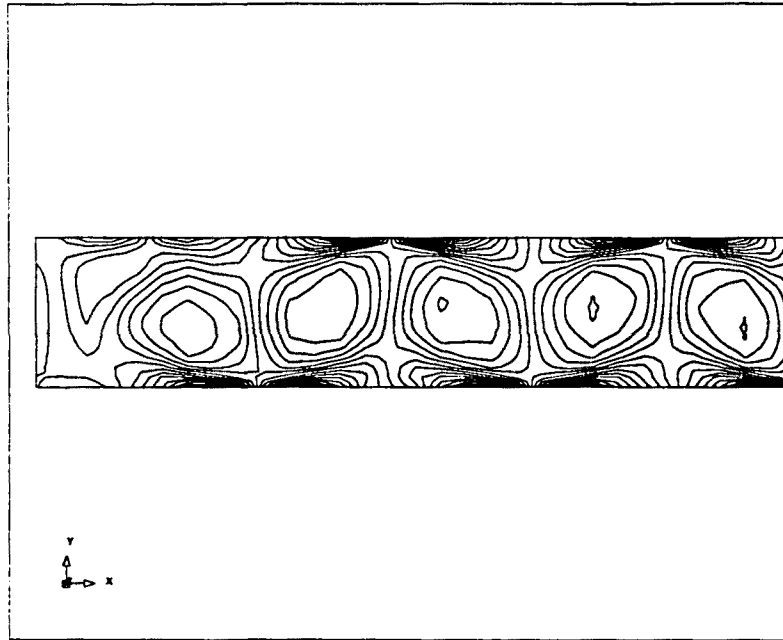


Figure 5.14 Vorticity contours for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

The Gauss-Seidel block iterative procedure with a Newton-Raphson linearization of the energy equation has been used, solving first the Navier-Stokes equations and then the temperature equation. The first problem has been linearized only up to first order (Picard method). The convergence tolerance has been taken as 0.1 % in the relative L^2 norm.

Of special interest is the choice of the parameter θ of the generalized trapezoidal rule. As it has already been mentioned, we have found that the Crank-Nicolson method is very sensitive to the convergence tolerance (see Remark 5.3.(3)). The time step size has been taken as $\Delta t = 0.01$. For $TOL = 10\%$, instability problems have been found at time step number 5, whereas for $TOL = 1\%$ they do not appear until time step number 37 and for $TOL = 0.1\%$ until time step number 121. Using the backward Euler method ($\theta = 1$) the time stepping algorithm has been found to be stable in all the cases. The results presented here have been obtained using this method.

Numerical results are shown in Figures 5.2 to 5.18 (only one half of the computational domain is shown). The acronym 'PPF' (standing for Plane Poiseuille Flow) has been used to identify the problem to which figures correspond. The transient evolution from the basic Poiseuille flow to the periodic flow pattern finally obtained has been plotted in the eight snapshots of Figures 5.2 and 5.3 (times are given in the captions). After a time $t = 3.3$, the travelling waves are fully developed and a new run has been carried out, redefining $t = 0$ for $t = 3.3$. The period of the oscillations has been found to be approximately 2.5 time units. This can be observed from the transient evolution of the temperature at the central point ($x = 5, y = 0.5$) depicted in Figure 5.4. This value is very sensitive to the Péclet number, since in Reference [EP] and for $Pe = 20/3$ this period was found to be approximately 1.5 time units.

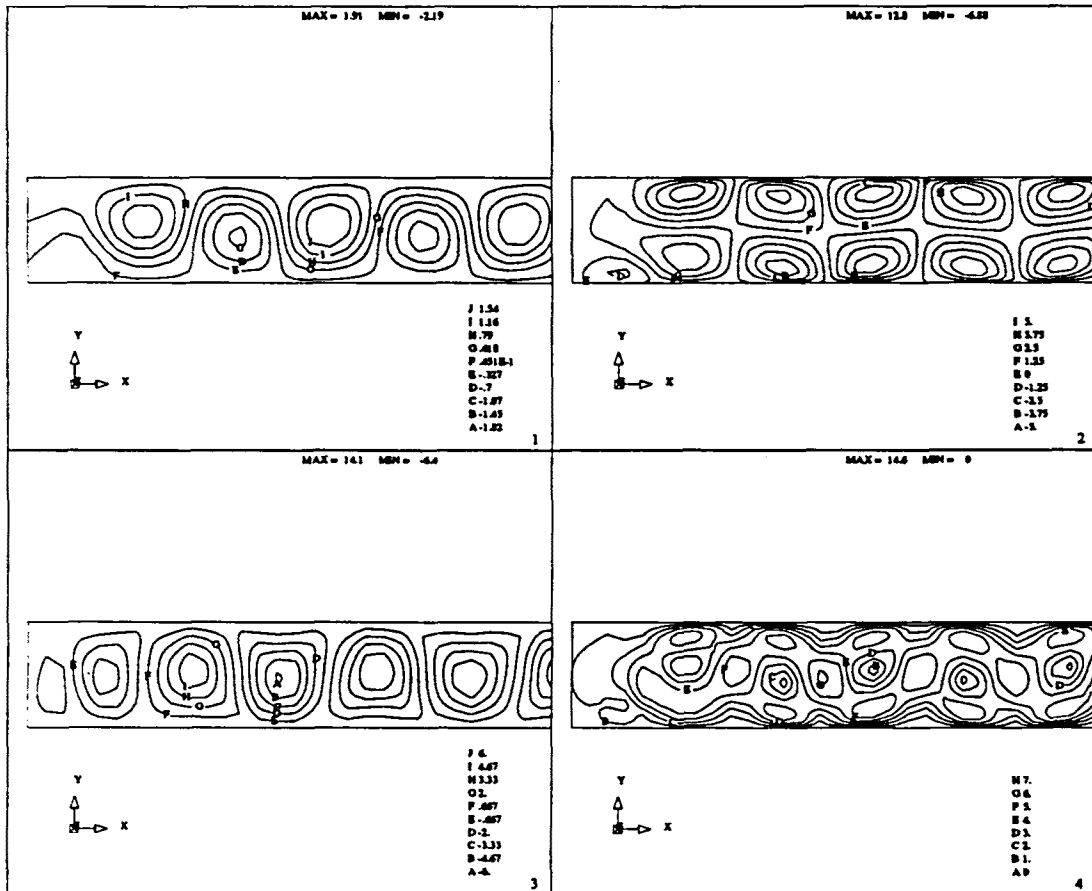


Figure 5.15 Numerical values for the velocity and streamfunction at $t = 1.3$ (PPF). (1): Streamfunction; (2): x -velocity component contours; (3): y -velocity component contours; (4): Norm of the velocity contours.

The streamlines, velocity vectors, temperature contours, isobars and vorticity contours for $t = 1.3$ and $t = 2.6$ are plotted in Figures 5.5 to 5.14. The first time corresponds to a maximum value for the temperature at the central point and the latter to a minimum. The periodicity of all these fields can be observed from the plots. Numerical values are given for $t = 1.3$ in Figures 5.15 and 5.16.

Finally, Figures 5.17 and 5.18 show the streamlines for $t = 1.3$ and $t = 2.6$ in the whole computational domain. The bad influence of the artificial boundary conditions can be observed, especially in what concerns the outlet wall. It is clear that the zero traction prescription does not reproduce the effect of an infinitely long channel. The proper evaluation of boundary conditions necessary for the numerical simulation of flows in infinite domains is an area that still deserves a lot of research.

Once the numerical strategy and the physical results have been described, let us discuss now the numerical behavior of the algorithm. The convergence history and the evolution of the incompressibility constraint for the first two time steps, starting from the Poiseuille flow, are shown in Figure 5.19. These results correspond to the iterative penalty method with $\epsilon = 10^{-4}$. The same plots for time step number 532 ($t = 5.32$)

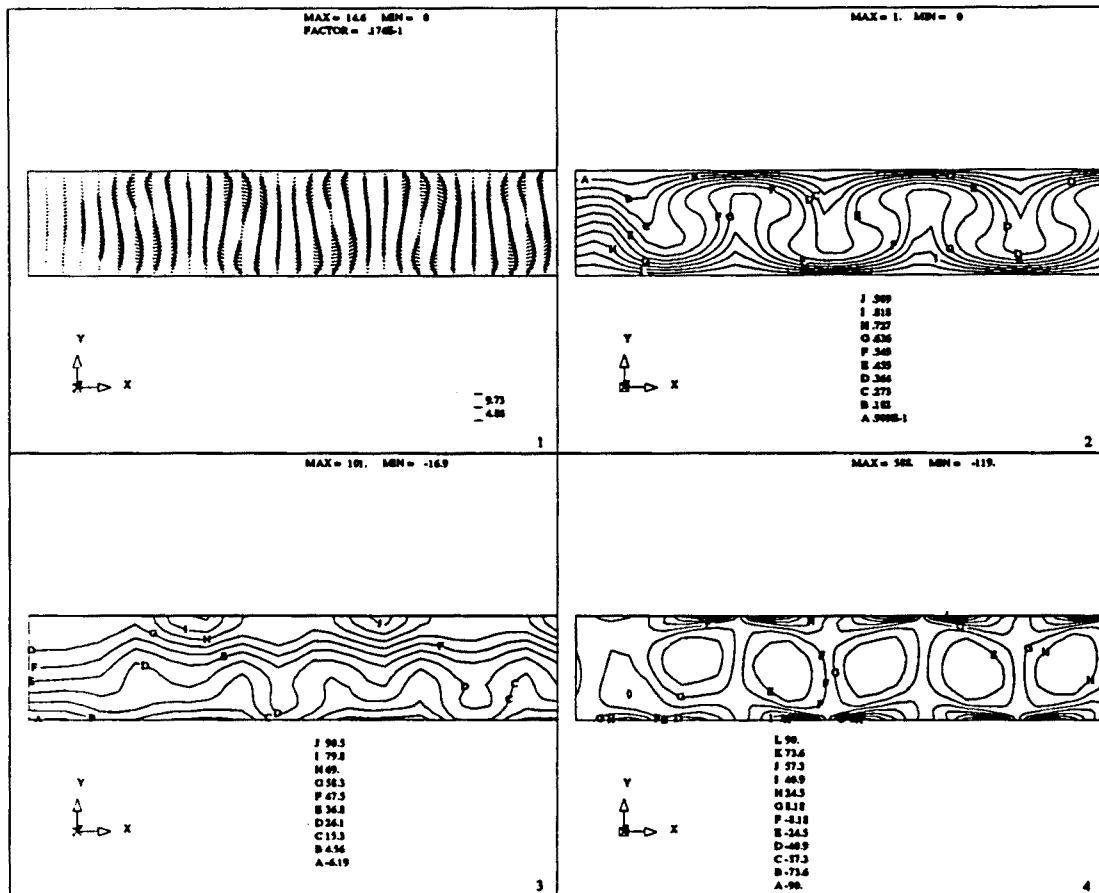


Figure 5.16 Numerical values for the velocity vectors, temperature, pressure and vorticity at $t = 1.3$ (PPF). (1): Velocity vectors; (2): Temperature; (3): Pressure; (4): Vorticity.

are shown in Figure 5.20, now for both the iterative penalty method with $\epsilon = 10^{-4}$ and the classical penalization with $\epsilon = 10^{-7}$. It is observed that the convergence history in both cases is almost the same, whereas the norm of the discrete velocity divergence decreases in the first case to 0.5×10^{-7} and in the second case it remains constant and equal to 0.15×10^{-7} . The excellent behavior of the iterative penalty method is again observed for this type of problems.

Between three and five iterations have been required to converge for each time step. The CPU time per iteration has been 22.1 seconds. It is important to remark that the solution of the Navier-Stokes equation requires the 78.98% of CPU, whereas the temperature equation only the 15.24%. The increase of the computing time is not only due to the formation of the element matrices and assembly, more costly for the Navier-Stokes equations, but also to the solution of the final algebraic system of equations. The time required for the temperature equation is the 29% of the time needed for the Navier-Stokes equations. This gives an idea of the rapid increase of the computing time with the number of equations of the system using a direct solver, that is what we have employed. Having this in mind and observing that the iterations needed for each time step are mainly due to the nonlinearity of the Navier-Stokes equations, the block

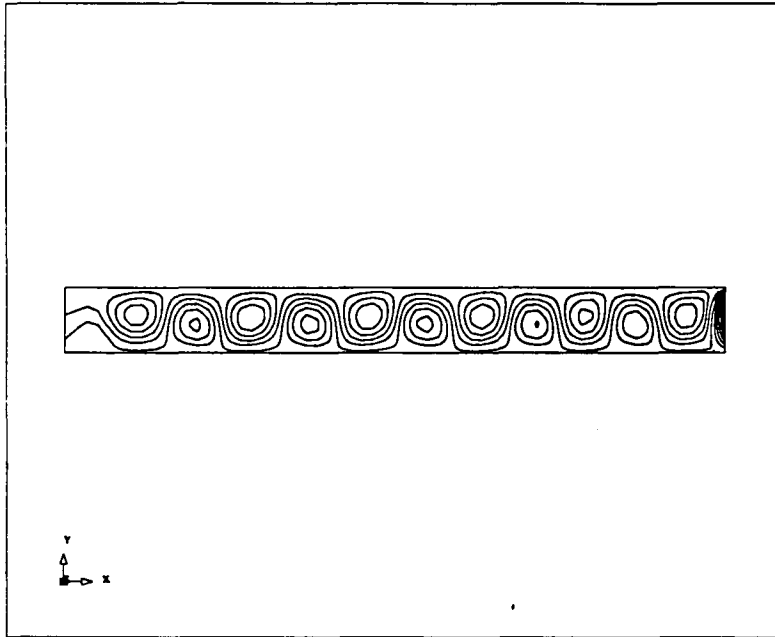


Figure 5.17 Influence of the outflow boundary condition for $t = 1.3$, corresponding approximately to the maximum value of the temperature at the central point (PPF).

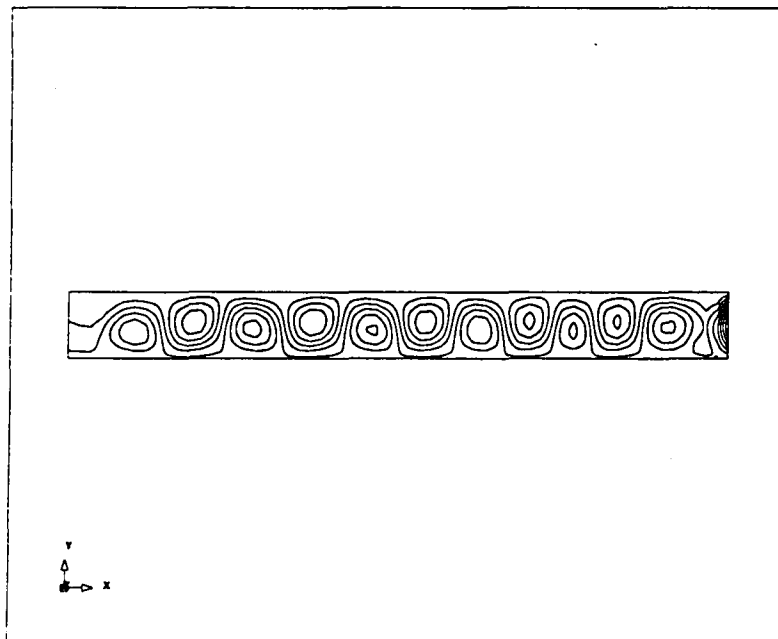


Figure 5.18 Influence of the outflow boundary condition for $t = 2.6$, corresponding approximately to the minimum value of the temperature at the central point (PPF).

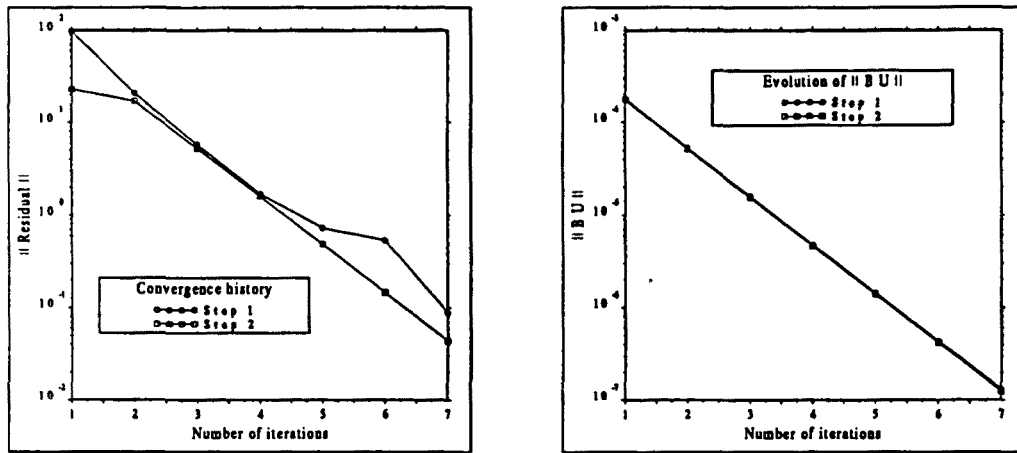


Figure 5.19 Convergence history and evolution of the norm of the incompressibility constraint for the first and second time steps (PPF).

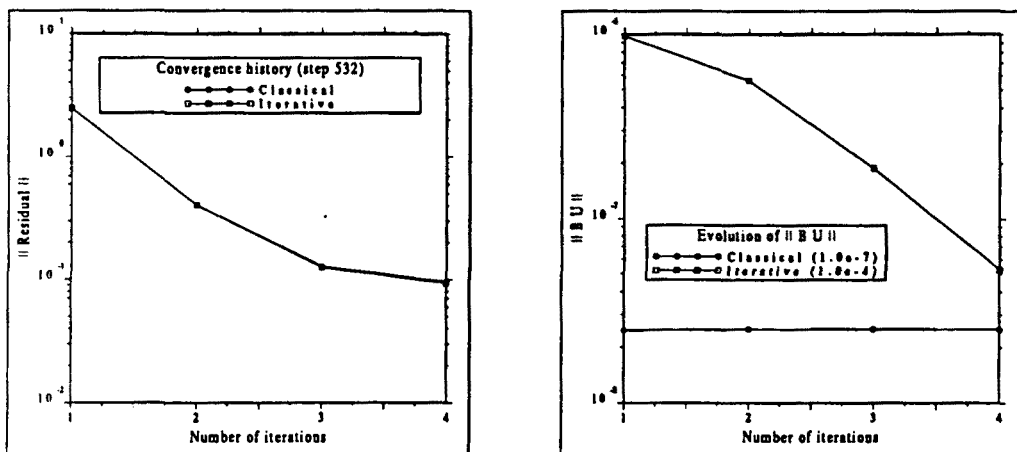


Figure 5.20 Convergence history and evolution of the norm of the incompressibility constraint using the classical penalty method with $\epsilon = 10^{-7}$ and the iterative penalization with $\epsilon = 10^{-4}$ for time step No. 532 (PPF).

iterative algorithm used to uncouple the thermal and mechanical problems seems to be a very efficient procedure.

5.5.2 Transient natural convection of low-Prandtl-number fluids

In this example, the transient convective motion of a fluid enclosed in a square cavity driven by a temperature gradient will be numerically analysed. The left vertical wall is suddenly heated and maintained at a constant temperature, while the right vertical wall is maintained at the initial temperature. Horizontal walls are assumed to be adiabatic, i.e., the zero heat flux boundary condition is prescribed. Homogeneous Dirichlet boundary conditions are prescribed everywhere on the boundary for the velocity.

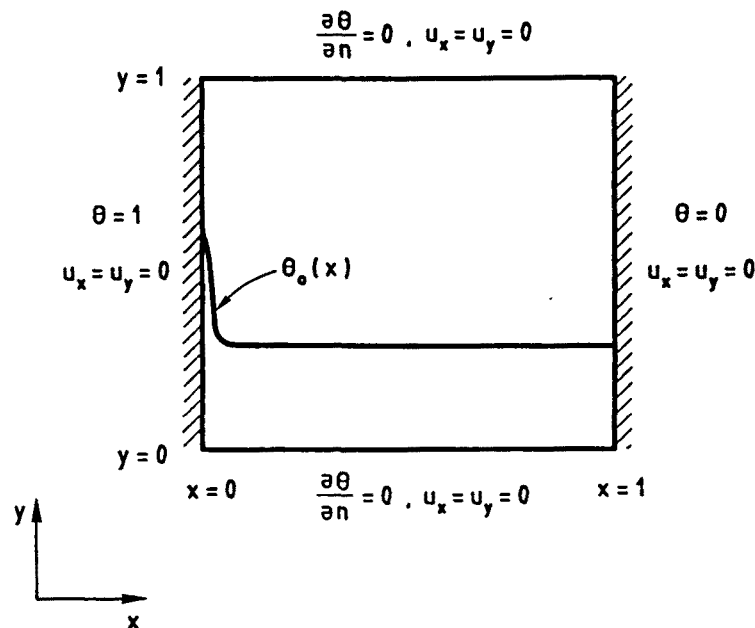


Figure 5.21 Geometry, initial and boundary conditions for the problem of transient natural convection of low-Prandtl-number fluids. Coordinates, velocity and temperature are assumed to be dimensionless.

The problem definition is represented in Figure 5.21. All the variables of the problem have been nondimensionalized using the length of the cavity and the temperature difference between the two vertical walls as reference values for length and temperature, respectively. The reference velocity has been taken as κ/L , as explained in Section 5.2.1. Referring to Eqns. (5.7), the only dimensionless parameters involved in the problem are the Prandtl number Pr and the Rayleigh number Ra or, equivalently, the Grashof number Gr . Numerical results will be presented for $Pr = 0.005$ and the values $Gr = 3 \times 10^6$ and $Gr = 5 \times 10^6$.

The value $Pr = 0.005$ is very small and not often encountered in common fluids. For example, the Prandtl number is 0.71 for air, 7.03 for water and 0.0249 for mercury (at 293 K). Small values of Pr are typical of liquid metals and semiconductors. The problem to be studied now is relevant to the solidification of ingots and casting, crystal growth from melts, materials processing, nuclear reactor safety and other applications (cf. [MV]).

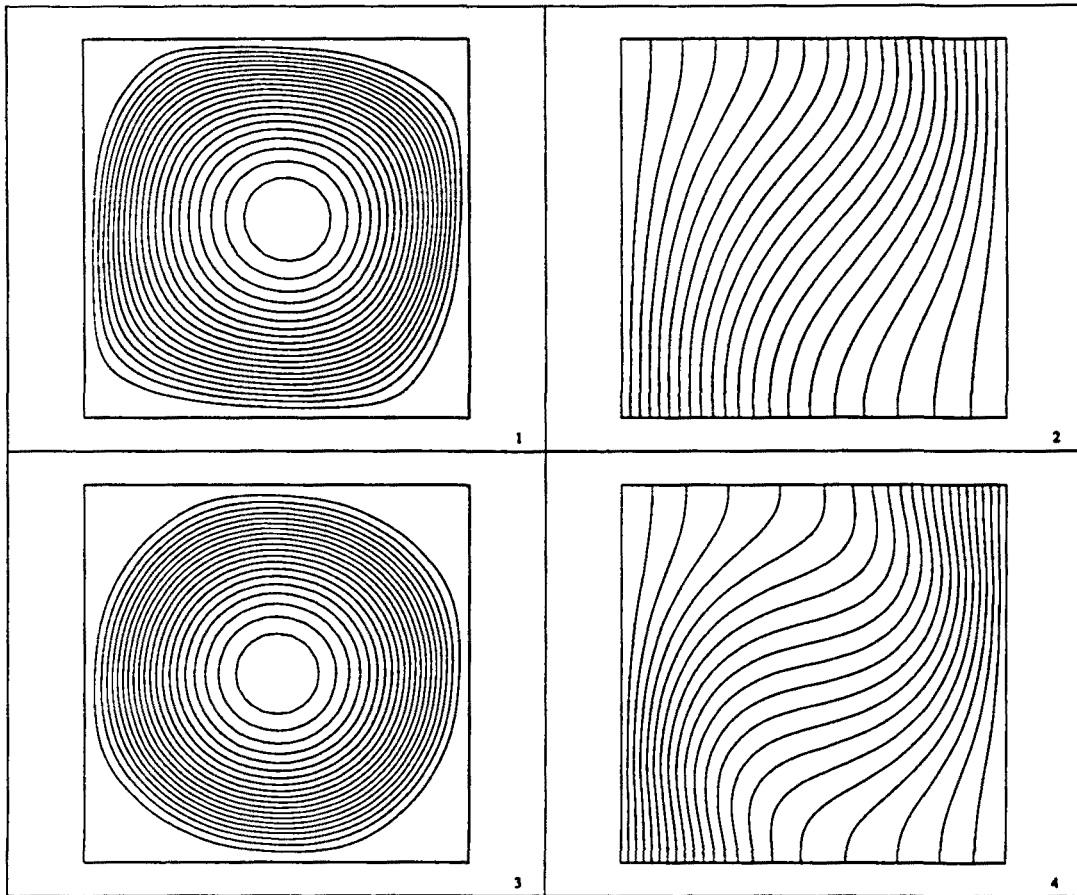


Figure 5.22 Transient evolution of the streamlines and the temperature contours for the problem of natural convection of low-Prandtl-number fluids (LPN), $Gr = 3 \times 10^6$. (1): Streamlines, $t = 0.4$; (2): Temperature contours, $t = 0.4$; (3): Streamlines, $t = 1.2$; (4): Temperature contours, $t = 1.2$.

Although the problem just described is a very popular test for thermally coupled flows when Pr is high, the interest for solving low-Prandtl-number flows is that this problem is not yet well understood. It is found that the flow exhibits a periodic oscillation when the Grashof number exceeds a critical value. In particular, for $Pr = 0.005$ a steady-state solution is obtained for $Gr = 3 \times 10^6$ but the solution bifurcates and for $Gr = 5 \times 10^6$ an oscillatory flow field is found. For further information about this problem the reader is referred to the work of Mohamad & Viskanta [MP], from where this problem has been taken. Our purpose here is to demonstrate the efficiency of the numerical method proposed in this work and also to get more insight in the physics of the problem now considered. A much more detailed information about the recirculation zones at the corners of the cavity and the dynamics of the vorticity than in the above quoted reference will be given.

The numerical strategy employed is as follows. The finite element mesh used to discretize the unit square is the same as in Example 4.1 and shown in Figure 4.6. It consists of 671 Q_2/P_1 elements and 2809 nodal points. The SD formulation has been

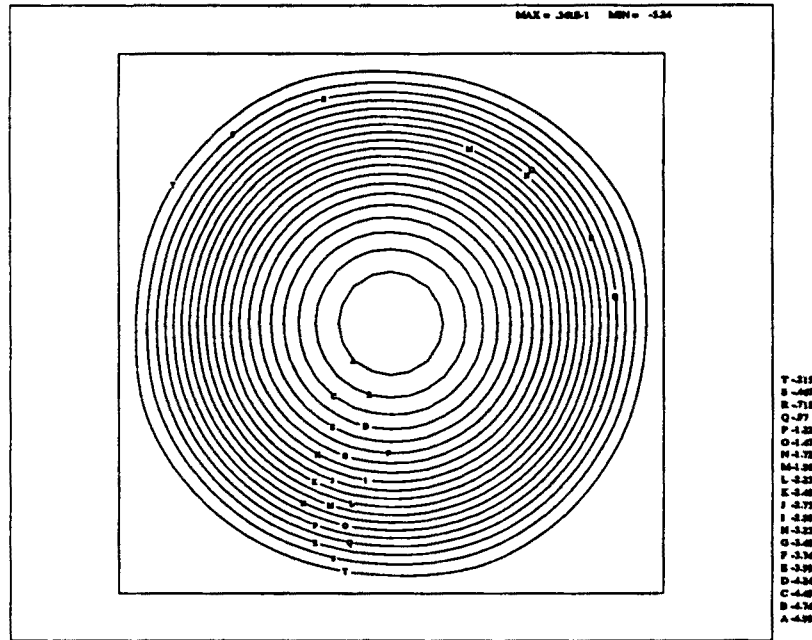


Figure 5.23 Numerical values of the streamfunction when the steady-state has already been reached, $Gr = 3 \times 10^6$ (LPN).

used for both the Navier-Stokes and the energy equation, with $\alpha_0 = 0.5$ and $h_0 = 2$ as upwind factors and length of the parent domain, respectively. The iterative penalization with $\epsilon = 10^{-4}$ has been chosen, yielding a final value of order 10^{-12} for the norm of the discrete velocity divergence in all the time steps. The Navier-Stokes equations have been linearized up to first order, and the Gauss-Seidel block iterative method and Newton-Raphson linearization of the temperature equation have been adopted. The convergence tolerance has been taken as 0.1% in the relative L^2 norm. Based on the results and comments of the previous section, $\theta = 1$ (backward Euler) has been taken for the generalized trapezoidal rule to advance in time.

