

- Douglas, J., 1962, Alternating direction methods for three space variables. *Numerische Mathematik*, **4**, 41–63.
- Douglas, J., and Gunn, J. E., 1964, A general formulation of alternating direction methods. *Numerische Mathematik*, **6**, 428–53.
- DuFort, E. C., and Frankel, S. P., 1953, Stability conditions in the numerical treatment of parabolic differential equations. *Math. Tables and Other Aids to Computation*, **7**, 135–52.
- Ferziger, J. H., 1981, *Numerical Methods for Engineering Application*. Wiley.
- Fletcher, C. A. J., 1988, *Computational Techniques for Fluid Dynamics*. 2 volumes. Springer-Verlag.
- Gourlay, A. R., 1970, Hopscotch: a fast second-order partial differential equation solver. *J. Inst. Maths Applics.* **6**, 375–90.
- Greig, I. S., and Morris, J. L., 1976, A hopscotch method for the Korteweg-de Vries equation. *J. Comp. Phys.* **20**, 64–80.
- Hindmarsh, A. C., Gresho, P. M., and Griffiths, D. F., 1984, The stability of explicit Euler time-integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation. *Int. J. Num. Meth. Fluids*, **4**, 853–97.
- Hirsch, C., 1988, *Numerical Computation of Internal and External Flows*. 2 volumes. Wiley.
- Hoffman, J. D., 1992, *Numerical Methods for Engineers and Scientists*. McGraw-Hill.
- Hoffmann, K. A., and Chiang, S. T., 1993, *Computational Fluid Dynamics for Engineers*, Vols. I and II, Engineering Education System, Wichita, Kansas 67208-1078.
- Lax, P. D., 1954, Weak solutions of nonlinear hyperbolic equations and their computation. *Comm. Pure Appl. Math.* **7**, 159–93.
- Lax, P. D., and Richtmyer, R. D., 1956, Survey of the stability of linear finite difference equations. *Comm. Pure Appl. Math.* **9**, 267–93.
- Lax, P., and Wendroff, B., 1960, Systems of conservation laws. *Comm. Pure Appl. Math.* **13**, 217–37.
- Lee, M., 1962, Alternating direction methods for hyperbolic differential equations. *J. Soc. Indust. Appl. Math.* **10**, 611–16.
- Leonard, B. P., Leschziner, M. A., and McGuirk, J., 1978, Third-order finite-difference method for steady two-dimensional convection. *Num. Meth. in Laminar and Turbulent Flow*, 807–19.
- MacCormack, R. W., 1969, *The effect of viscosity in hypervelocity impact cratering*. AIAA paper No. 69-354.
- Mitchell, A. R., 1969, *Computational Methods in Partial Differential Equations*. Wiley.
- Mitchell, A. R., Griffiths, D. F., 1980, *The Finite Difference Method in Partial Differential Equations*. Wiley.
- Peaceman, D. W., and Rachford, H. H., 1955, The numerical solution of parabolic and elliptic differential equations. *J. Soc. Indust. Appl. Math.* **3**, 28–41.
- Peyret, R., and Taylor, T. D., 1983, *Computational Methods for Fluid Flow*. Springer-Verlag.
- Richtmyer, R. D., 1963, *A survey of difference methods for non-steady fluid dynamics*. NCAR Technical Notes 63-2.
- Richtmyer, R. D., and Morton, K. W., 1967, *Difference Methods for Initial-Value Problems*. Interscience.
- Sod, G. A., 1985, *Numerical Methods in Fluid Dynamics. Initial and Initial Boundary-Value Problems*. Cambridge University Press.
- Warming, R. F., and Hyett, B. J., 1974, The modified equation approach to the stability and accuracy analysis of finite-difference methods. *J. Comp. Phys.* **14**, 159–79.
- Wilkinson, J. H., 1965, *The Algebraic Eigenvalue Problem*. Oxford University Press.
- Yanenko, N. N., 1970, *The Method of Fractional Steps*. Springer-Verlag.

# CHAPTER 13

## Finite-Difference Methods for Incompressible Newtonian Flow

Having discussed finite-difference methods for computing numerical solutions to the convection-diffusion equation in its general form, we proceed to develop corresponding methods for solving the equations of steady and unsteady incompressible Newtonian flow. The set of governing equations includes the Navier-Stokes equation and the continuity equation, and the primary unknowns are the velocity and the pressure. We recall, however, that a general rotational flow may also be described and therefore computed in terms of the secondary variables discussed in Chapter 2, including the vorticity, the stream functions, and the vector potential.

Considering the evolution of an unsteady flow, we regard the Navier-Stokes equation as an evolution equation for the velocity, providing us with the rate of change of the velocity at a particular point in the flow in terms of the instantaneous velocity and pressure. We then note that if the pressure gradient were absent, the simplified evolution equation would be identical to the nonlinear convection-diffusion equation, and could therefore be integrated in time using the finite-difference methods discussed in Chapter 12. Unfortunately, as discussed in Section 9.1, an evolution equation for the pressure is not available in an explicit form. In its place we have the restriction of incompressibility, which requires that the pressure evolve so as to ensure that the rate of expansion vanish and the velocity field remain solenoidal at all times. As we saw in Section 9.1, the restriction of incompressibility may be expressed in terms of a Poisson equation either for the pressure or for the rate of change of the pressure with a time-dependent forcing function. These equations determine the evolution of the pressure in an implicit fashion.

Computing the evolution of an incompressible Newtonian flow is thus distinguished by the necessity to solve, simultaneously, a parabolic differential equation in time, which is the equation of motion, and an elliptic differential equation in space, which is the Poisson equation for the pressure or for the rate of change of the pressure.

It is instructive to note at this point that the continuity equation for a compressible fluid has the form of an evolution equation for the density that is related to the pressure by means of an equation of state. Since the full set of governing equations is parabolic in time, it may be integrated using a standard time-marching method for initial-value problems. Shock waves aside, computing the evolution of a compressible flow is in this respect more straightforward than computing the evolution of an incompressible flow.

An additional concern that arises in computing the structure of a steady flow or the evolution of an unsteady flow in terms of the velocity and the pressure, pertains to the derivation and numerical implementation of boundary conditions for the pressure. In the vast majority of fluid-dynamics applications, these are not available in the statement of the problem, but must be derived from the equation of motion subject to the required boundary conditions for the velocity or surface stress. We shall see in this chapter that the accurate implementation of the derived pressure boundary conditions requires careful attention.

There are a number of finite-difference procedures for solving the equations of steady and unsteady incompressible Newtonian flow, and a choice must be made according to the tolerated level of programming complexity and available computational resources. In the present chapter we shall outline the fundamental principles that underlie several alternative procedures, and shall

discuss the basic steps involved in their numerical implementation. Extensions and discussions of specific issues and specialized topics may be found in the cited references, as well as in general reviews and monographs on finite-difference methods in fluid dynamics including those by Orszag and Israeli (1974), Cebeci (1982), Roach (1982), Peyret and Taylor (1983), Anderson, Tannehill, and Pletcher (1984), Hirsch (1988), Fletcher (1988), Gresho (1991), and Hoffmann and Chiang (1993). Numerical methods for interfacial and free-surface flows are discussed in a comprehensive review article by Floryan and Rasmussen (1989).

### 13.1 METHODS BASED ON THE VORTICITY TRANSPORT EQUATION

We begin by discussing a class of methods for computing the structure of a steady flow or the evolution of an unsteady flow on the basis of the vorticity transport equation. The numerical procedure involves computing the evolution of the vorticity field, and obtaining the simultaneous evolution of the velocity field by inverting the fundamental equation that relates the vorticity to the velocity,  $\omega = \nabla \times \mathbf{u}$ , subject to the continuity equation. One advantage of this approach is that the pressure does not have to be considered, which results in computational efficiency and ease of implementation. One inevitable concern is the need to derive boundary conditions for the vorticity.

The present class of methods may be regarded as extensions of the vortex methods for inviscid fluids or slightly viscous fluids discussed in Chapter 11. The distinguishing feature of the vortex methods is that the velocity field is obtained from the vorticity field most efficiently using the Biot-Savart integral or a related contour integral. For viscous flow, the support of the vorticity is not compact, and it is more expedient to recover the velocity from the vorticity by solving differential equations using finite-difference methods.

We begin the discussion by presenting the classical stream function–vorticity formulation for two-dimensional flow, and continue to address more general formulations for three-dimensional flow.

#### Stream Function–Vorticity Formulation for Two-Dimensional Flow

For two-dimensional flow in the  $xy$  plane, solving for the velocity in terms of the vorticity is done with the least amount of computational effort by introducing the stream function  $\psi$ . The two components of the velocity in the  $x$  and  $y$  direction are given by  $u = \partial\psi/\partial y$  and  $v = -\partial\psi/\partial x$ , and the vorticity is  $\omega = \omega\mathbf{k}$ , where  $\mathbf{k}$  is the unit vector along the  $z$  axis, and

$$\nabla^2\psi = -\omega \quad (13.1.1)$$

The computations proceed according to the two fundamental steps of vortex methods. In the first step, we compute the evolution of the vorticity field using the simplified form of the vorticity transport equation for two-dimensional flow, written in the stream function–vorticity form as

$$\frac{\partial\omega}{\partial t} + \frac{\partial\psi}{\partial y}\frac{\partial\omega}{\partial x} - \frac{\partial\psi}{\partial x}\frac{\partial\omega}{\partial y} = v\nabla^2\omega \quad (13.1.2)$$

The sum of the second and third terms on the left-hand side of Eq. (13.1.2) is sometimes designated as the Jacobian  $J(\omega, \psi)$ . In the second step, we update the stream function by solving Poisson's equation (13.1.1) for  $\psi$  in terms of  $\omega$ . Boundary conditions are required during both the integration of Eq. (13.1.2) and the inversion of Eq. (13.1.1).

It is instructive to note that the absence of an explicit evolution equation for the pressure in the original system of governing equations is reflected in the absence of an explicit evolution equation for the stream function.

We return to emphasize the lack of a need to compute the pressure. If the instantaneous pressure field is desired, it may be computed a posteriori by solving a Poisson equation that emerges by taking the divergence of the Navier-Stokes equation and using the continuity equation to obtain

$$\nabla^2 P = 2\rho \left[ \frac{\partial^2\psi}{\partial x^2} \frac{\partial^2\psi}{\partial y^2} - \left( \frac{\partial^2\psi}{\partial x \partial y} \right)^2 \right] \quad (13.1.3)$$

Boundary conditions for the pressure are derived by applying the equation of motion at the boundaries, projecting it onto either the normal or tangential unit vector, and then simplifying the various terms taking into account the boundary conditions for the velocity, as will be discussed in Section 13.2.

#### Flow in a rectangular cavity

To illustrate the practical implementation of the finite-difference method, we consider the classical problem of flow in a rectangular cavity driven by a lid that translates parallel to itself with a generally time-dependent velocity  $V(t)$ , as illustrated in Figure 13.1.1.

The no-penetration condition requires that the component of the velocity normal to each one of the four walls vanish. In terms of the stream function, we obtain the equivalent statement

$$\psi = c \quad \text{over all walls} \quad (13.1.4)$$

where  $c$  is an arbitrary constant that, for simplicity, will be set equal to zero. The no-slip boundary condition requires that the tangential component of the velocity over the bottom, left, and right

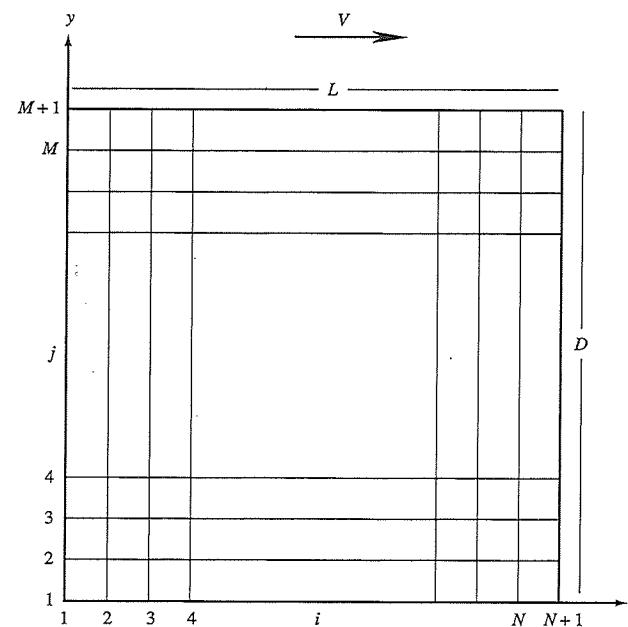


Figure 13.1.1 A non-staggered finite-difference grid for computing the two-dimensional flow in a rectangular cavity driven by a moving lid. The two components of the velocity and the pressure are defined at the grid nodes.

walls vanish, whereas the tangential component at the upper wall be equal to  $V(t)$ . In terms of the stream function, we obtain the statements

$$\begin{aligned}\frac{\partial \psi}{\partial y} &= 0 && \text{at the bottom}, & \frac{\partial \psi}{\partial x} &= 0 && \text{at the sides} \\ \frac{\partial \psi}{\partial y} &= V && \text{at the lid}\end{aligned}\quad (13.1.5)$$

Based on these boundary conditions for the velocity, we derive simplified expressions for the boundary values of the vorticity in terms of the stream function. Beginning with Eq. (13.1.1) and noting, for example, that at the bottom wall  $\partial^2 \psi / \partial x^2 = -\partial v / \partial x = 0$  because of the no-penetration condition, we find

$$\begin{aligned}\omega &= -\frac{\partial^2 \psi}{\partial y^2} && \text{at the top and bottom walls} \\ \omega &= -\frac{\partial^2 \psi}{\partial x^2} && \text{at the side walls}\end{aligned}\quad (13.1.6)$$

which are restatements of the first of Eqs. (3.6.14).

To implement a finite-difference method, we introduce a two-dimensional grid of size  $N$  by  $M$ , as illustrated in Figure 13.1.1. For simplicity, we have assumed that the grid lines are evenly spaced, which means that  $\Delta x$  and  $\Delta y$  are uniform but not necessarily equal to each other. This stipulation may be relaxed with straightforward changes in the finite-difference equations. We then assign to the stream function  $\psi$  and vorticity  $\omega$  discrete values at all internal and boundary grid points, and replace the differential equations (13.1.1) and (13.1.2) with difference equations as discussed in Chapter 12. The subsequent strategy of computation depends upon whether we wish to compute a steady or an unsteady flow.

### Steady Flow

Two classes of distinct but somewhat related methods are available for computing a steady flow. The first class proceeds by solving the equations of steady flow using iterative methods. The second class proceeds by computing the solution of a fictitious transient flow problem that is governed by a modified set of differential equations, from a given initial condition up to the steady state. The solution of the modified problem at the steady state satisfies the original differential equations of steady two-dimensional incompressible Newtonian flow.

#### Direct approach

In one version of the direct approach, we regard the governing equations (13.1.1) and (13.1.2) as a coupled, nonlinear system of Poisson equations for  $\psi$  and  $\omega$ , and recast them into the form

$$\nabla^2 \psi = -\omega \quad (13.1.7)$$

$$\nabla^2 \omega = \frac{1}{\nu} \left( u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} \right) \quad (13.1.8)$$

In the special case of Stokes flow, the right-hand side of Eq. (13.1.8) vanishes, yielding Laplace's equation for the vorticity. Equation (13.1.7) then shows that the stream function satisfies the bi-harmonic equation as discussed in Section 6.1.

