CHAPTER
NINE

NUMERICAL METHODS FOR THE NAVIER-STOKES EQUATIONS

9.1 INTRODUCTION

For certain viscous flow problems, it is not possible to obtain an accurate solution using the simplified flow equations discussed in Chapters 6–8. Examples of such flow problems include shock–boundary-layer interactions, leading edge flows, certain wake flows, and other flows that involve strong viscous-inviscid interactions with large separated flow regions. For these cases, it becomes necessary to solve the complete set of Navier-Stokes (N-S) equations (or the Reynolds averaged form of these equations). Unfortunately, these equations are very complex and require a substantial amount of computer time and storage in order to obtain a solution. However, if the flow is incompressible, the equations can be simplified, and the required computer time is decreased accordingly.

Numerical schemes for solving the N-S equations are based on the same methods described in Chapter 6 for the Euler equations. Since the N-S equations consist of the Euler equations plus shear-stress and heat flux terms, the only change that is required is to discretize the additional terms in an appropriate manner. Because of the dissipative nature of the viscous terms, they are almost always discretized using central differences. One of the major differences that occurs when solving the N-S equations, as compared to the Euler equations, is the need to use fine meshes in order to properly resolve viscous layers. In many cases, this requirement will lead to meshes (cells) with large aspect ratios in the

viscous regions of the flow field. When the aspect ratio of the cells becomes large, a typical numerical scheme produces larger truncation errors (T.E.s), and the rate of convergence is decreased. Another consideration that must be taken into account when solving the N-S equations is the amount of numerical dissipation that is present in the numerical method. This numerical dissipation may be inherent in the scheme or could be explicitly added (i.e., smoothing terms). Obviously, the numerical dissipation should be substantially less than the actual physical dissipation in order to obtain an accurate solution.

The unsteady compressible N-S equations are a mixed set of hyperbolic-parabolic equations in time, while the unsteady incompressible N-S equations are a mixed set of elliptic-parabolic equations. As a consequence, different numerical techniques have been used in the past to solve the N-S equations in the compressible and incompressible flow regimes. These techniques are discussed in this chapter beginning with the techniques for solving the compressible N-S equations. Recently, methods have been developed to efficiently solve the compressible N-S equations at very low Mach numbers. With these methods, it becomes possible to solve both compressible and incompressible flows using the same approach. These methods are discussed in Section 9.2.6.

9.2 COMPRESSIBLE NAVIER-STOKES EQUATIONS

The compressible N-S equations in Cartesian coordinates without body forces or external heat addition can be written (see Section 5.1.6) as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{G}}{\partial z} = 0 \tag{9.1}$$

where **U**, **E**, **F**, and **G** are vectors given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix} \tag{9.2}$$

$$\mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ (E_t + p)u - u\tau_{xx} - v\tau_{xy} - w\tau_{xz} + q_x \end{bmatrix} \tag{9.3}$$

$$\mathbf{F} = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{yz} \\ (E_t + p)v - u\tau_{xy} - v\tau_{yy} - w\tau_{yz} + q_y \end{bmatrix} \tag{9.4}$$

$$\mathbf{G} = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ (E_t + p)w - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} + q_z \end{bmatrix} \tag{9.5}$$

and the components of the shear-stress tensor and heat flux vector are given by

$$\begin{aligned}
\tau_{xx} &= \frac{2}{3}\mu\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} - \frac{\partial w}{\partial z}\right) \\
\tau_{yy} &= \frac{2}{3}\mu\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} - \frac{\partial w}{\partial z}\right) \\
\tau_{zz} &= \frac{2}{3}\mu\left(2\frac{\partial w}{\partial z} - \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right) \\
\tau_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) = \tau_{yx} \\
\tau_{xz} &= \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right) = \tau_{zx} \\
\tau_{yz} &= \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) = \tau_{zy} \\
q_x &= -k\frac{\partial T}{\partial x} \\
q_y &= -k\frac{\partial T}{\partial y} \\
q_z &= -k\frac{\partial T}{\partial z}
\end{aligned} \tag{9.6}$$

These equations can be expressed in terms of a generalized orthogonal curvilinear coordinate system (x_1, x_2, x_3) using the formulas in Section 5.1.8. In addition, the compressible N-S equations can be written in terms of a generalized

nonorthogonal curvilinear coordinate system (ξ, η, ζ) using the general transformation described in Section 5.6.2:

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{9.7}$$

The transformed equations are given in Chapter 8 as Eqs. (8.34)–(8.36).

The thin-layer approximation to the compressible N-S equations is discussed in Section 8.2. This approximation allows one to drop a number of terms from the complete N-S equations. However, the mathematical character of the resulting equations is identical to that of the complete N-S equations, and as a result, the two sets of equations are normally solved in the same manner. The thin-layer N-S equations are given in Chapter 8 for a Cartesian coordinate system [Eqs. (8.2)–(8.6)] and for a general nonorthogonal coordinate system [Eqs. (8.9)–(8.12)].

For turbulent flows, it is convenient to use the Reynolds averaged equations instead of the N-S equations. Employing the Boussinesq approximation (see Section 5.4.2), the N-S equations can be changed to a modeled form of the Reynolds averaged equations by replacing the coefficient of viscosity μ with

$$\mu + \mu_T$$

and by also replacing the coefficient of thermal conductivity k with

$$k + k_T$$

where μ_T is the eddy viscosity and k_T is the turbulent thermal conductivity. The turbulent thermal conductivity can be expressed in terms of the eddy viscosity using the turbulent Prandtl number Pr_T:

$$k_T = \frac{c_p \mu_T}{\mathrm{Pr}_T} \tag{9.8}$$

Techniques for determining μ_T are described in detail in Section 5.4.

As mentioned previously, the unsteady compressible N-S equations are a mixed set of hyperbolic-parabolic equations in time. If the unsteady terms are dropped from these equations, the resulting equations become a mixed set of hyperbolic-elliptic equations, which are difficult to solve because of the differences in numerical techniques required for hyperbolic and elliptic types of equations. As a consequence, most solutions of the compressible N-S equations have employed the unsteady form of the equations. The steady-state solution is obtained by marching the solution in time (or pseudo-time) until convergence is achieved. This procedure is called the *time-dependent approach* and is the method that will be discussed in this chapter for solving the compressible N-S equations.

Both explicit and implicit schemes have been used with the time-dependent approach to solve the compressible N-S equations. Nearly all of these methods are at least second-order accurate in space and are either first- or second-order

accurate in "time." If an accurate time evolution of the flow is required, the numerical scheme should at least be second-order accurate in time. On the other hand, if only the steady-state solution is desired, it is often advantageous to employ a scheme that is not time accurate, since the steady-state solution can usually be achieved with fewer time steps (iterations). Because of the added complexity, only a handful of third-order (or higher) time-accurate methods have appeared in the literature to solve the compressible N-S equations. Many feel that a second-order method is the optimum choice, since higher-order accuracy is at the expense of more computer time. For a complete review of nearly all papers that report solutions to the compressible N-S equations prior to 1976, the reader is urged to consult the excellent survey paper of Peyret and Viviand (1975). More recent reviews are given by MacCormack (1985, 1993). We will now begin our detailed discussion of methods for solving the compressible N-S equations.

9.2.1 Explicit MacCormack Method

When the original MacCormack (1969) scheme is applied to the 3-D compressible N-S equations given by Eq. (9.1), the following algorithm results.

Predictor:

$$\mathbf{U}_{i,j,k}^{\overline{n+1}} = \mathbf{U}_{i,j,k}^{n} - \frac{\Delta t}{\Delta x}(\mathbf{E}_{i+1,j,k}^{n} - \mathbf{E}_{i,j,k}^{n}) - \frac{\Delta t}{\Delta y}(\mathbf{F}_{i,j+1,k}^{n} - \mathbf{F}_{i,j,k}^{n}) - \frac{\Delta t}{\Delta z}(\mathbf{G}_{i,j,k+1}^{n} - \mathbf{G}_{i,j,k}^{n}) \tag{9.9}$$

Corrector:

$$\mathbf{U}_{i,j,k}^{n+1} = \frac{1}{2}\left[\mathbf{U}_{i,j,k}^{n} + \mathbf{U}_{i,j,k}^{\overline{n+1}} - \frac{\Delta t}{\Delta x}\left(\mathbf{E}_{i,j,k}^{\overline{n+1}} - \mathbf{E}_{i-1,j,k}^{\overline{n+1}}\right) - \frac{\Delta t}{\Delta y}\left(\mathbf{F}_{i,j,k}^{\overline{n+1}} - \mathbf{F}_{i,j-1,k}^{\overline{n+1}}\right) - \frac{\Delta t}{\Delta z}\left(\mathbf{G}_{i,j,k}^{\overline{n+1}} - \mathbf{G}_{i,j,k-1}^{\overline{n+1}}\right)\right] \tag{9.10}$$

where $x = i\,\Delta x$, $y = j\,\Delta y$, and $z = k\,\Delta z$. This explicit scheme is second-order accurate in both space and time. In the present form of this scheme, forward differences are used for all spatial derivatives in the predictor step, while backward differences are used in the corrector step. The forward and backward differencing can be alternated between predictor and corrector steps as well as between the three spatial derivatives in a sequential fashion. This eliminates any bias due to the one-sided differencing. An example of a suitable sequence is given in Table 9.1.

The derivatives appearing in the viscous terms of **E**, **F**, and **G** must be differenced correctly in order to maintain second-order accuracy. This is accomplished in the following manner. The x derivative terms appearing in **E**

Table 9.1 Differencing sequence for MacCormack scheme

	Predictor			Corrector		
Step	x Derivative	y Derivative	z Derivative	x Derivative	y Derivative	z Derivative
1	F	F	F	B	B	B
2	B	B	F	F	F	B
3	F	F	B	B	B	F
4	B	F	B	F	B	F
5	F	B	F	B	F	B
6	B	F	F	F	B	B
7	F	B	B	B	F	F
8	B	B	B	F	F	F
9	F	F	F	B	B	B
.	.	.	.	.	.	.
.	.	.	.	.	.	.

F, forward difference; B, backward difference.

are differenced in the opposite direction to that used for $\partial \mathbf{E}/\partial x$, while the y derivatives and the z derivatives are approximated with central differences. Likewise, the y derivative terms appearing in $\mathbf{F}$ and the z derivative terms appearing in $\mathbf{G}$ are differenced in the opposite direction to that used for $\partial \mathbf{F}/\partial y$ and $\partial \mathbf{G}/\partial z$, respectively, while the cross-derivative terms in $\mathbf{F}$ and $\mathbf{G}$ are approximated with central differences. For example, consider the following term in $\mathbf{F}$, which corresponds to the x-momentum equation:

$$F_2 = \rho uv - \mu \frac{\partial u}{\partial y} - \mu \frac{\partial v}{\partial x} \tag{9.11}$$

In the predictor step, given by Eq. (9.9), this term in $\mathbf{F}^n_{i,j,k}$ is differenced as

$$(F_2)^n_{i,j,k} = (\rho uv)^n_{i,j,k} - \mu^n_{i,j,k} \frac{u^n_{i,j,k} - u^n_{i,j-1,k}}{\Delta y} - \mu^n_{i,j,k} \frac{v^n_{i+1,j,k} - v^n_{i-1,j,k}}{2\,\Delta x} \tag{9.12}$$

while in the corrector step, given by Eq. (9.10), this term in $\mathbf{F}^{\overline{n+1}}_{i,j-1,k}$ is differenced as

$$(F_2)^{\overline{n+1}}_{i,j-1,k} = (\rho uv)^{\overline{n+1}}_{i,j-1,k} - \mu^{\overline{n+1}}_{i,j-1,k} \frac{u^{\overline{n+1}}_{i,j,k} - u^{\overline{n+1}}_{i,j-1,k}}{\Delta y} - \mu^{\overline{n+1}}_{i,j-1,k} \frac{v^{\overline{n+1}}_{i+1,j-1,k} - v^{\overline{n+1}}_{i-1,j-1,k}}{2\,\Delta x} \tag{9.13}$$

Because of the complexity of the compressible N-S equations, it is not possible to obtain a closed-form stability expression for the MacCormack

scheme applied to these equations. However, the following empirical formula (Tannehill et al., 1975) can normally be used:

$$\Delta t \leqslant \frac{\sigma(\Delta t)_{\text{CFL}}}{1 + 2/\text{Re}_\Delta} \tag{9.14}$$

where σ is the safety factor ($\cong 0.9$), $(\Delta t)_{\text{CFL}}$ is the inviscid Courant-Friedrichs-Levy (CFL) condition (MacCormack, 1971)

$$(\Delta t)_{\text{CFL}} \leqslant \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + a\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}} \right)^{-1} \tag{9.15}$$

Re_Δ is the minimum mesh Reynolds number given by

$$\text{Re}_\Delta = \min(\text{Re}_{\Delta x}, \text{Re}_{\Delta y}, \text{Re}_{\Delta z}) \tag{9.16}$$

where

$$\begin{aligned} \text{Re}_{\Delta x} &= \frac{\rho|u|\,\Delta x}{\mu} \\ \text{Re}_{\Delta y} &= \frac{\rho|v|\,\Delta y}{\mu} \\ \text{Re}_{\Delta z} &= \frac{\rho|w|\,\Delta z}{\mu} \end{aligned} \tag{9.17}$$

and a is the local speed of sound,

$$a = \sqrt{\frac{\gamma p}{\rho}}$$

Before each step, Δt can be computed for each grid point using Eq. (9.14). The smallest value of Δt is then used to advance the solution over the entire mesh. If only the steady-state solution is desired, Li (1973) has suggested that the solution at each point be advanced using the maximum possible Δt, as computed from Eq. (9.14), in order to accelerate the convergence of the solution. This procedure is referred to as *local time stepping*. In addition, multigrid procedures (see Section 4.3.5) can also be used to accelerate the convergence of N-S calculations.

After each predictor or corrector step, the primitive variables (ρ, u, v, w, e, p, T) can be found by "decoding" the **U** vector,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E_t \end{bmatrix} = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \\ U_5 \end{bmatrix} \tag{9.18}$$

in the following manner

$$\begin{aligned} \rho &= U_1 \\ u &= \frac{U_2}{U_1} \\ v &= \frac{U_3}{U_1} \\ w &= \frac{U_4}{U_1} \\ e &= \frac{U_5}{U_1} - \frac{u^2 + v^2 + w^2}{2} \\ p &= p(\rho, e) \\ T &= T(\rho, e) \end{aligned} \tag{9.19}$$

MacCormack (1971) modified his original method by incorporating time splitting into the scheme. This revised method, which was applied to the viscous Burgers equation in Section 4.5.8, "splits" the original MacCormack scheme into a sequence of one-dimensional (1-D) operations. As a result, the stability condition is based on a 1-D scheme that is less restrictive than the original 3-D scheme. Thus it becomes possible to advance the solution in each direction with the maximum possible time step. This is particularly advantageous if the allowable time steps $(\Delta t_x, \Delta t_y, \Delta t_z)$ are much different because of large differences in the mesh spacings $(\Delta x, \Delta y, \Delta z)$. In order to apply this algorithm to Eq. (9.1), we define the 1-D difference operators $L_x(\Delta t_x)$, $L_y(\Delta t_y)$, and $L_z(\Delta t_z)$ in the following manner. The $L_x(\Delta t_x)$ operator applied to $\mathbf{U}^*_{i,j,k}$,

$$\mathbf{U}^{**}_{i,j,k} = L_x(\Delta t_x)\mathbf{U}^*_{i,j,k} \tag{9.20}$$

is equivalent to the two-step formula

$$\begin{aligned} \overline{U^{**}_{i,j,k}} &= \mathbf{U}^*_{i,j,k} - \frac{\Delta t_x}{\Delta x}(\mathbf{E}^*_{i+1,j,k} - \mathbf{E}^*_{i,j,k}) \\ \mathbf{U}^{**}_{i,j,k} &= \frac{1}{2}\left[\mathbf{U}^*_{i,j,k} + \overline{\mathbf{U}^{**}_{i,j,k}} - \frac{\Delta t_x}{\Delta x}\left(\overline{\mathbf{E}^{**}_{i,j,k}} - \overline{\mathbf{E}^{**}_{i-1,j,k}}\right)\right] \end{aligned} \tag{9.21}$$

These expressions make use of the dummy time indices * and **. The $L_y(\Delta t_y)$ and $L_z(\Delta t_z)$ operators are defined in a similar manner. That is, the $L_y(\Delta t_y)$ operator applied to $\mathbf{U}^*_{i,j,k}$,

$$\mathbf{U}^{**}_{i,j,k} = L_y(\Delta t_y)\mathbf{U}^*_{i,j,k} \tag{9.22}$$

is equivalent to

$$\begin{aligned}\mathbf{U}^{\overline{**}}_{i,j,k} &= \mathbf{U}^{*}_{i,j,k} - \frac{\Delta t_y}{\Delta y}(\mathbf{F}^{*}_{i,j+1,k} - \mathbf{F}^{*}_{i,j,k}) \\ \mathbf{U}^{**}_{i,j,k} &= \frac{1}{2}\left[\mathbf{U}^{*}_{i,j,k} + \mathbf{U}^{\overline{**}}_{i,j,k} - \frac{\Delta t_y}{\Delta y}\left(\mathbf{F}^{\overline{**}}_{i,j,k} - \mathbf{F}^{\overline{**}}_{i,j-1,k}\right)\right]\end{aligned} \tag{9.23}$$

and the $L_z(\Delta t_z)$ operator applied to $\mathbf{U}^{*}_{i,j,k}$,

$$\mathbf{U}^{**}_{i,j,k} = L_z(\Delta t_z)\mathbf{U}^{*}_{i,j,k} \tag{9.24}$$

is equivalent to

$$\begin{aligned}\mathbf{U}^{\overline{**}}_{i,j,k} &= \mathbf{U}^{*}_{i,j,k} - \frac{\Delta t_z}{\Delta z}(\mathbf{G}^{*}_{i,j,k+1} - \mathbf{G}^{*}_{i,j,k}) \\ \mathbf{U}^{**}_{i,j,k} &= \frac{1}{2}\left[\mathbf{U}^{*}_{i,j,k} + \mathbf{U}^{\overline{**}}_{i,j,k} - \frac{\Delta t_z}{\Delta z}\left(\mathbf{G}^{\overline{**}}_{i,j,k} - \mathbf{G}^{\overline{**}}_{i,j,k}\right)\right]\end{aligned} \tag{9.25}$$

As mentioned in Section 4.5.8, a sequence of operators is consistent if the sums of the time steps for each of the operators are equal and is second-order accurate if the sequence is symmetric. A sequence that satisfies these criteria and is applicable to Eq. (9.1) is given by

$$\mathbf{U}^{n+2}_{i,j,k} = L_x(\Delta t_x)L_y(\Delta t_y)L_z(\Delta t_z)L_z(\Delta t_z)L_y(\Delta t_y)L_x(\Delta t_x)\mathbf{U}^{n}_{i,j,k} \tag{9.26}$$

Another sequence that satisfies these criteria, and is applicable when $\Delta y \ll \min(\Delta x, \Delta z)$, is given by

$$\mathbf{U}^{n+2}_{i,j,k} = L_x(\Delta t_x)\left[L_y\left(\frac{\Delta t_y}{m}\right)\right]^m L_z(\Delta t_z)L_z(\Delta t_z)\left[L_y\left(\frac{\Delta t_y}{m}\right)\right]^m L_x(\Delta t_x)\mathbf{U}^{n}_{i,j,k} \tag{9.27}$$

where m is an integer.