Let us first discuss the results for $Gr = 3 \times 10^6$ and shown in Figures 5.22 to 5.28 (the abbreviation 'LPN', standing for Low-Prandtl-Number, has been included to identify the problem). It has already been said that in this case a stable steady-state solution is found. The time step size has been taken as $\Delta t = 0.04$, a high value, considering that it is of the same magnitude as the mesh diameter and the backward Euler scheme is only first order accurate. The steady-state solution is completely developed at $t = 6$, time for which results are presented.

Figure 5.22 shows the transient evolution from the motionless flow field to the thermally induced solution. It is observed that the streamlines are initially a little squared (for $t = 0.4$) and evolve to the almost circular shape shown in Figure 5.23 (for $t = 6$). It is also observed how the temperature contours accommodate from the initial constant temperature gradient to the final configuration of Figure 2.25. A detail of the vortices created at the corners of the cavity is shown in Figure 5.24. Two vortices appear at the top right and bottom left corners with similar strength, and only one in the other two corners. Let us remark that the maximum and minimum values for the streamfunction are *exactly* the same as those obtained in Reference [MV] using a much finer mesh ($81 \times 81 = 6561$ grid points) but a finite difference method.

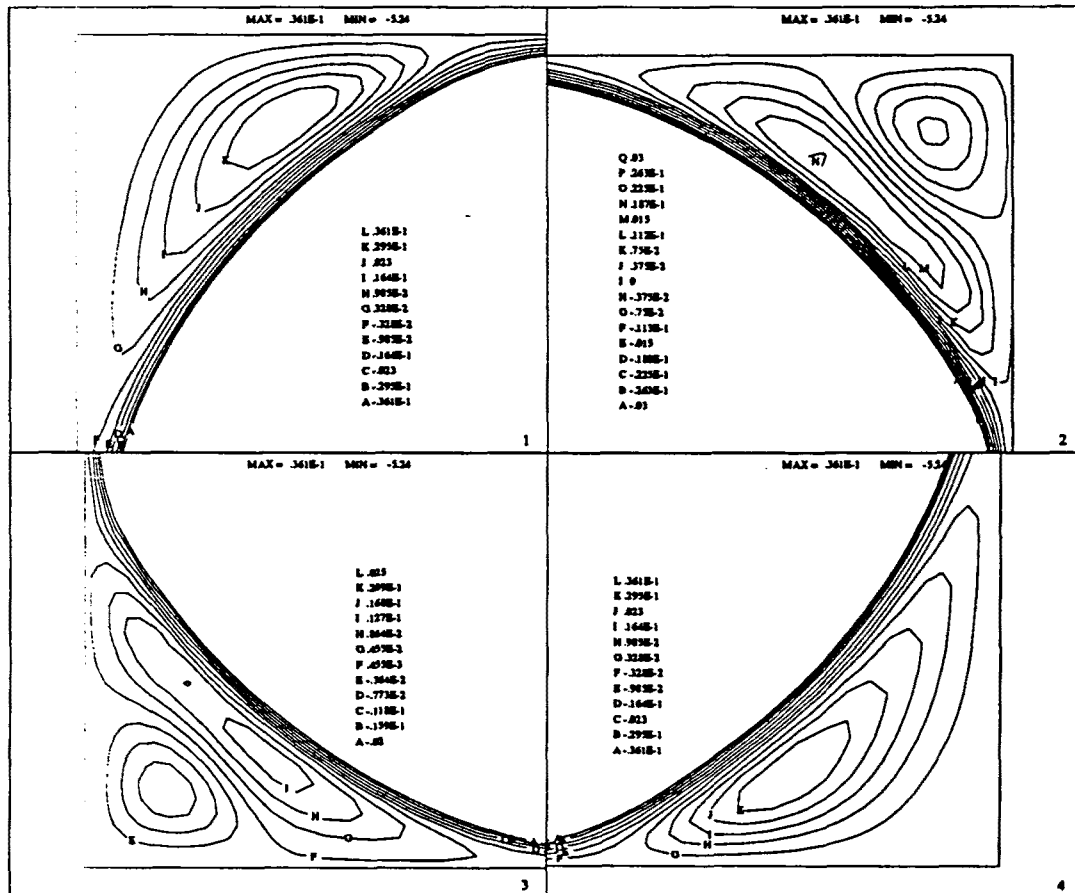


Figure 5.24 Details of the steady-state streamlines at the four corners of the cavity (LPN), $Gr = 3 \times 10^6$. (1): Top left corner; (2): Top right corner; (3): Bottom left corner; (4): Bottom right corner.

The velocity vectors, isobars and vorticity contours are shown in Figures 5.26, 5.27 and 5.28, respectively. It is interesting to observe that pressure gradients are almost constant at the middle of the walls and that high vorticity gradients are generated there. It is argued in Reference [MV] that the instability found for higher values of Gr is originated at the top right vortex. We believe that the sources of instability are these high gradients of vorticity just mentioned. They are due to the fact that the flow has to accommodate from the circular velocity field in the middle of the cavity to a zero velocity at the walls, without smooth transition. Figure 5.26 is illustrative of this situation.

The case $Gr = 5 \times 10^6$ is considered next. Now the time step size has been taken as $\Delta t = 0.004$. The computation has started with the steady-state solution found before, redefining $t = 0$ for $t = 6$. The transient evolution of the x -velocity component at the points of coordinates $(0.995, 0.5)$ and $(0.976, 0.5)$ is depicted in Figure 2.29. It is observed that an oscillatory flow pattern has been developed. The amplitude of the oscillations grows slowly as time goes on. Details of the streamlines, temperature contours, isobars and vorticity contours are shown for $t = 3.2$ in Figures 5.30 to 5.34. From the former it is seen that now two more secondary vortices appear at the top left

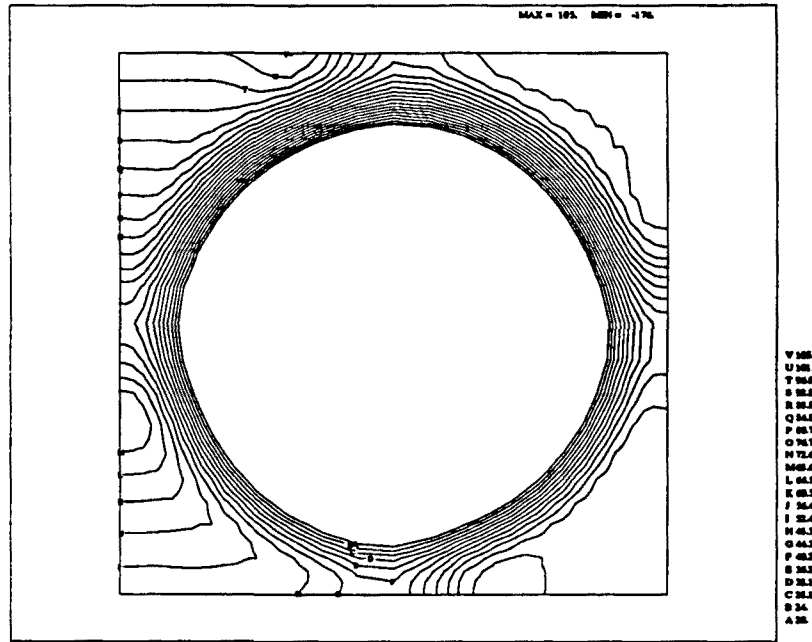


Figure 5.27 Steady-state pressure contours, $Gr = 3 \times 10^6$ (LPN).

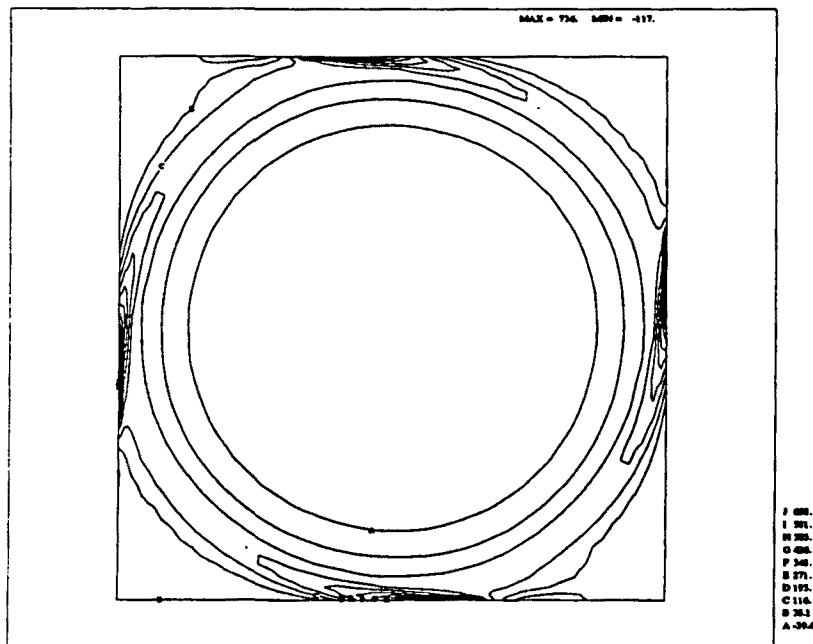


Figure 5.28 Steady-state vorticity contours, $Gr = 3 \times 10^6$ (LPN).

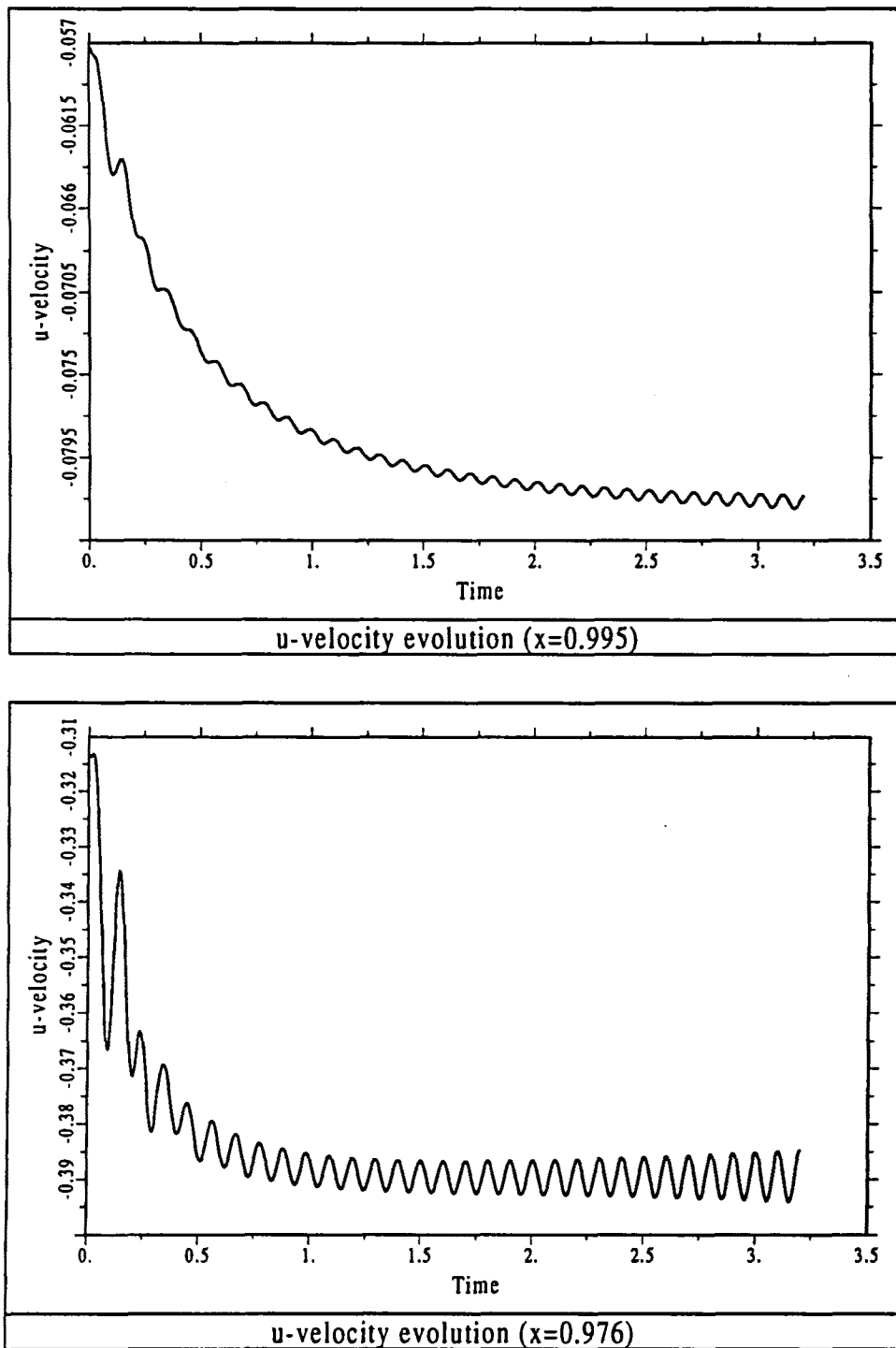


Figure 5.29 Transient evolution of the x -velocity component at points $(0.995, 0.5)$ and $(0.976, 0.5)$, $Gr = 5 \times 10^6$ (LPN).

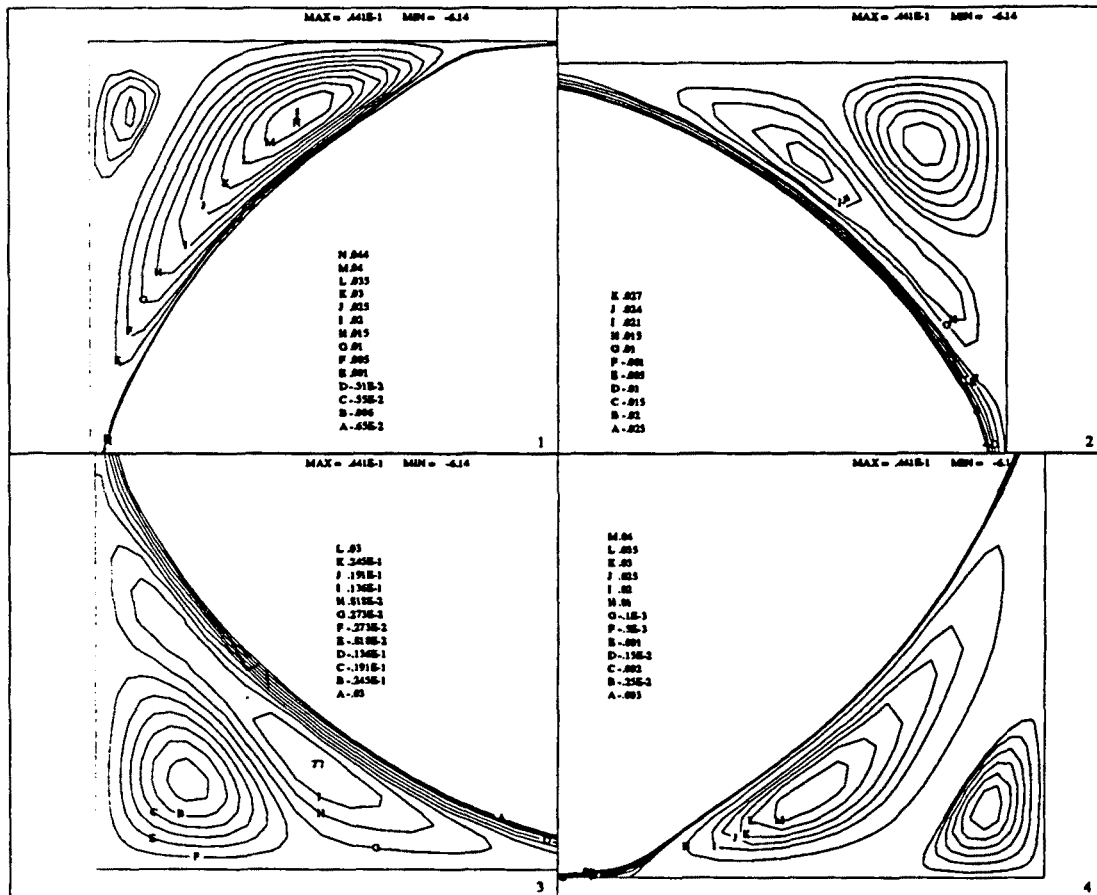


Figure 5.30 Details of the streamlines at the four corners of the cavity for $t = 3.2$, $Gr = 5 \times 10^6$ (LPN). (1): Top left corner; (2): Top right corner; (3): Bottom left corner; (4): Bottom right corner.

and bottom right corners. The center of the elongated vortex at the other two corners oscillates around the position found for $Gr = 3 \times 10^6$ (compare with Figure 5.24).

Finally, let us mention that the CPU time needed per iteration has been 45.483 seconds, of which the 83.41% are required by the Navier-Stokes solver and the 16.13% for the solution of the temperature equation. The solution of the linear algebraic system for this problem needs the 25.37% of what is needed for the Navier-Stokes equations. Between two and three iterations have been needed per time step for the case $Gr = 3 \times 10^6$ with $\Delta t = 0.04$ and only one in most of the time steps for the case $Gr = 5 \times 10^6$ with $\Delta t = 0.004$. This again indicates that using a block iterative method for thermally coupled problems is a good option.

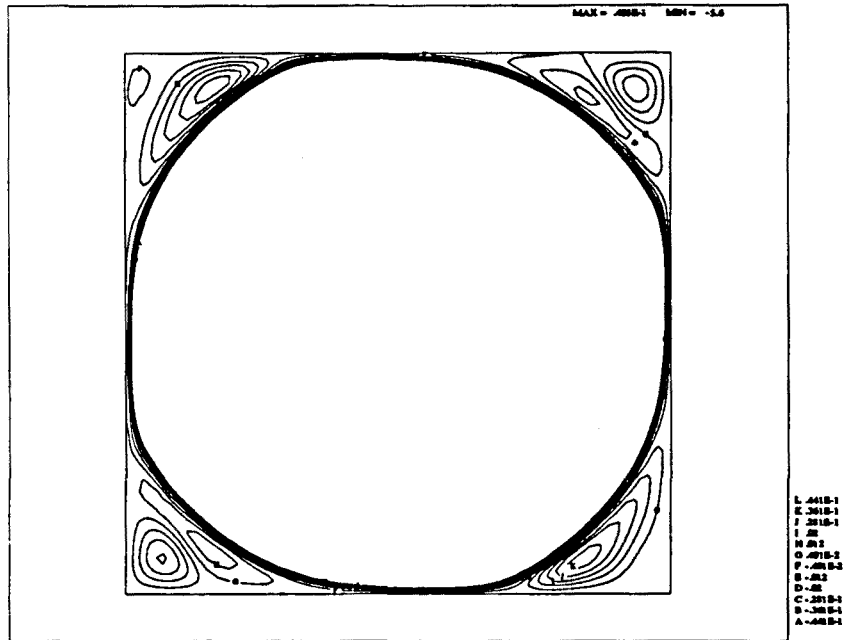


Figure 5.31 Global picture of the streamlines at the corners cavity for $t = 3.2$, $Gr = 5 \times 10^6$ (LPN).

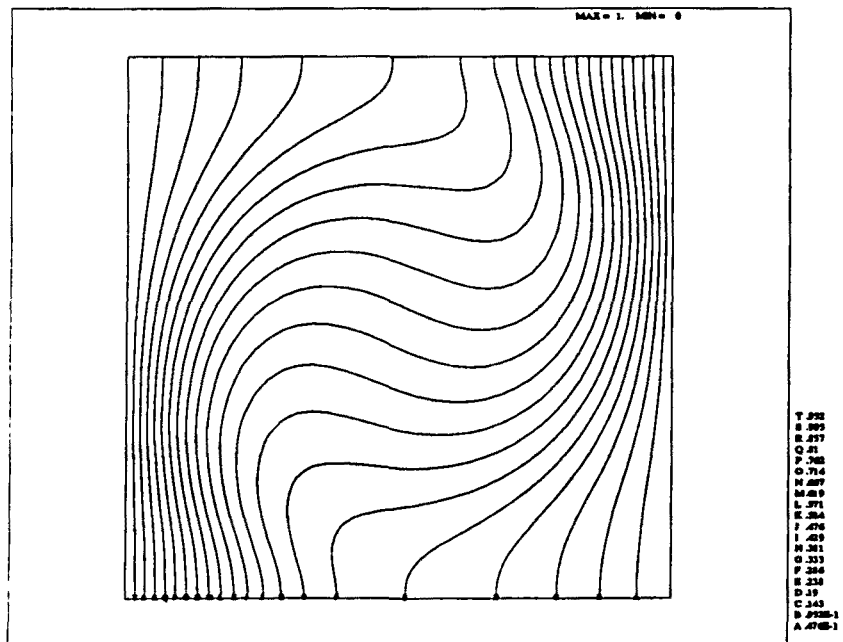


Figure 5.32 Temperature contours for $t = 3.2$, $Gr = 5 \times 10^6$ (LPN).

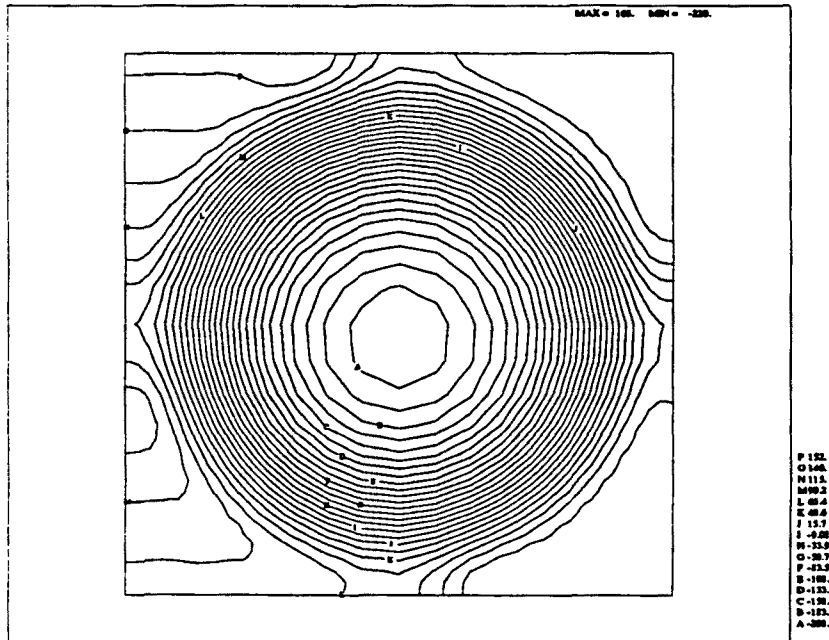


Figure 5.33 Pressure contours for $t = 3.2$, $Gr = 5 \times 10^6$ (LPN).

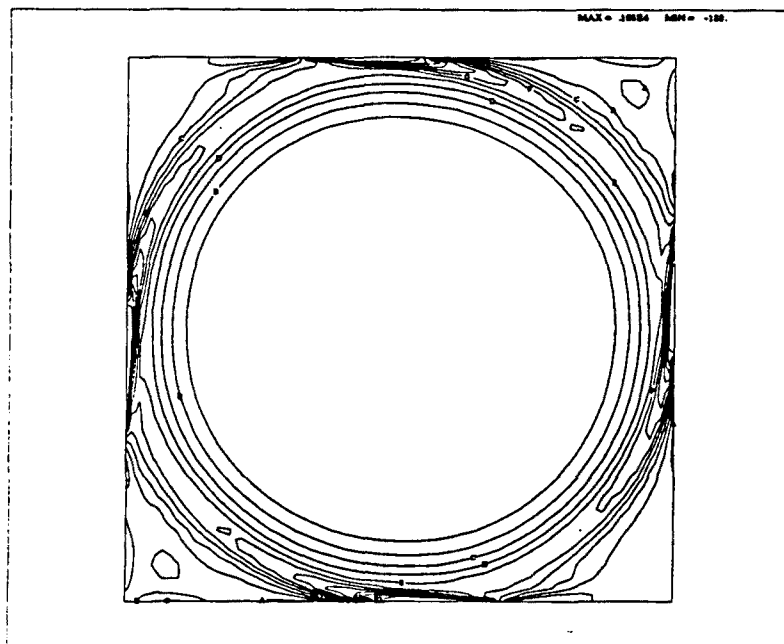


Figure 5.34 Vorticity contours for $t = 3.2$, $Gr = 5 \times 10^6$ (LPN).

5.5.3 The 4:1 plane extrusion of a power-law fluid

In this section we present some numerical results obtained for the well-known 4:1 plane extrusion problem. This is a very popular test for non-Newtonian flows, since all the flow features that characterize these fluids are present in this problem. It is also used as a test to check error estimators and adaptive remeshing techniques (see, e.g., Reference [DT]).

The geometry and the boundary conditions are depicted in Figure 5.35. The variation of the viscosity and the components of the velocity will be given for sections AA, BB and CC indicated in this Figure.

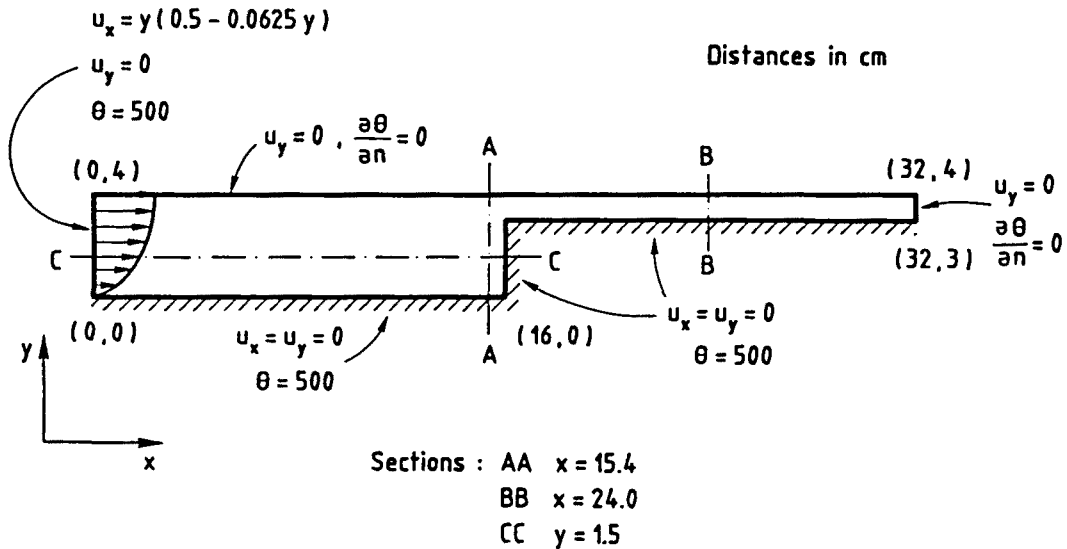


Figure 5.35 Geometry and boundary conditions for the 4:1 plane extrusion of a power-law fluid.

Here, Eqns. (5.62) modelling the creeping flow of nonlinear materials will be solved numerically. The finite element mesh employed for the space discretization is composed of 525 $Q2/P1$ elements (biquadratic interpolation for the velocity, piecewise linear pressure), with a total of 2201 nodal points. There are 15 elements in the y -direction from the coordinates $y = 3$ to $y = 4$ and only 12 from $y = 0$ to $y = 3$. The concentration of elements in the former zone is needed if one wants to reproduce accurately the shear thinning effect of fluids whose viscosity obeys the power law that we shall consider now, given by

$$\mu = K_0 [4I_2(\epsilon)]^{(n-1)/2} \exp\left(\frac{\beta}{\vartheta}\right) \tag{5.81}$$

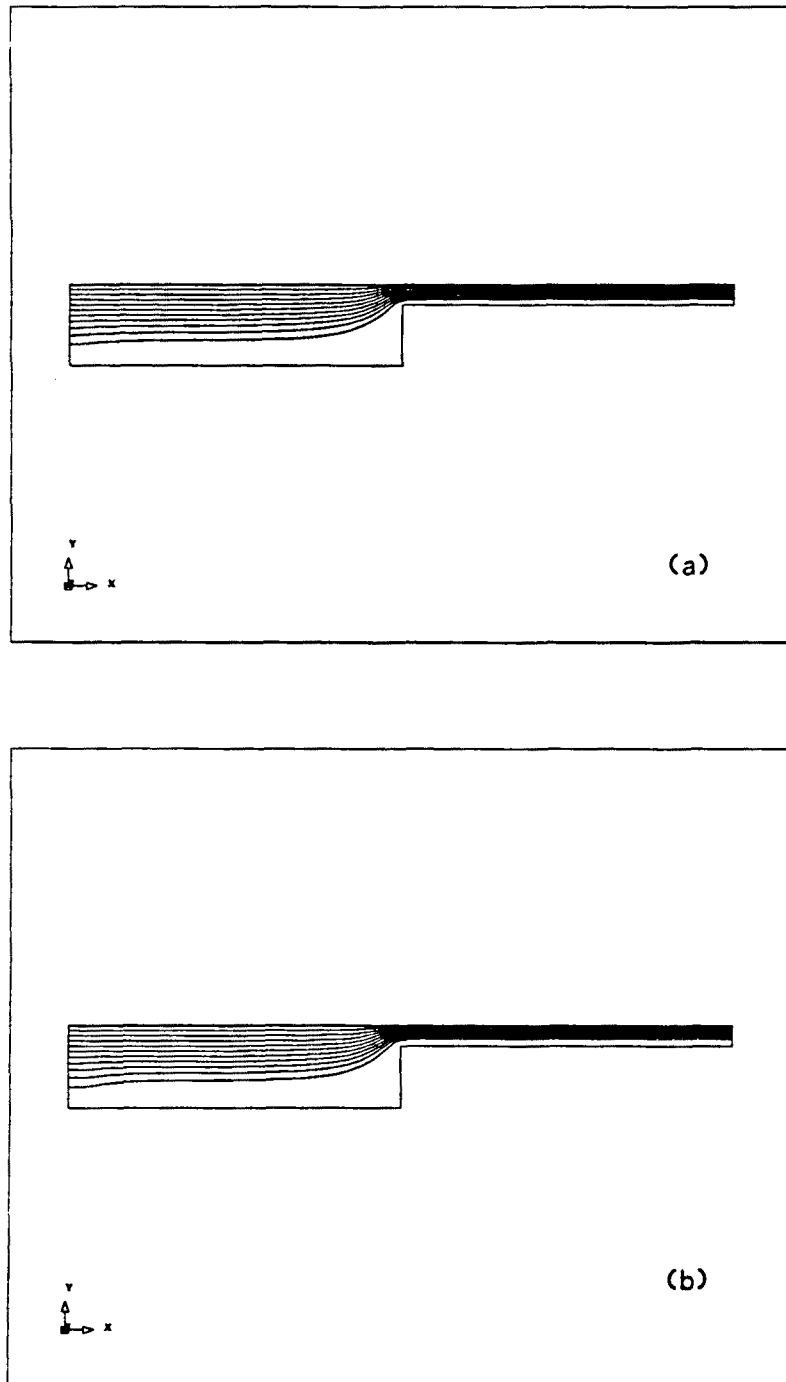


Figure 5.36 Streamlines for $\beta = 0$ (a) (thermally independent viscosity) and for $\beta = 2 \times 10^3$ (b) (thermally coupled flow). Observe the different curvature near the inflow vertical wall and the different gradient of the streamfunction in the exit channel (EPL).

