

# Computational Fluid Dynamics

*Howard H. Hu*

## OUTLINE

10.1. Introduction	421	10.6. Concluding Remarks	470
10.2. Finite-Difference Method	423	Exercises	470
10.3. Finite-Element Method	429	Literature Cited	471
10.4. Incompressible Viscous Fluid Flow	436	Supplemental Reading	472
10.5. Three Examples	449		

## CHAPTER OBJECTIVES

- To introduce the techniques for computational solutions of the fluid dynamic equations of motion
- To describe the finite-difference and finite-element formulations
- To specify finite-difference and finite-element equations for incompressible viscous flow
- To illustrate use of these equations via example calculations

## 10.1. INTRODUCTION

Computational fluid dynamics (CFD) is a science that, with the help of digital computers, produces quantitative predictions of fluid-flow phenomena based on the conservation laws (conservation of mass, momentum, and energy) governing fluid motion. These predictions

normally occur under those conditions defined in terms of flow geometry, the physical properties of a fluid, and the boundary and initial conditions of a flow field. Such predictions generally concern sets of values of the flow variables, for example, velocity, pressure, or temperature at selected locations in the domain and for selected times. The predictions may also involve evaluations of overall flow behavior, such as the flow rate or the hydrodynamic force acting on an object in the flow.

During the past several decades different types of numerical methods have been developed to simulate fluid flows involving a wide range of applications. These methods include finite-difference, finite-element, finite-volume, and spectral methods. Some of them are discussed in this chapter.

As time has passed, CFD has increased in importance and in accuracy; however, its predictions are never completely exact. Because many potential sources of error may be involved, one has to be very careful when interpreting the results produced by CFD techniques. The most common sources of error are:

- *Discretization error.* This error is intrinsic to all numerical methods, and is incurred whenever a continuous system is approximated by a discrete one. For example, a finite number of locations in space (grid points) or instants of time may be used to resolve the flow field. Different numerical schemes may have different orders of magnitude of the discretization error. Even with the same method, the discretization error will be different depending upon the distribution of grid points used in a simulation. In most applications, one needs to properly select a numerical method and choose a grid to control discretization error.
- *Input data error.* This error arises from the fact that both the flow geometry and fluid properties may be known to only a certain level of precision, or possibly in only an approximate way.
- *Initial and boundary condition error.* It is common that the initial and boundary conditions of a flow field may represent the real situation with imperfect precision. For example, flow information is needed at locations where fluid enters and leaves the computational domain. Here, flow properties may not be known exactly and are thus approximated to complete a numerical calculation.
- *Modeling error.* More complicated flows may involve physical phenomena that are not perfectly described by current scientific theories. Models used to solve these problems certainly contain errors, for example, turbulence modeling, atmospheric modeling, polymeric-fluid constitutive modeling, multiphase flow modeling, and so on.

As a research and design tool, CFD normally complements experimental and theoretical fluid dynamics. However, CFD has a number of distinct advantages:

- It can be produced inexpensively and quickly, without an extraordinary amount of training, although interpreting results often requires experience. Yet, while the price of many commodities increases, computing costs are falling. According to Moore's law (Intel Corporation, 2003) based on the observation of the data for the last 40 years, computational power will double every two years into the foreseeable future.

- It generates complete information. Full-field CFD produces detailed and comprehensive information of all relevant variables throughout the domain of interest. This information can also be easily accessed.
- It allows easy parameter changes. CFD permits input parameters to be varied easily over wide ranges, thereby facilitating design optimization. Such variations are often either impossible or prohibitively expensive in experimental studies.
- It has the ability to simulate realistic conditions. CFD can simulate flows directly under practical conditions, unlike experiments, where a small- or a large-scale model may be needed, or analytical theories that may only be valid for limiting cases where one parameter or another is very large or small.
- It has the ability to simulate ideal conditions. CFD provides the convenience of switching off certain terms in the governing equations, which allows one to focus attention on a few essential parameters and eliminate all irrelevant features. Such parametric control is typically impossible in experiments.
- It permits investigation of unnatural or unwanted situations. CFD allows events to be studied so that every attempt is made to prevent, for example, conflagrations, explosions, or nuclear power plant failures.

The remainder of the chapter provides a self-contained survey of CFD techniques, so that the interested reader, who might further investigate CFD, will be aware of language and techniques when pursuing more detailed sources and current literature.

## 10.2. FINITE-DIFFERENCE METHOD

---

The key to various numerical methods is to convert the partial differential equations that govern a physical phenomenon into a system of algebraic equations. Different techniques are available for this conversion. The finite-difference method is one of the most commonly used.

### Approximation to Derivatives

Consider the one-dimensional transport equation,

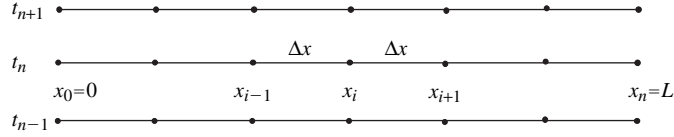
$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = D \frac{\partial^2 T}{\partial x^2} \quad \text{for } 0 \leq x \leq L. \quad (10.1)$$

This is the classic convection-diffusion problem for the scalar  $T(x, t)$ , where  $u$  is a convective velocity and  $D$  is a diffusion coefficient. For simplicity, assume that  $u$  and  $D$  are constants. This equation is written in dimensional form. The boundary conditions for this problem are

$$T(0, t) = g \quad \text{and} \quad \frac{\partial T}{\partial x}(L, t) = q, \quad (10.2)$$

where  $g$  and  $q$  are two constants. The initial condition is

$$T(x, 0) = T_0(x) \quad \text{for } 0 \leq x \leq L, \quad (10.3)$$



**FIGURE 10.1** Uniform grid in space and time. The time epoch is listed at the left. The horizontal lines indicate the spatial domain. The small black dots indicate the spatial grid points where field quantities are determined.

where  $T_0(x)$  is a given function that satisfies the boundary conditions (10.2).

Let us first discretize the transport equation (10.1) on a uniform grid with grid spacing  $\Delta x$ , as shown in Figure 10.1. Equation (10.1) is evaluated at spatial location  $x = x_i$  and time  $t = t_n$ . Define  $T(x_i, t_n)$  as the exact value of  $T$  at the location  $x = x_i$  and time  $t = t_n$ , and let  $T_i^n$  be its approximation. Using the Taylor series expansion, we have:

$$T_{i+1}^n = T_i^n + \Delta x \left[ \frac{\partial T}{\partial x} \right]_i^n + \frac{\Delta x^2}{2} \left[ \frac{\partial^2 T}{\partial x^2} \right]_i^n + \frac{\Delta x^3}{6} \left[ \frac{\partial^3 T}{\partial x^3} \right]_i^n + \frac{\Delta x^4}{24} \left[ \frac{\partial^4 T}{\partial x^4} \right]_i^n + O(\Delta x^5), \quad (10.4)$$

$$T_{i-1}^n = T_i^n - \Delta x \left[ \frac{\partial T}{\partial x} \right]_i^n + \frac{\Delta x^2}{2} \left[ \frac{\partial^2 T}{\partial x^2} \right]_i^n - \frac{\Delta x^3}{6} \left[ \frac{\partial^3 T}{\partial x^3} \right]_i^n + \frac{\Delta x^4}{24} \left[ \frac{\partial^4 T}{\partial x^4} \right]_i^n + O(\Delta x^5), \quad (10.5)$$

where  $O(\Delta x^5)$  means terms of the order of  $\Delta x^5$ . Therefore, the first spatial derivative may be approximated as

$$\begin{aligned} \left[ \frac{\partial T}{\partial x} \right]_i^n &= \frac{T_{i+1}^n - T_i^n}{\Delta x} + O(\Delta x) \quad (\text{forward difference}) \\ &= \frac{T_i^n - T_{i-1}^n}{\Delta x} + O(\Delta x) \quad (\text{backward difference}) \\ &= \frac{T_{i+1}^n - T_{i-1}^n}{2\Delta x} + O(\Delta x^2) \quad (\text{centered difference}) \end{aligned} \quad (10.6)$$

and the second-order derivative may be approximated as

$$\left[ \frac{\partial^2 T}{\partial x^2} \right]_i^n = \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} + O(\Delta x^2). \quad (10.7)$$

The orders of accuracy of the approximations (truncation errors) are also indicated in the expressions of (10.6) and (10.7). Higher order accuracy is generally desirable since the number of grid points is often a limitation for the size of a CFD computation. More accurate approximations generally require more values of the variable on the neighboring grid points. Similar expressions to (10.6) and (10.7) can be derived for nonuniform grids.

In the same fashion, the time derivative can be discretized as

$$\begin{aligned}
 \left[ \frac{\partial T}{\partial t} \right]_i^n &= \frac{T_i^{n+1} - T_i^n}{\Delta t} + O(\Delta t) \\
 &= \frac{T_i^n - T_i^{n-1}}{\Delta t} + O(\Delta t) \\
 &= \frac{T_i^{n+1} - T_i^{n-1}}{2\Delta t} + O(\Delta t^2),
 \end{aligned} \tag{10.8}$$

where  $\Delta t = t_{n+1} - t_n = t_n - t_{n-1}$  is the constant time step.

### Discretization and Its Accuracy

A discretization of the transport equation (10.1) is obtained by evaluating the equation at fixed spatial and temporal grid points and using the approximations for the individual derivative terms listed in the preceding section. When the first expression in (10.8) is used, together with (10.7) and the centered difference in (10.6), (10.1) may be discretized by:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} + u \frac{T_{i+1}^n - T_{i-1}^n}{2\Delta x} = D \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} + O(\Delta t, \Delta x^2), \tag{10.9}$$

or

$$\begin{aligned}
 T_i^{n+1} &\approx T_i^n - u\Delta t \frac{T_{i+1}^n - T_{i-1}^n}{2\Delta x} + D\Delta t \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} \\
 &= T_i^n - \alpha(T_{i+1}^n - T_{i-1}^n) + \beta(T_{i+1}^n - 2T_i^n + T_{i-1}^n),
 \end{aligned} \tag{10.10}$$

where

$$\alpha = u \frac{\Delta t}{2\Delta x}, \quad \beta = D \frac{\Delta t}{\Delta x^2}. \tag{10.11}$$

Once the values of  $T_i^n$  are known, starting with the initial condition (10.3), the expression (10.10) simply updates the variable for the next time step  $t = t_{n+1}$ . This scheme is known as an explicit algorithm. The discretization (10.10) is first-order accurate in time and second-order accurate in space.

As another example, when the backward difference expression in (10.8) is used, we will have

$$\frac{T_i^n - T_i^{n-1}}{\Delta t} + u \frac{T_{i+1}^n - T_{i-1}^n}{2\Delta x} = D \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2} + O(\Delta t, \Delta x^2), \tag{10.12}$$

or

$$T_i^n + \alpha (T_{i+1}^n - T_{i-1}^n) - \beta (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \approx T_i^{n-1}. \quad (10.13)$$

At each time step  $t = t_n$ , a system of algebraic equations needs to be solved to advance the solution. This scheme is known as an *implicit algorithm*. Obviously, for the same accuracy, the explicit scheme (10.10) is much simpler than the implicit one (10.13). However, the explicit scheme has limitations.

### Convergence, Consistency, and Stability

The result from the solution of the explicit scheme (10.10) or the implicit scheme (10.13) represents an approximate numerical solution to the original partial differential equation (10.1). One certainly hopes that the approximate solution will be close to the exact one. Thus we introduce the concepts of *convergence*, *consistency*, and *stability* of the numerical solution.

The approximate solution is said to be *convergent* if it approaches the exact solution as the grid spacings  $\Delta x$  and  $\Delta t$  tend to zero. We may define the solution error as the difference between the approximate solution and the exact solution,

$$e_i^n = T_i^n - T(x_i, t_n). \quad (10.14)$$

Thus, the approximate solution converges when  $e_i^n \rightarrow 0$  as  $\Delta x, \Delta t \rightarrow 0$ . For a convergent solution, some measure of the solution error can be estimated as

$$\|e_i^n\| \leq K \Delta x^a \Delta t^b, \quad (10.15)$$

where the measure may be the root mean square (rms) of the solution error on all the grid points;  $K$  is a constant independent of the grid spacing  $\Delta x$  and the time step  $\Delta t$ ; and the indices  $a$  and  $b$  represent the convergence rates at which the solution error approaches zero.

One may reverse the discretization process and examine the limit of the discretized equations (10.10) and (10.13), as the grid spacing tends to zero. The discretized equation is said to be *consistent* if it recovers the original partial differential equation (10.1) in the limit of zero grid spacing.

Let us consider the explicit scheme (10.10). Substitution of the Taylor series expansions (10.4) and (10.5) into this scheme (10.10) produces

$$\left[ \frac{\partial T}{\partial t} \right]_i^n + u \left[ \frac{\partial T}{\partial x} \right]_i^n - D \left[ \frac{\partial^2 T}{\partial x^2} \right]_i^n + E_i^n = 0, \quad (10.16)$$

where

$$E_i^n = \frac{\Delta t}{2} \left[ \frac{\partial^2 T}{\partial t^2} \right]_i^n + u \frac{\Delta x^2}{6} \left[ \frac{\partial^3 T}{\partial x^3} \right]_i^n - D \frac{\Delta x^2}{12} \left[ \frac{\partial^4 T}{\partial x^4} \right]_i^n + O(\Delta t^2, \Delta x^4) \quad (10.17)$$

is the truncation error. Obviously, as the grid spacing  $\Delta x, \Delta t \rightarrow 0$ , this truncation error is of the order of  $O(\Delta t, \Delta x^2)$  and tends to zero. Therefore, the explicit scheme (10.10) or expression

(10.16) recovers the original partial differential equation (10.1), so it is consistent. It is said to be first-order accurate in time and second-order accurate in space, according to the order of magnitude of the truncation error.

In addition to the truncation error introduced in the discretization process, other sources of error may be present in the approximate solution. Spontaneous disturbances (such as the round-off error produced from the finite precision arithmetic used by computers) may be introduced during either the evaluation or the numerical solution process. A numerical approximation is said to be *stable* if these disturbances decay and do not affect the solution.

The stability of the explicit scheme (10.10) may be examined using the von Neumann method in which numerical errors are represented via a Fourier decomposition. Let us consider the error at a grid point,

$$\xi_i^n = T_i^n - \bar{T}_i^n, \quad (10.18)$$

where  $T_i^n$  is the exact solution of the discretized system (10.10) and  $\bar{T}_i^n$  is the approximate numerical solution of the same system. This error could be introduced due to the round-off error at each step of the computation. We need to monitor its decay/growth with time. It can be shown that the evolution of this error satisfies the same homogeneous algebraic system (10.10) or

$$\xi_i^{n+1} = (\alpha + \beta) \xi_{i-1}^n + (1 - 2\beta) \xi_i^n + (\beta - \alpha) \xi_{i+1}^n. \quad (10.19)$$

The error distributed along the grid line can always be decomposed in Fourier space as

$$\xi_i^n = \sum_{k=-\infty}^{\infty} g^n(k) e^{i\pi k x_i}, \quad (10.20)$$

where  $i = \sqrt{-1}$ ,  $k$  is the wave number in Fourier space, and  $g^n$  represents the function  $g$  at time  $t = t_n$ . As the system is linear, we can examine one component of (10.20) at a time,

$$\xi_i^n = g^n(k) e^{i\pi k x_i}. \quad (10.21)$$

The component at the next time increment has a similar form:

$$\xi_i^{n+1} = g^{n+1}(k) e^{i\pi k x_i}. \quad (10.22)$$

Substituting the preceding two equations (10.21) and (10.22) into error equation (10.19), we obtain

$$g^{n+1} e^{i\pi k x_i} = g^n [(\alpha + \beta) e^{i\pi k x_{i-1}} + (1 - 2\beta) e^{i\pi k x_i} + (\beta - \alpha) e^{i\pi k x_{i+1}}] \quad (10.23)$$

or

$$\frac{g^{n+1}}{g^n} = [(\alpha + \beta) e^{-i\pi k \Delta x} + (1 - 2\beta) + (\beta - \alpha) e^{i\pi k \Delta x}]. \quad (10.24)$$

This ratio  $g^{n+1}/g^n$  is called the *amplification factor*. The condition for stability is that the magnitude of the error should decay with time, or

$$\left| \frac{g^{n+1}}{g^n} \right| \leq 1, \quad (10.25)$$

for any value of the wave number  $k$ . For this explicit scheme, the condition for stability (10.25) can be expressed as

$$\left(1 - 4\beta \sin^2\left(\frac{\theta}{2}\right)\right)^2 + (2\alpha \sin \theta)^2 \leq 1, \quad (10.26)$$

where  $\theta = k\pi \Delta x$ . The stability condition (10.26) also can be expressed as (Noye, 1983),

$$0 \leq 4\alpha^2 \leq 2\beta \leq 1. \quad (10.27)$$

For the pure diffusion problem ( $u = 0$ ), the stability condition (10.27) for this explicit scheme requires that

$$0 \leq \beta \leq \frac{1}{2} \quad \text{or} \quad \Delta t \leq \frac{1}{2} \frac{\Delta x^2}{D}, \quad (10.28)$$

which limits the size of the time step. For the pure convection problem ( $D = 0$ ), condition (10.27) will never be satisfied, which indicates that the scheme is always unstable and it means that any error introduced during the computation will explode with time. Thus, this explicit scheme is useless for pure convection problems. To improve the stability of the explicit scheme for the convection problem, one may use an upwind scheme to approximate the convective term,

$$T_i^{n+1} = T_i^n - 2\alpha (T_i^n - T_{i-1}^n), \quad (10.29)$$

where the stability condition requires that

$$u \frac{\Delta t}{\Delta x} \leq 1. \quad (10.30)$$

The condition (10.30) is known as the *Courant-Friedrichs-Lewy (CFL) condition*. This condition indicates that a fluid particle should not travel more than one spatial grid in one time step.

It can easily be shown that the implicit scheme (10.13) is also consistent and unconditionally stable.

It is normally difficult to show the convergence of an approximate solution theoretically. However, the *Lax Equivalence Theorem* (Richtmyer & Morton, 1967) states that: *for an approximation to a well-posed linear initial value problem, which satisfies the consistency condition, stability is a necessary and sufficient condition for the convergence of the solution.*

For convection-diffusion problems, the exact solution may change significantly in a narrow boundary layer. If the computational grid is not sufficiently fine to resolve the rapid variation of the solution in the boundary layer, the numerical solution may present unphysical oscillations adjacent to or in the boundary layer. To prevent the oscillatory solution, a condition on the cell Péclet number (or Reynolds number) is normally required (see Section 10.4),

$$R_{cell} = u \frac{\Delta x}{D} \leq 2. \quad (10.31)$$



### 10.3. FINITE-ELEMENT METHOD

The finite-element method was developed initially as an engineering procedure for stress and displacement calculations in structural analysis, where its success is impressive. The method was subsequently placed on a sound mathematical foundation with a variational interpretation of the potential energy of the system. For most fluid dynamics problems, finite-element applications have used the Galerkin finite-element formulation on which we will focus in this section.

#### Weak or Variational Form of Partial Differential Equations

Let us consider again the one-dimensional transport problem (10.1). The form of (10.1) with the boundary condition (10.2) and the initial conditions (10.3) is called the *strong* (or *classical*) form of the problem.

We first define a collection of trial solutions, which consists of all functions that have square-integrable first derivatives ( $H^1$  functions, i.e.,  $\int_0^L (\partial T / \partial x)^2 dx < \infty$  if  $T \in H^1$ ) and satisfy the Dirichlet type of boundary condition (where the value of the variable is specified) at  $x = 0$ . This is expressed as the trial functional space,

$$S = \{T | T \in H^1, T(0) = g\}. \quad (10.32)$$

The variational space of the trial solution is defined as

$$V = \{w | w \in H^1, w(0) = 0\}, \quad (10.33)$$

which requires a corresponding homogeneous boundary condition.