The computational algorithm in the general case of flow at finite Reynolds numbers involves the following steps:

1. Guess the vorticity distribution.
2. Solve Poisson's equation (13.1.7) for the stream function. For boundary conditions, we have the choice between the Dirichlet boundary condition that specifies the boundary distribution

of the stream function, the Neumann boundary condition that specifies the boundary distribution of the normal derivative of the stream function, which is equal to the tangential component of the velocity, or a combination of the Dirichlet and Neumann boundary conditions over the different boundaries.

Use of the Neumann boundary condition over all boundaries is not appropriate, for a solution to the Poisson equation will exist only when the following compatibility condition is fulfilled

$$\int_{\text{Walls}} \nabla \psi \cdot \mathbf{n} \, dl = \int_{\text{Flow}} \omega \, dA \quad (13.1.9)$$

where  $\mathbf{n}$  is the unit normal vector pointing into the flow. Even though Eq. (13.1.9) may be fulfilled for a certain fortuitous guess of the vorticity distribution in step 1, the singular nature of the linear system of equations that arises from the finite-difference discretization of Eq. (13.1.7) will present additional complications. We thus prefer to enforce the Dirichlet condition over all boundaries, expressed by Eq. (13.1.4).

3. Compute the right-hand side of Eq. (13.1.8) and the boundary values of the vorticity using the required boundary conditions for the velocity, and solve Poisson's equation (13.1.8) for the vorticity.
4. Check to see whether the computed vorticity agrees with the current vorticity at all grid points, and if it does not, replace the current with the computed vorticity and return to step 2.

The details of the numerical implementation of the various steps will be now discussed with reference to the flow in a cavity illustrated in Figure 13.1.1.

1. Assign initial values for the stream function to all  $(N+1) \times (M+1)$  internal and boundary grid points, and for the vorticity to all  $NM$  internal grid points. A simple choice is to set both the stream function and vorticity equal to zero.
2. Solve Poisson's equation (13.1.7) subject to the Dirichlet boundary condition (13.1.4) over all four walls. Since the vorticity, which is the forcing function of Poisson's equation (13.1.7), is only an approximation to the exact solution, an accurate solution for  $\psi$  at this stage is not warranted. To reduce the computational effort, we solve Poisson's equation using an iterative method and carry out only a small number of iterations.

One way of carrying out the iterations is to introduce a fictitious unsteady diffusion-reaction problem with a source term that is equal to  $\omega$  and diffusivity that is equal to  $\rho_1 \Delta x^2 / \Delta t$ , where  $\rho_1$  is a dimensionless constant, in which case the diffusion numbers in the  $x$  and  $y$  directions are  $\alpha_x = \rho_1$  and  $\alpha_y = \rho_1 \beta^2$ , where  $\beta = \Delta x / \Delta y$ . The solution at steady state satisfies Eq. (13.1.7). Implementing the FTCS discretization, which involves using the five-point formula to approximate the Laplacian, yields

$$\begin{aligned}\psi_{i,j}^{(k+1)} &= \psi_{i,j}^{(k)} + \rho_1 \left[ \psi_{i+1,j}^{(k)} - 2\psi_{i,j}^{(k)} + \psi_{i-1,j}^{(k)} \right. \\ &\quad \left. + \beta^2 (\psi_{i,j+1}^{(k)} - 2\psi_{i,j}^{(k)} + \psi_{i,j-1}^{(k)}) + \Delta x^2 \omega_{i,j} \right] \quad (13.1.10)\end{aligned}$$

where  $k$  plays the role of an iteration number (see first entry of Table 12.3.1). The von Neumann stability analysis discussed in Section 12.3 shows that the iterations will converge provided that  $\rho_1 \leq 1/[2(1 + \beta^2)]$ . When the iterations are executed for the first time, the initial values  $\psi_{i,j}^{(0)}$  are set equal to those guessed in step 1. After the iterations converge, the solution will be second-order accurate in  $\Delta x$  and  $\Delta y$ . Alternatives to the FTCS iterative scheme (13.1.10) are the Gauss-Siedel and successive over-relaxation schemes, as well as their SOR versions shown in Table 12.3.1.

3. Use Eqs. (13.1.6) to compute the vorticity at the boundary grid points by means of one-sided finite differences.

To compute the vorticity at a grid point that lies on the upper wall, we expand the stream function in a Taylor series with respect to  $y$  about a grid point that lies on the lid, and evaluate the series at the  $M$ th layer to obtain

$$\psi_{i,M} = \psi_{i,M+1} + \left( \frac{\partial \psi}{\partial y} \right)_{i,M+1} (-\Delta y) + \frac{1}{2} \left( \frac{\partial^2 \psi}{\partial y^2} \right)_{i,M+1} (-\Delta y)^2 + \dots \quad (13.1.11)$$

Using the no-slip boundary condition and the first of Eqs. (13.1.6), and rearranging, we obtain

$$\omega_{i,M+1} = 2 \frac{\psi_{i,M+1} - \psi_{i,M}}{\Delta y^2} - 2 \frac{V}{\Delta y} \quad (13.1.12)$$

which is first-order accurate in  $\Delta y$ . Working in a similar manner for the bottom and side walls, we derive the analogous expressions

$$\begin{aligned} \omega_{i,1} &= 2 \frac{\psi_{i,1} - \psi_{i,2}}{\Delta y^2}, & \omega_{1,j} &= 2 \frac{\psi_{1,j} - \psi_{2,j}}{\Delta x^2} \\ \omega_{N+1,j} &= 2 \frac{\psi_{N+1,j} - \psi_{N,j}}{\Delta x^2} \end{aligned} \quad (13.1.13)$$

The boundary condition (13.1.4) allows us to set  $\psi_{i,1}, \psi_{i,M+1}, \psi_{1,j}, \psi_{N+1,j}$  equal to zero.

To improve the accuracy to second order, we expand the stream function in a Taylor series about a grid point on the upper wall, evaluate the series at the two layers that are adjacent to the wall, and maintain terms up to third order to find

$$\begin{aligned} \psi_{i,M} &= \psi_{i,M+1} + \left( \frac{\partial \psi}{\partial y} \right)_{i,M+1} (-\Delta y) + \frac{1}{2} \left( \frac{\partial^2 \psi}{\partial y^2} \right)_{i,M+1} (-\Delta y)^2 \\ &\quad + \frac{1}{6} \left( \frac{\partial^3 \psi}{\partial y^3} \right)_{i,M+1} (-\Delta y)^3 + \dots \end{aligned} \quad (13.1.14)$$

$$\begin{aligned} \psi_{i,M-1} &= \psi_{i,M+1} + \left( \frac{\partial \psi}{\partial y} \right)_{i,M+1} (-2 \Delta y) + \frac{1}{2} \left( \frac{\partial^2 \psi}{\partial y^2} \right)_{i,M+1} (-2 \Delta y)^2 \\ &\quad + \frac{1}{6} \left( \frac{\partial^3 \psi}{\partial y^3} \right)_{i,M+1} (-2 \Delta y)^3 + \dots \end{aligned} \quad (13.1.15)$$

Combining these equations to eliminate the third derivative of the stream function, solving for the second derivative, using the boundary condition in Eqs. (13.1.5), and taking into account Eq. (13.1.6), we find

$$\omega_{i,M+1} = \frac{7\psi_{i,M+1} - 8\psi_{i,M} + \psi_{i,M-1}}{2 \Delta y^2} - 3 \frac{V}{\Delta y} \quad (13.1.16)$$

Working in a similar manner for the bottom and side walls we find

$$\begin{aligned} \omega_{i,1} &= \frac{7\psi_{i,1} - 8\psi_{i,2} + \psi_{i,3}}{2 \Delta y^2}, & \omega_{1,j} &= \frac{7\psi_{1,j} - 8\psi_{2,j} + \psi_{3,j}}{2 \Delta x^2} \\ \omega_{N+1,j} &= \frac{7\psi_{N+1,j} - 8\psi_{N,j} + \psi_{N-1,j}}{2 \Delta x^2} \end{aligned} \quad (13.1.17)$$

4. Differentiate the stream function to compute the velocity at the internal grid points subject to the boundary values (13.1.4). Differentiate the vorticity to compute the right-hand side

- of Eq. (13.1.8) at the internal grid points subject to the boundary values computed from Eqs. (13.1.12) and (13.1.13) or Eqs. (13.1.16) and (13.1.17). For convenience, denote the right-hand side of Eq. (13.1.8) at the  $i,j$  grid point by  $N_{i,j}$ , where  $N$  stands for *nonlinear*.
5. Solve Poisson's equation (13.1.8) subject to the Dirichlet boundary conditions computed in step 3. This may be done in an iterative manner according to the FTCS algorithm

$$\begin{aligned} \omega_{i,j}^{(k+1)} &= \omega_{i,j}^{(k)} + \rho_2 \left[ \omega_{i+1,j}^{(k)} - 2\omega_{i,j}^{(k)} + \omega_{i-1,j}^{(k)} \right. \\ &\quad \left. + \beta^2 (\omega_{i,j+1}^{(k)} - 2\omega_{i,j}^{(k)} + \omega_{i,j-1}^{(k)}) - \Delta x^2 N_{i,j} \right] \end{aligned} \quad (13.1.18)$$

discussed in step 2, where  $\rho_2$  is the diffusion number in the  $x$  direction, and carry out a small number of iterations. The values of the forcing function at the internal grid points  $N_{i,j}$  are available from step 4. As in step 2, the algorithm (13.1.18) may be replaced with one of the algorithms shown in Table 12.3.1.

6. If the vorticity  $\omega$  computed in step 5 does not agree with that previously available, return to step 2 and repeat the computations with the new grid values of  $\omega$ . This outer iteration is terminated when the absolute value of the difference of the vorticity between two successive iterations at each grid point becomes less than a preestablished threshold value  $\varepsilon$ , or the sum of the absolute values of the differences in the vorticity over all internal  $NM$  grid points becomes less than  $NM\varepsilon$ .

One noteworthy feature of this procedure is that the corner grid points do not enter the computations, and this eliminates ambiguities stemming from the fact that the velocity undergoes a discontinuity at the upper corner points. This discontinuity may cause local oscillations and decelerate the local convergence, but does not have a deleterious effect on the global convergence of the method.

The individual steps of the numerical procedure may be modified and improved in several ways (see, for instance, Gupta, 1991). While the basic philosophy of the algorithm remains unchanged, the rate of convergence of the iterations does depend on the details of the particular implementation (Israeli, 1972). In one variation of the method, instead of iterating on Poisson's equation for the vorticity in step 5, we iterate on the full convection-diffusion equation (13.1.8), which means that we recompute the nonlinear term  $N_{i,j}$  after each iteration using the updated values of the vorticity. These iterations may be done on the basis of an explicit or implicit finite-difference method for the convection-diffusion equation in two dimensions discussed in Section 12.7. At high Reynolds numbers, the flow near the center of the cavity is dominated by convection, and using upwind differencing improves the numerical stability.

Streamline patterns for flow in a square cavity are illustrated in Figure 13.1.2 for three values of the Reynolds number  $Re = VL/\nu$ . When  $Re = 1$ , we obtain a nearly creeping flow, and the streamline pattern is almost symmetric with respect to the midplane of the cavity. As the Reynolds number is increased, the recirculating eddy becomes unsymmetric and shifts toward the upper right corner. Small viscous eddies are always present at the two bottom corners. At even higher Reynolds numbers, the flow is composed of a central vortex with nearly uniform vorticity, and boundary layers lining the walls.

The method of computing the boundary values of the vorticity described in step 3 works well in most cases but it has been the subject of criticism (Gresho, 1991). It has been argued that it is not proper to specify the boundary values of the vorticity in an explicit manner, but instead, the boundary distribution of the vorticity must arise in an implicit manner as part of the solution, using the natural boundary conditions for the velocity or traction. This issue will be discussed further at the end of the present section in the more general context of three-dimensional flow.

(a)

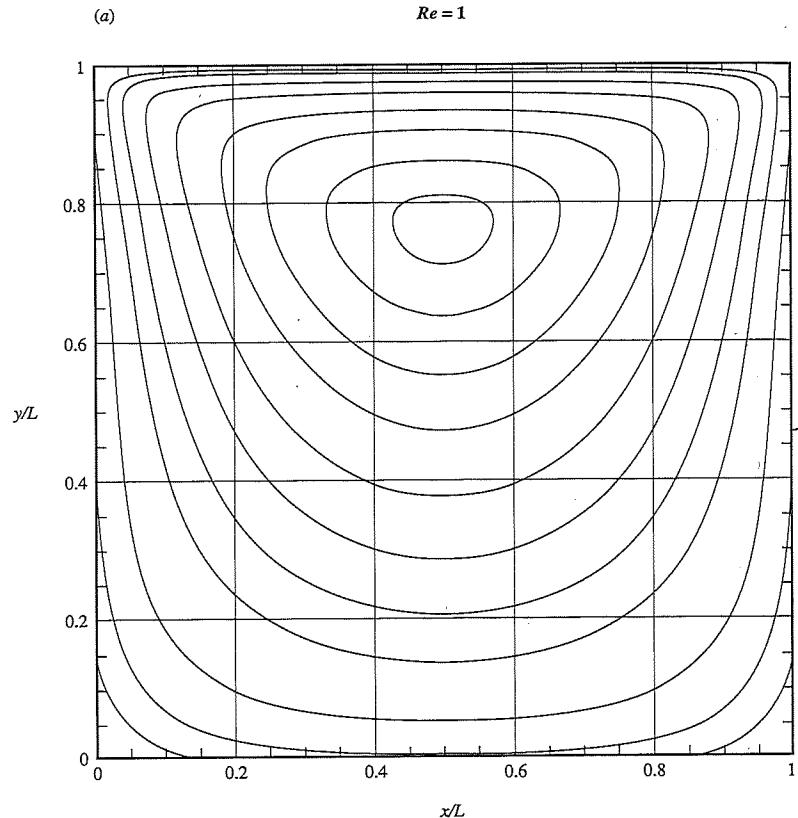
 $Re = 1$ 

Figure 13.1.2 Streamline patterns of steady flow in a square cavity driven by a translating lid, computed using the direct method discussed in the text for three values of the Reynolds number (a)  $Re = VL/\nu = 1$ . (continued)

#### *Method of modified dynamics or false transients*

A distinguishing part of the direct approach is the solution of a Poisson equation for the stream function, which reflects the elliptic nature of the equations governing the structure of a steady flow. We saw that one way to perform the associated inner iterations is to introduce a fictitious unsteady diffusion-reaction problem and then implement the explicit FTCS discretization. This observation suggests reformulating the problem by maintaining the unsteady vorticity transport equation (13.1.2) and transforming Eq. (13.1.1) into the following evolution equation for the stream function

$$\frac{\partial \psi}{\partial t} = \alpha(\nabla^2 \psi + \omega) \quad (13.1.19)$$

where  $\alpha$  is a positive constant that is a free parameter of the numerical method (Mallinson and de Vahl Davis, 1973). The idea is to compute the evolution of the flow from an arbitrary initial