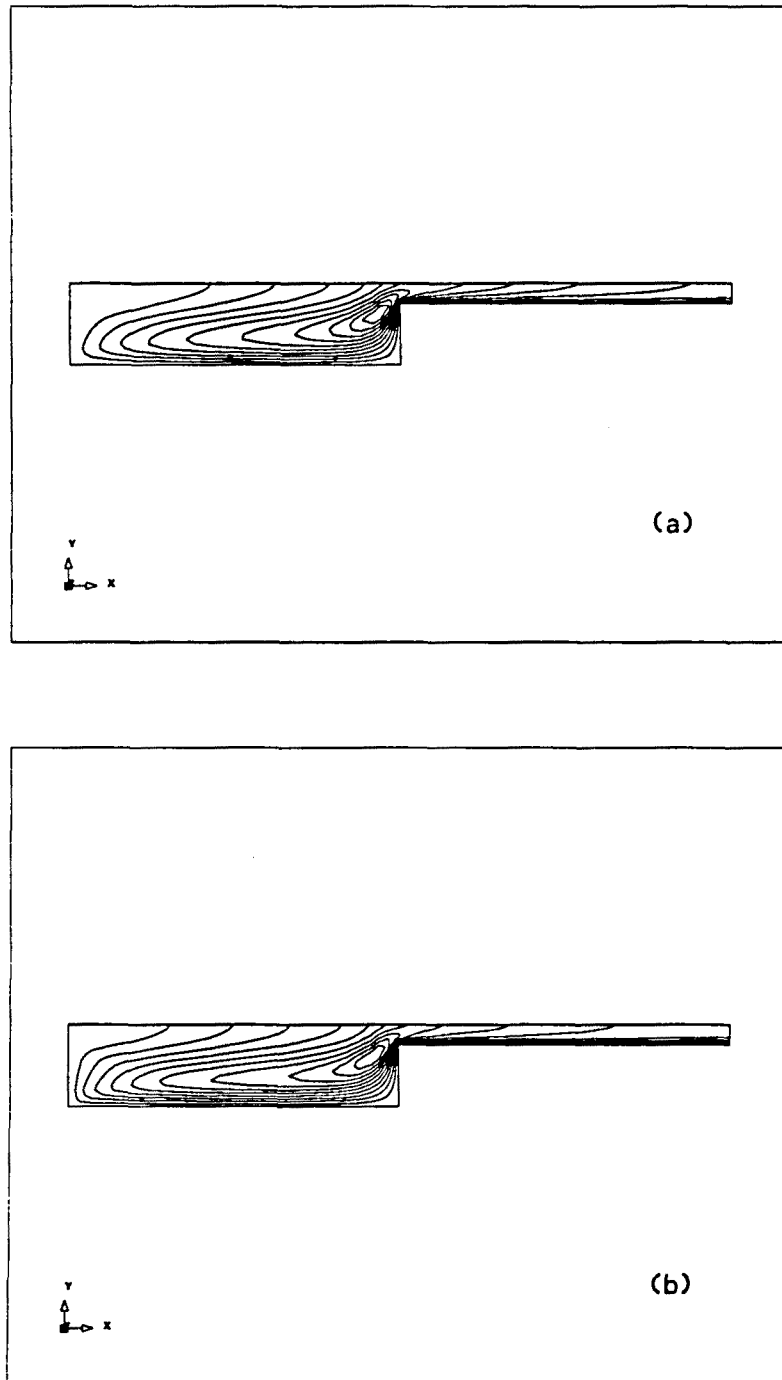


Figure 5.37 Temperature contours for $\beta = 0$ (a) (thermally independent viscosity) and for $\beta = 2 \times 10^3$ (b) (thermally coupled flow) (EPL).

In this expression, K_0 , n and β are physical constants (K_0 is the material consistency and n the rate sensitivity) and ϑ is the temperature. The power law given by Eqn. (5.50) has been combined with an exponential thermal dependence as dictated

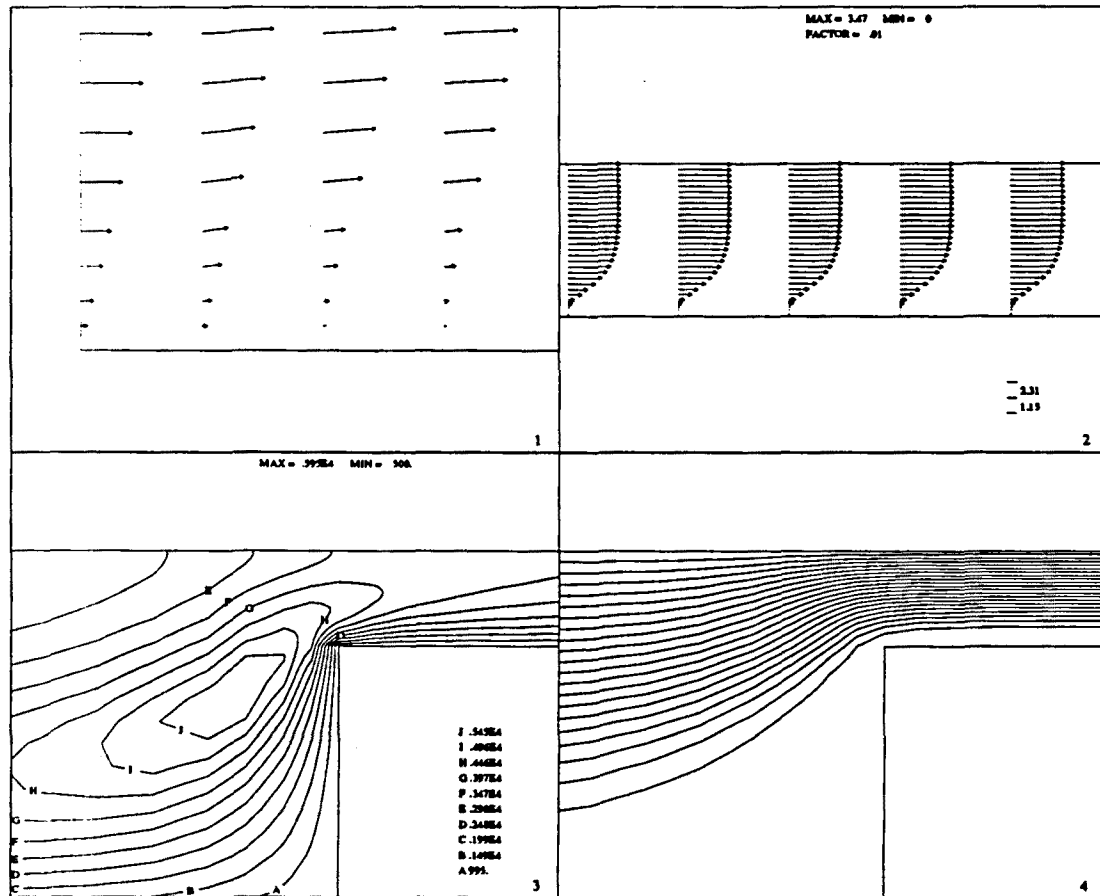


Figure 5.38 Details of the flow for $\beta = 0$ (EPL). (1): Accomodation of the parabolic velocity profile, corresponding to a Newtonian fluid, to the non-Newtonian velocity profile near the bottom left corner. Observe that velocity vectors have and positive y -component; (2): Velocity vectors in the exit channel. The shear thinning effect is apparent; (3): Temperature contours near the corner of the step; (4): Streamlines near the corner of the step.

by Eqn. (5.60), with a suitable identification of the physical parameters.

The values of the physical constants we have used are (all in SI units): $\rho = 1200$ (density), $c_p = 10$ (specific heat), $k = 2$ (thermal conduction coefficient), $K_0 = 10^6$ (material consistency) and $n = 0.2$ (rate sensitivity). For this value of n the effect of the non-constant viscosity is very pronounced. The values of the thermal properties are non-physical, but they have been used to accentuate the temperature effect on the viscosity. Numerical experiments have also been conducted with larger values of n , in which case convergence is easier to achieve. Since the expression of the viscosity (5.81) tends to infinity when $I_2(\varepsilon)$ tends to zero, we have introduced a cut-off value $\mu_c = 10^{11}$ for μ . The values of the viscosity for the converged solutions are always below this limit, except in isolated points.

The physical results for this problem are presented if Figures 5.36 to 5.43 (coordinates are given in decimeters. The problem has been identified with the acronym 'EPL', standing for plane Extrusion of a Power-Law fluid). Figure 5.36 shows the streamlines

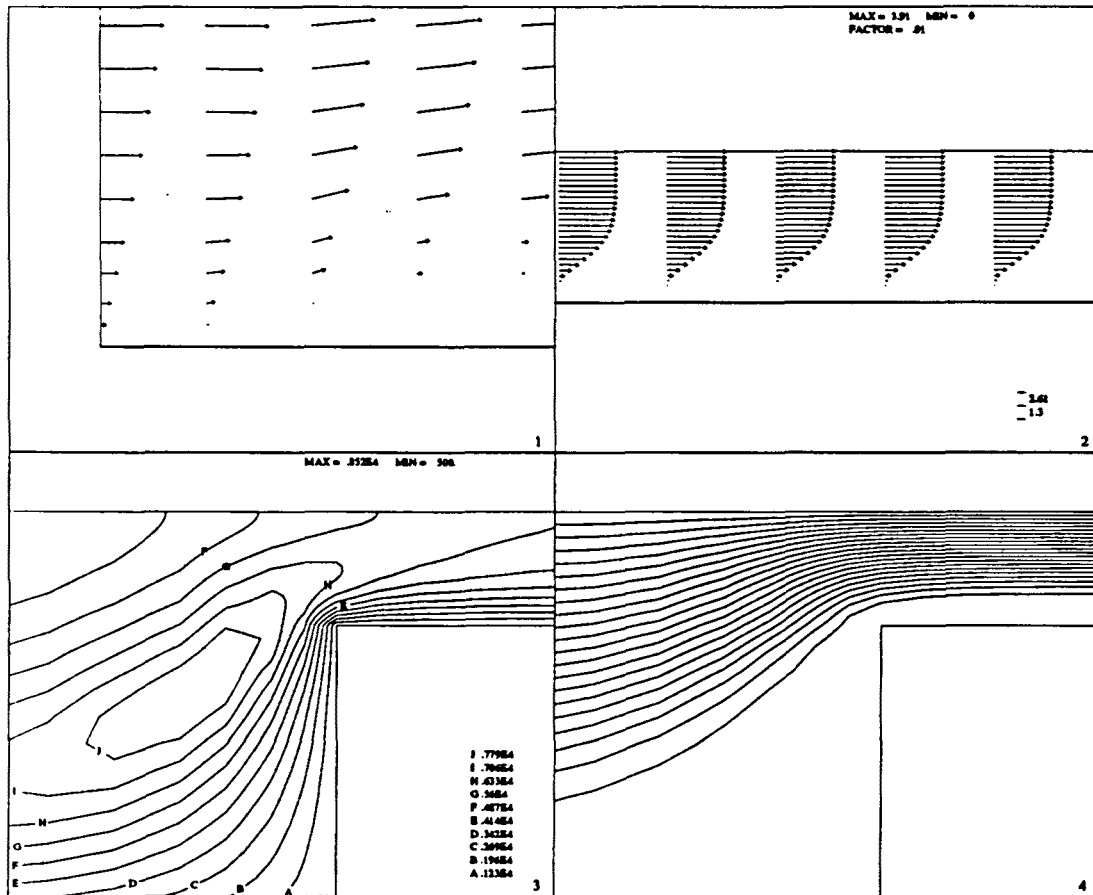


Figure 5.38 Details of the flow for $\beta = 2 \times 10^3$ (EPL). (1): Accomodation of the parabolic velocity profile, corresponding to a Newtonian fluid, to the non-Newtonian velocity profile near the bottom left corner; (2): Velocity vectors in the exit channel. The shear thinning effect is more pronounced than for $\beta = 0$; (3): Temperature contours near the corner of the step; (4): Streamlines near the corner of the step.

for $\beta = 0$ (thermally uncoupled flow) and for $\beta = 2 \times 10^3$, where the effect of the temperature on the viscosity (and thus on the velocity) is apparent. The temperature contours are plotted in Figure 5.37. From Eqn. (5.63) it is clear that the temperature will rise where the internal mechanical work is higher, that is, in the zones with high strain rate. This happens near the corner $(x, y) = (16, 3)$.

A detail of the flow features for both $\beta = 0$ and $\beta = 2 \times 10^3$ is shown in Figures 5.38 and 5.39, respectively. It is observed that the effect of the temperature on the viscosity for the latter case results in an even more pronounced shear thinning.

Figures 5.40, 5.41 and 5.42 show the variation of the x -velocity component and the viscosity along the sections AA , BB and CC indicated in Figure 5.35. The approximation of the viscosity in the AA section is not very good for $0 \leq y \leq 3$. As it has been already said, the discretization there is poor. However, the variation of the x - and y -velocity components (Figures 5.40 and 5.43) is smooth, since the shear thinning effect is not important in this section.

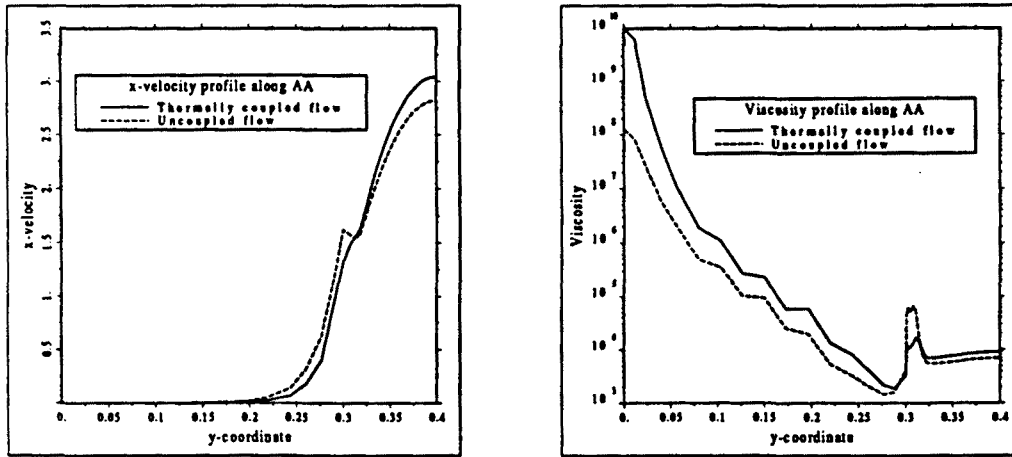


Figure 5.40 x -velocity and viscosity profiles along section AA for $\beta = 0$ and $\beta = 2 \times 10^3$ (EPL).

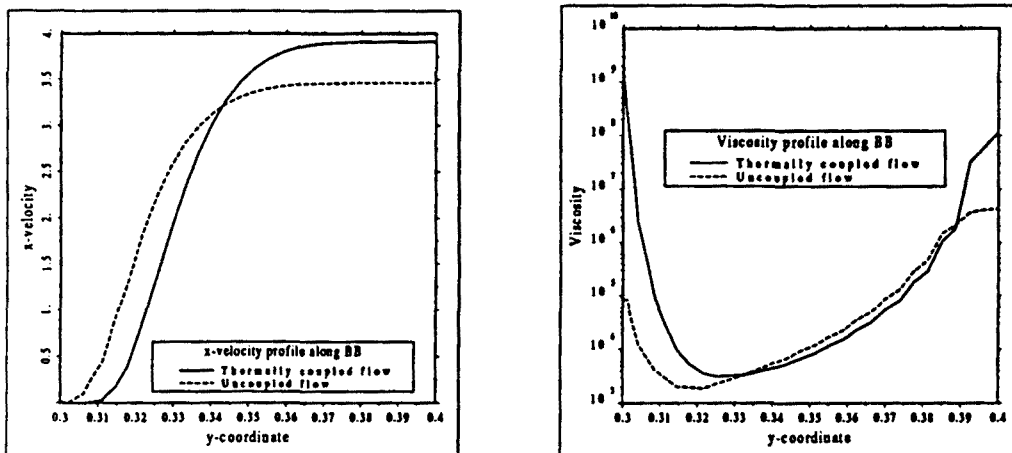


Figure 5.41 x -velocity and viscosity profiles along section BB for $\beta = 0$ and $\beta = 2 \times 10^3$ (EPL).

Let us discuss now the performance of the iterative penalization. For values of β between 0 and 2×10^3 the convergence history of the numerical simulation is similar. However, for larger values of β lack of convergence can occur. We have failed to obtain converged solutions for $\beta = 5 \times 10^3$, both for the standard penalty method and for the iterative version. As proposed in Reference [ZOH], under-relaxation techniques may be required when the dependence of the viscosity on the temperature is very pronounced. The convergence of the algorithm will be discussed in the case in which $\beta = 2 \times 10^3$, that is, when the viscosity depends on the temperature (thermally coupled flow).

Figure 5.44 shows the evolution of the discrete L^2 norm of the velocity residuals over the norm of the actual velocity (in %). As usual, this has been taken as the

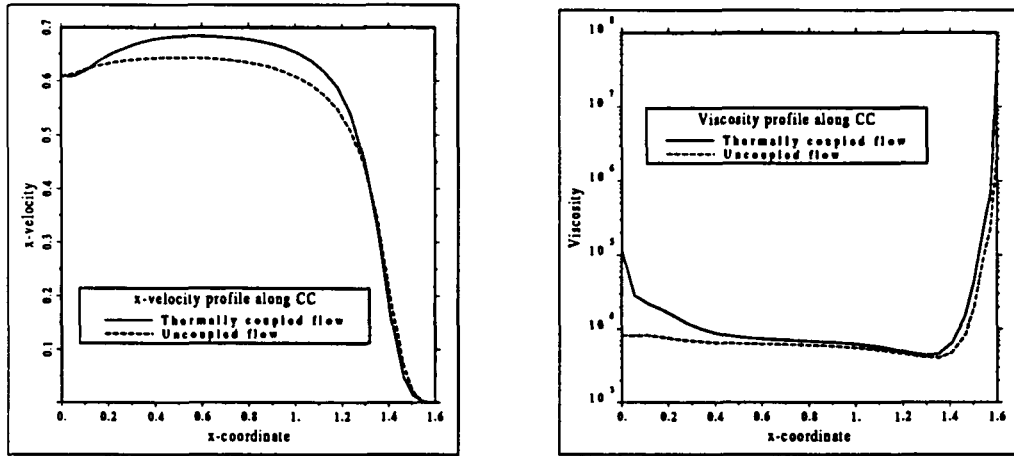


Figure 5.42 x -velocity and viscosity profiles along section CC for $\beta = 0$ and $\beta = 2 \times 10^3$ (EPL).

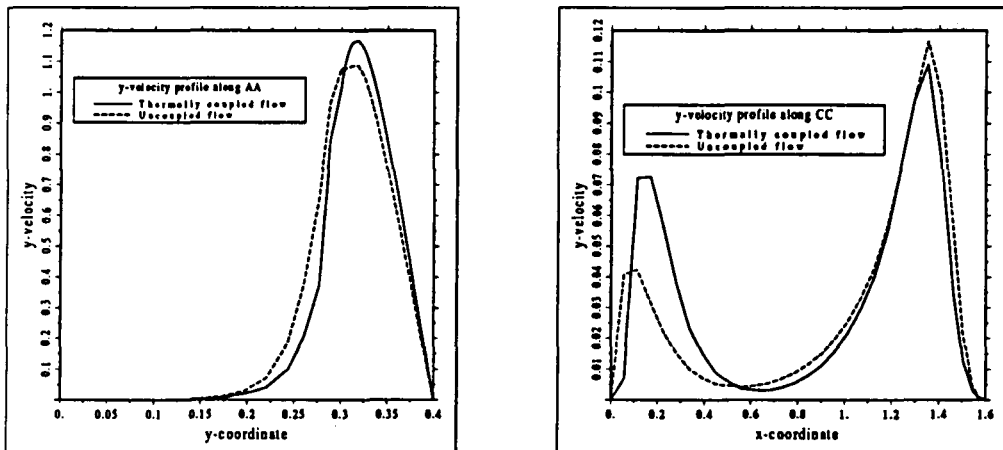


Figure 5.43 y -velocity profiles along sections AA and CC for $\beta = 0$ and $\beta = 2 \times 10^3$ (EPL).

parameter to decide whether convergence has been achieved or not. Both the curves corresponding to the classical and the iterative penalty methods have been plotted. Here, the penalty parameter that has been used is $\epsilon = 10^{-12}$. In the first iteration, the viscosity is set to its cut-off value. Thus, the effective initial guess for the second iteration is the Newtonian solution with this viscosity. A real non-Newtonian behavior will be first encountered in this second iteration and from there onwards iterations are required to reach the prescribed convergence tolerance. However, we see that one more iteration is needed if the iterative penalization is employed. The explanation we give is that in this method the second pass of the algorithm uses the Newtonian pressures obtained in the first one, and thus the complete non-Newtonian approximation is not

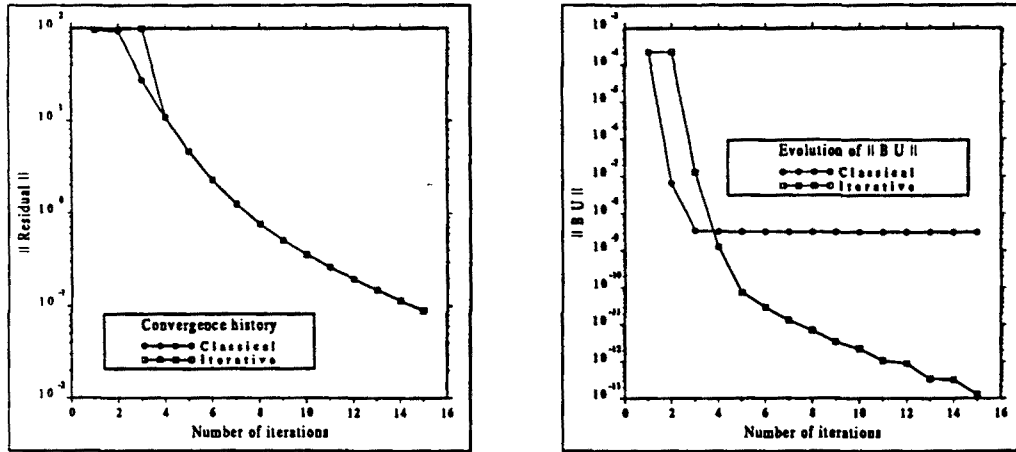


Figure 5.44 Convergence history and evolution of the incompressibility constraint for $\epsilon = 10^{-12}$ and $\beta = 2 \times 10^3$ (EPL).

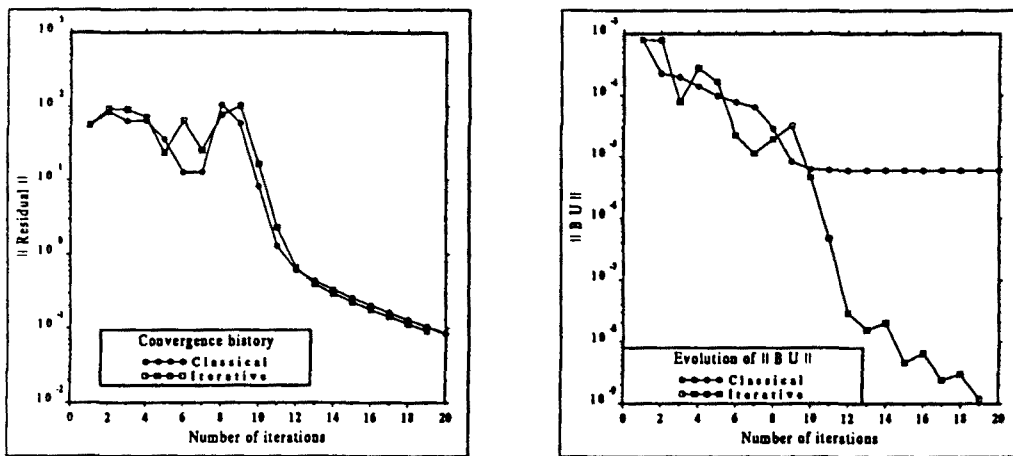


Figure 5.45 Convergence history and evolution of the incompressibility constraint for $\epsilon = 10^{-9}$ and $\beta = 2 \times 10^3$ (EPL).

obtained until the third iteration. In any case, it is interesting to observe that the final convergence rate and the number of iterations needed to achieve convergence have not been deteriorated because of the iterative penalization.

The important issue is to determine how well the incompressibility constraint has been approximated. The evolution of $\|BU\|$ ($B = G^T$, with the notation used earlier) as the iterative procedure goes on has been plotted in the second box of Figure 5.44 (we have normalised this norm by dividing it by $N_{tp}^{1/2}$). Observe that this value keeps constant for the classical penalty method and that it decreases uniformly up to a value of order 10^{-13} in 15 iterations if the iterative penalization is used. One might think that the value of order 10^{-8} obtained with the classical penalty method is a good enough

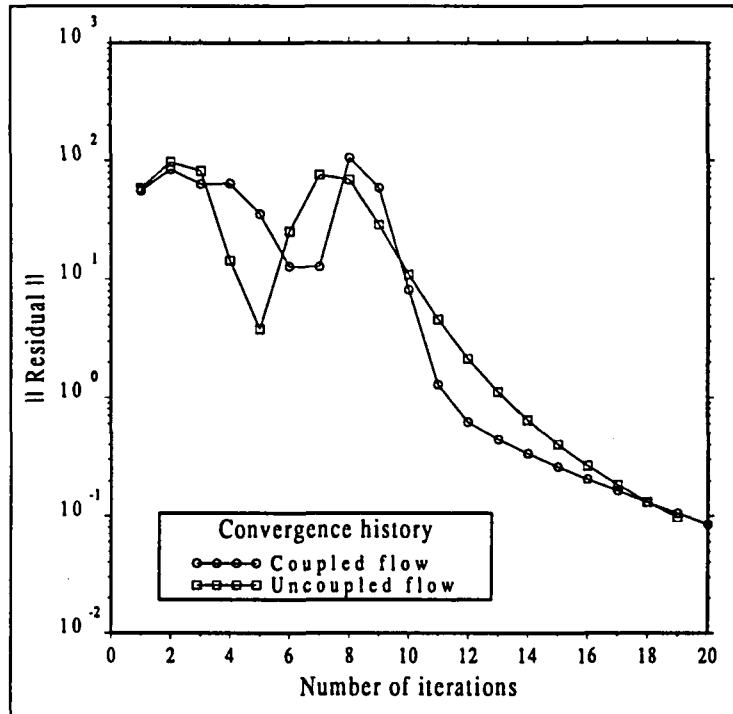


Figure 5.46 Convergence history for $\epsilon = 10^{-9}$ for thermally coupled flow ($\beta = 2 \times 10^3$) and uncoupled flow ($\beta = 0$) (EPL).

approximation. However, this may be somehow misleading, since the smallest value of the final viscosity, say μ_1 , is of order 10^3 , and thus $\epsilon \approx 10^{-9} \mu_1^{-1}$ whereas the largest viscosity value, say μ_2 , is of order 10^9 and then $\epsilon \approx 10^{-3} \mu_2^{-1}$. Thus, the parameter $\bar{\epsilon}$ introduced earlier is of order 10^{-9} in the low viscosity zones and of order 10^{-3} in the high viscosity zones. Recalling that the approximation of the incompressibility constraint is driven by $\bar{\epsilon}$, we may expect a much better satisfaction of this constraint in the low viscosity regions than in the zones where the viscosity is high. If smaller penalty parameters are employed, the solution is affected by the ill-conditioning of the stiffness matrix, even for the direct solver we use. For $\epsilon = 10^{-16}$ this ill-conditioning is so important that the algorithm fails to converge.

