An unstructured staggered scheme with quadrilaterals for the Navier-Stokes equations

Graduation thesis, TUDelft/ITS

G. van der Veer
Graduation committee:
A. Segal,
I. Wenneker,
P. Wesseling.

December 15, 1999

Abstract

problems in fluid dynamics are often described by the Navier-Stokes equations, which express the conservation of mass, momentum and energy. In physics such equations are usually made dimensionless. This is done by using a nondimensionalization method for compressible flows, but this method is not suitable for incompressible flows. Here we only consider subsonic flows. In reality it is impossible to solve the non-linear Navier-Stokes equations analytically, therefore they will be treated numerically. In order to do this we need a discretization. For that purpose the dimensionless equations are discretized by the finite volume method. We make use of a staggered placement of the variables on an unstructured grid with quadrilaterals. The momentum and enthalpy are calculated implicitly and the density explicitly during the time stepping. The next logical step is to make an implementation. Due to the limited available time the implementation has been restricted to the Euler equations. The equations are first tested separately and after that we consider a more realistic problem (a flow through a bend) with the coupled system of equations.

Contents

Chapter 1

Introduction

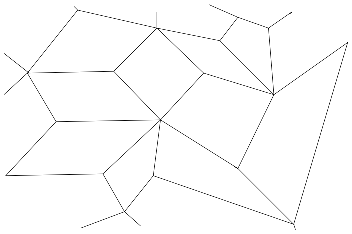

Figure 1.1: An unstructured grid.

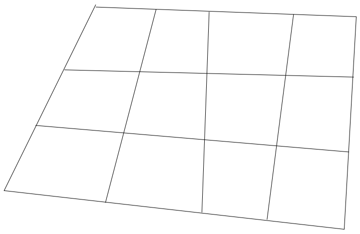

Figure 1.2: A structured grid.

Problems in fluid dynamics, for example the modeling of a flow in a pipe or the airflow around a wing, are usually described by the Navier-Stokes equations. These equations describe the conservation of mass, momentum and energy. The Navier-Stokes equations form a set of partial differential equations. Given appropriate initial and boundary conditions we can solve for the density, momentum and energy, which are called unknowns. All other interesting quantities (e.g. pressure, velocity, temperature) can be obtained easily.

In practice it is impossible to solve the Navier-Stokes equations analytically except for some very simple cases. For that reason one applies a discretization method. Examples of discretization methods are the finite elements method and the finite volume method.

For this purpose we generate a grid over the domain under consideration. In general we distinguish between two types of grids, namely structured and unstructured grids. A structured grid usually consists of blocks of cells. Each block can be mapped onto a rectangular grid with a constant number of cells in each of the coordinate directions. The number of cells that meet each other in an internal vertex is always 4 in a 2-dimensional space for quadrilaterals, see Figure 1.2. In an unstructured grid the arrangement of the cells is more or less arbitrary, see Figure 1.1.

In practice it is much easier to generate an unstructured grid for a complex

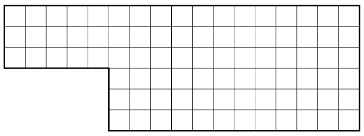

Figure 1.3: An unstructured grid.

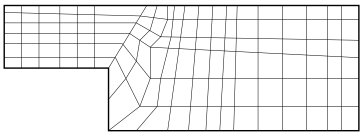

Figure 1.4: A structured grid.

domain than a structured grid (see Figures 1.3 and 1.4). However, the price to pay is that datastructure (the way in which data are ordered) becomes more complicated and that it is a harder task to develop a discretization. In the numerical analysis group of Delft University a start has been made to create a finite volume Navier-Stokes solver for an unstructured grid consisting of triangles. It is clear that when the aspect ratio of the cells (the ratio between the length and width) becomes large, it may be better to use quadrilaterals. Using triangles in this situation could result in sharp and large angles, which is not desirable. Quadrilateral cells are especially needed in the neighborhood of boundary layers. The ultimate goal is of course to use a mixture of quadrilaterals and triangles. However, in this report we shall limit ourselves to use quadrilaterals only.

Over the last two decades the finite volume method has become very popular to solve these equations because it is based on conservation laws, thus on physics. Hence we are using the finite volume method.