The algorithms resulting from a sequence of operators such as Eqs. (9.26) and (9.27) are stable if the time step size in the argument of each does not exceed the maximum allowed for that operator. Since it is not possible to analyze the stability of each operator applied to the complete N-S equations, a 1-D form of the empirical stability formula, given by Eq. (9.14), can be used for each operator:

$$\begin{aligned}\Delta t_x &\leqslant \frac{\sigma \Delta x}{(|u| + a)(1 + 2/\mathrm{Re}_{\Delta x})} \\ \Delta t_y &\leqslant \frac{\sigma \Delta y}{(|v| + a)(1 + 2/\mathrm{Re}_{\Delta y})} \\ \Delta t_z &\leqslant \frac{\sigma \Delta z}{(|w| + a)(1 + 2/\mathrm{Re}_{\Delta z})}\end{aligned} \tag{9.28}$$

where σ is the safety factor and a is the local speed of sound.

Computations involving the compressible N-S equations sometimes become unstable (i.e., "blow up") because of numerical oscillations. These oscillations are the result of inadequate mesh refinement in regions of large gradients such as shock waves and are accentuated when central differences are used for the spatial derivatives, as in the present MacCormack scheme. In many cases, it is impractical to refine the mesh in these regions, particularly if they are far removed from the region of interest. For such situations, MacCormack and Baldwin (1975) have devised a "product" fourth-order smoothing scheme, which is an alternative to the fourth-order type of smoothing given by Eq. (8.98). In the MacCormack type of smoothing, dissipation terms are added to each operator. For example, they are added to the $L_x(\Delta t_x)$ operator in the following manner:

$$\begin{aligned}
\mathbf{U}^{\overline{**}}_{i,j,k} &= \mathbf{U}^{*}_{i,j,k} - \frac{\Delta t_x}{\Delta x}\left(\mathbf{E}^{*}_{i+1,j,k} + \mathbf{S}^{*}_{i+1,j,k} - \mathbf{E}^{*}_{i,j,k} - \mathbf{S}^{*}_{i,j,k}\right) \\
\mathbf{U}^{**}_{i,j,k} &= \frac{1}{2}\left[\mathbf{U}^{*}_{i,j,k} + \mathbf{U}^{\overline{**}}_{i,j,k} - \frac{\Delta t_x}{\Delta x}\left(\mathbf{E}^{\overline{**}}_{i,j,k} + \mathbf{S}^{\overline{**}}_{i,j,k} - \mathbf{E}^{\overline{**}}_{i-1,j,k} - \mathbf{S}^{\overline{**}}_{i-1,j,k}\right)\right]
\end{aligned} \tag{9.29}$$

where

$$\begin{aligned}
\mathbf{S}^{*}_{i,j,k} &= \epsilon_e\left[\left(|u^{*}_{i,j,k}| + a^{*}_{i,j,k}\right)\frac{|\delta_x^2 p^{*}_{i,j,k}|}{\left(p^{*}_{i+1,j,k} + 2p^{*}_{i,j,k} + p^{*}_{i-1,j,k}\right)}\left(\mathbf{U}^{*}_{i,j,k} - \mathbf{U}^{*}_{i-1,j,k}\right)\right] \\
\mathbf{S}^{\overline{**}}_{i,j,k} &= \epsilon_e\left[\left(|u^{\overline{**}}_{i,j,k} + a^{\overline{**}}_{i,j,k}\right)\frac{|\delta_x^2 p^{\overline{**}}_{i,j,k}|}{\left(p^{\overline{**}}_{i+1,j,k} + 2p^{\overline{**}}_{i,j,k} + p^{\overline{**}}_{i-1,j,k}\right)}\left(\mathbf{U}^{\overline{**}}_{i+1,j,k} - \mathbf{U}^{\overline{**}}_{i,j,k}\right)\right]
\end{aligned} \tag{9.30}$$

and $0 \leqslant \epsilon_e \leqslant 0.5$ for stability. Thus an artificial viscosity term of the form

$$\epsilon_e(\Delta x)^4 \frac{\partial}{\partial x}\left(\frac{|u| + a}{4p}\left|\frac{\partial^2 p}{\partial x^2}\right|\frac{\partial \mathbf{U}}{\partial x}\right) \tag{9.31}$$

has been added to the N-S equations. This smoothing term has a very small magnitude except in regions of pressure oscillations, where the T.E. is already producing erroneous results.

The explicit MacCormack algorithm is a suitable method for solving both steady and unsteady flows at moderate to low Reynolds numbers. However, it is not a satisfactory method for solving high Reynolds numbers flows, where the viscous regions become very thin. For these flows, the mesh must be highly refined in order to accurately resolve the viscous regions. This leads to small time steps and, subsequently, long computing times if an explicit scheme such as the MacCormack method is used. In order to explain this further, let us consider the 2-D flow over a flat plate at high Reynolds number. In this case, a very fine mesh is required near the flat plate in order to resolve the boundary layer, but a coarser grid can be used in the inviscid portion of the flow field, as illustrated in Fig. 9.1. In the coarse grid region, the MacCormack time-split scheme can be

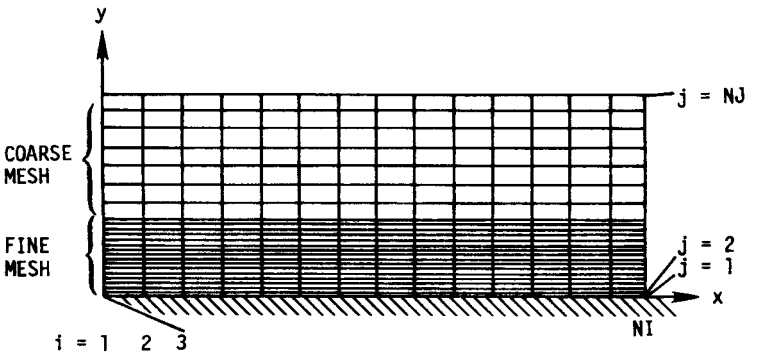

Figure 9.1 Mesh for high Reynolds number flow over a flat plate.

applied in the following manner:

$$\mathbf{U}_{i,j}^{n+1} = L_x\left(\frac{\Delta t}{2}\right) L_y(\Delta t) L_x\left(\frac{\Delta t}{2}\right) \mathbf{U}_{i,j}^{n} \tag{9.32}$$

where

$$\Delta t \leqslant \min\left(2\,\Delta t_x, \Delta t_y\right)_{\text{coarse mesh}} \tag{9.33}$$

In the fine grid region the following sequence of operators can be used:

$$\mathbf{U}_{i,j}^{n+1} = \left[L_y\left(\frac{\Delta t}{2m}\right) L_x\left(\frac{\Delta t}{m}\right) L_y\left(\frac{\Delta t}{2m}\right)\right]^m \mathbf{U}_{i,j} \tag{9.34}$$

where m is the smallest integer such that

$$\frac{\Delta t}{m} \leqslant \min\left(\Delta t_x, 2\,\Delta t_y\right)_{\text{fine mesh}} \tag{9.35}$$

For high Reynolds numbers, the fine-grid region becomes very thin, requiring Δy to be very small. This causes Δt_y in the L_y operator to be very small and the integer m to be very large. Consequently, a substantial amount of calculation time is required in the fine-grid region. To overcome this difficulty, MacCormack (1976) developed a hybrid version of his scheme, which is known as the MacCormack rapid solver method. This hybrid method is part explicit and part implicit. For the flat-plate problem described above, the rapid solver method is implemented by replacing the $L_y(\Delta t/2m)$ operator in Eq. (9.34) with

$$L_{y_H}\left(\frac{\Delta t}{2m}\right) L_{y_P}\left(\frac{\Delta t}{2m}\right)$$

where the L_{y_H} operator is applied to the inviscid (hyperbolic) portion of the N-S equations, i.e.,

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_H}{\partial y} = 0 \tag{9.36}$$

with $\mathbf{F}_H$ defined as

$$\mathbf{F}_H = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_t + p)v \end{bmatrix} \tag{9.37}$$

The L_{H_P} operator is applied to the viscous (parabolic) portion of the N-S equations:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_P}{\partial y} = 0 \tag{9.38}$$

where $\mathbf{F}_P = \mathbf{F} - \mathbf{F}_H$. The L_{y_H} operator solves Eq. (9.36) using either the method of characteristics or the original MacCormack scheme (Li, 1977; Shang, 1977). The L_{y_P} operator solves Eq. (9.38) using an implicit scheme such as the Crank-Nicolson or Laasonen schemes. Thus it is possible to solve Eq. (9.36) and (9.38) using a time step that is not limited by the viscosity stability constraint. The rapid solver method has proved to be from 10 to 100 times faster than the time-split scheme for high Reynolds number flows. However, because of its complexity, more traditional implicit schemes (see Sections 9.2.3 and 9.2.4) are preferred when solving high Reynolds number flows.

9.2.2 Other Explicit Methods

In addition to the MacCormack scheme, other explicit methods that have been used to solve the compressible N-S equations include the following:

1. Hopscotch method (Section 4.2.13)
2. Leap frog/DuFort-Frankel method (Section 4.5.2)
3. Brailovskaya method (Section 4.5.3)
4. Allen-Cheng method (Section 4.5.4)
5. Lax-Wendroff method (Section 4.5.5)
6. Runge-Kutta method (Section 4.1.13)

These methods were discussed in earlier sections (as indicated), where they were applied to model equations. When these methods are applied to the more complicated compressible N-S equations, certain difficulties can arise, as we have seen before. For example, the mixed-derivative terms create a problem for the hopscotch method. If these terms are differenced in the usual manner by applying Eq. (3.51), the hopscotch method is no longer explicit, since a matrix solution is required. This problem can be circumvented by lagging the mixed-derivative terms (i.e., evaluating them at the previous time level).

All of the above methods, except the Lax-Wendroff and Runge-Kutta schemes, are first-order accurate in time, so that they cannot be used to accurately compute the time evolution of a flow field. In addition, all of the methods have a stability restriction that limits the maximum time step. However, the stability conditions for the hopscotch and Allen-Cheng methods are independent of the viscosity, which gives them an advantage over the other methods. The allowable time step for the hopscotch method is given by the inviscid CFL condition, which for a 2-D problem becomes

$$(\Delta t)_{\text{CFL}} \leqslant \frac{\Delta x}{|u| + |v| + 2\sqrt{a}} \tag{9.39}$$

if $\Delta x = \Delta y$. An advantage of the Brailovskaya method is that the viscous terms need to be computed only once during the two-step procedure.

Of the six explicit methods listed above, only the Runge-Kutta method is widely used to solve the compressible N-S equations. The Runge-Kutta method, developed by Jameson et al. (1981, 1983) to solve the Euler equations, has been extended to the N-S equations by Swanson and Turkel (1985, 1987), Martinelli et al. (1986), Martinelli and Jameson (1988), and Turkel et al. (1991). This method utilizes a blend of second- and fourth-order damping terms and employs local time stepping, implicit residual smoothing, and multigrid to accelerate the convergence to steady state. The Runge-Kutta method is employed in the computer code TLNS3D (thin layer N-S program for 3-D flows) developed at NASA Langley Research Center (Vatsa and Wedan, 1989).

9.2.3 Beam-Warming Scheme

Prior to the mid-1970s, the numerical methods available to solve the compressible N-S equations were nearly all explicit and were limited in time step size by the CFL condition. As a consequence, it was difficult to compute high Reynolds number flows because of the fine meshes required to resolve the viscous regions. This difficulty was overcome with the application of noniterative implicit methods to the N-S equations. Briley and McDonald (1974) and Beam and Warming (1978) were the first to apply this type of scheme to solve the compressible N-S equations. We will describe the Beam-Warming scheme in this section.

The Beam-Warming numerical scheme for solving the compressible N-S equations belongs to the same class of alternating direction implicit (ADI) schemes developed by Lindemuth and Killeen (1973) and McDonald and Briley (1975). Under certain conditions, these schemes can be shown to be equivalent. The Briley-McDonald scheme is discussed in Section 4.5.7, where it is applied to the viscous Burgers equation.

For simplicity, we will apply the Beam-Warming finite-difference scheme to the 2-D compressible N-S equations, which can be written in the following

vector form:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial y} = \frac{\partial \mathbf{V}_1(\mathbf{U}, \mathbf{U}_x)}{\partial x} + \frac{\partial \mathbf{V}_2(\mathbf{U}, \mathbf{U}_y)}{\partial x} + \frac{\partial \mathbf{W}_1(\mathbf{U}, \mathbf{U}_x)}{\partial y} + \frac{\partial \mathbf{W}_2(\mathbf{U}, \mathbf{U}_y)}{\partial y} \tag{9.40}$$

where

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix} \quad \mathbf{E}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E_t + p)u \end{bmatrix} \quad \mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E_t + p)v \end{bmatrix}$$

$$\mathbf{V}_1 + \mathbf{V}_2 = \begin{bmatrix} 0 \\ \frac{2}{3}\mu(2u_x - v_y) \\ \mu(u_y + v_x) \\ \mu v(u_y + v_x) + \frac{2}{3}\mu u(2u_x - v_y) + kT_x \end{bmatrix} \tag{9.41}$$

$$\mathbf{W}_1 + \mathbf{W}_2 = \begin{bmatrix} 0 \\ \mu(u_y + v_x) \\ \frac{2}{3}\mu(2v_y - u_x) \\ \mu u(u_y + v_x) + \frac{2}{3}\mu v(2v_y - u_x) + kT_y \end{bmatrix}$$

In the Beam-Warming scheme the solution is marched in time using the following difference formula:

$$\Delta^n \mathbf{U} = \frac{\theta_1 \Delta t}{1 + \theta_2} \frac{\partial}{\partial t}(\Delta^n \mathbf{U}) + \frac{\Delta t}{1 + \theta_2} \frac{\partial}{\partial t}(\mathbf{U}^n) + \frac{\theta_2}{1 + \theta_2} \Delta^{n-1} \mathbf{U} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta t)^2 + (\Delta t)^3\right] \tag{9.42}$$

where $\Delta^n \mathbf{U} = \mathbf{U}^{n+1} - \mathbf{U}^n$. This general difference formula, with the appropriate choice of the parameters θ_1 and θ_2, represents many of the standard difference schemes, as we have seen from Section 8.3.3. For the compressible N-S equations, either the Euler implicit scheme ($\theta_1 = 1, \theta_2 = 0$), which is first-order accurate in time, or the three-point backward implicit scheme ($\theta_1 = 1, \theta_2 = \frac{1}{2}$), which is second-order accurate in time, is normally used.

After substituting Eq. (9.40) into Eq. (9.42), we obtain

$$\Delta^n \mathbf{U} = \frac{\theta_1 \Delta t}{1+\theta_2}\left[\frac{\partial}{\partial x}(-\Delta^n \mathbf{E} + \Delta^n \mathbf{V}_1 + \Delta^n \mathbf{V}_2) + \frac{\partial}{\partial y}(-\Delta^n \mathbf{F} + \Delta^n \mathbf{W}_1 + \Delta^n \mathbf{W}_2)\right] + \frac{\Delta t}{1+\theta_2}\left[\frac{\partial}{\partial x}(-\mathbf{E}^n + \mathbf{V}_1^n + \mathbf{V}_2^n) + \frac{\partial}{\partial y}(-\mathbf{F}^n + \mathbf{W}_1^n + \mathbf{W}_2^n)\right] + \frac{\theta_2}{1+\theta_2}\Delta^{n-1}\mathbf{U} + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta t)^2 + (\Delta t)^3\right] \tag{9.43}$$

This difference formula is in the so-called delta form, which is discussed in Section 4.4.7. The delta terms are linearized using truncated Taylor-series expansions. For example, $\Delta^n \mathbf{E}$ is linearized using

$$\mathbf{E}^{n+1} = \mathbf{E}^n + \left(\frac{\partial \mathbf{E}}{\partial \mathbf{U}}\right)^n (\mathbf{U}^{n+1} - \mathbf{U}^n) + O[(\Delta t)^2] \tag{9.44}$$

which can be rewritten as

$$\Delta^n \mathbf{E} = [A]^n \Delta^n \mathbf{U} + O[(\Delta t)^2] \tag{9.45}$$

where $[A]$ is the Jacobian matrix $\partial \mathbf{E}/\partial \mathbf{U}$ given by

$$[A] = -\left[\begin{array}{c|c|c|c} 0 & -1 & 0 & 0 \\ \dfrac{3-\gamma}{2}u^2 + \dfrac{1-\gamma}{2}v^2 & (\gamma-3)u & (\gamma-1)v & (1-\gamma) \\ uv & -v & -u & 0 \\ \dfrac{\gamma E_t u}{\rho} + (1-\gamma)u(u^2+v^2) & -\dfrac{\gamma E_t}{\rho} + \dfrac{\gamma-1}{2}(3u^2+v^2) & (\gamma-1)uv & -\gamma u \end{array}\right] \tag{9.46}$$

and γ is the ratio of specific heats. This Jacobian matrix is derived assuming a perfect gas. In a like manner, $\Delta^n \mathbf{F}$ can be linearized as

$$\Delta^n \mathbf{F} = [B]^n \Delta^n \mathbf{U} + O[(\Delta t)^2] \tag{9.47}$$

where $[B]$ is the Jacobian matrix $\partial \mathbf{F}/\partial \mathbf{U}$ given by

$$[B] = -\left[\begin{array}{c|c|c|c} 0 & 0 & -1 & 0 \\ uv & -v & -u & 0 \\ \dfrac{3-\gamma}{2}v^2 + \dfrac{1-\gamma}{2}u^2 & (\gamma-1)u & (\gamma-3)v & 1-\gamma \\ \dfrac{\gamma E_t v}{\rho} + (1-\gamma)v(u^2+v^2) & (\gamma-1)uv & -\dfrac{\gamma E_t}{\rho} + \dfrac{\gamma-1}{2}(3v^2+u^2) & -\gamma v \end{array}\right] \tag{9.48}$$

