

$$[\Delta u]_{i,j} := \frac{u(x_{i+1,j}) - 2u(x_{i,j}) + u(x_{i-1,j})}{\delta x^2} + \frac{u(x_{i,j+1}) - 2u(x_{i,j}) + u(x_{i,j-1})}{\delta y^2}. \quad (3.13)$$

The discretization of Poisson's equation with Dirichlet boundary conditions

$$\Delta u = f \quad \text{in } \Omega, \quad u = g \quad \text{on } \partial\Omega \quad (3.14)$$

then leads to the linear system of equations

$$\frac{1}{\delta x^2} (u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{\delta y^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = f(x_{i,j}), \quad (3.15)$$

$$i = 1, \dots, i_{\max} - 1, \quad j = 1, \dots, j_{\max} - 1,$$

of dimension $(i_{\max} - 1)(j_{\max} - 1)$ with unknowns $u_{i,j}$ approximating the solution at the grid points $x_{i,j}$. To satisfy the boundary condition we set $u_{i,j} = g(x_{i,j})$ for $i \in \{0, i_{\max}\}$ or $j \in \{0, j_{\max}\}$.

If the differential equation also contains derivatives of first order, then these can lead to stability problems like those we encountered in the one-dimensional case. These can again be circumvented by using upwind or donor cell discretizations.

3.1.2 Discretization of the Navier–Stokes Equations

Treatment of the Spatial Derivatives

When solving the Navier–Stokes equations, the region Ω is often discretized using a *staggered grid*, in which the different unknown variables are not located at the same grid points. In the grid we shall use, the pressure p is located in the cell centers, the horizontal velocity u in the midpoints of the vertical cell edges, and the vertical velocity v in the midpoints of the horizontal cell edges.¹⁷ Cell (i, j) occupies the spatial region $[(i-1)\delta x, i\delta x] \times [(j-1)\delta y, j\delta y]$, and the corresponding index (i, j) is assigned to the pressure at the cell center as well as to the u -velocity at the right edge and the v -velocity at the upper edge of this cell. Hence the pressure value $p_{i,j}$ is located at the coordinates $((i - 0.5)\delta x, (j - 0.5)\delta y)$, the horizontal velocity value $u_{i,j}$ at the coordinates $(i\delta x, (j - 0.5)\delta y)$, and the vertical velocity value $v_{i,j}$ at the coordinates $((i - 0.5)\delta x, j\delta y)$ (see Figure 3.4).

As a result, the discrete values of u , v , and p are actually located on three separate grids not shown here, each shifted by half a grid spacing to the bottom, to the left, and to the lower left, respectively.

Consequently, not all extremal grid points come to lie on the domain boundary. The vertical boundaries, for instance, carry no v -values, just as the horizontal boundaries carry no u -values. For this reason, an extra boundary strip of grid cells is introduced (see Figure 3.5), so that the boundary conditions may be applied by averaging the nearest grid points on either side (see the subsection "Boundary Values for the Discrete Equations" on page 30).

This staggered lations which co and p at the san conditions impos The solution of t

Were we to disc using central dif

$$u_{i,j} = 0, \quad v_i$$

would solve the

One alternati (cf. [Perić et al. and a finite volu determine the fi

¹⁷This phenomen condition [Brezzi, 1 problems, is violate

¹⁷The idea for this staggered arrangement of the variables comes from the finite volume method, in which the continuity equation is discretized in each volume cell by considering the mass flux across the cell edges determined by the velocities on these edges.

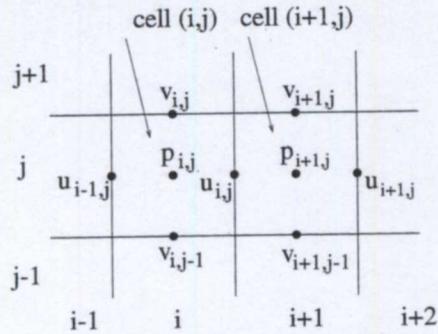


FIG. 3.4. Staggered grid.

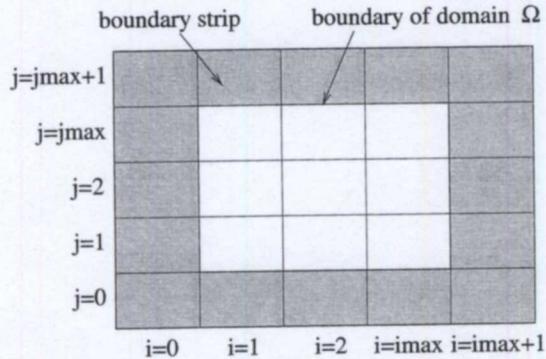


FIG. 3.5. Domain with boundary cells.

This staggered arrangement of the unknowns prevents possible pressure oscillations which could occur had we evaluated all three unknown functions u , v , and p at the same grid points. Consider briefly the problem with zero Dirichlet conditions imposed for both u and v along the entire boundary and $g_x = g_y = 0$. The solution of the continuous problem in this case is

$$u = 0, \quad v = 0, \quad p = \text{const.}$$

Were we to discretize the pressure derivative term in the momentum equations using central differences on a nonstaggered grid, then

$$u_{i,j} = 0, \quad v_{i,j} = 0, \quad p_{i,j} = P_1 \text{ for } i+j \text{ even}, \quad p_{i,j} = P_2 \text{ for } i+j \text{ odd}$$

would solve the discrete problem for arbitrary values of P_1 and P_2 .¹⁸

One alternative to a staggered grid is the use of so-called *colocated grids* (cf. [Perić et al., 1988]), for which u , v , and p are all evaluated at cell centers and a finite volume-based discretization employs special interpolation schemes to determine the flux across the cell edges.

¹⁸This phenomenon also occurs in finite element discretizations whenever the Babuška–Brezzi condition [Brezzi, 1974], which insures uniqueness and stability of the solution of saddlepoint problems, is violated.

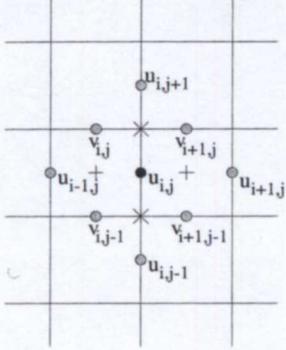


FIG. 3.6. Values required for the discretization of the u -momentum equation.

We return to the discretization on the staggered grid. The continuity equation (2.2c) is discretized at the center of each cell (i, j) , $i = 1, \dots, i_{\max}$, $j = 1, \dots, j_{\max}$, by replacing the spatial derivatives $\partial u / \partial x$ and $\partial v / \partial y$ by centered differences using half the mesh width:

$$\left[\frac{\partial u}{\partial x} \right]_{i,j} := \frac{u_{i,j} - u_{i-1,j}}{\delta x}, \quad \left[\frac{\partial v}{\partial y} \right]_{i,j} := \frac{v_{i,j} - v_{i,j-1}}{\delta y}. \quad (3.16)$$

The momentum equation (2.2a) for u , on the other hand, is discretized at the midpoints of the vertical cell edges, and the momentum equation (2.2b) for v at the midpoints of the horizontal cell edges.

The second derivatives $\partial^2 u / \partial x^2$, $\partial^2 u / \partial y^2$, $\partial^2 v / \partial x^2$, and $\partial^2 v / \partial y^2$ forming the so-called diffusive terms, in turn, can be replaced by their discrete counterparts according to (3.5) while the spatial derivatives of pressure are again treated using central differences with half the mesh width.

