Institute of Scientific Computing
Technical University Braunschweig
Prof. Hermann G. Matthies, Ph.D.

First we will introduce the fluid mechanics problem we will deal with in this practical course and present the partial differential equations that describe the problem. Then we develop an appropriate discretisation of these differential equations and an algorithm that solves the problem numerically.

1 The problem definition

In this practical course we are supposed to deal with *incompressible, viscous, laminar and unsteady* fluid motion. Fluids include both liquids and gases. The two fluids that occur most often, naturally or in flow machinery, are water (often in a liquid phase) and air.
As mentioned above we impose some restrictions on the fluid we consider in this practical course:

Incompressible The first restriction is that we consider only liquids (e.g. water) which is not *compressible*.

Viscosity The *viscosity* describes the "toughness" of the medium. For instance, the viscosity of air has a lower value as of water, but the viscosity of water has a lower one as of honey. In the considered flow the viscosity city of water is taken into account.

Laminar For the case of low velocities and low viscosity value (for the case of the low Reynolds number) the flow is rather regular and not mixing up, i.e. it is considered as *laminar* and not *turbulent*. We are not considering turbulent flows here.

Unsteady The meaning of *unsteady* flow is that the flow is supposed to change in time.

In this practical course only this particular kind of flow is considered. On this example one can learn about very common problems and techniques, which are present in the engineering computational fluid dynamics (CFD).

2 The mathematical description of viscous fluid flows: the Navier-Stokes equations

The fluid motion in the domain $\Omega \subset I\!R^N$ ($N \in \{2, 3\}$) along the time $t \in [0, T]$ could be described by the three physical properties, which are functions of space and time:

- the velocity field $\vec{u} : \Omega \times [0,T] \longrightarrow I\!R^N$,
- the pressure function $p : \Omega \times [0,T] \longrightarrow I\!R$,
- the scalar density $\rho : \Omega \times [0,T] \longrightarrow I\!R$.

For the incompressible, laminar, viscous and unsteady class of flow the density ρ is constant, e.g. $\rho(\vec{x},t) = \rho_c = \text{const}$, and the flow is governed by the *momentum equation*

$$\frac{\partial}{\partial t}\vec{u} + (\vec{u}\cdot \text{grad})\,\vec{u} + \text{ grad } p = \nu\Delta\vec{u} + \vec{g}, \tag{1}$$

and the *continuity equation*

$$\text{div } \vec{u} = 0. \tag{2}$$

The system of partial differential equations (1) and (2) is called *Navier-Stokes equations.* The pressure p is determined only up to a constant. The real number ν denotes the viscosity factor. In the non-dimensional case the viscosity factor is the inverse Reynolds number Re. $\vec{g} : \Omega \times [0,T] \longrightarrow I\!R^N$ is the given external volume force.

Task 1 *To simplify the task we just consider the flow behaviour around bodies in the two dimensional domain $\Omega \subset I\!R^2$. To obtain the Navier-Stokes equation (1) and (2) in two-dimensional Cartesian coordinates, set $\vec{x} = (x,y)^T$, $\vec{u} = (u,v)^T$, $\vec{g} = (g_x, g_y)^T$, simplify the term $(\vec{u}\cdot \text{grad})\,\vec{u}$ (hint: use the continuity equation (2) appropriately). For the notation of flow velocity use u for x-direction and v for the y-direction.*

In the following we just consider the two dimensional case (see task 1) on the rectangular domain $\Omega = [0,a]\times[0,b]$ and in the time interval $[0,T]$.
At time $t = 0$ the initial values $u = u_0(x,y)$ and $v = v_0(x,y)$ which fulfil the continuity equation (2), are given.
Boundary conditions at all time steps $t \in [0,T]$ are required so that we obtain an initial boundary problem.
For the formulation of the boundary conditions w_1 indicates the velocity component perpendicular to the boundary and w_2 the velocity component parallel to the boundary. $\partial w_1/\partial n$ and $\partial w_2/\partial n$ are the partial derivatives in perpendicular direction.
At the perpendicular boundary of Ω we have:

$$w_1 = u,\ w_2 = v,\ \frac{\partial w_1}{\partial n} = \frac{\partial u}{\partial x}\ ,\ \frac{\partial w_2}{\partial n} = \frac{\partial v}{\partial x}\ ,$$

at the horizontal boundary we have:

$$w_1 = v,\ w_2 = u,\ \frac{\partial w_1}{\partial n} = \frac{\partial v}{\partial y}\ ,\ \frac{\partial w_2}{\partial n} = \frac{\partial u}{\partial y}\ .$$

The following boundary conditions for points (x,y) of the solid boundary $\Gamma := \partial\Omega$ are appropriate:

No-slip no fluid flows through the walls and the fluid cling to the boundary, that means

$$w_1(x,y) = 0,\ w_2(x,y) = 0 \quad \text{for } (x,y) \in \Gamma = \partial\Omega.$$

Free-slip no fluid flows through the walls and in contrast to "no-slip" conditions there is no waste of friction along the boundary, that means

$$w_1(x,y) = 0,\ \frac{\partial w_2}{\partial n}(x,y) = 0 \quad \text{for } (x,y) \in \Gamma = \partial\Omega.$$

Inflow both velocities

$$w_1(x,y) = w_1^0,\ w_2(x,y) = w_2^0,$$

w_1^0, w_2^0 are given.

Outflow both velocities do not change in the direction perpendicular to the boundary, that means

$$\frac{\partial w_1}{\partial n}(x,y) = 0,\ \frac{\partial w_2}{\partial n}(x,y) = 0.$$

Are the velocities and not their normal derivatives at all boundaries given, the boundary integral of the velocities perpendicular to the boundaries must be zero. So we have

$$\int_\Gamma \begin{pmatrix} u \\ v \end{pmatrix} \cdot \vec{n}\, ds = 0 \tag{3}$$

.

Example 1 1. *In the first example a computational domain with four boundaries, each with "no-slip" conditions, is given. The upper wall moves with constant velocity to the right, that means $w_2 = w_2^0 = const$ at the upper boundary in contrast to ordinary "no-slip" condition $w_2 = 0$.*

2. *The second example shows a flow around a sloped timber. We have inflow conditions at the right boundary and outflow conditions at the left boundary. Both the upper and lower boundary has "no-slip" conditions.*

If we choose appropriate (physically correct) boundary conditions, we obtain a well-posed problem with an unique solution.

3 Discretisation

The process of converting the continuous partial differential equation into a discrete system of algebraic equations is called discretisation. Computational techniques are used to obtain an approximate solution of the governing equation and boundary conditions. For the discretisation of the PDE's we will use the finite difference method. Therefore we approximate the