The viscous delta term $\Delta^n \mathbf{V}_1(\mathbf{U}, \mathbf{U}_x)$ is linearized by writing

$$\begin{aligned}\Delta^n \mathbf{V}_1 &= \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}}\right)^n \Delta^n \mathbf{U} + \left(\frac{\partial \mathbf{V}_1}{\partial \mathbf{U}_x}\right)^n \Delta^n \mathbf{U}_x + O[(\Delta t)^2] \\ &= [P]^n \, \Delta^n \mathbf{U} + [R]^n \, \Delta^n \mathbf{U}_x + O[(\Delta t)^2] \\ &= ([P] - [R_x])^n \, \Delta^n \mathbf{U} + \frac{\partial}{\partial x}([R]^n \, \Delta^n \mathbf{U}) + O[(\Delta t)^2] \end{aligned} \tag{9.49}$$

where $[P]$ is the Jacobian $\partial \mathbf{V}_1 / \partial \mathbf{U}$, $[R]$ is the Jacobian $\partial \mathbf{V}_1 / \partial \mathbf{U}_x$, and $[R_x] = \partial [R] / \partial x$. These matrices can be written as

$$[P] - [R_x] = -\frac{1}{\rho} \left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -u\left(\frac{4}{3}\mu\right)_x & \left(\frac{4}{3}\mu\right)_x & 0 & 0 \\ -v\mu_x & 0 & \mu_x & 0 \\ -u^2\left(\frac{4}{3}\mu\right)_x - v^2\mu_x & u\left(\frac{4}{3}\mu\right)_x & v\mu_x & 0 \end{array}\right] \tag{9.50}$$

$$[R] = \frac{1}{\rho} \left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -\frac{4}{3}\mu u & \frac{4}{3}\mu & 0 & 0 \\ -\mu v & 0 & \mu & 0 \\ -\left(\frac{4}{3}\mu - \frac{k}{c_v}\right)u^2 - \left(\mu - \frac{k}{c_v}\right)v^2 - \frac{k}{c_v}\frac{E_t}{\rho} & \left(\frac{4}{3}\mu - \frac{k}{c_v}\right)u & \left(\mu - \frac{k}{c_v}\right)v & \frac{k}{c_v} \end{array}\right] \tag{9.51}$$

The matrix for $[P] - [R_x]$ is obtained by assuming that μ and k are locally independent of $\mathbf{U}$. In a like manner, $\Delta^n \mathbf{W}_2(\mathbf{U}, \mathbf{U}_y)$ is linearized as

$$\Delta^n \mathbf{W}_2 = ([Q] - [S_y])^n \, \Delta^n \mathbf{U} + \frac{\partial}{\partial y}([S]^n \, \Delta^n \mathbf{U}) + O[(\Delta t)^2] \tag{9.52}$$

where

$$[Q] - [S_y] = -\frac{1}{\rho} \left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -u\mu_y & \mu_y & 0 & 0 \\ -v\left(\frac{4}{3}\mu\right)_y & 0 & \left(\frac{4}{3}\mu\right)_y & 0 \\ -v^2\left(\frac{4}{3}\mu\right)_y - u^2\mu_y & u\mu_y & v\left(\frac{4}{3}\mu\right)_y & 0 \end{array}\right] \tag{9.53}$$

and

$$[S] = \frac{1}{\rho}\left[\begin{array}{c|c|c|c} 0 & 0 & 0 & 0 \\ -\mu u & \mu & 0 & 0 \\ -\frac{4}{3}\mu v & 0 & \frac{4}{3}\mu & 0 \\ -\left(\frac{4}{3}\mu - \frac{k}{c_v}\right)v^2 - \left(\mu - \frac{k}{c_v}\right)u^2 - \frac{k}{c_v}\frac{E_t}{\rho} & \left(\mu - \frac{k}{c_v}\right)u & \left(\frac{4}{3}\mu - \frac{k}{c_v}\right)v & \frac{k}{c_v} \end{array}\right] \tag{9.54}$$

The cross-derivative terms can be evaluated without loss of accuracy by noting that

$$\begin{aligned} \Delta^n \mathbf{V}_2 &= \Delta^{n-1}\mathbf{V}_2 + O[(\Delta t)^2] \\ \Delta^n \mathbf{W}_1^n &= \Delta^{n-1}\mathbf{W}_1 + O[(\Delta t)^2] \end{aligned} \tag{9.55}$$

for a uniform time step Δt. By evaluating the cross-derivative terms in this manner, the block tridiagonal form of the final equations is maintained. The Steger method (Steger, 1977) for linearizing viscous terms, described in Section 8.3.3, can be used in place of the linearizations given by Eqs. (9.49) and (9.52). The Steger form of linearization is particularly useful when coordinate transformations have been applied to the N-S equations.

Substituting Eqs. (9.45), (9.47), (9.49), (9.52), and (9.55) into Eq. (9.43) yields

$$\begin{aligned} &\left\{[I] + \frac{\theta_1 \Delta t}{1+\theta_2}\left[\frac{\partial}{\partial x}([A] - [P] + [R_x])^n - \frac{\partial^2}{\partial x^2}[R]^n \right.\right. \\ &\qquad \left.\left. + \frac{\partial}{\partial y}([B] - [Q] + [S_y])^n - \frac{\partial^2}{\partial y^2}[S]^n\right]\right\}\Delta^n \mathbf{U} \\ &= \frac{\Delta t}{1+\theta_2}\left[\frac{\partial}{\partial x}(-\mathbf{E} + \mathbf{V}_1 + \mathbf{V}_2)^n + \frac{\partial}{\partial y}(-\mathbf{F} + \mathbf{W}_1 + \mathbf{W}_2)^n\right] \\ &\quad + \frac{\theta_1 \Delta t}{1+\theta_2}\left[\frac{\partial}{\partial x}(\Delta^{n-1}\mathbf{V}_2) + \frac{\partial}{\partial y}(\Delta^{n-1}\mathbf{W}_1)\right] + \frac{\theta_2}{1+\theta_2}\Delta^{n-1}\mathbf{U} \\ &\quad + O\left[\left(\theta_1 - \frac{1}{2} - \theta_2\right)(\Delta t)^2, (\Delta t)^3\right] \end{aligned} \tag{9.56}$$

where $[I]$ is the unity matrix. In Eq. (9.56), expressions such as

$$\left[\frac{\partial}{\partial x}([A] - [P] + [R_x])^n\right]\Delta^n \mathbf{U}$$

should be interpreted as

$$\frac{\partial}{\partial x}\left[([A] - [P] + [R_x])^n \Delta^n \mathbf{U}\right]$$

The left-hand side (LHS) of Eq. (9.56) is approximately factored in the following manner:

$$\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\frac{\partial}{\partial x}([A] - [P] + [R_x])^n - \frac{\partial^2}{\partial x^2}[R]^n\right]\right\}$$
$$\times\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\frac{\partial}{\partial y}([B] - [Q] + [S_y])^n - \frac{\partial^2}{\partial y^2}[S]^n\right]\right\}\Delta^n \mathbf{U}$$
$$= \text{LHS of Eq. (9.56)} + O[(\Delta t)^3] \tag{9.57}$$

and the final form of the Beam-Warming algorithm becomes

$$\text{LHS of Eq. (9.57)} = \text{RHS of Eq. (9.56)} \tag{9.58}$$

It should be noted that the approximate factorization introduced by Eq. (9.57) may limit the size of the allowable time step because of the added T.E. The partial derivatives in Eq. (9.57) are evaluated using second-order accurate central differences.

The Beam-Warming algorithm is implemented in the following manner:

Step 1:

$$\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\frac{\partial}{\partial x}([A] - [P] + [R_x])^n - \frac{\partial^2}{\partial x^2}[R]^n\right]\right\}\Delta^n \mathbf{U}_1$$
$$= \text{RHS of Eq. (9.56)} \tag{9.59}$$

Step 2:

$$\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\frac{\partial}{\partial y}([B] - [Q] + [S_y])^n - \frac{\partial^2}{\partial y^2}[S]^n\right]\right\}\Delta^n \mathbf{U} = \Delta^n \mathbf{U}_1 \tag{9.60}$$

Step 3:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta^n \mathbf{U} \tag{9.61}$$

In Step 1, $\Delta^n \mathbf{U}_1$ represents the remaining terms on the LHS of Eq. (9.57). Equations (9.59) and (9.60) represent systems of equations that have the same block tridiagonal structure as shown in Eq. (8.96) except that for the 2-D compressible N-S equations, the blocks are 4×4 matrices.

Warming and Beam (1977) have studied the stability of their algorithm by applying it to both the 2-D wave equation,

$$u_t + c_1 u_x + c_2 u_y = 0 \tag{9.62}$$

and the diffusive equation,

$$u_t = a u_{xx} + b u_{xy} + c u_{yy} \tag{9.63}$$

The latter equation is parabolic if $b^2 < 4ac$ and $(a, c) > 0$. They found that the algorithm is unconditionally stable when applied to Eq. (9.62), provided that $\theta_2 > 0$. When applied to Eq. (9.63), the algorithm is unconditionally stable,

provided that $\theta_2 \geqslant 0.385$. Note that neither the leap frog scheme ($\theta_1 = 0, \theta_2 = -\frac{1}{2}$) nor the trapezoidal scheme ($\theta_1 = \frac{1}{2}, \theta_2 = 0$) is unconditionally stable when applied to Eq. (9.63). However, the three-point backward scheme ($\theta_1 = 1, \theta_2 = \frac{1}{2}$) is unconditionally stable and can be used when second-order temporal accuracy is desired.

In order to suppress oscillations that will occur near flow field discontinuities as a result of the central differences that are used for the spatial derivatives, it is necessary to add damping (artificial viscosity) to the Beam-Warming scheme. This can be accomplished by adding a fourth-order explicit dissipation term of the form given by Eq. (8.98) to the RHS of Eq. (9.56). In addition, if only the steady-state solution is of interest, a second-order implicit smoothing term can also be added to the LHS of Eq. (9.56). This latter smoothing term can be second order, since it has no effect on the steady-state solution where $\Delta^n \mathbf{U} = 0$. After the smoothing terms are added, the final differenced form of the algorithm becomes as follows:

Step 1:

$$\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\bar{\delta}_x([A] - [P] + [R_x])^n - \delta_x^2[R]^n - \epsilon_i \delta_x^2\right]\right\} \Delta^n \mathbf{U}_1$$
$$= \text{RHS of Eq. (9.56)} - \epsilon_e\left(\delta_x^4 + \delta_y^4\right)\mathbf{U}^n \tag{9.64}$$

Step 2:

$$\left\{[I] + \frac{\theta_1 \Delta t}{1 + \theta_2}\left[\bar{\delta}_y([B] - [Q] + [S_y])^n - \delta_y^2[S]^n - \epsilon_i \delta_y^2\right]\right\} \Delta^n \mathbf{U} = \Delta^n \mathbf{U}_1 \tag{9.65}$$

Step 3:

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta^n \mathbf{U} \tag{9.66}$$

where $\bar{\delta}$, δ^2, and δ^4 are the usual central-difference operators and ϵ_e and ϵ_i are the coefficients of the explicit and implicit smoothing terms, respectively. Using a Fourier stability analysis, it can be shown that the coefficient of the explicit smoothing term must be in the range

$$0 \leqslant \epsilon_e \leqslant \frac{1 + 2\theta_2}{8(1 + \theta_2)} \tag{9.67}$$

to ensure stability.

Désidéri et al. (1978) have investigated the possibility of maximizing the rate of convergence of the time-dependent solution by using the proper ratio of the coefficients of the smoothing terms. They found that when the Beam-Warming

scheme (with Euler implicit differencing) is applied to the Euler equations, the rate of convergence is optimized when

$$\frac{\epsilon_i}{\epsilon_e} = 2 \tag{9.68}$$

Beam and Warming have pointed out that their algorithm can be simplified considerably if μ is assumed locally constant. In this case, (μ_x, μ_y) = 0 and Eqs. (9.50) and (9.53) reduce to

$$\begin{aligned} [P] - [R_x] &= 0 \\ [Q] - [S_y] &= 0 \end{aligned} \tag{9.69}$$

If only the steady-state solution is desired, Tannehill et al. (1978) have suggested that all the viscous terms on the LHS of the algorithm (i.e., $[P], [R_x], [R], [Q]$, $[S_y], [S]$) can be set equal to zero, provided that implicit smoothing ($\epsilon_i > 0$) is retained. This takes advantage of the fact that the LHS of Eq. (9.57) approaches zero as the steady-state solution is approached. With this simplification, the complexity of the Beam-Warming algorithm is greatly reduced, particularly if a non-Cartesian coordinate system is employed. It is believed that this simplifying technique can be used in all moderate to high Reynolds number computations, since tests confirm that the convergence rate is not affected for these cases. To reduce computation time further, Chaussee and Pulliam (1981) have transformed the coupled set of thin-layer N-S equations into an uncoupled diagonal form.

The Beam-Warming scheme is employed in the widely used ARC3D code (Pulliam and Steger, 1980) developed at NASA Ames Research Center. This code has recently been incorporated into the OVERFLOW code (Buning et al., 1994), which is an outgrowth of both the ARC3D code and the flux-vector splitting F3D code (Steger et al., 1986). In addition, the Beam-Warming scheme is used in the Transonic Navier-Stokes (TNS) code developed by Holst et al. (1987).

9.2.4 Other Implicit Methods

MacCormack (1981) developed an implicit analog of his explicit finite-difference method. This method consists of two stages. The first stage uses the original MacCormack scheme, while the second stage employs an implicit scheme to eliminate any stability restrictions. The resulting matrix equations are either upper or lower block bidiagonal equations, which can be solved in an easier fashion than the usual block tridiagonal systems. A major disadvantage of this scheme is due to the difficulties encountered in applying non-Dirichlet boundary conditions.

Obayashi and Kuwahara (1986) modified the Beam-Warming scheme by applying lower-upper (LU) factorization (see Section 4.3.4) in conjunction with flux-vector splitting (see Section 6.4.1). As a result, each ADI operator is decomposed into the product of lower and upper bidiagonal matrices, which are

easier to solve. This technique is referred to as LU-ADI factorization and is similar to that used in the "implicit" MacCormack scheme described above.

The lower-upper symmetric Gauss-Seidel (LU-SGS) implicit scheme was developed by Yoon and Jameson (1987, 1988) to obtain steady-state solutions of the unsteady Euler and N-S equations. The LU-SGS method employs an approximate Newton iteration procedure that permits scaler diagonal inversions as opposed to the block matrix inversions required in the conventional line Gauss-Seidel (LGS) methods. The LU-SGS method ensures that the matrix is diagonally dominant without resorting to flux splitting. Rieger and Jameson (1988) extended the LU-SGS scheme to three dimensions. This scheme is widely used to solve the compressible N-S equations. It can be combined with an upwind scheme (see next section) to eliminate oscillations near flow field discontinuities.

9.2.5 Upwind Methods

The central-difference schemes described previously for solving the compressible N-S equations almost always require additional dissipation for numerical stability. Upwind schemes, on the other hand, inherently possess the needed dissipation to control these numerical instabilities. Upwind schemes were initially applied to the Euler equations in the early 1980s, as described in Chapter 6. Shortly thereafter, in the mid-1980s, they were applied to the compressible N-S equations. The extension to the compressible N-S equations is straightforward, since the additional shear-stress and heat flux terms are centrally differenced. However, an important consideration that must be taken into account when solving viscous flows with upwind schemes is whether they will produce excessive dissipation that will swamp the natural dissipation in boundary-layer regions. This problem is effectively eliminated by using higher-order upwind schemes.

Some of the earliest applications of upwind schemes to the compressible N-S equations were by Lombard et al. (1983), Coakley (1983b), MacCormack (1985), Chakravarthy et al. (1985), and Thomas and Walters (1985). Lombard et al. (1983) employed an implicit, upwind flux-difference splitting algorithm called the conservative supracharacteristics method (CSCM). Coakley (1983b) and MacCormack (1985) used implicit upwind finite-volume schemes similar to the flux-vector splitting method of Steger and Warming (1981). MacCormack applied LGS and Newton iteration procedures to solve the resulting matrix equations.

Chakravarthy et al. (1985) employed a family of high-order accurate total variation diminishing (TVD) schemes (Chakravarthy and Osher, 1985) based on Roe's approximate Riemann solver to model the convection terms in the compressible N-S equations. (See Section 4.4.12 for a discussion of TVD schemes.) This work has led to a series of widely used, unified computer programs (the USA-series of codes) that were developed at Rockwell International Science Center. These codes can be used for a wide variety of flow situations, including steady and unsteady flows; low-speed, subsonic, transonic, supersonic, and hypersonic flows; internal and external flows; and perfect-gas

and real-gas (equilibrium and nonequilibrium chemistry) flows (Palaniswamy et al., 1989).

Thomas and Walters (1985) initially used the flux-vector splitting method developed by van Leer and co-workers (van Leer et al., 1982; Anderson et al., 1985) to solve the thin-layer compressible N-S equations. This implicit upwind finite-volume method includes third-order accurate spatial differencing along with either approximate factorization or LGS relaxation to solve the resulting matrix equations. Since it has been shown (van Leer et al., 1987) that flux-vector splitting schemes will produce excessive dissipation in boundary layers, as compared with Roe and Osher's approximate Riemann solvers, Thomas and Walters have incorporated Roe's flux-difference splitting into their algorithm. This work has led to the widely used Computational Fluids Laboratory 3-D (CFL3D) code developed at NASA Langley Research Center (Vasta et al., 1987).

Since the initial applications of upwind schemes to the compressible N-S equations, numerous investigators have refined these procedures and have applied them to ever more complicated problems. For example, flow problems involving finite-rate chemistry and thermal nonequilibrium have been successfully computed using the compressible N-S equations. Included in this latter category is the work of Gnoffo (1986, 1989), who developed the Langley aerothermodynamic upwind relaxation algorithm (LAURA) code. This code was developed primarily to solve 3-D external hypersonic flows in chemical and thermal nonequilibrium. It uses an implicit upwind finite-volume algorithm based on Roe's scheme with second-order TVD corrections (Yee, 1985a, 1985b). The fluids, chemistry, and thermodynamics are fully coupled in the code.

Candler and MacCormack (1988) extended the upwind algorithm of MacCormack (1985) to account for chemical and thermal nonequilibrium processes. Molvik and Merkle (1989) developed the TUFF code to solve 3-D external reacting hypersonic flows. The TUFF code uses an implicit upwind finite-volume algorithm and employs a temporal Riemann solver that fully accounts for the multicomponent mixture of gases. Higher-order accuracy is obtained using Chakravarthy and Osher's (1985) TVD scheme. The code has been enhanced to permit the calculation of the internal reacting flow in scramjet engines (Molvik et al., 1993).