The same experiments discussed above have been performed using a penalty parameter $\epsilon = 10^{-9}$ and the results are presented in Figure 5.45 (convergence history and evolution of the L^2 norm of the discrete divergence). The conclusions are similar to the previous case. Observe now that oscillations are found for the first eight iterations and then the iterates converge uniformly. The reason for this behavior is the high value of ϵ , that is of the same order as μ_2^{-1} , the inverse of the maximum viscosity, and 100 times higher than μ_c^{-1} , the inverse of the cut-off value. To see that this behavior of the iterative procedure is not due to the block iterative algorithm (see Box 5.1), the convergence history for $\beta = 0$ (uncoupled flow) and $\beta = 2 \times 10^3$ (coupled flow) has been plotted in Figure 5.46. In both cases oscillations appear for the first eight iterations, although the final convergence rate of the uncoupled flow is slightly higher than for the coupled flow for the last iterations.

Concerning the computational cost of the simulation, the CPU required per iteration has been 24.41 seconds for $\beta = 2 \times 10^3$ and 18.58 seconds for $\beta = 0$. For the first case, the solution of the temperature equation needs the 20.46% of CPU time and the solution of the momentum equations (Stokes problem) the 74.85%. Most of the computing time now is needed to solve the linear algebraic system (about a 75%), both for the Stokes and the energy equation. Updating the physical properties and performing the smoothing technique explained in Chapter 4 in order to obtain nodal values of the viscosity and the pressure is inexpensive: these operations only need the 1.6% of CPU time.

5.6 Summary and conclusions

The basic numerical method described in Chapter 4 has been applied here to solve thermally coupled flows and flows of non-Newtonian fluids. The first issue to be considered is the algorithm used to couple the energy equation with the Navier-Stokes equations. This has been done by means of a block iterative coupling. Although its use seems to be quite 'natural', an effort has been made to analyse how can it be interpreted and to place it in the general framework of iterative algorithms. In particular, it has been proved for Newtonian fluids that the solution of the energy equation using the actual iterate of the velocity field can be thought of as a Newton-Raphson linearization of the convective term in this equation coupled with a Gauss-Seidel block iterative technique. In more general situations, numerical experiments have indicated that this coupling between the thermal and mechanical problems is very efficient from the computational standpoint.

The extension of the numerical methods used for the Navier-Stokes equations to the problems considered here has proved to be effective. In particular, the Streamline Diffusion method has proved to work very well when div-stable velocity-pressure finite element interpolations are used. Numerical results have demonstrated that a consistent Petrov-Galerkin weighting yields very accurate solutions, without any oscillations.

Concerning the time stepping algorithm, the Crank-Nicolson method has been found to be very sensitive to the convergence tolerance adopted within each time step. Using the block iterative technique, the coupling between the thermal and mechanical problems is only accomplished up to a certain tolerance and thus a certain stability limit will exist for the time step size. Using the backward Euler scheme no stability problems have been encountered in the numerical tests.

Perhaps the behavior of iterative penalization is the most salient result. It was derived and analysed for a much simpler problem than the one considered here (stationary Navier-Stokes equations with constant viscosity and under conditions that ensure uniqueness of solution) but happens to perform very well for thermally coupled flows and, what is more important, for non-Newtonian fluids. In this last case, the classical penalization is inappropriate due to the high variation of the viscosity in the fluid domain, and therefore either the incompressibility constraint is poorly approximated or the final stiffness matrix is ill-conditioned. The iterative penalty method allows to circumvent both problems using relatively high values of the penalty parameter.

References

- [BN] J. Baranger and K. Najib. Analyse numérique des écoulements quasi-Newtoniens dont la viscosité obéit à la loi puissance ou à la loi de carreau. *Numer. Math.*, vol. 58 (1990), 35–49
- [BL] J. Boland and W. Layton. An analysis of the finite element method for natural convection problems. *Numer. Meth. for Partial Diff. Eqns.*, vol. 2 (1990), 115–126
- [BLL] Y.A. Bukhman, V.I. Lipatov, A.I. Litvinov, B.I. Mitelman and Z.P. Shulman. Rheodynamics of nonlinear viscoplastic media. *Journal of Non-Newtonian Fluid Mechanics*, vol. 10 (1982), 215–233
- [BP] M.B. Bush and N. Phan-Thien. Drag force on a sphere in creeping motion through a Carreau model fluid. *Journal of Non-Newtonian Fluid Mechanics*, vol. 16 (1984), 303–313
- [CWJ] G.F. Carey, K.C. Wang and W.C. Joubert. Performance of iterative methods for Newtonian and generalized Newtonian flows. *Int. J. Numer. Meth. Fluids*, vol. 9 (1989), 127–150
- [CO] J.L. Chenot and E. Oñate (eds.). *Modelling of metal forming processes*. Proceedings of the Euromech 233 Colloquium, Sophia Antipolis, France (Kluwer Academic Publishers, 1988).
- [CC] L. Choplin and P.J. Carreau. Excess pressure losses in a slit. *Journal of Non-Newtonian Fluid Mechanics*, vol. 9 (1981), 119–146
- [CCO] R. Codina, M. Cervera and E. Oñate. A penalty finite element method for non-Newtonian creeping flows. *CIMNE Report Num. 13* (1991) (Submitted to *Int. J. Numer. Meth. Engrg.*)
- [Cu] C. Cuvelier. Résolution numérique d'un problème de control optimal d'un couplage des équations de Navier-Stokes et celle de la chaleur. *Calcolo*, vol. 15 (1978), 345–379
- [CSS] C. Cuvelier, A. Segal and A. van Steenhoven. *Finite element methods and Navier-Stokes equations* (Reidel, 1986).
- [DT] H.H. Dannelongue and P.A. Tanguy. An adaptive remeshing technique for non-Newtonian fluid flow. *Int. J. Numer. Meth. Engrg.*, vol. 30 (1990), 1555–1567
- [DK] H.D. Doctor and N.L. Kalthia. Spline collocation in the flow of non-Newtonian fluids. *Int. J. Numer. Meth. Engrg.*, vol. 26 (1988), 413–421
- [DR] A. Dutta and M.E. Ryan. A study of parison development in extrusion blow molding. *Journal of Non-Newtonian Fluid Mechanics*, vol. 10 (1982), 235–256
- [EP] G. Evans and S. Paolucci. The thermoconvective instability of plane Poiseuille flow heated from below: A proposed benchmark solution for open boundary flows. *Int. J. Numer. Meth. Fluids*, vol. 11 (1990), 1001–1013
- [GR] K.S. Gage and W.H. Reid. The stability of thermally stratified plane Poiseuille flow. *J. Fluid Mech.*, vol. 33 (1968), 21–32
- [Gu] M. Gunzburger. *Finite element methods for viscous incompressible flows* (Academic Press, 1989).
- [He] J.C. Heinrich. A finite element model for double diffusive convection. *Int. J. Numer. Meth. Engrg.*, vol. 20 (1984), 447–464
- [HTB] P. Hurez, P.A. Tanguy and F.H. Bertrand. A finite element analysis of die swell with pseudoplastic and viscoplastic fluids. *Comput. Meths. Appl. Mech. Engrg.*, vol. 86 (1991), 87–103

- [IOS] V.P. Isachenko, V.A. Osipova and A.S. Sukomel. *Heat transfer* (Mir, 1977).
- [Jo] D.D. Joseph. *Stability of fluid motions* (Springer-Verlag, 1976).
- [Ke] R. Kessler. Nonlinear transition in three-dimensional convection. *J. Fluid Mech.*, vol. 174 (1987), 359–379
- [LL] L.D. Landau and E.M. Lifshitz. *Mécanique des fluides* (Mir, 1969).
- [Li] J.L. Lions. *Quelques méthodes de résolution des problèmes aux limites non linéaires* (Dunod, 1968).
- [LLH] T.J. Liu, H.M. Lin and C.N. Hong. Comparison of two numerical methods for the solution of non-Newtonian flow in ducts. *Int. J. Numer. Meth. Fluids*, vol. 8 (1988), 845–861 (1988)
- [MV] A.A. Mohamad and R. Viskanta. Transient natural convection of low-Prandtl-number fluids in a differentially heated cavity. *Int. J. Numer. Meth. Fluids*, vol. 13 (1991), 61–81
- [Oñ] E. Oñate. *Plastic flow in metals, I: Thermal coupling behaviour, II: Thin sheet forming*. Ph. D. Thesis. University College of Swansea (1978).
- [OH] D.R.J. Owen and E. Hinton. *Finite elements in Plasticity* (Pineridge Press, 1980).
- [PF] K.C. Park and C.A. Felippa. Partitioned analysis of coupled systems. In: *Computational methods for transient analysis*. T. Belytschko and T.J.R. Hughes (eds.) (Elsevier, 1983).
- [SY] A.J.M. Shih and H.T.Y. Yang. Experimental and finite element simulation methods for rate-dependent metal forming processes. *Int. J. Numer. Meth. Engrg.*, vol. 31 (1991), 345–367
- [SB] J. Stoer and R. Burlisch. *Introduction to Numerical Analysis* (Springer-Verlag, 1983).
- [Ta] R.I. Tanner. *Engineering Rheology* (Clarendon Press, 1985).
- [TNB] R.I. Tanner, R.E. Nickell and R.W. Bilger. Finite element methods for the solution of some incompressible non-Newtonian fluid mechanics problems with free surfaces. *Comput. Meths. Appl. Mech. Engrg.*, vol. 6 (1975), 155–174
- [TWZ] E.G. Thompson, R.D. Wood, O.C. Zienkiewicz and A. Samuelson (eds.). *Nu-miform 89*. Proceedings of the 3rd International Conference on Numerical Methods for Industrial Forming Processes, Fort Collins, Colorado, USA (A. Balkema, 1989).
- [TTK] D.M. Tidd, R.W. Thatcher and A. Kaye. The free surface flow of Newtonian and non-Newtonian fluids trapped by surface tension. *Int. J. Numer. Meth. Fluids*, vol. 8 (1988), 1011–1027
- [WTS] K. Wisniewski, E. Turska, L. Simoni and B.A. Schrefler. Error analysis of staggered predictor-corrector scheme for consolidation of porous media. In: *Finite elements in the 90's*. E. Oñate, J. Periaux, A. Samuelson (eds.) (Springer-Verlag/CIMNE, 1991)
- [Zi] O.C. Zienkiewicz. Coupled problems—A simple time-stepping procedure. *Comm. Appl. Numer. Meth.*, vol. 1 (1985), 233–239
- [ZG] O.C. Zienkiewicz and P.N. Godbole. Flow of plastic and viscoplastic solids with special reference to extrusion and forming processes. *Int. J. Numer. Meth. Engrg.*, vol. 8 (1974), 3–16
- [ZJO] O.C. Zienkiewicz, P.C. Jain and E. Oñate. Flow of solids during forming and extrusion: some aspects of numerical solution. *Int. J. Numer. Meth. Engrg.*, vol. 14 (1978), 15–38

-
- [ZOH] O.C. Zienkiewicz, E. Oñate and J.C. Heinrich. A general formulation for coupled thermal flow of metals using finite elements. *Int. J. Numer. Meth. Engrg.*, vol. 17 (1981), 1497-1514
- [ZMS] A.I. Zografos, W.A. Martin and J.E. Sunderland. Equation of properties as a function of temperature for several fluids. *Comput. Meth. Appl. Mech. Engrg.*, vol. 61 (1987), 177-187

CHAPTER 6

MOULD FILLING SIMULATION

6.1 Introduction

This chapter will be devoted to present a specific application of the numerical tools developed previously: the numerical simulation of mould filling processes. Mould filling is an integral part of the casting process, an ancient metal forming technique. It starts with the pouring of a molten metal into a mould until it is filled and it is concluded when the solid nature of the metal is restored. The complete numerical simulation of these processes involves modelling of mould filling, prediction of thermal stresses in a solidifying material and micro-macro modelling in order to predict material micro-structure. Besides the inherent difficulty to model all these physical phenomena, another problem arises because of the identification of material properties, for which delicate experiments are needed. Mould filling as the first stage of the casting process will be the subject of this chapter.

The main difficulty for simulating the flow of a molten metal in a mould is the modelling of free surfaces. Most of the numerical approaches to this problem have been limited to simple geometries, due to the high computational cost of this simulation and that numerical models have been mainly based on finite difference techniques. Because of the available computer potential, it has become possible to deal with more complicated geometries for which finite element models are especially well suited. The representation of feeders, gating systems, risers and the overall mould geometry does not offer any difficulty using finite elements. Proper evaluation of the position of the melt during the transient analysis is the most important problem.

The model that we shall use here to track the free surface of the fluid is based upon the pseudo-concentration technique, which employs a fixed mesh. The moving fluid may fill the elements partially or fully. The version of this method we shall use is due to Thompson [Th1], [Th2], [TS], although it has also been used under the names *volume of fluid method* (VOF) [HN] or *saturation method* [SW]. The basic idea is to introduce a scalar function which is advected according to the velocity flow field obtained from the solution of the Navier-Stokes equations. This function is defined on the whole computational domain. A certain isovalue contour is used to define the front of the 'real' fluid. The unfilled region is assumed to be occupied by a fictitious material whose physical properties are such that its motion does not affect the dynamical behavior of the fluid under consideration. To fix ideas, we shall consider that this fictitious material

is air.

There are other popular methods to track free surfaces. One of them is the updated Lagrangian approach, in which the mesh moves with the fluid [Zi]. The main disadvantage of this method is that the mesh becomes distorted during the analysis and eventually remeshing is needed. Only when small distortions occur the method is successful [LDD], [ZJO], [ZOH]. A second approach is the Arbitrary Lagrangian-Eulerian method (see, e.g., References [Hu], [SFD]). In this method, a velocity is assigned to the mesh which is independent of the fluid velocity except at the boundary, and is chosen in order to minimize the mesh distortion and/or the convective terms. This requires some *a priori* knowledge of the fluid flow.

The pseudo-concentration technique has been used by several authors to follow free surfaces of creeping flows and viscoplastic flows in the context of metal forming processes, such as extrusion, forging or rolling. See, e.g., References [AI], [AID], [DP], [TS]. For applications of this method to mould filling, see References [DGB], [HS], [LUC].

This chapter is organized as follows. The basic pseudo-concentration technique is described in Section 6.2, whereas Section 6.3 is concerned with some problems encountered when this method is employed. These problems are either practical, arising from the computer implementation of the method, or conceptual. The coupling between the Navier-Stokes and energy equations with the free surface tracking is considered in Section 6.4. Practical numerical examples are presented in Section 6.5. The numerical simulation of two mould filling problems is first discussed. The final example is the classical lamination of a metal flat plate, now analysing the transient mechanical and thermal evolution since the metal contacts the roll until it leaves it.

6.2 The pseudo-concentration method

6.2.1 Basic formulation

The basic idea of the pseudo-concentration technique is to define a scalar function, say $\psi(\mathbf{x})$, over the computational domain Ω in such a manner that its value at a certain point $\mathbf{x} \in \Omega$ indicates the presence or absence of fluid. This function may be considered as a fictitious fluid property. For instance, we may assign the value 1 to regions where the fluid has already entered and the value 0 to air-filled regions. The position of the fluid front will be defined by the isovalue contour $\psi(\mathbf{x}) = \psi_c$, where $\psi_c \in [0, 1]$ is a critical value defined *a priori*. We usually take $\psi_c = 0.5$. This value is immaterial if ψ is a true step function, but is needed in the finite element discretization and for the smoothing to be described later.

The conservation of the pseudo-concentration in any control volume $V_t \subset \Omega$ which is moving with the velocity field \mathbf{u} leads to

$$\frac{d}{dt} \int_{V_t} \psi \, d\Omega = 0$$

If we further assume that ψ is smooth and \mathbf{u} is divergence-free, this implies that

$$\partial_t \psi + (\mathbf{u} \cdot \nabla) \psi = 0 \quad \text{in } \Omega, \, t \in (0, T) \quad (6.1)$$

where, as usual, $(0, T)$ denotes the time interval where the problem is to be solved. Equation (6.1) is hyperbolic and therefore boundary conditions for ψ have to be specified at the inflow boundary, that is,

$$\psi(\mathbf{x}, t) = \bar{\psi}(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_{inf}, \quad t \in (0, T) \quad (6.2)$$

where

$$\Gamma_{inf} := \{\mathbf{x} \in \partial\Omega \mid \mathbf{u} \cdot \mathbf{n} < 0\}$$

and $\bar{\psi}$ is a given function. Finally, an initial condition of the form

$$\psi(\mathbf{x}, 0) = \psi_0(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (6.3)$$

has to be appended to (6.1)–(6.2), $\psi_0(\mathbf{x})$ being chosen in order to define the initial position of the fluid front.

Solving problem (6.1)–(6.3) the position of the fluid will be identified by the values $\psi(\mathbf{x}, t) > \psi_c$ and the position of the air by $\psi(\mathbf{x}, t) < \psi_c$.

6.2.2 Numerical solution of the pseudo-concentration problem

The numerical techniques introduced in the previous chapters will be applied to the numerical solution of problem (6.1)–(6.3). Time derivatives will be discretized using the generalized trapezoidal rule and the Streamline Diffusion (SD) formulation will be employed for the space discretization.

The time discretization of Eqn. (6.1) leads to the following problem: Given $\psi^0(\mathbf{x}) = \psi_0(\mathbf{x})$, for $n = 1, 2, \dots, N$ find $\psi^n(\mathbf{x})$, approximation to $\psi(\mathbf{x}, t^n)$, such that

$$\psi^n + \theta \Delta t (\mathbf{u}^n \cdot \nabla) \psi^n = \psi^{n-1} - (1 - \theta) \Delta t (\mathbf{u}^{n-1} \cdot \nabla) \psi^{n-1} \quad (6.4)$$

After choosing a suitable finite element partition $\{\Omega^e\}$, $e = 1, \dots, N_{el}$, of the domain Ω , the SD method applied to Eqn. (6.4) leads to the variational equations

$$\begin{aligned} \int_{\Omega} \phi_h \psi_h^n d\Omega + \theta \Delta t \int_{\Omega} \phi_h (\mathbf{u}_h^n \cdot \nabla) \psi_h^n d\Omega + \sum_{e=1}^{N_{el}} S_f^{n,e}(\mathbf{u}_h, \psi_h; \phi_h) \\ = \int_{\Omega} \phi_h \psi_h^{n-1} d\Omega - (1 - \theta) \Delta t \int_{\Omega} \phi_h (\mathbf{u}_h^{n-1} \cdot \nabla) \psi_h^{n-1} d\Omega \end{aligned} \quad (6.5)$$

where the test functions ϕ_h and the trial solutions ψ_h^n belong to $H^1(\Omega)$, the former satisfying homogeneous boundary conditions on Γ_{inf} and the latter the essential boundary conditions (6.2). The SD term in Eqn. (6.5) is given by

$$\begin{aligned} S_f^{n,e}(\mathbf{u}_h, \psi_h; \phi_h) := \int_{\Omega^e} \tau_f^e(\mathbf{u}_h^n \cdot \nabla) \phi_h [\psi_h^n - \psi_h^{n-1} \\ + \theta \Delta t (\mathbf{u}_h^n \cdot \nabla) \psi_h^n + (1 - \theta) \Delta t (\mathbf{u}_h^{n-1} \cdot \nabla) \psi_h^{n-1}] d\Omega \end{aligned} \quad (6.6)$$

The intrinsic time τ_f^e is computed as explained in Chapter 1 (Eqn. (1.36)) using a Péclet number $\gamma = \infty$ (see Box 1.1) to compute the upwind function.

Let us denote by M_f the ‘mass’ (or Gramm) matrix for the pseudo-concentration interpolation and by J the matrix arising from the convective term in Eqn. (6.5)

(considering both the Galerkin and the SD contributions). The matrix version of Eqn. (6.5) will read as follows:

$$\mathbf{M}_f \cdot \Psi^n + \theta \Delta t \mathbf{J}(\mathbf{U}^n) \cdot \Psi^n = \mathbf{M}_f \cdot \Psi^{n-1} - (1 - \theta) \Delta t \mathbf{J}(\mathbf{U}^{n-1}) \cdot \Psi^{n-1} \quad (6.7)$$

the capital letter Ψ denoting the vector of nodal unknowns of the pseudo-concentration function. The dependence of matrix \mathbf{J} on the velocity has been explicitly indicated.

Remarks 6.1

- (1) The parameter θ of the generalized trapezoidal rule may be set different to that employed for the Navier-Stokes or the energy equations. In fact, when ψ is a step function or with a high gradient at the fluid front, the backward Euler scheme ($\theta = 1$) is inappropriate due to its high dissipation, even though it may be used for the Navier-Stokes and energy equations. In this case, the Crank-Nicolson scheme ($\theta = 1/2$) should be employed. However, this problem does not appear if the pseudo-concentration is a smooth function, since the position of the critical contour ψ_c will be advected properly, because the error of the backward Euler scheme is basically an amplitude error and not a phase error.
- (2) In our calculations we have chosen for the pseudo-concentration ψ the same finite element interpolation as for the components of the velocity field and the temperature. □

6.3 Some numerical techniques

6.3.1 General considerations

The use of the pseudo-concentration method described above provides a basic technique to track free-surfaces of viscous incompressible flows, although several problems appear when it is implemented in a computer code.

The first problem encountered is merely for post-processing the results. If the no-slip boundary condition is prescribed for the velocity field, the pseudo-concentration values for points of the finite element discretization located at the boundary of the computational domain will never be advected and therefore the final value obtained for them will be given by the initial condition $\psi_0(\mathbf{x})$. Assume that this initial value is zero. If fluid enters and occupies the neighboring nodal points, located at a distance h from the boundary, the pseudo-concentration value for them will be 1. When the discrete function ψ_h is interpolated, the critical contour ψ_c will be placed between the boundary and the contiguous points. In particular, for $\psi_c = 0.5$ the predicted position of the front will be at a distance $h/2$ from the boundary.

A possible way to artificially overcome this problem is to modify the pseudo-concentration values of the boundary nodal points. We have implemented the following method. Let ψ_m be the mean value of the function ψ for an element Ω^e adjacent to the boundary. The condition $\psi_m \geq \psi_c$ will indicate that most of the nodes of the element have already been filled. In this situation, the value of the pseudo-concentration for a node located at the boundary, ψ_b , is modified as follows:

$$\psi_b \leftarrow \psi_b + \rho(\psi_m - \psi_b) \quad (6.8)$$

where $0 \leq \rho \leq 1$. The constant ρ may be adjusted in order to control when the boundary nodes have to be considered part of the fluid or part of the air. As time advances, the application of (6.8) will yield a value 1 for ψ_b if this procedure enters the calculation, although it may also be used as a post-processing facility.

There are two more problems to be considered for the implementation of the pseudo-concentration technique. One of them is the choice of the function ψ . If we take a step function, as indicated before, numerical problems may be encountered when it is transported. We have seen in Chapter 1 that small oscillations in the vicinity of sharp gradients still remain using the SD formulation. These oscillations may propagate and yield to distorted front shapes, especially near corners. Since the basic idea of the method does not depend on the choice of the function ψ , it is preferable to use a smooth function instead of one with abrupt changes. The smoothing technique we employ will be discussed below. Nevertheless, we have found that if the peaks encountered when dealing with a step function are just eliminated for each time step, an accurate tracking of the front is obtained using the SD method.

The last problem to be considered is the evacuation of air bubbles. Since we deal with incompressible flows, air cannot shrink and air bubbles near the corners will remain if a method to evacuate them is not devised. In practice, moulds are made of porous materials, usually sand in casting applications. Therefore, air can leave the mould without resistance. Numerically, a possible way to evacuate air is to introduce holes on the boundary and to block them when the fluid touches the wall. This method will also be explained in the following.

6.3.2 Smoothing of the pseudo-concentration surface

Even if the initial condition $\psi_0(\mathbf{x})$ is a smooth function, if the pseudo-concentration is maintained unmodified over several time steps it may begin to lose its smoothness and numerical problems may be encountered. Since the only important factor is the location of the critical contour that defines the front, it is possible to smooth ψ while maintaining the position of this critical contour. Following Thompson [Th1], this can be performed redefining the pseudo-concentration for each node of the finite element mesh according to the following expression:

$$\psi = \psi_c + \text{sgn}(\psi_o - \psi_c) \sigma d \quad (6.9)$$

where ψ_o stands for the calculated value of ψ , σ is a given constant, d is the distance from the node under consideration to the front and $\text{sgn}(\cdot)$ is the signum of the value enclosed in the brackets.

Equation (6.9) indicates that the smoothed pseudo-concentration is obtained adding or subtracting to the critical value a quantity proportional to the distance to the front, according to which material occupies the point (the fluid analysed or air). The constant σ is the slope of the new pseudo-concentration surface in the direction normal to the front.

The crucial point is how to calculate the distance d from a point under consideration to the front. We have tested several possibilities that will be briefly described now.

Nodal-based distance

To calculate the distance d , we may first identify the points surrounding the front. This can be easily done by checking if their pseudo-concentration value is close to ψ_c , i.e., $|\psi - \psi_c|$ is less than a given tolerance that depends on the diameter of the elements and the constant σ . Once these points surrounding the front are identified, the required distance from a point of interest to the front is evaluated as the minimum of the distances to these points. We have found that this method yields a somewhat undulated (and therefore inappropriate) representation of the front, especially for coarse meshes. Moreover, the tolerance to be used depends strongly on the element dimensions and the slope of the pseudo-concentration surface.

Integration-points-based distance

Instead of using the nodal points surrounding the front to calculate d for a given point, we may also employ the integration points. Apart from this, the idea is the same as before and the problems encountered are also the same, perhaps to a lesser extent.

Interpolation of a straight line

Once we know the values of the pseudo-concentration for all the nodal points, it is possible to calculate the position of the points of the front located at the sides of the elements. This can be done by checking if the sign of $\psi - \psi_c$ changes when passing from a certain node of an element to the adjacent one. When this happens, the position where the value ψ_c is attained can be computed using a linear interpolation between the values of ψ at the two nodes identified and the coordinates of these two nodes. In the most common case in which only one front crosses the element, two front points which are part of the element sides will be found. Between these two points, the position of a specified number of additional front points may be calculated by interpolating the front within each element by a straight line. If more than a single front crosses the element, an even number of front points lying on the element sides will be found. The way to connect pairs of them is easily established by moving along the boundary and checking the sign of $\psi - \psi_c$.

When the process just described is finished, the front will be represented by a set of points lying on straight segments within each element. The distance d from a considered point to the front is then computed as the minimum of the distances to all these front points.

The accuracy of this method depends on the smoothness of the front (not on the pseudo-concentration), as well as on the number of front points to be interpolated within each element. Clearly, if the front presents a sharp corner within a certain element, the approximation by a straight segment will be indeed poor. Moreover, advancing in time the approximation error will sum up and the final representation of the front may be completely wrong. In these cases the smoothing of the pseudo-concentration is not recommended. We have solved some problems of this kind just using a step function for ψ and without smoothing. However, when the front is smooth, this method has proved to be quite effective. In general, we have found that four or five additional front points interpolated within each element are needed when quadratic elements are used.

For the particular case of finite elements with interior nodes, such as the Q_2/P_1 or the P_2^+/P_1 pairs, this smoothing technique has an additional problem that we have observed while running some test cases. Let us consider the situation illustrated in Figure 6.1 for the two-dimensional Q_2/P_1 element.