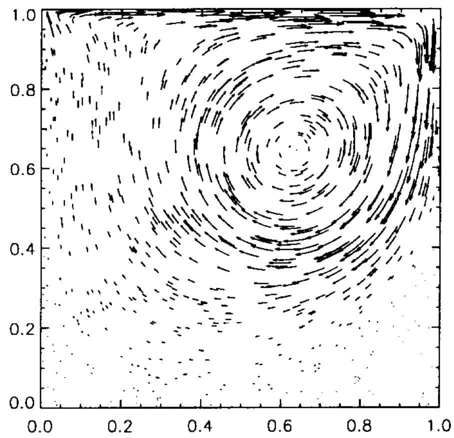

Figure 1: *Driven Cavity*

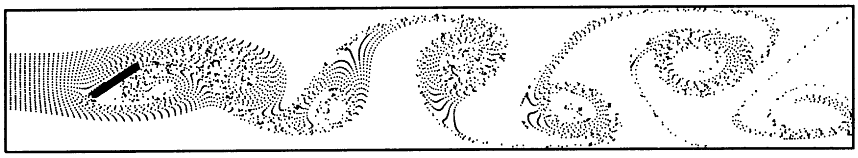

Figure 2: *Flow around a slope timber*

domain $\Omega = [0, a] \times [0, b] \subset I\!R^2$ with a grid and we just consider the grid's cross points as the discretisation of the continuous domain.
The grid has $imax$ cells in x-direction and $jmax$ cells in y-direction of equal sizes. So the distance between two lines of the grid can be computed by $\delta x = \frac{a}{imax}$, $\delta y = \frac{b}{jmax}$.
The cell with the index (i, j) indicates the sub domain $[(i-1)\delta x, i\delta x] \times [(j-1)\delta y, j\delta y]$.
In the case of discretisation of the Navier-Stokes equation often a „staggered grid“is used (stability). That means that the variables belong to index (i, j) do not lie in the same grid point.
We are using a grid where the pressure p is in the centre point of each cell, the velocity u in the centre points of the perpendicular cell boundary and the velocity v in the horizontal boundary of each cell.
The index (i, j) has the pressure at the centre point of the cell (i, j), the velocity u at the right cell boundary and the velocity v at the upper cell boundary of the cell (i, j)
The pressure $p_{i,j}$ has the coordinates $((i - \frac{1}{2})\delta x, (j - \frac{1}{2})\delta y)$, the velocities $u_{i,j}$ $(i\delta x, (j - \frac{1}{2})\delta y)$ and the velocities $v_{i,j}$ $((i - \frac{1}{2})\delta x, j\delta y)$ (see figure 3).
The points for u, v and p belong therefore to three, here not presented grids, which are respectively displaced. That means that not all values lie on the boundary of the domain. There are no v-values on the perpendicular boundaries and no u-values on the horizontal

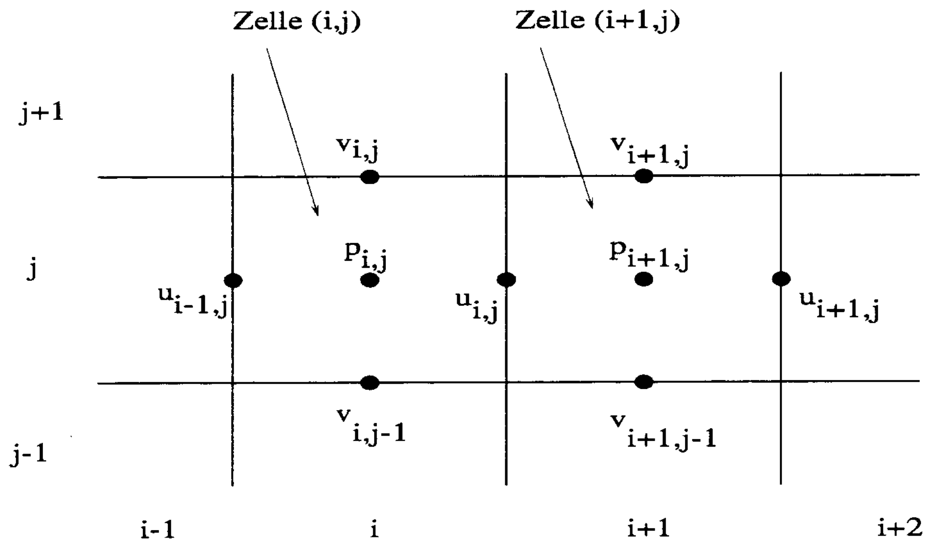

Figure 3: *Staggered Grid*

boundaries. Thus an additional boundary layer of grid cells is carried along (see figure 4), so that the boundary conditions can be fulfilled by a middle value of the nearest values (see subsection on the subject of „The edge values for the discrete equations“in the next chapter).