9.2.6 Compressible Navier-Stokes Equations at Low Speeds

Until recently, most algorithms designed for compressible flows were observed to become very inefficient or inaccurate at low Mach numbers. The traditional remedy was to solve the incompressible form of the equations for problems requiring solutions in the low-speed regime. This appears unreasonable. The incompressible equations are merely a subset of the compressible equations, and it is well known that the physics itself is usually no more complex merely because the Mach number is low. For most flows, no important changes would be observed if the Mach number were reduced from, say, 0.2 to 0.01 if all other

dimensionless parameters of the flow remained the same. If no significant changes in flow structure are noted as the Mach number drops from about 0.2 toward zero, any difficulties encountered must be due to an inappropriate construction of the numerical algorithm itself. The computational difficulties are believed to arise from two separate mechanisms: (1) an ill-conditioned algebraic problem and (2) round-off errors due to a disparity between magnitudes of variables. The first mechanism is the most troublesome. These difficulties have been addressed by several investigators, including Turkel (1987, 1992), Feng and Merkle (1990), Choi and Merkle (1991), and Peyret and Viviand (1985).

The key issues can be outlined sufficiently by using the 1-D N-S equations as they apply to an ideal gas. Nondimensional variables are defined as

$$t = \frac{\tilde{t}}{L_{\text{ref}}/u_{\text{ref}}} \qquad x = \frac{\tilde{x}}{L_{\text{ref}}} \qquad u = \frac{\tilde{u}}{u_{\text{ref}}}$$

$$p = \frac{\tilde{p}}{(\rho_{\text{ref}} u_{\text{ref}}^2)} \qquad T = \frac{\tilde{T}}{T_{\text{ref}}} \qquad \mu = \frac{\tilde{\mu}}{\mu_{\text{ref}}} \tag{9.70}$$

$$R = \frac{\tilde{R}}{(u_{\text{ref}}^2/T_{\text{ref}})} = \frac{1}{\gamma M^2} \qquad c_p = \frac{\tilde{c}_p}{(u_{\text{ref}}^2/T_{\text{ref}})} = \frac{1}{(\gamma - 1)M^2}$$

where the tildes denote the dimensional variables, subscript ref denotes dimensional reference quantities, and the Mach number M is based on reference quantities and the gas constant:

$$M = \frac{u_{\text{ref}}}{\sqrt{\gamma \tilde{R} T_{\text{ref}}}}$$

We will solve for the primitive variables p, u, and T rather than the "conserved" variables ρ, ρu, and E_t for two reasons. First, we know that the primitive variables are appropriate for low-speed flows because they are widely used in incompressible formulations. If we elect to solve the compressible equations for the same primitive variables, it should be possible to detect what, if anything, is causing the numerical difficulty as M approaches zero and the equations reduce to a variable-property version of the incompressible flow equations. Second, it is possible to identify and remedy the source of numerical difficulties more quickly if we choose to solve for the primitive variables rather than the conserved variables.

Substituting for density by using the ideal gas equation of state and utilizing primitive variables p, u, and T, the conservation equations for mass, momentum, and energy can be written

$$\frac{\partial \mathbf{Q}(\mathbf{q})}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{q})}{\partial x} - \frac{\partial \mathbf{E}_v(\mathbf{q})}{\partial x} = 0 \tag{9.71}$$

where

$$\mathbf{q} = \begin{bmatrix} p \\ u \\ T \end{bmatrix} \qquad \mathbf{Q} = \begin{bmatrix} \dfrac{p}{RT} \\ \dfrac{pu}{RT} \\ \dfrac{p}{\gamma R} + M^2 \dfrac{(\gamma - 1)}{2} \dfrac{pu^2}{RT} \end{bmatrix}$$

$$\mathbf{E} = \begin{bmatrix} \dfrac{pu}{RT} \\ \dfrac{pu^2}{RT} + p \\ \dfrac{pu}{R} + M^2 \dfrac{(\gamma - 1)}{2} \dfrac{pu^3}{RT} \end{bmatrix} \qquad \mathbf{E}_v = \begin{bmatrix} 0 \\ \dfrac{4\mu}{3\,\mathrm{Re}} \dfrac{\partial u}{\partial x} \\ \dfrac{4(\gamma - 1)M^2 \mu u}{3\,\mathrm{Re}} \dfrac{\partial u}{\partial x} + \dfrac{\mu}{\mathrm{Re\,Pr}} \dfrac{\partial T}{\partial x} \end{bmatrix} \tag{9.72}$$

The Reynolds and Prandtl numbers are defined as

$$\mathrm{Re} = \frac{\rho_{\mathrm{ref}} u_{\mathrm{ref}} L_{\mathrm{ref}}}{\mu_{\mathrm{ref}}} \qquad \mathrm{Pr} = \frac{\tilde{c}_p \tilde{\mu}}{\tilde{k}}$$

The Prandtl number of the fluid is assumed to be constant. The viscosity and thermal conductivity can be evaluated from Sutherland's equation. Note that the equations are in the strong conservation-law form even though primitive variables are used. Some of the computational fluid dynamics (CFD) literature gives the impression that "conserved" variables must be used to achieve the favorable properties of the conservation-law form of the equations. This is simply not true. It will be indicated below how the conservation-law form of the discretized equations can be satisfied using primitive variables.

First, we will consider the consequences of the reference Mach number approaching zero. Note that $p/RT = \rho/\rho_{\mathrm{ref}}$ and $R = 1/(\gamma M^2)$. We observe that the combination p/RT approaches a perfectly acceptable finite limit as M goes to zero. Note that $\mathbf{E}$ and $\mathbf{E}_v$ will reduce to a form appropriate for a fluid whose density may be a function of temperature but for which the density cannot be altered by changes in pressure alone, i.e., $\partial\rho/\partial p)_T = 0$. This is the sense in which the compressible N-S equations reduce to an incompressible form by virtue of the zero Mach number limit. Solutions to this system under isothermal conditions can be made to behave as close to the incompressible limit $[(1/\rho)(D\rho/Dt) = 0]$ as desired by choosing the reference Mach number to be sufficiently low.

To consider the mathematical properties of the 1-D conservation equations, it is convenient to write Eq. (9.71) as

$$[A_t]\frac{\partial \mathbf{q}}{\partial t} + [A_x]\frac{\partial \mathbf{q}}{\partial x} = \frac{\partial \mathbf{E}_v}{\partial x} \tag{9.73}$$

where $[A_t]$ and $[A_x]$ are Jacobian matrices to be evaluated at the most recent iteration level and $\mathbf{q}$ is the vector of unknown primitive variables given in Eq. (9.72). The Jacobian matrices can be written as

$$[A_t] = \begin{bmatrix} \dfrac{\gamma M^2}{T} & 0 & -\dfrac{p}{RT^2} \\ \dfrac{\gamma M^2 u}{T} & \dfrac{p}{RT} & -\dfrac{pu}{RT^2} \\ M^2 + \gamma M^4 u^2 \dfrac{(\gamma-1)}{2T} & \gamma M^4 p \dfrac{(\gamma-1)u}{T} & -\gamma M^4 p u^2 \dfrac{(\gamma-1)}{2T^2} \end{bmatrix} \tag{9.74}$$

$$[A_x] = \begin{bmatrix} \dfrac{\gamma M^2 u}{T} & \dfrac{p}{RT} & -\dfrac{pu}{RT^2} \\ \dfrac{\gamma M^2 u^2}{T} + 1 & \dfrac{2pu}{RT} & -\dfrac{pu^2}{RT^2} \\ \gamma M^2 u + \gamma M^4 \dfrac{(\gamma-1)u^3}{2T} & \dfrac{p}{R} + 3\gamma M^4 p u^2 \dfrac{(\gamma-1)}{2T} & -\gamma M^4 p u^3 \dfrac{(\gamma-1)}{2T^2} \end{bmatrix} \tag{9.75}$$

In the above, R has been replaced by $1/(\gamma M^2)$ in those terms that will vanish as M goes to zero. Notice that the variable property incompressible form (in the sense of $\partial\rho/\partial p)_T = 0$) of the equations is recovered as M goes to zero. Notice also that as M goes to zero, the terms containing the time derivative of pressure tend toward zero unless a vanishingly small time step is used. We shall show next that singular behavior of the coupled time-dependent system accompanies the vanishing of the pressure-time derivatives.

The mathematical nature of the time-marching problem can be established by writing the system as

$$\frac{\partial \mathbf{q}}{\partial t} + [A_t]^{-1}[A_x]\frac{\partial \mathbf{q}}{\partial x} = [A_t]^{-1}\frac{\partial \mathbf{E}_v}{\partial x} \tag{9.76}$$

and considering the eigenvalues of $[A_t]^{-1}[A_x]$, which in this case are u, $u + a$, and $u - a$ in terms of nondimensional variables. The parameter a is the nondimensional local speed of sound. The problem can be solved by a marching method, since the eigenvalues are real. As M becomes small, $[A_t]$ becomes ill conditioned. That is, the determinant of $[A_t]$ becomes small, and errors are expected to arise in computing $[A_t]^{-1}$. In the limit as M goes to zero, the

inverse of $[A_t]$ does not exist (the first column and third row vanish), and the system is singular.

The magnitudes of the eigenvalues of the matrix product $[A_t]^{-1}[A_x]$ also provide information about the properties of the system. As M is decreased in the subsonic regime, the eigenvalues of the matrix product $[A_t]^{-1}[A_x]$ begin to differ more and more in magnitude, the ratio of the smallest eigenvalue to the largest being approximately the ratio of the convective speed to the acoustic speed. The condition and degree of "stiffness" of the system can be related to the relative magnitudes of the eigenvalues. When the eigenvalues differ greatly in magnitude, convergence to a steady-state solution is usually slow, or for time-dependent solutions, the allowable time step becomes very small. This occurs because greatly varying signal speeds appear in the equations and the traditional solution schemes attempt to honor all of them, creating a "stiff" system. Since M has almost no influence on the physical characteristics of the flow at very low values of M (it effectively "cancels out" of the physics), it must be possible to devise a solution scheme in which M has very little influence on the convergence rate ("cancels out") of the numerical scheme over the range in which it is an unimportant physical parameter.

Another observation is that the pressure is eliminated as an unknown in the time derivative as M approaches zero. This indicates that the solution to the incompressible equations carry no direct pressure history and thus the pressure field is established anew at each time step.

To compute flows at very low M, one could of course, adopt a fully incompressible scheme, as has often been the case in the past. On the other hand, there are advantages to maintaining the variable-property fully coupled arrangement, and we will indicate that it is possible to modify the compressible formulation so that it will work well for vanishingly small Mach numbers with virtually no sacrifice in accuracy or efficiency.

To overcome the awkward mathematical situation that arises with the unsteady form of the coupled compressible equations at low M, it is necessary to make changes in the formulation of the time terms. At least two alternatives exist. The existing time terms can be modified to permit efficient solutions to be obtained to the steady-flow equations, or an efficient scheme can be obtained by adding a pseudo-time (artificial time) term that vanishes at convergence at each physical time step. This latter approach is recommended because it permits time-accurate solutions to the N-S equations to be obtained when they are needed.

Although the pseudo-time term can take many different forms and still be effective, a suitable arrangement can be obtained by simply adding a term to each equation having the same form as the physical time term but with M removed from the coefficients to the unknown pressure that cause the fatal ill conditioning. The equations then become

$$[A_p]\frac{\partial \mathbf{q}}{\partial \tau} + [A_t]\frac{\partial \mathbf{q}}{\partial t} + [A_x]\frac{\partial \mathbf{q}}{\partial x} = \frac{\partial \mathbf{E}_v}{\partial x} \tag{9.77}$$

where τ is a pseudo-time and the preconditioning matrix $[A_p]$ is given by

$$[A_p] = \begin{bmatrix} \frac{1}{T} & 0 & -\frac{p}{RT^2} \\ \frac{u}{T} & \frac{p}{RT} & -\frac{pu}{RT^2} \\ \frac{1}{\gamma} + M^2u^2\frac{(\gamma-1)}{2T} & \gamma M^4 pu\frac{(\gamma-1)}{T} & -\gamma M^4 pu^2\frac{(\gamma-1)}{2T^2} \end{bmatrix} \tag{9.78}$$

Note that $[A_p]$ is formed from $[A_t]$ by simply dividing the first column of $[A_t]$ by γM^2 (equivalent to multiplying by the nondimensional gas constant R). Presumably, dividing by only M^2 would have the same effect. It is also quite likely that the preconditioning matrix can be simplified somewhat by setting some of the off-diagonal entries in the second and third columns equal to zero. The form above has the conceptual advantage of being easily developed from $[A_t]$ (with minimal change) using commonsense logic.

The hyperbolic system is solved by advancing in pseudo-time until no changes occur at each physical time step. At that point, the time-accurate equations are satisfied. Obviously, this involves "subiterations," but that is consistent with the observation that for a completely incompressible flow, the pressure field must be established at each physical time step with no direct dependence on a previous pressure field. The addition of the pseudo-time term changes the eigenvalues of the hyperbolic system, so that they are clustered closer together in magnitude, effectively at speeds closer to the convective speed. The hyperbolic system being solved is in τ and x, and it is now the eigenvalues of the matrix product $[A_p]^{-1}[A_x]$ that characterize the system. This is similar to marching in pseudo-time to a "steady" solution at each physical time step. The eigenvalues of $[A_p]^{-1}[A_x]$ are

$$u, \tfrac{1}{2}\left[u(1+\gamma M^2) \pm \sqrt{u^2(1-\gamma M^2)^2 + 4\gamma T}\right]$$

If we assume a value of 1.4 for the ratio of specific heats and evaluate u and T at the reference values, the above expressions give 2.275 for the ratio of the magnitudes of the largest to the smallest eigenvalues in the limit of zero Mach number. Without preconditioning, this ratio approaches infinity at the zero Mach number limit. For $M = 0.3$ the ratio is 2.89 with preconditioning.

The second and minor source of difficulty with the coupled compressible formulation at low M is related to the differences in the relative magnitudes of the nondimensional dependent variables. The magnitudes of the velocity and temperature remain of order 1, but the nondimensional pressure tends to increase without limit as M decreases. This permits round-off errors to become significant. Again, this is not a problem of physical origin. Using double-precision arithmetic, no difficulties will usually be observed until M decreases to about

10^{-5} or 10^{-6}. To completely eliminate the problem, a "gauge" or relative pressure can be introduced, so that differences in pressure (originating from pressure derivatives) become differences in the gauge pressure. This procedure is very easy to implement. The nondimensional thermodynamic pressure p is replaced by the sum of p_c, a constant, and p_g, the variable part of the pressure. The constant part is selected to be as large as possible for the problem at hand.

Although the discussion above utilized primitive variables, the results can be interpreted in terms of the more traditional conserved variables (Feng and Merkle, 1990). However, there are no particular disadvantages to using the primitive variables for computations, and the required form of preconditioning is much easier to develop in terms of primitive variables. We shall indicate how the preconditioned system can be solved by an implicit scheme while maintaining the conservation-law form of the discretized equations. Putting Eq. (9.77) in conservation-law form except for the pseudo-time term gives

$$[A_p]\frac{\partial \mathbf{q}}{\partial \tau} + \frac{\partial \mathbf{Q}(\mathbf{q})}{\partial t} + \frac{\partial \mathbf{E}(\mathbf{q})}{\partial x} - \frac{\partial \mathbf{E}_v(\mathbf{q})}{\partial x} = 0 \tag{9.79}$$

The vectors $\mathbf{Q}$, $\mathbf{E}$, and $\mathbf{E}_v$ can be linearized by iterating at each pseudo-time step using a Newton method:

$$\mathbf{Q} = \tilde{\mathbf{Q}} + \left[\tilde{A}_t\right]\Delta\mathbf{q} \qquad \mathbf{E} = \tilde{\mathbf{E}} + \left[\tilde{A}_x\right]\Delta\mathbf{q} \qquad \mathbf{E}_v = \tilde{\mathbf{E}}_v + \left[\tilde{A}_v\right]\Delta\mathbf{q}$$

where the Jacobian matrices $[A_t]$ and $[A_x]$ have been defined previously and

$$[A_v] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \dfrac{4\mu}{3\,\mathrm{Re}}\dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{4(\gamma-1)M^2\mu}{3\,\mathrm{Re}}\dfrac{\partial u}{\partial x} & \dfrac{\mu}{\mathrm{Re\,Pr}}\dfrac{\partial}{\partial x} \end{bmatrix} \tag{9.80}$$

In the above, the tilde indicates evaluation at the most recently determined values (from the previous iteration) and the Δ indicates changes from the previous pseudo-time iterations. Equation (9.79) can then be written as

$$[A_p]\frac{\partial \Delta\mathbf{q}}{\partial \tau} + \frac{\partial\left(\left[\tilde{A}_t\right]\Delta\mathbf{q}\right)}{\partial t} + \frac{\partial\left(\left[\tilde{A}_x\right]\Delta\mathbf{q}\right)}{\partial x} - \frac{\partial\left(\left[\tilde{A}_v\right]\Delta\mathbf{q}\right)}{\partial x}$$
$$= -\left(\frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \frac{\partial \tilde{\mathbf{E}}}{\partial x} - \frac{\partial \tilde{\mathbf{E}}_v}{\partial x}\right) \tag{9.81}$$

Further details will depend somewhat on the discretization scheme selected. For example, if a centrally differenced fully implicit discretization is employed, the algebraic system can be solved using the block tridiagonal algorithm given in Appendix B. The algebraic system is solved for $\Delta\mathbf{q}$, the change between pseudo-time iterations. When the changes vanish at each physical step, the LHS goes to zero. The RHS is the discretized original partial differential equation

(PDE) (the residual) *in conservation-law form* evaluated using **q** from the most recent pseudo-time iteration. Thus the conservation-law form of the equation is satisfied at each physical time step when iterative (pseudo-time) convergence is achieved.

The procedure readily extends to the 2-D and 3-D N-S equations. For example, in 2-D the matrix $[A_p]$ becomes

$$[A_p] = \begin{bmatrix} \frac{1}{T} & 0 & 0 & -\frac{p}{RT^2} \\ \frac{u}{T} & \frac{p}{RT} & 0 & -\frac{pu}{RT^2} \\ \frac{v}{T} & 0 & \frac{p}{RT} & -\frac{pv}{RT^2} \\ \frac{1}{\gamma} + \frac{u^2 + v^2}{2c_pT} & \frac{pu}{Rc_pT} & \frac{pv}{Rc_pT} & -\frac{p(u^2 + v^2)}{2c_pRT^2} \end{bmatrix} \tag{9.82}$$

Further details on implementing this procedure for the 2-D N-S equations can be found in the work by Pletcher and Chen (1993).

9.3 INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

From the above discussion on solving the compressible N-S equations at low speeds, it should be evident that the preconditioned compressible form of the N-S equations can be solved efficiently at vanishingly small M. For low-speed, nearly "incompressible" gas flows where property variations may be important or where the presence of heat transfer requires a solution to the energy equation, solving the compressible form of the equations may be the best choice.

When heat transfer or significant property variations are not present, the traditional incompressible form of the N-S equations is usually selected for numerical solution. Panton (1984) provides a useful discussion of the limiting forms of the N-S equations and the range of applicability of the incompressible form of the equations.