The nodes of the element have been denoted by $N1, N2, \dots, N9$, the front points

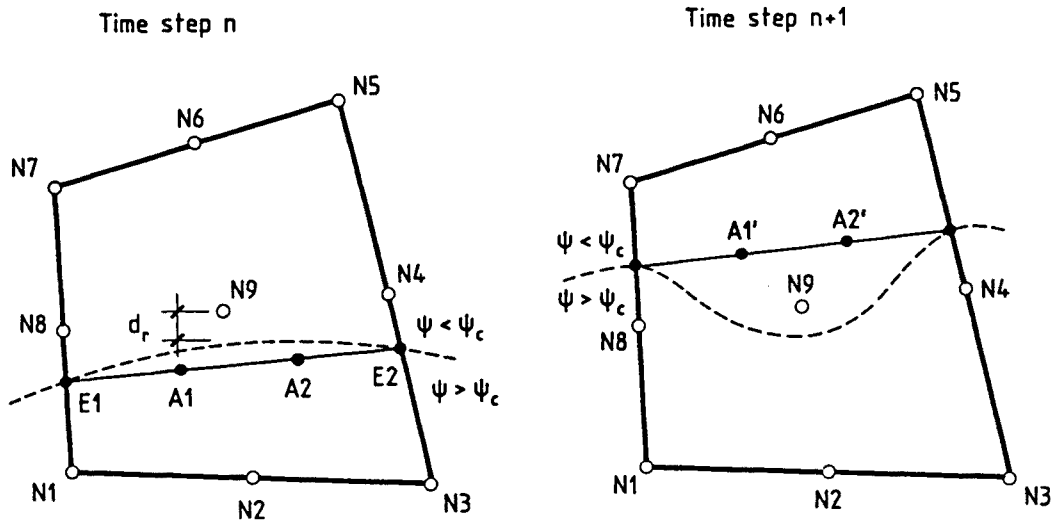


Figure 6.1 Formation of spurious air-bubbles for interior nodes. The dashed line denotes the critical contour ψ_c .

located on the sides of the element by $E1$, $E2$ and the additional front points by $A1$, $A2$. In the situation of the picture on the left of Figure 6.1, application of Eqn. (6.9) computing the distance d as explained above will lead to

$$\begin{aligned} \psi_9^n &= \psi_c - \sigma \min\{\text{dist}(N9, A1), \text{dist}(N9, A2)\} \\ &< \psi_c - \sigma d_r \end{aligned} \quad (6.10)$$

where d_r is the 'real' distance from node $N9$ to the front. Inequality (6.10) indicates that we are underestimating the value of ψ at node $N9$. The error will be much smaller for nodes $N4$ and $N8$. After solving the transport equation for the pseudo-concentration it may happen that nodes $N4$ and $N8$ are already part of the fluid whereas node $N9$ remains in the unfilled region (see the picture on the right of Figure 6.1). Since the fluid front will be again approximated by a straight segment, node $N9$ may be situated at the wrong side of this. Applying again (6.9) we will obtain

$$\psi_9^{n+1} = \psi_c - \sigma \min\{\text{dist}(N9, A1'), \text{dist}(N9, A2')\} < \psi_c$$

It is even possible that $\psi_9^{n+1} < \psi_9^n$. In any case, there exists the possibility that as the fluid front advances a spurious bubble around node $N9$ be left behind. We have observed this misbehavior in practice.

The way to circumvent this problem is quite simple. Once the values of the smoothed pseudo-concentration for the nodes lying on the element sides have been calculated using Eqn. (6.9), the value for the interior node is computed from interpolation. The serendipid interpolation (Q_2^-) is used for the biquadratic element (Q_2) and the quadratic simplicial interpolation (P_2^+) for the enriched simplex P_2^+ .

6.3.3 Air release—Introduction of holes

It has already been said that in practical problems air can leave the mould through its porous walls. Numerical models must incorporate a facility to evacuate air in order to prevent the appearance of air bubbles, especially near the corners.

The basic idea of the method to be described now is to place some holes in the walls of the mould and to block them when the fluid reaches these walls. Thus, air will be allowed to leave the mould but the fluid analysed will not.

To motivate the basic inconvenience of this method, let us describe how boundary conditions are implemented in the computer code developed in this work. If a boundary node has a Neumann type prescription, its velocity is one of the unknowns of the problem. But if a Dirichlet condition is prescribed there, the velocity vector is known. The columns and rows corresponding to the node under consideration of the assembled matrix of the final algebraic system are not needed. The product of the columns by the velocity components of the node are moved to the right-hand-side. The matrix of the resulting reduced system, say A , will be smaller than if these columns and rows are not eliminated. Since we work with dynamic memory allocation, the dimension of matrix A has to be known before starting the analysis, after reading the data of the problem. Hence, the change of a node from a Neumann boundary condition to a Dirichlet boundary condition during the analysis is not so simple as it might seem at first glance.

In order to avoid the need for changing the size of the problem, we leave the nodes located at the holes always free. When the fluid reaches them, the velocity (or perhaps only the component normal to the wall) is prescribed to zero not exactly, but through penalization.

To describe this method, let us consider a generic linear system of the form

$$Ax = b \quad (6.11)$$

where x is a vector of n unknowns. Suppose that the i -th component of x is to be prescribed to a value \bar{x} , i.e., $x_i = \bar{x}$. From Eqn. (6.11) we will have that

$$a_{ii}x_i = b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j \quad (6.12)$$

Assume that the component a_{ii} of matrix A is not zero and replace

$$\begin{aligned} a_{ii} &\leftarrow a_{ii}(1 + \lambda) \\ b_i &\leftarrow \bar{x}a_{ii}(1 + \lambda) \end{aligned} \quad (6.13)$$

From Eqn. (6.12) we will have that

$$x_i = \bar{x} - \frac{1}{a_{ii}(1 + \lambda)} \sum_{j=1, j \neq i}^n a_{ij}x_j \quad (6.14)$$

from where it follows that $x_i \rightarrow \bar{x}$ as $\lambda \rightarrow \infty$.

In practice, we have observed that values of λ of order 10^6 yield a good enough approximation to the prescription to be imposed (observe that λ is dimensionless).

The way to block the holes is now clear. For a certain time step, the value of the pseudo-concentration at the point of interest is computed. If this value ψ is lower than ψ_c , $\lambda = 0$ is taken for the system analogous to (6.11) arising from the fully discrete and linearized Navier-Stokes equations and the redefinition (6.13) is not performed. Otherwise, a high value of λ is selected, taking $\bar{x} = 0$ in (6.13).

Consider now the transport equation for the pseudo-concentration. If for a certain time step the velocity at a node lying on the hole is left free, it may point into the mould due to suction effects. In this situation, the hole must be considered as a part of the inflow boundary Γ_{inf} and therefore the function ψ must be prescribed there. Otherwise, it may happen that values of ψ higher than ψ_c be transported into the mould, thus introducing spurious fluid. The situation is similar to what happens for the one-dimensional hyperbolic equation

$$\partial_t \psi + u \partial_x \psi = 0, \quad 0 < x < 1$$

If $u > 0$ and the value of ψ at $x = 0$ is not prescribed, the solution is simply $\psi(x, t) = \psi_0(x - ut)$, where $\psi_0(x)$ is the initial condition extended by periodicity to the whole real line \mathbb{R} .

There is another way to see that if ψ is not prescribed at the nodes for which the velocity points into the mould then spurious material will be introduced. Let V_i be any control volume surrounding this node. Multiplying Eqn. (6.1) by ψ , integrating over V_i and using the fact that \mathbf{u} is divergence-free yields

$$\frac{d}{dt} \int_{V_i} \psi^2 d\Omega = -\frac{1}{2} \int_{\partial V_i} (\mathbf{n} \cdot \mathbf{u}) \psi^2 d\Gamma$$

If ψ is not prescribed where $\mathbf{n} \cdot \mathbf{u} < 0$, the integral of ψ^2 over V_i may increase as time goes on, and this happens for any control volume V_i , that is, a spurious fluid-filled region may appear around the hole.

Having these considerations in mind, it is clear that the pseudo-concentration must be prescribed at the temporary free wall nodes where $\mathbf{n} \cdot \mathbf{u} < 0$. For a certain time step, the value of the prescription will be the value obtained in the previous one. The way to implement this is the same as for the velocities in the Navier-Stokes equations. Let ψ_b^{n-1} the value of the pseudo-concentration at the node under consideration for time step $n - 1$. Considering that the system to be solved to find ψ for time step n is (6.11), the redefinition (6.13) will be employed, with $\bar{x} = \psi_b^{n-1}$. Again, we have found that good results are obtained taking λ of order 10^6 .

The checks to be performed for temporary free boundary nodes are summarized in Box 6.1. It is understood that all the variables (pseudo-concentration and velocity) refer to a certain node and that Dirichlet boundary conditions are prescribed according to the penalty technique described here.

Box 6.1 Checks for temporary free wall nodes

- IF $\psi_b^{n-1} < \psi_c$ then
 - IF $\mathbf{n} \cdot \mathbf{u} < 0$ then
 - Prescribe ψ_b^n to ψ_b^{n-1}
 - ELSE
 - Leave ψ_b^n free
 - END
 - Leave \mathbf{u} free (Neumann type condition)
 - ELSE
 - Prescribe $\mathbf{u} = \mathbf{0}$
 - Leave ψ_b free
 - END

6.4 The Navier-Stokes equations with a moving free surface

6.4.1 Statement of the problem

In Section 5.4 we have considered the general problem for an incompressible fluid in laminar regime and taking into account thermal effects. Now we will include the existence of a free surface within the domain Ω , which will be tracked using the techniques described in this chapter.

The mechanical and thermal equations describing the problem are (5.73), viz.

$$\begin{aligned}
 \rho[\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u}] - 2\nabla \cdot [\mu \boldsymbol{\varepsilon}(\mathbf{u})] + \nabla p &= \rho \mathbf{f} & \text{in } \Omega, t \in (0, T) \\
 \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega, t \in (0, T) \\
 \rho c_p [\partial_t \vartheta + (\mathbf{u} \cdot \nabla) \vartheta] - \nabla \cdot (k \nabla \vartheta) &= Q & \text{in } \Omega, t \in (0, T)
 \end{aligned} \tag{6.15}$$

It has been already explained in Section 5.4 the interest of considering the physical properties and the forcing terms as variable. The boundary conditions for the velocity and the temperature to be considered here are

$$\begin{aligned}
 \mathbf{u} &= \bar{\mathbf{u}} & \text{on } \Gamma_{du}, t \in (0, T) \\
 \mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} & \text{on } \Gamma_{nu}, t \in (0, T) \\
 \mathbf{u} \cdot \mathbf{n} = \bar{u}_n, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_1 = \bar{t}_1, \quad \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \mathbf{g}_2 = \bar{t}_2, & & \text{on } \Gamma_{mu}, t \in (0, T) \\
 \vartheta &= \bar{\vartheta} & \text{on } \Gamma_{dt} \\
 -k \mathbf{n} \cdot \nabla \vartheta &= \bar{\varphi} & \text{on } \Gamma_{nt}
 \end{aligned} \tag{6.16}$$

where the notation is the same as in Chapters 4 and 5. In particular, the boundary $\partial\Omega$ has been considered splitted into three sets of disjoint components Γ_{du} , Γ_{nu} and Γ_{mu} , the latter being the part of the boundary where mixed conditions are prescribed: the

normal velocity and the tangent stresses. Vectors \mathbf{g}_1 and \mathbf{g}_2 (for the three-dimensional case) span the space tangent to Γ_{mu} .

The initial conditions for Eqns. (6.15) are

$$\begin{aligned} \mathbf{u}(\mathbf{x}, 0) &= \mathbf{u}_0(\mathbf{x}) & \text{in } \Omega \\ \vartheta(\mathbf{x}, 0) &= \vartheta_0(\mathbf{x}) & \text{in } \Omega \end{aligned} \quad (6.17)$$

The definition of the position of the fluid front will be given by the physical properties. Let π be any of these, i.e., density (ρ), viscosity (μ), specific heat (c_p) or conduction coefficient (k). We will have that

$$\pi(\mathbf{x}, t) = \begin{cases} \pi_{fluid}(\mathbf{x}, t) & \text{if } \mathbf{x} \in \Omega_t \\ \pi_{air} & \text{if } \mathbf{x} \in \Omega \setminus \Omega_t \end{cases} \quad (6.18)$$

where

$$\Omega_t := \{\mathbf{x} \in \Omega \mid \psi(\mathbf{x}, t) \geq \psi_c\} \quad (6.19)$$

and the pseudo-concentration function ψ is the solution of the following problem:

$$\begin{aligned} \partial_t \psi + (\mathbf{u} \cdot \nabla) \psi &= 0 & \text{in } \Omega, t \in (0, T) \\ \psi &= \bar{\psi} & \text{on } \Gamma_{inf}, t \in (0, T) \\ \psi(\mathbf{x}, 0) &= \psi_0(\mathbf{x}) & \text{in } \Omega \end{aligned} \quad (6.20)$$

This is the formulation of the pseudo-concentration method. In Eqn. (6.18), the property π for the fluid-filled region is allowed to depend on the unknowns of the problem, whereas it has been considered constant for the air, i.e., for the fictitious fluid. Observe that since the physical properties will be discontinuous across the fluid front, the differential equations (6.15) will not exactly describe the conservation of momentum, mass and energy, since the jump of these properties has been simply ignored.

A particular case of the mixed boundary condition on Γ_{mu} will be employed here, namely, the von Karman law for the shear stress, also used in References [DGB] and [LUC] (see Reference [FCT] for another friction law for Bingham fluids). Introducing the tangent stress vector

$$\bar{\mathbf{t}}_\tau = \bar{t}_1 \mathbf{g}_1 + \bar{t}_2 \mathbf{g}_2$$

and the tangent velocity

$$\mathbf{u}_\tau = (\mathbf{u} \cdot \mathbf{g}_1) \mathbf{g}_1 + (\mathbf{u} \cdot \mathbf{g}_2) \mathbf{g}_2$$

the expression of this law is

$$\bar{\mathbf{t}}_\tau = -\rho \frac{|\mathbf{u}_\tau|}{(u^+)^2} \mathbf{u}_\tau \quad (6.21)$$

where the dimensionless friction coefficient u^+ depends on the rugosity of the wall and the position of the point considered through the relation

$$u^+ = A \log(y^+) + B$$

A and B being physical parameters. As in Reference [DGB] we will take $A = 2.5$, $B = 5.5$ (smooth walls) and $y^+ = 100$, yielding $u^+ = 17.01$.

Friction laws of this type are normally applied to turbulent flow problems, trying to emulate the frictional effect of boundary layers. As in the above quoted references, we shall use (6.21) to obtain tangential tractions at the walls of the mould.

6.4.2 Numerical treatment

It has already been explained in detail in Chapter 5 how to solve problem (6.15)–(6.17) and in Section 6.2.2 the numerical solution of problem (6.20). It only remains to link both problems through the updating of the physical properties given by Eqn. (6.18) for the continuous problem. First, let us discuss how to compute the matrices involving any of these properties. For example, consider the matrix \mathbf{K}_d arising from the viscous terms of the Navier-Stokes equations (see Box 4.2). Neglecting the contribution of the SD term, this matrix comes from $2 \int_{\Omega} \mu \boldsymbol{\varepsilon}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\Omega$. Using numerical integration within each element of the finite element partition with N_{gp} integration points of positions in the parent domain $\boldsymbol{\xi}_k$ and weights w_k , $k = 1, \dots, N_{gp}$, we will have that

$$\begin{aligned} \int_{\Omega} \mu \boldsymbol{\varepsilon}(\mathbf{u}_h) : \boldsymbol{\varepsilon}(\mathbf{v}_h) d\Omega &= \sum_{e=1}^{N_{el}} \int_{\Omega^e} \mu^e \boldsymbol{\varepsilon}(\mathbf{u}_h^e) : \boldsymbol{\varepsilon}(\mathbf{v}_h^e) d\Omega \\ &\approx \sum_{e=1}^{N_{el}} \sum_{k=1}^{N_{gp}} w_k \mu^e(\boldsymbol{\xi}_k) \boldsymbol{\varepsilon}[\mathbf{u}_h^e(\boldsymbol{\xi}_k)] : \boldsymbol{\varepsilon}[\mathbf{v}_h^e(\boldsymbol{\xi}_k)] |J^e(\boldsymbol{\xi}_k)| \end{aligned}$$

where J is the Jacobian determinant of the isoparametric mapping. It is therefore clear that the viscosity must be computed and stored for each integration point of each element. The same holds true for the rest of physical properties. Let us denote by π any of them and by π_k^e its value at the k -th integration point of the e -th element. To determine how to calculate it we first must know the value of the pseudo-concentration at this point, $\psi^e(\boldsymbol{\xi}_k)$, which is easily calculated from the standard interpolation from the nodal values of ψ for the element. Then,

$$\pi_k^e = \begin{cases} \pi_{fluid}^e(\boldsymbol{\xi}_k) & \text{if } \psi^e(\boldsymbol{\xi}_k) \geq \psi_c \\ \pi_{air} & \text{if } \psi^e(\boldsymbol{\xi}_k) < \psi_c \end{cases} \quad (6.22)$$

The property π for the fluid analysed may depend on the velocity and the temperature. For the 'air', it may be any constant provided that the motion of the resulting fictitious fluid do not affect the motion of the 'real' fluid. There is always the possibility of using the real air properties.

The final transient and iterative algorithm is given in Box 6.2. Only the basic steps of the scheme presented in Box 5.2 for the general problem of thermally coupled flows and nonlinear materials are indicated. The basic calculations needed to track the free surface are included.

Remarks 6.2

- (1) In Box 6.2 it is assumed that the Navier-Stokes equations are solved first and then the solution of the energy equation is performed. As mentioned in Chapter 5, there is no difficulty in swapping the order of block iterations.
- (2) The pseudo-concentration may be calculated at the beginning of the time step or at the end (staggered with respect to the other problems). This last choice is considered in Box 6.2. Both options are equally valid, but one must keep in mind that if the former is chosen the front will be 'delayed' one time step with respect to the velocity, pressure and temperature, whereas if the second possibility is adopted the situation will be the inverse. It could also be possible to include the calculation of the pseudo-concentration within the block iterative loop. We have found that this leads to convergence problems, which are due to the fact that an

- integration point may belong to the fluid in a certain iteration and to the air in the next one, thus having different physical properties from iteration to iteration.
- (3) The parameter θ of the generalized trapezoidal rule for the three different transient problems to be solved (velocity-pressure, temperature and pseudo-concentration) may be different and chosen according to the accuracy in time required for each problem.
- (4) In Box 6.2, it is understood that the boundary conditions for the temporary free wall nodes are always adjusted using the penalization method described in Section 6.3.3.
- (5) In the most common situation, the fluid front will cross an element. For some of its integration points the properties of the fluid will be used and for the others the properties of the air. Clearly, the accuracy of the integration rule will be poor for these elements, although this should not affect much the global accuracy. Also, there will be a jump in the fluxes of temperature and stresses that we have not considered. Let us denote by Γ_j^e the part of the front crossing element e . Considering for example the temperature equation, this jump (arising from the application of the divergence theorem) will be

$$\int_{\Gamma_j^e} \eta [k_{fluid} \mathbf{n} \cdot (\nabla \vartheta)_{fluid} - k_{air} \mathbf{n} \cdot (\nabla \vartheta)_{air}] d\Gamma$$

where η is the test function for the temperature. For the finite element discretization, the derivatives of ϑ within each element will be continuous, i.e., $(\nabla \vartheta)_{fluid} = (\nabla \vartheta)_{air}$, and therefore this integral will not vanish if the diffusions are different. The continuity of heat fluxes for the continuous problem implies that the bracketed term must be zero. The influence of the inclusion of the jump in the finite element problem is an aspect that deserves greater attention.

- (6) The use of the friction law given by Eqn. (6.21) will introduce another nonlinearity in the problem. Even if the Newton-Raphson linearization is employed for the convective term of the Navier-Stokes equations, this term is linearized only up to first order, since its influence is not very important, as we shall show in a numerical example. Moreover, the value of the friction at time step $n - 1$ is considered to be approximately the same as for time step n when the equations are written for this time step (this is the same approximation used for the density; see Section 5.4.2). This approximation is obviously unnecessary when the backward Euler scheme is used for the Navier-Stokes equations. Let us denote by $\mathbf{F}_{fric} = \mathbf{F}_{fric}(\mathbf{u})$ the contribution to the discrete forcing vector for Eqn. (5.78) arising from the friction law (6.21) (omitting the dependence on the SD contributions). The approximations just mentioned can be expressed as

$$\theta \mathbf{F}_{fric}^n(\mathbf{u}) + (1 - \theta) \mathbf{F}_{fric}^{n-1}(\mathbf{u}) \approx \mathbf{F}_{fric}(\mathbf{u}^{n,i-1})$$

where n and i are the actual time step and iteration, respectively.

- (7) The iterative penalty method has proved to be a fundamental ingredient for the success of the pseudo-concentration technique. In practical situations, especially for highly viscous flows, the viscosity of the fictitious material will be several orders of magnitude smaller than that of the fluid analysed. Even if the exact value for the air is not used, it must be between 3 and 5 orders of magnitude smaller. Hence, choosing *a priori* a penalty parameter for the classical penalty method will yield unavoidably a poor approximation to the incompressibility constraint or to

ill-conditioning. This is aggravated by the fact that contiguous elements may have stiffness matrices of an order of magnitude completely different. We have observed from several numerical experiments that if ϵ is taken as $\epsilon = 10^{-n} \mu_{fluid}^{-1}$ (assuming the fluid viscosity to be constant) and $\mu_{air} = 10^{-3} \mu_{fluid}$, ill-conditioning is observed for values of n as small as 3 (we have used the check proposed in Reference [ZT], Chapter 15, for ill-conditioning). \square

Box 6.2 General algorithm including free-surface tracking

- Set the initial condition $\mathbf{U}^0, \mathbf{P}^0 = 0, \Theta^0$ and Ψ^0
- $n := 0$
- WHILE $n < N$ and (*non-stationary*) DO:
 - $n \leftarrow n + 1$
 - $i := 0$
 - WHILE (*not converged*) DO:
 - $i \leftarrow i + 1$
 - Solve the Navier-Stokes equations (5.78)–(5.79)
 - Update the physical properties and forcing terms
 - Solve the temperature equation (5.80)
 - Update the physical properties and forcing terms
 - Check convergence
 - END while (*not converged*)
 - Check the sign of $\mathbf{n} \cdot \mathbf{u}$ for the temporary free wall nodes and adjust the boundary conditions for ψ (see Box 6.1)
 - Solve the pseudo-concentration equation (6.7)
 - Smooth the pseudo-concentration (see Eqn.(6.9))
 - Update the physical properties according to (6.22)
 - Check whether $\psi \geq \psi_c$ or $\psi < \psi_c$ for the temporary free wall nodes and adjust the boundary conditions for \mathbf{u} (see Box 6.1)
 - Check if the steady-state has been reached
- END while $n < N$ and (*non-stationary*)
- END

6.5 Application to some practical problems

The numerical model to simulate the mould filling process presented in this chapter will be applied now to three different problems. The first has been taken from Reference [DGC] and consists in the filling of a cavity due to the gravity acceleration. In the second problem the mould is filled by imposing an inflow velocity at the entrance of the cavity. The last problem is not directly related to the mould filling simulation, but demonstrates the possibility to apply the model described here to another metal forming problem: the plane strain hot rolling of a metal slab.

As in the two previous chapters, the numerical calculations have been carried out on a CONVEX-C320 computer using double arithmetic precision.

6.5.1 Mould filling by gravity

The problem definition for this first example is sketched in Figure 6.2. The mould is filled by a fluid that enters through the left vertical channel due to the action of the gravity acceleration. The data of the problem and the physical properties are those used in Reference [DGC]. In particular, the density and viscosity have been taken as $\rho_1 = 100$, $\mu_1 = 0.2$ for the fluid analysed (in SI units) and $\rho_2 = 0.1$, $\mu_2 = 0.02$ for the air (fictitious material). Part of the top wall of the square cavity has been left free in order to allow the air evacuation. No temporary holes have been used for this example.

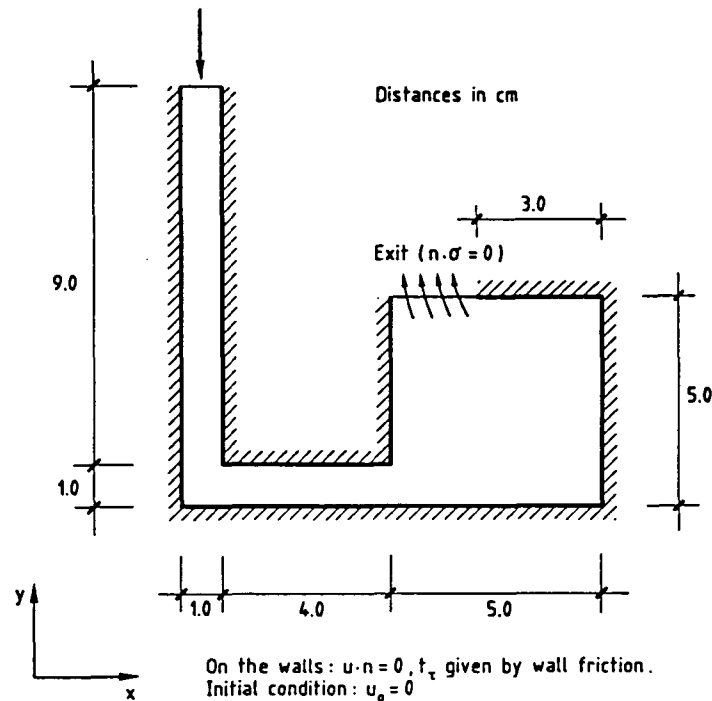


Figure 6.2 Geometry and boundary conditions for the problem of mould filling by gravity (MFG).

The boundary conditions for the Navier-Stokes equations are zero normal velocities and tangent stresses given by the wall friction law (6.21), with $u^+ = 17.01$. The initial condition is zero velocity everywhere and the fluid front located at the entrance of the left vertical channel. No thermal analysis will be performed for this example.

The computational domain has been discretized using a rather uniform mesh of 280 Q_2/P_1 elements (see Figure 6.3), yielding 1233 nodal points. The iterative penalty method has been employed with a penalty parameter $\epsilon = 10^{-4}$. The algorithmic constants of the SD formulation have been taken as $\alpha_0 = 0.5$ and $h_0 = 2$. The backward Euler scheme has been used to advance in time for the Navier-Stokes equations and the Crank-Nicolson method for the transport of the pseudo-concentration, the time step size being $\Delta t = 0.01$ in both cases.

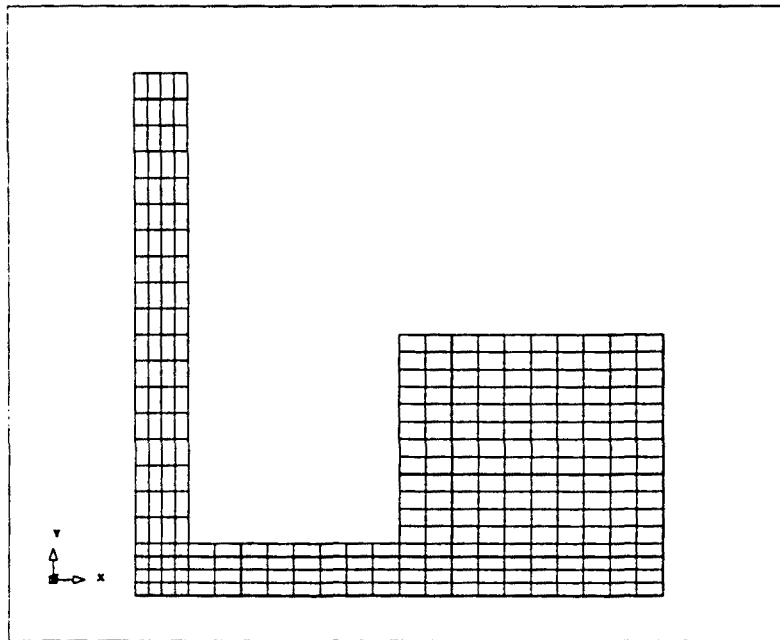


Figure 6.3 Finite element mesh for the problem of mould filling by gravity (280 Q_2/P_1 elements, 1233 nodal points) (MFG).

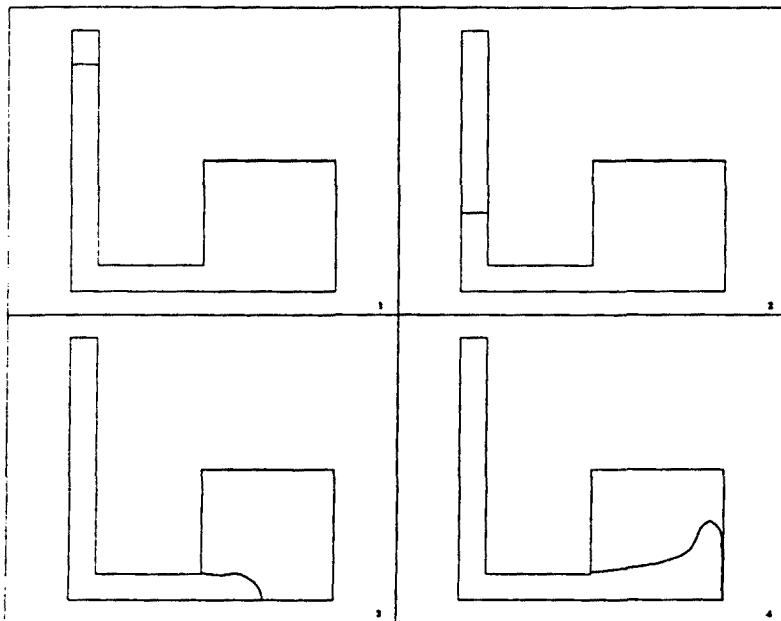


Figure 6.4 Positions of the fluid front using the Q_2/P_1 element (MFG). (1): $t = 0.1$; (2): $t = 0.2$; (3): $t = 0.3$; (4): $t = 0.4$.