The discretization of the convective terms $\partial(u^2) / \partial x$, $\partial(uv) / \partial y$, $\partial(uv) / \partial x$, and $\partial(v^2) / \partial y$, however, poses some difficulties. For example, to discretize $\partial(uv) / \partial y$ at the midpoint of the right edge of cell (i, j) (black dot in Figure 3.6), we need suitable values of the product uv lying in the two vertical directions. One solution that suggests itself here is to use averages of u and v taken at the locations marked with an \times in Figure 3.6, which gives us the discrete term

$$\begin{aligned} \left[\frac{\partial(uv)}{\partial y} \right]_{i,j} &:= \frac{1}{\delta y} \left(\frac{(v_{i,j} + v_{i+1,j})}{2} \frac{(u_{i,j} + u_{i,j+1})}{2} \right. \\ &\quad \left. - \frac{(v_{i,j-1} + v_{i+1,j-1})}{2} \frac{(u_{i,j-1} + u_{i,j})}{2} \right). \end{aligned} \quad (3.17)$$

Similarly, to discretize $\partial(u^2) / \partial x$, we use a central difference with half the mesh width of values averaged at the points marked with a $+$ in Figure 3.6, rather than employing a central difference between $u_{i+1,j}$ and $u_{i-1,j}$:

$$\left[\frac{\partial(u^2)}{\partial x} \right]_{i,j} := \frac{1}{\delta x} \left(\left(\frac{u_{i,j} + u_{i+1,j}}{2} \right)^2 - \left(\frac{u_{i-1,j} + u_{i,j}}{2} \right)^2 \right). \quad (3.18)$$

The Discretization

Because the convective terms involve high Reynolds numbers or high Froude numbers, the central differences described in section 3.12, we set

$$k_r := \frac{u_{i,j} + u_{i+1,j}}{2}$$

in the donor-cell term to discrete

$$k_r := \frac{v_{i,j} + v_{i,j+1}}{2}$$

for the discretization of $\partial(uv) / \partial x$. The convective terms of this type are treated analogously to the discretizations of the advection terms (cf. [Hirt et al., 1975]).

In equation (2.2a) for u at $i = 1, \dots, i_{\max} - 1$, $j = 1, \dots, j_{\max}$

$$\begin{aligned} \left[\frac{\partial(u^2)}{\partial x} \right]_{i,j} &:= \frac{1}{\delta x} \left(\left(\frac{u_{i,j} + u_{i+1,j}}{2} \right)^2 \right. \\ &\quad \left. + \gamma \frac{1}{\delta x} \left(\frac{|u_{i,j} + u_{i+1,j}|}{2} \right)^2 \right) \end{aligned}$$

$$\begin{aligned} \left[\frac{\partial(uv)}{\partial y} \right]_{i,j} &:= \frac{1}{\delta y} \left(\left(\frac{v_{i,j} + v_{i+1,j}}{2} \right)^2 \right. \\ &\quad \left. + \gamma \frac{1}{\delta y} \left(\frac{|v_{i,j} + v_{i+1,j}|}{2} \right)^2 \right) \end{aligned}$$

$$\left[\frac{\partial^2 u}{\partial x^2} \right]_{i,j} := \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2}$$

$$\left[\frac{\partial^2 u}{\partial y^2} \right]_{i,j} := \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\delta y)^2}$$

and in equation (2.2b) for v at $i = 1, \dots, i_{\max}$, $j = 1, \dots, j_{\max}$

$$\begin{aligned} \left[\frac{\partial(uv)}{\partial x} \right]_{i,j} &:= \frac{1}{\delta x} \left(\left(\frac{u_{i,j} + u_{i+1,j}}{2} \right)^2 \right. \\ &\quad \left. + \gamma \frac{1}{\delta x} \left(\frac{|u_{i,j} + u_{i+1,j}|}{2} \right)^2 \right) \end{aligned}$$

$$\begin{aligned} \left[\frac{\partial(v^2)}{\partial y} \right]_{i,j} &:= \frac{1}{\delta y} \left(\left(\frac{v_{i,j} + v_{i+1,j}}{2} \right)^2 \right. \\ &\quad \left. + \gamma \frac{1}{\delta y} \left(\frac{|v_{i,j} + v_{i+1,j}|}{2} \right)^2 \right) \end{aligned}$$

¹⁹Due to the identity $\partial(u^2) / \partial x = \partial u / \partial x$ in the convective terms.

The Discretization

Because the convective terms in the momentum equations become dominant at high Reynolds numbers or high velocities,¹⁹ it is necessary to use a mixture of the central differences described above and the donor-cell discretization. Following (3.12), we set

$$k_r := \frac{u_{i,j} + u_{i+1,j}}{2}, \quad k_l := \frac{u_{i-1,j} + u_{i,j}}{2}$$

in the donor-cell term to discretize $\partial(u^2)/\partial x$ as well as

$$k_r := \frac{v_{i,j} + v_{i+1,j}}{2}, \quad k_l := \frac{v_{i,j-1} + v_{i+1,j-1}}{2}$$

for the discretization of $\partial(uv)/\partial y$. The remaining two terms $\partial(uv)/\partial x$ and $\partial(v^2)/\partial y$ of this type are treated analogously. In sum, we obtain the following discrete expressions (cf. [Hirt et al., 1975]).

In equation (2.2a) for u at the midpoint of the *right edge* of cell (i, j) , $i = 1, \dots, i_{\max}, j = 1, \dots, j_{\max}$, we set

$$\begin{aligned} \left[\frac{\partial(u^2)}{\partial x} \right]_{i,j} &:= \frac{1}{\delta x} \left(\left(\frac{u_{i,j} + u_{i+1,j}}{2} \right)^2 - \left(\frac{u_{i-1,j} + u_{i,j}}{2} \right)^2 \right) \\ &\quad + \gamma \frac{1}{\delta x} \left(\frac{|u_{i,j} + u_{i+1,j}|}{2} \frac{(u_{i,j} - u_{i+1,j})}{2} - \frac{|u_{i-1,j} + u_{i,j}|}{2} \frac{(u_{i-1,j} - u_{i,j})}{2} \right), \\ \left[\frac{\partial(uv)}{\partial y} \right]_{i,j} &:= \frac{1}{\delta y} \left(\frac{(v_{i,j} + v_{i+1,j})(u_{i,j} + u_{i,j+1})}{2} - \frac{(v_{i,j-1} + v_{i+1,j-1})(u_{i,j-1} + u_{i,j})}{2} \right) \\ &\quad + \gamma \frac{1}{\delta y} \left(\frac{|v_{i,j} + v_{i+1,j}|}{2} \frac{(u_{i,j} - u_{i,j+1})}{2} - \frac{|v_{i,j-1} + v_{i+1,j-1}|}{2} \frac{(u_{i,j-1} - u_{i,j})}{2} \right), \\ \left[\frac{\partial^2 u}{\partial x^2} \right]_{i,j} &:= \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\delta x)^2}, \\ \left[\frac{\partial^2 u}{\partial y^2} \right]_{i,j} &:= \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\delta y)^2}, \quad \left[\frac{\partial p}{\partial x} \right]_{i,j} := \frac{p_{i+1,j} - p_{i,j}}{\delta x}, \end{aligned} \tag{3.19a}$$

and in equation (2.2b) for v at the midpoint of the *upper edge* of cell (i, j) , $i = 1, \dots, i_{\max}, j = 1, \dots, j_{\max} - 1$, we set