(b)

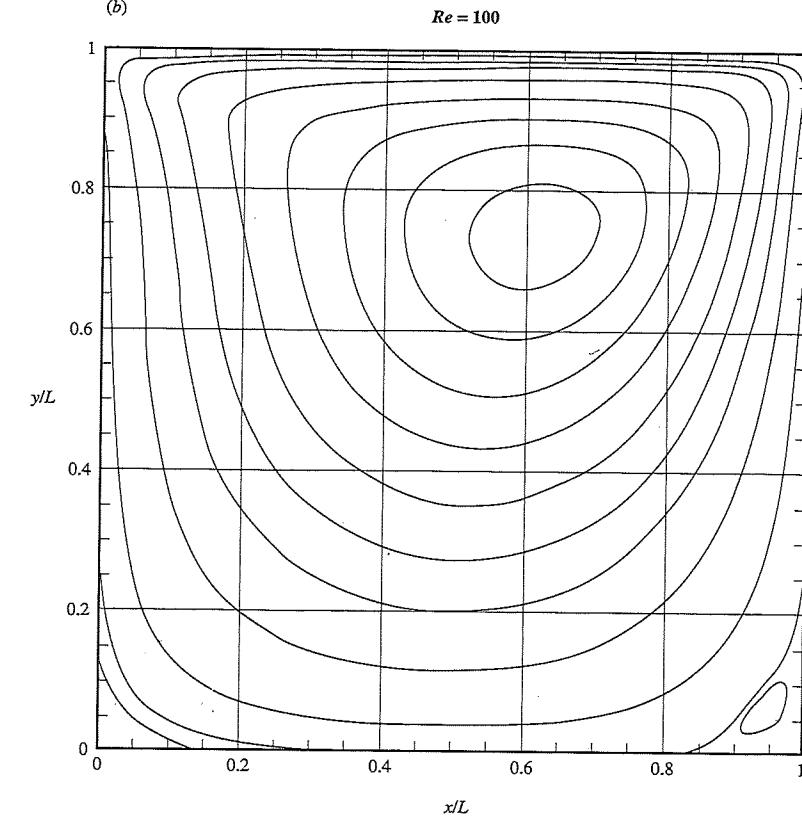
 $Re = 100$ 

Figure 13.1.2 (continued) Streamline patterns of steady flow in a square cavity driven by a translating lid, computed using the direct method discussed in the text for three values of the Reynolds number (b)  $Re = 100$ . (continued)

condition on the basis of Eqs. (13.1.2) and (13.1.19) until a steady state is established; at that point the solution will also satisfy the original Equations (13.1.7) and (13.1.8). In the implementation of the method, the right-hand side of Eq. (13.1.2) is also multiplied with a positive factor in order to expedite the convergence. The critical advantage of this approach is that the governing equations become parabolic in time, and this allows us to use time-marching methods similar to those developed in Chapter 12 for problems of convection-diffusion.

For the problem of flow in a cavity illustrated in Figure 13.1.1, the method of modified dynamics is implemented according to the following steps:

1. Assign initial values to the stream function and vorticity at all internal and boundary grid points; a simple choice is to set them both equal to zero.
2. Differentiate the stream function to compute the two components of the velocity at all internal grid points.
3. Compute the vorticity at the boundary grid points as in step 3 of the direct approach.

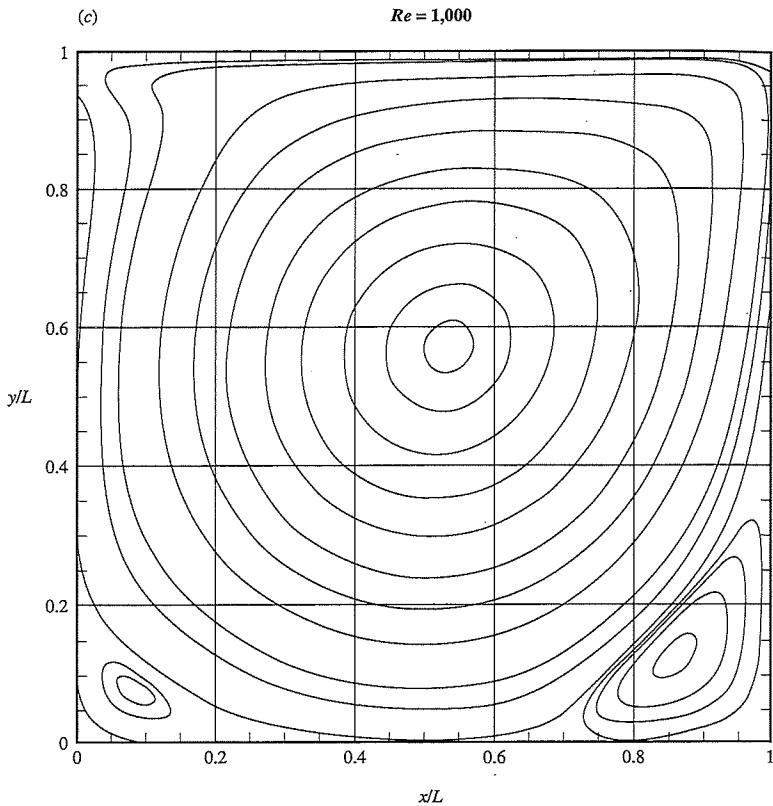


Figure 13.1.2 (continued) Streamline patterns of steady flow in a square cavity driven by a translating lid, computed using the direct method discussed in the text for three values of the Reynolds number (c)  $Re = 1,000$ .

4. Advance the vorticity at all internal grid points on the basis of Eq. (13.1.2) using, for instance, the ADI method for the convection-diffusion equation described in Section 12.7, while maintaining the vorticity at the boundary grid points constant.
5. Advance the stream function at all internal grid points on the basis of Eq. (13.1.19) using, for instance, the ADI method for the convection-diffusion equation described in Section 12.7, subject to the boundary condition (13.1.4).
6. Return to step 2 and repeat the computations for another time step.

#### Unsteady Flow

To compute the evolution of an unsteady flow, we follow a procedure that combines certain features of the direct approach and certain features of the method of modified dynamics for steady flow. The algorithm involves computing the evolution of the vorticity field using Eq. (13.1.2), while obtaining the simultaneous evolution of the velocity field on the basis of the stream function using Eq. (13.1.1).

A simple strategy for computing the evolution of the flow in a cavity when the lid begins translating suddenly at a constant velocity  $V$  proceeds according to the following steps:

1. Set the stream function and velocity equal to zero at all internal and boundary grid points at the initial instant. Then set the  $x$  component of the velocity at the grid points on the lid equal to  $V$ .
2. Differentiate the velocity to compute the vorticity. For the internal grid points use central differences. For the grid points that lie on the lid use the first of Eqs. (13.1.6), take into account the boundary conditions, and apply the second-order backward difference for the first derivative of  $u$  with respect to  $y$  to obtain

$$\omega_{i,M+1} = \frac{-3V + 4u_{i,M} - u_{i,M-1}}{2\Delta y} \quad (13.1.20)$$

(Table B.5.1). For the grid points that lie at the bottom and side walls, use the corresponding second-order finite-difference formulae

$$\begin{aligned} \omega_{i,1} &= \frac{-4u_{i,2} + u_{i,3}}{2\Delta y}, & \omega_{1,j} &= \frac{4v_{2,j} - v_{3,j}}{2\Delta x} \\ \omega_{N+1,j} &= \frac{-4v_{N,j} + v_{N-1,j}}{2\Delta x} \end{aligned} \quad (13.1.21)$$

3. Integrate Eq. (13.1.2) to compute the vorticity at the next time level at all internal grid points subject to the boundary conditions expressed by Eqs. (13.1.20) and (13.1.21) using, for example, an explicit method such as the FTCS method. At high Reynolds numbers, use upwind differencing.
4. Solve Poisson's equation (13.1.1) for the stream function at the next time level subject to the Dirichlet boundary condition (13.1.4).
5. Differentiate the stream function to compute the velocity at the next time level at all internal grid points.
6. Return to step 2 and repeat the computations for another time step.

To improve the temporal accuracy and enhance the stability of the method, one may update the vorticity using an implicit or a semi-implicit method such as the ADI method. In a simple implementation of the ADI method, we maintain the convection velocity constant during both substeps, equal to its value at the beginning of the first substep. In a more advanced implementation, we solve Poisson's equation for the intermediate stream function after completion of the first substep, and then set the convection velocity in the second substep equal to that computed by differentiating the intermediate stream function as described in step 5. Since, however, the convection velocity is kept constant during each step or substep, equal to its value at the beginning of the step or substep, the overall accuracy of the method will still be of first order in time.

To achieve second-order accuracy, we use the ADI method with time-dependent velocities described in Eqs. (12.7.6) and (12.7.7). Collecting the values of the vorticity at all grid points into the vector  $\omega$ , we obtain the two ADI equations written in the symbolic form

$$\begin{aligned} A(u^{n-1}, u^n, u^{n+1}) \cdot \omega^{n+1/2} &= B(u^{n-1}, u^n, u^{n+1}) \cdot \omega^n \\ C(u^{n-1}, u^n, u^{n+1}) \cdot \omega^{n+1} &= D(u^{n-1}, u^n, u^{n+1}) \cdot \omega^{n+1/2} \end{aligned} \quad (13.1.22)$$

where  $A$ ,  $B$ ,  $C$ , and  $D$  are tridiagonal matrices that are functions of their arguments. Eqs. (13.1.22) replace the explicit FTCS equation in step 3. Steps 3, 4, and 5 described above are now combined to yield the following inner iterative loop:

- i. Guess the velocities  $u^{n+1}$  and solve the two tridiagonal systems of Eqs. (13.1.22) with boundary conditions given in Eqs. (13.1.20) and (13.1.21) for both  $\omega^{n+1/2}$  and  $\omega^{n+1}$ .
- ii. Execute steps 4 and 5.

- iii. Solve Eqs. (13.1.22) with the computed values of  $\omega^{n+1}$  or with a weighted average of the old and new values.

If the boundary values of the velocity change in time, we solve the first tridiagonal system of Eqs. (13.1.22) with boundary conditions  $\omega^{n+1/2} = \frac{1}{2}(\omega^n + \omega^{n+1})$ , where  $\omega^{n+1}$  has been approximated from the previous inner iteration. To accelerate the convergence, we replace the boundary conditions for  $\omega^{n+1}$  during the inner iterations with a weighted average of its old and new values. Further details on the implementation of this method are given by Peyret and Taylor (1983, p. 197).

### Methods for Three-Dimensional Flow

Algorithms based on the vorticity transport equation for three-dimensional flow involve the following two basic steps:

1. Compute the evolution of the vorticity field on the basis of the vorticity transport equation written in the conservative or Eulerian form

$$\frac{\partial \omega}{\partial t} + \nabla \times (\omega \times \mathbf{u}) = \nu \nabla^2 \omega \quad (13.1.23)$$

Taking the divergence of Eq. (13.1.23), we find that  $\nabla \cdot \omega$  satisfies the unsteady diffusion equation, and this guarantees that the computed vorticity field will be solenoidal provided that (1) it is solenoidal at the initial instant, and (2) its divergence vanishes over the boundaries of the flow at all times (see also Problem 13.1.4).

To integrate Eq. (13.1.23), we require boundary conditions for the vorticity. In the majority of numerical procedures, the boundary values of the vorticity emerge by applying the definition  $\omega = \nabla \times \mathbf{u}$  at or near the boundaries, and then simplifying them, taking into consideration the prescribed boundary conditions for the velocity. The numerical procedure is analogous to that involved in the stream function–vorticity formulation discussed earlier in this section. This approach guarantees that an initially solenoidal vorticity field will remain solenoidal at all times (Guj and Stella, 1993; Trujillo, 1994).

It has been argued, however, that it is not entirely appropriate to impose local boundary conditions for the vorticity in an explicit fashion, but instead, the boundary distribution of the vorticity must be computed as part of the solution, taking into consideration the boundary conditions for the velocity or surface stress (Gresho, 1991). Computational experiments have shown that computing, instead of imposing, boundary values for the vorticity enhances the stability of the numerical method, but this comes at the cost of increased programming complexity and computational effort.

Quartapelle and Valz-Gris (1981), in particular, replaced the boundary conditions for the vorticity with an integral constraint. For two-dimensional flow with homogeneous boundary conditions for the velocity, this constraint requires that the vorticity be orthogonal to all nonsingular harmonic functions defined in the domain of flow, that is, the integral of the vorticity multiplied by any nonsingular harmonic function over the area of the flow vanish. The implementation of this method is discussed by Quartapelle (1981) and by Anderson (1989) in the context of Chorin's vortex sheet method (Section 11.5).

When the flow is described in a noninertial frame of reference that translates and rotates with time-dependent linear and angular velocities  $\mathbf{V}$  and  $\boldsymbol{\Omega}$ , we work with the modified vorticity  $\mathbf{W} = \omega + 2\boldsymbol{\Omega}$ , which evolves according to the standard vorticity transport equation (3.8.34) written for an inertial frame (Speziale, 1987). Since the effects of acceleration of the frame of reference enter the solution only through the boundary conditions, using the modified vorticity simplifies the numerical implementation and reduces the computational demands.

2. In the second step, we compute the evolution of the velocity field by inverting the definition  $\omega = \nabla \times \mathbf{u}$  subject to the continuity equation  $\nabla \cdot \mathbf{u} = 0$ . The inversion can be done in two

different ways according to the *vector potential–vorticity* and *velocity–vorticity* formulation discussed in the following subsections.

### Vector potential–vorticity formulation

This method proceeds by decomposing the velocity field into the sum of a solenoidal irrotational velocity field  $\nabla \phi$ , where  $\phi$  is a harmonic function, and a rotational velocity field that is expressed in terms of the curl of a solenoidal vector potential  $\mathbf{A}$ , so that  $\mathbf{u} = \nabla \phi + \nabla \times \mathbf{A}$  (Hirasaki and Hellums, 1970). The velocity potential  $\phi$  is found by solving Laplace's equation subject to the required no-penetration condition  $\nabla \phi \cdot \mathbf{n} = \mathbf{u} \cdot \mathbf{n}$ , which determines  $\phi$  uniquely up to an arbitrary but physically irrelevant constant.