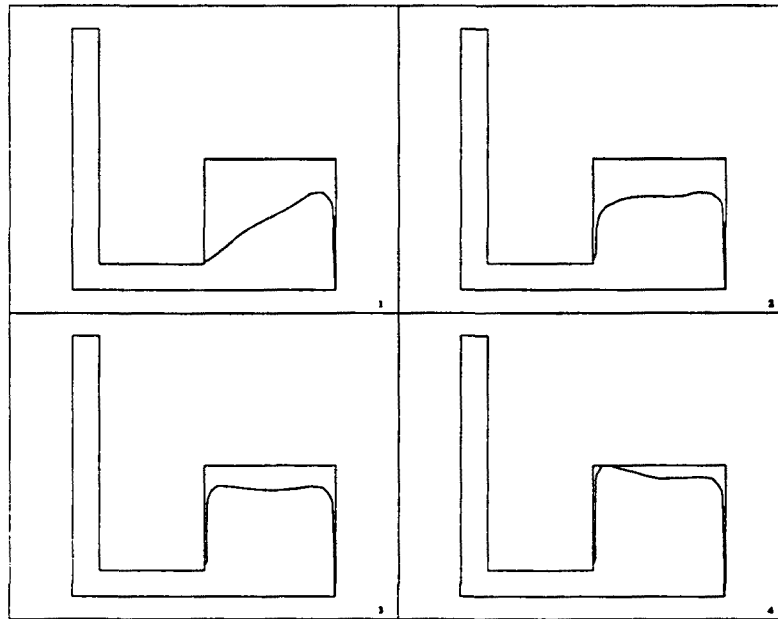


Figure 6.5 Positions of the fluid front using the Q_2/P_1 element (MFG). (1): $t = 0.5$; (2): $t = 0.6$; (3): $t = 0.7$; (4): $t = 0.8$.

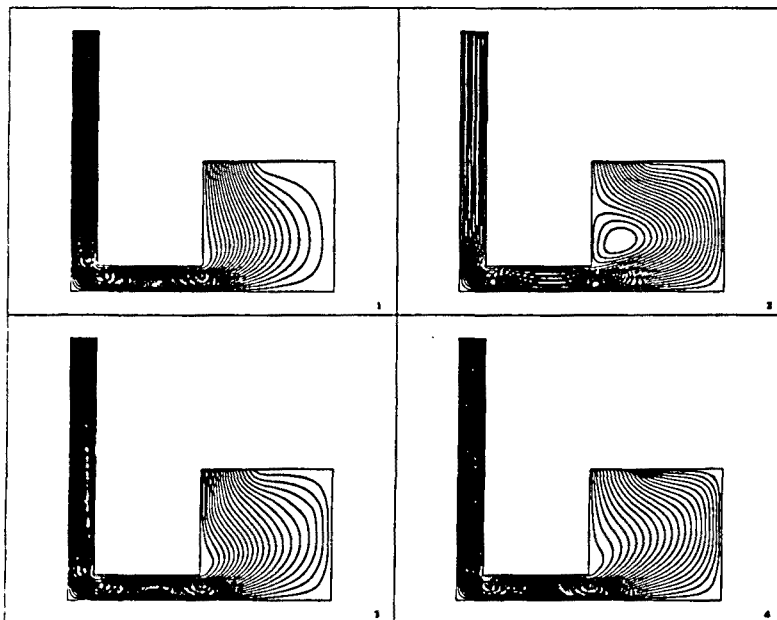


Figure 6.6 Evolution of the streamlines using the Q_2/P_1 element (MFG). (1): $t = 0.2$; (2): $t = 0.4$; (3): $t = 0.6$; (4): $t = 0.8$.

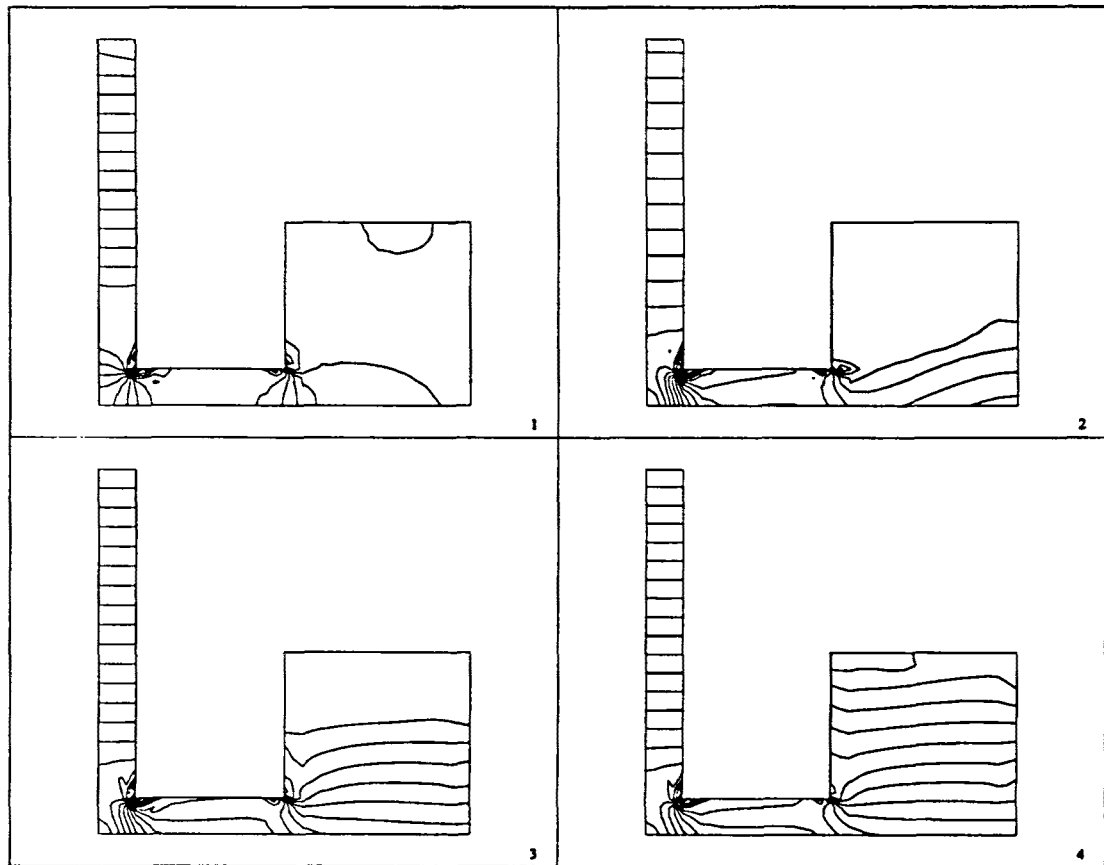


Figure 6.7 Evolution of the pressure contours using the Q_2/P_1 element (MFG). (1): $t = 0.2$; (2): $t = 0.4$; (3): $t = 0.6$; (4): $t = 0.8$.

Within each time step the convergence tolerance has been taken as 0.1%, first solving the transport of the pseudo-concentration and then iterating (using the Picard method) until a converged solution of the Navier-Stokes equation is found. Two iterations have been needed per time step. The final value of the norm of the incompressibility constraint has been found to be of order 10^{-11} for all the time steps.

Once the pseudo-concentration is calculated, the smoothing technique described in Section 6.3.2 has been employed, with a slope $\sigma = 1$ (see Eqn. (6.9)) and using five points within each element to compute the distance d .

Numerical results are shown in Figures 6.4 to 6.8. The evolution of the fluid front is depicted in the first two of them. In general, our results agree very well with those presented in Reference [DGC], although they are delayed since the initial position of the front is different and we have solved first the transport of the pseudo-concentration (cf. Remark 6.2.(2)). Looking at Figures 6.4.(1) and 6.4.(2) it is observed that the influence of the friction is very little at the walls of the vertical channel, since the shape of the front is almost straight there. To give a qualitative explanation to this fact, let us consider a single particle in contact with the walls and neglecting the effect of the contiguous particles, i.e., just considering the gravity acceleration. Denoting by

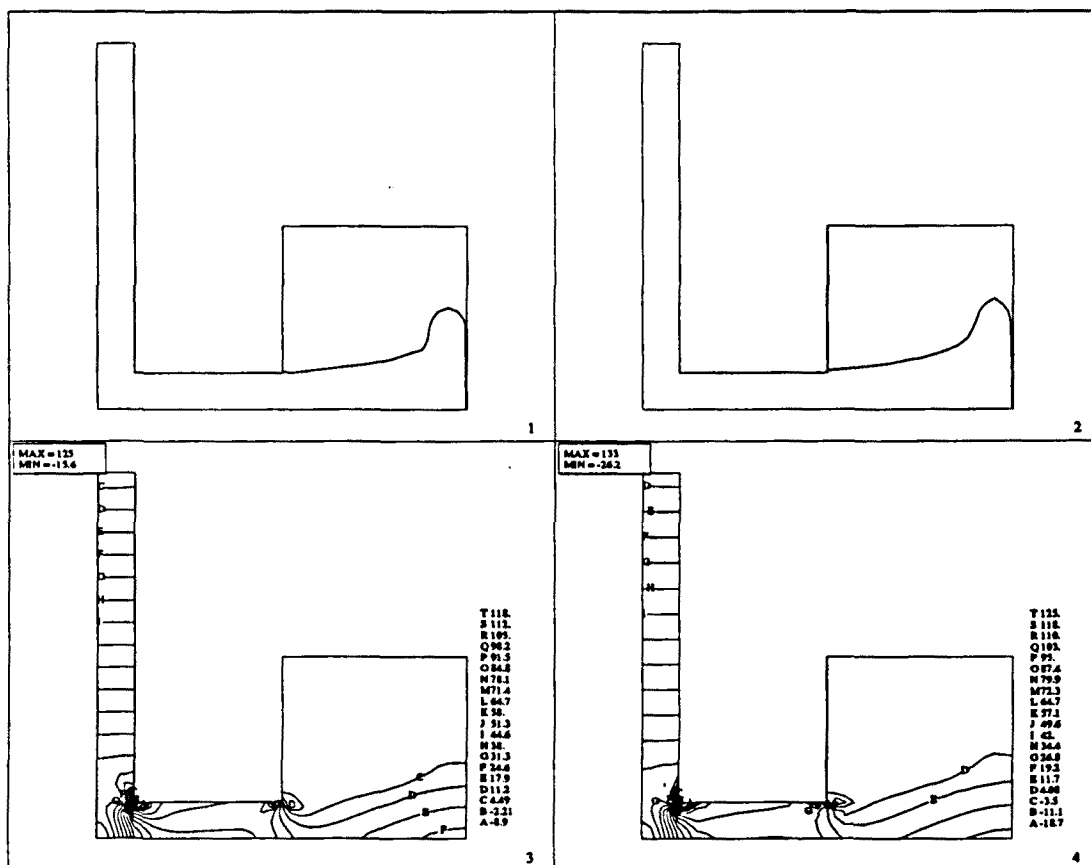


Figure 6.8 Comparison of the results obtained using the Q_2/P_1 and the Q_1/P_0 elements at time $t = 0.4$ (MFG). (1): Position of the front using the Q_1/P_0 element; (2): Position of the front using the Q_2/P_1 element; (3): Pressure contours using the Q_1/P_0 element; (4): Pressure contours using the Q_2/P_1 element.

$x = x(t)$ its vertical position measured from the entrance of the channel, this function will be the solution of the following non-linear equation

$$\ddot{x} = -\alpha(\dot{x})^2 + g \quad (6.23)$$

where $\alpha = (u^+)^{-2} \approx 3.46 \times 10^{-3}$ and g is the gravity acceleration. Assuming initial conditions $x(0) = 0$ and $\dot{x}(0) = 0$, the solution of Eqn. (6.23) is

$$\dot{x}(t) = \sqrt{\frac{g}{\alpha}} \tanh(\sqrt{g\alpha} t) \quad (6.24)$$

$$x(t) = \frac{1}{\alpha} \log(\cosh(\sqrt{g\alpha} t)) \quad (6.25)$$

Expanding the velocity given by (6.24) in Taylor series in the neighborhood of $t = 0$ (or $\alpha = 0$) it is found that

$$\dot{x}(t) = g \left[t - \frac{1}{3} \alpha g t^3 + O(\alpha^2 g^2 t^5) \right]$$

from where it follows that for α small or t small the influence of the friction is negligible with respect to the gravity effect. Friction will only be important once the vertical channel is completely filled. We have solved numerically the falling of the fluid in a very long channel (2 meters) and the position of the front agrees extremely well with the results predicted by Eqn. (6.25).

Let us return now to the discussion of the physical results. The evolution of the streamlines is shown in Figure 6.6. From the second box it is observed that a vortex is induced in the air due to the transmission of shear stresses from the fluid to the air. When the cavity is filled, this vortex disappears.

Pressure contours at different times are plotted in Figure 6.7. These contours allow to check the influence of the air on the motion of the fluid that fills the mould. Pressure gradients should be rapidly dissipated in the region occupied by the fictitious material and this in fact is observed to happen (see the positions of the front in Figures 6.4 and 6.5 corresponding to the plots of Figure 6.7).

A comparison between the behavior of the Q_2/P_1 and the Q_1/P_0 elements is shown in Figure 6.8. For the latter element, the mesh has been built up by splitting each element of mesh used for the former into four bilinear quadrilaterals and the SD parameters have been taken as $\alpha_0 = 1$ and $h_0 = 2$. Apart from this, the numerical strategy is the same in both cases. From the first and second boxes of Figure 6.8 it is observed that the Q_1/P_0 element shows a stiffer behavior than the Q_2/P_1 pair, for which the fluid front is smoother. Pressure contours in both cases are similar (slightly smaller absolute values are obtained using the Q_1/P_0 element). The possible stability problems that could be found using the bilinear-constant pair have not been observed for this particular example.

The cost of the numerical simulation is significantly smaller using the Q_1/P_0 than the Q_2/P_1 element. This is basically due to the formation and storage of the element matrices. The elemental stiffness matrix for the Q_2/P_1 element has $18 \times 18 = 324$ components, whereas four matrices for the Q_1/P_0 element have $4 \times 8 \times 8 = 256$ components. Also, the bandwidth of the assembled global matrix for the Q_1/P_0 pair is smaller than for the Q_2/P_1 . Using a profile storage, the maximum column height (after renumbering the equations) for the problem now considered is 96 for the former and 163 for the latter. The total memory required is 2.44 and 2.13 Mega-bytes, and the CPU time per iteration 10.8 and 7.8 seconds, respectively. Finally, let us mention that the tracking of the free surface has a very reduced computational cost compared to the solution of the Navier-Stokes equations (the 16.36% of the total CPU time for the Q_2/P_1 element).

6.5.2 Injection mould filling

In this second example the numerical simulation of the filling of the mould shown in Figure 6.9 is considered. The geometry and experimental results for this problem have been provided by RENAULT (Reference [RA]). The experiments have been carried out using Gallium, a metal well suited to experimentation because it has a low fusion point (30°C) and therefore it is easy to use it in the laboratory and to recover it once the experiments are finished. Moreover, its properties are close enough to those of the aluminium and other metals used in casting applications.

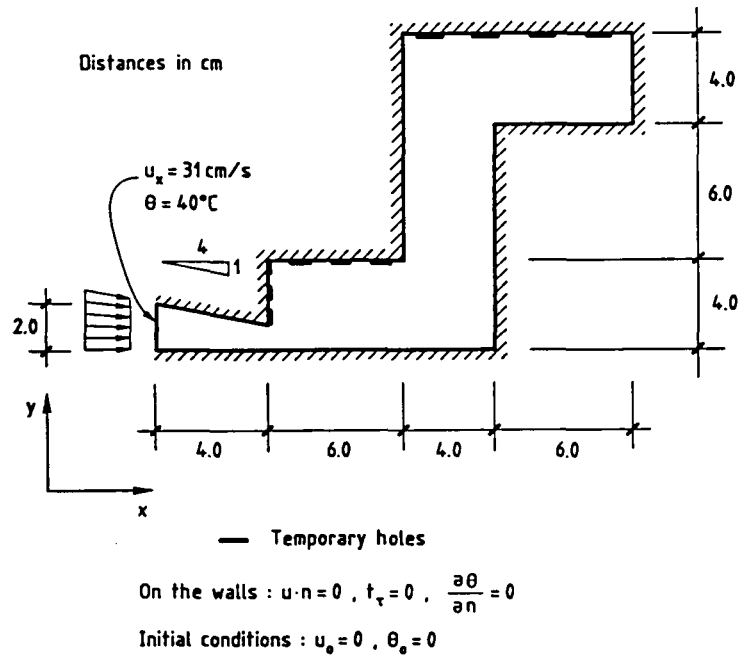


Figure 6.9 Geometry and boundary conditions for the problem of injection mould filling (IMF).

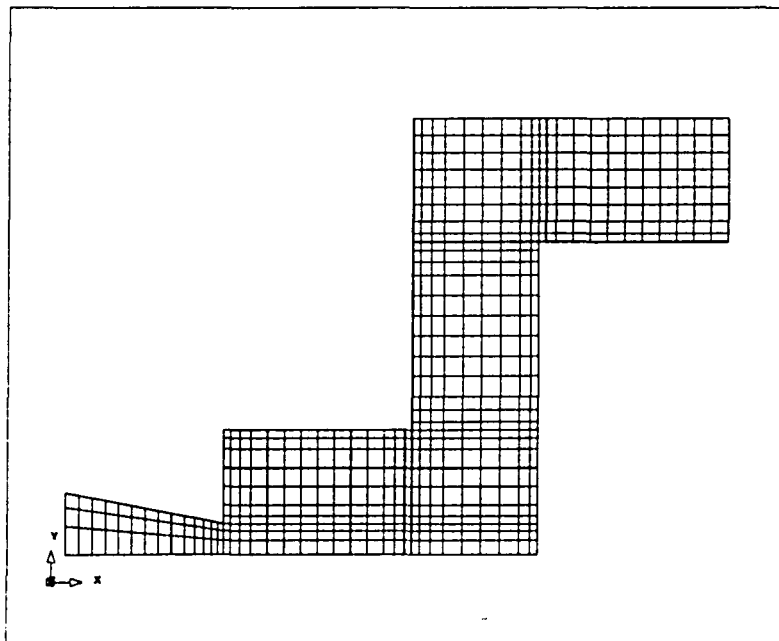


Figure 6.10 Finite element mesh for the problem of injection mould filling (548 Q_2/P_1 elements, 2351 nodal points) (IMF).

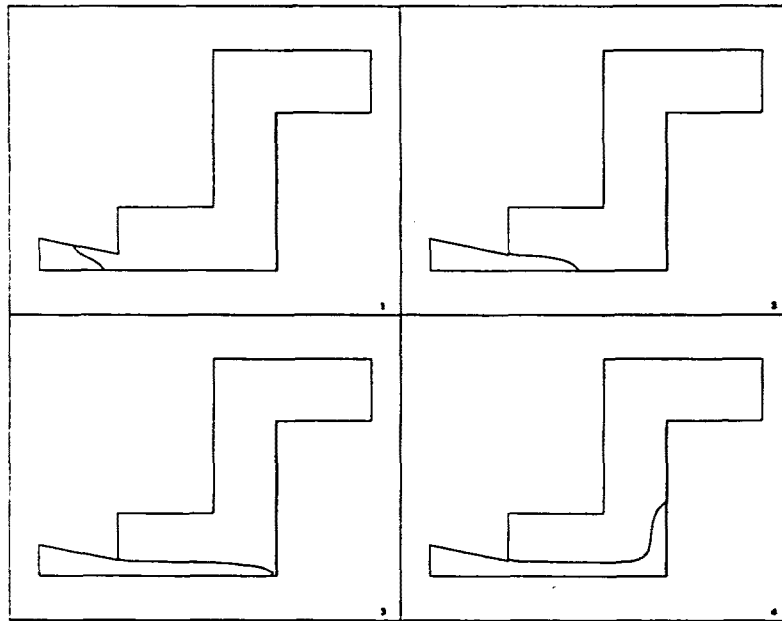


Figure 6.11 Positions of the fluid front (IMF). (1): $t = 0.1$; (2): $t = 0.2$; (3): $t = 0.3$; (4): $t = 0.4$.

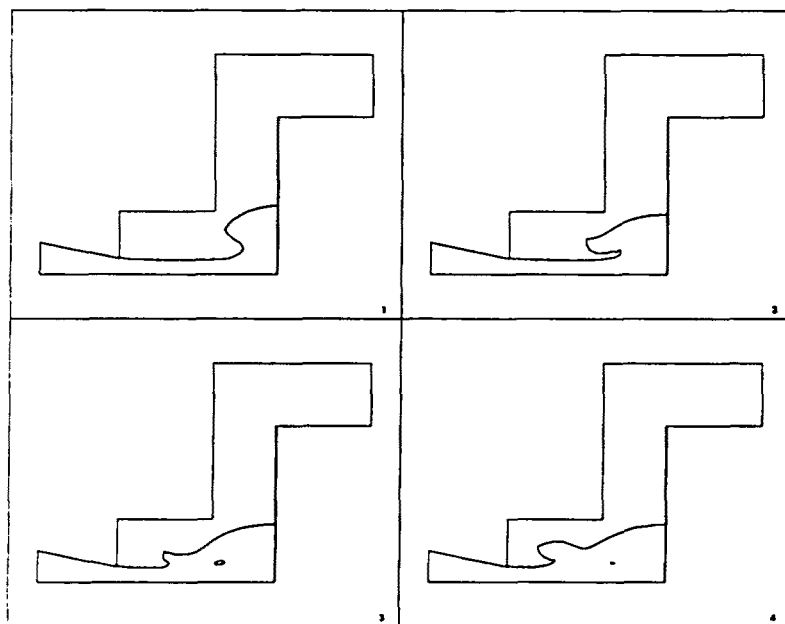


Figure 6.12 Positions of the fluid front (IMF). (1): $t = 0.50$; (2): $t = 0.55$; (3): $t = 0.60$; (4): $t = 0.65$.

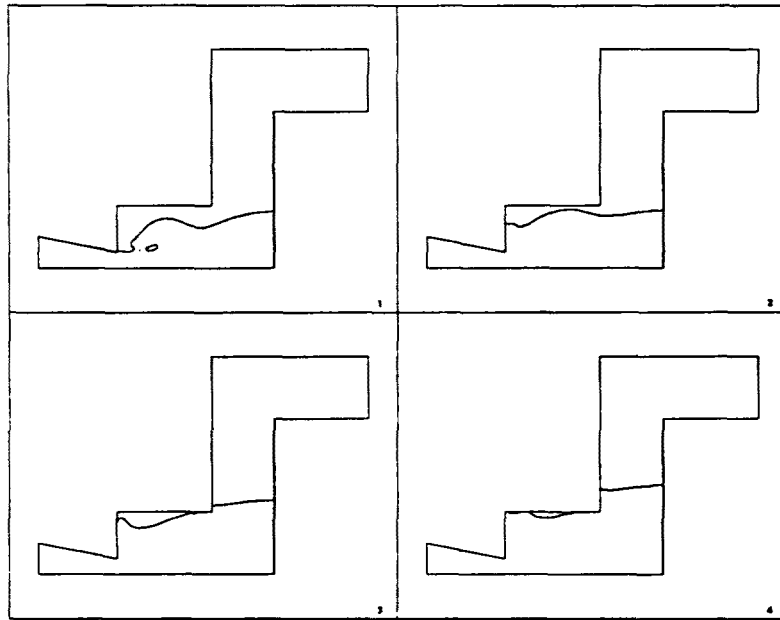


Figure 6.13 Positions of the fluid front (IMF). (1): $t = 0.7$; (2): $t = 0.8$; (3): $t = 0.9$; (4): $t = 1.0$.

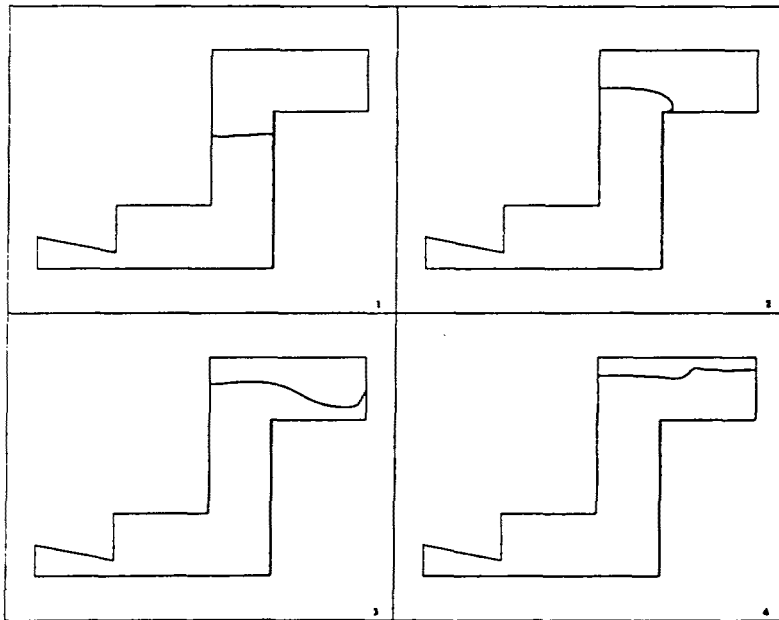


Figure 6.14 Positions of the fluid front (IMF). (1): $t = 1.2$; (2): $t = 1.4$; (3): $t = 1.6$; (4): $t = 1.8$.

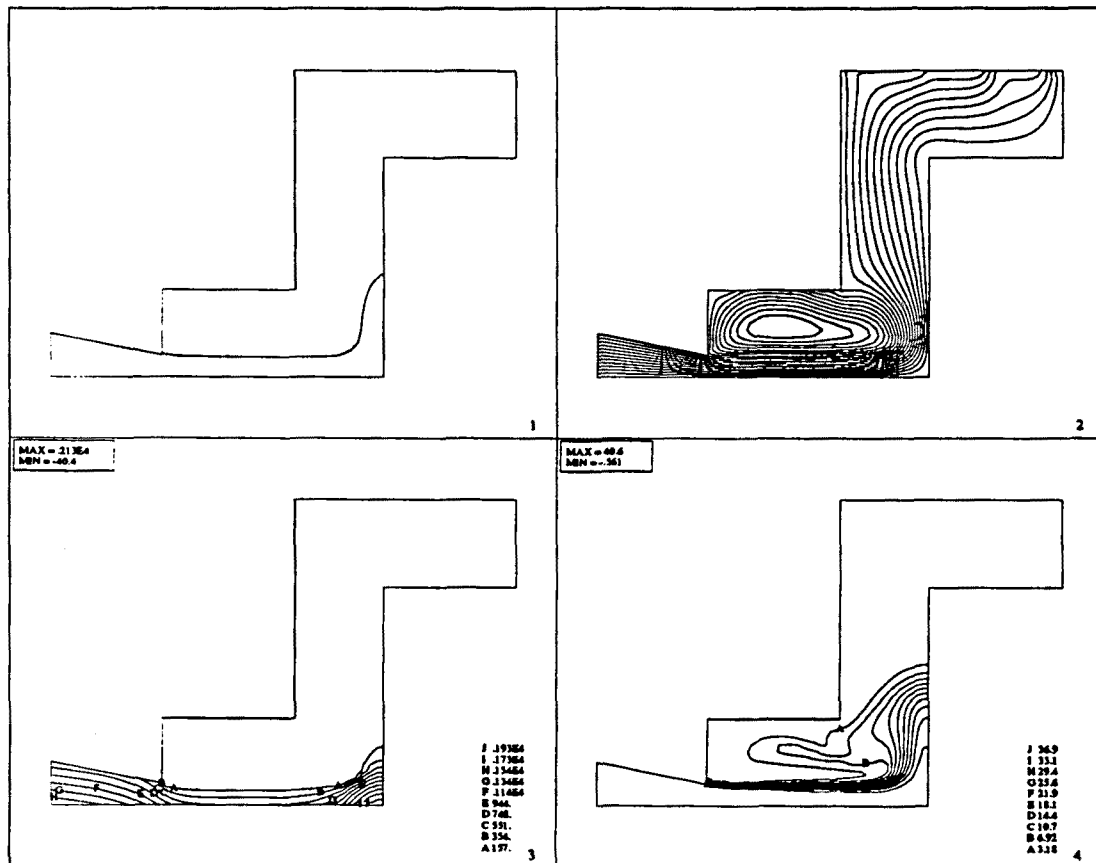


Figure 6.15 Some relevant physical results at $t = 0.4$ (IMF). (1): Position of the front; (2): Streamlines; (3): Pressure contours; (4): Temperature contours.