$$\begin{aligned} \left[\frac{\partial(uv)}{\partial x} \right]_{i,j} &:= \frac{1}{\delta x} \left(\frac{(u_{i,j} + u_{i,j+1})(v_{i,j} + v_{i,j+1})}{2} - \frac{(u_{i-1,j} + u_{i-1,j+1})(v_{i-1,j} + v_{i,j})}{2} \right) \\ &\quad + \gamma \frac{1}{\delta x} \left(\frac{|u_{i,j} + u_{i,j+1}|}{2} \frac{(v_{i,j} - v_{i,j+1})}{2} - \frac{|u_{i-1,j} + u_{i-1,j+1}|}{2} \frac{(v_{i-1,j} - v_{i,j})}{2} \right), \\ \left[\frac{\partial(v^2)}{\partial y} \right]_{i,j} &:= \frac{1}{\delta y} \left(\left(\frac{v_{i,j} + v_{i,j+1}}{2} \right)^2 - \left(\frac{v_{i,j-1} + v_{i,j}}{2} \right)^2 \right) \\ &\quad + \gamma \frac{1}{\delta y} \left(\frac{|v_{i,j} + v_{i,j+1}|}{2} \frac{(v_{i,j} - v_{i,j+1})}{2} - \frac{|v_{i,j-1} + v_{i,j}|}{2} \frac{(v_{i,j-1} - v_{i,j})}{2} \right), \end{aligned}$$

¹⁹Due to the identity $\partial(u^2)/\partial x = 2u \partial u / \partial x$, u enters as a factor in front of the first derivative $\partial u / \partial x$ in the convective terms.

for the discretization of the continuity equation.

d. The continuity equation $1, \dots, i_{\max}, j = 1, \dots, j_{\max}$, $\nabla \cdot \mathbf{v}$ centered differences using

$$\frac{v_{i,j} - v_{i,j-1}}{\delta y}. \tag{3.16}$$

On the hand, is discretized at the um equation (2.2b) for v at

$\nabla^2 v$, and $\partial^2 v / \partial y^2$ forming the their discrete counterparts sure are again treated using

$\nabla \cdot \mathbf{v}$, $\partial(uv)/\partial y$, $\partial(uv)/\partial x$, and nple, to discretize $\partial(uv)/\partial y$ dot in Figure 3.6), we need tical directions. One solution aken at the locations marked term

$$\frac{u_{i,j+1}}{2} \tag{3.17}$$

$$\frac{u_{i,j-1} + u_{i,j}}{2}.$$

ifference with half the mesh $i + 1$ in Figure 3.6, rather than $i-1,j$:

$$\frac{-1,j + u_{i,j}}{2} \Bigg)^2 \Bigg). \tag{3.18}$$

$$\begin{aligned} \left[\frac{\partial^2 v}{\partial x^2} \right]_{i,j} &:= \frac{v_{i+1,j} - 2v_{i,j} + v_{i-1,j}}{(\delta x)^2}, \\ \left[\frac{\partial^2 v}{\partial y^2} \right]_{i,j} &:= \frac{v_{i,j+1} - 2v_{i,j} + v_{i,j-1}}{(\delta y)^2}, \quad \left[\frac{\partial p}{\partial y} \right]_{i,j} := \frac{p_{i,j+1} - p_{i,j}}{\delta y}. \end{aligned} \quad (3.19b)$$

The parameter γ in the above formulas lies between 0 and 1. For $\gamma = 0$ we recover the central difference discretization, and for $\gamma = 1$, a pure donor-cell scheme results. According to [Hirt et al., 1975], γ should be chosen such that

$$\gamma \geq \max_{i,j} \left(\left| \frac{u_{i,j} \delta t}{\delta x} \right|, \left| \frac{v_{i,j} \delta t}{\delta y} \right| \right) \quad (3.20)$$

is satisfied.

Boundary Values for the Discrete Equations

The discretization (3.19a) of the momentum equation (2.2a) for u involves u -values on the boundary for $i \in \{1, i_{\max} - 1\}$. Moreover, for $j \in \{1, j_{\max}\}$, v -values lying on the boundary are required as well as additional u -values lying outside the domain Ω . Similarly, boundary values of v are required in the discretization (3.19b) of the momentum equation (2.2b) for v . In total, we require the values

$$\begin{aligned} u_{0,j}, \quad u_{i_{\max},j}, \quad &j = 1, \dots, j_{\max}, \\ v_{i,0}, \quad v_{i,j_{\max}}, \quad &i = 1, \dots, i_{\max}, \end{aligned}$$

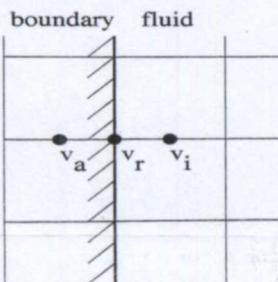
on the boundary as well as the values

$$\begin{aligned} u_{i,0}, \quad u_{i,j_{\max}+1}, \quad &i = 1, \dots, i_{\max}, \\ v_{0,j}, \quad v_{i_{\max}+1,j}, \quad &j = 1, \dots, j_{\max}, \end{aligned}$$

outside the domain Ω . These velocity values are obtained from a discretization of the boundary conditions of the continuous problem.

1. *No-slip condition:* The continuous velocities should vanish at the boundary to satisfy the no-slip condition.²⁰ For the values lying directly on the boundary we thus set

$$\begin{aligned} u_{0,j} = 0, \quad u_{i_{\max},j} = 0, \quad &j = 1, \dots, j_{\max}, \\ v_{i,0} = 0, \quad v_{i,j_{\max}} = 0, \quad &i = 1, \dots, i_{\max}. \end{aligned} \quad (3.21)$$



Since the vertical boundaries contain no v -values and the horizontal boundaries contain no u -values, the zero boundary value is enforced in these cases by averaging the values on either side of the boundary:

$$v_r := \frac{v_a + v_i}{2} = 0 \quad \Rightarrow \quad v_a = -v_i. \quad (3.22)$$

²⁰In the case of a moving boundary, the velocities are set to nonzero values, namely, those of the wall velocities. This occurs, for example, in the driven cavity flow in Section 5.1.

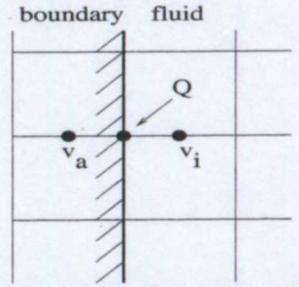
The Discretization

On the four boundaries, we

$$\begin{aligned} v_{0,j} &= -v_{1,j}, & v_{i_{\max},j} &= v_{i-1,j}, \\ u_{i,0} &= -u_{i,1}, & u_{i,j_{\max}} &= u_{i,j-1}. \end{aligned}$$

2. *Free-slip condition:* In case component normal to the derivative of the velocity \mathbf{c} angular domain discretized normal to the boundary lie no-slip condition, we may s

$$\begin{aligned} u_{0,j} &= 0, \\ v_{i,0} &= 0, \end{aligned}$$



We thus obtain the further

$$\begin{aligned} v_{0,j} &= v_{1,j}, \\ u_{i,0} &= u_{i,1}, \end{aligned}$$

3. *Outflow conditions:* In the c of both velocity components the total velocity does not In the discrete case this c boundary equal to their ne

$$\begin{aligned} u_{0,j} &= u_{1,j}, & u_{i_{\max},j} &= u_{i-1,j}, \\ v_{0,j} &= v_{1,j}, & v_{i_{\max},j} &= v_{i-1,j}, \end{aligned}$$

$$\begin{aligned} u_{i,0} &= u_{i,1}, & u_{i,j_{\max}} &= u_{i,j-1}, \\ v_{i,0} &= v_{i,1}, & v_{i,j_{\max}} &= v_{i,j-1}, \end{aligned}$$

4. *Inflow conditions:* On an i we impose this for the velo boundary) by directly fixin; components tangential to achieve this by averaging t as in (3.22).