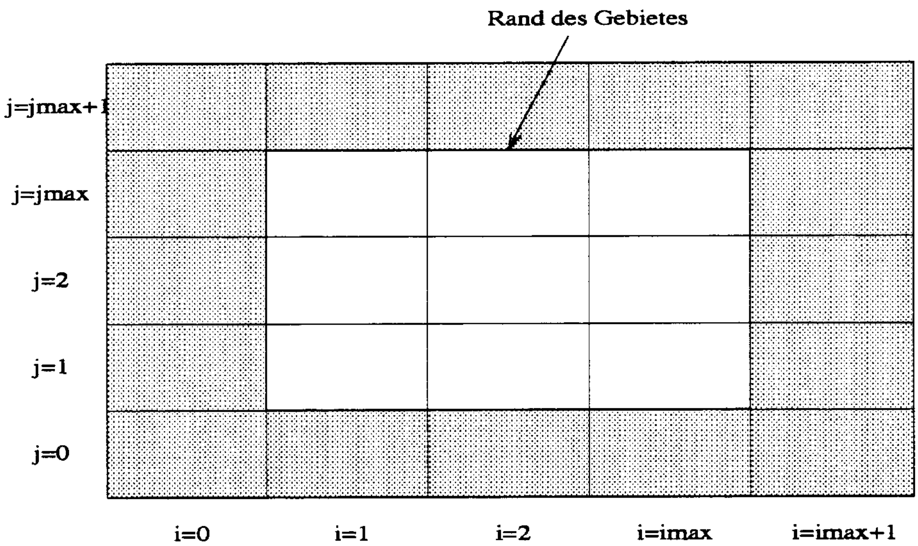

Figure 4: *Domain with boundary cells*

Discretisation of the Navier-Stokes equation: First we consider the spatial derivatives. The momentum equation (1) will be evaluated in the perpendicular and horizontal edges and the continuity equation (2) in the central points.

The terms of equation (1) are replaced at the central point of the right edge of cell (i,j), $i = 1,\ldots,imax-1$, $j = 1,\ldots,jmax$ by

$$\begin{aligned}
\left[\frac{\partial(u^2)}{\partial x}\right]_{i,j} &:= \frac{1}{\delta x}\left(\left(\frac{u_{i,j}+u_{i+1,j}}{2}\right)^2 - \left(\frac{u_{i-1,j}+u_{i,j}}{2}\right)^2\right) \\
&\quad + \alpha\frac{1}{\delta x}\left(\frac{|u_{i,j}+u_{i+1,j}|}{2}\frac{u_{i,j}-u_{i+1,j}}{2} - \frac{|u_{i-1,j}+u_{i,j}|}{2}\frac{u_{i-1,j}-u_{i,j}}{2}\right), \qquad (4) \\
\left[\frac{\partial(uv)}{\partial y}\right]_{i,j} &:= \frac{1}{\delta y}\left(\frac{v_{i,j}+v_{i+1,j}}{2}\frac{u_{i,j}+u_{i,j+1}}{2} - \frac{v_{i,j-1}+v_{i+1,j-1}}{2}\frac{u_{i,j-1}+u_{i,j}}{2}\right) \\
&\quad + \alpha\frac{1}{\delta y}\left(\frac{|v_{i,j}+v_{i+1,j}|}{2}\frac{u_{i,j}-u_{i,j+1}}{2} - \frac{|v_{i,j-1}+v_{i+1,j-1}|}{2}\frac{u_{i,j-1}-u_{i,j}}{2}\right) \qquad (5) \\
\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} &:= \frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\delta x)^2}, \qquad (6) \\
\left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j} &:= \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{(\delta y)^2}, \qquad (7) \\
\left[\frac{\partial p}{\partial x}\right]_{i,j} &:= \frac{p_{i+1,j}-p_{i,j}}{\delta x}, \qquad (8)
\end{aligned}$$

and at the central point of the upper edge of cell (i,j) for $i = 1,\ldots,imax$ and $j = 1,\ldots,jmax-1$ by

