Table 12.3.4 *Friction coefficient computed using the different meshes.*

	Number of cells in the boundary layer (first one millimeter)	*Cf*	*Relative error in Cf (%)*
Analytical		0.000713	–
65 × 65	28	0.00075	5.2
65 × 33	14	0.000803	12.7
65 × 17	7	0.000917	28.7
65 × 9	4	0.001371	92.3
65 × 5	2	0.002196	208.0

in Hemsch and Morrison (2004), also referred to in the general introduction to this book.

The errors on the friction coefficient are a consequence of the reduced accuracy on the velocity profiles, when the mesh is coarsened. This can be seen from Figure 12.3.9, where the distributions of skin friction and velocity are displayed on the two successive coarser grids, namely 65 × 33 and 65 × 17, having respectively, 14 and 7 points in the boundary layer. You will notice that 14 points might still be acceptable, for 'engineering' accuracy, but the error on the skin friction is of 12%, as seen from Table 12.3.4.

12.4 PRESSURE CORRECTION METHOD

The methods known as ***pressure correction*** are among the first developed for the numerical solutions of the full Navier–Stokes equations for incompressible flows. The method was originally applied by Harlow and Welch (1965) in the MAC, Marker-and-Cell, method for the computation of free surface incompressible flows. It is closely related to the ***fractional step method***, also called ***projection method***, developed independently by Chorin (1967), (1968) and Temam (1969); see also Temam (1977).

It has been adapted to industrial flow simulations by Patankar and Spalding (1972) and described in details in several books (Patankar (1980); Anderson (1995); Ferziger and Peric (1997); Wesseling (2001)).

The methods falling in this class can be applied to the stationary as well as to the time-dependent incompressible flow equations. They consist of a basic iterative procedure between the velocity and the pressure fields. For an initial approximation of the pressure, the momentum equation can be solved to determine the velocity field. The obtained velocity field does not satisfy the divergence free, continuity equation and has therefore to be corrected. Since this correction has an impact on the pressure field, a related pressure correction is defined, obtained by expressing that the corrected velocity satisfies the continuity equation. This leads to a Poisson equation for the pressure correction.

The pressure correction methods have been extended to compressible flows, and various approaches can be defined. Weak compressibility can be handled through a simplified system of equations obtained by developing a low Mach number expansion of the Navier–Stokes equations, omitting then the terms that are of higher order in

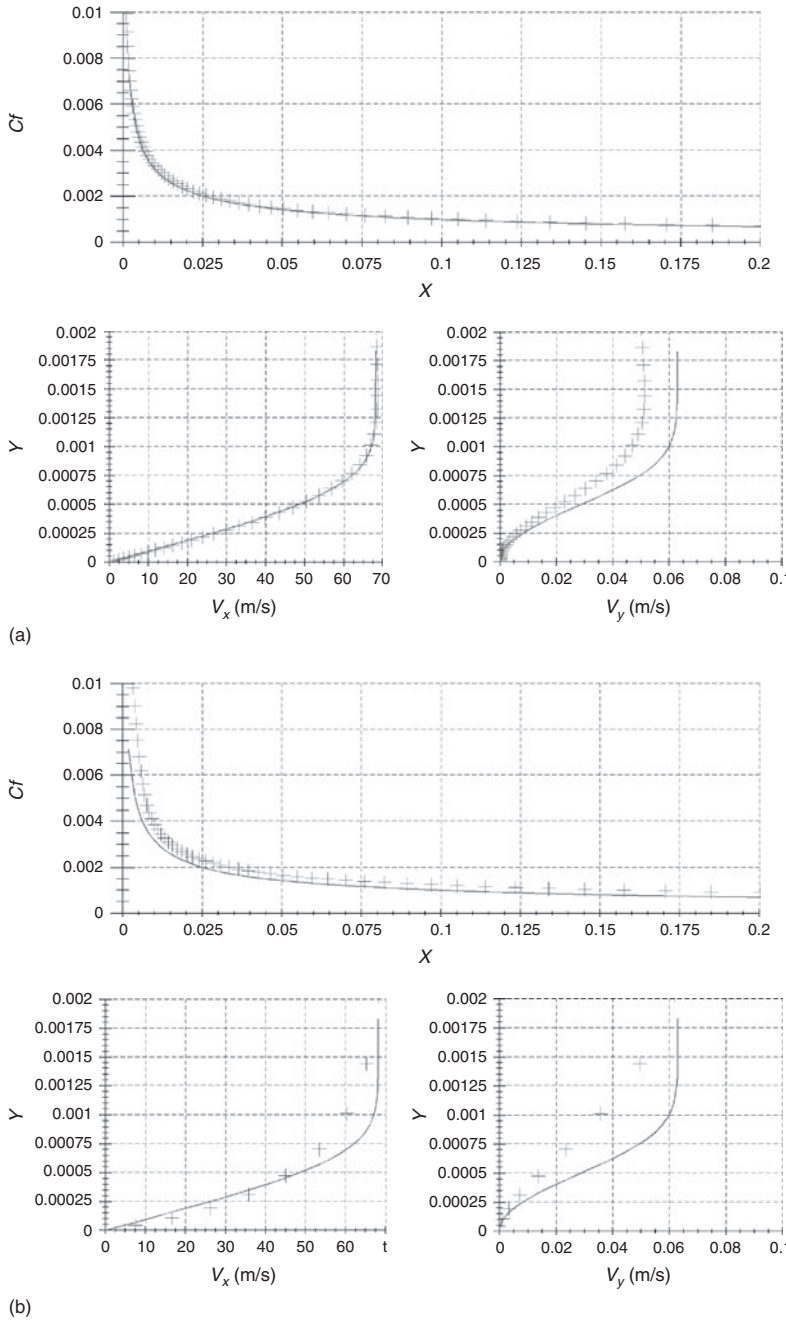

Figure 12.3.9 *Distributions of skin friction and velocity on the two successive coarser grids of 65 × 33 and 65 × 17, having respectively, 14 and 7 points in the boundary layer. (a) Numerical solution on 65 × 33 mesh with 14 points in the boundary layer and (b) numerical solution on 65 × 17 mesh with 7 points in the boundary layer.*

the Mach number (see, for instance, Majda and Sethian (1985)). A full compressible extension can be found in the book of Ferziger and Peric (1997).

12.4.1 Basic Approach of Pressure Correction Methods

We restrict the presentation here to strictly incompressible, isothermal flows, defined in Section 1.4.4 by the system formed by the continuity and momentum equations, written here in non-conservative form.

The mass conservation equation reduces in the case of incompressible flows to

$$\vec{\nabla} \cdot \vec{v} = 0 \tag{12.4.1}$$

which appears as a constraint to the general time-dependent equation of motion, written here in absence of external volume forces

$$\frac{\partial \vec{v}}{\partial t} + (\vec{v} \cdot \vec{\nabla})\vec{v} = -\frac{1}{\rho}\vec{\nabla} p + \nu \Delta \vec{v} \tag{12.4.2}$$

The only unknowns are velocity and pressure.

An equation for the pressure can be obtained by taking the divergence of the momentum equation (12.4.2), and introducing the divergence free velocity condition (12.4.1), leading to

$$\frac{1}{\rho}\Delta p = -\vec{\nabla} \cdot (\vec{v} \cdot \vec{\nabla})\vec{v} \tag{12.4.3}$$

which can be considered as a Poisson equation for the pressure for a given velocity field. Note that the right-hand side contains only products of first order velocity derivatives, because of the incompressibility condition (12.4.1). Indeed, in tensor notations, the velocity term in the right-hand side term is equal to $(\partial_j v_i) \cdot (\partial_i v_j)$.