To compute the vector potential, we write  $\omega = \nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$  and stipulate that  $\mathbf{A}$  be solenoidal, thus obtaining the vectorial Poisson equation

$$\nabla^2 \mathbf{A} = -\omega \quad (13.1.24)$$

Taking the divergence of Eq. (13.1.24) and remembering that the vorticity field is solenoidal, shows that  $\nabla \cdot \mathbf{A}$  satisfies Laplace's equation, which means that the computed  $\mathbf{A}$  will be solenoidal provided that the boundary conditions on  $\mathbf{A}$  ensure that  $\nabla \cdot \mathbf{A} = 0$  over the boundaries. For simply connected domains, one way of ensuring that this constraint is fulfilled is to require that the tangential components of  $\mathbf{A}$  vanish; that is,  $\mathbf{n} \times (\mathbf{A} \times \mathbf{n}) = 0$ . This is consistent with the requirement that  $(\nabla \times \mathbf{A}) \cdot \mathbf{n} = 0$ . To derive a boundary condition for the normal component of  $\mathbf{A}$ , we introduce a local coordinate system with the  $x$  and  $z$  axes tangential to the boundary and the  $y$  axis normal to the boundary at a point, and use the condition  $\nabla \cdot \mathbf{A} = 0$ , to find that, at the origin,

$$\begin{aligned} \nabla \cdot \mathbf{A} &= \mathbf{n} \cdot (\nabla \mathbf{A}) \cdot \mathbf{n} + \frac{\partial \mathbf{A}}{\partial x} \cdot \mathbf{t}_x + \frac{\partial \mathbf{A}}{\partial z} \cdot \mathbf{t}_z \\ &= \mathbf{n} \cdot (\nabla \mathbf{A}) \cdot \mathbf{n} + \frac{\partial(\mathbf{A} \cdot \mathbf{t}_x)}{\partial x} + \frac{\partial(\mathbf{A} \cdot \mathbf{t}_z)}{\partial z} - \mathbf{A} \cdot \left( \frac{\partial \mathbf{t}_x}{\partial x} + \frac{\partial \mathbf{t}_z}{\partial z} \right) \\ &= \mathbf{n} \cdot (\nabla \mathbf{A}) \cdot \mathbf{n} + 2\kappa_m \mathbf{A} \cdot \mathbf{n} = 0 \end{aligned} \quad (13.1.25)$$

where  $\mathbf{n} \cdot (\nabla \mathbf{A}) \cdot \mathbf{n}$  is the derivative of the normal component of  $\mathbf{A}$  in a direction normal to the boundary,  $\mathbf{t}_x$  and  $\mathbf{t}_z$  are the unit tangential vectors in the directions of the  $x$  and  $z$  axes, and  $\kappa_m$  is the mean curvature of the boundary. When the boundary is flat, the mean curvature vanishes, and Eq. (13.1.25) assumes the simpler form  $\mathbf{n} \cdot (\nabla \mathbf{A}) \cdot \mathbf{n} = 0$ . Richardson and Cornish (1977) developed boundary conditions for multiply connected domains.

For two-dimensional or axisymmetric flow, we set all components of  $\mathbf{A}$  equal to zero except for the  $z$  component or azimuthal component, which is identified, respectively, with the stream function or with the Stokes stream function divided by the radial distance  $\sigma$ . We then find that the vector potential–vorticity formulation reduces to the stream function–vorticity formulation discussed at the beginning of this section.

The numerical implementation of the vector potential–vorticity formulation for three-dimensional flow has been discussed by several authors, including Aziz and Hellums (1967), Aragbesola and Burley (1977), and Mallinson and de Vahl Davis (1977); the reader is referred to their works for specific details.

### Velocity–vorticity formulation

In the most popular version of this formulation, the velocity is computed from the vorticity by solving the vectorial Poisson equation

$$\nabla^2 \mathbf{u} = -\nabla \times \omega \quad (13.1.26)$$

which arises by taking the curl of the definition  $\omega = \nabla \times \mathbf{u}$  and requiring that  $\mathbf{u}$  be a solenoidal function. The solution is found subject to the boundary conditions for the velocity specified in the statement of the problem.

To validate the method, we must show that the curl of the computed velocity will indeed be equal to  $\omega$ , provided that  $\omega$  is solenoidal and its boundary values are computed from  $\omega = \nabla \times \mathbf{u}$  (Daube, 1992; Trujillo, 1994). For this purpose, we use the vector identity  $\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u}$  in conjunction with Eq. (13.1.26) and find  $\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) + \nabla \times \omega$ . Taking the curl of both sides of this equation to eliminate the first term on the right-hand side yields  $\nabla \times \nabla \times \mathbf{F} = \nabla(\nabla \cdot \mathbf{F}) - \nabla^2 \mathbf{F} = 0$ , where we have defined  $\mathbf{F} = \nabla \times \mathbf{u} - \omega$ . Since  $\omega$  and thus  $\mathbf{F}$  is solenoidal, the components of  $\mathbf{F}$  must be harmonic functions. But the boundary values of  $\mathbf{F}$  are equal to zero, and this requires that  $\mathbf{F}$  vanish and thus  $\omega$  be equal to  $\nabla \times \mathbf{u}$  throughout the domain of flow.

One important consequence of this result is that the velocity computed from the solution of Eq. (13.1.26) will surely be solenoidal. This can be shown beginning, once again, with the identity  $\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u}$ , which now shows that  $\nabla(\nabla \cdot \mathbf{u}) = 0$ . Straightforward integration in the spatial variables reveals that  $\nabla \cdot \mathbf{u}$  is constant throughout the domain of flow; conservation of mass requires this constant to be equal to zero (Daube, 1992; Trujillo, 1994).

The numerical implementation of the velocity–vorticity formulation based on Eq. (13.1.26) has been discussed by several authors, including Chien (1976), Dennis, Ingham, and Cook (1979), Daube (1992), Guj and Stella (1993), Trujillo (1994). Guj and Stella (1988) developed a method of false transients for steady flow by replacing the elliptic equation (13.1.26) with the parabolic equation

$$Re \frac{\partial \mathbf{u}}{\partial t} = \nabla^2 \mathbf{u} + \nabla \times \omega \quad (13.1.27)$$

In another version of the velocity–vorticity formulation, the velocity field is computed directly by solving the Cauchy–Riemann-type system of equations  $\omega = \nabla \times \mathbf{u}$  and  $\nabla \cdot \mathbf{u} = 0$  for the velocity, subject to the no-penetration condition at the boundaries. The implementation of this method is discussed by Osswald, Ghia, and Ghia (1987) and Gatski, Grosh, and Rose (1989).

## PROBLEMS

- 13.1.1 **Poisson's equation for the pressure.** Take the divergence of the two-dimensional Navier–Stokes equation and introduce the stream function to derive the pressure Poisson equation (13.1.3).
- 13.1.2 **Axisymmetric flow.** Write the counterparts of Eqs. (13.1.1)–(13.1.3) for axisymmetric flow in terms of the Stokes stream function.
- 13.1.3 **Boundary condition for the vorticity.** Derive the expressions given in Eqs. (13.1.20) and (13.1.21).
- 13.1.4 **Integration of the vorticity transport equation.** Consider the temporal integration of the vorticity transport equation written in the nonconservative form with the vortex stretching term explicit on the right-hand side, subject to the boundary condition  $\omega = \nabla \times \mathbf{u}$ . Discuss whether the vorticity field will remain solenoidal during the time integration (Gatski, Grosh, and Rose, 1989).

## Computer Problems

- 13.1.5 **Steady flow in a cavity.** (a) Write a computer program called *CV2DSI* that computes the steady flow in a square cavity with width and depth equal to  $L$ , generated by the steady translation of

the lid, using the direct approach discussed in the text. The inner iterations should be conducted using the FTCS method. Carry out computations at a sequence of increasing Reynolds numbers  $Re = VL/\nu = 1, 10, 100, 500, \dots$  and discuss the changes in the structure of the flow. Study the convergence of the method as a function of the two numerical parameters  $\rho_1$  and  $\rho_2$  and number of iterations. Estimate the critical Reynolds number where your spatial resolution appears to be inadequate. To compute flow at the higher Reynolds numbers, it is helpful to use a continuation method in which the initial guesses for the stream function and vorticity are identified with the corresponding converged values at a lower Reynolds number. (b) Repeat part (a) but with the inner iterations carried out using the LSOR method and comment on the improvement. (c) Repeat part (a) with the method of modified dynamics.

- 13.1.6 **Unsteady flow in a cavity.** Write a computer program called *CV2DUI* that solves the unsteady version of Problem 13.1.5 with a lid that is set in motion impulsively at a constant velocity, using the first-order method discussed in the text.

## 13.2 VELOCITY–PRESSURE FORMULATION

In this section we proceed to discuss a class of methods for computing steady and unsteady, two-dimensional and three-dimensional flows in primitive variables including the velocity and the pressure. To simplify the derivations, we shall assume that the density and the viscosity of the fluid are uniform throughout the domain of flow.

We begin developing these methods by rewriting the Navier–Stokes equation in the form of an evolution equation as

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u}) - \frac{1}{\rho} \nabla P + \nu L(\mathbf{u}) \quad (13.2.1)$$

where  $P$  is the modified pressure and  $\mathbf{N}$  and  $L$  are, respectively, the nonlinear-inertial and linear-viscous operators defined as

$$\mathbf{N}(\mathbf{u}) = -\mathbf{u} \cdot \nabla \mathbf{u} = -\nabla \cdot (\mathbf{u}\mathbf{u}) \quad (13.2.2)$$

$$L(\mathbf{u}) = \nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times \omega \quad (13.2.3)$$

Because the velocity field is solenoidal, the first term on the right-hand side of Eq. (13.2.3) vanishes and could have been discarded; maintaining it, however, will allow us to take into account and filter out the accumulation of the numerical error. The equation of motion is accompanied by the continuity equation for incompressible fluids

$$\nabla \cdot \mathbf{u} = 0 \quad (13.2.4)$$

which states that the velocity field is and must remain solenoidal at all times.

### Pressure Poisson Equation

Taking the divergence of Eq. (13.2.1) and interchanging the gradient with the temporal derivative, we obtain an evolution equation for the rate of expansion

$$\frac{\partial \nabla \cdot \mathbf{u}}{\partial t} = \nabla \cdot \mathbf{N}(\mathbf{u}) - \frac{1}{\rho} \nabla^2 P + \nu \nabla \cdot L(\mathbf{u}) \quad (13.2.5)$$

The continuity equation requires that the left-hand side of Eqs. (13.2.5) vanish at all times, and this makes it necessary for the pressure to satisfy the pressure Poisson equation, (PPE),

$$\nabla^2 P = \rho \nabla \cdot \mathbf{N}(\mathbf{u}) + \mu \nabla \cdot L(\mathbf{u}) \quad (13.2.6)$$

One might argue that since the divergence and the linear operator  $L$  commute, the second term on the right-hand side of Eq. (13.2.6) vanishes, yielding the simplified form

$$\nabla^2 P = \rho \nabla \cdot \mathbf{N}(\mathbf{u}) \quad (13.2.7)$$

We shall see, however, that issues of numerical stability require the use of the more cumbersome form (13.2.6). Following Gresho and Sani (1987), we call Eq. (13.2.6) the *consistent PPE*, and Eq. (13.2.7) the *simplified PPE*.

### Alternative Systems of Governing Equations

To this end, we consider replacing the original system of governing equations (13.2.1) and (13.2.4) with either (1) the modified system of Eqs. (13.2.1) and (13.2.6), or (2) the modified system of Eqs. (13.2.1) and (13.2.7). These replacements will be acceptable as long as the modified systems guarantee that the velocity remains solenoidal at all times.

Substituting Eq. (13.2.6) into Eq. (13.2.5), we find

$$\frac{\partial \nabla \cdot \mathbf{u}}{\partial t} = 0 \quad (13.2.8)$$

which states that, if the velocity field is solenoidal at the initial time, it will remain solenoidal at all times. Thus, if the initial velocity field is solenoidal, it is permissible to replace the continuity equation with the consistent PPE (13.2.6). When the initial rate of expansion, however, is not equal to zero, the divergence of the velocity will remain finite throughout the evolution.

Substituting Eq. (13.2.7) into Eq. (13.2.5) and interchanging the divergence with the Laplacian, we obtain the unsteady diffusion equation for the rate of expansion

$$\frac{\partial \nabla \cdot \mathbf{u}}{\partial t} = \nu L(\nabla \cdot \mathbf{u}) \quad (13.2.9)$$

The general properties of the unsteady diffusion equation in a bounded domain show that the rate of expansion will vanish at all times provided that (1) the initial velocity field is solenoidal, and (2) the rate of expansion or its normal derivative over all boundaries are held equal to zero at all times. When these conditions are met, it is permissible to replace the continuity equation with the simplified PPE (13.2.6). The second condition, in particular, underlines the importance of accurately satisfying conservation of mass at the grid points near or at the boundaries. When the initial rate of expansion is not equal to zero, but its boundary distribution is kept equal to zero, the magnitude of the divergence of the velocity will keep decreasing and eventually will tend to vanish during the evolution.

### Boundary Conditions for the Pressure

The consistent and modified PPEs, and their finite-difference counterparts, must be solved subject to one scalar boundary condition over each boundary of the flow. According to the preceding discussion, this condition must ensure that the boundary distribution of the divergence of the velocity vanish at all times.

The derivation of boundary conditions for the pressure is discussed in two illuminating articles by Orszag, Israeli, and Deville (1986) and Gresho and Sani (1987). Their analyses show that the Neumann boundary condition

$$\nabla P \cdot \mathbf{n} = \rho \left( -\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{n} + \mathbf{N}(\mathbf{u}) \cdot \mathbf{n} \right) + \mu L(\mathbf{u}) \cdot \mathbf{n} \quad (13.2.10)$$

which emerges by applying Eq. (13.2.1) at the boundaries of the flow and then projecting it onto the normal vector  $\mathbf{n}$ , is always appropriate, for *it is another manifestation of the condition of incompressibility at the boundaries*.

Using the boundary condition (13.2.10) guarantees that replacing the continuity equation with the simplified PPE is a valid substitution. The solution for the pressure computed using Eq. (13.2.10) also satisfies the Dirichlet boundary condition, which emerges by projecting the equation of motion onto a tangential vector and then integrating it with respect to the tangential arc length. On the contrary, the pressure field that is computed by solving the PPE using the Dirichlet

condition will satisfy the Neumann condition only when the initial velocity field is sufficiently regular (Gresho and Sani, 1987).

For a planar wall that translates parallel to itself with constant velocity, we use the no-slip and no-penetration conditions to find that the Neumann condition (13.2.10) simplifies to

$$\nabla P \cdot \mathbf{n} = \mu \frac{\partial^2 \mathbf{u}}{\partial l_n^2} \cdot \mathbf{n} \quad (13.2.11)$$

where  $\mathbf{n}$  is the unit normal vector directed into the flow, and  $l_n$  designates the arc length normal to the wall measured toward the fluid. When the Reynolds number is sufficiently large, the right-hand side of Eq. (13.2.11) is small and is sometimes set equal to zero.

In a certain class of finite-difference methods, to be discussed in Section 13.4, the Neumann boundary condition (13.2.10) is not enforced in an explicit manner. Instead, the numerical method employs an alternative boundary condition that emerges by satisfying the continuity equation at the grid nodes that are adjacent to the boundaries or are located at the boundaries. The consistent implementation of this procedure is equivalent to requiring the Neumann condition (13.2.10).

Quartapelle and Napolitano (1986) replaced the boundary condition for the pressure with an integral constraint involving the projection of the boundary distribution of the pressure onto a nonsingular solution of the vectorial Helmholtz equation defined within the domain of flow. Assessing the efficiency of their method, however, awaits further research.