(3.19b)

$$:= \frac{p_{i,j+1} - p_{i,j}}{\delta y}$$

en 0 and 1. For $\gamma = 0$ we or $\gamma = 1$, a pure donor-cell ould be chosen such that

(3.20)

sion (2.2a) for u involves u -er, for $j \in \{1, j_{\max}\}$, v -values tional u -values lying outside required in the discretization total, we require the values

$$, j_{\max}, \\ , i_{\max},$$

$$, i_{\max}, \\ , j_{\max},$$

btained from a discretization em.

uld vanish at the boundary to ing directly on the boundary

$$= 1, \dots, j_{\max}, \\ = 1, \dots, i_{\max}. \quad (3.21)$$

boundaries contain no v -values al boundaries contain no u -boundary value is enforced in aging the values on either side

$$= 0 \Rightarrow v_a = -v_i. \quad (3.22)$$

to nonzero values, namely, those of cavity flow in Section 5.1.

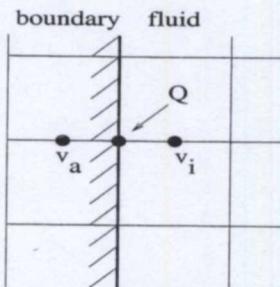
The Discretization

On the four boundaries, we thus obtain the conditions

$$\begin{aligned} v_{0,j} &= -v_{1,j}, & v_{i_{\max}+1,j} &= -v_{i_{\max},j}, & j &= 1, \dots, j_{\max}, \\ u_{i,0} &= -u_{i,1}, & u_{i,j_{\max}+1} &= -u_{i,j_{\max}}, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.23)$$

2. *Free-slip condition:* In case of a free-slip boundary condition, the velocity component normal to the boundary should vanish along with the normal derivative of the velocity component tangent to the boundary. In our rectangular domain discretized using the staggered grid, the values of velocities normal to the boundary lie directly on the boundary; hence, just as for the no-slip condition, we may set

$$\begin{aligned} u_{0,j} &= 0, & u_{i_{\max},j} &= 0, & j &= 1, \dots, j_{\max}, \\ v_{i,0} &= 0, & v_{i,j_{\max}} &= 0, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.24)$$



The normal derivative $\partial v / \partial n$ of the tangential velocity at a boundary point Q may be discretized by the expression $(v_i - v_a) / \delta x$; hence the requirement $\partial v / \partial n = 0$ leads to the condition

$$v_a = v_i.$$

We thus obtain the further boundary conditions

$$\begin{aligned} v_{0,j} &= v_{1,j}, & v_{i_{\max}+1,j} &= v_{i_{\max},j}, & j &= 1, \dots, j_{\max}, \\ u_{i,0} &= u_{i,1}, & u_{i,j_{\max}+1} &= u_{i,j_{\max}}, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.25)$$

3. *Outflow conditions:* In the outflow boundary condition the normal derivatives of both velocity components are set to zero at the boundary, which means that the total velocity does not change in the direction normal to the boundary. In the discrete case this can be realized by setting velocity values at the boundary equal to their neighboring velocities inside the domain, i.e.,

$$\begin{aligned} u_{0,j} &= u_{1,j}, & u_{i_{\max},j} &= u_{i_{\max}-1,j}, & j &= 1, \dots, j_{\max}, \\ v_{0,j} &= v_{1,j}, & v_{i_{\max}+1,j} &= v_{i_{\max},j}, \\ u_{i,0} &= u_{i,1}, & u_{i,j_{\max}+1} &= u_{i,j_{\max}}, & i &= 1, \dots, i_{\max}. \\ v_{i,0} &= v_{i,1}, & v_{i,j_{\max}} &= v_{i,j_{\max}-1}, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.26)$$

4. *Inflow conditions:* On an inflow boundary the velocities are explicitly given; we impose this for the velocities normal to the boundary (e.g., u on the left boundary) by directly fixing the values on the boundary line. For the velocity components tangential to the boundary (e.g., v at the left boundary), we achieve this by averaging the values on either side of the boundary similarly as in (3.22).

5. *Periodic boundary conditions:* For periodic boundary conditions in the x -direction, in which the boundary values on the left and right boundaries coincide, the values for u , v , and p are to be set as follows:

$$\begin{aligned} u_{0,j} &= u_{i_{\max}-1,j}, & u_{i_{\max},j} &= u_{1,j}, & p_{1,j} &= p_{i_{\max},j}, \\ v_{0,j} &= v_{i_{\max}-1,j}, & v_{1,j} &= v_{i_{\max},j}, & v_{i_{\max}+1,j} &= v_{2,j}. \end{aligned} \quad (3.27)$$

In contrast to the continuous problem, in which the same values are attained at $x = 0$ and $x = a$, the domain boundaries overlap here by one cell width so that the width of the domain must be chosen to exceed the period length by δx . Periodic boundary conditions in the y -direction are treated analogously.

Discretization of the Time Derivatives

We are left with the discretization of the time derivatives $\partial u / \partial t$ and $\partial v / \partial t$, for which we subdivide the time interval $[0, t_{\text{end}}]$ into equal subintervals $[n \delta t, (n + 1) \delta t]$, $n = 0, \dots, t_{\text{end}} / \delta t - 1$. This means that values of u , v , and p are considered only at times $n \delta t$. To discretize the time derivatives at time t_{n+1} we use *Euler's method*, which employs first-order difference quotients

$$\left[\frac{\partial u}{\partial t} \right]^{(n+1)} := \frac{u^{(n+1)} - u^{(n)}}{\delta t}, \quad \left[\frac{\partial v}{\partial t} \right]^{(n+1)} := \frac{v^{(n+1)} - v^{(n)}}{\delta t}, \quad (3.28)$$

where the superscript (n) denotes the time level. If all remaining terms in the differential equation, in particular the spatial derivatives, are evaluated at time t_n , one obtains an *explicit method*, in which the solution values at time t_{n+1} can be computed directly from those at time t_n . In contrast, *implicit methods*, in which spatial derivatives are evaluated at time t_{n+1} instead, permit much larger time steps to be taken while still maintaining stability. In every time step, however, these methods require the solution of a linear or even nonlinear system of equations.

3.2 The Algorithm

3.2.1 The Time-Stepping Loop

Beginning at time $t = 0$ with given initial values for u and v , time is incremented by δt in each step of an outer loop until the final time t_{end} is reached. At time step n the values of all variables are known and those at time t_{n+1} are to be computed.

We begin our description of the time-stepping loop by first performing the time discretization (3.28) of the terms $\partial u / \partial t$ and $\partial v / \partial t$ in the momentum equations (2.2a,b):

$$\begin{aligned} u^{(n+1)} &= u^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x - \frac{\partial p}{\partial x} \right], \\ v^{(n+1)} &= v^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y - \frac{\partial p}{\partial y} \right], \end{aligned}$$

The Algorithm

and, introducing the abbreviations

$$F := u^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x \right],$$

$$G := v^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y \right],$$

we obtain the form

To complete the discretization on the right-hand side of (3.28), we evaluate F and G at time t_n —while $\partial p / \partial x$ and $\partial p / \partial y$ are obtained by the time discretization of the corresponding terms.

u

v

This manner of discretization is called *semi-implicit* and *implicit* in the literature, since the implicit terms are computed once the corresponding explicit terms are computed.