We next multiply the transport equation (10.1) by a function in the variational space ( $w \in V$ ), and integrate the product over the domain where the problem is defined,

$$\int_0^L \left( \frac{\partial T}{\partial t} w \right) dx + u \int_0^L \left( \frac{\partial T}{\partial x} w \right) dx = D \int_0^L \left( \frac{\partial^2 T}{\partial x^2} w \right) dx. \quad (10.34)$$

Integrating the right-hand side of (10.34) by parts, we have:

$$\int_0^L \left( \frac{\partial T}{\partial t} w \right) dx + u \int_0^L \left( \frac{\partial T}{\partial x} w \right) dx + D \int_0^L \left( \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} \right) dx = D \left[ \frac{\partial T}{\partial x} w \right]_0^L = Dqw(L), \quad (10.35)$$

where the boundary conditions  $\partial T / \partial x = q$  and  $w(0) = 0$  are applied. The integral equation (10.35) is called the *weak* form of this problem. Therefore, the weak form can be stated as: Find  $T \in S$  such that for all  $w \in V$ ,

$$\int_0^L \left( \frac{\partial T}{\partial t} w \right) dx + u \int_0^L \left( \frac{\partial T}{\partial x} w \right) dx + D \int_0^L \left( \frac{\partial T}{\partial x} \frac{\partial w}{\partial x} \right) dx = Dqw(L). \quad (10.36)$$

It can be formally shown that the solution of the weak problem is identical to that of the strong problem, or that the strong and weak forms of the problem are *equivalent*. Obviously, if  $T$  is a solution of the strong problems (10.1) and (10.2), it must also be a solution of the weak

problem (10.36) using the procedure for derivation of the weak formulation. However, let us assume that  $T$  is a solution of the weak problem (10.36). By reversing the order in deriving the weak formulation, we have:

$$\int_0^L \left( \frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} - D \frac{\partial^2 T}{\partial x^2} \right) w dx + D \left[ \frac{\partial T}{\partial x}(L) - q \right] w(L) = 0. \quad (10.37)$$

Satisfying (10.37) for all possible functions of  $w \in V$  requires that

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} - D \frac{\partial^2 T}{\partial x^2} = 0 \text{ for } x \in (0, L), \quad \text{and} \quad \frac{\partial T}{\partial x}(L) - q = 0, \quad (10.38)$$

which means that the solution  $T$  will be also a solution of the strong problem. It should be noted that the Dirichlet type of boundary condition (where the value of the variable is specified) is built into the trial functional space  $S$ , and is thus called an *essential* boundary condition. However, the Neumann type of boundary condition (where the derivative of the variable is imposed) is implied by the weak formulation as indicated in (10.38) and is referred to as a *natural* boundary condition.

### Galerkin's Approximation and Finite-Element Interpolations

As shown earlier, the strong and weak forms of the problem are equivalent, and there is no approximation involved between these two formulations. Finite-element methods start with the weak formulation of the problem. Let us construct finite-dimensional approximations of  $S$  and  $V$ , which are denoted by  $S^h$  and  $V^h$ , respectively. The superscript refers to a discretization with a characteristic grid size  $h$ . The weak formulation (10.36) can be rewritten using these new spaces, as: Find  $T^h \in S^h$  such that for all  $w^h \in V^h$ ,

$$\int_0^L \left( \frac{\partial T^h}{\partial t} w^h \right) dx + u \int_0^L \left( \frac{\partial T^h}{\partial x} w^h \right) dx + D \int_0^L \left( \frac{\partial T^h}{\partial x} \frac{\partial w^h}{\partial x} \right) dx = D q w^h(L). \quad (10.39)$$

Normally,  $S^h$  and  $V^h$  will be subsets of  $S$  and  $V$ , respectively. This means that if a function  $\phi \in S^h$  then  $\phi \in S$ , and if another function  $\psi \in V^h$  then  $\psi \in V$ . Therefore, (10.39) defines an approximate solution  $T^h$  to the exact weak form of the problem (10.36).

It should be noted that, up to the boundary condition  $T(0) = g$ , the function spaces  $S^h$  and  $V^h$  are composed of identical collections of functions. We may take out this boundary condition by defining a new function:

$$v^h(x, t) = T^h(x, t) - g^h(x), \quad (10.40)$$

where  $g^h$  is a specific function that satisfies the boundary condition  $g^h(0) = g$ . Thus, the functions  $v^h$  and  $w^h$  belong to the same space  $V^h$ . Equation (10.39) can be rewritten in terms of the new function  $v^h$ : Find  $T^h = v^h + g^h$ , where  $v^h \in V^h$ , such that for all  $w^h \in V^h$ ,

$$\int_0^L \left( \frac{\partial v^h}{\partial t} w^h \right) dx + a(w^h, v^h) = D q w^h(L) - a(w^h, g^h). \quad (10.41)$$

The operator  $a(\cdot, \cdot)$  is defined as

$$a(w, v) = u \int_0^L \left( \frac{\partial v}{\partial x} w \right) dx + D \int_0^L \left( \frac{\partial v}{\partial x} \frac{\partial w}{\partial x} \right) dx. \quad (10.42)$$

The formulation (10.41) is called a *Galerkin formulation*, because the solution and the variational functions are in the same space. Again, the Galerkin formulation of the problem is an approximation to the weak formulation (10.36). Other classes of approximation methods, called *Petrov-Galerkin methods*, are those in which the solution function may be contained in a collection of functions other than  $V^h$ .

Next we need to explicitly construct the finite-dimensional variational space  $V^h$ . Let us assume that the dimension of the space is  $n$  and that the basis (shape or interpolation) functions for the space are

$$N_A(x), \quad A = 1, 2, \dots, n. \quad (10.43)$$

Each shape function has to satisfy the boundary condition at  $x = 0$ ,

$$N_A(0) = 0, \quad A = 1, 2, \dots, n, \quad (10.44)$$

which is required by the space  $V^h$ . The form of the shape functions will be discussed later. Any function  $w^h \in V^h$  can be expressed as a linear combination of these shape functions:

$$w^h = \sum_{A=1}^n c_A N_A(x), \quad (10.45)$$

where the coefficients  $c_A$  are independent of  $x$  and uniquely define this function. We may introduce one additional function  $N_0$  to specify the function  $g^h$  in (10.40) related to the essential boundary condition. This shape function has the property

$$N_0(0) = 1. \quad (10.46)$$

Therefore, the function  $g^h$  can be expressed as

$$g^h(x) = g N_0(x), \quad \text{and} \quad g^h(0) = g. \quad (10.47)$$

With these definitions, the approximate solution can be written as

$$v^h(x, t) = \sum_{A=1}^n d_A(t) N_A(x), \quad (10.48)$$

and

$$T^h(x, t) = \sum_{A=1}^n d_A(t) N_A(x) + g N_0(x), \quad (10.49)$$

where  $d_A$  is a function of time only for time-dependent problems.

## Matrix Equations, Comparison with Finite-Difference Method

With the construction of the finite-dimensional space  $V^h$ , the Galerkin formulation of the problem (10.41) leads to a coupled system of ordinary differential equations. Substitution of

the expressions for the variational function (10.45) and for the approximate solution (10.48) into the Galerkin formulation (10.41) yields

$$\begin{aligned} & \int_0^L \left( \sum_{B=1}^n \dot{d}_B N_B \sum_{A=1}^n c_A N_A \right) dx + a \left( \sum_{A=1}^n c_A N_A, \sum_{B=1}^n d_B N_B \right) \\ &= Dq \sum_{A=1}^n c_A N_A(L) - a \left( \sum_{A=1}^n c_A N_A, g N_0 \right) \end{aligned} \quad (10.50)$$

where  $\dot{d}_B = d(d_B)/dt$ . Rearranging the terms, (10.50) reduces to

$$\sum_{A=1}^n c_A G_A = 0, \quad (10.51)$$

where

$$G_A = \sum_{B=1}^n \dot{d}_B \int_0^L (N_A N_B) dx + \sum_{B=1}^n d_B a(N_A, N_B) - Dq N_A(L) + g a(N_A, N_0). \quad (10.52)$$

As the Galerkin formulation (10.41) should hold for all possible functions of  $w^h \in V^h$ , the coefficients,  $c_A$ , should be arbitrary. The necessary requirement for (10.51) to hold is that each  $G_A$  must be zero, that is,

$$\sum_{B=1}^n \dot{d}_B \int_0^L (N_B N_A) dx + \sum_{B=1}^n d_B a(N_A, N_B) = Dq N_A(L) - g a(N_A, N_0), \quad (10.53)$$

for  $A = 1, 2, \dots, n$ . The system of equations (10.53) constitutes a system of  $n$  first-order ordinary differential equations (ODEs) for  $d_B$ . It can be put into a more concise matrix form. Let us define:

$$\mathbf{M} = [M_{AB}], \quad \mathbf{K} = [K_{AB}], \quad \mathbf{F} = \{F_A\}, \quad \mathbf{d} = \{d_B\}, \quad (10.54)$$

where

$$M_{AB} = \int_0^L (N_A N_B) dx, \quad (10.55)$$

$$K_{AB} = u \int_0^L (N_{B,x} N_A) dx + D \int_0^L (N_{B,x} N_{A,x}) dx, \quad (10.56)$$

$$F_A = Dq N_A(L) - gu \int_0^L (N_{0,x} N_A) dx - gD \int_0^L (N_{0,x} N_{A,x}) dx. \quad (10.57)$$

Equation (10.53) can then be written as

$$\mathbf{M}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{F}. \quad (10.58)$$

The system of equations (10.58) is also termed the *matrix form* of the problem. Usually,  $\mathbf{M}$  is called the *mass matrix*,  $\mathbf{K}$  is the *stiffness matrix*,  $\mathbf{F}$  is the *force vector*, and  $\mathbf{d}$  is the *displacement vector*. This system of ODEs can be integrated by numerical methods, for example

Runge-Kutta methods, or discretized (in time) by finite-difference schemes as described in the previous section. The initial condition (10.3) will be used for integration. An alternative approach is to use a finite-difference approximation to the time-derivative term in the transport equation (10.1) at the beginning of the process, for example, by replacing  $\partial T / \partial t$  with  $(T^{n+1} - T^n) / \Delta t$ , and then using the finite-element method to discretize the resulting equation.

Now let us consider the actual construction of the shape functions for the finite-dimensional variational space. The simplest example is to use piecewise-linear, finite-element space. We first partition the domain  $[0, L]$  into  $n$  nonoverlapping subintervals (elements). A typical one is denoted as  $[x_A, x_{A+1}]$ . The shape functions associated with the interior nodes,  $A = 1, 2, \dots, n - 1$ , are defined as

$$N_A(x) = \begin{cases} \frac{x - x_{A-1}}{x_A - x_{A-1}}, & x_{A-1} \leq x < x_A, \\ \frac{x_{A+1} - x}{x_{A+1} - x_A}, & x_A \leq x \leq x_{A+1}, \\ 0, & \text{elsewhere.} \end{cases} \quad (10.59)$$

Further, for the boundary nodes, the shape functions are defined as

$$N_n(x) = \frac{x - x_{n-1}}{x_n - x_{n-1}}, \quad x_{n-1} \leq x \leq x_n, \quad (10.60)$$

and

$$N_0(x) = \frac{x_1 - x}{x_1 - x_0}, \quad x_0 \leq x \leq x_1. \quad (10.61)$$

These shape functions are graphically plotted in Figure 10.2. It should be noted that these shape functions have very compact (local) support and satisfy  $N_A(x_B) = \delta_{AB}$ , where  $\delta_{AB}$  is the Kronecker delta (i.e.,  $\delta_{AB} = 1$  if  $A = B$ , whereas  $\delta_{AB} = 0$  if  $A \neq B$ ).

With the construction of the shape functions, the coefficient,  $d_A$ , in the expression for the approximate solution (10.49) represents the values of  $T^h$  at the nodes  $x = x_A$  ( $A = 1, 2, \dots, n$ ), or

$$d_A = T^h(x_A) = T_A. \quad (10.62)$$

To compare the discretized equations generated from the finite-element method with those from finite-difference methods, we substitute (10.59) into (10.53) and evaluate the integrals. For an interior node  $x_A$  ( $A = 1, 2, \dots, n - 1$ ), we have:

$$\frac{d}{dt} \left( \frac{T_{A-1}}{6} + \frac{2T_A}{3} + \frac{T_{A+1}}{6} \right) + \frac{u}{2h} (T_{A+1} - T_{A-1}) - \frac{D}{h^2} (T_{A-1} - 2T_A + T_{A+1}) = 0, \quad (10.63)$$

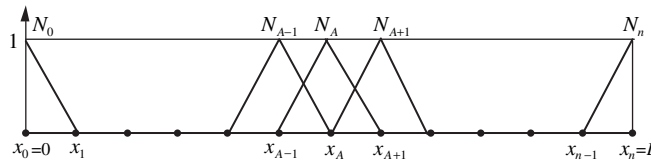


FIGURE 10.2 Piecewise-linear, finite-element space. These shape functions are compact.

where  $h$  is the uniform mesh size. The convective and diffusive terms in expression (10.63) have the same forms as those discretized using the standard second-order finite-difference method (centered difference) in (10.12). However, in the finite-element scheme, the time derivative term is presented with a three-point spatial average of the variable  $T$ , which differs from the finite-difference method. In general, the Galerkin finite-element formulation is equivalent to a finite-difference method. The advantage of the finite-element method lies in its flexibility to handle complex geometries.

### Element Point of View of the Finite-Element Method

So far we have been using a global view of the finite-element method. The shape functions are defined on the global domain, as shown in Figure 10.2. However, it is also convenient to present the finite-element method using a local (or element) point of view. This viewpoint is useful for the evaluation of the integrals in (10.55) through (10.57) and the actual computer implementation of the finite-element method.

Figure 10.3 depicts the global and local descriptions of the  $e$ th element. The global description of the element  $e$  is just the “local” view of the full domain shown in Figure 10.2. Only two shape functions are nonzero within this element,  $N_{A-1}$  and  $N_A$ . Using the local coordinate in the standard element (parent domain) as shown on the right in Figure 10.3, we can write the standard shape functions as

$$N_1(\xi) = \frac{1}{2}(1 - \xi) \quad \text{and} \quad N_2(\xi) = \frac{1}{2}(1 + \xi). \quad (10.64)$$

Clearly, the standard shape function  $N_1$  (or  $N_2$ ) corresponds to the global shape function  $N_{A-1}$  (or  $N_A$ ). The mapping between the domains of the global and local descriptions can easily be generated with the help of these shape functions,

$$x(\xi) = N_1(\xi) x_1^e + N_2(\xi) x_2^e = \frac{1}{2} \left[ (x_A - x_{A-1}) \xi + x_A + x_{A-1} \right], \quad (10.65)$$

with the notation that  $x_1^e = x_{A-1}$  and  $x_2^e = x_A$ . One can also solve (10.65) for the inverse map:

$$\xi(x) = \frac{2x - x_A - x_{A-1}}{x_A - x_{A-1}}. \quad (10.66)$$

Within the element  $e$ , the derivative of the shape functions can be evaluated using the mapping equation (10.66):

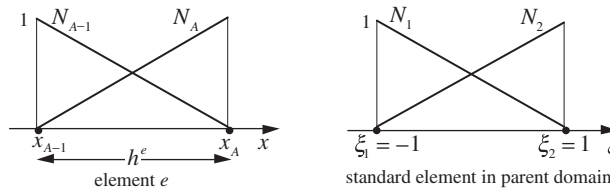


FIGURE 10.3 Global and local descriptions of an element.

$$\frac{dN_A}{dx} = \frac{dN_A}{d\xi} \frac{d\xi}{dx} = \frac{2}{x_A - x_{A-1}} \frac{dN_1}{d\xi} = \frac{-1}{x_A - x_{A-1}} \quad (10.67)$$

and

$$\frac{dN_{A+1}}{dx} = \frac{dN_{A+1}}{d\xi} \frac{d\xi}{dx} = \frac{2}{x_A - x_{A-1}} \frac{dN_2}{d\xi} = \frac{1}{x_A - x_{A-1}}. \quad (10.68)$$

The global mass matrix (10.55), the global stiffness matrix (10.56), and the global force vector (10.57) have been defined as the integrals over the global domain  $[0, L]$ . These integrals may be written as the summation of integrals over each element's domain. Thus

$$\mathbf{M} = \sum_{e=1}^{n_{el}} \mathbf{M}^e, \quad \mathbf{K} = \sum_{e=1}^{n_{el}} \mathbf{K}^e, \quad \mathbf{F} = \sum_{e=1}^{n_{el}} \mathbf{F}^e, \quad (10.69)$$

$$\mathbf{M}^e = [M_{AB}^e], \quad \mathbf{K}^e = [K_{AB}^e], \quad \mathbf{F}^e = \{F_A^e\} \quad (10.70)$$

where  $n_{el}$  is the total number of finite elements (in this case  $n_{el} = n$ ), and

$$M_{AB}^e = \int_{\Omega^e} (N_A N_B) dx, \quad (10.71)$$

$$K_{AB}^e = u \int_{\Omega^e} (N_{B,x} N_A) dx + D \int_{\Omega^e} (N_{B,x} N_{A,x}) dx, \quad (10.72)$$

$$F_A^e = Dq\delta_{en_A}\delta_{An} - gu \int_{\Omega^e} (N_{0,x} N_A) dx - gD \int_{\Omega^e} (N_{0,x} N_{A,x}) dx, \quad (10.73)$$

and  $\Omega^e = [x_1^e, x_2^e] = [x_{A-1}, x_A]$  is the domain of the  $e^{\text{th}}$  element; and the first term on the right-hand side of (10.73) is nonzero only for  $e = n_{el}$  and  $A = n$ .

Given the construction of the shape functions, most of the element matrices and force vectors in (10.71) through (10.73) will be zero. The nonzero ones require that  $A = e$  or  $e + 1$  and  $B = e$  or  $e + 1$ . We may collect these nonzero terms and arrange them into the element mass matrix, stiffness matrix, and force vector as follows:

$$\mathbf{m}^e = [m_{ab}^e], \quad \mathbf{k}^e = [k_{ab}^e], \quad \mathbf{f}^e = \{f_a^e\}, \quad a, b = 1, 2, \quad (10.74)$$

where

$$m_{ab}^e = \int_{\Omega^e} (N_a N_b) dx, \quad (10.75)$$

$$k_{ab}^e = u \int_{\Omega^e} (N_{b,x} N_a) dx + D \int_{\Omega^e} (N_{b,x} N_{a,x}) dx, \quad (10.76)$$

$$f_a^e = \begin{cases} -gk_{a1}^e & e = 1, \\ 0 & e = 2, 3, \dots, n_{el} - 1, \\ Dq\delta_{a2} & e = n_{el}, \end{cases} \quad (10.77)$$

Here,  $\mathbf{m}^e$ ,  $\mathbf{k}^e$ , and  $\mathbf{f}^e$  are defined with the local (element) ordering, and represent the nonzero terms in the corresponding  $\mathbf{M}^e$ ,  $\mathbf{K}^e$ , and  $\mathbf{F}^e$  with the global ordering. The terms in the local ordering need to be mapped back into the global ordering. For this example, the mapping is defined as

$$A = \begin{cases} e - 1 & \text{if } a = 1 \\ e & \text{if } a = 2 \end{cases} \quad (10.78)$$

for element  $e$ .

Therefore, in the element viewpoint, the global matrices and the global vector can be constructed by summing the contributions of the element matrices and the element vector, respectively. The evaluation of the element matrices and the element vector can be performed on a standard element using the mapping between the global and local descriptions.

The finite-element methods for two- or three-dimensional problems will follow the same basic steps introduced in this section. However, the data structure and the forms of the elements or the shape functions will be more complicated. Refer to [Hughes \(1987\)](#) for a detailed discussion. Section 10.5 presents an example of a two-dimensional flow over a circular cylinder.

## 10.4. INCOMPRESSIBLE VISCOUS FLUID FLOW

This section discusses numerical schemes for solving incompressible viscous fluid flows. It focuses on techniques using the primitive variables (velocity and pressure). Other formulations using the stream function and vorticity are available in the literature (see [Fletcher, 1988](#), Vol. II) and will not be discussed here since their extensions to three-dimensional flows are not straightforward. The schemes to be discussed normally apply to laminar flows. However, by incorporating additional appropriate turbulence models, these schemes will also be effective for turbulent flows.

For an incompressible Newtonian fluid, the fluid motion satisfies the constant-viscosity Navier-Stokes equation,

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} \right) = \rho \mathbf{g} - \nabla p + \mu \nabla^2 \mathbf{u}, \quad (10.79)$$

and the continuity equation,

$$\nabla \cdot \mathbf{u} = 0, \quad (10.80)$$

where (as in Chapter 4)  $\mathbf{u}$  is the velocity vector,  $\mathbf{g}$  is the body force per unit mass, which could be the gravitational acceleration,  $p$  is the pressure, and  $\rho$  and  $\mu$  are the density and viscosity of the fluid, respectively. With the proper scaling, (10.79) can be written in the dimensionless form,

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = \mathbf{g} - \nabla p + \frac{1}{\text{Re}} \nabla^2 \mathbf{u}, \quad (10.81)$$



where  $Re$  is the Reynolds number of the flow. In some approaches, the convective term is rewritten in conservative form,

$$(\mathbf{u} \cdot \nabla) \mathbf{u} = \nabla \cdot (\mathbf{u}\mathbf{u}), \quad (10.82)$$

because  $\mathbf{u}$  is solenoidal.