### Compatibility Condition for the PPE

The emerging computational procedure involves solving a Poisson equation of the form  $\nabla^2 P = g$  subject to the Neumann boundary condition  $\nabla P \cdot \mathbf{n} = q$  over all boundaries. Here  $g$  can be identified with the right-hand side of either Eq. (13.2.6) or (13.2.7), and  $q$  is identified with the right-hand side of Eq. (13.2.10). Integrating the pressure Poisson equation over the domain of flow, and using the divergence theorem, we find that a solution will exist only when the following compatibility condition is fulfilled

$$\int_{\text{Flow}} \nabla^2 P \, dV = - \int_{\text{Boundaries}} \nabla P \cdot \mathbf{n} \, dS$$

or

$$\int_{\text{Flow}} g \, dV = - \int_{\text{Boundaries}} q \, dS \quad (13.2.12)$$

The counterpart of Eq. (13.2.12) in two dimensions is

$$\int_{\text{Flow}} g \, dA = - \int_{\text{Boundaries}} q \, dl \quad (13.2.13)$$

The satisfaction of Eq. (13.2.12) is guaranteed in the continuous version of the problem, but discretization errors may destroy the exact equality in the corresponding finite-difference formulation (Problem 13.2.2). In solving Poisson's equation using iterative methods, this inconsistency may result in slow convergence or even divergence of the numerical solution; in solving it using a direct method, it has the inevitable consequence that one of the linear equations expressing the Poisson equation at a particular grid point will not be satisfied, and the pressure may exhibit local oscillations.

Two ways of ensuring that the compatibility condition is fulfilled in the discrete formulation of the problem are:

1. Modify the source term  $g$  of Poisson's equation in the discrete statement of the problem by a proper amount (Ghia, Hankey, and Hodge 1977; Sheth and Pozrikidis, 1995; see also Problem 13.2.2);

2. Use a custom-made finite-difference method that coordinates the discretization of the equation of motion, the PPE, and the boundary conditions, so as to automatically satisfy the discrete version of the compatibility condition. This procedure is sometimes called the *consistent finite-difference discretization*. The method was originally developed by Abdallah (1987) for a uniform two-dimensional Cartesian grid, and was subsequently extended to three-dimensional and curvilinear grids by several authors including Mansour and Hamed (1990), Sotiropoulos and Abdallah (1991), and Babu and Korpela (1994).

When the discrete version of the compatibility condition is fulfilled, the linear system of equations associated with the discrete Poisson equation will have a multiplicity of solutions reflecting the fact that the pressure may be determined only up to an arbitrary constant. To render the solution unique, we must specify the value of the pressure at an arbitrary grid point, or set the average value of the pressure over all grid points at an arbitrary level.

### An Explicit Evolution Equation for the Pressure

An evolution equation for the pressure may be obtained by differentiating the pressure Poisson equation in time and using the equation of motion to eliminate the time derivatives of the velocity in favor of the velocity and pressure, as discussed in Section 9.1. The result is the Poisson equation for  $\partial P/\partial t$  shown in Eq. (9.1.5), to be solved subject to the Neumann boundary condition that arises by differentiating Eq. (13.2.10) in time and interchanging the order of the normal spatial derivative and temporal derivative on the left-hand side. Unfortunately, this method appears to be untested in practice.

In closing this section, we compare the formulation in primitive variables to the formulations based on the vorticity transport equation discussed in the preceding section and find relative weaknesses and strengths. One weakness is the need to derive boundary conditions for the pressure; one strength is ease of extension to multifluid and interfacial flows.

## PROBLEMS

- 13.2.1 Compatibility condition for the PPE.** Show that the compatibility condition (13.2.12) for the pressure Poisson equation Eq. (13.2.6) with boundary conditions given in Eq. (13.2.10) is fulfilled (Gresho and Sani, 1987).
- 13.2.2 Solving Poisson's equation with Neumann boundary conditions.** Consider Poisson's equation  $\nabla^2 \phi = g$  in a three-dimensional domain  $\Omega$  with Neumann boundary conditions  $\nabla \phi \cdot \mathbf{n} = q$  all around the boundaries  $\partial\Omega$ . For a solution to exist, the compatibility condition (13.2.12) must be fulfilled. In a discrete finite-difference formulation, the problem is expressed in terms of an  $N \times N$  linear system of equations  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}(g, q)$ , which incorporates Poisson's equation applied at the internal grid points and the boundary conditions. This notation emphasizes that the constant vector  $\mathbf{b}$  is a function of the source term  $g$  as well as of the specified boundary flux  $q$ . Unless the corresponding discrete version of Eq. (13.2.12) is fulfilled to machine accuracy, the linear system will not have a solution. One way of circumventing this difficulty is to perturb the source term  $g$ , thus modifying the constant vector  $\mathbf{b}$ . This can be done by replacing  $g$  with  $g + \epsilon f$ , where  $\epsilon$  is a small number to be found as part of the solution, and  $f$  is an arbitrarily specified function that is independent of  $g$ . The proper value of  $\epsilon$  is given by

$$\epsilon \int_{\Omega} f dV = - \int_{\Omega} g dV - \int_{\partial\Omega} q dS \quad (13.2.14)$$

The method can be implemented numerically according to the following steps: First, we solve the first  $N - 1$  equations of the system  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}(g, q)$  for the first  $N - 1$  unknowns, set the last unknown equal to zero,  $x_n = 0$ , and call the solution  $\mathbf{x}^{(1)}$ . We then compute the residual of the

last equation  $R^{(1)} = A_{nn}x_1^{(1)} - b_n$ . Next, we solve the first  $N - 1$  equations of the linear system  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}(f, q)$  for the first  $N - 1$  unknowns, set the last unknown equal to zero,  $x_n = 0$ , and call the solution  $\mathbf{x}^{(2)}$ . Then, we compute the residual of the last equation  $R^{(2)} = A_{nn}x_1^{(2)} - b_n$ . Finally, we set  $\epsilon = R^{(1)}/R^{(2)}$  and compute the final solution  $\mathbf{x} = \mathbf{x}^{(1)} + \epsilon \mathbf{x}^{(2)}$ . Show that this solution satisfies all  $N$  equations of the linear system  $\mathbf{A} \cdot \mathbf{x} = \mathbf{b}(g + \epsilon f, q)$ .

### Computer Problem

- 13.2.3 Solving the Poisson equation in a rectangular domain.** Write a routine called *RPPE* that uses the method described in Problem 13.2.2 to solve Poisson's equation in a two-dimensional rectangular domain with an arbitrary source term assigned at the grid points, and arbitrary Neumann boundary conditions all around the boundaries. The Laplacian should be approximated using the five-point formula, and the normal derivatives at the boundaries should be approximated using second-order, one-side finite differences (Section B.5, Appendix, B).

## 13.3 IMPLEMENTATION OF METHODS IN PRIMITIVE VARIABLES

Having discussed the basic considerations that enter the computation of an incompressible Newtonian flow in terms of the velocity and pressure, we proceed to develop specific implementations.

### The Explicit Method of Harlow and Welch on a Staggered Grid

The marker and cell (MAC) method of Harlow and Welch (1965) combines a finite-difference method for solving the equations of incompressible flow and a marker-tracing method for tracking the motion of free surfaces or fluid interfaces.

The finite-difference method is based on the explicit forward-time discretization of the equation of motion (13.2.1) yielding the velocity at the next time-level in terms of the velocity and the pressure at the current time level,

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \left( \mathbf{N}(\mathbf{u}^n) + \nu L(\mathbf{u}^n) - \frac{1}{\rho} \nabla P^n \right) \quad (13.3.1)$$

To compute the pressure  $P^n$ , we discretize the evolution equation for the rate of expansion, Eq. (13.2.5), using a forward difference in time and obtain

$$\frac{(\nabla \cdot \mathbf{u})^{n+1} - (\nabla \cdot \mathbf{u})^n}{\Delta t} = \nabla \cdot \mathbf{N}(\mathbf{u}^n) - \frac{1}{\rho} \nabla^2 P^n + \nu L(\nabla \cdot \mathbf{u})^n \quad (13.3.2)$$

Requiring that the divergence of the velocity at the  $n + 1$  level vanish, we obtain the following Poisson equation for the pressure

$$\nabla^2 P^n = \frac{\rho}{\Delta t} (\nabla \cdot \mathbf{u})^n + \rho \nabla \cdot \mathbf{N}(\mathbf{u}^n) + \mu L(\nabla \cdot \mathbf{u})^n \quad (13.3.3)$$

which is a modified version of the consistent PPE, Eq. (13.2.6). Although small, the first and third terms on the right-hand side must be retained in order to prevent the onset of numerical instabilities. The computational algorithm involves the following steps:

- Specify an initial solenoidal velocity field that satisfies the prescribed boundary conditions.

2. Compute the pressure field by solving Poisson's equation (13.3.3) subject to the Neumann boundary conditions (13.2.10). In practice, this is done using an alternative but equivalent set of boundary conditions to be discussed shortly.
3. Use Eq. (13.3.1) to advance the velocity field by one step in time subject to the prescribed boundary conditions. The spatial derivatives on the right-hand side of Eq. (13.3.1) are computed using central differences, as will be discussed shortly. The size of the time step must be kept sufficiently small in order to suppress the growth of numerical oscillations.
4. Return to step 2 and repeat the computations for another time step.

#### Staggered grid

Harlow and Welch (1965) implemented their method on the staggered grid shown in Figure 13.3.1. In two dimensions, the staggered grid is composed of two superposed grids that are displaced with respect to each other by distances equal to half the grid spacings. The *primary* grid is drawn with solid lines, and the *secondary* grid is drawn with broken lines. Note that the secondary grid conforms with the physical boundaries of the flow where the velocity is known.

Discrete values of the pressure are assigned at the primary nodes  $(i, j)$  shown with *circles*, values of the  $x$  component of the velocity are assigned at the intersections between the primary and secondary grids  $(i + 1/2, j)$  shown with *squares*, and values of the  $y$  component of the velocity are assigned at the intersections between the primary and secondary grids  $(i, j + 1/2)$  shown with *triangles* in Figure 13.3.1. Note that the squares and triangles are located at the faces of a primary cell.

The distinguishing feature of the staggered-grid method is that the unknown functions and governing equations are defined or enforced at different nodes, and this decoupling simplifies the numerical implementation and enhances the numerical stability of the method. We shall see, in particular, that the fact that the pressure nodes are located in the interior of the flow simplifies the implementation of the boundary conditions for the pressure. Unfortunately, the staggered-grid

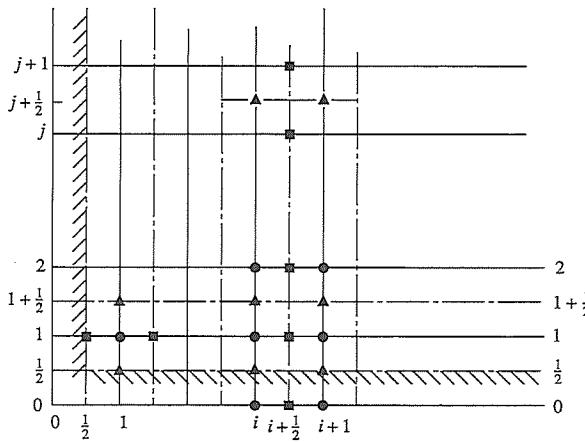


Figure 13.3.1 Staggered grid for flow in a rectangular domain near a corner. The primary grid is drawn with heavy lines, and the secondary grid is drawn with broken lines. Note that the secondary grid conforms with the physical boundaries of the flow. The pressure is defined at the circles, the  $x$  component of the velocity is defined at the squares; and the  $y$  component is defined at the triangles.

method becomes considerably more involved, and sometimes prohibitively expensive, when applied to grids that are defined in curvilinear coordinates.

#### Interpolation and extrapolation of the velocity

The velocity components at the vertices of the primary or secondary grids are computed by linear interpolation from the closest nodes. For instance,

$$\begin{aligned} u_{i,j} &= \frac{1}{2}(u_{i-1/2,j} + u_{i+1/2,j}), & u_{i+1/2,j+1/2} &= \frac{1}{2}(u_{i+1/2,j} + u_{i+1/2,j+1}) \\ v_{i,j} &= \frac{1}{2}(v_{i,j+1/2} + v_{i,j-1/2}), & v_{i+1/2,j+1/2} &= \frac{1}{2}(v_{i,j+1/2} + v_{i+1,j+1/2}) \end{aligned} \quad (13.3.4)$$

For a rectangular domain of flow with a grid of size  $N \times M$ , the velocity components at the external nodes corresponding to  $i = 0, i = N + 1, j = 0, j = M + 1$  are computed taking into account the boundary conditions at the physical boundaries of flow that are located at  $i = \frac{1}{2}, i = N + \frac{1}{2}, j = \frac{1}{2}, j = M + \frac{1}{2}$ . For instance, requiring that  $u = U$  at  $j = \frac{1}{2}$ , and applying linear extrapolation yields

$$u_{i+1/2,0} = 2U - u_{i+1/2,1} \quad (13.3.5)$$

which is called the *reflection formula*. Equation (13.3.5) allows us to compute the first partial derivative  $\partial u / \partial y$  at the level  $j = 1$  using central differences with first-order accuracy, that is, by setting  $(\partial u / \partial y)_{i+1/2,1} = (u_{i+1/2,2} - u_{i+1/2,0}) / (2 \Delta y)$ . To compute the corresponding second partial derivative with comparable accuracy using central differences, we must use the alternative formula

$$u_{i+1/2,0} = \frac{1}{3}(8U - 6u_{i+1/2,1} + u_{i+1/2,2}) \quad (13.3.6)$$

Similar equations can be derived for use with non-centered finite differences for the first or second derivatives near the boundaries (Peyret and Taylor, 1983, p. 151).

#### Advancing the velocity field

To advance the components of the velocity in time, we apply Eq. (13.3.1) at the *internal velocity nodes*, express the nonlinear convection term in conservative form, and approximate the spatial derivatives using central differences to obtain

$$u_{i+1/2,j}^{n+1} = u_{i+1/2,j}^n + \Delta t \left( \Delta u_{i+1/2,j}^n - \frac{1}{\rho} \frac{P_{i+1,j}^n - P_{i,j}^n}{\Delta x} \right) \quad (13.3.7)$$

and

$$v_{i,j+1/2}^{n+1} = v_{i,j+1/2}^n + \Delta t \left( \Delta v_{i,j+1/2}^n - \frac{1}{\rho} \frac{P_{i,j+1}^n - P_{i,j}^n}{\Delta y} \right) \quad (13.3.8)$$

where

$$\begin{aligned} \Delta u_{i+1/2,j} &= - \frac{u_{i+1,j}^2 - u_{i,j}^2}{\Delta x} - \frac{1}{\Delta y} (u_{i+1/2,j+1/2} v_{i+1/2,j+1/2} - u_{i+1/2,j-1/2} v_{i+1/2,j-1/2}) \\ &+ \nu \left( \frac{u_{i+3/2,j} - 2u_{i+1/2,j} + u_{i-1/2,j}}{\Delta x^2} + \frac{u_{i+1/2,j+1} - 2u_{i+1/2,j} + u_{i+1/2,j-1}}{\Delta y^2} \right) \end{aligned} \quad (13.3.9)$$

and

$$\begin{aligned} \Delta v_{i,j+1/2} &= - \frac{1}{\Delta x} (u_{i+1/2,j+1/2} v_{i+1/2,j+1/2} - u_{i-1/2,j+1/2} v_{i-1/2,j+1/2}) - \frac{v_{i,j+1}^2 - v_{i,j}^2}{\Delta y} \\ &+ \nu \left( \frac{v_{i+1,j+1/2} - 2v_{i,j+1/2} + v_{i-1,j+1/2}}{\Delta x^2} + \frac{v_{i,j+3/2} - 2v_{i,j+1/2} + v_{i,j-1/2}}{\Delta y^2} \right) \end{aligned} \quad (13.3.10)$$

Before we can compute the right-hand sides of Eqs. (13.3.7) and (13.3.8), however, we must solve the Poisson equation for the pressure.