The latter is now determined at time t_{n+1} . We substitute the values of u and v into the continuity equation

$$0 = \frac{\partial u^{(n+1)}}{\partial x} + \frac{\partial v^{(n+1)}}{\partial y},$$

which, after rearranging, yields the value of $p^{(n+1)}$:

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2}$$

In summary, the $(n + 1)$ st time step consists of the following steps:

- Step 1: Compute $F^{(n)}$,
- Step 2: Solve the Poisson equation for $p^{(n+1)}$,
- Step 3: Compute the new pressure values $p^{(n+1)}$.

²¹This equation together with the continuity equation determines the new pressure values $p^{(n+1)}$.

The Algorithm

and, introducing the abbreviations

$$F := u^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x \right], \quad (3.29)$$

$$G := v^{(n)} + \delta t \left[\frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y \right],$$

we obtain the form

$$\begin{aligned} u^{(n+1)} &= F - \delta t \frac{\partial p}{\partial x}, \\ v^{(n+1)} &= G - \delta t \frac{\partial p}{\partial y}. \end{aligned} \quad (3.30)$$

To complete the discretization of the momentum equations in time, the terms on the right-hand side of (3.30) must also be associated with a time level: we evaluate F and G at time level n —i.e., all velocities in (3.29) belong to time level t_n —while $\partial p / \partial x$ and $\partial p / \partial y$ are associated with time level t_{n+1} . This gives us the *time discretization of the momentum equations* (2.2a,b):

$$\begin{aligned} u^{(n+1)} &= F^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial x}, \\ v^{(n+1)} &= G^{(n)} - \delta t \frac{\partial p^{(n+1)}}{\partial y}. \end{aligned} \quad (3.31)$$

This manner of discretization may be characterized as being *explicit* in the velocities and *implicit* in the pressure; i.e., the velocity field at time step t_{n+1} can be computed once the corresponding pressure is known.

The latter is now determined by evaluating the continuity equation (2.2c) at time t_{n+1} . We substitute the relation (3.31) for the velocity field $(u^{(n+1)}, v^{(n+1)})^T$ into the continuity equation (2.2c) and obtain

$$0 = \frac{\partial u^{(n+1)}}{\partial x} + \frac{\partial v^{(n+1)}}{\partial y} = \frac{\partial F^{(n)}}{\partial x} - \delta t \frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial G^{(n)}}{\partial y} - \delta t \frac{\partial^2 p^{(n+1)}}{\partial y^2},$$

which, after rearranging, becomes a *Poisson equation for the pressure* $p^{(n+1)}$ at time t_{n+1} :²¹

$$\frac{\partial^2 p^{(n+1)}}{\partial x^2} + \frac{\partial^2 p^{(n+1)}}{\partial y^2} = \frac{1}{\delta t} \left(\frac{\partial F^{(n)}}{\partial x} + \frac{\partial G^{(n)}}{\partial y} \right). \quad (3.32)$$

In summary, the $(n+1)$ st time step consists of the following parts.

- Step 1: Compute $F^{(n)}$, $G^{(n)}$ according to (3.29) from the velocities $u^{(n)}$, $v^{(n)}$.
- Step 2: Solve the Poisson equation (3.32) for the pressure $p^{(n+1)}$.
- Step 3: Compute the new velocity field $(u^{(n+1)}, v^{(n+1)})^T$ using (3.31) with the pressure values $p^{(n+1)}$ computed in Step 2.

²¹This equation together with (3.31) ensures a divergence-free velocity field.

$$\begin{aligned} \frac{\partial(u^2)}{\partial x} - \frac{\partial(uv)}{\partial y} + g_x - \frac{\partial p}{\partial x}, \\ \frac{\partial(uv)}{\partial x} - \frac{\partial(v^2)}{\partial y} + g_y - \frac{\partial p}{\partial y}, \end{aligned}$$

Solving the pressure Poisson equation in Step 2 requires boundary values for the pressure. These result from multiplying the time-discrete momentum equations (3.31) with the exterior unit normal vector $\vec{n} := (n_1, n_2)^T$ on the boundary Γ , yielding²²

$$\begin{aligned}\text{grad} p^{(n+1)} \cdot \vec{n} &= \frac{\partial p^{(n+1)}}{\partial x} n_1 + \frac{\partial p^{(n+1)}}{\partial y} n_2 \\ &= -\frac{1}{\delta t} ((u^{(n+1)} - F^{(n)}) n_1 + (v^{(n+1)} - G^{(n)}) n_2).\end{aligned}\quad (3.33)$$

This approach corresponds to the *Chorin projection method* developed by Chorin [Chorin, 1968] and Temam [Temam, 1969].

It is equivalent to the SMAC-method [Amsden & Harlow, 1970a], [Amsden & Harlow, 1970b] (cf. also [Tome & McKee, 1994]), in which, using the continuity equation, only a pressure correction $\delta p^{(n)}$ is computed. This is then added to $p^{(n)}$, yielding $p^{(n+1)}$. A third alternative consists in computing pressure and velocity simultaneously by iteration (cf. [Hirt & Cook, 1972], [Hirt et al., 1975]).

3.2.2 The Discrete Momentum Equations

To obtain the fully discrete momentum equations, we have yet to discretize the spatial derivatives occurring in the time-discretized momentum equations (3.31). Making use of the formulas (3.19a) and (3.19b) we obtain

$$u_{i,j}^{(n+1)} = F_{i,j}^{(n)} - \frac{\delta t}{\delta x} (p_{i+1,j}^{(n+1)} - p_{i,j}^{(n+1)}), \quad i = 1, \dots, i_{\max} - 1, \quad j = 1, \dots, j_{\max}, \quad (3.34)$$

$$v_{i,j}^{(n+1)} = G_{i,j}^{(n)} - \frac{\delta t}{\delta y} (p_{i,j+1}^{(n+1)} - p_{i,j}^{(n+1)}), \quad i = 1, \dots, i_{\max}, \quad j = 1, \dots, j_{\max} - 1, \quad (3.35)$$

in which the quantities F and G from (3.29) are discretized at the right and upper edges of cell (i, j) , respectively:

$$\begin{aligned}F_{i,j} &:= u_{i,j} \\ &+ \delta t \left(\frac{1}{Re} \left(\left[\frac{\partial^2 u}{\partial x^2} \right]_{i,j} + \left[\frac{\partial^2 u}{\partial y^2} \right]_{i,j} \right) - \left[\frac{\partial(u^2)}{\partial x} \right]_{i,j} - \left[\frac{\partial(uv)}{\partial y} \right]_{i,j} + g_x \right), \\ &\quad i = 1, \dots, i_{\max} - 1, \quad j = 1, \dots, j_{\max},\end{aligned}\quad (3.36)$$

²²This results in the directional derivative of the pressure in the direction of the exterior normal vector n , $\partial p / \partial n := \lim_{\varepsilon \rightarrow 0} (p(\vec{x} + \varepsilon \vec{n}) - p(\vec{x})) / \varepsilon = \text{grad} p \cdot \vec{n}$. This leads to a *Neumann boundary condition*, as opposed to a *Dirichlet boundary condition*, which would prescribe the value of the function itself at the boundary. The Poisson equation with a Neumann boundary condition, $\Delta p = f$ in Ω , $\partial p / \partial n = g$ on Γ , is solvable only if the compatibility condition $\int_{\Gamma} g ds = \int_{\Omega} f dx$ holds. In this case the solution is determined uniquely up to an additive constant.