The incompressible N-S equations for a constant property flow without body forces or external heat addition are given by (see Chapter 5)

continuity:

$$\nabla \cdot \mathbf{V} = 0 \tag{9.83}$$

momentum:

$$\rho \frac{D\mathbf{V}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{V} \tag{9.84}$$

energy:

$$\rho c_v \frac{DT}{Dt} = k \nabla^2 T + \Phi \tag{9.85}$$

These equations (one vector, two scalar) are a mixed set of elliptic-parabolic equations that contain the unknowns $(\mathbf{V}, p, T)$. Note that the temperature appears directly only in the energy equation, so that we can uncouple this equation from the continuity and momentum equations. This uncoupling is exact if the fluid properties are independent of temperature. For many applications, the temperature changes are either insignificant or unimportant, and it is not necessary to solve the energy equation. However, if we wish to find the temperature distribution, this can be easily accomplished, since the unsteady energy equation is a parabolic PDE, provided that V has already been computed. With this in mind, we will focus our attention on methods for solving the continuity and momentum equations during the remainder of this chapter.

The 2-D incompressible N-S equations written in Cartesian coordinates (without the energy equation) are

continuity:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{9.86}$$

x momentum:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{9.87}$$

y momentum:

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{9.88}$$

where ν is the kinematic viscosity μ/ρ. These equations are written in the *primitive-variable* form, where p, u, v are the primitive variables. Incompressible flows have been solved successfully by using both primitive and derived (such as vorticity and stream function) variables. Both approaches will be discussed here. We start with the vorticity–stream function technique in Section 9.3.1. Methods for solving Eqs. (9.86)–(9.88) for the primitive variables will follow in Section 9.3.2.

9.3.1 Vorticity–Stream Function Approach

The vorticity–stream function approach has been one of the most popular methods for solving the 2-D incompressible N-S equations. In this approach, a change of variables is made that replaces the velocity components with the vorticity ζ and the stream function ψ. The vorticity vector $\boldsymbol{\zeta}$ was defined in Chapter 5 as

$$\boldsymbol{\zeta} = \nabla \times \mathbf{V} \tag{9.89}$$

The magnitude of the vorticity vector is

$$|\boldsymbol{\zeta}| = |\nabla \times \mathbf{V}| \tag{9.90}$$

while the scaler value of vorticity for 2-D flows can be written as

$$\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{9.91}$$

for a 2-D Cartesian coordinate system. Also in this coordinate system, the stream function ψ is defined by the equations

$$\begin{aligned} \frac{\partial \psi}{\partial y} &= u \\ \frac{\partial \psi}{\partial x} &= -v \end{aligned} \tag{9.92}$$

Using these new dependent variables, the two momentum equations [Eqs. (9.87) and (9.88)] can be combined (thereby eliminating pressure) to give

$$\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y} = \nu\left(\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 \zeta}{\partial y^2}\right) \tag{9.93}$$

or

$$\frac{D\zeta}{Dt} = \nu \nabla^2 \zeta \tag{9.94}$$

This parabolic PDE is called the *vorticity transport equation*. The 1-D form of this equation,

$$\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} = \nu\frac{\partial^2 \zeta}{\partial x^2} \tag{9.95}$$

is the 1-D advection-diffusion equation, which is often used as a model equation. In addition, the nonlinear Burgers equation can be used to model the vorticity transport equation. In fact, the numerical techniques described in Section 4.5 to solve the nonlinear Burgers equation can be directly applied to the vorticity transport equation.

An additional equation involving the new dependent variables ζ and ψ can be obtained by substituting Eqs. (9.92) into Eq. (9.91), which gives

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\zeta \tag{9.96}$$

or

$$\nabla^2 \psi = -\zeta \tag{9.97}$$

This elliptic PDE is the *Poisson equation*. Methods for solving equations of this type are discussed in Section 4.3.

As a result of the change of variables, we have been able to separate the mixed elliptic-parabolic 2-D incompressible N-S equations into one parabolic equation (the vorticity transport equation) and one elliptic equation (the Poisson

equation). These equations are normally solved sequentially using a time-marching procedure, which is described by the following steps:

1. Specify initial values for ζ and ψ at time $t = 0$.
2. Solve the vorticity transport equation for ζ at each interior grid point at time $t + \Delta t$.
3. Iterate for new ψ values at all points by solving the Poisson equation using new ζ at interior points.
4. Find the velocity components from $u = \psi_y$ and $v = -\psi_x$.
5. Determine values of ζ on the boundaries using ψ and ζ values at interior points.
6. Return to Step 2 if the solution is not converged.

An alternative sequential procedure has been used successfully by Mallinson and de Vahl Davis (1973) that is based on a pseudo-transient representation of Eq. (9.96):

$$\frac{\partial \psi}{\partial \tau} - \left(\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \zeta \right) = 0$$

The pseudo-time step is an additional parameter in the scheme that can be varied in order to accelerate convergence. Upon convergence, the pseudo-time term vanishes, and Eq. (9.96) is satisfied.

The vorticity–stream function system can also be solved efficiently in a coupled rather than sequential manner. Rubin and Khosla (1981) solved the 2×2 coupled system for ψ and ζ using the modified strongly implicit procedure. The coupled procedure was also used in combination with multigrid acceleration by Ghia et al. (1982).

The pressure does not appear explicitly in the vorticity–stream function formulation. However, in those applications where the pressure is of interest, it can be readily determined from the velocity solution by solving an additional Poisson equation. This equation is derived by differentiating Eq. (9.87) with respect to x:

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}\right) + \left(\frac{\partial u}{\partial x}\right)^2 + u\frac{\partial^2 u}{\partial x^2} + \frac{\partial v}{\partial x}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial x \partial y} = -\frac{1}{\rho}\frac{\partial^2 p}{\partial x^2} + \nu\frac{\partial}{\partial x}(\nabla^2 u) \tag{9.98}$$

differentiating Eq. (9.88) with respect to y,

$$\frac{\partial}{\partial t}\left(\frac{\partial v}{\partial y}\right) + \left(\frac{\partial v}{\partial y}\right)^2 + v\frac{\partial^2 v}{\partial y^2} + \frac{\partial u}{\partial y}\frac{\partial v}{\partial x} + u\frac{\partial^2 v}{\partial x \partial y} = -\frac{1}{\rho}\frac{\partial^2 p}{\partial y^2} + \nu\frac{\partial}{\partial y}(\nabla^2 v) \tag{9.99}$$

and adding the results to obtain

$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)+\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+2\left(\frac{\partial v}{\partial x}\right)\left(\frac{\partial u}{\partial y}\right)+u\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x\,\partial y}\right)$$
$$+v\left(\frac{\partial^2 u}{\partial x\,\partial y}+\frac{\partial^2 v}{\partial y^2}\right)=-\frac{1}{\rho}\nabla^2 p+\nu\left[\frac{\partial}{\partial x}(\nabla^2 u)+\frac{\partial}{\partial y}(\nabla^2 v)\right] \quad (9.100)$$

Using the continuity equation, Eq. (9.100) can be reduced to

$$\nabla^2 p=2\rho\left(\frac{\partial u}{\partial x}\frac{\partial v}{\partial y}-\frac{\partial u}{\partial y}\frac{\partial v}{\partial x}\right) \quad (9.101)$$

In terms of the stream function this equation can be rewritten as

$$\nabla^2 p=S \quad (9.102)$$

where

$$S=2\rho\left[\left(\frac{\partial^2\psi}{\partial x^2}\right)\left(\frac{\partial^2\psi}{\partial y^2}\right)-\left(\frac{\partial^2\psi}{\partial x\,\partial y}\right)^2\right] \quad (9.103)$$

Thus we have obtained a Poisson equation for pressure that is analogous to Eq. (9.97). In fact, all the methods discussed in Section 4.3 for solving Eq. (9.97) will also apply to Eq. (9.102) if S is differenced in an appropriate manner. A suitable second-order difference representation is given by

$$S_{i,j}=2\rho_{i,j}\left[\left(\frac{\psi_{i+1,j}-2\psi_{i,j}+\psi_{i-1,j}}{(\Delta x)^2}\right)\left(\frac{\psi_{i,j+1}-2\psi_{i,j}+\psi_{i,j-1}}{(\Delta y)^2}\right)\right.$$
$$\left.-\left(\frac{\psi_{i+1,j+1}-\psi_{i+1,j-1}-\psi_{i-1,j+1}+\psi_{i-1,j-1}}{4\,\Delta x\,\Delta y}\right)^2\right] \quad (9.104)$$

For a steady flow problem, the Poisson equation for pressure is only solved once, i.e., after the steady-state values of ζ and ψ have been computed. If only the wall pressures are desired, it is not necessary to solve the Poisson equation over the entire flow field. Instead, a simpler equation can be solved for the wall pressures. This equation is obtained by applying the tangential momentum equation to the fluid adjacent to the wall surface. For a wall located at $y=0$ in a Cartesian coordinate system (see Fig. 9.2), the steady tangential momentum equation (x momentum equation) reduces to

$$\left.\frac{\partial p}{\partial x}\right)_{\text{wall}}=\left.\mu\frac{\partial^2 u}{\partial y^2}\right)_{\text{wall}} \quad (9.105)$$

or

$$\left.\frac{\partial p}{\partial x}\right)_{\text{wall}}=\left.-\mu\frac{\partial\zeta}{\partial y}\right)_{\text{wall}} \quad (9.106)$$

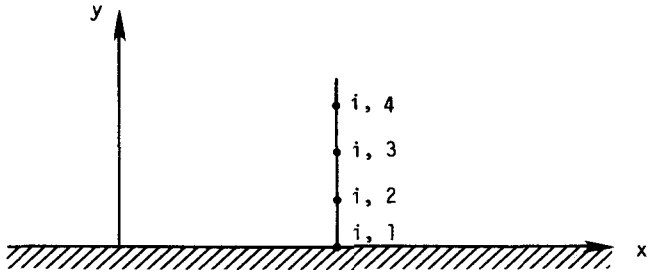

Figure 9.2 Grid points normal to a flat plate at $y = 0$.

which can be differenced as

$$\frac{p_{i+1,1} - p_{i-1,1}}{2\,\Delta x} = -\mu\left(\frac{-3\zeta_{i,1} + 4\zeta_{i,2} - \zeta_{i,3}}{2\,\Delta y}\right) \tag{9.107}$$

In order to apply Eq. (9.107) the pressure must be known for at least one point on the wall surface. The pressure at the adjacent point can be determined using a first-order one-sided difference expression for $\partial p/\partial x$ in Eq. (9.107). Thereafter, Eq. (9.107) can be used to find the pressure at all other wall points. For a body intrinsic coordinate system, Eq. (9.106) becomes

$$\left.\frac{\partial p}{\partial s}\right)_{\text{wall}} = -\mu\left.\frac{\partial \zeta}{\partial n}\right)_{\text{wall}} \tag{9.108}$$

where s is measured along the body surface and n is normal to it.

The time-marching procedure described earlier for solving the vorticity transport equation and the Poisson equation requires that appropriate expressions for ψ and ζ be specified at the boundaries. The specification of these boundary conditions is extremely important, since it directly affects the stability and accuracy of the solution. Let us examine the application of boundary conditions on a wall located at $y = 0$. At the wall surface, ψ is a constant that is usually set equal to zero. In order to find ζ at the wall surface, we expand ψ using a Taylor series about the wall point $(i, 1)$:

$$\psi_{i,2} = \psi_{i,1} + \left.\frac{\partial \psi}{\partial y}\right)_{i,1} \Delta y + \left.\frac{1}{2}\frac{\partial^2 \psi}{\partial y^2}\right)_{i,1} (\Delta y)^2 + \cdots \tag{9.109}$$

Since

$$\begin{aligned} \left.\frac{\partial \psi}{\partial y}\right)_{i,1} &= u_{i,1} = 0 \\ \left.\frac{\partial^2 \psi}{\partial y^2}\right)_{i,1} &= \left.\frac{\partial u}{\partial y}\right)_{i,1} \end{aligned} \tag{9.110}$$

and

$$\zeta_{i,1} = \cancel{\left.\frac{\partial v}{\partial x}\right)^{0}_{i,1}} - \left.\frac{\partial u}{\partial y}\right)_{i,1} = -\left.\frac{\partial^2 \psi}{\partial y^2}\right)_{i,1} \tag{9.111}$$

we can rewrite Eq. (9.109) as

$$\psi_{i,2} = \psi_{i,1} - \tfrac{1}{2}\zeta_{i,1}(\Delta y)^2 + O\left[(\Delta y)^3\right]$$

or

$$\zeta_{i,1} = \frac{2(\psi_{i,1} - \psi_{i,2})}{(\Delta y)^2} + O(\Delta y) \tag{9.112}$$

This first-order expression for $\zeta_{i,1}$ often gives better results than higher-order expressions, which are susceptible to instabilities at higher Reynolds numbers. For example, the following second-order expression, which was first used by Jensen (1959), leads to unstable calculations at moderate to high Reynolds numbers:

$$\zeta_{i,1} = \frac{7\psi_{i,1} - 8\psi_{i,2} + \psi_{i,3}}{2(\Delta y)^2} + O\left[(\Delta y)^2\right] \tag{9.113}$$

Briley (1970) explained the instability by noting that the polynomial expression for ψ, assumed in the derivation of Eq. (9.113), is inconsistent with the evaluation of $u = \partial\psi/\partial y$ at $(i, 2)$ using a central difference. By evaluating u at $(i, 2)$ using the following expression, which is consistent with Eq. (9.113),

$$u_{i,2} = \left.\frac{\partial \psi}{\partial y}\right)_{i,2} = \frac{-5\psi_{i,1} + 4\psi_{i,2} + \psi_{i,3}}{4\,\Delta y} + O\left[(\Delta y)^2\right] \tag{9.114}$$

Briley found his computations to be stable even at high Reynolds numbers.

A classical problem that has wall boundaries surrounding the entire computational region is the driven cavity problem illustrated in Fig. 9.3. In this problem the incompressible viscous flow in the cavity is driven by the uniform translation of the upper surface (lid). The boundary conditions for this problem are indicated in Fig. 9.3. The driven cavity problem is an excellent test case for comparing methods that solve the incompressible N-S equations. A standard test condition of $\mathrm{Re}_l = 100$ is frequently chosen in these comparisons, where

$$\mathrm{Re}_l = \frac{Ul}{\nu} \tag{9.115}$$

and l is the width of the cavity. Two-dimensional computational results are available from numerous investigators, including Burggraf (1966), Bozeman and Dalton (1973), Rubin and Harris (1975), and Ghia et al. (1982). The Ghia et al. results are among the most detailed, having been computed on a 257×257 grid. Unsteady 2-D driven cavity results can be found in the works by Soh and Goodrich (1988) and Pletcher and Chen (1993). Three-dimensional computational results have been reported by Iwatsu et al. (1993) and Freitas et

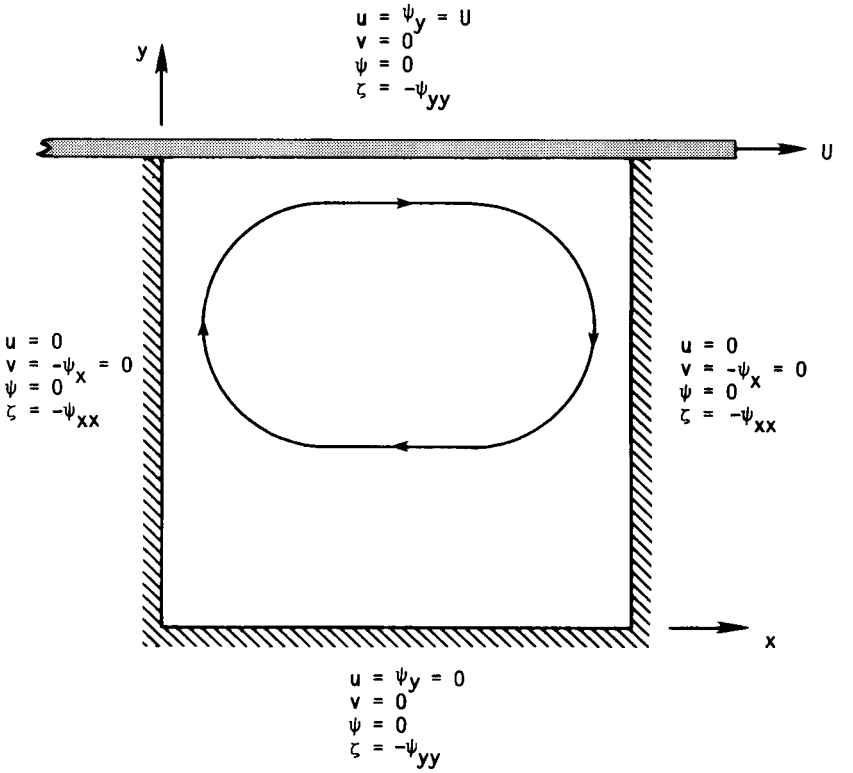

Figure 9.3 Driven cavity problem.

al. (1985). Experimental data are available in the works by Mills (1965), Pan and Acrivos (1967), and Koseff and Street (1984).

The specification of appropriate values for ζ and ψ at other types of boundaries, such as symmetry lines, upper surfaces, inflow and outflow planes, and slip lines, is extremely important, and care must be taken to ensure that the physics of the problem is correctly modeled. An excellent discussion on how to treat these various boundaries can be found in the works by Roache (1972) and Fletcher (1988).

An alternate way of solving the incompressible N-S equations written in the vorticity–stream function formulation, involves using the steady form of the vorticity transport equation

$$u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y} = \nu \nabla^2 \zeta \tag{9.116}$$

This equation is elliptic and can be solved using methods similar to those employed for the Poisson equation. This approach has been successfully used by

several investigators, but it appears to be susceptible to instabilities. For this reason, the transient approach is recommended over this steady-state method.