Before describing the pressure correction method we have to select a time integration scheme for the momentum equations, considering the pressure gradient as known. For reasons of simplicity and in order to point out the essential properties of the pressure correction approach, we will select an explicit method of first order accuracy in time, although it is not recommended in practice. Even for time-dependent problems the time step restriction imposed by stability conditions for the parabolic, convection–diffusion momentum equations is generally smaller than the physical time constant of the flow. Hence, the time steps allowed by the requirements of physical accuracy are large enough to allow the larger numerical time steps of implicit schemes. Typically, semi-implicit time integration schemes are recommended. For instance, the viscous fluxes can be treated implicitly by means of the Crank–Nicholson formulation, while convective fluxes can be handled with an Adams–Bashworth second order method. This provides a higher accuracy for time-dependent simulations and allows for large time steps leading to more efficient calculations in terms of computational costs.

The fundamental approach of pressure correction methods is the decoupling of the pressure field from the velocity field. This is expressed by solving the momentum

equation with a known pressure field, for instance with the pressure obtained at the previous iteration.

A variety of methods have been developed and applied in practice, based on various decoupling approaches. In its simplest form, with an explicit time discretization, we solve for an intermediate velocity field $\vec{v}^*$, solution of

$$\frac{\vec{v}^* - \vec{v}^n}{\Delta t} = -\vec{\nabla} \cdot (\vec{v} \otimes \vec{v})^n - \frac{1}{\rho}\vec{\nabla} p^n + \nu \Delta \vec{v}^n \tag{12.4.4}$$

The solution $\vec{v}^*$ of this equation does not satisfy the continuity equation. Hence the final values are defined by adding corrections to the intermediate values

$$\vec{v}^{n+1} = \vec{v}^* + \vec{v}' \quad p^{n+1} = p^n + p' \tag{12.4.5}$$

where the final values with superscript $n+1$ have to be solutions of

$$\begin{aligned} \frac{\vec{v}^{n+1} - \vec{v}^n}{\Delta t} &= -\vec{\nabla} \cdot (\vec{v} \otimes \vec{v})^n - \frac{1}{\rho}\vec{\nabla} p^{n+1} + \nu \Delta \vec{v}^n \\ \vec{\nabla} \cdot \vec{v}^{n+1} &= 0 \end{aligned} \tag{12.4.6}$$

Introducing (12.4.5) in the above equation and subtracting (12.4.4), leads to the following relation between the pressure and velocity corrections:

$$\vec{v}' = -\frac{\Delta t}{\rho}\vec{\nabla} p' \tag{12.4.7}$$

Note that expressing the velocity correction as a gradient of a scalar function conserves the vorticity of the intermediate velocity field. That is, the correction field is a potential flow.

Taking the divergence of the first of the equations (12.4.6) gives the Poisson equation for the pressure correction:

$$\Delta p' = \frac{\rho}{\Delta t}\vec{\nabla} \cdot \vec{v}^* \tag{12.4.8}$$

Equation (12.4.3) assumes that the solution at time level n satisfies exactly the divergence-free condition. In the numerical process, the velocity at level n might not satisfy exactly this condition. In this case, the non-zero value of $D^n \triangleq \vec{\nabla} \cdot \vec{v}^n$ should be introduced in the pressure Poisson equation. This situation is more likely to occur in stationary computations where n represents an iteration count. With time-dependent calculations, it is recommended to satisfy accurately mass conservation at each time step, in particular by discretizing the integral form of the mass conservation law on a finite volume mesh.

The Poisson equation for the pressure is solved with Neumann boundary conditions, on the normal pressure gradient, obtained by taking the normal component of equations (12.4.6). The details of the implementation depend on the selected space discretization and on the mesh.

An alternative approach is the ***fractional step, or projection***, method based on a slightly different definition of the intermediate pressure, whereby the pressure term is simply omitted, leading to the complete decoupling of the intermediate velocity field

$$\frac{\vec{v}^* - \vec{v}^n}{\Delta t} = -\vec{\nabla} \cdot (\vec{v} \otimes \vec{v})^n + \nu \Delta \vec{v}^n \tag{12.4.9}$$

followed by the pressure equation

$$\frac{\vec{v}^{n+1} - \vec{v}^*}{\Delta t} = -\frac{1}{\rho} \vec{\nabla} p^{n+1} \tag{12.4.10}$$

The final value of velocity field is obtained from equation (12.4.10). The pressure is calculated in such a way that the velocity field at level $(n+1)$ satisfies the divergence free condition:

$$\frac{\rho}{\Delta t} [\vec{\nabla} \cdot \vec{v}^{n+1} - \vec{\nabla} \cdot \vec{v}^*] = -\Delta p^{n+1} \tag{12.4.11}$$

leading to the an equation similar to (12.4.8)

$$\Delta p^{n+1} = \frac{\rho}{\Delta t} \vec{\nabla} \cdot \vec{v}^* \tag{12.4.12}$$

The final value of the velocity field $\vec{v}^{n+1}$ velocities are updated from (12.4.10).

The numerical resolution of the pressure Poisson equation is a crucial step of the whole approach, since the overall efficiency of the code will depend on its performance. Hence all possible convergence optimization and acceleration techniques should be applied. In particular preconditioning and multigrid techniques are strongly recommended for this step of the computation, and eventually for other steps.

12.4.2 The Issue of Staggered Versus Collocated Grids

The choice of a space discretization is, as for compressible flows, between centered or upwind methods, at least for the convection terms, since the diffusive contributions are always centrally discretized.

The most current choice is the central discretization of the convection terms, which raises a particular problem with the pressure correction approach.

The central discretization for the convection terms requires the addition of higher order artificial dissipation terms to create the required damping of high frequency errors, as introduced in the previous examples. However the absence of the time derivative of the density in the continuity equation creates an additional decoupling in the centrally discretized equations, with pressure correction methods. This is best illustrated on the one-dimensional system of incompressible Navier–Stokes equations. The conservation equations take the form

$$\begin{aligned} &\frac{\partial u}{\partial x} = 0 \\ &\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2} \end{aligned} \tag{12.4.13}$$

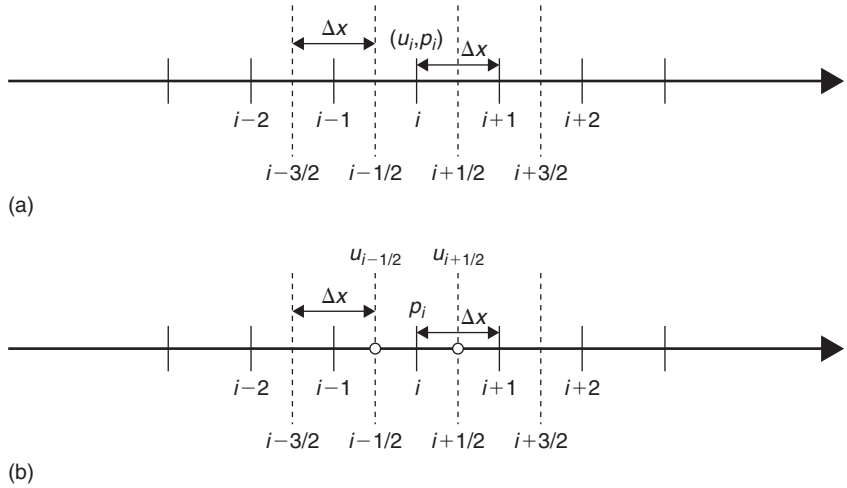

Figure 12.4.1 *Standard 'collocated' and staggered grids: (a) standard collocated grid and (b) staggered grid.*

We consider a central finite difference discretization of the above equations on a standard uniform grid, where all the variables are defined on the same mesh points. This is called a ***collocated mesh***, in the context of pressure correction methods (see Figure 12.4.1a). The centrally discretized equations become

$$\frac{u_{i+1}^{n+1} - u_{i-1}^{n+1}}{2\Delta x} = 0 \tag{12.4.14}$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{(u_{i+1}^n)^2 - (u_{i-1}^n)^2}{2\Delta x} = -\frac{1}{\rho}\frac{p_{i+1} - p_{i-1}}{2\Delta x} + \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{12.4.15}$$