In order to guarantee that a flow problem is well posed, appropriate initial and boundary conditions for the problem must be specified. For time-dependent flow problems, the initial condition for the velocity,

$$\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x}), \quad (10.83)$$

is required. The initial velocity field has to satisfy the continuity equation  $\nabla \cdot \mathbf{u}_0 = 0$ . At a solid surface, the fluid velocity should equal the surface velocity (no-slip condition). No boundary condition for the pressure is required at a solid surface. If the computational domain contains a section where the fluid enters the domain, the fluid velocity (and the pressure) at this inflow boundary should be specified. If the computational domain contains a section where the fluid leaves the domain (outflow section), appropriate outflow boundary conditions include zero tangential velocity and zero normal stress, or zero velocity derivatives, as further discussed in [Gresho \(1991\)](#). Because the conditions at the outflow boundary are artificial, it should be checked that the numerical results are not sensitive to the location of this boundary. In order to solve the Navier-Stokes equations, it is also appropriate to specify the value of the pressure at one reference point in the domain, because the pressure appears only as a gradient and can be determined up to a constant.

There are two major difficulties in solving the Navier-Stokes equations numerically. One is related to the unphysical oscillatory solution often found in a convection-dominated problem. The other is the treatment of the continuity equation that is a constraint on the flow to determine the pressure.

## Convection-Dominated Problems

As mentioned in Section 10.2, the exact solution may change significantly in a thin boundary layer for convection-dominated transport problems. If the computational grid is not sufficiently fine to resolve the rapid variation of the solution in the boundary layer, the numerical solution may present unphysical oscillations adjacent the boundary. Let us examine the steady transport problem in one dimension,

$$u \frac{\partial T}{\partial x} = D \frac{\partial^2 T}{\partial x^2} \quad \text{for } 0 \leq x \leq L, \quad (10.84)$$

with two boundary conditions:

$$T(0) = 0 \quad \text{and} \quad T(L) = 1. \quad (10.85)$$

The exact solution for this problem is

$$T = \frac{e^{Rx/L} - 1}{e^R - 1}, \quad (10.86)$$

where

$$R = uL/D \quad (10.87)$$

is the global Péclet number. For large values of  $R$ , the solution (10.86) behaves as

$$T = e^{-R(1-x/L)}. \quad (10.88)$$

The essential feature of this solution is the existence of a boundary layer at  $x = L$ , and its thickness  $\delta$  is of the order of

$$\frac{\delta}{L} = O\left(\frac{1}{|R|}\right). \quad (10.89)$$

At  $1 - x/L = 1/R$ ,  $T$  is about 37% of the boundary value; while at  $1 - x/L = 2/R$ ,  $T$  is about 13.5% of the boundary value.

If centered differences are used to discretize the steady transport equation (10.84) using the grid shown in Figure 10.1, the resulting finite-difference scheme is

$$\frac{u\Delta x}{2D}(T_{j+1} - T_{j-1}) = (T_{j+1} - 2T_j + T_{j-1}), \quad (10.90)$$

or

$$0.5R_{cell}(T_{j+1} - T_{j-1}) = (T_{j+1} - 2T_j + T_{j-1}), \quad (10.91)$$

where the grid spacing  $\Delta x = L/n$  and the cell Péclet number  $R_{cell} = u\Delta x/D = R/n$ . From the scaling of the boundary thickness (10.89) we know that it is of the order

$$\delta = O\left(\frac{L}{nR_{cell}}\right) = O\left(\frac{\Delta x}{R_{cell}}\right). \quad (10.92)$$

Physically, if  $T$  represents the temperature in the transport problem (10.84), the convective term brings the heat toward the boundary  $x = L$ , whereas the diffusive term conducts the heat away through the boundary. These two terms have to be balanced. The discretized equation (10.91) has the same physical meaning. Let us examine this balance for a node next to the boundary,  $j = n - 1$ . When the cell Péclet number  $R_{cell} > 2$ , according to (10.92) the thickness of the boundary layer is less than half the grid spacing, and the exact solution (10.86) indicates that the temperatures  $T_j$  and  $T_{j-1}$  are already outside the boundary layer and are essentially zero. Thus, the two sides of the discretized equation (10.91) cannot balance, or the conduction term is not strong enough to remove the heat convected to the boundary, assuming the solution is smooth. In order to force the heat balance, an unphysical oscillatory solution with  $T_j < 0$  is generated to enhance the conduction term in the discretized problem (10.91). To prevent the oscillatory solution, the cell Péclet number is normally required to be less than two, which can be achieved by refining the grid to resolve the flow inside the boundary layer. In some respect, an oscillatory solution may be a virtue since it provides a warning that a physically important feature is not being properly resolved. To reduce the overall computational cost, nonuniform grids with local fine grid spacing inside the boundary layer will frequently be used to resolve the variables there.

Another common method to avoid the oscillatory solution is to use a first-order upwind scheme,

$$R_{cell} (T_j - T_{j-1}) = (T_{j+1} - 2T_j + T_{j-1}), \quad (10.93)$$

where a forward-difference scheme is used to discretize the convective term. It is easy to see that this scheme reduces the heat convected to the boundary and thus prevents the oscillatory solution. However, the upwind scheme is not very accurate (only first-order accurate). It can be easily shown that the upwind scheme (10.93) does not recover the original transport equation (10.84). Instead it is consistent with a slightly different transport equation (when the cell Péclet number is kept finite during the process),

$$u \frac{\partial T}{\partial x} = D (1 + 0.5 R_{cell}) \frac{\partial^2 T}{\partial x^2}. \quad (10.94)$$

Thus, another way to view the effect of the first-order upwind scheme (10.93) is that it introduces a numerical diffusivity of the value of  $0.5 R_{cell} D$ , which enhances the conduction of heat through the boundary. For an accurate solution, one normally requires that  $0.5 R_{cell} \ll 1$ , which is very restrictive and does not offer any advantage over the centered difference scheme (10.91).

Higher order upwind schemes may be introduced to obtain more accurate nonoscillatory solutions without excessive grid refinement. However, those schemes may be less robust. Refer to [Fletcher \(1988\)](#); see Vol. I, Chapter 9) for discussions.

Similarly, there are upwind schemes for finite-element methods to solve convection-dominated problems. Most of those are based on the Petrov-Galerkin approach that permits an effective upwind treatment of the convective term along local streamlines ([Brooks & Hughes, 1982](#)). Since then, stabilized finite-element methods have been developed where a least-squares term is added to the momentum balance equation to provide the necessary stability for convection-dominated flows (see [Franca et al., 1992](#)).

## Incompressibility Condition

In solving the Navier-Stokes equations using the primitive variables (velocity and pressure), another numerical difficulty lies in the continuity equation: The continuity equation can be regarded either as a constraint on the flow field to determine the pressure, or the pressure plays the role of the Lagrange multiplier to satisfy the continuity equation.

In a flow field, the information (or disturbance) travels with both the flow and the speed of sound in the fluid. Since the speed of sound is infinite in an incompressible fluid, part of the information (pressure disturbance) is propagated instantaneously throughout the domain. In many numerical schemes the pressure is often obtained by solving a Poisson equation. The Poisson equation may occur in either continuous form or discrete form. Some of these schemes will be described here. In some of them, solving the pressure Poisson equation is the most costly step.

Another common technique to surmount the difficulty of the incompressible limit is to introduce an artificial compressibility ([Chorin, 1967](#)). This formulation is normally used for steady problems with a pseudotransient formulation. In the formulation, the continuity equation is replaced by

$$\frac{\partial p}{\partial t} + c^2 \nabla \cdot \mathbf{u} = 0, \quad (10.95)$$

where  $c$  is an arbitrary constant and could be the artificial speed of sound in a corresponding compressible fluid with the equation of state  $p = c^2 \rho$ . The formulation is called *pseudotransient* because (10.95) does not have any physical meaning before the steady state is reached. However, when  $c$  is large, (10.95) can be considered as an approximation to the unsteady solution of (10.80) in the incompressible Navier-Stokes problem.

### Explicit MacCormack Scheme

Instead of using the artificial compressibility in (10.95), one may start with the exact compressible Navier-Stokes equations. In Cartesian coordinates, the component form of the continuity equation and compressible Navier-Stokes equation (with  $\mu_v = 0$ ) in two dimensions can be explicitly written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} + \frac{\partial (\rho v)}{\partial y} = 0, \quad (10.96)$$

$$\frac{\partial}{\partial t}(\rho u) + \frac{\partial}{\partial x}(\rho u^2) + \frac{\partial}{\partial y}(\rho v u) = \rho g_x - \frac{\partial p}{\partial x} + \mu \nabla^2 u + \frac{\mu}{3} \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \quad (10.97)$$

$$\frac{\partial}{\partial t}(\rho v) + \frac{\partial}{\partial x}(\rho u v) + \frac{\partial}{\partial y}(\rho v^2) = \rho g_y - \frac{\partial p}{\partial y} + \mu \nabla^2 v + \frac{\mu}{3} \frac{\partial}{\partial y} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \quad (10.98)$$

with the equation of state,

$$p = c^2 \rho, \quad (10.99)$$

where  $c$  is the speed of sound in the medium. As long as the flows are limited to low Mach numbers and the conditions are almost isothermal, the solution to this set of equations should approximate the incompressible limit.

The explicit MacCormack scheme, after R. W. [MacCormack \(1969\)](#), is essentially a predictor-corrector scheme, similar to a second-order Runge-Kutta method commonly used to solve ordinary differential equations. For a system of equations of the form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial y} = 0, \quad (10.100)$$

the explicit MacCormack scheme consists of two steps:

$$\text{predictor } \mathbf{U}_{i,j}^* = \mathbf{U}_{i,j}^n - \frac{\Delta t}{\Delta x} (\mathbf{E}_{i+1,j}^n - \mathbf{E}_{i,j}^n) - \frac{\Delta t}{\Delta y} (\mathbf{F}_{i,j+1}^n - \mathbf{F}_{i,j}^n), \quad (10.101)$$

$$\text{corrector } \mathbf{U}_{i,j}^{n+1} = \frac{1}{2} \left[ \mathbf{U}_{i,j}^n + \mathbf{U}_{i,j}^* - \frac{\Delta t}{\Delta x} (\mathbf{E}_{i,j}^* - \mathbf{E}_{i-1,j}^*) - \frac{\Delta t}{\Delta y} (\mathbf{F}_{i,j}^* - \mathbf{F}_{i,j-1}^*) \right] \quad (10.102)$$

Notice that the spatial derivatives in (10.100) are discretized with opposite one-sided finite differences in the predictor and corrector stages. The star variables are all evaluated at time level  $t_{n+1}$ . This scheme is second-order accurate in both time and space.

Applying the MacCormack scheme to the compressible Navier-Stokes equations (10.96) through (10.98) and replacing the pressure with (10.99), we have the predictor step:

$$\rho_{i,j}^* = \rho_{i,j}^n - c_1 \left[ (\rho u)_{i+1,j}^n - (\rho u)_{i,j}^n \right] - c_2 \left[ (\rho v)_{i,j+1}^n - (\rho v)_{i,j}^n \right] \quad (10.103)$$

$$\begin{aligned} (\rho u)_{i,j}^* &= (\rho u)_{i,j}^n - c_1 \left[ (\rho u^2 + c^2 \rho)_{i+1,j}^n - (\rho u^2 + c^2 \rho)_{i,j}^n \right] \\ &\quad - c_2 \left[ (\rho uv)_{i,j+1}^n - (\rho uv)_{i,j}^n \right] + \frac{4}{3} c_3 \left( u_{i+1,j}^n - 2u_{i,j}^n + u_{i-1,j}^n \right) \\ &\quad + c_4 \left( u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n \right) \\ &\quad + c_5 \left( v_{i+1,j+1}^n + v_{i-1,j-1}^n - v_{i+1,j-1}^n - v_{i-1,j+1}^n \right) \end{aligned} \quad (10.104)$$

$$\begin{aligned} (\rho v)_{i,j}^* &= (\rho v)_{i,j}^n - c_1 \left[ (\rho uv)_{i+1,j}^n - (\rho uv)_{i,j}^n \right] \\ &\quad - c_2 \left[ (\rho v^2 + c^2 \rho)_{i,j+1}^n - (\rho v^2 + c^2 \rho)_{i,j}^n \right] \\ &\quad + c_3 \left( v_{i+1,j}^n - 2v_{i,j}^n + v_{i-1,j}^n \right) + \frac{4}{3} c_4 \left( v_{i,j+1}^n - 2v_{i,j}^n + v_{i,j-1}^n \right) \\ &\quad + c_5 \left( u_{i+1,j+1}^n + u_{i-1,j-1}^n - u_{i+1,j-1}^n - u_{i-1,j+1}^n \right). \end{aligned} \quad (10.105)$$

Similarly, the corrector step is given by:

$$2\rho_{i,j}^{n+1} = \rho_{i,j}^n + \rho_{i,j}^* - c_1 \left[ (\rho u)_{i,j}^* - (\rho u)_{i-1,j}^* \right] - c_2 \left[ (\rho v)_{i,j}^* - (\rho v)_{i,j-1}^* \right] \quad (10.106)$$

$$\begin{aligned} 2(\rho u)_{i,j}^{n+1} &= (\rho u)_{i,j}^n + (\rho u)_{i,j}^* - c_1 \left[ (\rho u^2 + c^2 \rho)_{i,j}^* - (\rho u^2 + c^2 \rho)_{i-1,j}^* \right] \\ &\quad - c_2 \left[ (\rho uv)_{i,j}^* - (\rho uv)_{i,j-1}^* \right] + \frac{4}{3} c_3 \left( u_{i+1,j}^* - 2u_{i,j}^* + u_{i-1,j}^* \right) \\ &\quad + c_4 \left( u_{i,j+1}^* - 2u_{i,j}^* + u_{i,j-1}^* \right) \\ &\quad + c_5 \left( v_{i+1,j+1}^* + v_{i-1,j-1}^* - v_{i+1,j-1}^* - v_{i-1,j+1}^* \right) \end{aligned} \quad (10.107)$$

$$\begin{aligned} 2(\rho v)_{i,j}^{n+1} &= (\rho v)_{i,j}^n + (\rho v)_{i,j}^* - c_1 \left[ (\rho uv)_{i,j}^* - (\rho uv)_{i-1,j}^* \right] \\ &\quad - c_2 \left[ (\rho v^2 + c^2 \rho)_{i,j}^* - (\rho v^2 + c^2 \rho)_{i,j-1}^* \right] \\ &\quad + c_3 \left( v_{i+1,j}^* - 2v_{i,j}^* + v_{i-1,j}^* \right) + \frac{4}{3} c_4 \left( v_{i,j+1}^* - 2v_{i,j}^* + v_{i,j-1}^* \right) \\ &\quad + c_5 \left( u_{i+1,j+1}^* + u_{i-1,j-1}^* - u_{i+1,j-1}^* - u_{i-1,j+1}^* \right) \end{aligned} \quad (10.108)$$

The coefficients are defined as

$$c_1 = \frac{\Delta t}{\Delta x}, \quad c_2 = \frac{\Delta t}{\Delta y}, \quad c_3 = \frac{\mu \Delta t}{(\Delta x)^2}, \quad c_4 = \frac{\mu \Delta t}{(\Delta y)^2}, \quad c_5 = \frac{\mu \Delta t}{12 \Delta x \Delta y}. \quad (10.109)$$

In both the predictor and corrector steps, the viscous terms (the second-order derivative terms) are all discretized with centered differences to maintain second-order accuracy. For brevity, body force terms in the momentum equations are neglected here.

During the predictor and corrector stages of the explicit MacCormack scheme (10.103) through (10.108), one-sided differences are arranged in the *FF* and *BB* fashion, respectively. Here, in the notation *FF*, the first *F* denotes the forward difference in the *x*-direction and the second *F* denotes the forward difference in the *y*-direction. Similarly, *BB* stands for backward differences in both *x* and *y* directions. We denote this arrangement as *FF/BB*. Similarly, one may get *BB/FF*, *FB/BF*, *BF/FB* arrangements. It is noted that some balanced cyclings of these arrangements generate better results than others.

Tannehill, Anderson, and Pletcher (1997) give the following semi-empirical stability criterion for the explicit MacCormack scheme:

$$\Delta t \leq \frac{\sigma}{(1 + 2/Re_\Delta)} \left[ \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + c \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \right]^{-1}, \quad (10.110)$$

where  $\sigma$  is a safety factor ( $\approx 0.9$ ), and  $Re_\Delta = \min(\rho|u|\Delta x/\mu, \rho|v|\Delta y/\mu)$  is the minimum mesh Reynolds number. This condition is quite conservative for flows with small-mesh Reynolds numbers.

One key issue for the explicit MacCormack scheme to work properly is the boundary conditions for density (thus pressure). We leave this issue to the next section where its implementation in two sample problems will be demonstrated.

## MAC Scheme

Most numerical schemes developed for computational fluid dynamics problems can be characterized as operator-splitting algorithms. The operator-splitting algorithms divide each time step into several substeps. Each substep solves one part of the operator and thus decouples the numerical difficulties associated with each part of the operator. For example, consider a system,

$$\frac{d\phi}{dt} + A(\phi) = f, \quad (10.111)$$

with initial condition  $\phi(0) = \phi_0$ , where the operator  $A$  may be split into two operators

$$A(\phi) = A_1(\phi) + A_2(\phi). \quad (10.112)$$

Using a simple first-order accurate Marchuk-Yanenko fractional step scheme (Marchuk, 1975; Yanenko, 1971), the solution of the system at each time step  $\phi^{n+1} = \phi((n+1)\Delta t)$  ( $n = 0, 1, \dots$ ) is approximated by solving the following two successive problems:

$$\frac{\phi^{n+1/2} - \phi^n}{\Delta t} + A_1(\phi^{n+1/2}) = f_1^{n+1}, \quad (10.113)$$

$$\frac{\phi^{n+1} - \phi^{n+1/2}}{\Delta t} + A_2(\phi^{n+1}) = f_2^{n+1}, \quad (10.114)$$

where  $\phi^0 = \phi_0$ ,  $\Delta t = t_{n+1} - t_n$ , and  $f_1^{n+1} + f_2^{n+1} = f^{n+1} = f((n+1)\Delta t)$ . The time discretizations in (10.113) and (10.114) are implicit. Some schemes to be discussed in what follows actually use explicit discretizations. However, the stability conditions for those explicit schemes must be satisfied.

The MAC (marker-and-cell) method was first proposed by Harlow and Welch (1965) to solve flow problems with free surfaces. There are many variations of this method. It basically uses a finite-difference discretization for the Navier-Stokes equations and splits the equations into two operators:

$$\mathbf{A}_1(\mathbf{u}, p) = \begin{pmatrix} (\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\text{Re}} \nabla^2 \mathbf{u} \\ 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{A}_2(\mathbf{u}, p) = \begin{pmatrix} \nabla p \\ \nabla \cdot \mathbf{u} \end{pmatrix}. \quad (10.115)$$

Each time step is divided into two substeps as discussed in the Marchuk-Yanenko fractional-step scheme (10.113) and (10.114). The first step solves a convection and diffusion problem, which is discretized explicitly:

$$\frac{\mathbf{u}^{n+1/2} - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{1}{\text{Re}} \nabla^2 \mathbf{u}^n = \mathbf{g}^{n+1}. \quad (10.116)$$

In the second step, the pressure gradient operator is added (implicitly) and, at the same time, the incompressible condition is enforced:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1/2}}{\Delta t} + \nabla p^{n+1} = \mathbf{0}, \quad (10.117)$$

and

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \quad (10.118)$$

This step is also called a *projection step* to satisfy the incompressibility condition.

Normally, the MAC scheme is presented in a discretized form. A preferred feature of the MAC method is the use of the staggered grid. An example of a staggered grid in two dimensions is shown in Figure 10.4. On this staggered grid, pressure variables are defined at the centers of the cells and velocity components are defined at the cell faces.