The extension of the vorticity–stream function approach to 3-D problems is complicated by the fact that a stream function does not exist for a truly 3-D flow. However, the vorticity–stream function approach can be generalized to 3-D by making use of a *vector potential.* Several variations have been used. The earliest method, generally referred to as the *vorticity-potential method* (see, for example, Aziz and Hellums, 1967), expressed the velocity as the curl of a vector potential:

$$\boldsymbol{\psi} = \psi_x \mathbf{i} + \psi_y \mathbf{j} + \psi_z \mathbf{k} \tag{9.117}$$

which satisfies the continuity equation

$$\nabla \cdot \mathbf{V} = 0 \tag{9.118}$$

so that

$$\mathbf{V} = \nabla \times \boldsymbol{\psi} \tag{9.119}$$

and

$$u = \frac{\partial \psi_z}{\partial y} - \frac{\partial \psi_y}{\partial z}$$

$$v = -\frac{\partial \psi_z}{\partial x} + \frac{\partial \psi_x}{\partial z}$$

$$w = \frac{\partial \psi_y}{\partial x} - \frac{\partial \psi_x}{\partial y}$$

After inserting Eq. (9.119) into Eq. (9.89), we obtain

$$\nabla \times (\nabla \times \boldsymbol{\psi}) = \boldsymbol{\zeta} \tag{9.120}$$

Since the vector potential can be arbitrarily chosen to satisfy

$$\nabla \cdot \boldsymbol{\psi} = 0$$

we can simplify Eq. (9.120) to yield

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\zeta} \tag{9.121}$$

This vector Poisson equation represents three scalar Poisson equations that must be solved after each time step. Likewise, the vorticity transport equation for a 3-D problem is a vector equation, which must be separated into three scalar parabolic equations:

$$\begin{aligned}
&\frac{\partial \zeta_x}{\partial t} + u\frac{\partial \zeta_x}{\partial x} + v\frac{\partial \zeta_x}{\partial y} + w\frac{\partial \zeta_x}{\partial z} - \zeta_x\frac{\partial u}{\partial x} - \zeta_y\frac{\partial u}{\partial y} - \zeta_z\frac{\partial u}{\partial z} = \nu \nabla^2 \zeta_x \\
&\frac{\partial \zeta_y}{\partial t} + u\frac{\partial \zeta_y}{\partial x} + v\frac{\partial \zeta_y}{\partial y} + w\frac{\partial \zeta_y}{\partial z} - \zeta_x\frac{\partial v}{\partial x} - \zeta_y\frac{\partial v}{\partial y} - \zeta_z\frac{\partial v}{\partial z} = \nu \nabla^2 \zeta_y \\
&\frac{\partial \zeta_z}{\partial t} + u\frac{\partial \zeta_z}{\partial x} + v\frac{\partial \zeta_z}{\partial y} + w\frac{\partial \zeta_z}{\partial z} - \zeta_x\frac{\partial w}{\partial x} - \zeta_y\frac{\partial w}{\partial y} - \zeta_z\frac{\partial w}{\partial z} = \nu \nabla^2 \zeta_z
\end{aligned} \tag{9.122}$$

to find the three components $(\zeta_x, \zeta_y, \zeta_z)$ of the vorticity vector, although only

two of the above vorticity transport equations are actually independent. Given two components of the vorticity vector, the third can be found from a linear combination of derivatives of the other two. In practice, this approach requires the solution of three parabolic and three elliptic equations at each time level. Although the vorticity-potential method would appear to require more computational effort than the primitive-variable formulation, Aziz and Hellums (1967) reported that the vorticity-potential method was faster and more accurate than a method based on the primitive-variable approach.

More recent studies utilizing derived variables have made use of a variation of the vorticity-potential method known as the *dual-potential method.* This variation was motivated by the discovery by Hirasaki and Hellums (1970) that the boundary conditions for problems with inflow and outflow can be simplified if the velocity is composed of the sum of a scalar potential and a vector potential. This formulation is based on the Helmholtz decomposition theorem, which states that any vector field can be split into a curl-free and a divergence-free part. Thus the dual-potential approach represents the velocity as

$$\mathbf{V} = \nabla\phi + \nabla \times \boldsymbol{\psi} \tag{9.123}$$

where $\nabla\phi$ is the curl-free part and $\nabla \times \boldsymbol{\psi}$ is the divergence-free (or *solenoidal*) part. In addition, it is possible to select $\boldsymbol{\psi}$, such that $\nabla \cdot \boldsymbol{\psi} = 0$ as before. Because $\nabla \times \boldsymbol{\psi}$ is divergence free, the continuity equation requires that

$$\nabla^2\phi = 0 \tag{9.124}$$

The relation between the vector potential and vorticity is obtained by taking the curl of Eq. (9.123),

$$\nabla^2\boldsymbol{\psi} - \nabla(\nabla \cdot \boldsymbol{\psi}) = -\boldsymbol{\zeta} \tag{9.125}$$

Because the vector potential has been selected to be solenoidal, it follows that

$$\nabla^2\boldsymbol{\psi} = -\boldsymbol{\zeta} \tag{9.126}$$

The solution procedure for the dual-potential formulation is nearly the same as given above for the vector-potential formulation, except that in the general case, an additional Laplace equation, Eq. (9.124), needs to be solved for the scalar potential ϕ. The velocity components are then determined from the scalar and vector potentials as

$$\begin{aligned} u &= \frac{\partial\phi}{\partial x} + \frac{\partial\psi_z}{\partial y} - \frac{\partial\psi_y}{\partial z} \\ v &= \frac{\partial\phi}{\partial y} - \frac{\partial\psi_z}{\partial x} + \frac{\partial\psi_x}{\partial z} \\ w &= \frac{\partial\phi}{\partial z} + \frac{\partial\psi_y}{\partial x} - \frac{\partial\psi_x}{\partial y} \end{aligned} \tag{9.127}$$

Notice that the part of the velocity components that is derived from the vector-potential function is the same as observed earlier for the vector-potential formulation. In fact, when the dual-potential formulation is applied to a viscous

problem in which there is no throughflow, i.e., a flow in which the boundary conditions on all boundaries are the no-slip conditions (such as the driven cavity problem), a trivial solution of $\phi = 0$ satisfies the Laplace equation for the scalar potential as well as the required boundary conditions. Thus, for such problems, the scalar potential is not needed. It is only when throughflow exists that the merits of including the scalar potential become evident. Accommodating throughflow boundary conditions with the vector potential alone can be done but is overly complex (Hiraski and Hellums, 1970). Further details on the treatment of boundary conditions for the dual-potential method can be found in the works of Aregbesola and Burley (1977), Richardson and Cornish (1977), Morino (1986), and Gegg et al. (1989).

Before moving on to a discussion of the primitive-variable approach, we will briefly describe an approach that can be considered a hybrid of the stream function–vorticity approach and the primitive-variable approach. In this hybrid vorticity-velocity approach, the dependent variables are the vorticity components $(\zeta_x, \zeta_y, \zeta_z)$ and the velocity components (u, v, w). The vorticity components are obtained by solving Eq. (9.122), and the velocity components are determined from

$$\nabla^2 \mathbf{V} = -\nabla \times \boldsymbol{\zeta} \tag{9.128}$$

This vector equation is derived by taking the vector cross product of the del operator with the definition of the vorticity vector and then simplifying the resulting expression,

$$\nabla \times (\nabla \times \mathbf{V}) = \nabla \times \boldsymbol{\zeta} \tag{9.129}$$

using the appropriate vector identity. Agarwal (1981) states that this hybrid vorticity-velocity approach avoids the necessity of using a staggered-grid arrangement, which is required in some primitive-variable approaches. Other applications of the vorticity-velocity approach can be found in the works of Dennis et al. (1979), Gastski et al. (1982), Fasel and Booz (1984), Farouk and Fusegi (1985), Osswald et al. (1987), and Guj and Stella (1988).

9.3.2 Primitive-Variable Approach

General. The approaches based on derived variables such as the vorticity–stream function and dual-potential methods lose some of their attractiveness when applied to a 3-D flow, as discussed in the last section. Consequently, the incompressible N-S equations are most often solved in their primitive-variable form (u, v, w, p) for 3-D problems. Even for 2-D problems, the use of primitive variables is quite common.

The incompressible N-S equations written in nondimensional primitive-variable form for a Cartesian coordinate system are given by the following.

continuity:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} + \frac{\partial w^*}{\partial z^*} = 0 \tag{9.130}$$

x momentum:

$$\frac{\partial u^*}{\partial t^*} + u^*\frac{\partial u^*}{\partial x^*} + v^*\frac{\partial u^*}{\partial y^*} + w^*\frac{\partial u^*}{\partial z^*} = -\frac{\partial p^*}{\partial x^*} + \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} + \frac{\partial^2 u^*}{\partial z^{*2}}\right) \tag{9.131}$$

y momentum:

$$\frac{\partial v^*}{\partial t^*} + u^*\frac{\partial v^*}{\partial x^*} + v^*\frac{\partial v^*}{\partial y^*} + w^*\frac{\partial v^*}{\partial z^*} = -\frac{\partial p^*}{\partial y^*} + \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} + \frac{\partial^2 v^*}{\partial z^{*2}}\right) \tag{9.132}$$

z momentum:

$$\frac{\partial w^*}{\partial t^*} + u^*\frac{\partial w^*}{\partial x^*} + v^*\frac{\partial w^*}{\partial y^*} + w^*\frac{\partial w^*}{\partial z^*} = -\frac{\partial p^*}{\partial z^*} + \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2 w^*}{\partial x^{*2}} + \frac{\partial^2 w^*}{\partial y^{*2}} + \frac{\partial^2 w^*}{\partial z^{*2}}\right) \tag{9.133}$$

These equations are nondimensionalized using

$$\begin{aligned} u^* &= \frac{u}{V_\infty} & x^* &= \frac{x}{L} & p^* &= \frac{p}{\rho_\infty V_\infty^2} \\ v^* &= \frac{v}{V_\infty} & y^* &= \frac{y}{L} & t^* &= \frac{tV_\infty}{L} \\ w^* &= \frac{w}{V_\infty} & z^* &= \frac{z}{L} & \mathrm{Re}_L &= \frac{V_\infty L}{\nu_\infty} \end{aligned} \tag{9.134}$$

Notice that no time derivative of pressure appears in these equations. For an incompressible fluid, pressure waves propagate at infinite speed. The pressure is determined through the governing equations and boundary conditions, but it also can be shown to be governed by an elliptic PDE.

Methods for solving the incompressible N-S equations in primitive variables can be grouped into two broad categories. The first we shall refer to as the *coupled approach.* In this approach the discretized conservation equations are solved, treating all dependent variables as simultaneous unknowns. For the time-dependent N-S equations this is implemented in what is known as the *artificial compressibility* (also known as the *pseudo-compressibility*) *method.* An artificial-time derivative of pressure is added to the continuity equation to permit the solution to the coupled hyperbolic system to be advanced in time. Without the addition of such a time derivative, the algebraic system of equations resulting from a coupled time-dependent discretization is singular. This occurs in a manner similar to the singular behavior noted in Section 9.2.6 for the compressible time-dependent formulation taken to the incompressible limit. The

addition of the artificial-time term plays a role similar to that of preconditioning applied to the compressible formulation. Upon convergence, the artificial-time term vanishes, as will be explained below. On the other hand, it will be seen that the steady-flow incompressible equations can be solved by a coupled discretization without the addition of artificial terms.

The second strategy often employed for the incompressible N-S equations will be referred to as the *pressure correction* approach. Such methods are also known as pressure-based, uncoupled, sequential, or segregated methods. The distinguishing feature of this approach is the use of a derived equation to determine the pressure. Typically, the momentum equations are solved for the velocity components in an uncoupled manner. The x momentum equation is solved for the x component of velocity, the y momentum equation is solved for the y component of velocity, etc. In doing so, the equations are linearized by using values lagged in iteration level for the other unknowns, including pressure. The velocity components have thus been computed without using the continuity equation as a constraint. Usually, a Poisson equation is developed for the pressure, or changes in the pressure, that will alter the velocity field in a direction such as to satisfy the continuity equation. Such an equation for pressure can be derived from the conservation equations in a rigorous manner. However, several well-known schemes use an approximate formulation for the pressure equation, which is justified as long as the iterative procedure produces a solution that satisfies all of the discretized conservation equations, including the continuity equation.

The literature on numerical schemes for the incompressible N-S equations is quite extensive. A great many numerical schemes, each differing from others in some detail but falling within the two basic approaches defined above, have been proposed for solving the incompressible N-S equations. Only a few specific examples will be given here.

Coupled approach: The method of artificial compressibility. One of the early techniques proposed for solving the incompressible N-S equations in primitive-variable form was the artificial compressibility method of Chorin (1967). In this method, the continuity equation is modified to include an artifical compressibility term that vanishes when the steady-state solution is reached. With the addition of this term to the continuity equation, the resulting N-S equations are a mixed set of hyperbolic-parabolic equations, which can be solved using a standard time-dependent approach. In order to explain this method, let us apply it to Eqs. (9.130)–(9.133). The continuity equation is replaced by

$$\frac{\partial \tilde{\rho}^*}{\partial \tilde{t}^*} + \frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} + \frac{\partial w^*}{\partial z^*} = 0 \tag{9.135}$$

where $\tilde{\rho}^*$ is an artificial density and $\tilde{t}^*$ is a fictitious time that is analogous to real time in a compressible flow. The artificial density is related to the pressure

by the artificial equation of state,

$$p^* = \beta \tilde{\rho}^* \tag{9.136}$$

where β is the artificial compressibility factor to be determined later. Note that at steady-state the solution is independent of $\tilde{\rho}^*$ and $\tilde{t}^*$, since $\partial \tilde{\rho}^* / \partial \tilde{t}^* \to 0$. After replacing t^* with $\tilde{t}^*$ in Eqs. (9.131)–(9.133) and substituting Eq. (9.136) into Eq. (9.135), we can apply a suitable numerical technique to the resulting equations and march the solution in $\tilde{t}^*$ to obtain a final steady-state incompressible solution. Obviously, this technique is applicable only to steady-flow problems, since it is not time accurate.

In order to facilitate the application of the numerical scheme, Eqs. (9.130)–(9.133) and Eqs. (9.135)–(9.136) can be combined into the following vector form:

$$\frac{\partial \mathbf{u}^*}{\partial \tilde{t}^*} + \frac{\partial \mathbf{e}^*}{\partial x^*} + \frac{\partial \mathbf{f}^*}{\partial y^*} + \frac{\partial \mathbf{g}^*}{\partial z^*} = \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2}{\partial x^{*2}} + \frac{\partial^2}{\partial y^{*2}} + \frac{\partial^2}{\partial z^{*2}}\right)[D]\mathbf{u}^* \tag{9.137}$$

where

$$\mathbf{u}^* = \begin{bmatrix} p^* \\ u^* \\ v^* \\ w^* \end{bmatrix} \quad \mathbf{e}^* = \begin{bmatrix} \beta u^* \\ p^* + (u^*)^2 \\ u^* v^* \\ u^* w^* \end{bmatrix}$$

$$\mathbf{f}^* = \begin{bmatrix} \beta v^* \\ u^* v^* \\ p^* + (v^*)^2 \\ v^* w^* \end{bmatrix} \quad \mathbf{g}^* = \begin{bmatrix} \beta w^* \\ u^* w^* \\ v^* w^* \\ p^* + (w^*)^2 \end{bmatrix} \tag{9.138}$$

$$[D] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that since Eq. (9.136) represents an artificial equation of state, then $\beta^{1/2}$ plays the role of an artificial sound speed. Defining Jacobians,

$$[A] = \frac{\partial \mathbf{e}^*}{\partial \mathbf{u}^*} \qquad [B] = \frac{\partial \mathbf{f}^*}{\partial \mathbf{u}^*} \qquad [C] = \frac{\partial \mathbf{g}^*}{\partial \mathbf{u}^*} \tag{9.139}$$

the LHS of Eq. (9.137) can be rewritten as

$$\frac{\partial \mathbf{u}^*}{\partial \tilde{t}^*} + [A]\frac{\partial \mathbf{u}^*}{\partial x^*} + [B]\frac{\partial \mathbf{u}^*}{\partial y^*} + [C]\frac{\partial \mathbf{u}^*}{\partial z^*} \tag{9.140}$$

where

$$[A] = \begin{bmatrix} 0 & \beta & 0 & 0 \\ 1 & 2u^* & 0 & 0 \\ 0 & v^* & u^* & 0 \\ 0 & w^* & 0 & u^* \end{bmatrix} \quad [B] = \begin{bmatrix} 0 & 0 & \beta & 0 \\ 0 & v^* & u^* & 0 \\ 1 & 0 & 2v^* & 0 \\ 0 & 0 & w^* & v^* \end{bmatrix}$$

$$[C] = \begin{bmatrix} 0 & 0 & 0 & \beta \\ 0 & w^* & 0 & u^* \\ 0 & 0 & w^* & v^* \\ 1 & 0 & 0 & 2w^* \end{bmatrix}$$

The eigenvalues of $[A]$, $[B]$, and $[C]$ are

$$\left(u, u, u \pm \sqrt{u^2 + \beta}\right), \quad \left(v, v, v \pm \sqrt{v^2 + \beta}\right), \quad \left(w, w, w \pm \sqrt{w^2 + \beta}\right)$$

respectively. This suggests that the magnitude of β should be close to that of the convective velocities to avoid the stiffness associated with a disparity in the magnitudes of the eigenvalues. Although the artificial equation of state suggests that $\beta^{1/2}$ is an artificial speed of sound, the eigenvalues above indicate that the effective acoustic wave speeds are really the quantities under the radicals in the eigenvalues above ($\sqrt{u^2 + \beta}$, for example), which are functions of the velocity components as well as β. The optimum value of β may be somewhat problem dependent. Kwak et al. (1986) suggest that a value of β in the range 0.1–10.0 will work well for most problems. On the high side, the problem is one of stiffness, which retards the convergence rate. On the low side, the value of $\beta \Delta \tilde{t}^*$ should be large enough to permit pressure waves (which actually should move at infinite speed in the incompressible limit) to propagate far enough to reasonably balance viscous effects during the artificial transient, or the pseudo-time iterations will tend not to converge.

In the original paper of Chorin, the leap frog/DuFort-Frankel finite-difference scheme (see Section 4.5.2) was used. Since that time, a variety of numerical schemes have been used to solve the hyperbolic system of equations, including the multistage Runge-Kutta explicit method, approximate-factorization implicit schemes, the LU-SGS implicit scheme, and the coupled strongly implicit scheme. Generally, it is believed that any numerical solution strategy that is appropriate for solving the time-dependent compressible N-S equations as a coupled system will also work for solving the discretized equations resulting from the artificial compressibility formulation. On balance, it seems that implicit formulations have been favored over explicit methods for incompressible viscous flow applications.

Over the years, a number of investigators have reported good success with the artificial compressibility method in a number of impressive applications using a variety of algorithms. Among these are the works of Steger and Kutler (1976), Choi and Merkle (1985), Kwak et al. (1986), Hartwich and Hsu (1987), and Hartwich et al. (1988). Initially, such methods were considered to be only

suitable for obtaining steady solutions because the solutions had to be iterated to time convergence for the artificial term to vanish. It has now been demonstrated (Merkle and Athavale, 1987; Pan and Chakravarthy, 1989; Rogers et al., 1989; Chen and Pletcher, 1993) that this approach can be made time accurate by considering the time terms in the momentum equations to be the real, physical time terms, and the time term added to the continuity equation to be in pseudo-time. The solution is then iterated to pseudo-time convergence at each real (physical) time step. Pseudo-time terms can also be added to the momentum equations (while leaving physical time terms intact) as an option to aid in maintaining diagonal dominance of the algebraic system. When the pseudo-time term (or terms) vanish, the solution obtained satisfies the complete time-dependent N-S equations. This approach employing pseudo-time terms in all equations will be discussed next in some detail.