In the framework of the pressure correction method, the momentum equation is split into two parts by introducing the intermediate velocity u^*, based on the fractional step method, solution of

$$\frac{u_i^* - u_i^n}{\Delta t} + \frac{(u_{i+1}^n)^2 - (u_{i-1}^n)^2}{2\Delta x} = \nu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2} \tag{12.4.16}$$

$$\frac{u_i^{n+1} - u_i^*}{\Delta t} = -\frac{1}{\rho}\frac{p_{i+1} - p_{i-1}}{2\Delta x} \tag{12.4.17}$$

The expression for u_i^{n+1}, obtained from equation (12.4.17), can be substituted into (12.4.14), resulting in the following 1D equation:

$$\frac{p_{i+2} - 2p_i + p_{i-2}}{4\Delta x^2} = \frac{\rho}{\Delta t}\frac{u_{i+1}^* - u_{i-1}^*}{2\Delta x} \tag{12.4.18}$$

This is a Poisson equation for the pressure, which ensures that the continuity equation (12.4.14) is satisfied for the newly updated velocity u_i^{n+1}. The stencil on which the Laplace operator is discretized $(i+2, i, i-2)$ contains, however, only odd or even indices, which leads to a decoupling of the discrete pressure field and often results in high frequency oscillations of pressure.

As can be seen indeed, the pressure at point i is not influenced by the velocity component u_i^n of the same point and in return u_i^n is not affected by p_i. Hence velocity and pressure are decoupled on even and odd points; see also Section 4.2 and the discussion around the Lax–Friedrichs scheme in Chapter 7, for an illustration of analog cases. This decoupling is not present with compressible flows due to the density–velocity coupling in the continuity equation. It will generate additional high frequency oscillations, requesting the introduction of artificial dissipation terms.

A solution to the odd–even decoupling problem, has been introduced by Harlow and Welch (1965), by defining a ***staggered mesh***, where the velocity and pressure are not defined in the same mesh points. As seen in Figure 12.4.1b, the velocity is directly defined at the half mesh points, while the pressure remains defined at the central mesh point. The central discretization of the continuity equation of (12.4.13) now becomes

$$\frac{u_{i+1/2}^{n+1} - u_{i-1/2}^{n+1}}{\Delta x} = 0 \tag{12.4.19}$$

With the fractional step method, equation (12.4.17) becomes on the staggered mesh

$$\frac{u_{i+1/2}^{n+1} - u_{i+1/2}^{*}}{\Delta t} = -\frac{1}{\rho}\frac{p_{i+1} - p_i}{\Delta x} \tag{12.4.20}$$

By substituting expressions for $u_{i+1/2}^{n+1}$ and $u_{i-1/2}^{n+1}$ derived from (12.4.20) into this equation, we obtain

$$\frac{p_{i+1} - 2p_i + p_{i-1}}{\Delta x^2} = \frac{\rho}{\Delta t}\frac{u_{i+1/2}^{*} - u_{i-1/2}^{*}}{\Delta x} \tag{12.4.21}$$

In this discrete Poisson equation the pressure and velocity in all nodes are fully coupled, which completely eliminates the problem of odd–even decoupling.

Staggered meshes are currently applied with central discretization and the most popular two-dimensional arrangement is shown in Figure 12.4.2, where the u and v velocity components are located on different cell faces. The equations are discretized in conservation form, the control volumes depending on the considered equations. The mass equation is discretized on the volume centered on the point (i, j), while the x-momentum conservation is expressed on the volume centered for the location of u, i.e. $(i+1/2, j)$. Similarly, the y-momentum conservation is expressed on the volume centered on the location of v, i.e. $(i, j+1/2)$.

The Poisson equation for the pressure is obtained from the divergence of the discretized momentum equation. *This step should be performed by exactly the same discrete operations as applied to express mass conservation.* This is required for global consistency and conservation. It is fairly straightforward on a Cartesian mesh, but becomes essential on arbitrary meshes.

The Poisson equation for the pressure is solved with Neumann boundary conditions on the normal pressure gradient on the walls and at the inlet, while at the outlet

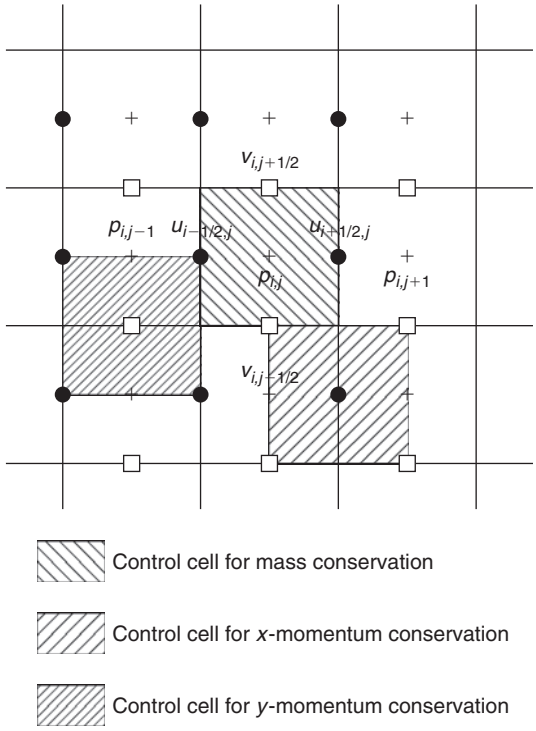

Figure 12.4.2 *Staggered, two-dimensional finite difference mesh for centrally discretized pressure correction methods.*

the pressure is set to a certain value (e.g. atmospheric pressure). The details of the implementation depend on the selected space discretization and on the mesh.

An additional condition is essential for the numerical accuracy of the resolution of the pressure equation, namely that the compatibility condition, obtained from Green's theorem applied to the Poisson equation, should be identically satisfied by the space discretization. Applied to equation (12.4.12), we should have identically, for the integral of the normal pressure gradient on boundary $\vec{S}$ of the computational domain Ω:

$$\int_{\Omega} \Delta p^{n+1} \mathrm{d}\Omega = \oint_{S} \vec{\nabla} p \cdot \mathrm{d}\vec{S} = \oint_{S} \frac{\partial p}{\partial n} \mathrm{d}S = \frac{\rho}{\Delta t} \int_{\Omega} \nabla \cdot \vec{v}^{*} \mathrm{d}\Omega = \frac{\rho}{\Delta t} \oint_{S} \vec{v}^{*} \cdot \mathrm{d}\vec{S} \tag{12.4.22}$$

12.4.3 Implementation of a Pressure Correction Method

We consider the application of the fractional step method to the simulation of incompressible flows. The method is described for a cell-centered finite volume method on a Cartesian mesh, with all flow variables defined in the same points at the centers of the computational cells, although the staggered grid approach is applied to connect values at the cell centers with face defined values, as seen on Figure 12.4.3, where different control volumes are shown. Boundaries of the computational domain are

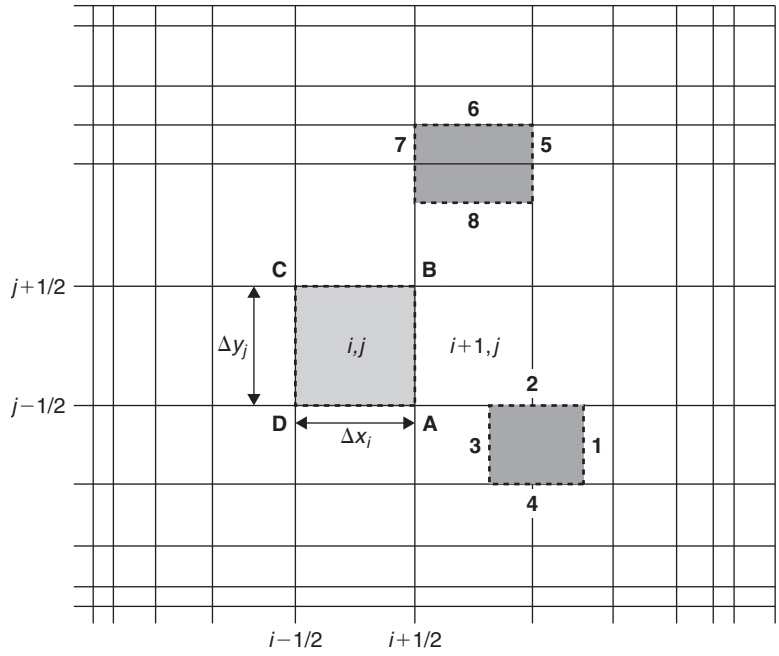

Figure 12.4.3 *Cell-centered finite volume mesh for pressure correction method on non-uniform Cartesian grid.*

located on cell faces and you can follow the treatment of the boundary conditions as described in Section 12.2.