Another important issue is the positions of the unknowns. In a collocated grid all variables are stored in the same position, usually either the center of the cells or the vertices. For an incompressible flow however, it is known that a staggered grid arrangement, where scalars and vector components are positioned at different places (for example the cell centers, the midpoints of the cell faces or the cell vertices) is more accurate. Since we want to develop a scheme that is suited both for compressible and incompressible flows which has these advantages, we have chosen to apply this staggering. However, in this report we shall limit ourselves to compressible flows only.

In Chapter 2 the Navier-Stokes equations and the equations of state are discussed. In order to use these equations in (numerical) models these equations are made dimensionless. The nondimensionalization method used is only suitable for compressible fluids. In Section 2.7 boundary conditions are discussed for inviscid flows (Euler) and viscous flows (Navier-Stokes) in a channel.

Chapter 3 deals with the discretization of the dimensionless Navier-Stokes equations using finite volume method. In the first section this method is explained. Notations are introduced, the idea of a staggered grid is explained and the discretization of the equations is briefly discussed. The next sections are devoted to the discretization of the dimensionless equations. In Section 3.5 upwind discretization is discussed for the convection terms.

In Chapter 4 implementation is discussed. Since we only had a limited amount of time available, we had to make some restrictions with respect to the implementation. This is the subject of Section 4.1. The implementation embodies the fundamental arrays, which are an extension to the already existing datastructure, and the arrays for the matrices, which are needed by the solver already available. In numerical calculations one usually meets memory and time constraints. In order to keep these costs low the matrices are stored in a special way.

With the aid of this implementation the Euler equations are solved for several problems on different meshes. Chapter 5 gives a brief overview of the results. First the equations are tested separately and in the second part of the chapter we consider a more realistic problem with the coupled system of equations. The results are verified with the results obtained by Deft for the same problems. Deft is a structured finite volume package, in which quadrilaterals are mapped onto rectangles. The discretization and interpolations performed in Deft differ from the ones we will use.

The final chapter contains a summary about the activities performed and a small discussion about the results that were obtained. In order for the implementation to perform well on a larger scala of problems some improvements are needed. A few hints are given in this last chapter.

Chapter 2

The basic equations of fluid dynamics

In the first three sections of this chapter we will consider the basic equations of fluid dynamics. These equations, the Navier-Stokes equations, describe: the conservation of mass, the conservation of momentum and the conservation of energy. In order to have as many equations as unknowns one or more additional relations are required. These so-called equations of state are discussed in Section 2.4. In Section 2.5 the Euler equations are introduced. The Navier-Stokes equations are made dimensionless in Section 2.6. In order to get a unique solution for a given problem we also need to define initial conditions and boundary conditions. In Section 2.7 the boundary conditions are given for viscous and subsonic flows in a channel.

2.1 Conservation of mass

Consider a fixed volume V in the fluid enclosed by a surface S. Let $\mathbf{n}$ be the outward unit normal on the surface. Conservation of mass tells us that the change of mass in the volume V equals the netto flux through the surface S (we take for granted that there are no sources inside volume V). In mathematical form we get the following integral equation:

$$\frac{\partial M}{\partial t} = \frac{\partial}{\partial t} \int\int\int_V \rho dV = -\int\int_S \rho \mathbf{u} \cdot \mathbf{n} dS. \tag{2.1}$$

Here M is the total mass in volume V. Application of the Gauss-theorem to the right-hand side gives:

$$\frac{\partial}{\partial t} \int\int\int_V \frac{\partial \rho}{\partial t} dV = -\int\int\int_V \operatorname{div}(\rho \mathbf{u}) dV. \tag{2.2}$$

Since this has to be true for all V it is clear that:

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho \mathbf{u}) = 0, \tag{2.3}$$

This is also called the continuity equation.