Using the staggered grid, two components of the transport equation (10.116) can be written as

$$u_{i+1/2, j}^{n+1/2} = u_{i+1/2, j}^n - \Delta t \left( u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \frac{1}{\text{Re}} \nabla^2 u \right)_{i+1/2, j}^n + \Delta t f_{i+1/2, j}^{n+1}, \quad (10.119)$$

$$v_{i, j+1/2}^{n+1/2} = v_{i, j+1/2}^n - \Delta t \left( u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \frac{1}{\text{Re}} \nabla^2 v \right)_{i, j+1/2}^n + \Delta t g_{i, j+1/2}^{n+1}, \quad (10.120)$$

where  $\mathbf{u} = (u, v)$ ,  $\mathbf{g} = (f, g)$ ,  $\left( u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \frac{1}{\text{Re}} \nabla^2 u \right)_{i+1/2, j}^n$ , and  $\left( u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \frac{1}{\text{Re}} \nabla^2 v \right)_{i, j+1/2}^n$  are the functions interpolated at the grid locations for the  $x$ -component of the velocity at  $(i+1/2, j)$  and for the  $y$ -component of the velocity at  $(i, j+1/2)$ , respectively, and at the previous time  $t = t_n$ . The discretized form of (10.117) is

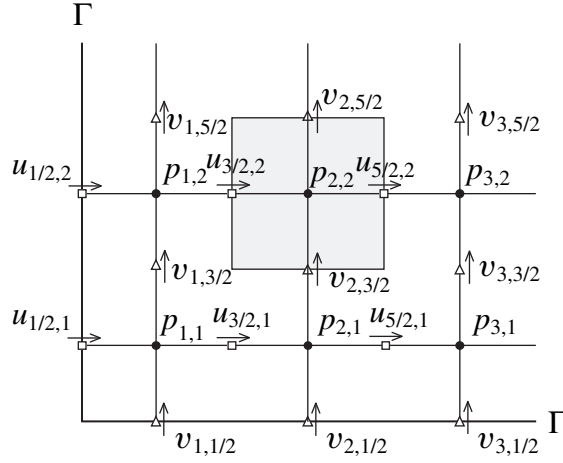


FIGURE 10.4 Staggered grid and a typical cell around  $p_{2,2}$ . Here velocity component values are computed halfway between the grid points for the pressure field.

$$u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^{n+1/2} - \frac{\Delta t}{\Delta x} (p_{i+1,j}^{n+1} - p_{i,j}^{n+1}), \quad (10.121)$$

$$v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^{n+1/2} - \frac{\Delta t}{\Delta y} (p_{i,j+1}^{n+1} - p_{i,j}^{n+1}), \quad (10.122)$$

where  $\Delta x = x_{i+1} - x_i$  and  $\Delta y = y_{j+1} - y_j$  are the uniform grid spacing in the  $x$  and  $y$  directions, respectively. The discretized continuity equation (10.118) can be written as:

$$\frac{u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}}{\Delta y} = 0. \quad (10.123)$$

Substitution of the two velocity components from (10.121) and (10.122) into the discretized continuity equation (10.123) generates a discrete Poisson equation for the pressure:

$$\begin{aligned} \nabla_d^2 p_{i,j}^{n+1} &\equiv \frac{1}{\Delta x^2} (p_{i+1,j}^{n+1} - 2p_{i,j}^{n+1} + p_{i-1,j}^{n+1}) + \frac{1}{\Delta y^2} (p_{i,j+1}^{n+1} - 2p_{i,j}^{n+1} + p_{i,j-1}^{n+1}) \\ &= \frac{1}{\Delta t} \left( \frac{u_{i+1/2,j}^{n+1/2} - u_{i-1/2,j}^{n+1/2}}{\Delta x} + \frac{v_{i,j+1/2}^{n+1/2} - v_{i,j-1/2}^{n+1/2}}{\Delta y} \right). \end{aligned} \quad (10.124)$$

The major advantage of the staggered grid is that it prevents the appearance of oscillatory solutions. On a normal grid, the pressure gradient would have to be approximated using two alternate grid points (not the adjacent ones) when a central difference scheme is used, that is,

$$\left( \frac{\partial p}{\partial x} \right)_{i,j} = \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} \quad \text{and} \quad \left( \frac{\partial p}{\partial y} \right)_{i,j} = \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y}. \quad (10.125)$$



Thus, a wavy pressure field (in a zigzag pattern) would be felt like a uniform one by the momentum equation. However, on a staggered grid, the pressure gradient is approximated by the difference of the pressures between two adjacent grid points. Consequently, a pressure field with a zigzag pattern would no longer be felt as a uniform pressure field and could not arise as a possible solution. It is also seen that the discretized continuity equation (10.123) contains the differences of the adjacent velocity components, which would prevent a wavy velocity field from satisfying the continuity equation.

Another advantage of the staggered grid is its accuracy. For example, the truncation error for (10.123) is  $O(\Delta x^2, \Delta y^2)$  even though only four grid points are involved. The pressure gradient evaluated at the cell faces,

$$\left(\frac{\partial p}{\partial x}\right)_{i+1/2, j} = \frac{p_{i+1, j} - p_{i, j}}{\Delta x}, \quad \text{and} \quad \left(\frac{\partial p}{\partial y}\right)_{i, j+1/2} = \frac{p_{i, j+1} - p_{i, j}}{\Delta y}, \quad (10.126)$$

are all second-order accurate.

On the staggered grid, the MAC method does not require boundary conditions for the pressure equation (10.124). Let us examine a pressure node next to the boundary, for example,  $p_{1,2}$  as shown in Figure 10.4. When the normal velocity is specified at the boundary,  $u_{1/2, 2}^{n+1}$  is known. In evaluating the discrete continuity equation (10.123) at the pressure node (1, 2), the velocity  $u_{1/2, 2}^{n+1}$  should not be expressed in terms of  $u_{1/2, 2}^{n+1/2}$  using (10.121). Therefore  $p_{0,2}$  will not appear in equation (10.120), and no boundary condition for the pressure is needed. It should also be noted that (10.119) and (10.120) only update the velocity components for the interior grid points, and their values at the boundary grid points are not needed in the MAC scheme. Peyret and Taylor (1983) also noticed that the numerical solution in the MAC method is independent of the boundary values of  $u^{n+1/2}$  and  $v^{n+1/2}$ , and a zero normal pressure gradient on the boundary would give satisfactory results. However, their explanation was more cumbersome.

In summary, for each time step in the MAC scheme, the intermediate velocity components,  $u_{i+1/2, j}^{n+1/2}$  and  $v_{i, j+1/2}^{n+1/2}$ , in the interior of the domain are first evaluated using (10.119) and (10.120), respectively. Next, the discrete pressure Poisson equation (10.124) is solved. Finally, the velocity components at the new time step are obtained from (10.121) and (10.122). In the MAC scheme, the most costly step is the solution of the Poisson equation for the pressure (10.124).

Chorin (1968) and Temam (1969) independently presented a numerical scheme for the incompressible Navier-Stokes equations, termed the *projection method*. The projection method was initially proposed using the standard grid. However, when it is applied in an explicit fashion on the MAC staggered grid, it is identical to the MAC method as long as the boundary conditions are not considered, as shown in Peyret and Taylor (1983).

A physical interpretation of the MAC scheme or the projection method is that the explicit update of the velocity field does not generate a divergence-free velocity field in the first step. Thus an irrotational correction field, in the form of a velocity potential which is proportional to the pressure, is added to the nondivergence-free velocity field in the second step in order to enforce the incompressibility condition.

As the MAC method uses an explicit scheme in the convection-diffusion step, the stability conditions for this method are (Peyret & Taylor, 1983),

$$\frac{1}{2}(u^2 + v^2) \Delta t \text{ Re} \leq 1, \quad (10.127)$$

and

$$\frac{4\Delta t}{\text{Re}\Delta x^2} \leq 1, \quad (10.128)$$

when  $\Delta x = \Delta y$ . The stability conditions (10.127) and (10.128) are quite restrictive on the size of the time step. These restrictions can be removed by using implicit schemes for the convection-diffusion step.

### Θ-Scheme

The MAC algorithm described in the preceding section is only first-order accurate in time. In order to have a second-order accurate scheme for the Navier-Stokes equations, the Θ-scheme of Glowinski (1991) may be used. The Θ-scheme splits each time step symmetrically into three substeps, which are described here.

- Step 1:

$$\frac{\mathbf{u}^{n+\theta} - \mathbf{u}^n}{\theta \Delta t} - \frac{\alpha}{\text{Re}} \nabla^2 \mathbf{u}^{n+\theta} + \nabla p^{n+\theta} = \mathbf{g}^{n+\theta} + \frac{\beta}{\text{Re}} \nabla^2 \mathbf{u}^n - (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n, \quad (10.129)$$

$$\nabla \cdot \mathbf{u}^{n+\theta} = 0. \quad (10.130)$$

- Step 2:

$$\frac{\mathbf{u}^{n+1-\theta} - \mathbf{u}^{n+\theta}}{(1-2\theta) \Delta t} - \frac{\beta}{\text{Re}} \nabla^2 \mathbf{u}^{n+1-\theta} + (\mathbf{u}^* \cdot \nabla) \mathbf{u}^{n+1-\theta} = \mathbf{g}^{n+1-\theta} + \frac{\alpha}{\text{Re}} \nabla^2 \mathbf{u}^{n+\theta} - \nabla p^{n+\theta}. \quad (10.131)$$

- Step 3:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+1-\theta}}{\theta \Delta t} - \frac{\alpha}{\text{Re}} \nabla^2 \mathbf{u}^{n+1} + \nabla p^{n+1} = \mathbf{g}^{n+1} + \frac{\beta}{\text{Re}} \nabla^2 \mathbf{u}^{n+1-\theta} - (\mathbf{u}^{n+1-\theta} \cdot \nabla) \mathbf{u}^{n+1-\theta}, \quad (10.132)$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \quad (10.133)$$

It was shown that when  $\theta = 1 - 1/\sqrt{2} = 0.29289\dots$ ,  $\alpha + \beta = 1$ , and  $\beta = \theta/(1 - \theta)$ , the scheme is second-order accurate. The first and third steps of the Θ-scheme are identical and are the Stokes flow problems. The second step, (10.131), represents a nonlinear convection-diffusion problem if  $\mathbf{u}^* = \mathbf{u}^{n+1-\theta}$ . However, it was concluded that there is practically no loss in accuracy and stability if  $\mathbf{u}^* = \mathbf{u}^{n+\theta}$  is used. Numerical techniques for solving these substeps are discussed in Glowinski (1991).

## Mixed Finite-Element Formulation

The weak formulation described in Section 10.3 can be directly applied to the Navier-Stokes equations (10.80) and (10.81), and it gives

$$\int_{\Omega} \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{g} \right) \cdot \tilde{\mathbf{u}} d\Omega + \frac{2}{\text{Re}} \int_{\Omega} \mathbf{D}[\mathbf{u}] : \mathbf{D}[\tilde{\mathbf{u}}] d\Omega - \int_{\Omega} p (\nabla \cdot \tilde{\mathbf{u}}) d\Omega = 0, \quad (10.134)$$

$$\int_{\Omega} \tilde{p} \nabla \cdot \mathbf{u} d\Omega = 0, \quad (10.135)$$

where  $\tilde{\mathbf{u}}$  and  $\tilde{p}$  are the variations of the velocity and pressure, respectively. The rate of strain tensor is given by

$$\mathbf{D}[\mathbf{u}] = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T]. \quad (10.136)$$

The Galerkin finite-element formulation for the problem is identical to (10.134) and (10.135), except that all the functions are chosen from finite-dimensional subspaces and represented in the form of basis or interpolation functions.

The main difficulty with this finite-element formulation is the choice of the interpolation functions (or the types of the elements) for velocity and pressure. The finite-element approximations that use the same interpolation functions for velocity and pressure suffer from a highly oscillatory pressure field. As described in the previous section, a similar behavior in the finite-difference scheme is prevented by introducing the staggered grid. There are a number of options to overcome this problem with spurious pressure. One of them is the mixed finite-element formulation that uses different interpolation functions (or finite elements) for velocity and pressure. The requirement for the mixed finite-element approach is related to the so-called Babuska-Brezzi (or LBB) stability condition, or *inf-sup* condition. The detailed discussions for this condition can be found in Oden and Carey (1984). A common practice in the mixed finite-element formulation is to use a pressure interpolation function that is one order lower than a velocity interpolation function. As an example in two dimensions, a triangular element is shown in Figure 10.5a. On this mixed element, quadratic interpolation functions are used for the velocity components and are defined on all six nodes,

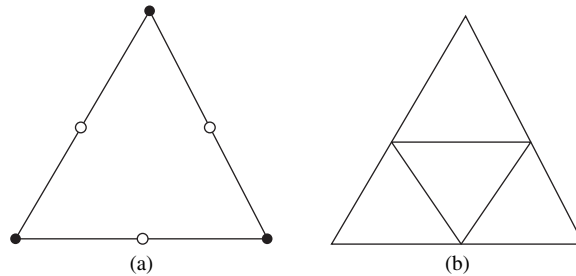


FIGURE 10.5 Mixed finite elements. (a) Velocities are determined at all six points while pressure is only determined at the triangle's vertices (filled circles). (b) The velocity grid involves all four triangles while the pressure only involves the one larger composite triangle.

while linear interpolation functions are used for the pressure and are defined on three vertices only. A slightly different approach is to use a pressure grid that is twice coarser than the velocity one, and then use the same interpolation functions on both grids (Glowinski, 1991). For example, a piecewise-linear pressure is defined on the outside (coarser) triangle, while a piecewise-linear velocity is defined on all four subtriangles, as shown in Figure 10.5b.

Another option to prevent a spurious pressure field is to use the stabilized finite-element formulation while keeping the equal order interpolations for velocity and pressure. A general formulation in this approach is the Galerkin/least-squares (GLS) stabilization (Tezduyar, 1992). In the GLS stabilization, the stabilizing terms are obtained by minimizing the squared residual of the momentum equation integrated over each element domain. The choice of the stabilization parameter is discussed in Franca et al. (1992) and Franca and Frey (1992).

Comparing the mixed and the stabilized finite-element formulations, the mixed finite-element method is parameter free, as pointed out in Glowinski (1991). There is no need to adjust the stabilization parameters, which could be a delicate problem. More importantly, for a given flow problem the desired finite-element mesh size is generally determined based on the velocity behavior (e.g., it is defined by the boundary or shear layer thickness). Therefore, equal order interpolation will be more costly from the pressure point of view but without further gains in accuracy. However, the GLS-stabilized finite-element formulation has the additional benefit of preventing oscillatory solutions produced in the Galerkin finite-element method due to the large convective term in high Reynolds number flows.

Once the interpolation functions for the velocity and pressure in the mixed finite-element approximations are determined, the matrix form of equations (10.134) and (10.135) can be written as

$$\begin{pmatrix} \mathbf{M}\dot{\mathbf{u}} \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_u \\ \mathbf{f}_p \end{pmatrix}, \quad (10.137)$$

where  $\mathbf{u}$  and  $\mathbf{p}$  are the vectors containing all unknown values of the velocity components and pressure defined on the finite-element mesh, respectively;  $\dot{\mathbf{u}}$  is the first-time derivative of  $\mathbf{u}$ ; and  $\mathbf{M}$  is the mass matrix corresponding to the time derivative term in equation (10.134). Matrix  $\mathbf{A}$  depends on the value of  $\mathbf{u}$  due to the nonlinear convective term in the momentum equation. The symmetry in the pressure terms in (10.134) and (10.135) results in the symmetric arrangement of  $\mathbf{B}$  and  $\mathbf{B}^T$  in the algebraic system (10.137). Vectors  $\mathbf{f}_u$  and  $\mathbf{f}_p$  come from the body-force term in the momentum equation and from the application of the boundary conditions.

The ordinary differential equation (10.137) can be further discretized in time with finite-difference methods. The resulting nonlinear system of equations is typically solved iteratively using Newton's method. At each stage of the nonlinear iteration, the sparse linear algebraic equations are normally solved either by using a direct solver such as the Gauss elimination procedure for small system sizes or by using an iterative solver such as the generalized minimum residual method (GMRES) for large systems. Other iterative solution methods for sparse nonsymmetric systems can be found in Saad (1996). An application of the mixed finite-element method is discussed as one of the examples in the next section.

## 10.5. THREE EXAMPLES

In this section, we will solve three sample problems. The first one is the classic driven-cavity flow problem. The second is flow around a square block confined between two parallel plates. These two problems will be solved by using the explicit MacCormack scheme, with details in Perrin and Hu (2006). The contribution by Andrew Perrin in preparing results for these two problems is greatly appreciated. The last problem is flow around a circular cylinder confined between two parallel plates. It will be solved by using a mixed finite-element formulation.

### Explicit MacCormack Scheme for Driven-Cavity Flow Problem

The driven-cavity flow problem, in which a fluid-filled square box (or cavity) is swirled by a uniformly translating lid, as shown in Figure 10.6, is a classic problem in CFD. This problem is unambiguous with easily applied boundary conditions and has a wealth of documented analytical and computational results, for example Ghia et al. (1982). We will solve this flow using the explicit MacCormack scheme discussed in the previous section.

We may nondimensionalize the problem with the following scaling: length with  $D$ , velocity with  $U$ , time with  $D/U$ , density with a reference density  $\rho_0$ , and pressure with  $\rho_0 U^2$ . Using this scaling, the equation of state (10.99) becomes  $p = \rho/M^2$ , where  $M = U/c$  is the Mach number. The Reynolds number is defined as  $Re = \rho_0 U D / \mu$ .

The boundary conditions for this problem are relatively simple. The velocity components on all four sides of the cavity are well defined. There are two singularities of velocity gradient at the two top corners where velocity  $u$  changes from 0 to  $U$  and from  $U$  to 0 directly underneath the sliding lid. However, these singularities will be smoothed out on a given grid, since the change of the velocity occurs linearly between two grid points. The boundary conditions for the density (hence the pressure) are more involved. Since the density is not specified on a solid surface, we need to generate an update scheme for values of density on all boundary points. A natural option is to derive it using the continuity equation.

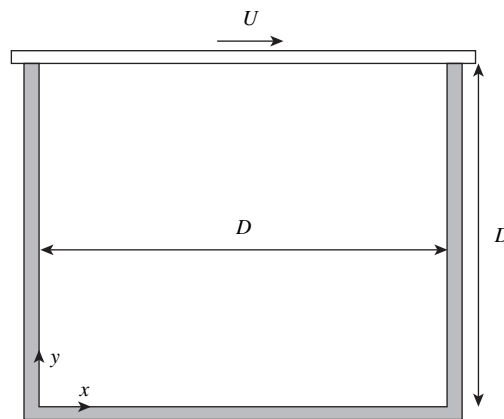


FIGURE 10.6 Driven-cavity flow problem. The cavity is filled with a fluid with the top lid sliding at a constant velocity  $U$ . The resulting flow field involves a large main vortex that is not symmetrical or centered in the box.

Consider the boundary on the left (at  $x = 0$ ). Since  $v = 0$  along the surface, the continuity equation (10.96) reduces to

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0. \quad (10.138)$$

We may use a predictor-corrector scheme to update density on this surface with a one-sided, second-order accurate discretization for the spatial derivative:

$$\begin{aligned} \left( \frac{\partial f}{\partial x} \right)_i &= \frac{1}{2\Delta x} (-f_{i+2} + 4f_{i+1} - 3f_i) + O(\Delta x^2) \\ \text{or } \left( \frac{\partial f}{\partial x} \right)_i &= \frac{-1}{2\Delta x} (-f_{i-2} + 4f_{i-1} - 3f_i) + O(\Delta x^2). \end{aligned}$$

Therefore, on the surface of  $x = 0$  (for  $i = 0$ , including two corner points on the left), we have the following update scheme for density:

$$\text{predictor } \rho_{i,j}^* = \rho_{i,j}^n - \frac{\Delta t}{2\Delta x} [-(\rho u)_{i+2,j}^n + 4(\rho u)_{i+1,j}^n - 3(\rho u)_{i,j}^n], \quad (10.139)$$

$$\text{corrector } 2\rho_{i,j}^{n+1} = \rho_{i,j}^n + \rho_{i,j}^* - \frac{\Delta t}{2\Delta x} [-(\rho u)_{i+2,j}^* + 4(\rho u)_{i+1,j}^* - 3(\rho u)_{i,j}^*]. \quad (10.140)$$

Similarly, on the right side of the cavity  $x = D$  (for  $i = n_x - 1$ , where  $n_x$  is the number of grid points in the  $x$ -direction, including two corner points on the right), we have:

$$\text{predictor } \rho_{i,j}^* = \rho_{i,j}^n + \frac{\Delta t}{2\Delta x} [-(\rho u)_{i-2,j}^n + 4(\rho u)_{i-1,j}^n - 3(\rho u)_{i,j}^n], \quad (10.141)$$

$$\text{corrector } 2\rho_{i,j}^{n+1} = \rho_{i,j}^n + \rho_{i,j}^* + \frac{\Delta t}{2\Delta x} [-(\rho u)_{i-2,j}^* + 4(\rho u)_{i-1,j}^* - 3(\rho u)_{i,j}^*]. \quad (10.142)$$

On the bottom of the cavity  $y = 0$  ( $j = 0$ ),

$$\text{predictor } \rho_{i,j}^* = \rho_{i,j}^n - \frac{\Delta t}{2\Delta y} [-(\rho v)_{i,j+2}^n + 4(\rho v)_{i,j+1}^n - 3(\rho v)_{i,j}^n], \quad (10.143)$$

$$\text{corrector } 2\rho_{i,j}^{n+1} = \rho_{i,j}^n + \rho_{i,j}^* - \frac{\Delta t}{2\Delta y} [-(\rho v)_{i,j+2}^* + 4(\rho v)_{i,j+1}^* - 3(\rho v)_{i,j}^*]. \quad (10.144)$$

Finally, on the top of the cavity  $y = D$  ( $j = n_y - 1$ ) where  $n_y$  is the number of grid points in the  $y$ -direction), the density needs to be updated from slightly different expressions since  $\partial \rho u / \partial x = U \partial \rho / \partial x$  is not zero there:

$$\text{predictor } \rho_{i,j}^* = \rho_{i,j}^n - \frac{\Delta t U}{2\Delta x} [\rho_{i+1,j}^n - \rho_{i-1,j}^n] + \frac{\Delta t}{2\Delta y} [-(\rho v)_{i,j-2}^n + 4(\rho v)_{i,j-1}^n - 3(\rho v)_{i,j}^n], \quad (10.145)$$

$$\begin{aligned} \text{corrector } 2\rho_{i,j}^{n+1} &= \rho_{i,j}^n + \rho_{i,j}^* - \frac{\Delta t U}{2\Delta x} [\rho_{i+1,j}^* - \rho_{i-1,j}^*] \\ &\quad + \frac{\Delta t}{2\Delta y} [-(\rho v)_{i,j-2}^* + 4(\rho v)_{i,j-1}^* - 3(\rho v)_{i,j}^*]. \end{aligned} \quad (10.146)$$

In summary, we may organize the explicit MacCormack scheme at each time step (10.103) to (10.108) into the following six substeps.