$$\begin{aligned}
\left[\frac{\partial(v^2)}{\partial y}\right]_{i,j} &:= \frac{1}{\delta y}\left(\left(\frac{v_{i,j}+v_{i,j+1}}{2}\right)^2 - \left(\frac{v_{i,j-1}+v_{i,j}}{2}\right)^2\right) \\
&\quad + \alpha\frac{1}{\delta y}\left(\frac{|v_{i,j}+v_{i,j+1}|}{2}\frac{v_{i,j}-v_{i,j+1}}{2} - \frac{|v_{i,j-1}+v_{i,j}|}{2}\frac{v_{i,j-1}-v_{i,j}}{2}\right), \qquad (9) \\
\left[\frac{\partial(uv)}{\partial x}\right]_{i,j} &:= \frac{1}{\delta x}\left(\frac{u_{i,j}+u_{i,j+1}}{2}\frac{v_{i,j}+v_{i+1,j}}{2} - \frac{u_{i-1,j}+u_{i-1,j+1}}{2}\frac{v_{i-1,j}+v_{i,j}}{2}\right) \\
&\quad + \alpha\frac{1}{\delta x}\left(\frac{|u_{i,j}+u_{i,j+1}|}{2}\frac{v_{i,j}-v_{i+1,j}}{2} - \frac{|u_{i-1,j}+u_{i-1,j+1}|}{2}\frac{v_{i-1,j}-v_{i,j}}{2}\right) \qquad (10) \\
\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} &:= \frac{v_{i+1,j}-2v_{i,j}+v_{i-1,j}}{(\delta x)^2}, \qquad (11) \\
\left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j} &:= \frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{(\delta y)^2}, \qquad (12) \\
\left[\frac{\partial p}{\partial y}\right]_{i,j} &:= \frac{p_{i,j+1}-p_{i,j}}{\delta y}. \qquad (13)
\end{aligned}$$

The parameter α is between 0 and 1. For $\alpha = 0$ one receives an approximation of second order for the differential operators, i.e. the approximation error has the precision $O(\frac{1}{(\delta x)^2})$ respectively $O(\frac{1}{(\delta y)^2})$. These approximation can however for slight viscosity values, or if the derivative terms $\frac{\partial}{\partial x}$ resp. $\frac{\partial}{\partial y}$ are affect with essentially larger factors the problem more than the derivative terms $\frac{\partial^2}{\partial x^2}$ and $\frac{\partial^2}{\partial y^2}$, lead to oscillations in the solution. In such cases the so-called "Donor-Cell-Scheme" ($\alpha = 1$) must be used, which delivers a first order approximation. In practise one uses a mixture of both methods with $\alpha \in [0, 1]$. The parameter α should be chosen thereby in such a manner, that the inequality

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

is fulfilled. The spatial derivatives of the continuity equation (2) in the central points of the cell (i, j), $i = 1, \ldots, imax$, $j = 1, \ldots, jmax$ are replaced by the terms

$$\left[\frac{\partial u}{\partial x}\right]_{i,j} := \frac{u_{i,j} - u_{i-1,j}}{\delta x}, \qquad \left[\frac{\partial v}{\partial y}\right]_{i,j} := \frac{v_{i,j} - v_{i,j-1}}{\delta y} \tag{14}$$

.

Now we consider the discretisation of the time derivatives $\frac{\partial u}{\partial t}$ and $\frac{\partial v}{\partial t}$.
Therefore the time interval $[0, T]$ is decomposed into equal sub intervals $[n\delta t, (n + 1)\delta t]$, $n = 0, 1, \ldots, \frac{T}{\delta t} - 1$. The discrete values for u, v and p are considered in the time steps $n\delta t$. For the discretisation of the time derivatives at time step t_{n+1} we use the Euler method, that means first order differential quotients.

$$\left[\frac{\partial u}{\partial t}\right]^{(n+1)} := \frac{u^{(n+1)} - u^{(n)}}{\delta t}, \quad \left[\frac{\partial v}{\partial t}\right]^{(n+1)} := \frac{v^{(n+1)} - v^{(n)}}{\delta t}, \tag{15}$$

where the upper index n is the time-step.
The continuity equation (2) is evaluated at time-step $n + 1$. This corresponds to an explicit time step-method with the help of the Euler-formula. There are as well implicit methods where values at time step $n + 1$ occur also on the right hand side. In general these methods allow essentially larger time steps, but there must be solved a linear system of equations in each time-step.