The molten metal enters through the left gate shown in Figure 6.9 with a horizontal velocity of 0.31 m/s. The vertical velocity is accommodated to the slope of the top wall of the entering gate. The physical properties of the molten Gallium at 55°C are (all in SI units) $\rho = 5.9 \times 10^3$ (density), $\mu = 1.9 \times 10^{-3}$ (dynamical viscosity), $k = 30.4$ (thermal conduction coefficient) and $c_p = 250$ (specific heat at constant pressure). Thus, the Reynolds number based on the velocity that enters the cavity (0.62) and its longitudinal length (0.1) is $Re = 1.93 \times 10^5$. The flow is clearly turbulent for such a high Reynolds number and it is impossible to simulate it with a laminar model as ours. The physical properties of air are $\rho = 1.2$, $\mu = 1.8 \times 10^{-5}$, $c_p = 1005$ and $k = 0.0256$. In order to reproduce the relative importance of all the physical effects, we have used the real properties of the Gallium and the air except for the dynamical viscosity, which has been taken 10^n times higher for both the Gallium and the air. Results are qualitatively similar for $n = 3$ and $n = 2$. We have failed to obtain a converged solution for lower values of this exponent.

The boundary conditions for this problem are zero normal velocities at the walls and zero tangent stresses, i.e., no friction with the walls is considered. The fluid is

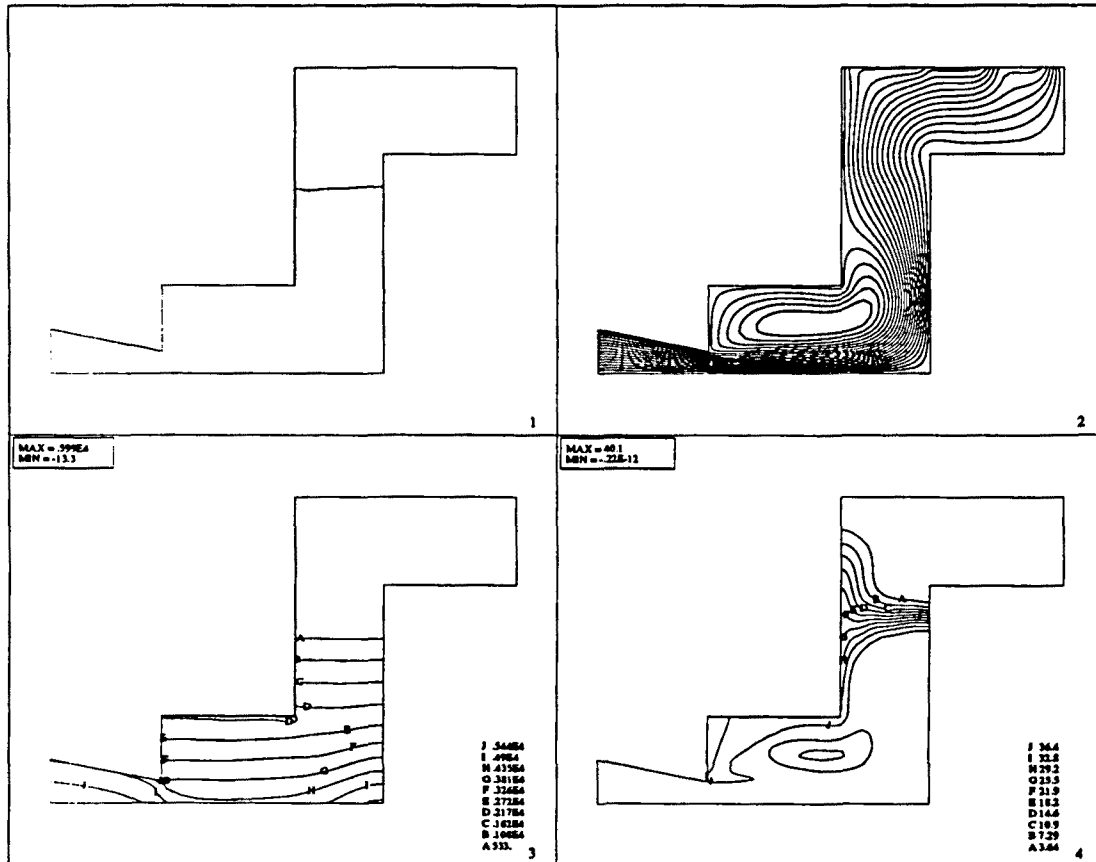


Figure 6.17 Some relevant physical results at $t = 1.2$ (IMF). (1): Position of the front; (2): Streamlines; (3): Pressure contours; (4): Temperature contours.

In order to allow the air evacuation, some holes have been introduced on the walls of the mould. They are schematically shown in Figure 6.9 (three or four nodes of the finite element discretization correspond to each hole). The parameter λ to block them when the Gallium touches the wall (see Section 6.3.3) has been taken as $\lambda = 10^6$.

Numerical results are shown in Figures 6.11 to 6.20. The position of the fluid front at different times is depicted in Figures 6.11 to 6.14. It is observed how several air bubbles appear in the Gallium. This fact is also observed in the experimental results [RA], with which the numerical simulation shows a good qualitative agreement. The differences should be attributed to the different Reynolds number of the numerical calculation. Air bubbles disappear as time goes on due to the artificial effect of the smoothing technique. Observe from Figure 6.1 that the interpolation of the front by a straight segment within each element will advance or delay artificially the front depending on its curvature. Physically, air bubbles disappear because air can escape through the porous lateral walls of the sand mould.

Figures 6.15 to 6.18 show the position of the fluid front, the streamline pattern, the pressure contours and the temperature contours at different times. From the streamline plots, it is observed how air enters or leaves the mould through the holes, as well as

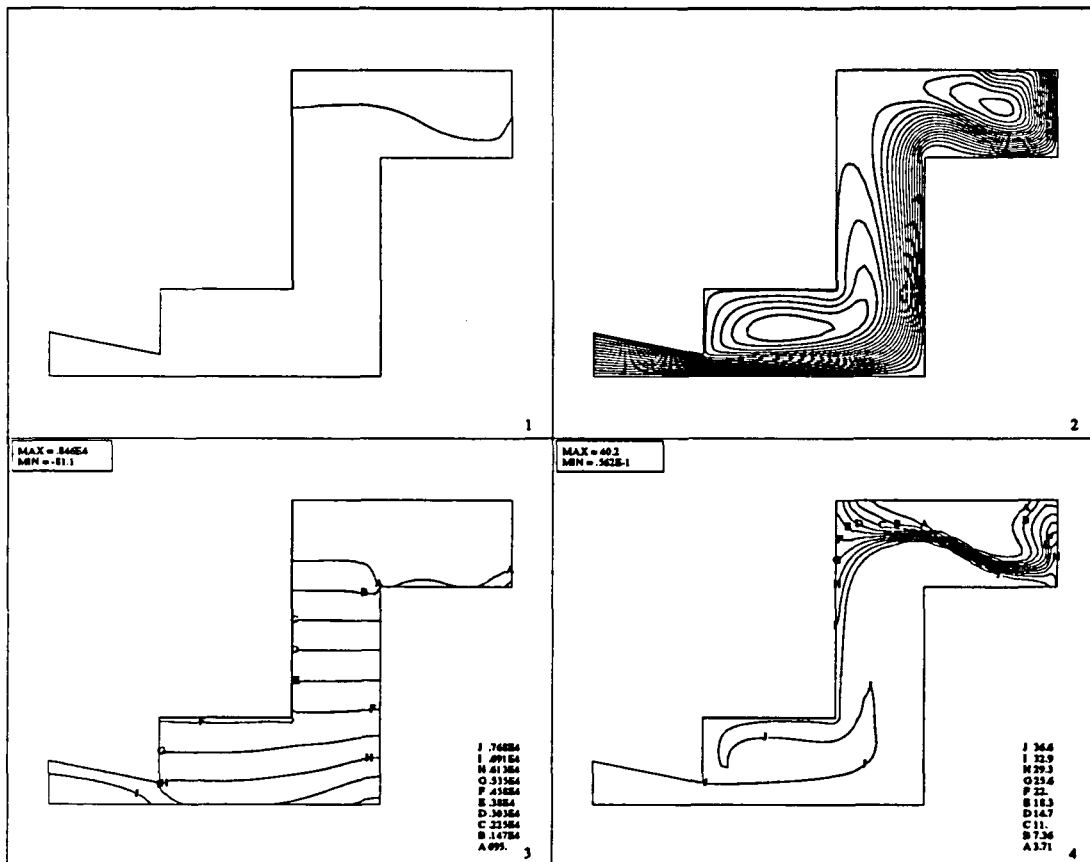
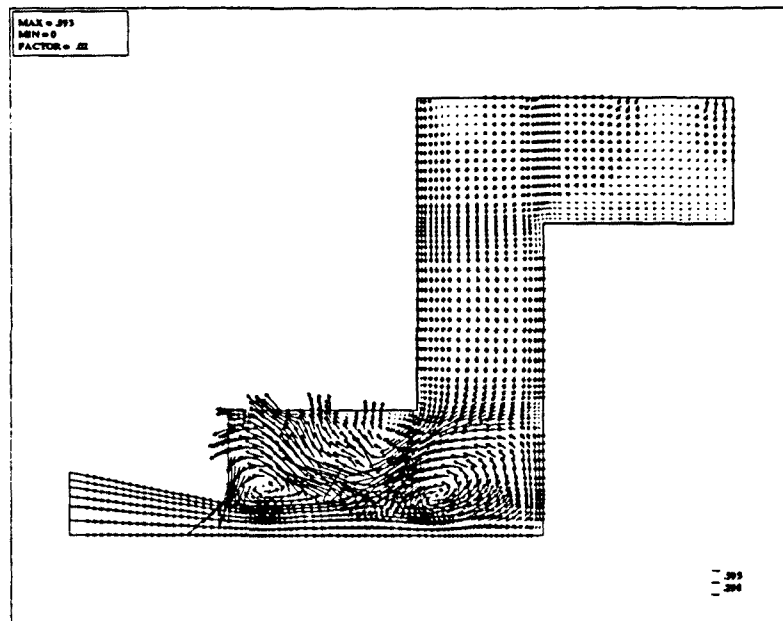
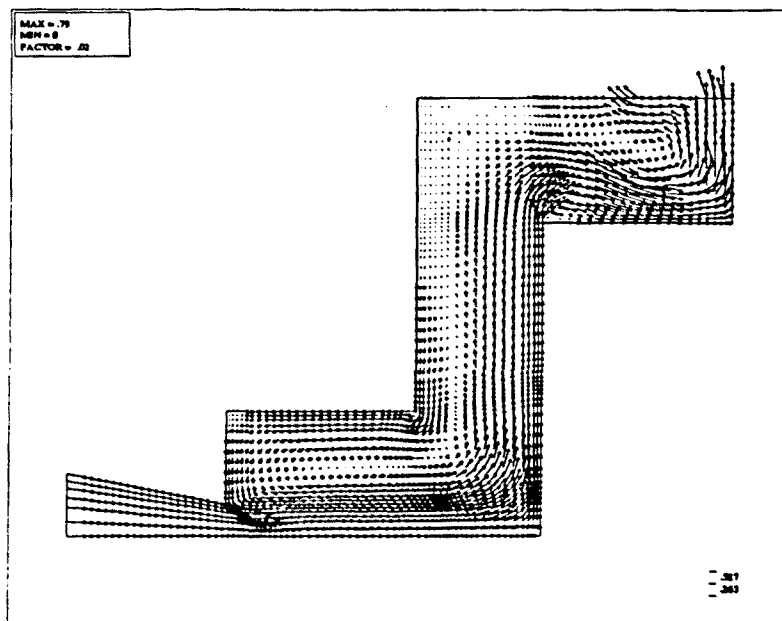


Figure 6.18 Some relevant physical results at $t = 1.6$ (IMF). (1): Position of the front; (2): Streamlines; (3): Pressure contours; (4): Temperature contours.

the creation of vortices due to the transmission of shear stresses. All these results are in accordance with what physical intuition predicts. From the pressure plots it is seen that pressure gradients are rapidly dissipated in the air region. This indicates that the motion of air does not influence much that of the Gallium. Isotherm curves show how heat is basically transported through convection. Conduction transport is only apparent in regions occupied by Gallium that has first entered the cavity. It is remarkable to note the high temperature gradients that the numerical method is able to capture at the interface between hot Gallium and cold air.

Velocity vectors at times $t = 0.6$ and $t = 1.6$ are plotted in Figures 6.19 and 6.20, respectively. From the former it is observed how air enters the mould through the holes placed at the top wall and leaves it through the holes of the bottom left corner. The pseudo-concentration is prescribed at the temporary inflow using the penalization technique described in Section 6.3.3. Otherwise, spurious fluid would enter the cavity. The effect of the blocking of the holes when Gallium contacts the walls is clearly appreciated from Figure 6.20. It is also observed that a vortex remains in the fluid-filled region.

Figure 6.19 Velocity vectors at $t = 0.6$ (IMF).Figure 6.20 Velocity vectors at $t = 1.6$ (IMF).

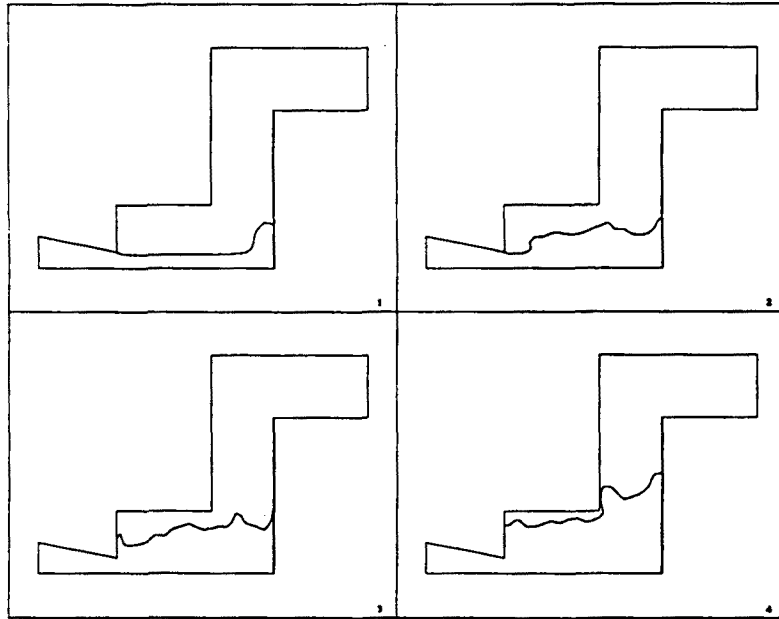


Figure 6.21 Positions of the fluid front without the introduction of holes on the walls (IMF). (1): $t = 0.5$; (2): $t = 1.0$; (3): $t = 1.5$; (4): $t = 2.0$.

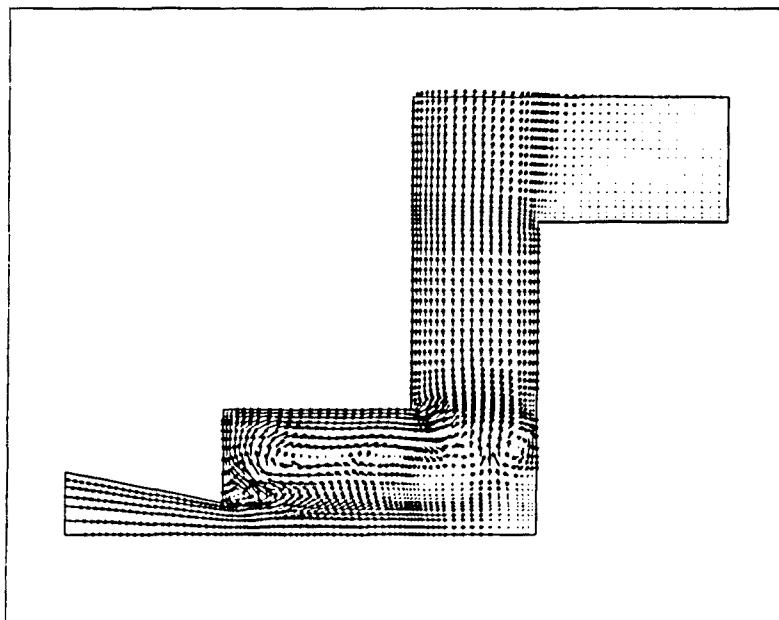


Figure 6.22 Velocity vectors at $t = 1.0$ without the introduction of holes on the walls (IMF).

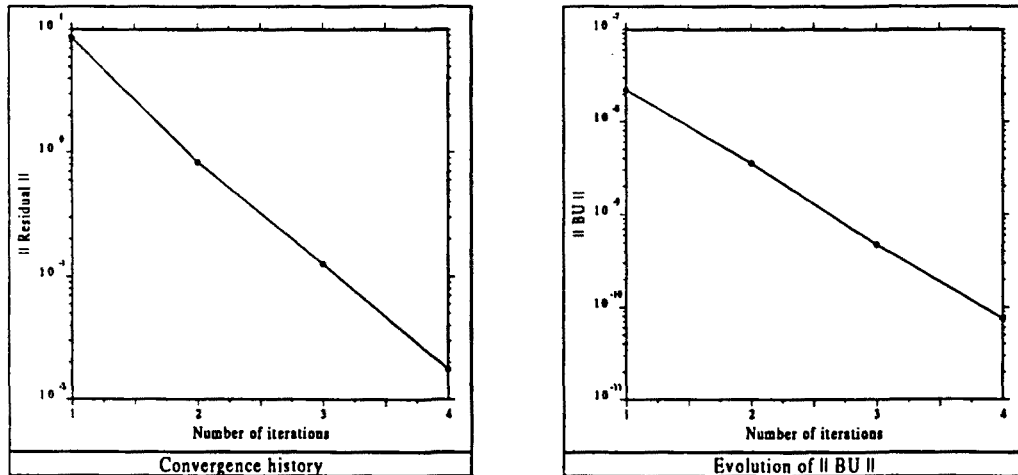


Figure 6.23 Convergence history and evolution of the norm of the incompressibility constraint for time step number 77 (IMF).

If air is not allowed to escape, the flow features are much more complicated. If only the top wall is left free, the position of the fluid front at different times is depicted in Figure 6.21. It could be argued that the ondulations of the fluid surface are due to a misbehavior of the pseudo-concentration technique. To show that this is not the case, the velocity vectors for $t = 1.0$ have been plotted in Figure 6.22. It is observed how small recirculation zones are created in the air region, thus inducing the shape of the fluid front.

Referring now to some computational aspects of the calculation, the behavior of the iterative penalty method has been found to be again very effective. The convergence history and the evolution of the norm of the incompressibility constraint for time step number 77 are shown in Figure 6.23. It is observed that this norm decreases almost three orders of magnitude in four iterations. This decrease is even more accentuated for the first time steps (not shown). Starting from a value of order 10^{-7} , a final value of order 10^{-11} is obtained in four iterations.

Most of the computational cost of the simulation is due to the solution of the Navier-Stokes equations. The CPU time per iteration has been 24.27 seconds (54.96% for the formation of the element matrices, 44.96% for the solution of the linear system). The pseudo-concentration and the temperature are solved only once per time step. The CPU time needed has been 5.03 (11.54% for the element matrices, 52.98% for the linear system, 35.48% for the smoothing and updating of physical properties) and 3.05 seconds (24.83% for the element matrices, 75.16% for the linear system), respectively.

6.5.3 Hot rolling of a rectangular slab

The plane stress hot rolling of a metal has been chosen for this last example. The problem definition is sketched in Figure 6.24. All the data except for slight changes in the geometry have been taken from Reference [ZOH]. In particular, the constitutive law (5.58) has been adopted for the metal, with $\gamma = \infty$ (pure plastic flow) and σ_y

depending on the temperature through the following empirical law:

$$\sigma_y = \frac{1}{C_1} \left[\left(\frac{Z}{C_2} \right)^{1/C_3} + \sqrt{\left(\frac{Z}{C_2} \right)^{2/C_3} + 1} \right] \quad (6.26)$$

with

$$Z := \sqrt{3}\hat{\epsilon} \exp\left(\frac{C_4}{R\vartheta}\right) \quad (6.27)$$

The parameter $\hat{\epsilon}$ is defined by Eqn. (5.57) and the experimental constants C_i , $i = 1, 2, 3, 4$, and R are

$$\begin{aligned} C_1 &= 0.01901 \quad \text{m}^2/\text{MN} \\ C_2 &= 7.92 \times 10^8 \quad \text{s}^{-1} \\ C_3 &= 5.0 \\ C_4 &= 1.39 \times 10^5 \quad \text{J/g mole} \\ R &= 8.311 \quad \text{J/g mole K} \end{aligned}$$

The use of the pseudo-concentration technique will allow us to follow the metal since it first contacts the roll until the steady-state is reached. Besides the inherent interest of this numerical simulation, a classical problem concerning free surfaces will be solved: the swelling problem (see, e.g., Reference [WC] and references therein).

The values of the physical properties we have used are the following:

	Metal	Fictitious fluid
ρ Kg/cm ³	0.00275	0.001
k cal/cm s K	0.4302	0.01
c_p cal/Kg K	239.01	1000.0
μ N s/cm ²		1.0

For the viscosity law (5.58), a cut-off value $\mu_c = 10^5$ has been chosen. The viscosity values in the metal for the converged solution are always below this limit, except where the strain rate is small, i.e., in regions where the flow approach used here is not valid. This happens before the metal contacts the roll. Since we assume that the initial position of the metal is the first contact with this roll (see Figure 6.24), this simplification is immaterial for the results.

In practice, there is no rigid contact between the metal and the roll. In order to simulate the friction between them, we have just assigned a smaller viscosity (100 times smaller) to a boundary zone defined by very narrow elements (see Figure 6.25). In Reference [ZOH], the friction was introduced by means of a law relating σ_y with the pressure and using a somehow arbitrary friction coefficient. The introduction of proper friction laws between roll and slab surfaces is an aspect that deserves further investigation.

Not all the plastic work has been considered to be transformed into heat, but only the 90%. Therefore, the source term for the energy equation given by expression (5.63) has been multiplied by 0.9.

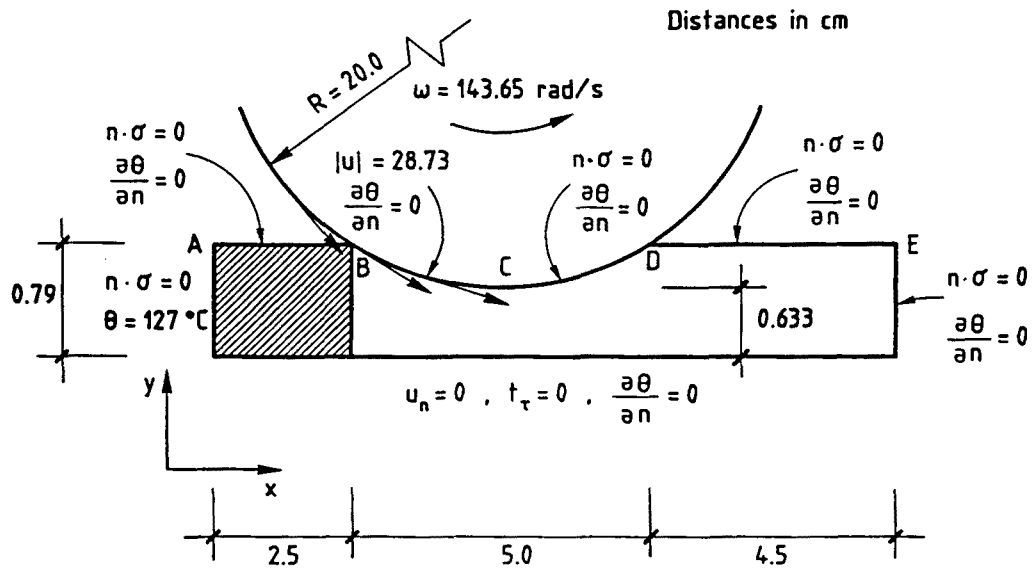


Figure 6.24 Geometry and boundary conditions for the problem of hot rolling of a metal slab (HRM).

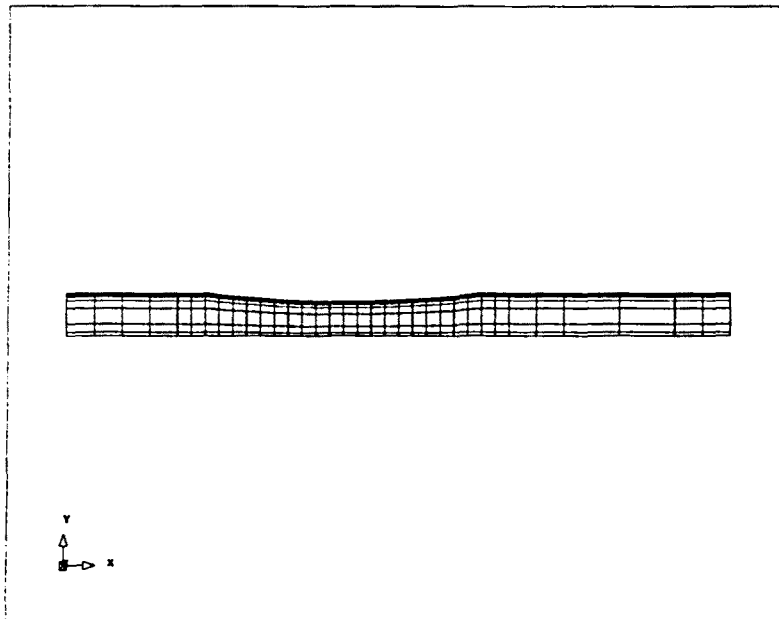


Figure 6.25 Finite element mesh for the problem of hot rolling a metal slab (HRM) (340 Q_2/P_1 elements, 1449 nodal points).

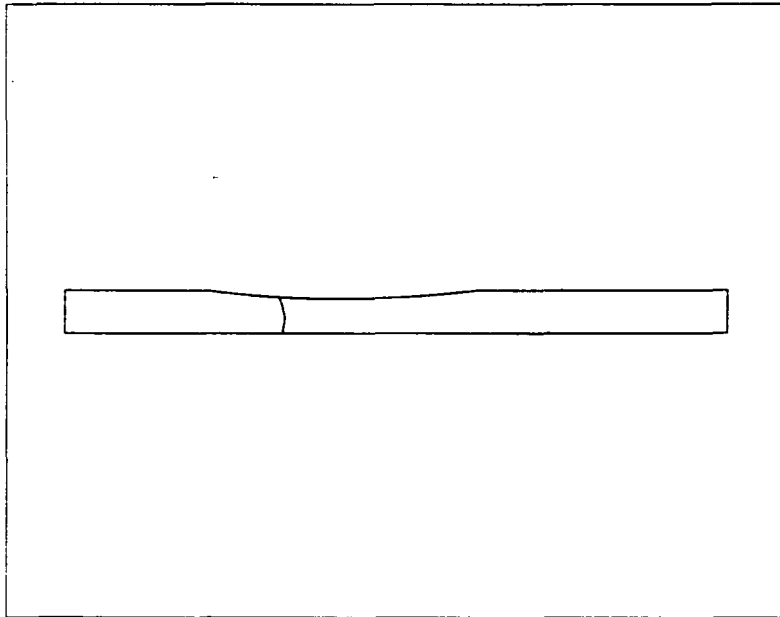


Figure 6.26.(a) Position of the metal front at $t = 0.5$ (HRM).

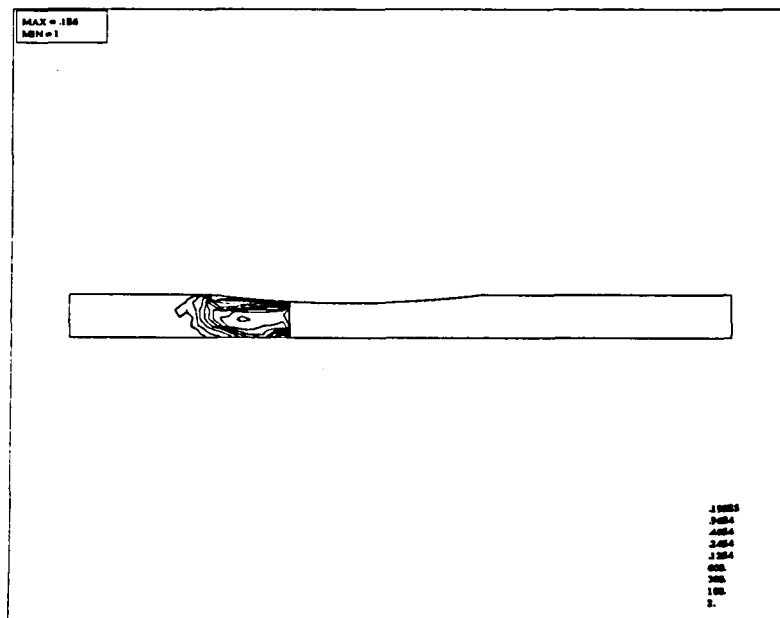


Figure 6.26.(b) Some viscosity contours at $t = 0.5$ (HRM).