Another way to impose boundary conditions in such a configuration can be considered through introduction of ghost or dummy cells, i.e. rows of cells neighboring the computational domain and having the same size as the first row of cells inside the domain.

You can extend the generality of your code by allowing for non-uniform Cartesian grids to account for clustered grids near solid walls. We consider a rectangular domain with side lengths L_x and L_y with N_x and N_y mesh points in both directions and variable mesh sizes. The Cartesian mesh has $N_x \times N_y$ nodes, dividing the domain into $(N_x - 1)(N_y - 1)$ rectangular cells, the sides of varying sizes: $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$.

The two-dimensional incompressible Navier–Stokes equations are discretized on the non-uniform Cartesian grid of Figure 12.4.3. A cell-centered second order finite volume discretization is selected, with an explicit first order time integration.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$
$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xx}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} + \nu\frac{\partial^2 u}{\partial y^2}$$
$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{1}{\rho}\frac{\partial \tau_{xy}}{\partial x} + \frac{1}{\rho}\frac{\partial \tau_{yy}}{\partial y} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\frac{\partial^2 v}{\partial x^2} + \nu\frac{\partial^2 v}{\partial y^2} \tag{12.4.23}$$

12.4.3.1 *Numerical discretization*

In the pressure correction method, the velocities are updated from the momentum equations, while pressure is obtained by solving the Poisson equation derived by taking a divergence of the momentum equations and taking conservation of mass into account.

We refer to Figure 12.4.3 and apply a finite volume formulation on the contour ABCD for the velocity components u and v at the cell centers.

Applying the fractional step formulation of equations (12.4.9) to (12.4.12), we write the scheme as follows, for the intermediate velocity components u^* and v^*

$$\frac{u^*_{i,j}-u^n_{i,j}}{\Delta t}+\frac{(u^2)^n_{i+1/2,j}-(u^2)^n_{i-1/2,j}}{\Delta x_i}+\frac{(uv)^n_{i,j+1/2}-(uv)^n_{i,j-1/2}}{\Delta y_j}$$
$$=\frac{1}{\rho}\frac{(\tau_{xx})^n_{i+1/2,j}-(\tau_{xx})^n_{i-1/2,j}}{\Delta x_i}+\frac{1}{\rho}\frac{(\tau_{xy})^n_{i,j+1/2}-(\tau_{xy})^n_{i,j-1/2}}{\Delta y_j} \tag{12.4.24}$$

$$\frac{v^*_{i,j}-v^n_{i,j}}{\Delta t}+\frac{(uv)^n_{i+1/2,j}-(uv)^n_{i-1/2,j}}{\Delta x_i}+\frac{(v^2)^n_{i,j+1/2}-(v^2)^n_{i,j-1/2}}{\Delta y_j}$$
$$=\frac{1}{\rho}\frac{(\tau_{yx})^n_{i+1/2,j}-(\tau_{yx})^n_{i-1/2,j}}{\Delta x_i}+\frac{1}{\rho}\frac{(\tau_{yy})^n_{i,j+1/2}-(\tau_{yy})^n_{i,j-1/2}}{\Delta y_j} \tag{12.4.25}$$

where the interface values are obtained by the weighted averages

$$u_{i+1/2,j}=\frac{\Delta x_i u_{i+1,j}+\Delta x_{i+1}u_{i,j}}{\Delta x_i+\Delta x_{i+1}}$$
$$v_{i,j+1/2}=\frac{\Delta y_j v_{i,j+1}+\Delta y_{j+1}v_{i,j}}{\Delta y_j+\Delta y_{j+1}} \tag{12.4.26}$$

and similarly for the other variables.

You can calculate the shear stress components as follows:

$$(\tau_{xx})^n_{i+1/2,j}=2\mu^n_{i+1/2,j}(u_x)^n_{i+1/2,j}$$
$$(\tau_{xy})^n_{i+1/2,j}=\mu^n_{i+1/2,j}((u_y)^n_{i+1/2,j}+(v_x)^n_{i+1/2,j})$$
$$(u_x)^n_{i+1/2,j}=\frac{u^n_{i+1,j}-u^n_{i,j}}{(\Delta x_{i+1}+\Delta x_i)/2}\quad (v_x)^n_{i+1/2,j}=\frac{v^n_{i+1,j}-v^n_{i,j}}{(\Delta x_{i+1}+\Delta x_i)/2}$$
$$(u_y)^n_{i+1/2,j}=\frac{1}{2}\left[\frac{u^n_{i+1,j+1}-u^n_{i+1,j-1}}{(\Delta y_{j+1}+2\Delta y_j+\Delta y_{j-1})/2}+\frac{u^n_{i,j+1}-u^n_{i,j-1}}{(\Delta y_{j+1}+2\Delta y_j+\Delta y_{j-1})/2}\right]$$
$$(v_y)^n_{i+1/2,j}=\frac{1}{2}\left[\frac{v^n_{i+1,j+1}-v^n_{i+1,j-1}}{(\Delta y_{j+1}+2\Delta y_j+\Delta y_{j-1})/2}+\frac{v^n_{i,j+1}-v^n_{i,j-1}}{(\Delta y_{j+1}+2\Delta y_j+\Delta y_{j-1})/2}\right] \tag{12.4.27}$$

$$
\begin{aligned}
(\tau_{yy})^n_{i,j+1/2} &= 2\mu^n_{i,j+1/2}(v_y)^n_{i+1/2,j} \\
(\tau_{yx})^n_{i,j+1/2} &= \mu^n_{i,j+1/2}((u_y)^n_{i,j+1/2} + (v_x)^n_{i,j+1/2}) \\
(u_y)^n_{i,j+1/2} &= \frac{u^n_{i,j+1} - u^n_{i,j}}{(\Delta y_{j+1} + \Delta y_j)/2} \quad (v_y)^n_{i,j+1/2} = \frac{v^n_{i,j+1} - v^n_{i,j}}{(\Delta y_{j+1} + \Delta y_j)/2} \\
(u_x)^n_{i,j+1/2} &= \frac{1}{2}\left[\frac{u^n_{i+1,j+1} - u^n_{i-1,j+1}}{(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})/2} + \frac{u^n_{i+1,j} - u^n_{i-1,j}}{(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})/2}\right] \\
(v_x)^n_{i,j+1/2} &= \frac{1}{2}\left[\frac{v^n_{i+1,j+1} - v^n_{i-1,j+1}}{(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})/2} + \frac{v^n_{i+1,j} - v^n_{i-1,j}}{(\Delta x_{i+1} + 2\Delta x_i + \Delta x_{i-1})/2}\right]
\end{aligned}
\tag{12.4.28}
$$