The Algorithm

$$G_{i,j} := v_{i,j}$$

$$+ \delta t \left(\frac{1}{Re} \left(\left[\frac{\partial^2 v}{\partial x^2} \right]_{i,j} + \left[\frac{\partial^2 v}{\partial y^2} \right]_{i,j} \right) - \left[\frac{\partial(v^2)}{\partial x} \right]_{i,j} - \left[\frac{\partial(uv)}{\partial y} \right]_{i,j} + g_y \right),$$

3.2.3 The Poisson Equation

The discrete quantities introduced in the previous section are required for the Laplacian (3.13), resulting in

$$\frac{p_{i+1,j}^{(n+1)} - 2p_{i,j}^{(n+1)} + p_{i-1,j}^{(n+1)}}{(\delta x)^2}$$

For $i \in \{1, i_{\max}\}$ or $j \in \{1, j_{\max}\}$ the right-hand side of (3.38) is given by

$$\begin{aligned}p_{0,j}, \\ p_{i,0},\end{aligned}$$

In addition, we need the following terms for the right-hand side of (3.38):

$$\begin{aligned}F_{0,j} \\ G_{i,0}\end{aligned}$$

which have not yet been specified.

To determine these boundary values, we use the boundary condition (3.33). The exterior normal vector $\vec{n} = (-1, 0)^T$, for instance,

$$\frac{p_{0,j}^{(n+1)}}{(\delta x)^2}$$

If we now insert this into the right-hand side of (3.38), we get

$$\frac{p_{2,j}^{(n+1)} - p_{1,j}^{(n+1)}}{(\delta x)^2} + \frac{p_{1,j}^{(n+1)}}{(\delta x)^2}$$

$$\begin{aligned}
G_{i,j} &:= v_{i,j} \\
&+ \delta t \left(\frac{1}{Re} \left(\left[\frac{\partial^2 v}{\partial x^2} \right]_{i,j} + \left[\frac{\partial^2 v}{\partial y^2} \right]_{i,j} \right) - \left[\frac{\partial(uv)}{\partial x} \right]_{i,j} - \left[\frac{\partial(v^2)}{\partial y} \right]_{i,j} + g_y \right), \tag{3.37} \\
i &= 1, \dots, i_{\max}, \quad j = 1, \dots, j_{\max} - 1.
\end{aligned}$$

3.2.3 The Poisson Equation for the Pressure

The discrete quantities introduced in Section 3.2.2, along with the discretization of the Laplacian (3.13), result in the discrete Poisson equation

$$\begin{aligned}
&\frac{p_{i+1,j}^{(n+1)} - 2p_{i,j}^{(n+1)} + p_{i-1,j}^{(n+1)}}{(\delta x)^2} + \frac{p_{i,j+1}^{(n+1)} - 2p_{i,j}^{(n+1)} + p_{i,j-1}^{(n+1)}}{(\delta y)^2} \\
&= \frac{1}{\delta t} \left(\frac{F_{i,j}^{(n)} - F_{i-1,j}^{(n)}}{\delta x} + \frac{G_{i,j}^{(n)} - G_{i,j-1}^{(n)}}{\delta y} \right), \tag{3.38} \\
i &= 1, \dots, i_{\max}, \quad j = 1, \dots, j_{\max}.
\end{aligned}$$

For $i \in \{1, i_{\max}\}$ or $j \in \{1, j_{\max}\}$, the following pressure boundary values are required in (3.38):

$$\begin{aligned}
p_{0,j}, \quad p_{i_{\max}+1,j}, \quad &j = 1, \dots, j_{\max}, \\
p_{i,0}, \quad p_{i,j_{\max}+1}, \quad &i = 1, \dots, i_{\max}.
\end{aligned}$$

In addition, we need the following values of F and G at the boundary to compute the right-hand side of (3.38):

$$\begin{aligned}
F_{0,j}, \quad F_{i_{\max},j}, \quad &j = 1, \dots, j_{\max}, \\
G_{i,0}, \quad G_{i,j_{\max}}, \quad &i = 1, \dots, i_{\max},
\end{aligned}$$

which have not yet been specified.

To determine these boundary values we look at the continuous pressure boundary condition (3.33). The resulting discretization along the left boundary (i.e., $\vec{n} = (-1, 0)^T$), for instance, is given by

$$\frac{p_{0,j}^{(n+1)} - p_{1,j}^{(n+1)}}{\delta x} = \frac{1}{\delta t} \left(u_{0,j}^{(n+1)} - F_{0,j}^{(n)} \right). \tag{3.39}$$

If we now insert this into the discrete pressure equation (3.38) for $i = 1$, we obtain

$$\begin{aligned}
&\frac{p_{2,j}^{(n+1)} - p_{1,j}^{(n+1)}}{(\delta x)^2} + \frac{p_{1,j+1}^{(n+1)} - 2p_{1,j}^{(n+1)} + p_{1,j-1}^{(n+1)}}{(\delta y)^2} \\
&= \frac{1}{\delta t} \left(\frac{F_{1,j}^{(n)} - u_{0,j}^{(n+1)}}{\delta x} + \frac{G_{1,j}^{(n)} - G_{1,j-1}^{(n)}}{\delta y} \right). \tag{3.40}
\end{aligned}$$

This reveals that this equation does not depend on the value of $F_{0,j}^{(n)}$, since $F_{0,j}^{(n)}$ occurs simultaneously in the right-hand side and the boundary condition of the discrete pressure equation. Hence $F_{0,j}^{(n)}$ can be selected arbitrarily. The simplest choice is $F_{0,j}^{(n)} = u_{0,j}^{(n+1)}$, which leads to $p_{0,j}^{(n+1)} = p_{1,j}^{(n+1)}$.²³²⁴ The right, upper, and lower boundaries are treated analogously, which results in the following boundary values for p , F , and G :

$$\begin{aligned} p_{0,j} &= p_{1,j}, & p_{i_{\max}+1,j} &= p_{i_{\max},j}, & j &= 1, \dots, j_{\max}, \\ p_{i,0} &= p_{i,1}, & p_{i,j_{\max}+1} &= p_{i,j_{\max}}, & i &= 1, \dots, i_{\max}, \end{aligned} \quad (3.41)$$

and

$$\begin{aligned} F_{0,j} &= u_{0,j}, & F_{i_{\max},j} &= u_{i_{\max},j}, & j &= 1, \dots, j_{\max}, \\ G_{i,0} &= v_{i,0}, & G_{i,j_{\max}} &= v_{i,j_{\max}}, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.42)$$