#### Pressure Poisson equation

As a first step, we approximate the rate of expansion at the pressure nodes using central differences and obtain

$$D_{i,j} \equiv (\nabla \cdot \mathbf{u})_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \quad (13.3.11)$$

Requiring that the right-hand side of Eq. (13.3.11) vanish at the interior pressure nodes  $i = 2, \dots, N-1$  and  $j = 2, \dots, M-1$ , at the  $n+1$  time level, and expressing the velocities at the  $n+1$  level in terms of the velocities and pressure at the  $n$ th level using Eqs. (13.3.7) and (13.3.8), we obtain

$$\begin{aligned} D_{i,j}^n + \Delta t \left( \frac{\Delta u_{i+1/2,j}^n - \Delta u_{i-1/2,j}^n}{\Delta x} + \frac{\Delta v_{i,j+1/2}^n - \Delta v_{i,j-1/2}^n}{\Delta y} \right) \\ - \frac{1}{\rho} \Delta t \left( \frac{P_{i+1,j}^n - 2P_{i,j}^n + P_{i-1,j}^n}{\Delta x^2} + \frac{P_{i,j+1}^n - 2P_{i,j}^n + P_{i,j-1}^n}{\Delta y^2} \right) = 0 \end{aligned} \quad (13.3.12)$$

Rearranging, and using the definitions (13.3.9) and (13.3.10), we derive the discrete version of the pressure Poisson equation (13.3.3), where the Laplacian is approximated using central differences over intervals equal to  $\Delta x$  and  $\Delta y$

$$\begin{aligned} \frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{\Delta x^2} + \frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{\Delta y^2} = \frac{\rho}{\Delta t} D_{i,j} + \rho Q_{i,j} \\ + \mu \left( \frac{D_{i+1,j} - 2D_{i,j} + D_{i-1,j}}{\Delta x^2} + \frac{D_{i,j+1} - 2D_{i,j} + D_{i,j-1}}{\Delta y^2} \right) \end{aligned} \quad (13.3.13)$$

where we have defined

$$\begin{aligned} Q_{i,j} = - \frac{u_{i+1,j}^2 - 2u_{i,j}^2 + u_{i-1,j}^2}{\Delta x^2} - \frac{v_{i,j+1}^2 - 2v_{i,j}^2 + v_{i,j-1}^2}{\Delta y^2} \\ - \frac{2}{\Delta x \Delta y} (u_{i+1/2,j+1/2} v_{i+1/2,j+1/2} + u_{i-1/2,j-1/2} v_{i-1/2,j-1/2} \\ - u_{i+1/2,j-1/2} v_{i+1/2,j-1/2} - u_{i-1/2,j-1/2} v_{i-1/2,j-1/2}) \end{aligned} \quad (13.3.14)$$

All variables in Eqs. (13.3.13) and (13.3.14) are evaluated at the  $n$ th time level.

#### Pressure boundary condition

For the pressure nodes that are adjacent to the boundaries corresponding to  $i = 1, N$ , and  $j = 1, M$ , we follow a slightly different approach that takes into account the boundary conditions for the velocity. Considering the first horizontal layer  $j = 1$ , we require that the discrete form of the rate of expansion, given by the right-hand side of Eq. (13.3.11), vanish at the  $n+1$  time level. Focusing on the pressure nodes  $j = 1$  and  $i = 2, \dots, N-1$  that are located away from the corners, we express the velocities at the nodes  $(i - \frac{1}{2}, 1), (i, \frac{3}{2}), (i + \frac{1}{2}, 1)$  at the  $n+1$  level in terms of the velocities and pressure at the  $n$ th level using Eqs. (13.3.7) and (13.3.8), and obtain the counterpart of Eq. (13.3.12),

$$\begin{aligned} \frac{u_{i+1/2,1}^n - u_{i-1/2,1}^n}{\Delta x} + \frac{v_{i,3/2}^n - v_{i,1/2}^{n+1}}{\Delta y} + \Delta t \left( \frac{\Delta u_{i+1/2,1}^n - \Delta u_{i-1/2,1}^n}{\Delta x} + \frac{\Delta v_{i,3/2}^n}{\Delta y} \right) \\ - \frac{1}{\rho} \Delta t \left( \frac{P_{i+1,1}^n - 2P_{i,1}^n + P_{i-1,1}^n}{\Delta x^2} + \frac{P_{i,2}^n - P_{i,1}^n}{\Delta y^2} \right) = 0 \end{aligned} \quad (13.3.15)$$

The velocity  $v^{n+1}$  at the  $(i, \frac{1}{2})$  node is available from the prescribed boundary conditions. A straightforward modification of Eq. (13.3.15) is required for the corner nodes  $i = 1$  and  $N$ .

Working similarly for the last horizontal layer corresponding to  $j = M$  with  $i = 2, \dots, N-1$ , we find

$$\begin{aligned} \frac{u_{i+1/2,M}^n - u_{i-1/2,M}^n}{\Delta x} + \frac{v_{i,M+1/2}^{n+1} - v_{i,M-1/2}^n}{\Delta y} + \Delta t \left( \frac{\Delta u_{i+1/2,M}^n - \Delta u_{i-1/2,M}^n}{\Delta x} - \frac{\Delta v_{i,M-1/2}^n}{\Delta y} \right) \\ - \frac{1}{\rho} \Delta t \left( \frac{P_{i+1,M}^n - 2P_{i,M}^n + P_{i-1,M}^n}{\Delta x^2} - \frac{P_{i,M}^n - P_{i,M-1}^n}{\Delta y^2} \right) = 0 \end{aligned} \quad (13.3.16)$$

Straightforward modifications are necessary for the two corner nodes. Similar equations may be derived for the first and last vertical layers corresponding to  $i = 1, N$  and  $j = 2, \dots, M-1$ .

Equations (13.3.15) and (13.3.16), and their counterparts for the first and last vertical layers, provide us with boundary conditions for the discrete Poisson equation (13.3.13). It appears that by using a staggered grid, we have circumvented the derivation of explicit boundary conditions for the pressure. Gresho and Sani (1987), however, demonstrated that Eq. (13.3.15) and its counterparts for the other three walls amount to the Neumann boundary condition (13.2.10).

To implement the explicit Neumann boundary condition for the pressure, let us refer to Figure 13.3.1, and introduce the external pressure nodes  $P_{i,0}$ . Using the continuity equation, we find that  $\partial v/\partial y$  vanishes at the bottom wall, and then apply Eq. (13.2.11) and approximate the derivatives using central differences to obtain

$$(\nabla P \cdot \mathbf{n})_{i,1/2} = \left( \frac{\partial P}{\partial y} \right)_{i,1/2} \cong \frac{P_{i,1} - P_{i,0}}{\Delta y} \cong \mu \left( \frac{\partial^2 v}{\partial y^2} \right)_{i,1/2} \cong 2\mu \frac{v_{i,3/2}}{\Delta y^2} \quad (13.3.17)$$

which may be rearranged to yield

$$P_{i,0} = P_{i,1} - 2\mu \frac{v_{i,3/2}}{\Delta y} \quad (13.3.18)$$

Working similarly with the upper, left, and right walls, we find

$$\begin{aligned} P_{i,M+1} = P_{i,M} + 2\mu \frac{v_{i,M-1/2}}{\Delta y}, & \quad P_{0,j} = P_{1,j} - 2\mu \frac{u_{3/2,j}}{\Delta y} \\ P_{N+1,j} = P_{N,j} + 2\mu \frac{u_{N-1/2,j}}{\Delta y} \end{aligned} \quad (13.3.19)$$

Eqs. (13.3.18) and (13.3.19) provide us with an alternative set of boundary conditions for Poisson's equation (13.3.13), which is now also applied at the pressure nodes that are adjacent to the boundaries, for instance, at  $j = 1$ .

#### The Explicit Method of Harlow and Welch on a Non-staggered Grid

Consider next the implementation of the explicit method of Harlow and Welch on the non-staggered grid shown in Figure 13.1.1. The discretized equation of motion (13.3.1) and pressure Poisson equation (13.3.3) are enforced at all internal grid points, and the spatial derivatives are approximated using centered differences.

To derive boundary conditions for the pressure, we require that the discretized form of the rate of expansion  $D$  vanish at the grid points that are located on the four walls. Considering the first horizontal layer  $j = 1$  with  $i = 2, \dots, N$ , we use a central difference in the  $x$  direction and a second-order one-sided difference in the  $y$  direction to write

$$D_{ij} = \frac{u_{i+1,1} - u_{i-1,1}}{2\Delta x} + \frac{-v_{i,3} + 4v_{i,2} - 3v_{i,1}}{2\Delta y} \quad (13.3.20)$$

(Chorin, 1968). Requiring that the right-hand side of Eq. (13.3.20) vanish at the  $n + 1$  time level and expressing the velocities at the  $(i, 2)$  and  $(i, 3)$  grid points in terms of the discrete version of the right-hand side of Eq. (13.3.1) yields an equation that is similar to Eq. (13.3.15). Working in a similar manner, we derive corresponding equations for the upper, left, and right walls.

It would appear that we have again circumvented the explicit derivation of boundary conditions for the pressure, but Gresho and Sani (1987) demonstrated that Eq. (13.3.20) and its counterparts for the other three walls amount to the Neumann boundary condition (13.2.10). This is true even when first-order one-sided differences are used to approximate the second partial derivative in Eq. (13.3.20).

### Solenoidality of the Discrete Velocity Field

Let us assume that the solution of the discrete version of Poisson's equation for the pressure has been computed, and the results are substituted back into Eq. (13.3.1) to advance the velocity. To this end, we must ask whether the divergence of the updated velocity  $\nabla \cdot \mathbf{u}^{n+1}$  will be equal to zero to machine accuracy. The answer is negative, unless the finite-difference methods for solving the pressure Poisson equation and for computing the divergence of the velocity have been coordinated in an appropriate manner (Sotiropoulos and Abdallah, 1991).

One way to ensure that the discrete velocity field is solenoidal to machine accuracy is to solve the pressure Poisson equation using the five-point formula with intervals of size  $2\Delta x$ ,  $2\Delta y$ , and  $2\Delta z$ , and then compute the divergence of the velocity using central differences with intervals of size  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$ . Unfortunately, the solution of the Poisson equation using this method may exhibit numerical instabilities due to the decoupling of the values of the pressure at neighboring grid points. In practice, a small value of the rate of expansion at the grid points is tolerated and may be reduced by grid refinement.

### Higher-Order Methods

The explicit method of Harlow and Welch is first-order accurate in time and conditionally stable. To improve the accuracy and relax the stability constraints, we resort to semi-implicit, fully implicit, or predictor-corrector iterative methods.

Adopting the Crank-Nicolson method, for example, we apply the equation of motion at the intermediate  $n + \frac{1}{2}$  time level and obtain

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \frac{1}{2} [\mathbf{N}(\mathbf{u}^{n+1}) + \mathbf{N}(\mathbf{u}^n) + \nu L(\mathbf{u}^{n+1}) + \nu L(\mathbf{u}^n)] - \frac{\Delta t}{\rho} \nabla P^{n+1/2} \quad (13.3.21)$$

which is second-order accurate in time and enjoys an enhanced numerical stability. Since the method is implicit in the nonlinear terms, carrying out each time step requires solving a system of nonlinear algebraic equations for the velocity at the  $n + 1$  time level and pressure at the  $n + \frac{1}{2}$  level. In practice, this is done using an iterative method, as discussed by Peyret and Taylor (1983, p. 167).

## PROBLEMS

- 13.3.1 Pressure boundary conditions for flow in a cavity with a staggered grid. Derive the counterparts of Eqs. (13.3.15) and (13.3.16) for (a) the first and last vertical layers corresponding to  $i = 1, N$  and  $j = 2, \dots, M - 1$ , (b) the four corner pressure nodes.
- 13.3.2 Neumann boundary conditions for the pressure for flow in a cavity with a nonstaggered grid. Derive the finite-difference statement of the Neumann boundary condition for the pressure for flow in a cavity on the non-staggered grid shown in Figure 13.1.1.

### Computer Problems

- 13.3.3 Flow in a rectangular cavity with a staggered grid. Write a program called CV2DSG that uses the explicit method of Harlow and Welch on a staggered grid, as described in the text, to compute transient flow in a square cavity due to the impulsive translation of the lid discussed in Problem 13.1.5. The pressure Poisson equation should be solved using an iterative method of your choice, including the method of the routine RPPE described in Problem 13.2.3.
- 13.3.4 Flow in a rectangular cavity with a non-staggered grid. Repeat Problem 13.3.3 with a non-staggered grid.

### 13.4 OPERATOR SPLITTING, PROJECTION, AND PRESSURE-CORRECTION METHODS

The Navier-Stokes equation states that the velocity at a particular point in a flow changes because of the simultaneous action of the nonlinear convection term, the linear viscous term, and the pressure gradient. In the operator splitting method, individual terms or groups of these terms are decoupled and considered to operate sequentially for time intervals of equal duration.

Since the absence of an evolution equation for the pressure is an important consideration, it is natural to decouple the convective-diffusive term from the pressure gradient term, thereby replacing Eq. (13.2.1) with the system of two component equations

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u}) + \nu L(\mathbf{u}) \quad (13.4.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla P \quad (13.4.2)$$

which are assumed to apply sequentially, each one for the full time interval  $\Delta t$  within each time step.

The nonlinear convection-diffusion equation (13.4.1) advances the velocity from the initial state  $\mathbf{u}^n$  to an intermediate state  $\mathbf{u}^*$ , and the pressure-correction equation (13.4.2) advances the intermediate state  $\mathbf{u}^*$  to the final state  $\mathbf{u}^{n+1}$  and completes the  $n$ th time step.

#### Solenoidal Projection and the Role of Pressure

To analyze the nature of the fractional-step decomposition expressed by Eqs. (13.4.1) and (13.4.2), it is helpful to introduce the concept of solenoidal projections (Chorin, 1968; see also Section 9.1 of this book). First we introduce the space of all vector functions, and note that the velocity field of an unsteady incompressible flow must evolve within the subspace of solenoidal functions. Next we observe that the intermediate velocity  $\mathbf{u}^*$  at the end of the convection-diffusion step will not necessarily be solenoidal. The evolution according to Eq. (13.4.1) allows for a departure from the subspace of solenoidal functions, and the pressure-correction step makes up for this departure by projecting  $\mathbf{u}^*$  back into the subspace of solenoidal functions, thus producing the final velocity  $\mathbf{u}^{n+1}$ .

Since the pressure  $P$  has not been updated during the convection-diffusion step, it loses its physical significance and must be regarded as an auxiliary function whose main purpose is to project  $\mathbf{u}^*$  onto the subspace of solenoidal functions. Thus, to be rigorous, we must replace  $P$  in Eq. (13.4.2) with a projection function  $\phi$ , obtaining

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla \phi \quad (13.4.3)$$

The relation between  $P$  and  $\phi$  is discussed in detail by Gresho (1990).