Step 1: For  $0 \leq i < n_x$  and  $0 \leq j < n_y$  (all nodes):

$$u_{i,j} = (\rho u)_{i,j}^n / \rho_{i,j}^n, \quad v_{i,j} = (\rho v)_{i,j}^n / \rho_{i,j}^n.$$

Step 2: For  $1 \leq i < n_x - 1$  and  $1 \leq j < n_y - 1$  (all interior nodes):

$$\begin{aligned} \rho_{i,j}^* &= \rho_{i,j}^n - a_1 \left[ (\rho u)_{i+1,j}^n - (\rho u)_{i,j}^n \right] - a_2 \left[ (\rho v)_{i,j+1}^n - (\rho v)_{i,j}^n \right], \\ (\rho u)_{i,j}^* &= (\rho u)_{i,j}^n - a_3 \left( \rho_{i+1,j}^n - \rho_{i,j}^n \right) - a_1 \left[ (\rho u^2)_{i+1,j}^n - (\rho u^2)_{i,j}^n \right] \\ &\quad - a_2 \left[ (\rho uv)_{i,j+1}^n - (\rho uv)_{i,j}^n \right] - a_{10} u_{i,j} + a_5 (u_{i+1,j} + u_{i-1,j}) \\ &\quad + a_6 (u_{i,j+1} + u_{i,j-1}) + a_9 (v_{i+1,j+1} + v_{i-1,j-1} - v_{i+1,j-1} - v_{i-1,j+1}), \\ (\rho v)_{i,j}^* &= (\rho v)_{i,j}^n - a_4 \left( \rho_{i,j+1}^n - \rho_{i,j}^n \right) - a_1 \left[ (\rho uv)_{i+1,j}^n - (\rho uv)_{i,j}^n \right] \\ &\quad - a_2 \left[ (\rho v^2)_{i,j+1}^n - (\rho v^2)_{i,j}^n \right] - a_{11} v_{i,j} + a_7 (v_{i+1,j} + v_{i-1,j}) \\ &\quad + a_8 (v_{i,j+1} + v_{i,j-1}) + a_9 (u_{i+1,j+1} + u_{i-1,j-1} - u_{i+1,j-1} - u_{i-1,j+1}). \end{aligned}$$

Step 3: Impose boundary conditions (at time  $t_{n+1}$ ) for  $\rho_{i,j}^*$ ,  $(\rho u)_{i,j}^*$  and  $(\rho v)_{i,j}^*$ .

Step 4: For  $0 \leq i < n_x$  and  $0 \leq j < n_y$  (all nodes):

$$u_{i,j}^* = (\rho u)_{i,j}^* / \rho_{i,j}^*, \quad v_{i,j}^* = (\rho v)_{i,j}^* / \rho_{i,j}^*.$$

Step 5: For  $1 \leq i < n_x - 1$  and  $1 \leq j < n_y - 1$  (all interior nodes):

$$\begin{aligned} 2\rho_{i,j}^{n+1} &= \left( \rho_{i,j}^n + \rho_{i,j}^* \right) - a_1 \left[ (\rho u)_{i,j}^* - (\rho u)_{i-1,j}^* \right] - a_2 \left[ (\rho v)_{i,j}^* - (\rho v)_{i,j-1}^* \right], \\ 2(\rho u)_{i,j}^{n+1} &= (\rho u)_{i,j}^n + (\rho u)_{i,j}^* - a_3 \left( \rho_{i,j}^* - \rho_{i-1,j}^* \right) - a_1 \left[ (\rho u^2)_{i,j}^* - (\rho u^2)_{i-1,j}^* \right] \\ &\quad - a_2 \left[ (\rho uv)_{i,j}^* - (\rho uv)_{i,j-1}^* \right] - a_{10} u_{i,j}^* + a_5 \left( u_{i+1,j}^* + u_{i-1,j}^* \right) \\ &\quad + a_6 \left( u_{i,j+1}^* + u_{i,j-1}^* \right) + a_9 \left( v_{i+1,j+1}^* + v_{i-1,j-1}^* - v_{i+1,j-1}^* - v_{i-1,j+1}^* \right), \\ 2(\rho v)_{i,j}^{n+1} &= (\rho v)_{i,j}^n + (\rho v)_{i,j}^* - a_4 \left( \rho_{i,j}^* - \rho_{i,j-1}^* \right) - a_1 \left[ (\rho uv)_{i,j}^* - (\rho uv)_{i-1,j}^* \right] \\ &\quad - a_2 \left[ (\rho v^2)_{i,j}^* - (\rho v^2)_{i,j-1}^* \right] - a_{11} v_{i,j}^* + a_7 \left( v_{i+1,j}^* + v_{i-1,j}^* \right) \\ &\quad + a_8 \left( v_{i,j+1}^* + v_{i,j-1}^* \right) + a_9 \left( u_{i+1,j+1}^* + u_{i-1,j-1}^* - u_{i+1,j-1}^* - u_{i-1,j+1}^* \right). \end{aligned}$$

Step 6: Impose boundary conditions for  $\rho_{i,j}^{n+1}$ ,  $(\rho u)_{i,j}^{n+1}$ , and  $(\rho v)_{i,j}^{n+1}$ . The coefficients are defined as:

$$\begin{aligned} a_1 &= \frac{\Delta t}{\Delta x}, & a_2 &= \frac{\Delta t}{\Delta y}, & a_3 &= \frac{\Delta t}{\Delta x M^2}, & a_4 &= \frac{\Delta t}{\Delta y M^2}, & a_5 &= \frac{4\Delta t}{3\text{Re}(\Delta x)^2}, \\ a_6 &= \frac{\Delta t}{\text{Re}(\Delta y)^2}, & a_7 &= \frac{\Delta t}{\text{Re}(\Delta x)^2}, & a_8 &= \frac{4\Delta t}{3\text{Re}(\Delta y)^2}, & a_9 &= \frac{\Delta t}{12\text{Re}\Delta x\Delta y}, \\ a_{10} &= 2(a_5 + a_6), & a_{11} &= 2(a_7 + a_8). \end{aligned}$$

For coding purposes, the variables  $u_{i,j}$  ( $v_{i,j}$ ) and  $u_{i,j}^*$  ( $v_{i,j}^*$ ) can take the same storage space. At the end of each time step, the starting values of  $\rho_{i,j}^n$ ,  $(\rho u)_{i,j}^n$ , and  $(\rho v)_{i,j}^n$  will be replaced with the corresponding new values of  $\rho_{i,j}^{n+1}$ ,  $(\rho u)_{i,j}^{n+1}$ , and  $(\rho v)_{i,j}^{n+1}$ .

Next we present some of the results and compare them with those in the paper by Hou et al. (1995) obtained by a lattice Boltzmann method. To keep the flow almost incompressible, the Mach number is chosen as  $M = 0.1$ . Flows with two Reynolds numbers,  $\text{Re} = \rho_0 UD / \mu = 100$  and 400 are simulated. At these Reynolds numbers, the flow will eventually be steady. Thus calculations need to be run long enough to get to the steady state. A uniform grid of 256 by 256 was used for this example.

Figure 10.7 shows comparisons of the velocity field calculated by the explicit MacCormack scheme with the streamlines from Hou et al. (1995) at  $\text{Re} = 100$  and 400. The agreement seems reasonable. It was also observed that the location of the center of the primary eddy agrees even better. When  $\text{Re} = 100$ , the center of the primary eddy is found at  $(0.62 \pm 0.02, 0.74 \pm 0.02)$  from the MacCormack scheme in comparison with  $(0.6196, 0.7373)$  from Hou. When  $\text{Re} = 400$ , the center of the primary eddy is found at  $(0.57 \pm 0.02, 0.61 \pm 0.02)$  from the MacCormack scheme in comparison with  $(0.5608, 0.6078)$  from Hou.

For a more quantitative comparison, Figure 10.8 plots the velocity profile along a vertical line cut through the center of the cavity ( $x = 0.5D$ ). The velocity profiles for two Reynolds numbers,  $\text{Re} = 100$  and 400, are compared. The results from the explicit MacCormack scheme

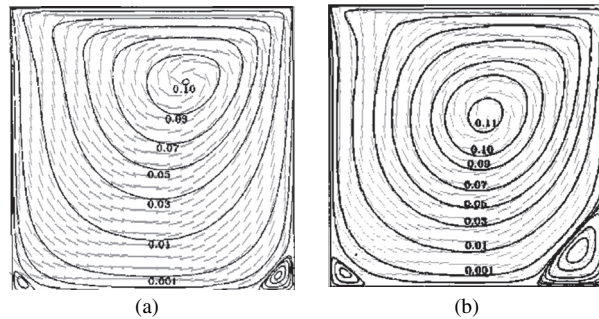
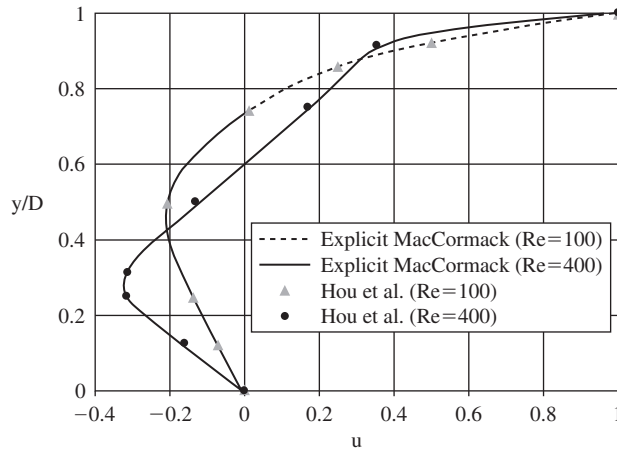


FIGURE 10.7 Comparisons of lid-driven square cavity results from the explicit MacCormack scheme (light gray, velocity vector field) and those from Hou et al. (1995) (dark solid streamlines) calculated using a lattice Boltzmann method. (a)  $\text{Re} = 100$ , (b)  $\text{Re} = 400$ .





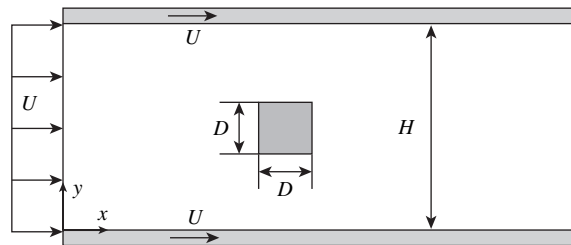
**FIGURE 10.8** Comparison of velocity profiles along a line cut through the center of the cavity ( $x = 0.5D$ ) at  $Re = 100$  and  $400$ . There is agreement between the results at the same Reynolds number from different computational schemes.

are shown in solid and dashed lines. The data points in symbols were directly converted from Hou's paper. The agreement is excellent.

### Explicit MacCormack Scheme for Flow Over a Square Block

For the second example, we consider flow around a square block confined between two parallel plates. Fluid comes in from the left with a uniform velocity profile  $U$ , and the plates are sliding with the same velocity, as indicated in Figure 10.9. This flow corresponds to the block moving left with velocity  $U$  along the channel's centerline. In the calculation we set the channel width  $H = 3D$  and the channel length  $L = 35D$  with  $15D$  ahead of the block and  $19D$  behind. The Mach number is set at  $M = 0.05$  to approximate the incompressible limit.

The velocity boundary conditions in this problem are specified as shown in Figure 10.9, except that at the outflow section, conditions  $\partial \rho u / \partial x = 0$  and  $\partial \rho v / \partial x = 0$  are used. The density (or pressure) boundary conditions are much more complicated, especially on the block surface. On all four sides of the outer boundary (top and bottom plates, inflow and



**FIGURE 10.9** Computational domain for two-dimensional flow past a square obstruction (block) between two parallel plates.

outflow), the continuity equation is used to update density as in the previous example. However, on the block surface, it was found that the conditions derived from the momentum equations give better results. Let us consider the front section of the block, and evaluate the  $x$ -component of the momentum equation (10.97) with  $u = v = 0$ :

$$\begin{aligned}\frac{\partial \rho}{\partial x} &= M^2 \left[ \frac{1}{\text{Re}} \left( \frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{1}{3} \frac{\partial^2 v}{\partial x \partial y} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial}{\partial x}(\rho u^2) - \frac{\partial}{\partial y}(\rho v u) - \frac{\partial}{\partial t}(\rho u) \right]_{\text{front surface}} \\ &= \frac{M^2}{\text{Re}} \left( \frac{4}{3} \frac{\partial^2 u}{\partial x^2} + \frac{1}{3} \frac{\partial^2 v}{\partial x \partial y} \right).\end{aligned}\quad (10.147)$$

In (10.147), the variables are nondimensionalized with the same scaling as the driven-cavity flow problem except that the block size  $D$  is used for length. Furthermore, the density gradient may be approximated with a second-order backward finite-difference scheme:

$$\left( \frac{\partial \rho}{\partial x} \right)_{i,j} = \frac{-1}{2\Delta x} (-\rho_{i-2,j} + 4\rho_{i-1,j} - 3\rho_{i,j}) + O(\Delta x^2). \quad (10.148)$$

The second-order derivatives for the velocities are expressed as

$$\left( \frac{\partial^2 u}{\partial x^2} \right)_{i,j} = \frac{1}{\Delta x^2} (2u_{i,j} - 5u_{i-1,j} + 4u_{i-2,j} - u_{i-3,j}) + O(\Delta x^2), \quad (10.149)$$

and

$$\begin{aligned}\left( \frac{\partial^2 v}{\partial x \partial y} \right)_{i,j} &= \frac{-1}{4\Delta x \Delta y} \left[ -(v_{i-2,j+1} - v_{i-2,j-1}) + 4(v_{i-1,j+1} - v_{i-1,j-1}) - 3(v_{i,j+1} - v_{i,j-1}) \right] \\ &\quad + O(\Delta x^2, \Delta x \Delta y, \Delta y^2).\end{aligned}\quad (10.150)$$

Substituting (10.148) through (10.150) into (10.147), we have an expression for density at the front of the block:

$$\begin{aligned}\rho_{i,j}|_{\text{front}} &= \frac{1}{3} (4\rho_{i-1,j} - \rho_{i-2,j}) + \frac{8}{9\Delta x} \frac{M^2}{\text{Re}} (-5u_{i-1,j} + 4u_{i-2,j} - u_{i-3,j}) \\ &\quad - \frac{1}{18\Delta y} \frac{M^2}{\text{Re}} \left[ -(v_{i-2,j+1} - v_{i-2,j-1}) + 4(v_{i-1,j+1} - v_{i-1,j-1}) - 3(v_{i,j+1} - v_{i,j-1}) \right].\end{aligned}\quad (10.151)$$

Similarly, at the back of the block:

$$\begin{aligned}\rho_{i,j}|_{\text{back}} &= \frac{1}{3} (4\rho_{i+1,j} - \rho_{i+2,j}) - \frac{8}{9\Delta x} \frac{M^2}{\text{Re}} (-5u_{i+1,j} + 4u_{i+2,j} - u_{i+3,j}) \\ &\quad - \frac{1}{18\Delta y} \frac{M^2}{\text{Re}} \left[ -(v_{i+2,j+1} - v_{i+2,j-1}) + 4(v_{i+1,j+1} - v_{i+1,j-1}) - 3(v_{i,j+1} - v_{i,j-1}) \right].\end{aligned}\quad (10.152)$$

At the top of the block, the  $y$ -component of the momentum equation should be used, and it is easy to find:

$$\begin{aligned} \rho_{i,j}|_{top} = & \frac{1}{3}(4\rho_{i,j+1} - \rho_{i,j+2}) - \frac{8}{9\Delta y} \frac{M^2}{Re}(-5v_{i,j+1} + 4v_{i,j+2} - v_{i,j+3}) \\ & - \frac{1}{18\Delta x} \frac{M^2}{Re} \left[ -(u_{i+1,j+2} - u_{i-1,j+2}) + 4(u_{i+1,j+1} - u_{i-1,j+1}) - 3(u_{i+1,j} - u_{i-1,j}) \right], \end{aligned} \quad (10.153)$$

and, finally, at the bottom of the block:

$$\begin{aligned} \rho_{i,j}|_{bottom} = & \frac{1}{3}(4\rho_{i,j-1} - \rho_{i,j-2}) + \frac{8}{9\Delta y} \frac{M^2}{Re}(-5v_{i,j-1} + 4v_{i,j-2} - v_{i,j-3}) \\ & - \frac{1}{18\Delta x} \frac{M^2}{Re} \left[ -(u_{i+1,j-2} - u_{i-1,j-2}) + 4(u_{i+1,j-1} - u_{i-1,j-1}) - 3(u_{i+1,j} - u_{i-1,j}) \right]. \end{aligned} \quad (10.154)$$

At the four corners of the block, the average values from the two corresponding sides may be used.

In computation, double precision numbers should be used: Otherwise cumulative round-off error may corrupt the simulation, especially for long runs. It is also helpful to introduce a new variable for density,  $\rho' = \rho - 1$ , such that only the density variation is computed. For this example, we may extend the *FF/BB* form of the explicit MacCormack scheme to have an *FB/BF* arrangement for one time step and a *BF/FB* arrangement for the subsequent time step. This cycling seems to generate better results.

We first plot the drag coefficient,  $C_D$  (4.107) and the lift coefficient,  $C_L$  (4.108) as functions of time for flows at two Reynolds numbers,  $Re = 20$  and  $100$ , in [Figure 10.10](#). For  $Re = 20$ , after the initial messy transient (corresponding to sound waves bouncing around the block and reflecting at the outflow) the flow eventually settles into a steady state. The drag coefficient stabilizes at a constant value around  $C_D = 6.94$  (obtained on a grid of  $701 \times 61$ ). Calculation on a finer grid ( $1401 \times 121$ ) yields  $C_D = 7.003$ . This is in excellent agreement with the value of  $C_D = 7.005$  obtained from an implicit finite-element calculation for incompressible flows (similar to the one used in the next example in this section) on a similar mesh to  $1401 \times 121$ . There is a small lift ( $C_L = 0.014$ ) due to asymmetries in the numerical scheme. The lift reduces to  $C_L = 0.003$  on the finer grid of  $1401 \times 121$ . For  $Re = 100$ , periodic vortex shedding occurs. Drag and lift coefficients are shown in [Figure 10.10b](#). The mean value of the drag coefficient and the amplitude of the lift coefficient are  $C_D = 3.35$  and  $C_L = 0.77$ , respectively. The finite-element results are  $C_D = 3.32$  and  $C_L = 0.72$  under similar conditions.

The flow field around the block at  $Re = 20$  is shown in [Figure 10.11](#). A steady wake is attached behind the block, and the circulation within the wake is clearly visible. [Figure 10.12](#) displays a sequence of the flow field around the block during one cycle of vortex shedding at  $Re = 100$ .