Due to (3.41) the pressure equation (3.38) must be modified in cells adjacent to the boundary of Ω . Using the definition of F and G at the boundary according to (3.42), we obtain

$$\begin{aligned} &\frac{\epsilon_i^E(p_{i+1,j}^{(n+1)} - p_{i,j}^{(n+1)}) - \epsilon_i^W(p_{i,j}^{(n+1)} - p_{i-1,j}^{(n+1)})}{(\delta x)^2} \\ &+ \frac{\epsilon_j^N(p_{i,j+1}^{(n+1)} - p_{i,j}^{(n+1)}) - \epsilon_j^S(p_{i,j}^{(n+1)} - p_{i,j-1}^{(n+1)})}{(\delta y)^2} \\ &= \frac{1}{\delta t} \left(\frac{F_{i,j}^{(n)} - F_{i-1,j}^{(n)}}{\delta x} + \frac{G_{i,j}^{(n)} - G_{i,j-1}^{(n)}}{\delta y} \right), \\ &i = 1, \dots, i_{\max}, \quad j = 1, \dots, j_{\max}, \end{aligned} \quad (3.43)$$

in place of (3.38). The parameters

$$\epsilon_i^W := \begin{cases} 0, & i = 1, \\ 1, & i > 1, \end{cases} \quad \epsilon_i^E := \begin{cases} 1, & i < i_{\max}, \\ 0, & i = i_{\max}, \end{cases} \quad \epsilon_j^S := \begin{cases} 0, & j = 1, \\ 1, & j > 1, \end{cases} \quad \epsilon_j^N := \begin{cases} 1, & j < j_{\max}, \\ 0, & j = j_{\max}, \end{cases}$$

indicate whether cell (i,j) lies adjacent to the domain boundary, in which case the corresponding pressure values in (3.38) must be eliminated. The superscripts W , E , N , and S indicate in which direction (west, east, north, or south) the boundary lies.²⁵

²³The condition that the normal derivative of the pressure be zero is purely a mathematical artifact.

²⁴Alternatively, $F_{0,j}^{(n)}$ could also be determined from the discrete momentum equation (3.34) requiring the value $u_{-1,j}$, which in turn can be determined from the discrete continuity equation in cell $(0,j)$. This particular boundary condition is employed in the *MAC method* [Harlow & Welch, 1965].

²⁵An efficient implementation of an iterative solver for (3.43) should avoid making this distinction at every step of the iteration. All cells not adjacent to a boundary cell can, for instance, always use (3.38).

The Algorithm

As a result, (3.43) represents a system of linear equations and i_{\max}, j_{\max} are solved using a suitable algorithm for linear systems of equations, direct methods of computer time and storage space being usually applied.²⁶ A classical variant, starting from an initial guess once in every cycle, corresponds to the following equation:

An improved variant is given by

$$\begin{aligned} it &= 1, \dots, it_{\max}, \\ i &= 1, \dots, i_{\max}, \\ j &= 1, \dots, j_{\max}, \\ p_{i,j}^{it+1} &:= (1 - \omega) p_{i,j}^{it} + \omega rhs_{i,j} \end{aligned}$$

The abbreviation $rhs_{i,j}$ stands for the right-hand side of (3.38) in cell (i,j) . The parameter ω is the relaxation parameter in the iterative process. The optimal choice of ω depends on the eigenvalue estimates for the matrix (3.43). Setting $\omega = 1$ yields the Jacobi method, either once a maximal number of iterations is reached or of the residual

$$r_{i,j}^{it} := \frac{\epsilon_i^E(p_{i+1,j}^{it} - p_{i,j}^{it}) - \epsilon_i^W(p_{i,j}^{it} - p_{i-1,j}^{it})}{(\delta x)^2} + \frac{\epsilon_j^N(p_{i,j+1}^{it} - p_{i,j}^{it}) - \epsilon_j^S(p_{i,j}^{it} - p_{i,j-1}^{it})}{(\delta y)^2}$$

has fallen below an absolute tolerance or a commonly used norm is reached.

²⁶A thorough description of iterative solvers for linear systems of equations can be found in the classical treatise [Varga, 1962] or [Barrett et al., 1993] and the references therein.

²⁷For very large pressure values, the Jacobi method is too lenient. Including the *red-black* variant is recommended.

f $F_{0,j}^{(n)}$, since $F_{0,j}^{(n)}$ condition of the boundary. The simplest right, upper, and following boundary

$$\text{ax}, \quad (3.41)$$

$$\text{ax}, \quad (3.42)$$

n cells adjacent to boundary according

$$(3.43)$$

$$- G_{i,j-1}^{(n)} \Big),$$

$$, \dots, j_{\max},$$

$$\epsilon_j^N := \begin{cases} 1, & j < j_{\max}, \\ 0, & j = j_{\max}, \end{cases}$$

dary, in which case d. The superscripts r south) the bound-

purely a mathematical

momentum equation (3.34) etc continuity equation IAC method [Harlow &

avoid making this dis- ry cell can, for instance,

The Algorithm

As a result, (3.43) represents a linear system of equations containing $i_{\max} j_{\max}$ equations and $i_{\max} j_{\max}$ unknowns $p_{i,j}$, $i = 1, \dots, i_{\max}$, $j = 1, \dots, j_{\max}$, to be solved using a suitable algorithm. For the solution of these very large, sparse linear systems of equations arising from the discretization of partial differential equations, direct methods such as Gaussian elimination are too costly in terms of computer time and storage, and thus iterative solution methods are generally applied.²⁶ A classical representative of these is the Gauss–Seidel method in which, starting from an initial approximation, each cell (i, j) is successively processed once in every cycle by modifying its pressure value in such a way that the corresponding equation is satisfied exactly.

An improved variant is given by the successive overrelaxation (SOR) method:

$$\begin{aligned} it &= 1, \dots, it_{\max}, \\ i &= 1, \dots, i_{\max}, \\ j &= 1, \dots, j_{\max}, \\ p_{i,j}^{it+1} &:= (1 - \omega) p_{i,j}^{it} + \frac{\omega}{\left(\frac{\epsilon_i^E + \epsilon_i^W}{(\delta x)^2} + \frac{\epsilon_j^N + \epsilon_j^S}{(\delta y)^2} \right)} \\ &\cdot \left(\frac{\epsilon_i^E p_{i+1,j}^{it} + \epsilon_i^W p_{i-1,j}^{it+1}}{(\delta x)^2} + \frac{\epsilon_j^N p_{i,j+1}^{it} + \epsilon_j^S p_{i,j-1}^{it+1}}{(\delta y)^2} - rhs_{i,j} \right). \end{aligned} \quad (3.44)$$

The abbreviation $rhs_{i,j}$ stands for the right-hand side of the pressure equation (3.38) in cell (i, j) . The parameter ω must be chosen from the interval $[0, 2]$. This choice can strongly affect the rate of convergence. The optimal choice of the relaxation parameter is discussed in Chapter 8.3 of [Stoer & Bulirsch, 1980]. The optimal choice of ω depends on the matrix of the linear system and requires eigenvalue estimates for the iteration matrix. A value often used in practice is $\omega = 1.7$. Setting $\omega = 1$ yields the Gauss–Seidel method. The iteration is terminated either once a maximal number of steps it_{\max} has been taken or when the norm of the residual

$$\begin{aligned} r_{i,j}^{it} &:= \frac{\epsilon_i^E (p_{i+1,j}^{it} - p_{i,j}^{it}) - \epsilon_i^W (p_{i,j}^{it} - p_{i-1,j}^{it})}{(\delta x)^2} \\ &+ \frac{\epsilon_j^N (p_{i,j+1}^{it} - p_{i,j}^{it}) - \epsilon_j^S (p_{i,j}^{it} - p_{i,j-1}^{it})}{(\delta y)^2} - rhs_{i,j}, \\ &i = 1, \dots, i_{\max}, \quad j = 1, \dots, j_{\max}, \end{aligned} \quad (3.45)$$

has fallen below an absolute tolerance eps or a relative tolerance $eps \|p^0\|$.²⁷ Two commonly used norms are the discrete L^2 -norm

²⁶A thorough description of iterative methods for solving linear systems of equations can be found in the classical treatise [Varga, 1962]; for more recent results, cf., e.g., [Hackbusch, 1994] or [Barrett et al., 1993] and the references therein.