### Boundary Conditions for the Intermediate Variables

An important issue in the implementation of the fractional-step method is the choice of boundary conditions for the intermediate velocity  $\mathbf{u}^*$  and for the projection function  $\phi$ . Ideally, the boundary conditions for  $\mathbf{u}^*$  should be chosen so as to minimize its divergence, subject to the constraint that the required boundary conditions will be observed at the end of a complete time step.

The derivation of boundary conditions for  $\mathbf{u}^*$  and  $\phi$  has been the subject of extensive discussions (Orszag et al., 1986; Gresho, 1990). Gresho (1990) derived several sets of optimal and simplified sets of boundary conditions with varying degrees of accuracy, sophistication, and ease of implementation. In the simplest scheme, the intermediate velocity  $\mathbf{u}^*$  satisfies the boundary conditions that are required at the  $n + 1$  time level, while the projection function  $\phi$  satisfies the homogeneous Neumann condition  $\nabla \phi \cdot \mathbf{n} = 0$ . Far from the boundaries,  $\phi$  reduces to  $P$ .

### First-Order Projection Method

We proceed now to discuss the implementation of a particular projection method developed by Chorin (1968). We begin by splitting the convection-diffusion step expressed by Eq. (13.4.1) into four sequential convection-diffusion fractional steps expressed by the equations

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + u_x \frac{\partial \mathbf{u}}{\partial x} &= \nu \frac{\partial^2 \mathbf{u}}{\partial x^2}, & \frac{\partial \mathbf{u}}{\partial t} + u_y \frac{\partial \mathbf{u}}{\partial y} &= \nu \frac{\partial^2 \mathbf{u}}{\partial y^2} \\ \frac{\partial \mathbf{u}}{\partial t} + u_z \frac{\partial \mathbf{u}}{\partial z} &= \nu \frac{\partial^2 \mathbf{u}}{\partial z^2} \end{aligned} \quad (13.4.4)$$

Each component equation operates for the full time interval  $\Delta t$ , and the time at the end of each fractional step is reset to the initial value  $t^n$ . The three intermediate velocity fields, denoted by  $\mathbf{u}^{n+1/4}$ ,  $\mathbf{u}^{n+2/4}$ ,  $\mathbf{u}^{n+3/4} \equiv \mathbf{u}^*$ , are not generally solenoidal. For two-dimensional flow, only the first two steps in Eq. (13.4.4) are present, and the intermediate velocity fields are  $\mathbf{u}^{n+1/3}$  and  $\mathbf{u}^{n+2/3} \equiv \mathbf{u}^*$ .

The three steps in Eqs. (13.4.4) may be carried out using either the implicit BTCS method or the Crank-Nicolson method, with first- or second-order accuracy in time, respectively, both of which are unconditionally stable and require the relatively easy task of solving tridiagonal systems of equations. Setting the convection velocity in all three steps equal to the velocity at the beginning of the step, at the time level  $t^n$ , allows us to express the components of the convection-diffusion equations in conservative form. The boundary conditions for the intermediate velocity will be discussed shortly.

The projection step expressed by Eq. (13.4.3) is carried out using the BTCS method, yielding

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho} \nabla \phi^{n+1} \quad (13.4.5)$$

Computing the right-hand side of Eq. (13.4.5) requires a knowledge of the function  $\phi$  at the  $n + 1$  time level. This is obtained by taking the divergence of Eq. (13.4.5) and requiring that  $\nabla \cdot \mathbf{u}^{n+1} = 0$ , thereby deriving the following Poisson equation for  $\phi^{n+1}$

$$\nabla^2 \phi^{n+1} = \frac{\Delta t}{\rho} \nabla \cdot \mathbf{u}^* \quad (13.4.6)$$

which is to be solved subject to the boundary condition  $\nabla \phi^{n+1} \cdot \mathbf{n} = 0$ . As in the case of Poisson's equation for the pressure discussed in Section 13.3, the explicit implementation of this boundary

condition may be circumvented either by using a staggered grid and requiring that  $\nabla \cdot \mathbf{u}^{n+1}$  vanish at the pressure nodes that are adjacent to the boundaries, or by using a non-staggered grid and requiring that  $\nabla \cdot \mathbf{u}^{n+1}$  vanish at the nodes that are located at the boundaries.

Given an initial velocity field along with a prescribed boundary condition  $\mathbf{u} = \mathbf{U}$ , we compute the evolution of the flow according to the following steps:

1. Assign the initial values of the velocity field to the velocity nodes, and provide an estimate for  $\phi^{n+1}$  at the pressure nodes.
2. Advance the velocity field in a sequential manner according to the three equations (13.4.4) with boundary conditions: (a)  $\mathbf{u}^* \cdot \mathbf{n} = \mathbf{U} \cdot \mathbf{n}$  for the normal component of the velocity, and (b)  $\mathbf{u}^* \cdot (\mathbf{I} - \mathbf{nn}) = [\mathbf{U} + (\Delta t/\rho) \nabla \phi^{n+1}] \cdot (\mathbf{I} - \mathbf{nn})$  for the tangential component of the velocity. Here  $\mathbf{I}$  is the identity matrix and  $\mathbf{I} - \mathbf{nn}$  is the tangential projection operator.
3. Solve Poisson's equation (13.4.6) with boundary condition  $\nabla \phi^{n+1} \cdot \mathbf{n} = 0$ , or one of its modified versions. Since the flow rate of  $\mathbf{u}^*$  and thus of  $\nabla \phi$  across the boundaries vanishes, because of the boundary conditions required for  $\mathbf{u}^*$ , the compatibility condition is automatically fulfilled and Poisson's equation has an infinity of solutions.
4. Compute the velocity  $\mathbf{u}^{n+1}$  at all internal and boundary grid points according to Eq. (13.4.5). If the tangential boundary velocity is not equal to that required by the boundary conditions, that is, if there is a finite slip velocity, return to step 2 and repeat the computations with the new boundary distribution of  $\phi$ . Otherwise, proceed to step 5.
5. Set the time to  $t^{n+1}$ , return to step 2, and repeat the computations for another time step.

To further illustrate the implementation of the method on a non-staggered grid, consider the familiar problem of flow in a cavity driven by a moving lid depicted in Figure 13.1.1. The two convection-diffusion equations are integrated in time using an implicit method such as the Crank-Nicolson method. When integrating in the  $x$  direction, we use the boundary conditions

$$\begin{aligned} u_{1,j} &= 0, & v_{1,j} &= \frac{\Delta t}{\rho} \left( \frac{\partial \phi^{n+1}}{\partial y} \right)_{1,j} \\ u_{N+1,j} &= 0, & v_{N+1,j} &= \frac{\Delta t}{\rho} \left( \frac{\partial \phi^{n+1}}{\partial y} \right)_{N+1,j} \end{aligned} \quad (13.4.7)$$

at the side walls; boundary conditions over the upper and lower wall are not required. When integrating in the  $y$  direction, we use the boundary conditions

$$\begin{aligned} u_{i,1} &= \frac{\Delta t}{\rho} \left( \frac{\partial \phi^{n+1}}{\partial x} \right)_{i,1}, & v_{i,1} &= 0 \\ u_{i,M+1} &= U + \frac{\Delta t}{\rho} \left( \frac{\partial \phi^{n+1}}{\partial x} \right)_{i,M+1}, & v_{i,M+1} &= 0 \end{aligned} \quad (13.4.8)$$

at the upper and lower wall; boundary conditions over the side walls are not required.

A more advanced method of carrying out the fractional steps involves integrating the convection-diffusion equation (13.4.1) by means of the ADI method. This modification reduces the temporal error due to the spatial decoupling involved in Eqs. (13.4.4), and renders the convection-diffusion step second-order accurate in  $\Delta t$ .

### A Semi-Implicit Three-Level Method

Kim and Moin (1985) proposed advancing the velocity field according to the two fractional steps

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{1}{2}[3\mathbf{N}(\mathbf{u}^n) - \mathbf{N}(\mathbf{u}^{n-1})] + \frac{1}{2}\nu[\mathbf{L}(\mathbf{u}^*) + \mathbf{L}(\mathbf{u}^n)] \quad (13.4.9)$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla\phi^{n+1} \quad (13.4.10)$$

Equation (13.4.9) uses the explicit second-order Adams–Bashforth method for the nonlinear term and the implicit second-order Crank–Nicolson method for the viscous term. The projection function  $\phi$  is computed by solving Poisson's equation (13.4.6), which ensures that  $\mathbf{u}^{n+1}$  will be solenoidal at the end of a complete step.

It is instructive to solve Eq. (13.4.10) for  $\mathbf{u}^*$  and substitute the result back into Eq. (13.4.9). We thus find

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\frac{1}{\rho}\nabla(\phi^{n+1} + \frac{1}{2}\nu\Delta t\nabla^2\phi^{n+1}) + \frac{1}{2}(3\mathbf{N}^n - \mathbf{N}^{n-1}) + \frac{1}{2}\nu(\mathbf{L}(\mathbf{u}^n) + \mathbf{L}(\mathbf{u}^{n+1})) \quad (13.4.11)$$

which shows that the function  $\phi + \frac{1}{2}\nu\Delta t\nabla^2\phi$  plays the role of the pressure, thereby making a clear distinction between  $P$  and  $\phi$ . Defining  $\phi + (\Delta t/2)\nabla^2\phi$  at the  $n + 1$  time level to be equal to the pressure at the  $n + \frac{1}{2}$  level suggests that Eq. (13.4.10) is second-order accurate in  $\Delta t$ .

To carry out the convection–diffusion step, we rewrite Eq. (13.4.9) in the form

$$(1 - \frac{1}{2}\Delta t\nu\nabla^2)(\mathbf{u}^* - \mathbf{u}^n) = \frac{1}{2}\Delta t[3\mathbf{N}(\mathbf{u}^n) - \mathbf{N}(\mathbf{u}^{n-1})] + \Delta t\nu\nabla^2\mathbf{u}^n \quad (13.4.12)$$

and then factorize the operator on the left-hand side in an approximate manner to obtain

$$\begin{aligned} & \left(1 - \frac{1}{2}\Delta t\nu\frac{\partial^2}{\partial x^2}\right)\left(1 - \frac{1}{2}\Delta t\nu\frac{\partial^2}{\partial y^2}\right)\left(1 - \frac{1}{2}\Delta t\nu\frac{\partial^2}{\partial z^2}\right)(\mathbf{u}^* - \mathbf{u}^n) \\ &= \frac{1}{2}\Delta t[3\mathbf{N}(\mathbf{u}^n) - \mathbf{N}(\mathbf{u}^{n-1})] + \Delta t\nu\nabla^2\mathbf{u}^n \end{aligned} \quad (13.4.13)$$

To compute the difference  $\mathbf{u}^* - \mathbf{u}^n$ , we solve three tridiagonal systems of equations, which can be done using the efficient Thomas algorithm (Problem 13.4.2). The boundary conditions for  $\mathbf{u}^*$  are the same as those discussed previously for the first-order method. Poisson's equation (13.4.6) may be solved on a staggered grid in order to avoid the explicit implementation of boundary conditions for  $\phi$ .

### A Second-Order Method

Bell, Colella, and Glaz (1989) developed a projection method that is second-order accurate in  $\Delta t$ . The method involves introducing values for the velocity and pressure at the integral time levels  $n$  as well as at the intermediate time levels  $n + \frac{1}{2}$ , computing the auxiliary velocity  $\mathbf{u}^*$  on the basis of the semi-discrete version of the modified equation of motion

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mathbf{N}(\mathbf{u}^{n+1/2}) - \frac{1}{\rho}\nabla\psi + \frac{1}{2}\nu[\mathbf{L}(\mathbf{u}^*) + \mathbf{L}(\mathbf{u}^n)] \quad (13.4.14)$$

and then obtaining the velocity at the next time level by means of the projection

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho}\nabla\phi \quad (13.4.15)$$

where  $\psi$  and  $\phi$  are two auxiliary functions. Solving Eq. (13.4.15) for  $\mathbf{u}^*$  and substituting the result into Eq. (13.4.14) yields

$$\begin{aligned} \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} &= \mathbf{N}(\mathbf{u}^{n+1/2}) - \frac{1}{\rho}\nabla(\psi + \phi) + \frac{1}{2}\nu[\mathbf{L}(\mathbf{u}^{n+1}) + \mathbf{L}(\mathbf{u}^n)] \\ &+ \frac{1}{2}\mu\Delta t\nabla L(\phi) \end{aligned} \quad (13.4.16)$$

which represents a second-order discretization of the equation of motion provided that (1) we set  $P^{n+1/2} = \psi + \phi$  and (2) the last term on the right-hand side vanishes. The computational algorithm is designed so that progressively  $\psi$  tends to  $P^{n+1/2}$  and  $\phi$  tends to vanish, and is implemented according to the following steps:

1. Given the pressure  $P^{n-1/2}$  and the velocity  $\mathbf{u}^n$ , estimate  $\mathbf{u}^{n+1/2}$  by extrapolation and set  $\psi = P^{n-1/2}$ .
2. Calculate  $\mathbf{N}(\mathbf{u}^{n+1/2})$ , compute  $\mathbf{u}^*$  from Eq. (13.4.14), and call the solution  $\mathbf{u}^{*,k}$  where  $k$  is an inner iteration number; the first time, set  $k = 0$ .
3. Introduce the discrete form of the equation of motion

$$\frac{\mathbf{u}^{n+1,k} - \mathbf{u}^n}{\Delta t} = \mathbf{N}(\mathbf{u}^{n+1/2}) - \frac{1}{\rho}\nabla P^{n+1/2,k} + \frac{1}{2}\nu[\mathbf{L}(\mathbf{u}^{*,k}) + \mathbf{L}(\mathbf{u}^n)] \quad (13.4.17)$$

take its divergence, and require that  $\mathbf{u}^{n+1,k}$  be solenoidal to obtain a Poisson equation for  $P^{n+1/2,k}$ . Solve the Poisson equation subject to appropriate boundary conditions and then use Eq. (13.4.17) to compute  $\mathbf{u}^{n+1,k}$ .

4. Return to step 2, set  $\psi = P^{n+1/2,k}$ , and repeat the computations with  $\mathbf{u}^{n+1} = \mathbf{u}^{n+1,k}$ , where  $\mathbf{u}^{n+1/2}$  is computed by interpolation or extrapolation, increasing  $k$  by one.

### PROBLEMS

- 13.4.1 Solenoidal projection.** (a) Consider a nonsolenoidal rotational velocity field that is defined over the whole three-dimensional space and vanishes at infinity, and develop a procedure for removing the nonsolenoidal component while leaving the vorticity unaffected. (b) Consider part (a) for a bounded flow, and discuss the boundary conditions for the projection function.
- 13.4.2 Approximate factorization.** Write the three tridiagonal systems of equations corresponding to the factorized form of Eq. (13.4.13).



### Computer Problem

- 13.4.3 Flow in a rectangular cavity.** Write a program called *CV2DPRI* that uses the first-order projection method described in the text to compute the transient flow in a rectangular cavity due to the impulsive translation of a lid described in Problem 13.1.5. The Poisson equation should be solved using the routine of Problem 13.2.3.