To obtain time-accurate solutions to the N-S equations by the artificial compressibility method, Eq. (9.137) is modified by retaining the physical time terms but adding pseudo-time terms to give

$$[A_p]\frac{\partial \mathbf{u}^*}{\partial \tilde{t}^*} + [A_t]\frac{\partial \mathbf{u}^*}{\partial t^*} + [A]\frac{\partial \mathbf{u}^*}{\partial x^*} + [B]\frac{\partial \mathbf{u}^*}{\partial y^*} + [C]\frac{\partial \mathbf{u}^*}{\partial z^*} = \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2}{\partial x^{*2}} + \frac{\partial^2}{\partial y^{*2}} + \frac{\partial^2}{\partial z^{*2}}\right)[D]\mathbf{u}^* \tag{9.141}$$

where $\mathbf{u}^*$, $[A]$, $[B]$, $[C]$, and $[D]$ are as defined previously and

$$[A_p] = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & d \end{bmatrix} \qquad [A_t] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The above formulation works well with the parameters a, b, c, d set equal to 1, although other values may enhance the convergence rate or robustness of some algorithms. At each physical time step, the computations are advanced in pseudo-time until convergence (no further changes in the variables are observed). The equations are often solved in conservation-law form with the linearization achieved by a Newton method, whereby the Jacobians are evaluated using the most recently computed values. Notice that numerical errors associated with the linearization can be driven to zero during the pseudo-time iteration cycle, and at iterative convergence, the conservation-law form of the equations is satisfied. This strategy follows the pattern described in Section 9.2.6 for solving the compressible N-S equations in primitive variables. The eigenvalues of the system are the eigenvalues of $[A_p]^{-1}[A]$, $[A_p]^{-1}[B]$, and $[A_p]^{-1}[C]$. The eigenvalues of $[A_p]^{-1}[A]$ are $u^*/c, u^*/d, (u^*/b) \pm \sqrt{(u^{*2}/b^2) + (\beta/ab)}$. The other

eigenvalues are obtained from the above expression by replacing u^* with v^* to obtain the eigenvalues of $[A_p]^{-1}[B]$ and by replacing u^* with w^* to obtain the eigenvalues of $[A_p]^{-1}[C]$. For $a = b = c = d = 1$ the eigenvalues are seen to be the same as for the original steady-flow pseudo-compressibility formulation of Chorin.

Some flexibility exists in establishing $[A_p]$, and for some choices, the eigenvalues are altered somewhat. For example, we can add a pseudo-time derivative of pressure to each momentum equation. Taking some liberties with the form of the coefficients of that term, we find the following candidate preconditioning matrix:

$$[A_p] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \dfrac{au}{\beta} & 1 & 0 & 0 \\ \dfrac{av}{\beta} & 0 & 1 & 0 \\ \dfrac{aw}{\beta} & 0 & 0 & 1 \end{bmatrix}$$

where a is an arbitrary constant. Using this last form for $[A_p]$, we compute the eigenvalues of $[A_p]^{-1}[A]$ to be $u, u, u - \frac{1}{2}(au \pm \sqrt{a^2u^2 - 4au^2 + 4u^2 + 4\beta})$. Notice that this result reduces to the eigenvalues for the original Chorin scheme if $a = 0$. However, if $a = 2$, the eigenvalues become $u, u, \pm \beta^{1/2}$, and the effective acoustic speed becomes independent of the velocity and equal to $\beta^{1/2}$. We close this discussion with the reminder that adding the pseudo-time terms to the momentum equations is not essential for time accuracy if pseudo-time convergence is achieved at each physical time step (see Pan and Chakravarthy et al. 1989).

Note that the strategy employed in the time-accurate version of the artificial compressibility scheme bears some resemblance to that suggested in Section 9.2.6 for solving the preconditioned compressible N-S equations. In fact, if isothermal conditions are assumed, the application of the preconditioned compressible formulation to low-speed flows becomes effectively an artificial compressibility scheme to the extent that percentage changes in the density remain small.

Coupled approach: Space marching. It is possible to solve the steady form of both the compressible and incompressible N-S equations by coupled space-marching methods. Because the steady-flow system of equations is elliptic for subsonic flows, repeated calculation sweeps are made from inflow to outflow until convergence is achieved. Examples of coupled space-marching methods applied to the steady N-S equations can be found in the works of Bentson and Vradis (1987), TenPas and Pletcher (1987, 1991), Vradis et al. (1992), and

TenPas and Hancock (1992). Following the formulation of TenPas and Hancock (1992), the 2-D incompressible continuity and momentum equations are written in the form

$$\frac{\partial \mathbf{e}^*}{\partial x^*} + \frac{\partial \mathbf{f}^*}{\partial y^*} = \frac{1}{\mathrm{Re}_L}\left(\frac{\partial^2}{\partial x^{*2}} + \frac{\partial^2}{\partial y^{*2}}\right)[D]\mathbf{u}^* \tag{9.142}$$

where

$$\mathbf{u}^* = \begin{bmatrix} p^* \\ u^* \\ v^* \end{bmatrix} \quad \mathbf{e}^* = \begin{bmatrix} u^* \\ u^{*2} + p^* \\ u^* v^* \end{bmatrix} \quad \mathbf{f}^* = \begin{bmatrix} v^* \\ u^* v^* \\ v^{*2} + p \end{bmatrix} \quad [D] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The solution is to be advanced by marching in the "streamwise" direction, which we shall assume to be the positive x direction. For an implicit space-marching strategy to work successfully, attention must be given to two details. The first is the manner in which the streamwise pressure gradient is treated. It is essential in the marching sweep to treat the *downstream* value of pressure as given either from an initial estimate or from the value computed from the most recent sweep. Note that this is the downstream and not the downwind value, in that the downstream value is fixed regardless of the local flow direction. For example, if the marching solution is being advanced from marching level i to $i + 1$, the pressure gradient term in the x momentum equation would be forward differenced (either first order or second order), treating the value of pressure at $i + 1$ as unknown. For a first-order representation, this would give

$$\left.\frac{\partial p}{\partial x}\right|_{i+1} \cong \frac{p_{i+2} - p_{i+1}}{\Delta x}$$

where p_{i+2} would be evaluated from the previous sweep (i.e., "fixed") as the solution at $i + 1$ is computed, but p_{i+1} would be an unknown. The transverse pressure gradient term in the y momentum equation is discretized at the unknown $i + 1$ level using forward (rather than central) second-order differences to prevent even-odd decoupling. The pressure is fixed by the boundary conditions at the downstream boundary. The fixing of the downstream pressure at each marching step is consistent with the physical and mathematical nature of the steady incompressible flow problem, in that the local solution should be influenced by information coming from all directions.

The second concern in the space-marching procedure is to ensure that the difference stencil honors the appropriate zone of dependence in a manner that maintains diagonal dominance in the implicit solution algorithm. To achieve this, the streamwise convective derivative in the momentum equations is invariably upwinded. In regions where the flow is reversed, as behind a step in a rearward-facing step flow, the differencing direction changes. The upwind value is treated as known, and the downwind value is unknown, which is the value at $i + 1$ in the present discussion. The streamwise second-derivative terms are differenced centrally, with the $i + 1$ values treated as unknowns, and the upstream and downstream values lagged to the most recently computed values.

The continuity equation is differenced in a form equivalent to a finite-volume approximation for a control volume shifted upstream between i and $i + 1$. This allows for global mass conservation to be ensured at each marching sweep. The streamwise derivative uses second-order central differences, while the derivative in the transverse direction is approximated by a second-order backward difference. The need to march in the predominant flow direction restricts somewhat the available options for grid construction when generalized coordinates are used. For most flow configurations, the use of space marching is limited to "H" grids (see Thompson et al., 1985).

Multiple marching sweeps are required in space-marching schemes in order to develop the final pressure distribution because the downstream value of pressure in the streamwise momentum equation is always fixed at the value determined at the previous sweep until convergence is achieved. It has been observed that convergence can be accelerated by "correcting" the pressure between streamwise sweeps. Presumably, this accelerates the rate at which information travels upstream. The pressure correction usually takes the form of a single sweep from downstream to upstream of an approximate Poisson equation for the pressure. More details on space-marching schemes for the complete N-S equations can be found in the work of TenPas (1990). Such schemes have been demonstrated for computing developing 2-D and 3-D flows in a channel, flows over a rearward-facing step, and flow over a cylinder at low Reynolds numbers. Space-marching procedures have also been employed for the compressible N-S equations in the subsonic regime (TenPas and Pletcher, 1991; Pappalexis and TenPas, 1993).

Pressure-correction approach: General. The general pressure-correction approach is characterized by a formulation in which the momentum equations are solved sequentially for the velocity components using the best available estimate for the pressure distribution. Such a procedure will not yield a velocity field that satisfies the continuity equation unless the correct pressure distribution is employed. If mass sources exist, the pressure is improved in a separate step in a manner that will eliminate the mass sources (satisfy continuity). If the pressure changes, the solution to the momentum equations will change, and particularly, in implicit schemes, the sequence is repeated iteratively until a divergence-free velocity field is established. Such schemes have often been called "pressure-based" schemes, as contrasted with coupled-solution schemes (such as the artificial compressibility scheme) which have been referred to as "density based." This terminology (pressure or density based) is being abandoned here because even in coupled approaches, it is increasingly common to employ primitive variables (u, v, p, T, for example), where the density no longer appears as a variable. This is especially evident in schemes employing low Mach number preconditioning. Thus the terminology "pressure correction" is being suggested as being more descriptive than "pressure based," since both coupled and sequential schemes are likely to employ pressure as a dependent variable.

Pressure-correction methods have been widely used for solving the incom-

pressible N-S equations. The methods differ primarily in the algorithms used to solve the component equations and the strategy employed to develop an equation to be solved for an improved pressure. Such an equation is most often a Poisson equation. Early forms of these methods employed staggered grids to avoid even-odd decoupling of the pressure. More recently, regular (colocated) grids have been employed satisfactorily.

Some of the most commonly used variations of the pressure-correction method are the marker-and-cell (MAC) method of Harlow and Welch (1965), the SIMPLE and SIMPLER methods of Caretto et al. (1972) and Patankar (1980), the fractional-step method (Chorin, 1968; Yanenko, 1971; Marchuk, 1975), and the primitive-variable implicit split operator (PISO) method of Issa (1986). A sampling of these methods will now be discussed.

Pressure-correction approach: Marker-and-cell method. Perhaps the earliest pressure-correction scheme for solving the incompressible N-S equations was the MAC method introduced by Harlow and Welch (1965) and Welch et al. (1966). The scheme was based on a staggered grid similar to that introduced in Chapter 8. The marker-and-cell terminology arose because the method had the capability to resolve time-dependent free surface flows by tracing the paths of fictitious massless marker particles introduced on the free surface. The solution was advanced in time by solving the momentum equations for velocity components using the best current estimate of the pressure distribution, much as indicated in Section 8.4.3. Such a solution initially would not satisfy the continuity equation unless the correct pressure distribution was used. The pressure is improved by numerically solving a Poisson equation derived in the same manner as Eq. (9.100). In nondimensional form this equation can be written as

$$\nabla^2 p^* = S_p^* - \frac{\partial D^*}{\partial t^*} \tag{9.143}$$

where

$$\begin{aligned} S_p^* = {} & \frac{d}{dx^*}\left[-(u^*u_x^* + v^*u_y^* + w^*u_z^*) + \frac{1}{\mathrm{Re}_L}(u_{xx}^* + u_{yy}^* + u_{zz}^*)\right] \\ & + \frac{d}{dy^*}\left[-(u^*v_x^* + v^*v_y^* + w^*v_z^*) + \frac{1}{\mathrm{Re}_L}(v_{xx}^* + v_{yy}^* + v_{zz}^*)\right] \\ & + \frac{d}{dz^*}\left[-(u^*w_x^* + v^*w_y^* + w^*w_z^*) + \frac{1}{\mathrm{Re}_L}(w_{xx}^* + w_{yy}^* + w_{zz}^*)\right] \end{aligned}$$

and D^* is the local dilatation term given by

$$D^* = u_x^* + v_y^* + w_z^*$$

and terms such as u_x^* denote $\partial u^*/\partial x^*$. The value of S_p^* is determined from the solution of the momentum equations using the provisional values of pressure.

counter and $D_{i,j,k}^{m+1}$ is set equal to zero. That is, the correction in pressure is required to compensate for the nonzero dilation at the m iteration level. The Poisson equation is then solved for the revised pressure field. Depending on the details of the discretization method employed (explicit, implicit, etc.), the improved pressure may then be used in the momentum equation for a better solution at the present time step. If the dilation (divergence of velocity field) is not zero, the cyclic process of solving the momentum equations and the Poisson equation is repeated until the velocity field is divergence free.

A point that needs careful attention in all schemes requiring the solution of a Poisson equation is the proper application of boundary conditions. These boundary conditions are Neumann as derived from the momentum equations. When a staggered grid is used, the pressure at the boundary itself is not required for the solution of the momentum equations. However, in representing the Neumann boundary condition, the pressure at a point below the boundary (a fictitious point) is called for. It is shown in Chapter 8 (Section 8.4.3) that this pressure cancels out of the representation of the boundary condition if it is evaluated implicitly at the current iteration level. To achieve convergence of the Poisson equation, the solution must satisfy the integral constraint,

$$\iint_A \nabla^2 p \, dA = \oint_C \frac{\partial p}{\partial n} \, ds \tag{9.144}$$

where C is the closed boundary of the solution domain of area A and ds is a differential length along C. On a staggered grid (see Chapter 8) this is satisfied automatically. When a nonstaggered mesh is used, the possible inconsistency arising from Eq. (9.144) can be circumvented (Ghia et al., 1977a, 1979, 1981; Briley, 1974) by reducing the source term of the Poisson equation [RHS of Eq. (9.143)] at every point by the same fixed amount computed as required to satisfy the global constraint of Eq. (9.144). With the nonstaggered mesh, the pressure gradient at the wall is required. A technique for its computation will be illustrated for a wall located at $y = 0$ as shown in Fig. 9.4. Note that a fictitious row of grid points for pressure has been added below the wall surface in this

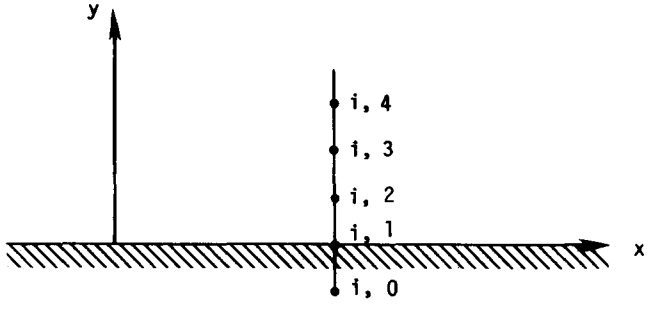

Figure 9.4 Grid points for determination of pressure boundary condition.

nonstaggered grid. At the wall surface, the y momentum equation [Eq. (9.132)] reduces to

$$\left.\frac{\partial p^*}{\partial y^*}\right)_{i,1} = \frac{1}{\mathrm{Re}_L}\left.\frac{\partial^2 v^*}{\partial y^{*2}}\right)_{i,1} \tag{9.145}$$

This equation can be differenced using the familiar second-order accurate central-difference expressions

$$\frac{p^*_{i,2} - p^*_{i,0}}{2\,\Delta y^*} = \frac{1}{\mathrm{Re}_L}\left(\frac{v^*_{i,2} - 2\overset{0}{\cancel{v^*_{i,1}}} + v^*_{i,0}}{(\Delta y^*)^2}\right) \tag{9.146}$$

where $v^*_{i,0}$ is the value of v^* at the fictitious point. An expression for $v^*_{i,0}$ can be determined from the continuity equation, which reduces to

$$\left.\frac{\partial v^*}{\partial y^*}\right)_{i,1} = 0 \tag{9.147}$$

at the wall. Using a third-order accurate finite-difference expression for this reduced form of the continuity equation,

$$\left.\frac{\partial v^*}{\partial y^*}\right)_{i,1} = \frac{-2v^*_{i,0} - 3\overset{0}{\cancel{v^*_{i,1}}} + 6v^*_{i,2} - v^*_{i,3}}{6(\Delta y^*)} + O\left[(\Delta y^*)^3\right] = 0 \tag{9.148}$$

allows us to compute $v^*_{i,0}$ and retain second-order accuracy in Eq. (9.146). Similar techniques can be used to find the pressure gradient at other boundaries in order to solve the Poisson pressure equation.

A very large number of methods can trace their ancestry back to the MAC method. In 1970, Amsden and Harlow (1970) introduced a simpler MAC procedure (SMAC), which employed a second Poisson equation for a velocity potential that would drive a corrective velocity in order to satisfy the continuity equation. Such a procedure is described in Chapter 8 in connection with the partially parabolized procedures for subsonic flows. The original MAC and SMAC schemes employed an explicit time-marching procedure. Implicit discretizations have been widely used in more recent variations of these schemes (Deville, 1974; Briley, 1974; Ghia et al., 1979).

Pressure-correction approach: Projection (fractional step) methods. A key feature of the MAC method is the splitting of the solution into two distinct steps, one of which is the solution of a Poisson equation for the pressure. A great many variations to this "splitting" of the solution procedure have been suggested. One such variation proposed by Chorin (1968) and Temam (1969) is known as the *projection method*, or the *method of fractional steps*. The projection

method was originally formulated on a regular rather than a staggered grid. Another difference is in the way the pressure Poisson equation is developed. In the original formulation proposed by Chorin, the pressure gradient terms are omitted from the momentum equations in the first step. The unsteady equations are advanced in time to obtain a provisional nondimensional velocity $\mathbf{V}^*$. In a second step, we wish to correct this provisional velocity by accounting for the pressure gradient and the continuity equation. This is achieved by considering

$$\frac{\mathbf{V}^{n+1} - \mathbf{V}^*}{\Delta t} + \nabla p^{n+1} = 0 \tag{9.149}$$

subject to the continuity constraint $\nabla \cdot \mathbf{V}^{n+1} = 0$. By taking the divergence of Eq. (9.149) subject to the continuity constraint above, we obtain the Poisson equation:

$$\nabla^2 p^{n+1} = \frac{\nabla \cdot \mathbf{V}^*}{\Delta t} \tag{9.150}$$

The solution procedure consists of first computing $\mathbf{V}^*$ from the momentum equations while neglecting the pressure gradient terms. The pressure Poisson equation is then solved for the pressure field, after which the velocities are computed from Eq. (9.149). More recently, investigators have found that the procedure also works well if a provisional pressure distribution is used in the momentum equations in the first step, where provisional velocities are determined. Then the p in Eq. (9.149) becomes a pressure correction, which can be determined from the solution to the Poisson equation.

The projection method has been implemented on both regular and staggered grids. On a staggered grid the scheme is very similar to the MAC method. Both explicit and implicit formulations have been employed. A rather detailed discussion of the projection method that points out similarities between it and the MAC scheme is provided by Peyret and Taylor (1983).