²⁷For very large pressure values the absolute tolerance is too strict while for very small values it is too lenient. Including the magnitude of the pressure in the relative tolerance is thus to be recommended.

$$\|r^{it}\|_2 := \left(\frac{1}{i_{\max} j_{\max}} \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (r_{i,j}^{it})^2 \right)^{1/2} \quad (3.46)$$

and the maximum norm

$$\|r^{it}\|_\infty := \max \left\{ |r_{i,j}^{it}| \mid i = 1, \dots, i_{\max}, j = 1, \dots, j_{\max} \right\}. \quad (3.47)$$

We have used the L^2 -norm together with an absolute tolerance in our simulations. Initial values for the iteration to compute the pressure values $p^{(n+1)}$ are provided by the pressure values at time level n so that, in order to compute $p^{(1)}$, we require initial values $p^{(0)}$ for the pressure (usually $p^{(0)} = 0$) in addition to those for the velocities $u^{(0)}$ and $v^{(0)}$.

This solution approach, however, introduces a difficulty: the system matrix of the linear system (3.43) is singular since the underlying boundary value problem for the pressure has only Neumann boundary conditions. This means that, for the system to have a solution, the right-hand side must lie in the range space of the matrix. In this case the solution still possesses one degree of freedom corresponding to the additive constant up to which the solution of the continuous Navier–Stokes equations is determined. If the velocity field at time t_n fails to approximately satisfy the discrete continuity equation, then the linear system has no solution and nonphysical pressure values result in the iteration (3.44).²⁸ Numerical experiments have shown that this difficulty can be overcome not by modifying the Poisson equation at the boundary as in (3.43), but instead by satisfying the boundary condition (3.41) for the pressure by copying the pressure values along the boundary to their neighboring cells in the boundary strip prior to each iteration step; i.e.,

$$\begin{aligned} p_{0,j}^{it+1} &= p_{1,j}^{it}, & p_{i_{\max}+1,j}^{it+1} &= p_{i_{\max},j}^{it}, & j &= 1, \dots, j_{\max}, \\ p_{i,0}^{it+1} &= p_{i,1}^{it}, & p_{i,j_{\max}+1}^{it+1} &= p_{i,j_{\max}}^{it}, & i &= 1, \dots, i_{\max}. \end{aligned} \quad (3.48)$$

The SOR iteration (3.44) and the calculation of the residual (3.45) then proceed as described above except that the distinction of the various cases involving the ϵ -parameters is no longer necessary, as these are now identically equal to 1.

Once the pressure values at time t_{n+1} have been calculated using either of these two methods, the velocity values u and v can be calculated according to (3.34) and (3.35).

When traditional iterative methods for solving linear systems of equations such as the Gauss–Seidel method or the SOR method are applied to systems originating from the discretization of partial differential equations, the convergence behavior deteriorates as the spacing between the grid lines is decreased. The result is that increasingly fine discretizations require more and more iteration steps to reduce the iteration error below a given tolerance.

²⁸Particularly for complicated geometries such as those described in Section 3.4, it is practically impossible to find initial velocity values which satisfy the discrete continuity equation.

A new class of iterative methods developed in the early 1980s improved the general, implementation-independent discretization methods. The basic idea underlies the solution on the unit square by doubling the number of iterations [Hackbusch et al. 1984], [Hackbusch and Schneider 1985].

3.2.4 The Stabilized Iteration

In order to ensure stability, oscillations, stabilization

In the literature, the term

$$\frac{2\delta t}{Re} < \left(\frac{\Delta x}{\Delta y} \right)^2$$

(cf., e.g., [Tomek et al. 1991]) is often used to estimate the values of the vorticity and the Courant–Friedrichs–Lax condition number. It is based on the fact that the numerical solution travels a distance Δx in time Δt and that the stability analysis requires that the ratio of the time step to the spatial step is sufficiently small.

An adaptive time-stepping scheme was proposed by [McKee, 1991] and it is based on the fact that each of the time steps is determined by the local stability analysis.

The factor $\tau \in (0, 1]$ is used to determine the time step size Δt using a fixed global time step Δt_0 .

3.2.5 Summary

The entire process of solving the Navier–Stokes equations is summarized as follows:

3.3 Implementation

We now turn to the implementation of the algorithm described in Section 3.2.

(3.46)

(3.47)

nce in our simu-
values $p^{(n+1)}$ are
to compute $p^{(1)}$,
0) in addition to

system matrix of
ary value problem
s means that, for
a the range space
degree of freedom
1 of the continuous
at time t_n fails to
the linear system
iteration (3.44).²⁸
e overcome not by
3), but instead by
opying the pressure
oundary strip prior

j_{\max} ,
 i_{\max} .
(3.48)

(3.45) then proceed
; cases involving the
ally equal to 1.
using either of these
l according to (3.34)

ems of equations such
l to systems originated
the convergence be-
decreased. The result
ore iteration steps to

n Section 3.4, it is practi-
te continuity equation.

A new class of iterative methods for solving discrete elliptic equations was developed in the late 1970s. In these *multigrid methods*, as they are known, the number of iteration steps is independent of the number of unknowns. Thus, in general, improving the approximation of the continuous problem with a finer discretization no longer leads to an increase in the number of iteration steps. The basic idea underlying multigrid methods is that correction terms for the discrete solution on the original grid are computed on successively coarse grids obtained by doubling the spacing between the grid lines of each previous grid (cf. [Brandt, 1984], [Hackbusch, 1985], [Wesseling, 1992]).

3.2.4 The Stability Condition

In order to ensure stability of the numerical algorithm and avoid generating oscillations, stability conditions must be imposed on the stepsizes δx , δy , and δt .

In the literature one most often finds the three conditions

$$\frac{2\delta t}{Re} < \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right)^{-1}, \quad |u_{\max}| \delta t < \delta x, \quad |v_{\max}| \delta t < \delta y \quad (3.49)$$

(cf., e.g., [Tome & McKee, 1994]). Here $|u_{\max}|$ and $|v_{\max}|$ are the maximal absolute values of the velocities occurring on the grid. The latter two are the famous *Courant–Friedrichs–Lewy (CFL) conditions*. They state that no fluid particle may travel a distance greater than the mesh spacing δx or δy in time δt . A detailed stability analysis can be found, e.g., in [Peyret & Taylor, 1983] or [Roache, 1976].

An adaptive stepsize control based on these stability conditions is used in [Tome & McKee, 1994]. This is implemented by selecting δt for the next time step so that each of the three conditions (3.49) is satisfied:

$$\delta t := \tau \min \left(\frac{Re}{2} \left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right)^{-1}, \frac{\delta x}{|u_{\max}|}, \frac{\delta y}{|v_{\max}|} \right). \quad (3.50)$$

The factor $\tau \in]0, 1]$ is a safety factor. The algorithm can, of course, also be run using a fixed given time step length which satisfies the stability condition for all time steps.

3.2.5 Summary

The entire procedure is outlined once more for clarity in Algorithm 1.

3.3 Implementation

We now turn to the implementation in the C programming language of the algorithm described above. For now, we adopt the restriction that only one type of