### 13.5 METHODS OF MODIFIED DYNAMICS OR FALSE TRANSIENTS

All methods discussed in the preceding two sections involve (1) advancing the velocity field using a time-marching method that is suitable for parabolic differential equations, and (2) updating the pressure field or projection function by solving the elliptic Poisson equation. The origin of this dual procedure may be traced back to the absence of an evolution equation for the pressure in the original system of governing equations. The solution of the elliptic equation consumes much of the computational effort and inhibits the development of methods with second-order temporal accuracy. If all governing equations had the form of evolution equations, a simple time-marching method would suffice.

The idea behind the methods of *modified dynamics* is to amend either the continuity equation or the equation of motion with an objective to render all governing equations parabolic in time. In certain cases, the error introduced by modifying the original equations is mild, and the transient solution obtained by solving the modified problem describes the physical evolution with acceptable accuracy. In other cases, however, the transient evolution is purely fictitious and physically irrelevant, and hence it is significant only insofar as to provide us with a vehicle that will lead us to the steady state. The modified equations are designed so that their steady solutions satisfy the equations of steady incompressible Newtonian flow with a certain degree of accuracy.

### Artificial Compressibility Method for Steady Flow

One way to render the set of governing equations parabolic in time is to transform the continuity equation into an evolution equation for the pressure. In the artificial compressibility method introduced by Chorin (1967), this is achieved by replacing the continuity equation with the modified evolution equation

$$\frac{\partial P}{\partial t} + \frac{\rho}{\delta} \nabla \cdot \mathbf{u} = 0 \quad (13.5.1)$$

where  $\delta$  is a small positive constant called the *artificial compressibility*. Setting  $P = \rho/\delta$ , where  $\rho$  is the density of the fluid, makes Eq. (13.5.1) resemble the continuity equation for a compressible fluid. At steady state, the first term on the left-hand side of Eq. (13.5.1) vanishes, and this ensures that the steady solution satisfies the equations of steady incompressible flow.

### Non-staggered grid

On a non-staggered grid, the modified continuity equation (13.5.1) is discretized using the standard central-time-central-space method. At the nodes that lie on the boundaries, the divergence of the velocity is computed using central differences for the velocity component that is tangential to the boundary, and first- or second-order one-sided differences for the velocity component that is normal to the boundary. Boundary conditions for the pressure are not required.

To expedite the approach toward steady state, it is desirable to use a larger time step, but then stability issues require the use of a semi-implicit or fully implicit method. As an alternative, Chorin (1967) implemented a variant of the explicit CTCS method, modified according to the DuFort–Frankel scheme. The finite-difference equation corresponding to the  $x$  component of the equation of motion at the  $(i, j)$  grid point is

$$u_{i,j}^{n+1} = u_{i,j}^n + 2\Delta t \left( N_x(\mathbf{u}^n) + \nu \frac{u_{i+1,j}^n - 2\frac{1}{2}(u_{i,j}^{n+1} + u_{i,j}^{n-1}) + u_{i-1,j}^n}{\Delta x^2} \right. \\ \left. + \nu \frac{u_{i,j+1}^n - 2\frac{1}{2}(u_{i,j}^{n+1} + u_{i,j}^{n-1}) + u_{i,j-1}^n}{\Delta y^2} - \frac{1}{\rho} \frac{\partial P^n}{\partial x} \right) \quad (13.5.2)$$

The nonlinear term  $N$  is discretized in its conservative form using central differences. The  $y$  component of the equation of motion is discretized in an analogous fashion (Peyret and Taylor, 1983, p. 157). In Chapter 12 we saw that the modified differential equation corresponding to the DuFort–Frankel approximation is not always consistent with the original differential equation, the difference being a small term involving a second partial derivative with respect to time. The modified and original equations, however, agree at steady state.

The method may be extended in a straightforward manner to three-dimensional flow, where the DuFort–Frankel modification is applied to the three second spatial derivatives. Chorin (1967) found that, given boundary conditions for the velocity and provided that the flow remains subsonic with respect to the artificial speed of sound  $1/\delta^{1/2}$ , the method will be stable as long as the maximum value of the magnitude of the Courant number is less than  $2(\delta/n)^{1/2}/(1 + 5^{1/2})$ , where  $n = 2$  or  $3$  for two- and three-dimensional flow.

Other explicit or implicit implementations of the artificial compressibility method can be developed in a straightforward manner. Implementing explicit upwind methods for the convection

term at high Reynolds numbers and implicit methods for the viscous term, in particular, allows the use of large time steps and accelerates the approach toward steady state.

### Staggered grid

Next we consider the implementation of the method on the staggered grid shown in Figure 13.3.1. Adopting an explicit formulation, we advance the velocity at the  $x$  and  $y$  velocity nodes according to Eqs. (13.3.7) and (13.3.8) subject to the boundary conditions discussed in Section 13.3. To advance the pressure, we apply Eq. (13.5.1) at the pressure nodes and introduce the implicit BTCS discretization to obtain

$$P_{i,j}^{n+1} = P_{i,j}^n - \frac{\rho \Delta t}{\delta} \left( \frac{u_{i+1/2,j}^{n+1} - u_{i-1/2,j}^{n+1}}{\Delta x} + \frac{v_{i,j+1/2}^{n+1} - v_{i,j-1/2}^{n+1}}{\Delta y} \right) \quad (13.5.3)$$

One notable feature of the artificial compressibility method is that boundary conditions for the pressure are not required. When, however, steady state is reached, the pressure will satisfy the pressure Poisson equation with boundary conditions that result from projecting the equation of motion onto a unit vector that is normal or tangential to the boundaries.

### Modified PPE

Sotiropoulos and Abdallah (1990) modified the evolution equation for the rate of expansion (13.2.5), transforming it into an evolution equation for the pressure,

$$\frac{\partial P}{\partial t} = \frac{1}{\beta} \left( \nabla^2 P - \rho \nabla \cdot \mathbf{N}(\mathbf{u}) + \rho \frac{\partial \nabla \cdot \mathbf{u}}{\partial t} \right) \quad (13.5.4)$$

where  $\beta$  is a positive constant. At steady state, Eq. (13.5.4) reduces to the familiar Poisson equation for the pressure. The transient solution lacks a physical meaning, and Eq. (13.5.4) is significant only insofar as to provide us with a route toward the steady state.

### Penalty-Function Formulation

The penalty-function formulation uses an artificial constitutive equation for the pressure in terms of the rate of expansion,

$$P = -\frac{1}{\varepsilon} \nabla \cdot \mathbf{u} \quad (13.5.5)$$

where  $\varepsilon$  is a small positive constant (Temam, 1968). Since the pressure is an order-one variable, the rate of expansion is restricted to remain small at all times. Substituting Eq. (13.5.5) into the equation of motion yields a modified evolution equation for the velocity alone. In practice, in order to render the method stable, it is necessary to enhance this equation with the addition of a small term involving the rate of expansion. The governing equation of motion is

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{N}(\mathbf{u}) + \frac{1}{\rho \varepsilon} \nabla (\nabla \cdot \mathbf{u}) - \frac{1}{2} \mathbf{u} \nabla \cdot \mathbf{u} + \nu L(\mathbf{u}) \quad (13.5.6)$$

The computations proceed by integrating Eq. (13.5.6) forward in time from a given initial state subject to the specified velocity boundary conditions. The penalty-function formulation has found extensive applications predominantly in numerical procedures based on finite-element methods (Hughes, Liu, and Brooks, 1979).

## PROBLEM

- 13.5.1 Artificial compressibility method. (a) Write the equivalent of Eq. (13.5.2) for the  $x$  and  $y$  components of the velocity on a staggered grid. (b) Develop an ADI method for two-dimensional flow.

## References

- Abdallah, S., 1987, Numerical solutions for the incompressible Navier-Stokes equations in primitive variables using a non-staggered grid II. *J. Comp. Phys.* **70**, 193–202.
- Anderson, C. R., 1989, Vorticity boundary conditions and boundary vorticity generation for two-dimensional viscous incompressible flows. *J. Comp. Phys.* **80**, 72–97.
- Anderson, P. A., Tannehill, J. C., and Pletcher, R. H., 1984, *Computational Fluid Dynamics and Heat Transfer*. Taylor & Francis.
- Aragbesola, Y. A. S., and Burley, D. M., 1977, The vector and scalar potential method for the numerical solution of two- and three-dimensional Navier-Stokes equations. *J. Comp. Phys.* **24**, 398–415.
- Aziz, K., and Hellums, J. D., 1967, Numerical solution of the three-dimensional equations of motion for laminar natural convection. *Phys. Fluids* **10**, 314–24.
- Babu, V., and Korpeila, S., 1994, Numerical solution of the incompressible, three-dimensional Navier-Stokes equations. *Computers & Fluids* **23**, 675–91.
- Bell, J., Colella, P., and Glaz, H., 1989, A second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.* **85**, 257–83.
- Cebeci, T., 1982, *Numerical and Physical Aspects of Aerodynamic Flows*. Springer-Verlag.
- Chien, J. C., 1976, A general finite-difference formulation with applications to Navier-Stokes equations. *J. Comp. Phys.* **20**, 268–78.
- Chorin, A. J., 1967, A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.* **2**, 12–26.
- Chorin, A., 1968, Numerical solution of the Navier-Stokes equations. *Math. Comp.* **22**, 745–62.
- Daube, O., 1992, Resolution of the 2D Navier-Stokes equations in velocity-vorticity form by means of an influence matrix technique. *J. Comp. Phys.* **103**, 402–14.
- Dennis, S. C. R., Ingham, D. B., and Cook, R. N., 1979, Finite-difference methods for calculating steady incompressible flows in three dimensions. *J. Comp. Phys.* **33**, 325–39.
- Fletcher, C. A. J., 1988, *Computational Techniques for Fluid Dynamics*. 2 volumes. Springer-Verlag.
- Floryan, J. M., and Rasmussen, H., 1989, Numerical methods for viscous flows with moving boundaries. *Appl. Mech. Rev.* **42**, 323–41.
- Gatski, T. B., Grosch, C. E., and Rose, M. E., 1989, The numerical solution of the Navier-Stokes equations for 3-dimensional, unsteady, incompressible flows by compact schemes. *J. Comp. Phys.* **82**, 298–329.
- Ghia, K. N., Hankey, W. L., and Hodge, J. K., 1977, Study of incompressible Navier-Stokes equations in primitive variables using implicit numerical technique. *AIAA paper 77-648*, 156–65.
- Gresho, P. M., 1990, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: Theory. *Int. J. Num. Meth. Fluids* **11**, 587–620.
- Gresho, P. M., 1991, Incompressible fluid dynamics: some fundamental formulation issues. *Ann. Rev. Fluid Mech.* **23**, 413–53.
- Gresho, P. M., and Sani, R. L., 1987, On pressure boundary conditions for the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids* **7**, 1111–45.
- Guj, G., and Stella, F., 1988, Numerical solutions of high-Re recirculating flows in vorticity–velocity form. *I. J. Num. Meth. Fluids* **8**, 405–16.
- Guj, G., and Stella, F., 1993, A vorticity–velocity method for the numerical solution of 3D incompressible flows. *J. Comp. Phys.* **106**, 286–98.
- Gupta, M. M., 1991, High-accuracy solutions of incompressible Navier-Stokes equations. *J. Comp. Phys.* **93**, 343–59.
- Harlow, H. H., and Welch, J. E., 1965, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids* **8**, 2182–89.
- Hirasaki, G. J., and Hellums, J. D., 1970, Boundary conditions on the vector and scalar potentials in viscous three-dimensional hydrodynamics. *Quart. Appl. Math.* **28**, 293–96.
- Hirsch, C., 1988, *Numerical Computation of Internal and External Flows*. 2 volumes. Wiley.
- Hoffman, J. D., 1992, *Numerical Methods for Engineers and Scientists*. McGraw-Hill.
- Hoffman, K. A., and Chiang, S. T., 1993, *Computational Fluid Dynamics for Engineers*, Vols. I and II, Engineering Education System.
- Hughes, T. J. R., Liu, W. K., and Brooks, A., 1979, Finite element analysis of incompressible viscous flows by the penalty function formulation. *J. Comp. Phys.* **30**, 1–60.
- Israeli, M., 1972, On the evaluation of iteration parameters for the boundary vorticity. *Stud. Appl. Math.* **51**, 67–71.

- Kim, J., and Moin, P., 1985, Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.* **59**, 308–23.
- Mallinson, G. D., and de Vahl Davis, G., 1973, The method of false transient for the solution of coupled elliptic equations. *J. Comp. Phys.* **12**, 435–61.
- Mallinson, G. D., and de Vahl Davis, G., 1977, Three-dimensional natural convection in a box: a numerical study. *J. Fluid Mech.* **83**, 1–31.
- Mansour, M. L., and Hamed, A., 1990, Implicit solution of the incompressible Navier-Stokes equations on a non-staggered grid. *J. Comp. Phys.* **86**, 147–67.
- Orszag, S. A., and Israeli, M., 1974, Numerical simulation of viscous incompressible flows. *Annu. Rev. Fluid Mech.* **5**, 281–318.
- Orszag, S. A., Israeli, M., and Deville, M. O., 1986, Boundary conditions for incompressible flows. *J. Scient. Comp.* **1**, 75–111.
- Osswald, G. A., Ghia, K. N., and Ghia, U., 1987, A direct algorithm for solution of incompressible three-dimensional unsteady Navier-Stokes equations. *AIAA paper 87-1139*, 408–21.
- Peyret, R., and Taylor, T. D., 1983, *Computational Methods for Fluid Flow*. Springer-Verlag.
- Quartapelle, L., 1981, Vorticity conditioning in the computation of two-dimensional viscous flows. *J. Comp. Phys.* **40**, 453–77.
- Quartapelle, L., and Napolitano, M., 1986, Integral conditions for the pressure in the computation of incompressible viscous flows. *J. Comp. Phys.* **62**, 340–48.
- Quartapelle, L., and Valz-Gris, F., 1981, Projection conditions on the vorticity in viscous incompressible flows. *Int. J. Num. Meth. Fluids* **1**, 129–44.
- Roach, P. J., 1982, *Computational Fluid Dynamics*. Hermosa Publishers, P.O. Box 8172, Albuquerque, NM 87108.
- Richardson, S. M., and Cornish, A. R. H., 1977, Solution of three-dimensional incompressible flow problems. *J. Fluid Mech.* **82**, 109–319.
- Sheth, K., and Pozrikidis, C., 1995, Effects of inertia on the deformation of liquid drops in simple shear flow. *Computers & Fluids* **94**, 101–10.
- Sotiropoulos, F., and Abdallah, S., 1990, Coupled fully implicit solution procedure for the steady incompressible Navier-Stokes equation. *J. Comp. Phys.* **87**, 328–48.
- Sotiropoulos, F., and Abdallah, S., 1991, The discrete continuity equation in primitive variable solutions of incompressible flow. *J. Comp. Phys.* **95**, 212–27.
- Speziale, C. G., 1987, On the advantages of the velocity–vorticity formulation of the equations of fluid dynamics. *J. Comp. Phys.* **73**, 476–80.
- Temam, R., 1968, Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bull. Soc. Math. France*, **96**, 115–52.
- Trujillo, J. R., 1994, *Spectral element vorticity–velocity algorithm for the incompressible Navier-Stokes equations*. Doctoral dissertation, Princeton University.