Once the intermediate velocity components are estimated, you can obtain the values of the updated velocity and pressure variables, satisfying the continuity equation, following (12.4.10). Applying a finite volume formulation on the contour ABCD to this equation and projecting in the x- and y-directions, we obtain

$$
\begin{aligned}
\frac{u^{n+1}_{i,j} - u^*_{i,j}}{\Delta t} &= -\frac{1}{\rho}\frac{p_{i+1/2,j} - p_{i-1/2,j}}{\Delta x_i} \\
\frac{v^{n+1}_{i,j} - v^*_{i,j}}{\Delta t} &= -\frac{1}{\rho}\frac{p_{i,j+1/2} - p_{i,j+1/2}}{\Delta y_j}
\end{aligned}
\tag{12.4.29}
$$

The pressure is obtained by expressing that the velocity components at level $(n+1)$ satisfy the divergence free continuity equation, as in (12.4.6). Discretized on the finite volume mesh ABCD we obtain

$$
\frac{u^{n+1}_{i+1/2,j} - u^{n+1}_{i-1/2,j}}{\Delta x_i} + \frac{v^{n+1}_{i,j+1/2} - v^{n+1}_{i,j-1/2}}{\Delta y_j} = 0 \tag{12.4.30}
$$

The interface velocities at the $(n+1)$ level are obtained by applying once again the finite volume formulation of equation (12.4.10), but this time on a staggered control volume such as 1234, Figure 12.4.3, centered on face AB, for the x component equation

$$
\rho\frac{u^{n+1}_{i+1/2,j} - u^*_{i+1/2,j}}{\Delta t}\Delta y_j\frac{\Delta x_{i+1} + \Delta x_i}{2} = (p^{n+1}_{i+1,j} - p^{n+1}_{i,j})\Delta y_j \tag{12.4.31}
$$

and on the control volume 5678, centered around BC, for the vertical component, leading to

$$
\rho\frac{v^{n+1}_{i,j+1/2} - v^*_{i,j+1/2}}{\Delta t}\Delta x_i\frac{\Delta y_{j+1} + \Delta y_j}{2} = (p^{n+1}_{i,j+1} - p^{n+1}_{i,j})\Delta x_i \tag{12.4.32}
$$

The values on faces CD and DA are obtained similarly.

The intermediate values u^* and v^* on the cell faces are obtained from the relations (12.4.26).

Substituting these relations in equation (12.4.30), leads to the pressure Poisson equation:

$$\begin{aligned}
&\frac{1}{\Delta x_i}\left[\frac{p^{n+1}_{i+1,j}-p^{n+1}_{i,j}}{(\Delta x_{i+1}+\Delta x_i)/2}-\frac{p^{n+1}_{i,j}-p^{n+1}_{i-1,j}}{(\Delta x_i+\Delta x_{i-1})/2}\right]\\
&\quad+\frac{1}{\Delta y_j}\left[\frac{p^{n+1}_{i,j+1}-p^{n+1}_{i,j}}{(\Delta y_{j+1}+\Delta y_j)/2}-\frac{p^{n+1}_{i,j}-p^{n+1}_{i,j-1}}{(\Delta y_j+\Delta y_{j-1})/2}\right]\\
&\quad=\frac{\rho}{\Delta t}\left[\frac{u^*_{i+1/2,j}-u^*_{i-1/2,j}}{\Delta x_i}+\frac{v^*_{i,j+1/2}-v^*_{i,j-1/2}}{\Delta y_j}\right]
\end{aligned}\tag{12.4.33}$$

The whole procedure of updating the solution is the following:

- Calculate the intermediate velocity field $u^*_{i,j}, v^*_{i,j}$ from (12.4.24) and (12.4.25).
- Obtain the pressure by solving the Poisson equation (12.4.33).
- Obtain the solution at the next time step $u^{n+1}_{i,j}, v^{n+1}_{i,j}$ from (12.4.29), where the pressure at the cell faces is obtained by applying relations (12.4.26).

We have now to focus on the most critical issue of pressure correction methods, namely the efficient resolution of the pressure Poisson equation. This is a crucial step of the whole approach, since the overall efficiency of your code will depend on its performance.

12.4.3.2 *Algorithm for the pressure Poisson equation*

The pressure Poisson equation (12.4.33) is a standard elliptic equation and you can call upon the various methods introduced in Chapter 10. In advanced codes, various convergence optimization and acceleration techniques are applied, in particular preconditioning and multigrid techniques are strongly recommended for this step of the computation, and many of these techniques are described in the literature on pressure correction methods.

Here, we suggest you to choose a simple line Gauss–Seidel method along a vertical line. As a first approximation of the pressure, its discrete values obtained on the previous time level are used. Given an approximation of the pressure p^k, the next one is obtained from the following relation:

$$\begin{aligned}
&\frac{1}{\Delta x_i}\left[\frac{p^{k}_{i+1,j}-p^{k+1}_{i,j}}{(\Delta x_{i+1}+\Delta x_i)/2}-\frac{p^{k+1}_{i,j}-p^{k+1}_{i-1,j}}{(\Delta x_i+\Delta x_{i-1})/2}\right]\\
&\quad+\frac{1}{\Delta y_j}\left[\frac{p^{k+1}_{i,j+1}-p^{k+1}_{i,j}}{(\Delta y_{j+1}+\Delta y_j)/2}-\frac{p^{k+1}_{i,j}-p^{k+1}_{i,j-1}}{(\Delta y_j+\Delta y_{j-1})/2}\right]=Q_{i,j}
\end{aligned}\tag{12.4.34}$$

where $Q_{i,j}$ is the right-hand side of (12.4.33). Not that the k index denotes an iteration number and not a time level (n in the previous section). Equation (12.4.34) can be

rewritten as follows:

$$a_{i,j}p_{i,j-1}^{k+1} + b_{i,j}p_{i,j}^{k+1} + c_{i,j}p_{i,j+1}^{k+1} = Q_{i,j} - \frac{1}{\Delta x_i}\left[\frac{p_{i+1,j}^{k}}{(\Delta x_{i+1} + \Delta x_i)/2} + \frac{p_{i-1,j}^{k+1}}{(\Delta x_i + \Delta x_{i-1})/2}\right] \tag{12.4.35}$$

where

$$\begin{aligned} a_{i,j} &= \frac{2}{\Delta y_j}\frac{1}{\Delta y_j + \Delta y_{j-1}} \\ c_{i,j} &= \frac{2}{\Delta y_j}\frac{1}{\Delta y_{j+1} + \Delta y_j} \\ b_{i,j} &= -\frac{2}{\Delta x_i}\left[\frac{1}{\Delta x_{i+1} + \Delta x_i} + \frac{1}{\Delta x_i + \Delta x_{i-1}}\right] - (a_{i,j} + c_{i,j}) \end{aligned} \tag{12.4.36}$$

We have to add the boundary conditions, for instance for $i = 1$ and a Neumann boundary condition (12.4.35) can be rewritten taking into account the boundary condition at $i = 1/2$:

$$a_{1,j}p_{1,j-1}^{k+1} + \left(b_{1,j} + \frac{1}{\Delta x_1}\frac{2}{\Delta x_1 + \Delta x_0}\right)p_{1,j}^{k+1} + c_{1,j}p_{1,j+1}^{k+1} = Q_{1,j} - \frac{1}{\Delta x_1}\frac{p_{2,j}^{k}}{(\Delta x_2 + \Delta x_1)/2} \tag{12.4.37}$$

where the Neumann boundary condition is expressed as $p_0^{k+1} = p_1^{k+1}$.

You can apply this similarly for the other boundary conditions. If the pressure is imposed at certain boundaries, as a Dirichlet condition, then you can introduce this value directly in the corresponding equation.

The algebraic system can be efficiently solved with the Thomas Algorithm (see Appendix A in Chapter 10).

The iterations are to be repeated until a prescribed convergence criterion is satisfied (e.g. $\max_{i,j} |p_{i,j}^{k+1} - p_{i,j}^{k}| < \varepsilon$).