[Figure 10.13](#) shows the convergence of the drag coefficient as the grid spacing is reduced. Tests for two Reynolds numbers,  $Re = 20$  and  $100$ , are plotted. It seems that the solution with

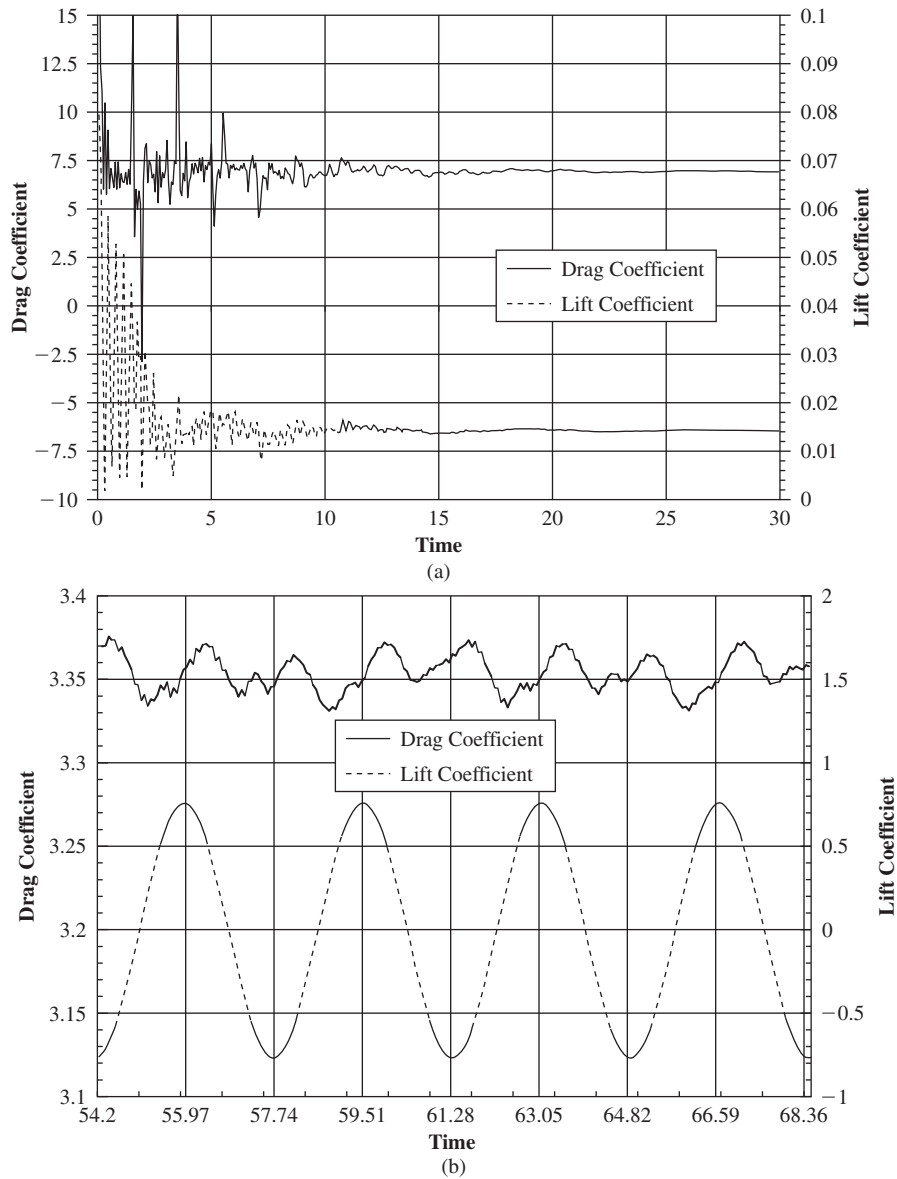


FIGURE 10.10 Drag and lift coefficients as functions of time for flow over a block. (a)  $Re = 20$  on a grid of  $701 \times 61$ , (b)  $Re = 100$  on a grid of  $1401 \times 121$ . Clearly, this flow is oscillatory at the higher Reynolds number.

20 grid points across the block ( $\Delta x = \Delta y = 0.05$ ) reasonably resolves the drag coefficient and the singularity at the block corners does not affect this convergence very much.

The explicit MacCormack scheme can be quite efficient to compute flows at high Reynolds numbers where small time steps are naturally needed to resolve high frequencies in the flow,

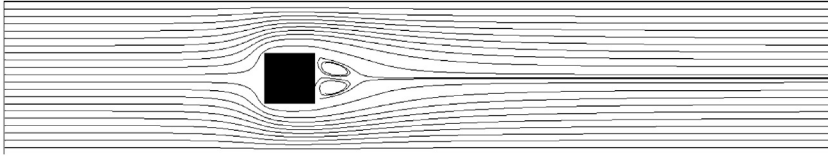


FIGURE 10.11 Streamlines for flow around a block at  $Re = 20$ . Although there is flow separation, the flow is steady.

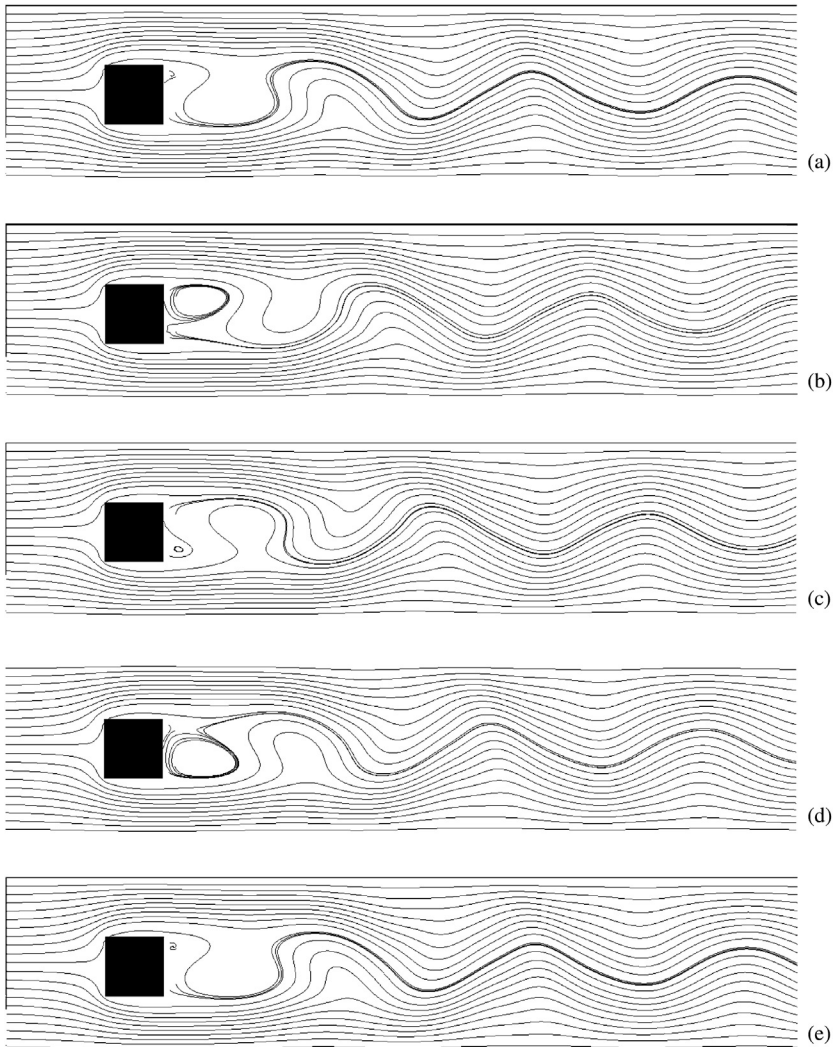
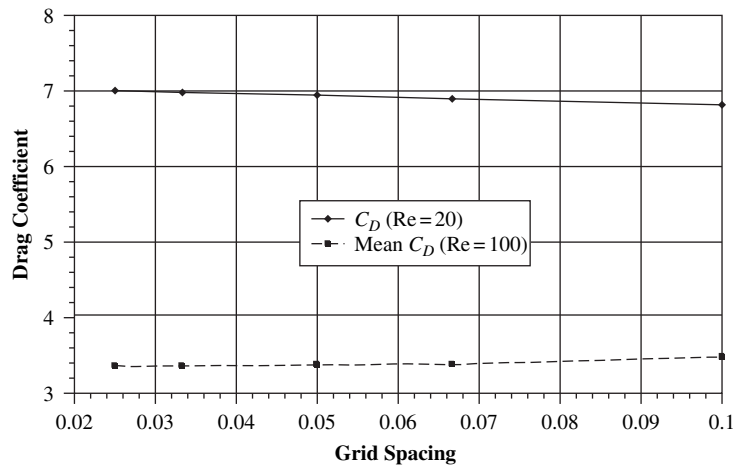


FIGURE 10.12 A sequence of flow fields around a block at  $Re = 100$  during one period of vortex shedding. (a)  $t = 40.53$ , (b)  $t = 41.50$ , (c)  $t = 42.48$ , (d)  $t = 43.45$ , (e)  $t = 44.17$ . Here the unsteadiness fully washes away the attached near-wake vortices.

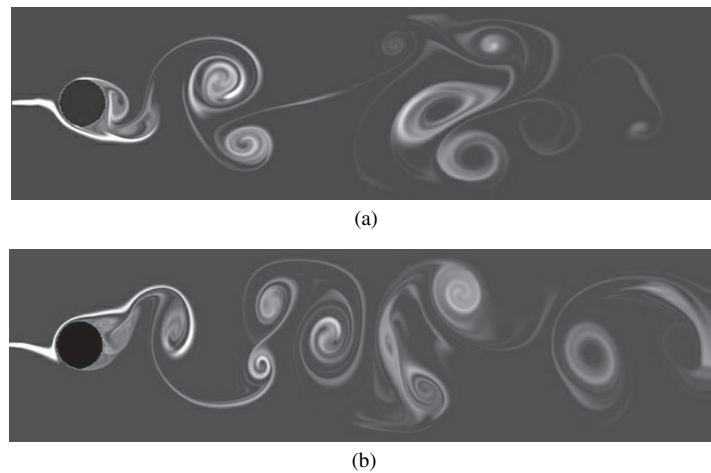


**FIGURE 10.13** Convergence tests for the drag coefficient as the grid spacing decreases. The grid spacing is equal in both directions  $\Delta x = \Delta y$ , and time step  $\Delta t$  is determined by the stability condition. Confidence in computed results increases when such results are shown to be independent of the resolution of the computational grid for decreasing grid-point spacing.

and the stability condition for the time step is no longer too restrictive. With  $\Delta x = \Delta y$  and large (grid) Reynolds numbers, the stability condition (10.110) becomes approximately:

$$\Delta t \leq \frac{\sigma}{\sqrt{2}} M \Delta x. \quad (10.155)$$

As a more complicated example, the flow around a circular cylinder confined between two parallel plates (the same geometry as the fourth example later in this section) is calculated at  $Re = 1000$  using the explicit MacCormack scheme. For flow visualization, a smoke line is introduced at the inlet. Numerically, an additional convection-diffusion equation for smoke concentration is solved similarly, with an explicit scheme at each time step coupled with the computed flow field. Two snapshots of the flow field are displayed in Figure 10.14.



**FIGURE 10.14** Smoke lines in flow around a circular cylinder between two parallel plates at  $Re = 1000$ . The flow geometry is the same as in the fourth example presented later in this section.

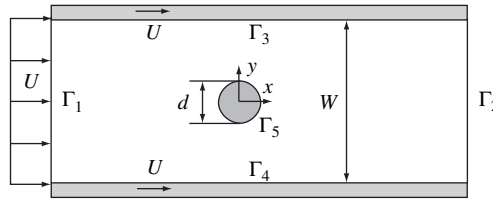
In this calculation, the flow Mach number is set at  $M = 0.3$ , and a uniform fine grid with 100 grid points across the cylinder diameter is used.

### Finite-Element Formulation for Flow Over a Cylinder Confined in a Channel

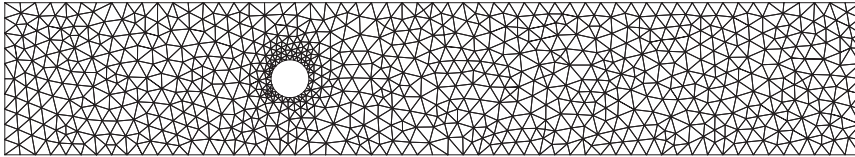
We next consider the flow over a circular cylinder moving along the center of a channel. In the computation, we fix the cylinder and use the flow geometry as shown in [Figure 10.15](#). The flow comes from the left with a uniform velocity  $U$ . Both plates of the channel are sliding to the right with the same velocity  $U$ . The diameter of the cylinder is  $d$  and the width of the channel is  $W = 4d$ . The boundary sections for the computational domain are indicated in the figure. The location of the inflow boundary  $\Gamma_1$  is selected to be at  $x_{\min} = -7.5d$ , and the location of the outflow boundary section  $\Gamma_2$  is at  $x_{\max} = 15d$ . They are both far away from the cylinder so as to minimize their influence on the flow field near the cylinder. In order to compute the flow at higher Reynolds numbers, we relax the assumptions that the flow is symmetric and steady. We will compute unsteady flow (with vortex shedding) in the full geometry and using the Cartesian coordinates shown in [Figure 10.15](#).

The first step in the finite-element method is to discretize (mesh) the computational domain described in [Figure 10.15](#). We cover the domain with triangular elements. A typical mesh is presented in [Figure 10.16](#). The mesh size is distributed in a way that finer elements are used next to the cylinder surface to better resolve the local flow field. For this example, the mixed finite-element method will be used, such that each triangular element will have six nodes as shown [Figure 10.5a](#). This element allows for curved sides that better capture the surface of the circular cylinder. The mesh in [Figure 10.16](#) has 3320 elements, 6868 velocity nodes, and 1774 pressure nodes.

The weak formulation of the Navier-Stokes equations is given in (10.134) and (10.135). For this example the body-force term is zero,  $\mathbf{g} = 0$ . In Cartesian coordinates, the weak form of the momentum equation (10.134) can be written explicitly as



**FIGURE 10.15** Computational domain for two-dimensional flow past a circular cylinder in a channel formed by parallel plates.



**FIGURE 10.16** A finite-element mesh around a cylinder. Note that mesh spacing is finer near the cylinder to properly resolve the boundary layers that form on the cylinder.

$$\begin{aligned} \int_{\Omega} \left( \frac{\partial \mathbf{u}}{\partial t} + u \frac{\partial \mathbf{u}}{\partial x} + v \frac{\partial \mathbf{u}}{\partial y} \right) \cdot \tilde{\mathbf{u}} d\Omega + \frac{2}{\text{Re}} \int_{\Omega} \left[ \frac{\partial u}{\partial x} \frac{\partial \tilde{u}}{\partial x} + \frac{1}{2} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \left( \frac{\partial \tilde{u}}{\partial y} + \frac{\partial \tilde{v}}{\partial x} \right) + \frac{\partial v}{\partial y} \frac{\partial \tilde{v}}{\partial y} \right] d\Omega \\ - \int_{\Omega} p \left( \frac{\partial \tilde{u}}{\partial x} + \frac{\partial \tilde{v}}{\partial y} \right) d\Omega = 0, \end{aligned} \quad (10.156)$$

where  $\Omega$  is the computational domain and  $\tilde{\mathbf{u}} = (\tilde{u}, \tilde{v})$ . Since the variational functions  $\tilde{u}$  and  $\tilde{v}$  are independent, the weak formulation (10.156) can be separated into two equations:

$$\int_{\Omega} \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) \tilde{u} d\Omega - \int_{\Omega} p \frac{\partial \tilde{u}}{\partial x} d\Omega + \frac{1}{\text{Re}} \int_{\Omega} \left[ 2 \frac{\partial u}{\partial x} \frac{\partial \tilde{u}}{\partial x} + \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial \tilde{u}}{\partial y} \right] d\Omega = 0, \quad (10.157)$$

$$\int_{\Omega} \left( \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) \tilde{v} d\Omega - \int_{\Omega} p \frac{\partial \tilde{v}}{\partial y} d\Omega + \frac{1}{\text{Re}} \int_{\Omega} \left[ \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \frac{\partial \tilde{v}}{\partial x} + 2 \frac{\partial v}{\partial y} \frac{\partial \tilde{v}}{\partial y} \right] d\Omega = 0. \quad (10.158)$$

The weak form of the continuity equation (10.135) is expressed as

$$- \int_{\Omega} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \tilde{p} d\Omega = 0. \quad (10.159)$$

Given a triangulation of the computational domain, for example, the mesh shown in Figure 10.16, the weak formulation of (10.157) through (10.159) can be approximated by the Galerkin finite-element formulation based on the finite-dimensional discretization of the flow variables. The Galerkin formulation can be written as:

$$\begin{aligned} \int_{\Omega^h} \left( \frac{\partial u^h}{\partial t} + u^h \frac{\partial u^h}{\partial x} + v^h \frac{\partial u^h}{\partial y} \right) \tilde{u}^h d\Omega - \int_{\Omega^h} p^h \frac{\partial \tilde{u}^h}{\partial x} d\Omega \\ + \frac{1}{\text{Re}} \int_{\Omega^h} \left[ 2 \frac{\partial u^h}{\partial x} \frac{\partial \tilde{u}^h}{\partial x} + \left( \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \right) \frac{\partial \tilde{u}^h}{\partial y} \right] d\Omega = 0, \end{aligned} \quad (10.160)$$

$$\begin{aligned} \int_{\Omega^h} \left( \frac{\partial v^h}{\partial t} + u^h \frac{\partial v^h}{\partial x} + v^h \frac{\partial v^h}{\partial y} \right) \tilde{v}^h d\Omega - \int_{\Omega^h} p^h \frac{\partial \tilde{v}^h}{\partial y} d\Omega \\ + \frac{1}{\text{Re}} \int_{\Omega^h} \left[ \left( \frac{\partial u^h}{\partial y} + \frac{\partial v^h}{\partial x} \right) \frac{\partial \tilde{v}^h}{\partial x} + 2 \frac{\partial v^h}{\partial y} \frac{\partial \tilde{v}^h}{\partial y} \right] d\Omega = 0, \end{aligned} \quad (10.161)$$

and

$$- \int_{\Omega^h} \left( \frac{\partial u^h}{\partial x} + \frac{\partial v^h}{\partial y} \right) \tilde{p}^h d\Omega = 0, \quad (10.162)$$

where  $h$  indicates a given triangulation of the computational domain.

The time derivatives in (10.160) and (10.161) can be discretized by finite-difference methods. We first evaluate all the terms in (10.160) to (10.162) at a given time instant



$t = t_{n+1}$  (fully implicit discretization). Then the time derivative in (10.160) and (10.161) can be approximated as

$$\frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t_{n+1}) \approx \alpha \frac{\mathbf{u}(\mathbf{x}, t_{n+1}) - \mathbf{u}(\mathbf{x}, t_n)}{\Delta t} - \beta \frac{\partial \mathbf{u}}{\partial t}(\mathbf{x}, t_n), \quad (10.163)$$

where  $\Delta t = t_{n+1} - t_n$  is the time step. The approximation in (10.163) is first-order accurate in time when  $\alpha = 1$  and  $\beta = 0$ . It can be improved to second-order accurate by selecting  $\alpha = 2$  and  $\beta = 1$  which is a variation of the well-known Crank-Nicolson scheme.

As (10.160) and (10.161) are nonlinear, iterative methods are often used for the solution. In Newton's method, the flow variables at the current time  $t = t_{n+1}$  are often expressed as:

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}, t_{n+1}) &= \mathbf{u}^*(\mathbf{x}, t_{n+1}) + \mathbf{u}'(\mathbf{x}, t_{n+1}), \\ p^h(\mathbf{x}, t_{n+1}) &= p^*(\mathbf{x}, t_{n+1}) + p'(\mathbf{x}, t_{n+1}), \end{aligned} \quad (10.164)$$

where  $\mathbf{u}^*$  and  $p^*$  are the guesstimated values of velocity and pressure during the iteration;  $\mathbf{u}'$  and  $p'$  are the corrections sought at each iteration.