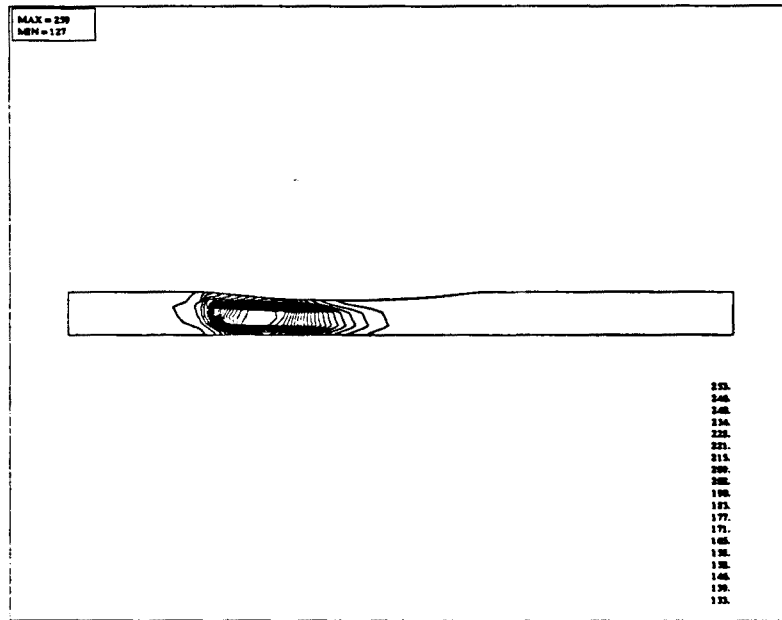


Figure 6.26.(c) Temperature contours at $t = 0.5$ (HRM).

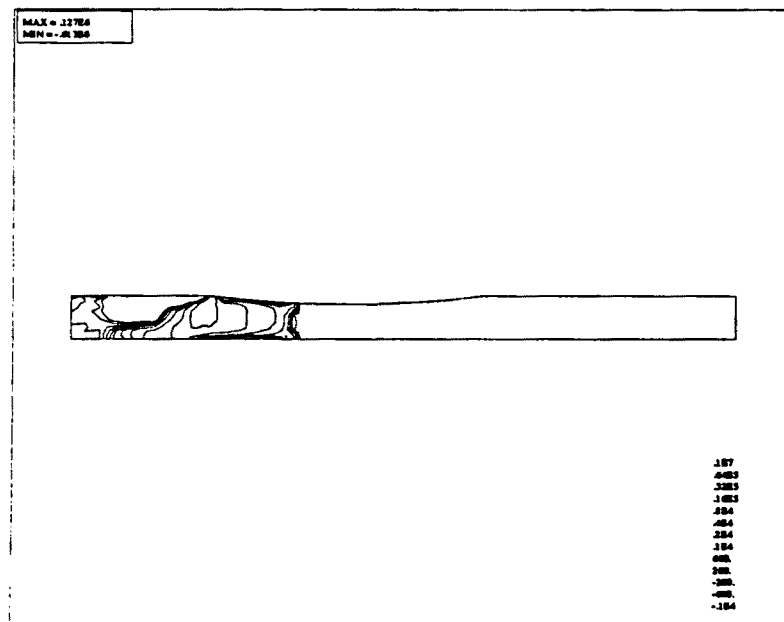


Figure 6.26.(d) Some pressure contours at $t = 0.5$ (HRM).

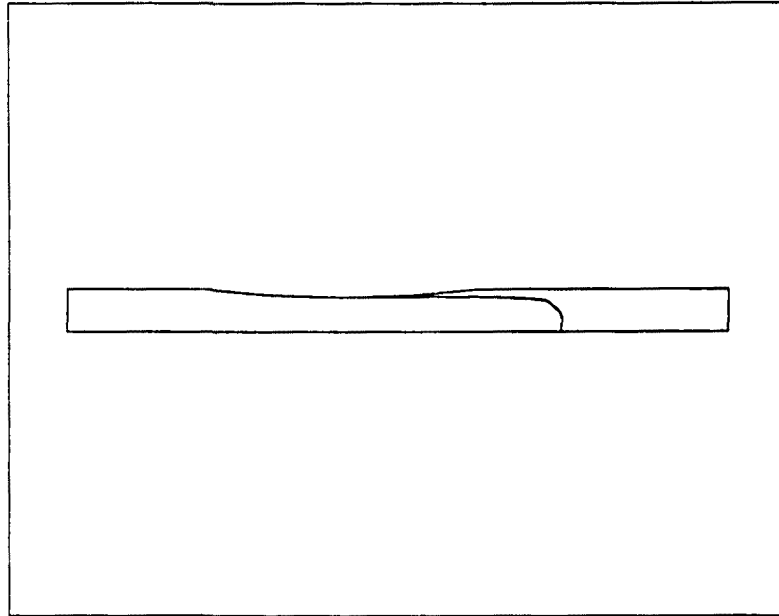


Figure 6.27.(a) Position of the metal front at $t = 2.0$ (HRM).

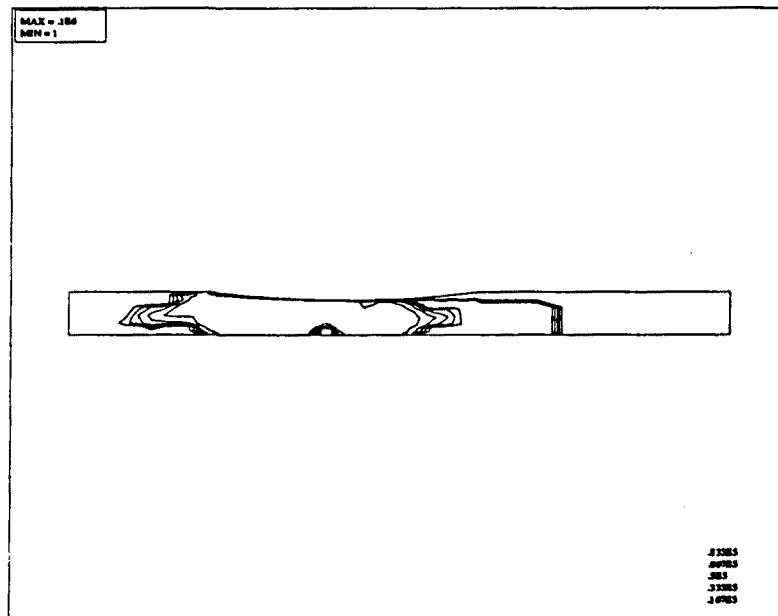


Figure 6.27.(b) Viscosity contours at $t = 2.0$ (HRM).

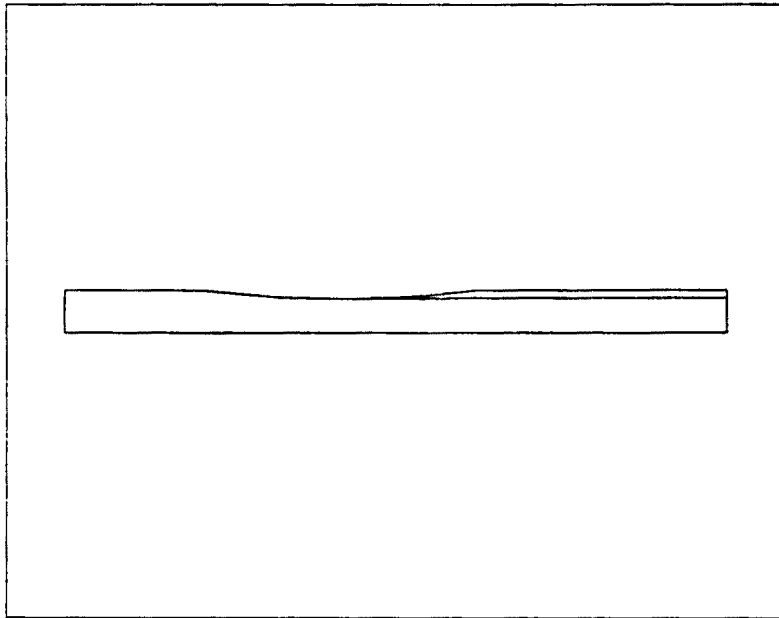


Figure 6.28.(a) Position of the metal front at $t = 5.0$ (HRM).

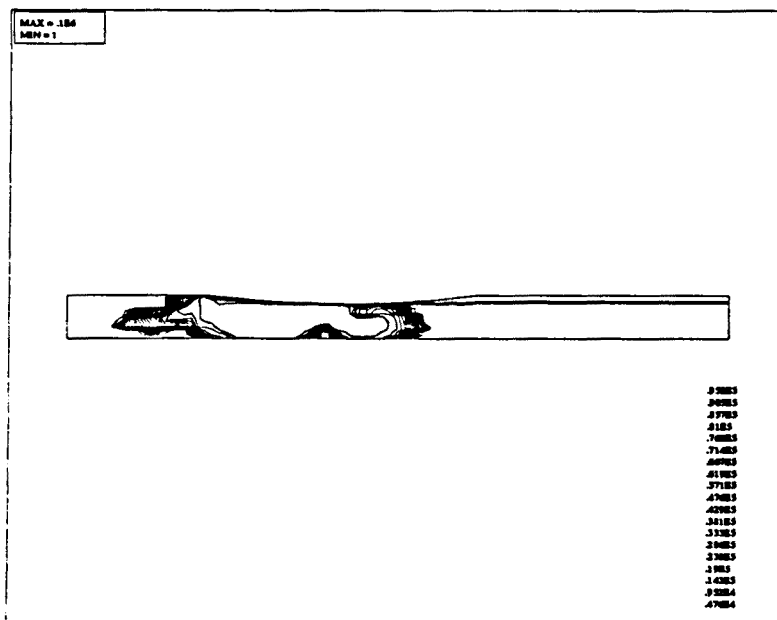


Figure 6.28.(b) Viscosity contours at $t = 5.0$ (HRM).

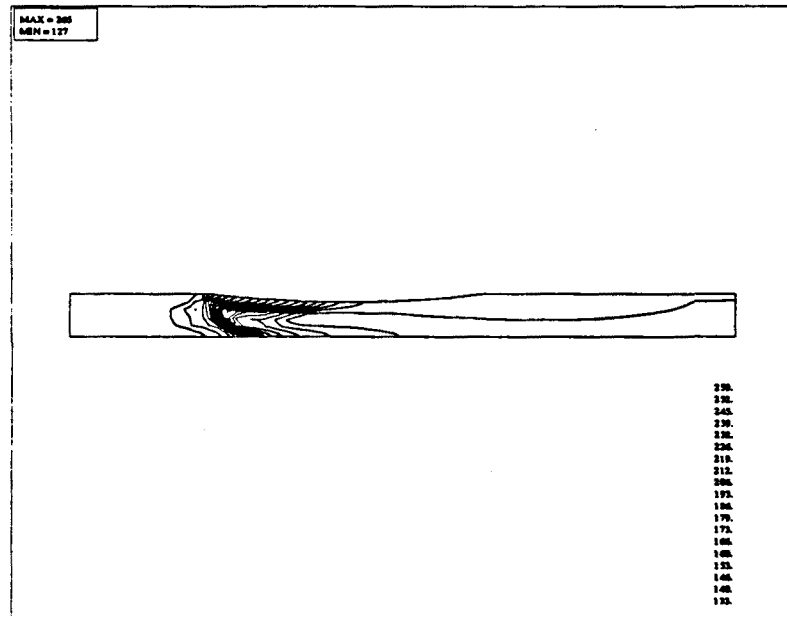


Figure 6.28.(c) Temperature contours at $t = 5.0$ (HRM).

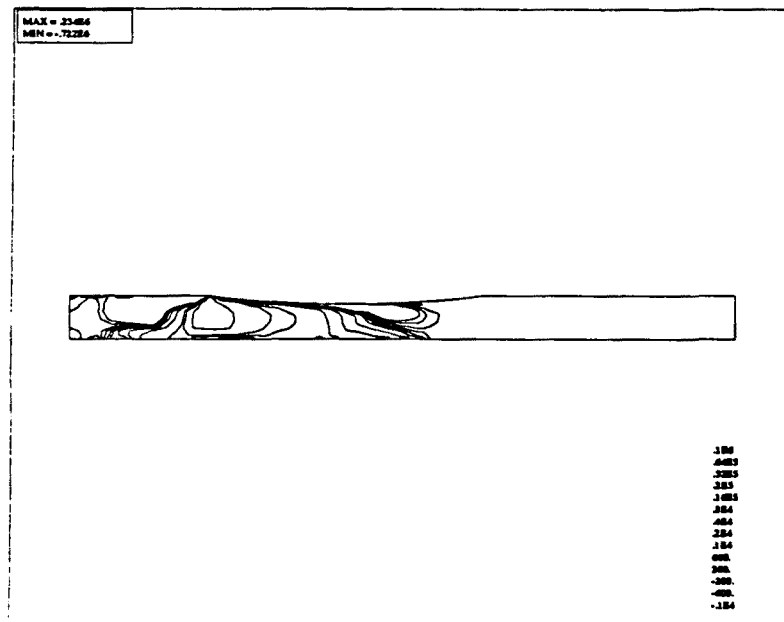


Figure 6.28.(d) Some pressure contours at $t = 5.0$ (HRM).

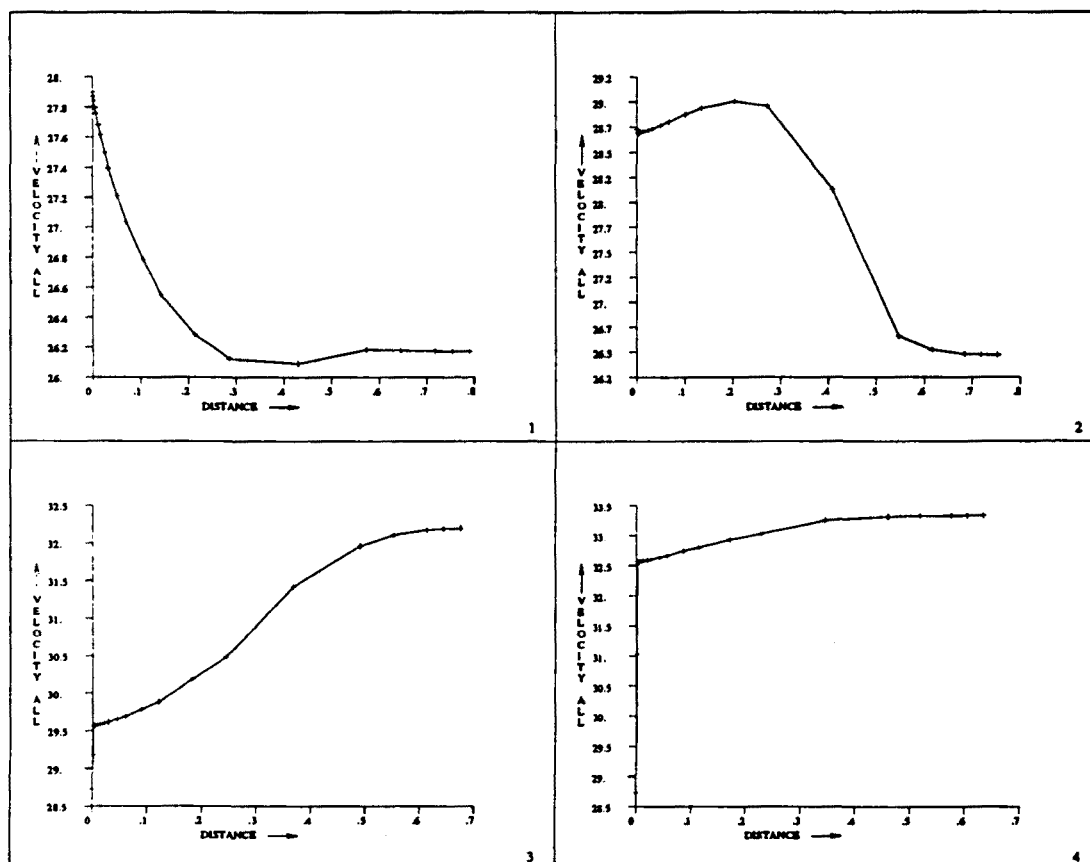


Figure 6.29 Velocity profiles (norms) for some vertical sections at $t = 5.0$ (HRM). (1): Contact with the roll ($x = 2.5$); (2): $x = 3.125$; (3): $x = 3.750$; (4): Center of the roll ($x = 5.0$).

The computational domain has been discretized using the mesh shown in Figure 6.25. It consists of 340 Q_2/P_1 elements, with 1449 nodal points. A parameter $\epsilon = 10^{-9}$ has been used for the iterative penalty method. Convective terms for the Navier-Stokes equations have been neglected (creeping flow). Since convection is not very important for the temperature equation either, the Galerkin approach has been used to solve it. The transport of the pseudo-concentration has been solved using the SD formulation, with $\alpha_0 = 0.5$ and $h_0 = 2$. As for the previous examples, within each time step this transport equation is solved first. Since now the flow is thermally coupled, the block iterative scheme described in Chapter 5 has been employed to deal with the mechanical and thermal problems. The time step size has been taken as $\Delta t = 0.01$, using the backward Euler scheme to advance in time. Now, *no smoothing* of the pseudo-concentration has been performed, i.e., a true step function is transported. The metal is defined by the value 1 and the 'air' for the value 0. The metal front is assumed to be defined by $\psi_c = 0.5$.

In all the previous examples, no comment has been made about the numerical integration rule. We have always used the Gauss-Legendre 3×3 integration for the 2D

Q_2/P_1 element. However, when the viscosity at the nodes of the mesh is computed using the least-squares smoothing technique described in Chapter 4, oscillations appear in the vicinity of sharp viscosity gradients, i.e., at the metal front in this case. This problem was not encountered for the problem solved in Section 5.5.3 because the variation we obtained for the viscosity was smooth. In order to avoid this problem, for this particular example we have used the nodal 3×3 rule (Lobatto), i.e., with the integration points placed at the nodes of the elements. Therefore, no smoothing is needed to obtain nodal viscosity values.

Numerical results at different times are shown in Figures 6.26 to 6.28. The position of the metal front, some viscosity contours, temperature contours and pressure contours are plotted for each case. It is observed that the viscosity is low where the temperature is high, in accordance with the constitutive law given by Eqns. (5.58), (6.26) and (6.27). High temperatures appear in the region where the strain rate is higher, arising from the transformation of plastic work into heat. From Figure 6.19 it is observed how the swelling effect is perfectly well reproduced using the pseudo-concentration technique.

The velocity profiles for different sections $x = \text{const.}$ are shown in Figure 6.29. Recalling that the velocity of the roll is 28.73 cm/s, it is observed that the no-slip point is placed approximately at $x = 3.125$. The relative velocity between the roll and the metal depends on the friction coefficient to be used for the narrow elements in contact with the roll.

We have also computed the roll force and the roll torque, assuming as in Reference [ZOH] that the former acts midway along the angular arc of contact and that it is directed towards the roll center. The values we have obtained are $F = 0.4955$ N for the force and $T = 0.1207$ N cm for the torque.

Concerning the numerical behavior of the algorithm, between three and six iterations have been needed to convergence for a tolerance of the 0.1% in the relative L^2 norm. The iterative penalty method yields a value of order 10^{-8} for the norm of the incompressibility constraint, starting from a value of order 10^{-5} .

The computational cost of the simulation has been of 14.16 CPU seconds per iteration (solution of the Stokes problem, temperature equation and updating of the physical properties, including the calculation of the viscosity). The 68.77% of the total CPU time has been needed to solve the Stokes equations, the 19.64% for the temperature equation and the 8.69% for the pseudo-concentration transport equation.

6.6 Summary and conclusions

In this chapter we have given a complete description of the pseudo-concentration technique as a numerical method to track free surfaces of viscous incompressible flows. Besides the application of the techniques developed in the previous chapters applied now to the solution of its transport equation (Streamline Diffusion formulation, generalized trapezoidal rule to advance in time), some specific issues have been introduced here. The most important one is with no doubt the introduction of temporary holes on the walls in order to allow the air release, an essential ingredient for the success of this method. Two aspects have to be considered when one deals with temporary free wall nodes. The first is that one must check whether the fluid has touched the wall or not and, if so, to block the holes. The other is that the sign of the normal velocity has to be computed. If the velocity points into the computational domain, the temporary

free node is part of the inflow boundary and thus the pseudo-concentration must be prescribed there. We have used a penalty technique to prescribe both the velocity and the pseudo-concentration when necessary.

Also related to the free surface tracking, some problems arising from the smoothing technique described here have been noticed. In particular, special reference has been given to the calculation of the distance from a certain point to the fluid front.

Concerning the solution of the Navier-Stokes and temperature equations when a free surface has to be simulated, a comprehensive description of the transient algorithm, its implications and some approximations has been given. Once again, the numerical methods developed previously have demonstrated their robustness for the problem considered in this chapter. The SD formulation and the iterative penalty method have been shown to be extremely effective.

The main interest of this chapter relies however on the numerical results that have been presented. They show that the pseudo-concentration technique is an effective method to track free surfaces with complicated shapes. If the physical properties of the fictitious material are properly chosen, its motion does not affect that of the fluid that one wishes to analyse. It is observed that pressure gradients are rapidly dissipated in the region occupied by this fictitious fluid. Moreover, an accurate thermal analysis can be performed. This is an aspect of vital importance in casting applications, the subject that has motivated the work presented in this chapter.

References

- [AI] H.J. Antúnez and S.R. Idelshon. Using pseudo-concentrations in the analysis of transient forming processes. *Engineering Computations* (to appear).
- [AID] H.J. Antúnez, S.R. Idelshon and E.N. Dvorkin. Metal forming analysis by Fourier series expansion and further uses of pseudo-concentrations. *Computers & Structures* (to appear).
- [DGB] G. Dhatt, D.M. Gao and A. Ben Cheikh. A finite element simulation of metal flow in moulds. *Int. J. Numer. Meth. Engrg.*, vol. 30 (1990), 821–831
- [DP] E. Dvorkin and E.G. Petöcz. On the modelling of 2D metal forming processes using the flow formulation and the pseudo-concentration technique. In: *COMPLAS III*. Proceedings of the 3rd International Conference on Computational Plasticity, Barcelona, Spain (Pineridge Press/CIMNE, 1992).
- [HN] C.W. Hirt and B.D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *J. Comput. Phys.*, vol. 39 (1981), 201–225
- [Hu] J. Huetink. Analysis of metal forming processes based on a combined Eulerian-Lagrangian finite element formulation. In: *Numerical analysis of forming processes*. J.F.T. Pittman, O.C. Zienkiewicz, R.D. Wood and J.M. Alexander (eds.) (Wiley, 1984).
- [HS] W.S. Hwang and R.S. Stoehr. Molten metal flow prediction for complete solidification analysis of near net shape castings. *Materials Science Technology*, vol. 4 (1988), 240–250
- [FCT] A. Fortin, D. Cote and P.A. Tanguy. On the imposition of friction boundary conditions for the numerical simulation of Bingham fluid flows. *Comput. Meth. Appl. Mech. Engrg.*, vol. 88 (1991), 97–109

- [LDD] Y.S. Lee, P.R. Dawson and T.B. Dewhurst. Bulge predictions in steady state bar rolling processes. *Int. J. Numer. Meth. Engrg.*, vol. 30 (1990), 1403–1413
- [LUC] R.W. Lewis, A.S. Usmani and J.T. Cross. Finite element modelling of mould filling. In: *Finite elements in the 90's*. E. Oñate, J. Periaux, A. Samuelson (eds.) (Springer-Verlag/CIMNE, 1991)
- [RA] P. Le Roy and P. Angibault. Remplissage d'un 1/2 moule sable par du Gallium: mesures, visualisations et comparaisons avec le calcul. *Direction de Méthodes Organes Mécaniques, RENAULT – Service 0968*. Note de Service N° 90/299.
- [SW] T.J. Smith and D.B. Welbourn. The integration of geometric modelling with finite element analysis for the computer-aided design of castings. *Applied Scientific Research*, vol. 44 (1987), 139–160
- [SFD] A. Soulaïmani, M. Fortin, G. Dhatt and Y. Ouellet. Finite element simulation of two- and three-dimensional free surface flows. *Comput. Meth. Appl. Mech. Engrg.*, vol. 86 (1991), 265–296
- [Th1] E. Thompson. Use of pseudo-concentrations to follow creeping viscous flows during transient analysis. *Int. J. Numer. Meth. Fluids*, vol. 6 (1986), 749–761
- [Th2] E. Thompson. Transient analysis of metal forming operations using pseudo-concentrations. In: *Numiform 86*. Proceedings of the 2nd International Conference on Numerical Methods for Industrial Forming Processes, Göteborg, Sweden (A. Balkema, 1986).
- [TS] E. Thompson and R.E. Smelser. Transient analysis of forging operations by the pseudo-concentration method. *Int. J. Numer. Meth. Engrg.*, vol. 25 (1988), 177–189
- [WC] O. Wambersie and M.J. Crochet. Transient finite element method for calculating steady-state three-dimensional free surfaces. *Int. J. Numer. Meth. Fluids*, vol. 14 (1992), 343–360
- [Zi] O.C. Zienkiewicz. Flow formulation for the numerical solution of forming processes. In: *Numerical analysis of forming processes*. J.F.T. Pittman, O.C. Zienkiewicz, R.D. Wood and J.M. Alexander (eds.) (Wiley, 1984).
- [ZJO] O.C. Zienkiewicz, P.C. Jain and E. Oñate. Flow of solids during forming and extrusion: some aspects of numerical solution. *Int. J. Numer. Meth. Engrg.*, vol. 14 (1978), 15–38 (1978)
- [ZOH] O.C. Zienkiewicz, E. Oñate and J.C. Heinrich. A general formulation for coupled thermal flow of metals using finite elements. *Int. J. Numer. Meth. Engrg.*, vol. 17 (1981), 1497–1514
- [ZT] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. Fourth Edition, vol. 1 (McGraw-Hill, 1989)

AFTERWORD

It is not my purpose now to give a detailed description of the results obtained in this work, but rather to give an overall assessment about the numerical behavior of the techniques developed here and to discuss some possible future research lines that could emanate from this thesis.

The first question is the extension of the Streamline Diffusion (SD) method to problems with very sharp gradients of the solution or even discontinuities. The SD is not a monotone method and therefore small oscillations still remain in the vicinity of sharp gradients (internal or boundary layers). Nonlinear algorithms have to be devised to overcome this problem, adding the so called discontinuity or shock capturing terms. This is a point that has not been treated in this work and that deserves further research. A method based on the introduction of a nonlinear crosswind diffusion designed in order to satisfy the discrete maximum principle is currently being developed. In any case, the original version of the SD method has proved to be very effective for the problems considered in this work, both using linear and quadratic finite elements.

Time stepping algorithms is another point of interest. The discontinuous Galerkin method for the time discretization is very attractive, at least from the conceptual point of view. Nevertheless, I think that further experience is needed and that appropriate iterative algorithms have to be developed to convince users that the additional computational cost that it involves compared to finite differences is worth affording. The trapezoidal rule employed here yields satisfactory results, but maybe in a near future the finite element method will also dominate the time domain and space-time unstructured finite element meshes will prevail over classical finite difference techniques.

Concerning the treatment of the incompressibility constraint of the Navier-Stokes equations, the discussion is focussed on the use of stabilization techniques, such as the Galerkin/least-squares (GLS) method, or mixed finite element interpolations. The former simplifies the computer implementation, the latter allows the use of penalty methods and therefore the reduction of the number of unknowns. If iterative methods have to be used to solve linear systems, the main drawback of penalty methods is the ill-conditioning of the matrix. This can be alleviated using the iterative penalization analysed in this work, whose behavior in general situations has been found to be excellent. Numerical experiments have to help to decide in favor of one approach or the other, although no absolute answer is to be expected.

From the theoretical standpoint, it would be interesting to extend the analysis of the iterative penalty method of Chapter 3 to branches of nonsingular solutions for the stationary Navier-Stokes equations. The condition that ensures uniqueness should be replaced by the weaker condition of local invertibility of the Navier-Stokes operator. I don't expect much changes in the conclusions. Another question that still remains open is the convergence of the SD method presented in Chapter 4, at least for simple problems. As mentioned in the text, some partial results are known for the GLS approach. The idea would be to drop the pressure stabilization term of this method and to assume that the Babuška-Brezzi stability condition holds.

Whichever the basic numerical tool for the solution of the Navier-Stokes equations be, it will not be enough to solve many real flow problems. Micro-scale phenomena are beyond the actual possibilities of discretization and the need for appropriate turbulence models emerges. This is not only a numerical problem, but a deeper understanding of

Afterword

turbulent behavior is needed. The implementation of some more or less well established turbulence models must be a forthcoming step in the development of the code written for this thesis.

Besides the applications to thermally coupled flows, flows of nonlinear materials and free surface tracking presented in chapters 5 and 6, a lot of possibilities are open once the basic finite element model is established. Considering only incompressible flow problems, the numerical simulation of reactive flows, viscoelastic materials, electromagnetically coupled flows and numerical techniques such as error estimation, mesh adaptivity and fast solvers are some of the fascinating challenges that computational fluid dynamics has ahead. I expect that the techniques presented in this thesis and the code that implements them will be a valid starting point.