For reasons of accuracy, it is recommended to alternate this algorithm with a line Gauss–Seidel method in the horizontal direction (at $j = \text{constant}$), applying the Thomas algorithm in two different mesh directions. Equation (12.4.34) is replaced by

$$\frac{1}{\Delta x_i}\left[\frac{p_{i+1,j}^{k+1} - p_{i,j}^{k+1}}{(\Delta x_{i+1} + \Delta x_i)/2} - \frac{p_{i,j}^{k+1} - p_{i-1,j}^{k+1}}{(\Delta x_i + \Delta x_{i-1})/2}\right] + \frac{1}{\Delta y_j}\left[\frac{p_{i,j+1}^{k} - p_{i,j}^{k+1}}{(\Delta y_{j+1} + \Delta y_j)/2} - \frac{p_{i,j}^{k+1} - p_{i,j-1}^{k+1}}{(\Delta y_j + \Delta y_{j-1})/2}\right] = Q_{i,j} \tag{12.4.38}$$

We leave it to you as an exercise to work out the details of its implementation.

12.5 NUMERICAL SOLUTIONS WITH THE PRESSURE CORRECTION METHOD

We apply now the developed method to the incompressible lid driven cavity flow. This well-known flow configuration results from the uniform motion of the upper wall of a square box, wherein the flow is induced by the viscous stresses, similarly to the Couette flow. The flow within the lid driven rectangular two-dimensional cavity is maintained by the continuous diffusion of kinetic energy injected by the moving wall. This energy is initially confined to a thin viscous layer of fluid next to the moving boundary. After a period of time, which depends on the Reynolds number, the redistribution of energy reaches an equilibrium leading to a steady state laminar flow. In case of high Reynolds number flows, this steady state solution is never reached, due to instabilities leading to transition to turbulence.

It can actually be considered as the two-dimensional extension of the Couette flow.

12.5.1 Lid Driven Cavity

We consider the domain included in a square of unit length, with $0 \leq x, y \leq 1$, where the upper boundary at $y = 1$, moves with a constant velocity $U = 1$.

The Reynolds number based on the size of the domain, the velocity of the moving wall, density $\rho = 1$ and viscosity $\mu = 0.01$ is $\mathrm{Re} = 100$.

The boundary conditions are set as follows

$$\begin{array}{ll} u(x,0) = 0 & v(x,0) = 0 \\ u(x,1) = 1 & v(x,1) = 0 \\ u(0,y) = 0 & v(0,y) = 0 \\ u(1,y) = 0 & v(1,y) = 0 \end{array} \tag{12.5.1}$$

The calculation is performed on a 41×41 Cartesian uniform mesh, which divides the computational domain in 1600 equidistant cells, with $\Delta x = \Delta y = 0.025$. In the solution of the pressure Poisson equation a Neumann boundary condition is imposed on the boundaries. The time step is taken as $\Delta t = 0.01$. A line Gauss–Seidel method is used to solve the pressure equation iteratively, iterations repeated till the maximum absolute value of the residual is smaller that 10^{-3}. In the selected time-dependent approach, the Poisson equation is converged for each time step. This option is selected to enable you also to handle unsteady flows or to detect spontaneous unsteadiness when they occur.

Figure 12.5.1 shows the convergence history of the pressure equation plotted for the first 10 time steps. As can be seen from the graph, the residual value for the first iteration of the Poisson solver at each time step decreases as the calculation proceeds, which means that less and less iterations are needed to reach the convergence criterion. This is typical for pressure correction methods when applied to unsteady flows with a steady state limit, at which the Poisson equation is satisfied automatically. Each jump in this figure represents the passage at the next time step, while the residual reduction in between represents the convergence behavior of the Gauss–Seidel relaxation method.

The diagrams on Figure 12.5.2 display the streamlines of the solution at different transient stages, at $t = 0.5$, 1, 10 and 30. The flow undergoes a recirculation motion

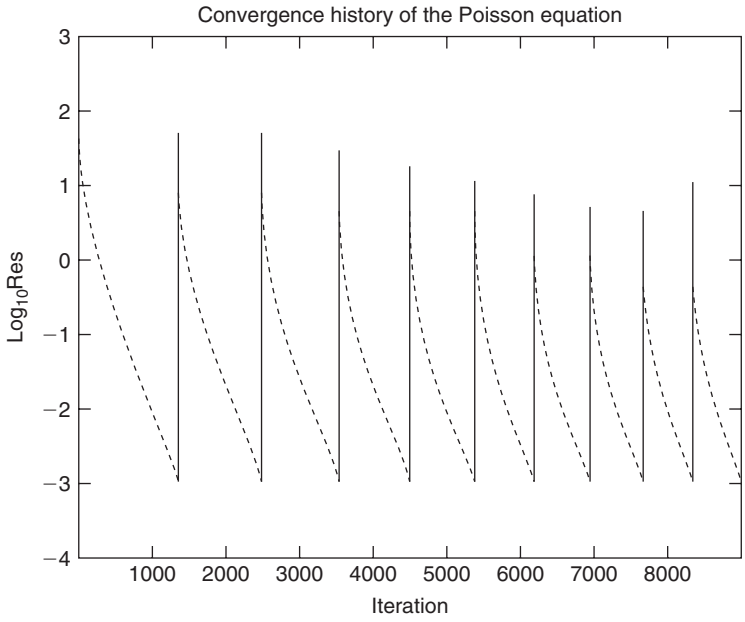

Figure 12.5.1 *Convergence history of the Poisson equation for the pressure.*

imposed by the viscous effects. The solution at $t = 0.5$ and $t = 1$ represent a transient stage of the solution, where the circular motion of the fluid is still developing. At $t = 10$ and $t = 30$ the flow in the cavity is fully developed and has reached its steady state.

On a quantitative basis, Figure 12.5.3 compares the velocity distributions along the centerlines $x = 0.5$ and $y = 0.5$, with a reference solution obtained by Dr. Sergey Smirnov, at the Vrije Universiteit Brussel, Belgium, on a fine grid of 161×161 with a fourth order accurate compact scheme for the space discretization.

The maximum error on the velocity distribution is of the order of 10% on this 41×41 mesh, indicating that a finer resolution is required.

12.5.2 Additional Suggestions

You can now run your code on many other cases, such as:

- The lid driven cavity by increasing the Reynolds number of your simulation, until you start detecting the initial process toward transition. This will require you to increase the grid resolution.
- The flat plate problem.
- Other cases with a Cartesian grid, such as the backward facing step.
- You could also extend now your code to more general grids and run the cylinder case in laminar mode.