Substituting (10.163) and (10.164) into Galerkin formulation, (10.160) through (10.162), and linearizing the equations with respect to the correction variables, we have:

$$\begin{aligned} & \int_{\Omega^h} \left( \frac{\alpha}{\Delta t} u' + u^* \frac{\partial u'}{\partial x} + v^* \frac{\partial u'}{\partial y} + \frac{\partial u^*}{\partial x} u' + \frac{\partial u^*}{\partial y} v' \right) \tilde{u}^h d\Omega - \int_{\Omega^h} p' \frac{\partial \tilde{u}^h}{\partial x} d\Omega \\ & + \frac{1}{\text{Re}} \int_{\Omega^h} \left[ 2 \frac{\partial u'}{\partial x} \frac{\partial \tilde{u}^h}{\partial x} + \left( \frac{\partial u'}{\partial y} + \frac{\partial v'}{\partial x} \right) \frac{\partial \tilde{u}^h}{\partial y} \right] d\Omega \\ & = - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (u^* - u(t_n)) - \beta \frac{\partial u}{\partial t}(t_n) + u^* \frac{\partial u^*}{\partial x} + v^* \frac{\partial u^*}{\partial y} \right] \tilde{u}^h d\Omega + \int_{\Omega^h} p^* \frac{\partial \tilde{u}^h}{\partial x} d\Omega \\ & - \frac{1}{\text{Re}} \int_{\Omega^h} \left[ 2 \frac{\partial u^*}{\partial x} \frac{\partial \tilde{u}^h}{\partial x} + \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial \tilde{u}^h}{\partial y} \right] d\Omega, \end{aligned} \quad (10.165)$$

$$\begin{aligned} & \int_{\Omega^h} \left( \frac{\alpha}{\Delta t} v' + u^* \frac{\partial v'}{\partial x} + v^* \frac{\partial v'}{\partial y} + \frac{\partial v^*}{\partial x} u' + \frac{\partial v^*}{\partial y} v' \right) \tilde{v}^h d\Omega \\ & - \int_{\Omega^h} p' \frac{\partial \tilde{v}^h}{\partial y} d\Omega + \frac{1}{\text{Re}} \int_{\Omega^h} \left[ \left( \frac{\partial u'}{\partial y} + \frac{\partial v'}{\partial x} \right) \frac{\partial \tilde{v}^h}{\partial x} + 2 \frac{\partial v'}{\partial y} \frac{\partial \tilde{v}^h}{\partial y} \right] d\Omega \\ & = - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (v^* - v(t_n)) - \beta \frac{\partial v}{\partial t}(t_n) + u^* \frac{\partial v^*}{\partial x} + v^* \frac{\partial v^*}{\partial y} \right] \tilde{v}^h d\Omega \\ & + \int_{\Omega^h} p^* \frac{\partial \tilde{v}^h}{\partial y} d\Omega - \frac{1}{\text{Re}} \int_{\Omega^h} \left[ \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial \tilde{v}^h}{\partial x} + 2 \frac{\partial v^*}{\partial y} \frac{\partial \tilde{v}^h}{\partial y} \right] d\Omega, \end{aligned} \quad (10.166)$$

and

$$-\int_{\Omega^h} \left( \frac{\partial u'}{\partial x} + \frac{\partial v'}{\partial y} \right) \tilde{p}^h d\Omega = \int_{\Omega^h} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) \tilde{p}^h d\Omega. \quad (10.167)$$

As the functions in the integrals, unless specified otherwise, are all evaluated at the current time instant  $t_{n+1}$ , the temporal discretization in (10.165) and (10.166) is fully implicit and unconditionally stable. The terms on the right-hand side of (10.165) through (10.167) represent the residuals of the corresponding equations and can be used to monitor the convergence of the nonlinear iteration.

Similar to the one-dimensional case in Section 10.3, the finite-dimensional discretization of the low variables can be constructed using shape (or interpolation) functions:

$$u' = \sum_A u_A N_A^u(x, y), \quad v' = \sum_A v_A N_A^u(x, y), \quad p' = \sum_B p_B N_B^p(x, y), \quad (10.168)$$

where  $N_A^u(x, y)$  and  $N_B^p(x, y)$  are the shape functions for the velocity and the pressure, respectively. They are not necessarily the same. In order to satisfy the LBB stability condition, the shape function  $N_A^u(x, y)$  in the mixed finite-element formulation should be one order higher than  $N_B^p(x, y)$ , as discussed in Section 10.4. The summation over  $A$  is through all the velocity nodes, while the summation over  $B$  runs through all the pressure nodes. The variational functions may be expressed in terms of the same shape functions:

$$\tilde{u}^h = \sum_A \tilde{u}_A N_A^u(x, y), \quad \tilde{v}^h = \sum_A \tilde{v}_A N_A^u(x, y), \quad \tilde{p}^h = \sum_B \tilde{p}_B N_B^p(x, y). \quad (10.169)$$

Since the Galerkin formulation (10.165) through (10.167) is valid for all possible choices of the variational functions, the coefficients in (10.169) should be arbitrary. In this way, the Galerkin formulation (10.165) through (10.167) reduces to a system of algebraic equations:

$$\begin{aligned} & \sum_{A'} u_{A'} \int_{\Omega^h} \left[ \left( \frac{\alpha}{\Delta t} N_{A'}^u + u^* \frac{\partial N_{A'}^u}{\partial x} + v^* \frac{\partial N_{A'}^u}{\partial y} + \frac{\partial u^*}{\partial x} N_{A'}^u \right) N_A^u + \frac{1}{Re} \left( 2 \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial x} + \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial y} \right) \right] d\Omega \\ & + \sum_{A'} v_{A'} \int_{\Omega^h} \left( \frac{\partial u^*}{\partial y} N_{A'}^u N_A^u + \frac{1}{Re} \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial y} \right) d\Omega - \sum_{B'} p_{B'} \int_{\Omega^h} N_{B'}^p \frac{\partial N_A^u}{\partial x} d\Omega \\ & = - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (u^* - u(t_n)) - \beta \frac{\partial u}{\partial t}(t_n) + u^* \frac{\partial u^*}{\partial x} + v^* \frac{\partial u^*}{\partial y} \right] N_A^u d\Omega + \int_{\Omega^h} p^* \frac{\partial N_A^u}{\partial x} d\Omega \\ & - \frac{1}{Re} \int_{\Omega^h} \left[ 2 \frac{\partial u^*}{\partial x} \frac{\partial N_A^u}{\partial x} + \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial N_A^u}{\partial y} \right] d\Omega, \end{aligned} \quad (10.170)$$

$$\begin{aligned}
& \sum_{A'} v_{A'} \int_{\Omega^h} \left[ \left( \frac{\alpha}{\Delta t} N_{A'}^u + u^* \frac{\partial N_{A'}^u}{\partial x} + v^* \frac{\partial N_{A'}^u}{\partial y} + \frac{\partial u^*}{\partial y} N_{A'}^u \right) N_A^u + \frac{1}{Re} \left( \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial x} + 2 \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial y} \right) \right] d\Omega \\
& + \sum_{A'} u_{A'} \int_{\Omega^h} \left( \frac{\partial v^*}{\partial x} N_{A'}^u N_A^u + \frac{1}{Re} \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial x} \right) d\Omega - \sum_{B'} p_{B'} \int_{\Omega^h} N_{B'}^p \frac{\partial N_A^u}{\partial y} d\Omega \\
& = - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (v^* - v(t_n)) - \beta \frac{\partial v^*}{\partial t}(t_n) + u^* \frac{\partial v^*}{\partial x} + v^* \frac{\partial v^*}{\partial y} \right] N_A^u d\Omega + \int_{\Omega^h} p^* \frac{\partial N_A^u}{\partial y} d\Omega \\
& - \frac{1}{Re} \int_{\Omega^h} \left[ \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial N_A^u}{\partial x} + 2 \frac{\partial v^*}{\partial y} \frac{\partial N_A^u}{\partial y} \right] d\Omega,
\end{aligned} \tag{10.171}$$

and

$$- \sum_{A'} u_{A'} \int_{\Omega^h} \left( \frac{\partial N_{A'}^u}{\partial x} N_B^p \right) d\Omega - \sum_{A'} u_{A'} \int_{\Omega^h} \left( \frac{\partial N_{A'}^u}{\partial y} N_B^p \right) d\Omega = \int_{\Omega^h} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) N_B^p d\Omega, \tag{10.172}$$

for all the velocity nodes  $A$  and pressure nodes  $B$ . Equations (10.170) through (10.172) can be organized into matrix form:

$$\begin{pmatrix} \mathbf{A}_{uu} & \mathbf{A}_{uv} & \mathbf{B}_{up} \\ \mathbf{A}_{vu} & \mathbf{A}_{vv} & \mathbf{B}_{vp} \\ \mathbf{B}_{up}^T & \mathbf{B}_{vp}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_u \\ \mathbf{f}_v \\ \mathbf{f}_p \end{pmatrix}, \tag{10.173}$$

where:

$$\begin{aligned}
\mathbf{A}_{uu} &= [A_{AA'}^{uu}], \quad \mathbf{A}_{uv} = [A_{AA'}^{uv}], \quad \mathbf{B}_{up} = [B_{AB'}^{up}], \\
\mathbf{A}_{vu} &= [A_{AA'}^{vu}], \quad \mathbf{A}_{vv} = [A_{AA'}^{vv}], \quad \mathbf{B}_{vp} = [B_{AB'}^{vp}], \\
\mathbf{u} &= \{u_{A'}\}, \quad \mathbf{v} = \{v_{A'}\}, \quad \mathbf{p} = \{p_{B'}\}, \\
\mathbf{f}_u &= \{f_A^u\}, \quad \mathbf{f}_v = \{f_A^v\}, \quad \mathbf{f}_p = \{f_B^p\},
\end{aligned} \tag{10.174}$$

and:

$$A_{AA'}^{uu} = \int_{\Omega^h} \left[ \left( \frac{\alpha}{\Delta t} N_{A'}^u + u^* \frac{\partial N_{A'}^u}{\partial x} + v^* \frac{\partial N_{A'}^u}{\partial y} + \frac{\partial u^*}{\partial y} N_{A'}^u \right) N_A^u + \frac{1}{Re} \left( 2 \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial x} + \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial y} \right) \right] d\Omega, \tag{10.175}$$

$$A_{AA'}^{uv} = \int_{\Omega^h} \left( \frac{\partial u^*}{\partial y} N_{A'}^u N_A^u + \frac{1}{Re} \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial y} \right) d\Omega, \tag{10.176}$$

$$A_{AA'}^{vu} = \int_{\Omega^h} \left( \frac{\partial v^*}{\partial x} N_{A'}^u N_A^u + \frac{1}{Re} \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial x} \right) d\Omega, \tag{10.177}$$

$$A_{AA'}^{vv} = \int_{\Omega^h} \left[ \left( \frac{\alpha}{\Delta t} N_{A'}^u + u^* \frac{\partial N_{A'}^u}{\partial x} + v^* \frac{\partial N_{A'}^u}{\partial y} + \frac{\partial v^*}{\partial y} N_{A'}^u \right) N_A^u + \frac{1}{Re} \left( \frac{\partial N_{A'}^u}{\partial x} \frac{\partial N_A^u}{\partial x} + 2 \frac{\partial N_{A'}^u}{\partial y} \frac{\partial N_A^u}{\partial y} \right) \right] d\Omega, \tag{10.178}$$

$$B_{AB'}^{up} = - \int_{\Omega^h} N_{B'}^p \frac{\partial N_A^u}{\partial x} d\Omega, \quad (10.179)$$

$$B_{AB'}^{vp} = - \int_{\Omega^h} N_{B'}^p \frac{\partial N_A^v}{\partial y} d\Omega, \quad (10.180)$$

$$\begin{aligned} f_A^u = & - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (u^* - u(t_n)) - \beta \frac{\partial u}{\partial t}(t_n) + u^* \frac{\partial u^*}{\partial x} + v^* \frac{\partial u^*}{\partial y} \right] N_A^u d\Omega \\ & + \int_{\Omega^h} p^* \frac{\partial N_A^u}{\partial x} d\Omega - \frac{1}{Re} \int_{\Omega^h} \left[ 2 \frac{\partial u^*}{\partial x} \frac{\partial N_A^u}{\partial x} + \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial N_A^u}{\partial y} \right] d\Omega, \end{aligned} \quad (10.181)$$

$$\begin{aligned} f_A^v = & - \int_{\Omega^h} \left[ \frac{\alpha}{\Delta t} (v^* - v(t_n)) - \beta \frac{\partial v}{\partial t}(t_n) + u^* \frac{\partial v^*}{\partial x} + v^* \frac{\partial v^*}{\partial y} \right] N_A^v d\Omega \\ & + \int_{\Omega^h} p^* \frac{\partial N_A^v}{\partial y} d\Omega - \frac{1}{Re} \int_{\Omega^h} \left[ \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial N_A^v}{\partial x} + 2 \frac{\partial v^*}{\partial y} \frac{\partial N_A^v}{\partial y} \right] d\Omega, \end{aligned} \quad (10.182)$$

$$f_B^p = \int_{\Omega^h} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) N_B^p d\Omega. \quad (10.183)$$

The practical evaluation of the integrals in (10.175) through (10.183) is done element-wise. We need to construct the shape functions locally and transform these global integrals into local integrals over each element.

In the finite-element method, the global shape functions have very compact support. They are zero everywhere except in the neighborhood of the corresponding grid point in the mesh. It is convenient to cast the global formulation using the element point of view (Section 10.3). In this element view, the local shape functions are defined inside each element. The global shape functions are the assembly of the relevant local ones. For example, the global shape function corresponding to the grid point  $A$  in the finite-element mesh consists of the local shape functions of all the elements that share this grid point. An element in the physical space can be mapped into a standard element, as shown in Figure 10.17, and the local shape functions can be defined on this standard element. The mapping is given by

$$x(\xi, \eta) = \sum_{a=1}^6 x_a^e \phi_a(\xi, \eta) \quad \text{and} \quad y(\xi, \eta) = \sum_{a=1}^6 y_a^e \phi_a(\xi, \eta), \quad (10.184)$$

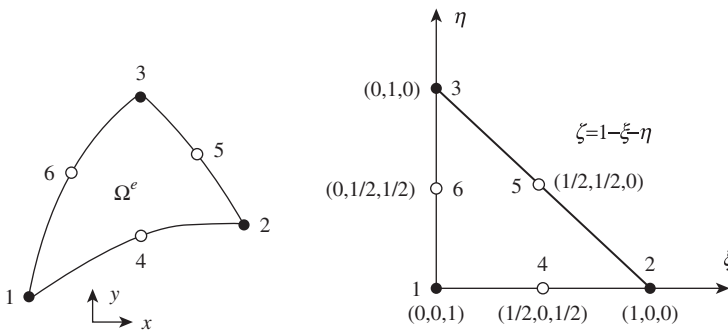


FIGURE 10.17 A quadratic triangular finite-element mapping into the standard element. The single-digit numbers indicate equivalent points on the left- and right-side elements.

where  $(x_a^e, y_a^e)$  are the coordinates of the nodes in the element  $e$ . The local shape functions are  $\phi_a$ . For a quadratic triangular element they are defined as

$$\phi_1 = \zeta(2\zeta - 1), \quad \phi_2 = \xi(2\xi - 1), \quad \phi_3 = \eta(2\eta - 1), \quad \phi_4 = 4\xi\zeta, \quad \phi_5 = 4\xi\eta, \quad \phi_6 = 4\eta\zeta, \quad (10.185)$$

where  $\zeta = 1 - \xi - \eta$ . As shown in Figure 10.17, the mapping (10.184) is able to handle curved triangles. The variation of the flow variables within this element can also be expressed in terms of their values at the nodes of the element and the local shape functions,

$$u' = \sum_{a=1}^6 u_a^e \phi_a(\xi, \eta), \quad v' = \sum_{a=1}^6 v_a^e \phi_a(\xi, \eta), \quad p' = \sum_{b=1}^3 p_b^e \psi_b(\xi, \eta). \quad (10.186)$$

Here the shape functions for velocities are quadratic and the same as the coordinates. The shape functions for the pressure are chosen to be linear, thus one order less than those for the velocities. They are given by

$$\psi_1 = \zeta, \quad \psi_2 = \xi, \quad \psi_3 = \eta. \quad (10.187)$$

Furthermore, the integration over the global computational domain can be written as the summation of the integrations over all the elements in the domain. As most of these integrations will be zero, the nonzero ones are grouped as element matrices and vectors:

$$\begin{aligned} \mathbf{A}_{uu}^e &= [A_{aa'}^{euu}], \quad \mathbf{A}_{uv}^e = [A_{aa'}^{euv}], \quad \mathbf{B}_{up}^e = [B_{ab'}^{eup}], \\ \mathbf{A}_{vu}^e &= [A_{aa'}^{evu}], \quad \mathbf{A}_{vv}^e = [A_{aa'}^{evv}], \quad \mathbf{B}_{vp}^e = [B_{ab'}^{evp}], \\ \mathbf{f}_u^e &= \{f_a^{eu}\}, \quad \mathbf{f}_v^e = \{f_a^{ev}\}, \quad \mathbf{f}_p^e = \{f_b^{ep}\}, \end{aligned} \quad (10.188)$$

where:

$$A_{aa'}^{euu} = \int_{\Omega^e} \left[ \left( \frac{\alpha}{\Delta t} \phi_{a'} + u^* \frac{\partial \phi_{a'}}{\partial x} + v^* \frac{\partial \phi_{a'}}{\partial y} + \frac{\partial u^*}{\partial x} \phi_{a'} \right) \phi_a + \frac{1}{\text{Re}} \left( 2 \frac{\partial \phi_{a'}}{\partial x} \frac{\partial \phi_a}{\partial x} + \frac{\partial \phi_{a'}}{\partial y} \frac{\partial \phi_a}{\partial y} \right) \right] d\Omega, \quad (10.189)$$

$$A_{aa'}^{euv} = \int_{\Omega^e} \left( \frac{\partial u^*}{\partial y} \phi_{a'} \phi_a + \frac{1}{\text{Re}} \frac{\partial \phi_{a'}}{\partial x} \frac{\partial \phi_a}{\partial y} \right) d\Omega, \quad (10.190)$$

$$A_{aa'}^{evu} = \int_{\Omega^e} \left( \frac{\partial v^*}{\partial x} \phi_{a'} \phi_a + \frac{1}{\text{Re}} \frac{\partial \phi_{a'}}{\partial y} \frac{\partial \phi_a}{\partial x} \right) d\Omega, \quad (10.191)$$

$$A_{aa'}^{evv} = \int_{\Omega^e} \left[ \left( \frac{\alpha}{\Delta t} \phi_{a'} + u^* \frac{\partial \phi_{a'}}{\partial x} + v^* \frac{\partial \phi_{a'}}{\partial y} + \frac{\partial v^*}{\partial y} \phi_{a'} \right) \phi_a + \frac{1}{\text{Re}} \left( \frac{\partial \phi_{a'}}{\partial x} \frac{\partial \phi_a}{\partial x} + 2 \frac{\partial \phi_{a'}}{\partial y} \frac{\partial \phi_a}{\partial y} \right) \right] d\Omega, \quad (10.192)$$

$$B_{ab'}^{eup} = - \int_{\Omega^e} \psi_{b'} \frac{\partial \phi_a}{\partial x} d\Omega, \quad (10.193)$$

$$B_{ab'}^{evp} = - \int_{\Omega^e} \psi_{b'} \frac{\partial \phi_a}{\partial y} d\Omega, \quad (10.194)$$

$$f_a^{eu} = - \int_{\Omega^e} \left[ \frac{\alpha}{\Delta t} (u^* - u(t_n)) - \beta \frac{\partial u}{\partial t}(t_n) + u^* \frac{\partial u^*}{\partial x} + v^* \frac{\partial u^*}{\partial y} \right] \phi_a d\Omega + \int_{\Omega^e} p^* \frac{\partial \phi_a}{\partial x} d\Omega - \frac{1}{Re} \int_{\Omega^e} \left[ 2 \frac{\partial u^*}{\partial x} \frac{\partial \phi_a}{\partial x} + \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial \phi_a}{\partial y} \right] d\Omega, \quad (10.195)$$

$$f_a^{ev} = - \int_{\Omega^e} \left[ \frac{\alpha}{\Delta t} (v^* - v(t_n)) - \beta \frac{\partial v}{\partial t}(t_n) + u^* \frac{\partial v^*}{\partial x} + v^* \frac{\partial v^*}{\partial y} \right] \phi_a d\Omega + \int_{\Omega^e} p^* \frac{\partial \phi_a}{\partial y} d\Omega - \frac{1}{Re} \int_{\Omega^e} \left[ \left( \frac{\partial u^*}{\partial y} + \frac{\partial v^*}{\partial x} \right) \frac{\partial \phi_a}{\partial x} + 2 \frac{\partial v^*}{\partial y} \frac{\partial \phi_a}{\partial y} \right] d\Omega, \quad (10.196)$$

$$f_b^{ep} = \int_{\Omega^e} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right) \psi_b d\Omega. \quad (10.197)$$

The indices  $a$  and  $a'$  run from 1 to 6, and  $b$  and  $b'$  run from 1 to 3.

The integrals in the above expressions can be evaluated by numerical integration rules,

$$\int_{\Omega^e} f(x, y) d\Omega = \int_0^1 \int_0^{1-\eta} f(\xi, \eta) J(\xi, \eta) d\xi d\eta = \frac{1}{2} \sum_{l=1}^{N_{int}} f(\xi_l, \eta_l) J(\xi_l, \eta_l) W_l, \quad (10.198)$$

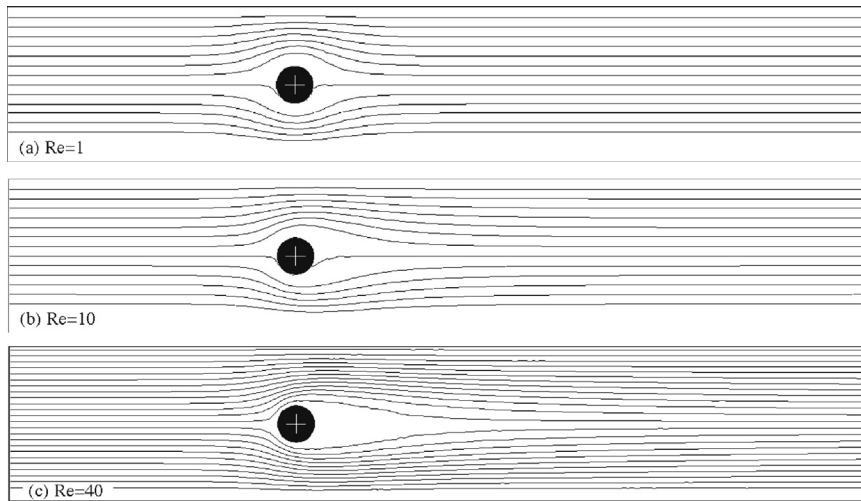
where the Jacobian of the mapping (10.184) is given by  $J = x_\xi y_\eta - x_\eta y_\xi$ . Here  $N_{int}$  is the number of numerical integration points and  $W_l$  is the weight of the  $l$ th integration point. For this example, a seven-point integration formula with degree of precision of five (see Hughes, 1987) was used.