Pressure-correction approach: SIMPLE family of methods. The semi-implicit method for pressure linked equations (SIMPLE) algorithm of Caretto et al. (1972) and Patankar and Spalding (1972) which is introduced in Section 8.4.1 for the solution of the partially parabolized N-S equations can also be applied to the incompressible N-S equations (see Caretto et al., 1972; Patankar, 1975, 1981). This procedure is based on a cyclic series of guess-and-correct operations to solve the governing equations. The velocity components are first calculated from the momentum equations using a guessed pressure field. The pressures and velocities are then corrected, so as to satisfy continuity. This procedure continues until the solution converges. The main distinction between this method and the MAC and projection methods is in the way in which the pressure and velocity corrections are achieved.

In this procedure, the actual pressure p is written as

$$p = p_0 + p' \tag{9.151}$$

where p_0 is the estimated (or intermediate) value of pressure and p' is the pressure correction. Likewise, the actual velocity components (for a 2-D flow) are written as

$$\begin{aligned} u &= u_0 + u' \\ v &= v_0 + v' \end{aligned} \tag{9.152}$$

where u_0, v_0 are the estimated (or intermediate) values of velocity and u', v' are the velocity corrections. The pressure corrections are related to the velocity corrections by approximate forms of the momentum equations:

$$\begin{aligned} \rho\frac{\partial u'}{\partial t} &= -\frac{\partial p'}{\partial x} \\ \rho\frac{\partial v'}{\partial t} &= -\frac{\partial p'}{\partial y} \end{aligned} \tag{9.153}$$

Since the velocity corrections can be assumed to be zero at the previous iteration step, the above equations can be written as

$$\begin{aligned} u' &= -A\frac{\partial p'}{\partial x} \\ v' &= -A\frac{\partial p'}{\partial y} \end{aligned} \tag{9.154}$$

where A is a fictitious time increment divided by density. After combining Eqs. (9.152) and (9.154) and substituting the result into the continuity equation, we obtain

$$\cancelto{0}{\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)} - \left(\frac{\partial u_0}{\partial x} + \frac{\partial v_0}{\partial y}\right) + A\left(\frac{\partial^2 p'}{\partial x^2} + \frac{\partial^2 p'}{\partial y^2}\right) = 0 \tag{9.155}$$

or

$$\nabla^2 p' = \frac{1}{A}(\nabla \cdot \mathbf{V}_0) \tag{9.156}$$

where $\mathbf{V}_0$ is the estimated velocity vector. This Poisson equation can be solved for the pressure correction. Note that if the estimated velocity vector satisfies continuity at every point, then the pressure correction is zero at every point. In the actual SIMPLE algorithm, an equivalent differenced form of Eq. (9.156) is used as shown by Raithby and Schneider (1979).

The SIMPLE procedure can now be described by the following steps:

1. Guess the pressure (p_0) at each grid point.
2. Solve the momentum equations to find the velocity components (u_0, v_0). A staggered grid in conjunction with a block-iterative method is recommended by Patankar and Spalding.

3. Solve the pressure-correction equation [i.e., Eq. (9.156)] to find p' at each grid point.
4. Correct the pressure and velocity using Eqs. (9.151) and (9.154):

$$p = p_0 + p'$$

$$u = u_0 - \frac{A}{2\,\Delta x}(p'_{i+1,j} - p'_{i-1,j})$$

$$v = v_0 - \frac{A}{2\,\Delta y}(p'_{i,j+1} - p'_{i,j-1})$$

5. Replace the previous intermediate values of pressure and velocity (p_0, u_0, v_0) with the new corrected values (p, u, v), and return to Step 2. Repeat this process until the solution converges.

The SIMPLE procedure has been used successfully to solve a number of incompressible flow problems. However, in certain cases it is found that the rate of convergence is not satisfactory. This is due to the fact that the pressure-correction equation tends to overestimate the value of p' even though the corresponding velocity corrections are reasonable. Because of this, Eq. (9.151) is often replaced with

$$p = p_0 + \alpha_p p'$$

where α_p is an underrelaxation constant. For the same reason, underrelaxation is also employed in the solution of the momentum equations. In the present formulation, underrelaxation can be accommodated by varying the parameter A in Eqs. (9.154) and (9.156).

A large number of variations of the SIMPLE strategy have been proposed for the purpose of improving the convergence rate of the scheme. Among them are the SIMPLE revised (SIMPLER) scheme (Patankar, 1981) and the SIMPLEC scheme (van Doormal and Raithby, 1984).

The SIMPLER algorithm appears to make use of a feature contained in the fractional-step method, whereby provisional velocities are defined from a momentum equation in which the pressure gradient is absent. Patankar refers to these as pseudo-velocities. The algorithm starts with a guessed velocity field. With this, coefficients are computed for the momentum equations, and pseudo-velocities are obtained by solving a form of the momentum equations in which the pressure gradient is missing. These velocities are treated much as the provisional velocities of the SIMPLE algorithm to obtain a Poisson equation for pressure, making use of the continuity equation. The Poisson equation is solved for the pressure, which is then used to obtain a solution to the momentum equations. Velocity corrections are then computed as required to satisfy the continuity equation using the p' (or an alternative) procedure of the SIMPLE algorithm. This usually will require the solution of another Poisson-like equation for the p' field that will drive the corrections. Only the velocities are corrected with the p' solution. This revised scheme makes use of a more exact (less approximate) procedure for revising the pressure. Looking more closely at the

algorithm, it should be evident that if the first velocity field is the correct one and continuity is satisfied, then the pressure field computed with the help of the pseudo-velocities will also be the correct one. Thus, when the momentum equations are solved with this pressure field, a mass-conserving velocity field will be computed, and no further iterations will be required.

The SIMPLEC procedure developed by van Doormal and Raithby (1984) attempts to improve on the convergence rate of SIMPLE by using a more complete or consistent approximation to the momentum equations to compute p'. The scheme attempts to approximate the effects of some terms in the momentum equations neglected in the SIMPLE algorithm for p'. This is equivalent to modifying the A in Eq. (9.154). With this modification, van Doormal and Raithby reported that it was no longer necessary to underrelax the pressure correction. They also observed that SIMPLEC performed more efficiently than both SIMPLE and SIMPLER for the several test cases they considered. Other suggested improvements to SIMPLE and SIMPLER were included in their paper, which should be consulted for further details.

Pressure-correction approach: SIMPLE on nonstaggered grids. Until the early 1980s the SIMPLE family of methods was generally only employed on staggered grids. It is not straightforward to implement staggered grid schemes on general nonorthogonal curvilinear coordinate systems. However, use of nonstaggered (colocated) grids with the SIMPLE family of methods was observed to result in decoupling of the velocity and pressure fields, yielding "wiggles" in solutions. This provided motivation for numerical experimentation, and in 1981, several investigators (Hsu, 1981; Prakash, 1981; Rhie, 1981) reported success in implementing pressure-correction schemes on a regular grid. In effect, fourth-order pressure smoothing or dissipation was added to the continuity equation to achieve this result. The description given here follows the report of Rhie and Chow (1983).

The colocated scheme of Rhie and Chow (1983) follows the SIMPLE sequence. However, all variables are located at the same grid points. The momentum equations are solved using the best guess for the pressure field. Until the correct pressure field is established, this velocity field will not satisfy the continuity equation. The "trick" comes into play in computing the mass sources needed to correct the velocities and pressure. It is required in the SIMPLE strategy to compute the mass sources in each computational control volume or cell. On a colocated grid in which the grid points are on the interior of each control volume, this means that velocities need to be interpolated to the cell faces. Assuming a uniform grid in the Cartesian coordinate system, a simple linear interpolation would ordinarily be considered, giving for example, in two dimensions

$$\tilde{u}_{i+1/2,j} = \frac{\tilde{u}_{i,j} + \tilde{u}_{i+1,j}}{2}$$

where $\tilde{u}$ is the provisional x component of velocity obtained from solving the x momentum equation with a provisional pressure field. Inspection of the

momentum equations used to compute the velocities at i and $i + 1$ reveals that the cell-face velocity indicated above depends on velocities computed using centrally differenced pressure gradient terms formed from differences over two grid spacings. The Rhie-Chow interpolation scheme can be interpreted as an attempt to estimate the cell-face velocity that would have been computed if the resultant pressure gradient influencing the cell-face velocity were $(p_{i+1,j} - p_{i,j})/\Delta x$ (adjacent nodes only) instead of $(p_{i+2,j} - p_{i,j} + p_{i+1,j} - p_{i-1,j})/4\,\Delta x$, which follows from the momentum equations. The problem with the second expression is that it is insensitive to "checkerboard" oscillations because the differences are between every second node. The interpolation scheme to obtain cell-face velocities for computing the mass sources, as in Eq. (9.155), can be written as

$$\tilde{u}_{i+1/2,j} = \tilde{u}_{\text{li}} + B_{\text{li}}\left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x} - \left.\frac{\partial p}{\partial x}\right|_{\text{li}}\right) \tag{9.157}$$

where the subscript li denotes a value linearly interpolated to the cell face at $i + 1/2$ and B is the coefficient of the pressure gradient term in the momentum equation after it has been rearranged to isolate $u_{i,j}$ on the LHS. Thus Eq. (9.157) can be thought of as correcting the linearly interpolated velocity by providing a local value of the pressure gradient instead of the one resulting from the centrally differenced forms used in the momentum equations. For the uniform Cartesian grid example initiated above, Eq. (9.157) can be written as

$$\tilde{u}_{i+1/2,j} = \frac{\tilde{u}_{i,j} + \tilde{u}_{i+1,j}}{2} + B_{\text{li}}\left(\frac{p_{i+1,j} - p_{i,j}}{\Delta x} - \frac{p_{i+2,j} - p_{i,j} + p_{i+1,j} - p_{i-1,j}}{4\,\Delta x}\right) \tag{9.158}$$

which can be further rearranged into the form

$$\tilde{u}_{i+1/2,j} = \frac{\tilde{u}_{i,j} + \tilde{u}_{i+1,j}}{2} + \frac{B_{\text{li}}}{4\,\Delta x}(3p_{i+2,j} - 3p_{i,j} + p_{i+1,j} - p_{i-1,j}) \tag{9.159}$$

The term in parentheses can be recognized as the fourth difference commonly used in fourth-order dissipation, which formed the basis for the earlier observation that fourth-order dissipation of pressure is being added to the continuity equation. The cell-face values of velocities computed as indicated above are then used to compute the mass source term. To enforce mass conservation, velocity and pressure corrections are introduced as in SIMPLE. The pressure Poisson equation is solved for the pressure corrections, and then the velocity corrections at the nodes can be computed from a simplified relationship between nodal velocities and the pressure gradient having the form of Eq. (9.154). These correction relationships follow the discretization used in the momentum equations and do not make further use of the special interpolation formula for cell-face velocities discussed above.

Several investigators have compared the accuracy and computational efficiency of the colocated and staggered-grid version of the SIMPLE family of

methods. Among these studies are the works of Burns et al. (1986), Peric et al. (1988), and Melaaen (1992). Generally, the accuracy and convergence rate of both formulations have been found comparable. Examples have been cited in which each of the two schemes has been slightly more accurate than the other. Generally, however, the difference between the two results has been less than the estimated numerical error in the calculations of either scheme. There seems to be general agreement that the colocated approach is more convenient for working in curvilinear nonorthogonal coordinate systems and in three dimensions. Implementation of multigrid is also more straightforward with the colocated arrangement.

The SIMPLE family of methods has also been applied to compressible flows. For details, refer to the works of van Doormaal et al. (1987), Karki and Patankar (1989), McGuirk and Page (1989), and Shyy et al. (1992).

Pressure-correction approach: PISO (pressure-implicit with splitting of operators) method. A predictor-corrector strategy was proposed by Issa (1985) for solving the discretized time-dependent N-S equations in a sequential uncoupled manner. The scheme is applicable to both the incompressible and compressible forms of the equations and has been implemented on both colocated and staggered grids. The scheme is largely implicit, and various strategies for solving the simultaneous algebraic equations can be employed. The splitting strategy will be outlined here for incompressible flow using symbolic operator notation.

One predictor step and two corrector steps are utilized. Let the asterisks denote intermediate values computed during the splitting process. The calculation proceeds in the following steps.

1. *Predictor step.* The pressure field prevailing at time level n is used in the implicit solution of the momentum equations. This step is identical to the first step in the SIMPLE algorithm when the latter is applied to a time-dependent flow:

$$\frac{\rho}{\Delta t}(u_i^* - u_i^n) = H(u_i^*) - \Delta_i p^n + S_i \tag{9.160}$$

   Index notation is employed in Eq. (9.160), and the operator H stands for the finite-difference representation of the spatial convective and diffusive fluxes of momentum. The operator Δ_i is the finite-difference equivalent of $\partial/\partial x_i$. This velocity field will not generally satisfy the continuity equation.
2. *First corrector step.* In this step a new pressure field p^* is sought along with a revised velocity field u_i^{**} that will satisfy conservation of mass. Treating the velocities explicitly, the momentum equation is considered in the form

$$\frac{\rho}{\Delta t}(u_i^{**} - u_i^n) = H(u_i^*) - \Delta_i p^* + S_i \tag{9.161}$$

Requiring that $\Delta_i u_i^{**} = 0$ and taking the divergence of Eq. (9.161) gives a discretized Poisson equation:

$$\Delta_i^2 p^* = \Delta_i H(u_i^*) + \Delta_i S_i + \frac{\rho}{\Delta t} \Delta_i u_i^n \tag{9.162}$$

Note that the Poisson equation can be immediately solved for pressure p^*, since the RHS contains quantities already determined in the predictor step. This pressure field can then be used in Eq. (9.161) to compute u_i^{**}, which by design should satisfy the continuity equation.

3. *Second corrector step.* This step is essentially a recorrection, using the strategy outlined above. The momentum equation is considered in the form

$$\frac{\rho}{\Delta t}(u_i^{***} - u_i^n) = H(u_i^{**}) - \Delta_i p^{**} + S_i \tag{9.163}$$

Taking the divergence of Eq. (9.163) and requiring the continuity equation to be satisfied with the new velocity field, $\Delta_i u_i^{***} = 0$, gives an equation that can be solved for an updated pressure field:

$$\Delta_i^2 p^{**} = \Delta_i H(u_i^{**}) + \Delta_i S_i + \frac{\rho}{\Delta t} \Delta_i u_i^n \tag{9.164}$$

The pressure field is first determined from Eq. (9.164) and then used in the momentum equation, Eq. (9.163), where the updated velocity field, u_i^{***}, is computed. Following this format, more recorrections can be made, but Issa (1985) suggests that the two correction steps should be sufficient for most purposes.

Issa (1985) discusses the errors associated with the method and argues that the splitting errors are sufficiently small that time-accurate solutions can be obtained without iterative application of the algorithm using time steps dictated only by the accuracy of the difference scheme. The favorable features of the scheme were demonstrated by Issa et al. (1986) in a paper that considered time-dependent and subsonic compressible flows. A successive overrelaxation by lines procedure was used to solve the simultaneous linearized algebraic equations resulting from a finite-volume discretization on a staggered grid.

PROBLEMS

9.1 Show how all the terms in the 2-D y momentum equation are differenced when the explicit MacCormack (1969) method is applied to the compressible N-S equations.

9.2 Repeat Prob. 9.1 for the 2-D energy equation.

9.3 Apply the explicit MacCormack scheme to the N-S equations written in cylindrical coordinates (see Section 5.1.8), and show how all the terms in the r momentum equation are differenced.

9.4 Apply the Allen-Cheng method instead of the explicit MacCormack method in Prob. 9.1.

9.5 Derive the Jacobian matrix $[A]$ given by Eq. (9.46).

9.6 Derive the Jacobian matrix $[B]$ given by Eq. (9.48).

9.7 Derive the Jacobian matrix $[R]$ given by Eq. (9.51).

9.8 Derive the Jacobian matrix $[S]$ given by Eq. (9.54).
9.9 Derive the matrix $[P] - [R_x]$ given by Eq. (9.50).
9.10 Derive the matrix $[Q] - [S_y]$ given by Eq. (9.53).
9.11 Determine the amplification factor G for the explicit MacCormack scheme applied to the linearized Burgers equation. Does Eq. (4.313) satisfy $|G| \leqslant 1$ for all values of β when $\nu = \frac{1}{2}$ and $r = \frac{1}{4}$?
9.12 Repeat Prob. 9.11 for $\nu = 1$ and $r = \frac{1}{2}$.
9.13 Use the explicit MacCormack method to solve the linearized Burgers equation for the initial condition

$$u(x,0) = 0 \qquad 0 \leqslant x \leqslant 1$$

and the boundary conditions

$$u(0,t) = 100$$
$$u(1,t) = 0$$

on a 21 grid point mesh. Find the steady-state solution for the conditions

$$r = 0.5$$
$$\nu = 0.5$$

and compare the numerical solution with the exact solution.
9.14 Derive the Jacobian matrix $[A_t]$ given by Eq. (9.74).
9.15 Derive the Jacobian matrix $[A_x]$ given by Eq. (9.75).
9.16 Formulate a preconditioned implicit scheme for solving the 2-D compressible N-S equations at low Mach numbers using primitive variables. Derive all necessary Jacobian matrices. Explain your work.
9.17 Explain how preconditioning can be applied to the 2-D compressible N-S equations when conserved variables are employed in an implicit formulation.
9.18 Obtain Eq. (9.124).
9.19 Solve the square driven cavity problem for $\mathrm{Re}_l = 50$. Use the forward-time, centered-space (FTCS) method (Section 4.5.1) to solve the vorticity transport equation and the successive overrelaxation method to solve the Poisson equation. Employ a first-order evaluation of the vorticity at the wall, and use an 8×8 grid.
9.20 Repeat Prob. 9.19 for $\mathrm{Re}_l = 100$ and a 15×15 grid.
9.21 Derive the vorticity transport equations for a 3-D Cartesian coordinate system.
9.22 Use the artificial compressibility method to solve the square driven cavity problem for $\mathrm{Re}_l = 100$. Apply the leap frog/DuFort-Frankel finite-difference scheme to the governing equations on a 15×15 grid. Determine pressure at the wall using a suitable finite-difference representation of the normal momentum equation applied at the wall.
9.23 Use the method of artificial compressibility to solve the steady square driven cavity problem for $\mathrm{Re}_l = 100$. Use a coupled implicit scheme. Compare your solution from a 21×21 grid with the results of Ghia et al. (1982).
9.24 Add a pseudo-time term to the method of Prob. 9.23, and compute the driven cavity problem for $\mathrm{Re}_l = 100$ in a time-accurate manner, starting the lid impulsively from rest. Plot the x component of velocity at the center of the cavity as a function of time.
9.25 Use a preconditioned compressible formulation for the 2-D N-S equations to solve the steady driven cavity problem for $\mathrm{Re}_l = 100$. Obtain solutions at Mach numbers of 0.2, 0.1, 0.01, and 0.001. Compare the convergence histories of the scheme for the Mach numbers specified.