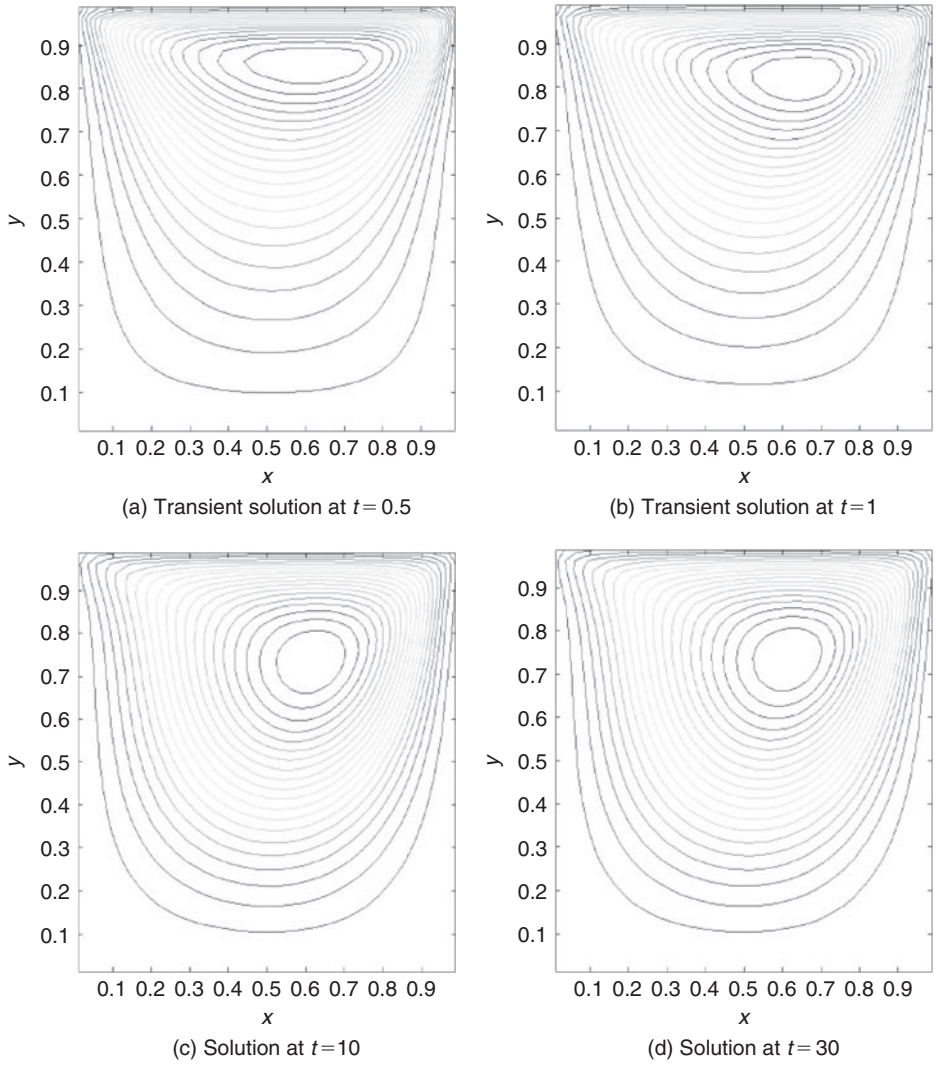

Figure 12.5.2 *Streamlines of the flow field at different transient stages, for t = 0.5, 1, 10 and 30 (for color image refer Plate 12.5.2).*

12.6 BEST PRACTICE ADVICE

CFD software systems form today an essential part of the world of Computer Aided Engineering (CAE), supporting the design and analysis of industrial products involving fluid flows. Many design decisions of systems, whose performance depends on their internal or external flow behavior, are based on the results of CFD simulations, either with in-house or commercial CFD codes. This raises the question of

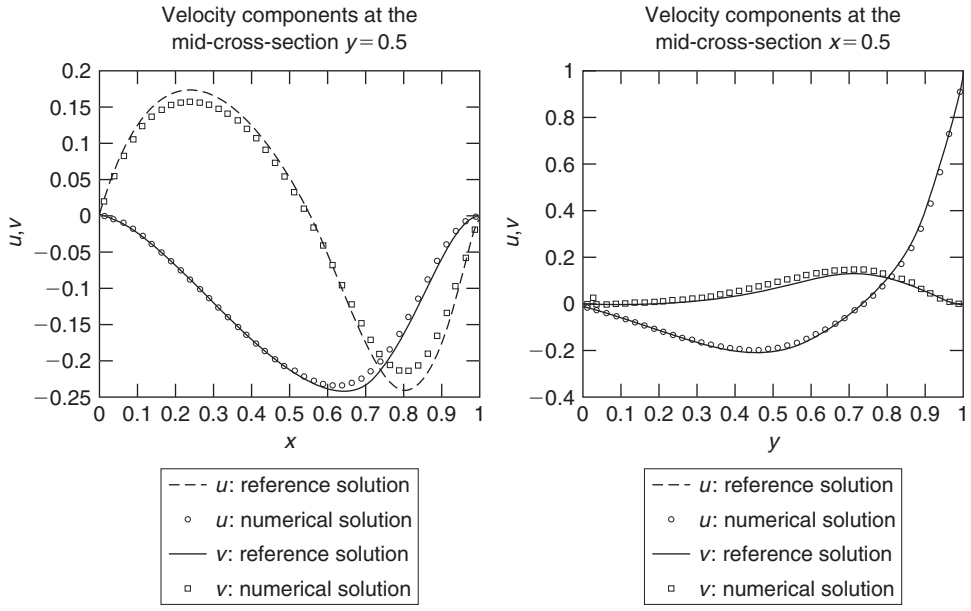

Figure 12.5.3 *Comparison of the computed velocity distributions along the centerlines $x = 0.5$ and $y = 0.5$, with a reference solution.*

the reliability and the confidence we can attach to the results of CFD, in presence of numerous sources of errors and of uncertainties.

The overwhelming majority of industrial and environmental fluid flow systems are turbulent and the modeling of turbulence remains a dominant factor of uncertainty, as none of the available models today are fully satisfactory in their prediction capability of the complex phenomena of turbulence.

We have not dealt with turbulence, in this introductory text on the basics of CFD, and left this important issue to the Volume II, although it has a considerable effect on the level of uncertainty of CFD results.

In order to respond to the needs of the increasing number of CFD users in industry, a demand has arisen for recommendations of best practices in the application of CFD codes, in regard of the complexity of industrial flow systems.

One of the first efforts toward the establishment of best practice guidelines (BPG) for CFD has been generated by the ERCOFTAC (European Research Community on Flow, Turbulence And Combustion) association; http://www.ercoftac.org. This effort has led to a document, Casey and Wintergerste (2000), providing an extensive set of recommendations, and is available from this organization.

As we have considered here only the numerical issues and since we attempted to develop your awareness of the various pitfalls and error sources, particularly in the last two chapters, we will summarize here some of the basic guidelines and recommendations in applying CFD codes, based essentially on the content of this ERCOFTAC document.

12.6.1 List of Possible Error Sources

The first issue is to attempt to summarize all the possible error sources. We can distribute them as follows:

- ***Discretization or numerical error***: These errors are due to the approximations resulting from the space and time numerical discretization, and have been analyzed in details in Parts III and IV of this book.
- ***Iteration or convergence error***: These errors occur due to the difference between a fully converged solution on a finite number of grid points and a solution that is not fully converged. Ideally, each calculation should be run up to the reduction of the residuals to machine accuracy. Although this can be performed on simple cases as illustrated in the last two chapters, it is hardly ever possible on industrial simulations with million of points.
- ***Round-off errors***: These errors are due to the fact that the difference between two values of a parameter is below the machine accuracy of the computer. This is caused by the limited number of computer digits available for storage of a given physical value. It might require to shift to double precision arithmetic, when dealing with very small variations between flow variables, or when very short distances between mesh points are introduced.
- ***Application uncertainties***: Many variables defining the flow conditions, such as operational and/or geometrical data are often not precisely defined or not well known. Examples of this are uncertainties in the precise geometry due to manufacturing tolerances, uncertain inflow data or models, such as turbulence properties or fluid properties.
- ***User errors***: Errors can arise from mistakes introduced by the user. They can cover various aspects, such as inadequate or poor grid generation; incorrect boundary condition; incorrect choice of numerical parameters, such as time step or relaxation coefficients; post-processing errors. Experience and great care are required to minimize their risk of occurrence.
- ***Code errors***: Errors due to bugs in the software cannot be excluded, despite all verification efforts, as it is humanly impossible to cover all possible combinations of code parameters with a finite number of verification tests.
- ***Model uncertainties***: It refers to the physical models that have to be introduced to describe complex flow properties, such as turbulence, multiphase flows, combustion, real Newtonian or non-Newtonian fluids.

To minimize the effects of these error sources, a series of recommendations can be collected, that we recommend to your attention, as a kind of checklist when running a CFD code.

12.6.2 Best Practice Recommendations

As seen in the previous chapters, an essential component of a CFD simulation and a major potential source of errors is the choice of the grid and the resulting grid quality.