The global matrices and vectors in (10.173) are the summations of the element matrices and vectors in (10.188) over all the elements. In the process of summation (assembly), a mapping of the local nodes in each element to the global node numbers is needed. This information is commonly available for any finite-element mesh.

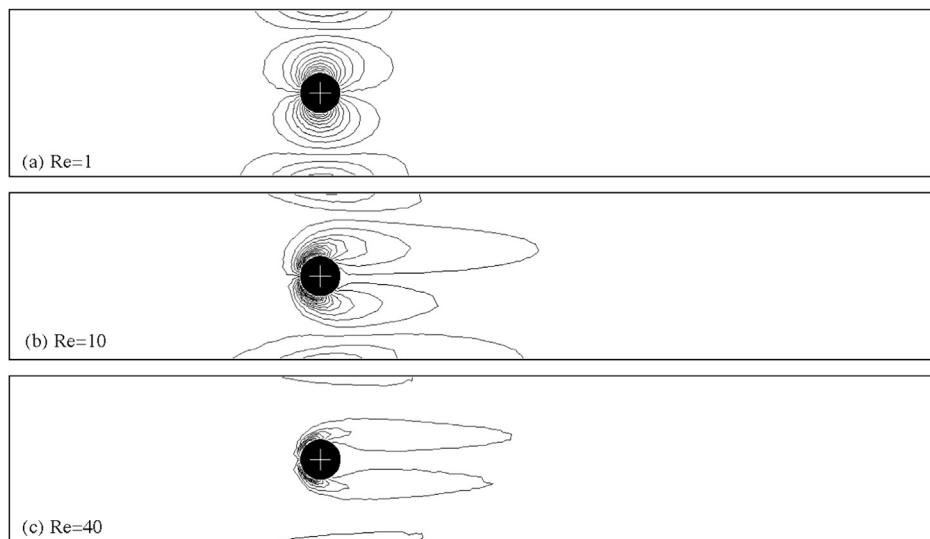
Once the matrix equation (10.173) is generated, we may impose the essential boundary conditions for the velocities. One simple method is to use the equation of the boundary condition to replace the corresponding equation in the original matrix or one can multiply a large constant by the equation of the boundary condition and add this equation to the original system of equations in order to preserve the structure of the matrix. The resulting matrix equation may be solved using common direct or iterative solvers for a linear algebraic system of equations.

Figures 10.18 and 10.19 display the streamlines and vorticity lines around the cylinder at three Reynolds numbers:  $Re = 1, 10$ , and  $40$ . For these Reynolds numbers, the flow is steady and should be symmetric above and below the cylinder. However, due to the imperfection in the mesh used for the calculation and as shown in Figure 10.16, the calculated flow field is not perfectly symmetric. From Figure 10.18 we observe the increase in the size of the wake behind the cylinder as the Reynolds number increases. In Figure 10.19, we see the effects of the Reynolds number in the vorticity build up in front of the cylinder, and in the convection of the vorticity by the flow.

We next compute the case with Reynolds number  $Re = 100$ . In this case, the flow is expected to be unsteady. Periodic vortex shedding occurs. In order to capture the details of the flow, we used a finer mesh than the one shown in Figure 10.16. The finer mesh has 9222 elements, 18,816 velocity nodes, and 4797 pressure nodes. In this calculation, the flow



**FIGURE 10.18** Streamlines for flow around a cylinder at three different Reynolds numbers: 1, 10, and 40. Here the streamlines are nearly fore-aft symmetric at the lowest  $Re$ , but the cylinder's wake grows with increasing Reynolds number.



**FIGURE 10.19** Vorticity contours for flow around a cylinder at the three different Reynolds numbers shown in [Figure 10.18](#). Top-to-bottom asymmetry in the flow field arises because the mesh above and below the cylinder are not mirror images.

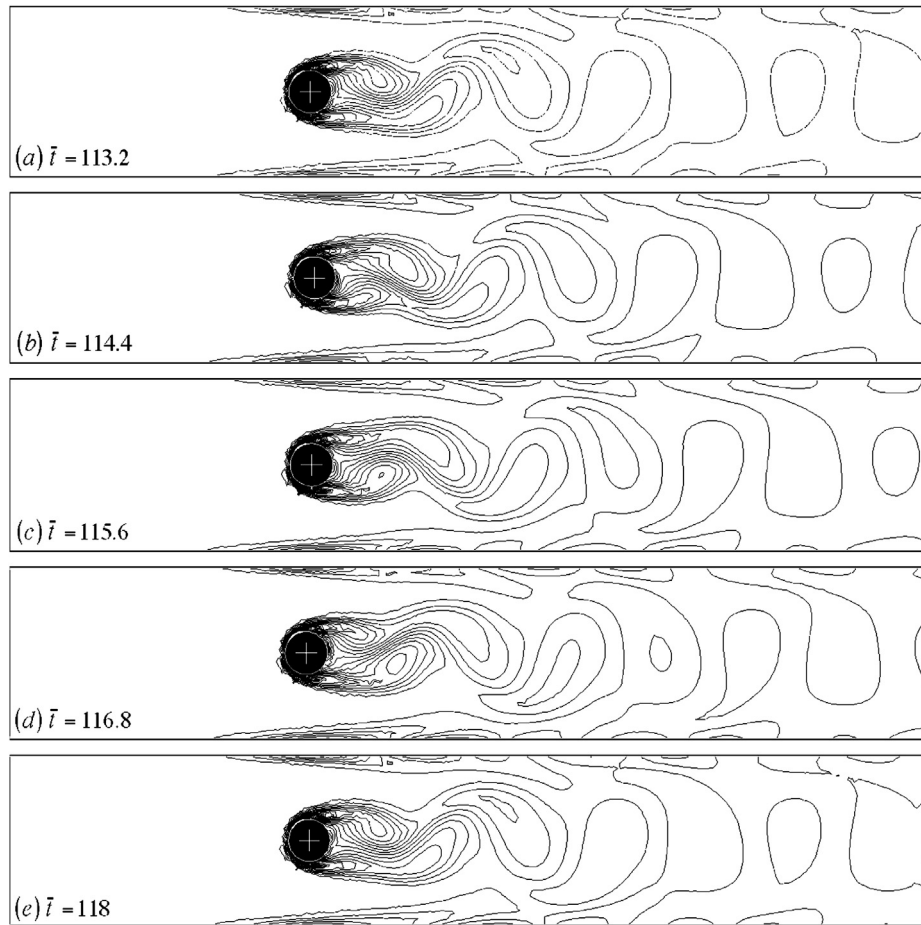
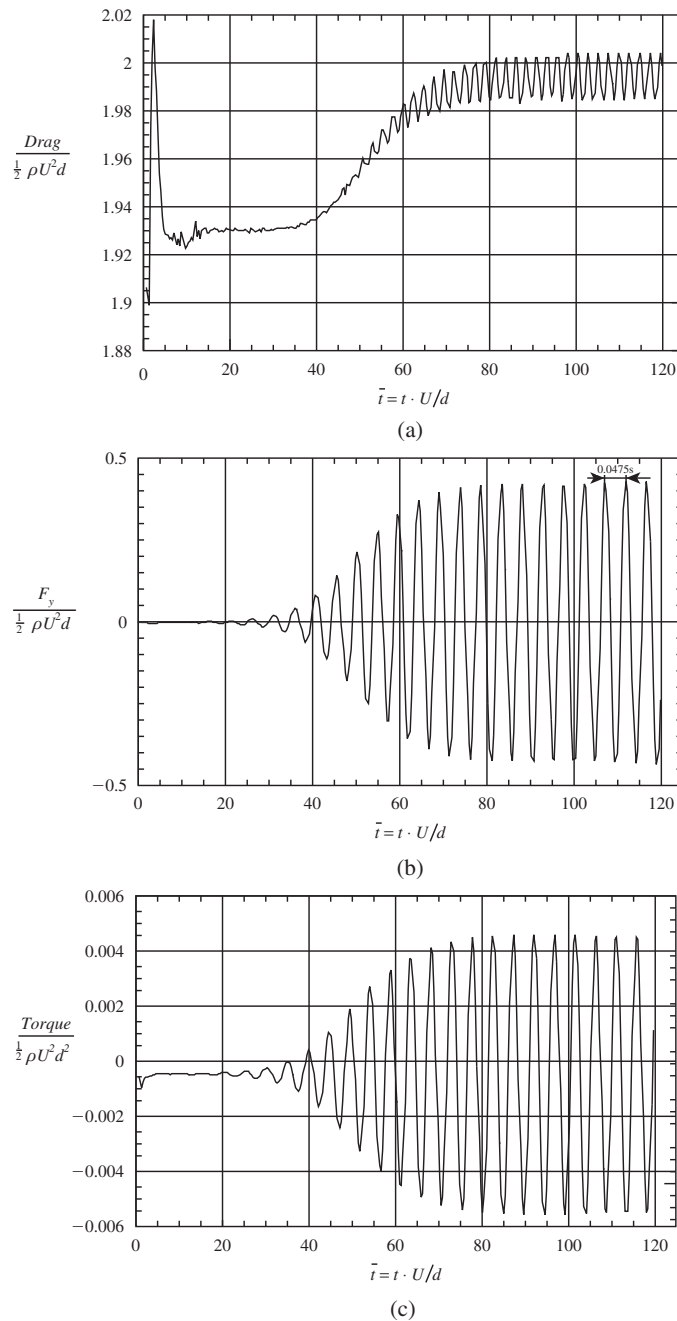


FIGURE 10.20 Vorticity lines for flow around a cylinder at Reynolds number  $Re = 100$ .  $\bar{t} = tU/d$  is the dimensionless time. Here the formation of a von Karman vortex street is evident.

starts from rest. Initially, the flow is symmetric, and the wake behind the cylinder grows bigger and stronger. Then, the wake becomes unstable, undergoes a supercritical Hopf bifurcation, and sheds periodically away from the cylinder. The periodic vortex shedding forms the well-known von Karman vortex street. The vorticity lines are presented in Figure 10.20 for a complete cycle of vortex shedding.

For this case with  $Re = 100$ , we plot in Figure 10.21 the history of the forces and torque acting on the cylinder. The oscillations shown in the lift and torque plots are typical for the supercritical Hopf bifurcation. The nonzero mean value of the torque shown in Figure 10.21c is due to the asymmetry in the finite-element mesh. It is clear that the flow becomes fully periodic at the times shown in Figure 10.20. The period of the oscillation is measured as  $\tau = 0.0475s$  or  $\bar{\tau} = 4.75$  in the nondimensional units. This period corresponds to a nondimensional Strouhal number  $S = nd/U = 0.21$ , where  $n$  is the frequency of the





**FIGURE 10.21** History of forces and torque acting on the cylinder at  $Re = 100$ : (a) drag coefficient; (b) lift coefficient; and (c) coefficient for the torque. Clearly the alternate shedding of vortices leads to oscillations in the fluid-dynamic loads on the cylinder.

shedding. In the literature, the value of the Strouhal number for an unbounded uniform flow around a cylinder is found to be around 0.167 at  $Re = 100$  (e.g., [Wen & Lin, 2001](#)). The difference could be caused by the geometry in which the cylinder is confined in a channel.

## 10.6. CONCLUDING REMARKS

It should be strongly emphasized that CFD is merely a tool for analyzing fluid-flow problems. If it is used correctly, it can provide useful information cheaply and quickly. However, it can easily be misused or even abused. In today's computer age, there is a tendency for people to trust the output from a computer, especially when they do not know the equations that were solved or how the computations were performed. One certainly should be aware of the assumptions used in producing the results from a CFD model.

As we have previously discussed, CFD is never exact. There are uncertainties involved in CFD predictions. However, one is able to gain more confidence in CFD predictions by following a few steps. Tests on some benchmark problems with known solutions are often encouraged. A mesh-refinement test is normally a must in order to be sure that the numerical solution converges to something meaningful. A similar test with the time step for unsteady flow problems is often desired. If the boundary locations and conditions are in doubt, their effects on the CFD predictions should be minimized. Furthermore, the sensitivity of the CFD predictions to some key parameters in the problem should be investigated for practical design problems.

In this chapter, we have presented the basics of the finite-difference and finite-element methods and their applications in CFD. There are other kinds of numerical methods, for example, the spectral method and the spectral element method, which are often used in CFD. They share the common approach that discretizes the Navier-Stokes equations into a system of algebraic equations. However, a class of new numerical techniques including the lattice gas cellular automata, the lattice Boltzmann method, and dissipative particle dynamics does not start from the continuum Navier-Stokes equations. Unlike the conventional methods discussed in this chapter, they are based on simplified kinetic models that incorporate the essential physics of the microscopic or mesoscopic processes so that the macroscopic-averaged properties obey the desired macroscopic Navier-Stokes equations.

## EXERCISES

- 10.1. Show that the stability condition for the explicit scheme (10.10) is the condition (10.26).  
 10.2. For the heat conduction equation  $\partial T / \partial t - D(\partial^2 T / \partial x^2) = 0$ , one of the discretized forms is

$$-sT_{j+1}^{n+1} + (1 + 2s) T_j^{n+1} - sT_{j-1}^{n+1} = T_j^n$$

where  $s = D \Delta t / \Delta x^2$ . Show that this implicit algorithm is always stable.

- 10.3. An insulated rod initially has a temperature of  $T(x, 0) = 0^\circ\text{C}$  ( $0 \leq x \leq 1$ ). At  $t = 0$  hot reservoirs ( $T = 100^\circ\text{C}$ ) are brought into contact with the two ends,  $A$  ( $x = 0$ ) and  $B$

( $x = 1$ ):  $T(0, t) = T(1, t) = 100^\circ\text{C}$ . Numerically find the temperature  $T(x, t)$  of any point in the rod. The governing equation of the problem is the heat conduction equation  $\partial T / \partial t - D (\partial^2 T / \partial x^2) = 0$ . The exact solution to this problem is

$$T^*(x_j, t_n) = 100 - \sum_{m=1}^M \frac{400}{(2m-1)\pi} \sin \left[ (2m-1) \pi x_j \right] \exp \left[ -D (2m-1)^2 \pi^2 t_n \right], \quad (10.199)$$

where  $M$  is the number of terms used in the approximation.

- a) Try to solve the problem with the explicit forward time, centered space (FTCS) scheme. Use the parameter  $s = D \Delta t / \Delta x^2 = 0.5$  and  $0.6$  to test the stability of the scheme.
  - b) Solve the problem with a stable explicit or implicit scheme. Test the rate of convergence numerically using the error at  $x = 0.5$ .
- 10.4. Derive the weak form, Galerkin form, and the matrix form of the following strong problem:

Given functions  $D(x)$ ,  $f(x)$ , and constants  $g, h$ , find  $u(x)$  such that  
 $[D(x)u, v]_x + f(x) = 0$  on  $\Omega = (0, 1)$ ,  
 with  $u(0) = g$  and  $-u_x(1) = h$ .

Write a computer program solving this problem using piecewise-linear shape functions. You may set  $D = 1$ ,  $g = 1$ , and  $h = 1$ . Check your numerical result with the exact solution.

- 10.5. Solve numerically the steady convective transport equation,

$$u \frac{\partial T}{\partial x} = D \frac{\partial^2 T}{\partial x^2}, \text{ for } 0 \leq x \leq 1,$$

with two boundary conditions  $T(0) = 0$  and  $T(1) = 1$ , where  $u$  and  $D$  are two constants:

- a) using the centered finite-difference scheme in equation (10.91), and compare with the exact solution; and
  - b) using the upwind scheme (10.93), and compare with the exact solution.
- 10.6. Code the explicit MacCormack scheme with the *FF/BB* arrangement for the driven-cavity flow problem as described in Section 10.5. Compute the flow field at  $\text{Re} = 100$  and  $400$ , and explore effects of Mach number and the stability condition (10.110).

## Literature Cited

- Brooks, A. N., & Hughes, T. J. R. (1982). Streamline-upwinding/Petrov-Galerkin formulation for convection dominated flows with particular emphasis on incompressible Navier-Stokes equation. *Comput. Methods Appl. Mech. Engrg.*, 30, 199–259.
- Chorin, A. J. (1967). A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2, 12–26.
- Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22, 745–762.
- Fletcher, C. A. J. (1988). *Computational Techniques for Fluid Dynamics, I—Fundamental and General Techniques, and II—Special Techniques for Different Flow Categories*. New York: Springer-Verlag.

- Franca, L. P., Frey, S. L., & Hughes, T. J. R. (1992). Stabilized finite element methods: I. Application to the advective-diffusive model. *Comput. Methods Appl. Mech. Engrg.*, 95, 253–276.
- Franca, L. P., & Frey, S. L. (1992). Stabilized finite element methods: II. The incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 99, 209–233.
- Ghia, U., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, 48, 387–411.
- Glowinski, R. (1991). *Finite element methods for the numerical simulation of incompressible viscous flow, introduction to the control of the Navier-Stokes equations*. In: *Lectures in Applied Mathematics*, Vol. 28. Providence, RI: American Mathematical Society. 219–301.
- Gresho, P. M. (1991). Incompressible fluid dynamics: Some fundamental formulation issues. *Annu. Rev. Fluid Mech.*, 23, 413–453.
- Harlow, F. H., & Welch, J. E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8, 2182–2189.
- Hou, S., Zou, Q., Chen, S., Doolen, G. D., & Cogley, A. C. (1995). Simulation of cavity flow by the lattice Boltzmann method. *J. Comp. Phys.*, 118, 329–347.
- Hughes, T. J. R. (1987). *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*. Englewood Cliffs, NJ: Prentice-Hall.
- Intel Corporation. (2003). Moore's Law and Intel Innovation: Advancing silicon technology. <<http://www.intel.com/about/companyinfo/museum/exhibits/moore.htm>>. Accessed 10/4/2011.
- MacCormack, R. W. (1969). *The effect of viscosity in hypervelocity impact cratering*. Cincinnati, Ohio: AIAA Paper. 69–354.
- Marchuk, G. I. (1975). *Methods of Numerical Mathematics*. New York: Springer-Verlag.
- Noye, J. (Ed.), (1983). Chapter 2 in *Numerical Solution of Differential Equations*. Amsterdam: North-Holland.
- Oden, J. T., & Carey, G. F. (1984). *Finite Elements: Mathematical Aspects*, Vol. IV. Englewood Cliffs, NJ: Prentice-Hall.
- Perrin, A., & Hu, H. H. (2006). An explicit finite-difference scheme for simulation of moving particles. *J. Comput. Phys.*, 212, 166–187.
- Peyret, R., & Taylor, T. D. (1983). *Computational Methods for Fluid Flow*. New York: Springer-Verlag.
- Richtmyer, R. D., & Morton, K. W. (1967). *Difference Methods for Initial-Value Problems*. New York: Interscience.
- Saad, Y. (1996). *Iterative Methods for Sparse Linear Systems*. Boston: PWS Publishing Company.
- Tannehill, J. C., Anderson, D. A., & Pletcher, R. H. (1997). *Computational Fluid Mechanics and Heat Transfer*. Washington, DC: Taylor & Francis.
- Temam, R. (1969). Sur l'approximation des équations de Navier-Stokes par la méthode de pas fractionnaires. *Archiv. Ration. Mech. Anal.*, 33, 377–385.
- Tezduyar, T. E. (1992). In J. W. Hutchinson, & T. Y. Wu (Eds.), *Advances in Applied Mechanics. Stabilized Finite Element Formulations for Incompressible Flow Computations*, Vol. 28 (pp. 1–44). New York: Academic Press.
- Wen, C. Y., & Lin, C. Y. (2001). Two dimensional vortex shedding of a circular cylinder. *Phys. Fluids*, 13, 557–560.
- Yanenko, N. N. (1971). *The Method of Fractional Steps*. New York: Springer-Verlag.

## Supplemental Reading

- Dennis, S. C. R., & Chang, G. Z. (1970). Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100. *J. Fluid Mech*, 42, 471–489.
- Sucker, D., & Brauer, H. (1975). Fluidodynamik bei der angeströmten Zylindern. *Wärme-Stoffübertrag*, 8, 149.
- Takami, H., & Keller, H. B. (1969). Steady two-dimensional viscous flow of an incompressible fluid past a circular cylinder. *Phys. Fluids*, 12(Suppl.II), II-51–II-56.