4 The algorithm

The algorithm is subdivided into the following steps:

4.1 The time loop

Within an outer time loop, which starts at time step $t = 0$, the time is incremented in each time step till the time T is reached. The discretisation of the PDE's as described above take place at time step n ($n = 0, 1, \ldots$). The values at time step n are given and the values at time step $n + 1$ should be computed.

4.2 The discrete momentum equation

The discrete momentum equation, one obtains by inserting the discretisations into equation (1), must be reordered so that all velocities $u_{i,j}^{(n+1)}$ and $v_{i,j}^{(n+1)}$ stand on the left hand side of the equation. The pressure $p_{i,j}^{(n+1)}$ will also be evaluated at time step t_{n+1}. We obtain an equation of the form

$$u_{i,j}^{(n+1)} = F_{i,j}^{(n)} - \frac{\delta t}{\delta x}\left(p_{i+1,j}^{(n+1)} - p_{i,j}^{(n+1)}\right) \tag{16}$$

for $i = 1, \ldots, imax-1,\ j = 1, \ldots, jmax$ and

$$v_{i,j}^{(n+1)} = G_{i,j}^{(n)} - \frac{\delta t}{\delta y}\left(p_{i,j+1}^{(n+1)} - p_{i,j}^{(n+1)}\right) \tag{17}$$

for $i = 1, \ldots, imax,\ j = 1, \ldots, jmax-1$.

The terms $F_{i,j}^{(n)}$ include the velocities at time step n of the discrete momentum equation. With the discrete expressions from the previous sections at time step n we obtain:

$$F_{i,j} = u_{i,j} + \delta t\left(\nu\left(\left[\frac{\partial^2 u}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 u}{\partial y^2}\right]_{i,j}\right) - \left[\frac{\partial (u^2)}{\partial x}\right]_{i,j} - \left[\frac{\partial (uv)}{\partial y}\right]_{i,j} + g_x\right) \tag{18}$$

for $i = 1, \ldots, imax-1,\ j = 1, \ldots, jmax$ and

$$G_{i,j} = v_{i,j} + \delta t\left(\nu\left(\left[\frac{\partial^2 v}{\partial x^2}\right]_{i,j} + \left[\frac{\partial^2 v}{\partial y^2}\right]_{i,j}\right) - \left[\frac{\partial (v^2)}{\partial y}\right]_{i,j} - \left[\frac{\partial (uv)}{\partial x}\right]_{i,j} + g_y\right) \tag{19}$$

for $i = 1, \ldots, imax,\ j = 1, \ldots, jmax-1$.

4.3 The pressure equation

Now we can insert the velocities $u_{i-1,j}^{(n+1)}$, $u_{i,j}^{(n+1)}$, $v_{i,j-1}^{(n+1)}$ and $v_{i,j}^{(n+1)}$ in the cell (i,j) of the discrete continuity equation and obtain an equation which contains only pressure values at time step $n+1$ and (already calculated) velocities at time-step n.

Task 2 *Transform the discretised equations to obtain the pressure equation!*

This discrete equation is a discrete Poisson equation for $p^{(n+1)}$

$$\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = f$$

on the domain Ω along with a given right hand side f. For a unique solution we still require boundary values for the pressure (for $i = 0, imax+1,\ j = 0, jmax+1$), which we can obtain from the impulse equations at the boundaries (see next paragraph). Now it is possible, to solve the pressure equation along with an arbitrary method for solving linear equations (e.g. SOR method). In our case it is reasonable to use the pressure values $p^{(n)}$ at time step n as start values of the iteration for the calculation of the pressure values $p^{(n+1)}$. With the calculated pressure values at time step $n+1$ the velocities u and v can be calculated according to (16) and (17).