RECOMMENDATIONS ON GRIDS

The key recommendation is to ensure smooth grids, avoiding abrupt changes in grid size or shape, as this can lead to a significant loss of accuracy. Hence take good care to:

- Define the computational domain, in order to minimize the influence and interactions between the flow and the far-field conditions. In particular,
  - Place inlet and outlet boundaries as far away as possible from the region of interest. In particular, if uniform far-field conditions are imposed, you should ensure that the boundary is not in a region where the flow may still vary significantly.
  - Avoid inlet or outlet boundaries in regions of strong geometrical changes or in regions of recirculation.
- Avoid jumps in grid density or in grid size.
- Avoid highly distorted cells or small grid angles.
- Ensure that the grid stretching is continuous.
- Avoid unstructured tetrahedral meshes in boundary layer regions.
- Refine the grids in regions with high gradients, such as boundary layers, leading edges of airfoils and any region where large changes in flow properties might occur.
- Make sure that the number of points in the boundary layers is sufficient for the expected accuracy. Avoid less than 10 points over the inner part of the boundary layer thickness.
- Monitor the grid quality by adequate mesh parameters, available in most of the grid generators, such as aspect ratio, internal angle, concavity, skewness, negative volume.

RECOMMENDATIONS ON SOLUTION ASSESSMENT

Once you run your code, the following recommendations will be useful to enhance your confidence in the results obtained:

- Check very carefully the selected boundary conditions for correctness and compatibility with the physics of the flow you are modeling.
- Verify all the numerical settings and parameters, before launching the CFD run.
- Verify that your initial solution is acceptable for the problem to be solved.
- Monitor the convergence to ensure that you reach machine accuracy. It is recommended to monitor, in addition to the residuals, the convergence of representative quantities of your problem, such as a drag force or coefficient, a velocity, temperature or pressure at selected points in the flow domain.
- Look carefully at the behavior of the residual convergence curve in function of number of iterations. If the behavior is oscillatory, or if the residual does not converge to machine accuracy by showing a limit cycle at a certain level of residual reduction, it tells you that some inaccuracy affects your solution process.

- Apply internal consistency and accuracy criteria, by verifying:
  - Conservation of global quantities such as total enthalpy and mass flow in steady flow calculations.
  - The entropy production and drag coefficients with inviscid flows, which are strong indicators of the influence of numerical dissipation, as they should be zero.
- Check, whenever possible, the grid dependence of the solution by comparing the results obtained on different grid sizes.
- Some quantities are more sensitive than others to error sources. Pressure curves are less sensitive than shear stresses, which in turn are less sensitive than temperature gradients or heat fluxes, which require finer grids for a given accuracy level.
- If your calculation appears difficult to converge, you can
  - Look at the residual distribution and associated flow field for possible hints, e.g. regions with large residuals or unrealistic levels of the relevant flow parameters.
  - Reduce the values of parameters controlling convergence, such as the CFL number or some under-relaxation parameter, when available.
  - Consider the effects of different initial flow conditions.
  - Check the effect of the grid quality on the convergence rate.
  - Use a more robust numerical scheme, such as a first order scheme, during the initial steps of the convergence and switch to more accurate numerical schemes as the convergence improves.

RECOMMENDATIONS ON EVALUATION OF UNCERTAINTIES

This is a very difficult issue, as the application uncertainties are generally not well defined and require a sound judgment about the physics of the considered flow problem. Some recommendations can be offered:

- Attempt to list the most important uncertainties, such as
  - Geometrical simplifications and manufacturing tolerances around the CAD definition.
  - Operational conditions, such as inlet velocity or inlet flow angle.
  - Physical approximations, such as handling an incompressible flow as a low Mach number compressible flow. This type of uncertainty is manageable, as it can more easily be quantified.
  - Uncertainties related to turbulence or other physical models.
- Perform a sensitivity analysis of the relevant uncertainty to investigate its influence.

CONCLUSIONS AND MAIN TOPICS TO REMEMBER

If you have followed closely the guidelines of this chapter, you have now available a general 2D finite volume density-based code, which allows you to handle practically

any flow configuration, from low to supersonic speed. You can even simulate incompressible flow conditions, by considering low Mach numbers, say below 0.2, for which the numerical solution is an excellent approximation of incompressible fluid flows.

You also have available another option, with a code based on the pressure correction method, suitable for compressible and incompressible flows, although it is restricted to the subsonic range.

You have certainly experienced, by following the steps of the last two chapters in running the various proposed test cases, that the way to achieve high accuracy and reliability of the CFD results on general grids is a difficult process, requiring a close attention to all the details of the implementation of a selected scheme.

Our main ambition with these two chapters was to introduce you to this awareness and to guide you in your ability to ask the 'right questions' when faced with the development of a CFD code or when using a third party code.

The main topics to remember are summarized in the best practice guidelines of Section 12.6. The main message being that you have to exercise critical judgment at all stages of the code development. If you apply a third party code, your critical judgment should apply to your assessment of all aspects of the schemes and its implementation as proposed by the options you select. Make sure that you have enough information on:

- Formal order of the scheme, but also on its behavior on a non-uniform grid.
- The level of numerical dissipation generated on your grid. This can be obtained by running the same case as an inviscid problem, monitoring the entropy distribution.
- The details of the boundary condition implementation and their effect on the accuracy and convergence.
- Convergence levels of the solution, in terms of residuals, but also by monitoring some of the quantities relevant for the problem you are interested in.

We also hope that these exercises will have stimulated your interest and enthusiasm for the beautiful world of numerical flow simulations.

REFERENCES

Anderson, J.D. (1995). *Computational Fluid Dynamics. The Basics with Applications*. McGraw-Hill, New York.

Casey, M. and Wintergerste, T. (2000). Best Practice Guidelines. *ERCOFTAC Special Interest Group on Quality and Trust in Industrial CFD*. ERCOFTAC, http://www.ercoftac.org.

Chorin, A.J. (1967). A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys*., 2, 12–26.

Chorin, A.J. (1968). Numerical solution of the Navier–Stokes equations. *Math. comput*., 23, 341–54.

Ferziger, J.H. and Peric, M. (1997). *Computational Methods for Fluid Dynamics*. Springer Verlag, Berlin.

Harlow, F.H. and Welch, J.E. (1965). Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluid*., 8, 2182–2189.

Hemsch, M.J. and Morrison, J.H. (2004). Statistical analysis of CFD solutions from 2nd drag prediction workshop. *42nd AIAA Aerospace Sciences Meeting*, Reno, AIAA Paper 2004-556.

Jameson, A. (1995a). Analysis and design of numerical schemes for gas dynamics. 1. Artificial diffusion, upwind biasing, limiters and their effect on multigrid convergence. *Int. J. Comp. Fluid Dyn*., 4, 171–218.

Jameson, A. (1995b). Analysis and design of numerical schemes for gas dynamics. 2. Artificial diffusion and discrete shock structure. *Int. J. Comp. Fluid Dyn*., 5, 1–38.

Majda, A. and Sethian, J. (1985). The derivation and numerical solution of the equations for zero Mach number combustion. *Combust. Sci. Technol*., 42, 185.

Patankar, S.V. (1980). *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publ. Co., New York.

Patankar, S.V. and Spalding, D.B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer*, 15, 1787–1806.

Schlichting, H. (1979). *Boundary Layer Theory*. McGraw-Hill, New York.

Temam, R. (1969). Sur l'approximation de la solution de equations de Navier–Stokes par la methode des pas fractionnnaires. *Arch. Rational Mech. Anal*., 32, 135–153; (II): *Arch. Rational Mech. Anal*., 33, 377–385.

Temam, R. (1977). *Navier–Stokes Equations*. North-Holland, Amsterdam.

Wesseling, P. (2001). *Principles of Computational Fluid Dynamics*. Springer Verlag, Berlin.