Task 3 *Formulate the algorithm for the SOR method, and write up the method and its characteristics.*

4.4 The boundary values of the discrete equations under "no-slip" conditions

For u, v and p we still require the boundary conditions for the discrete equation. In our first example we just consider "no-slip" conditions, so our focus in the section lies on this type of boundary condition. We will treat the other types of boundary conditions later.
To calculate $F_{i,j}$ and $G_{i,j}$ by (18) and (19) we are using values of u and v for $i = 1, \ldots, imax$ and $j = 1, \ldots, jmax$ which can lie on the boundary or even outside. According to "no-slip" conditions the continuous velocities at the boundary are zero, so we set

$$u_{0,j} = 0,\ u_{imax,j} = 0, \quad j = 1, \ldots, jmax,$$

$$v_{i,0} = 0,\ v_{i,jmax} = 0, \quad i = 1, \ldots, imax.$$

for the boundary values.
On the perpendicular boundary there are no appropriate v-values for edges and on the horizontal boundary no u-values. The boundary value 0 is obtained here by an averaging of the values on both sides of the boundary (see figure 5):

$$v_r := \frac{v_a + v_i}{2} = 0 \implies v_a = -v_i.$$

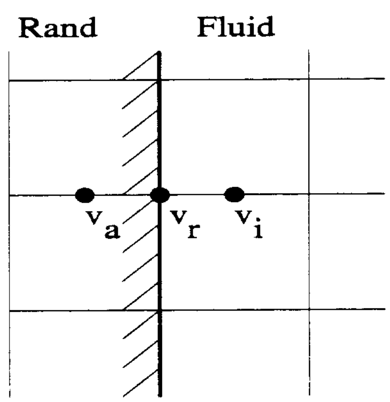

Figure 5: *Boundary conditions*

Therefore we obtain further conditions at the four boundaries of the domain Ω:

$$v_{0.j} = -v_{1,j},\ v_{imax+1,j} = -v_{imax,j} \quad \text{for } j = 1, \ldots, jmax, \tag{20}$$

$$u_{i,0} = -u_{i,1},\ u_{i,jmax+1} = -u_{i,jmax} \quad \text{for } i = 1, \ldots, imax. \tag{21}$$

In addition we still require boundary conditions for the pressure. These values result from the impulse equations at the boundary. We set

$$p_{0,j} = p_{1,j}, \; p_{imax+1,j} = p_{imax,j}, \tag{22}$$

for $j = 1, \ldots, jmax$ und

$$p_{i,0} = p_{i,1}, \; p_{i,jmax+1} = p_{i,jmax} \tag{23}$$

for $i = 1, \ldots, imax$.

During the calculation of the right hand side of the pressure equation in the near of the boundary we need values $F_{i,j}$ and $G_{i,j}$ which lie on the boundary, so we need a suitable modification of these values. Analogously to the pressure values at the boundary for $j = 1, \ldots, jmax$ we set

$$F_{0,j} = u_{0,j}, \; F_{imax,j} = u_{imax,j}, \tag{24}$$

und for $i = 1, \ldots, imax$

$$G_{i,0} = v_{i,0}, \; G_{i,jmax} = v_{i,jmax}. \tag{25}$$

4.5 The stability condition

To guarantee the stability of the numerical algorithm and to produce no oscillations, the following three stability conditions must be put to the step-sizes δx, δy and δt:

$$2\nu\delta t < \frac{(\delta x)^2(\delta y)^2}{(\delta x)^2 + (\delta y)^2}\,, \quad |u_{max}|\,\delta t < \delta x\,, \quad |v_{max}|\,\delta t < \delta y\,. \tag{26}$$

The last both conditions are the known Courant-Friedrichs-Levi conditions (CFL conditions). CLF condition indicates, that a particle of fluid should not travel more than one spacial step-size δx or δy in one time-step δt. We will fulfil the stability conditions at our examples thereby, that we choose the step-sizes small enough.

Task 4 *Represent the algorithm in form of a structured chart or a data flow graph.*