2.2 Conservation of momentum

The law of conservation of momentum says that the change in momentum in a volume equals the total force in that volume. There are two kinds of forces: *body forces* ($\mathbf{F}_b$) and *surface forces* ($\mathbf{F}_s$). A body force is a force proportional to its mass. An example is gravitation force. A surface force is proportional to the area of the volume element, for example friction. The body force and surface force can be defined in the following way:

$$\mathbf{F}_b = \mathbf{f}^b \rho \, dV(t), \tag{2.4}$$

$$\mathbf{F}_s = \mathbf{f}^s dS(t). \tag{2.5}$$

Here $\mathbf{f}^b$ is the force per mass unit and $\mathbf{f}^s$ is the force per area unit. Applying the conservation of momentum to a material volume, where u_i are the velocities in each direction ($i = 1 \ldots 3$) yields:

$$\frac{d}{dt}\int_{V(t)} \rho u_i dV = \int_{V(t)} f_i^b \rho dV + \int_{S(t)} f_i^s dS, \tag{2.6}$$

For the body forces we only consider the gravitation force. In case of incompressible flows the gravitation force is incorporated in the pressure term. The only compressible flows to be considered in this report are gas flows, where the gravitation force can be neglected. Therefore we set $\mathbf{f}^b = \mathbf{0}$.

Now we use the transport theorem which says that a material volume $V(t)$ permanently consists of the same material particles:

$$\frac{d}{dt}\int_{V(t)} \phi dV = \int_{V(t)} (\frac{\partial \phi}{\partial t} + (\phi u_j)_{,j}) dV. \tag{2.7}$$

Taking $\phi = \rho u_i$ and substituting the right-hand side of (2.7) into the left-hand side of (2.6) gives:

$$\int_{V(t)} \{\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j}\} dV = \int_{S(t)} f_i^s dS. \tag{2.8}$$

The notation $_{,j}$ means differentiation with respect to x_j. The summation convention says that if an index appears twice, a summation over that index is intended. Hence:

$$u_i u_i = \mathbf{u} \cdot \mathbf{u}. \tag{2.9}$$

It can be shown that there are nine quantities σ_{ij} (the so called stress tensor) such that:

$$f_i^s = \sigma_{ij} n_j. \tag{2.10}$$

The vector $\mathbf{n}$ is the outward unit normal on the surface dS. Substitution of (2.10) into (2.8) yields:

$$\int_{V(t)} \{\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j}\} dV = \int_{S(t)} \sigma_{ij} n_j dS. \tag{2.11}$$

From the Gauss-theorem follows:

$$\int_{V(t)} \{\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j}\} dV = \int_{V(t)} \sigma_{ij,j} dV. \tag{2.12}$$

This equation is true for every volume $V(t)$ and therefore the integrals can be dropped:

$$\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j} = \sigma_{ij,j}. \tag{2.13}$$

Tensor σ_{ij} has to be related to the movement of the fluid. We can use the constitutive relations for that purpose, namely:

$$\sigma_{ij} = -p\delta_{ij} + 2\mu(\epsilon_{ij} - \frac{1}{3}\Delta\delta_{ij}). \tag{2.14}$$

Here p is the pressure, δ_{ij} is the Kronecker delta, μ is the dynamic viscosity and ϵ_{ij} the *rate of strain tensor*:

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}), \tag{2.15}$$

and:

$$\Delta = u_{i,i}. \tag{2.16}$$

The last expression represents the change in volume of V. If we substitute (2.14) into (2.13) we get:

$$\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j} = -p_{,i} + \{2\mu(\epsilon_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j}. \tag{2.17}$$

These equations are the momentum equations.

2.3 Conservation of energy

Again we consider a volume element V with a surface S. The element moves along with the fluid, and therefore there is no transport of mass through the walls of the volume element. The first law of thermodynamics says the following:

$$\frac{d}{dt}\int_{V(t)} \rho E dV = Q + W. \tag{2.18}$$

In words: the change in total energy equals the heat Q that is added and the work W exposed to that volume element. The total energy is equal to the internal energy and the kinetic energy:

$$E = e + \frac{1}{2}u_i u_i. \tag{2.19}$$

The body force does work at rate $\rho u_i f_i^b dV(t)$ and the surface force does work at rate $u_i f_i^s dS(t)$. This means for the total work:

$$W = \int_{V(t)} \rho u_i f_i^b dV + \int_{S(t)} u_i f_i^s dS. \tag{2.20}$$

Again we set $\mathbf{f}^b$ to zero. With the aid of equation (2.10) and the divergence theorem of Gauss this can be written as:

$$W = \int_{V(t)} (u_i \sigma_{ij})_{,j} dV. \tag{2.21}$$

Supposing that there is no heat production inside the volume, Q equals:

$$Q = \int_{S(t)} \tau dS. \tag{2.22}$$

Flux τ follows Fourier's law, i.e.:

$$\tau = kT_{,i} n_i, \tag{2.23}$$

where k is the thermal conductivity and T is the temperature. Substituting (2.23) into (2.22) and using the divergence theorem we find:

$$Q = \int_{V(t)} (kT_{,i})_{,i} dV. \tag{2.24}$$

Applying the transport theorem to (2.18) and substituting (2.21) and (2.24) into the right-hand side of (2.18) yields:

$$\int_{V(t)} \{\frac{\partial \rho E}{\partial t} + (\rho u_i E)_{,i}\} dV = \int_{V(t)} \{(u_i \sigma_{ij})_{,j} + (kT_{,i})_{,i}\} dV. \tag{2.25}$$

This is the energy equation in integral form. Since this equation is applicable for all volumes $V(t)$ we can deduce the differential form of the heat equation:

$$\frac{\partial \rho E}{\partial t} + (\rho u_i E)_{,i} = (u_i \sigma_{ij})_{,j} + (kT_{,i})_{,i}. \tag{2.26}$$

Inserting the constitutive relations (2.14) we get the following relation for the energy:

$$\frac{\partial \rho E}{\partial t} + (\rho u_i E)_{,i} = -(u_i p)_{,i} + \{2\mu u_i (e_{ij} - \frac{1}{3}\Delta \delta_{ij})\}_{,j} + (kT_{,i})_{,i}, \tag{2.27}$$

or:

$$\frac{\partial \rho E}{\partial t} + (\rho u_i H)_{,i} = \{2\mu u_i(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} + (kT_{,i})_{,i}, \tag{2.28}$$

where E is given by (2.19) and H is the total enthalpy, i.e.:

$$H = h + \frac{1}{2}u_i u_i. \tag{2.29}$$

Here h is the enthalpy.
The first term in the right-hand side of (2.28), the viscous term, is usually very small.

We now have 5 equations, ie. the continuity equation (2.3), the momentum equations, given by (2.17) and the energy equation, given by (2.28). However, we have 8 unknowns, namely: u_i, ρ, p, E, H and T. So we need 3 extra equations. These equations are called the equations of state. In the next section these will be introduced in order to complete the system.

2.4 Equations of state

A perfect gas is a gas for which the following statement holds:

$$p = \rho RT \tag{2.30}$$

This is the first equation of state, where R is a constant that is characteristic for the gas. Furthermore, in case of a calorically perfect gas, e (the internal energy per mass unit) only depends on the temperature T. This yields our second equation of state:

$$e = c_v T, \tag{2.31}$$

where c_v is called the specific heat at constant volume. The enthalpy is also exclusively a function of the temperature, namely:

$$h = c_p T, \tag{2.32}$$

where c_p is the specific heat at a constant pressure. This is our third equation of state. Furthermore:

$$h = e + p/\rho. \tag{2.33}$$

We then get the following relationship for h and e:

$$\frac{h}{e} = \gamma, \tag{2.34}$$

where:

$$\gamma = \frac{c_p}{c_v}. \tag{2.35}$$

For a perfect diatomic gas $\gamma = \frac{7}{5}$.

It follows immediately from (2.30), (2.31), (2.32) and (2.33) that:

$$R = c_p - c_v. \tag{2.36}$$

We have here a so-called simple system, i.e. there are at most two independent variables of state, on which the other variables of state depend. We choose our independent variables of state to be ρ and h. Together with the momentum $\mathbf{m}$ they are the primary variables. Since we will use density and enthalpy as primary variables (and not the temperature), equation (2.30) is rewritten as:

$$p = \frac{\gamma - 1}{\gamma}\rho h, \tag{2.37}$$

where (2.32), (2.34), (2.35) and (2.36) are used.

2.5 Euler equations

A fluid is called ideal if there is no friction nor diffusion/conduction, thus $\mu = k = 0$.

With $\mu = 0$ in (2.17) and $\mu = k = 0$ in (2.28), together with (2.3) and (2.37) using the definition of the total energy (2.19) and the total enthalpy (2.29) we get:

$$\frac{\partial \rho}{\partial t} + (\rho u_i)_{,i} = 0, \tag{2.38}$$

$$\frac{\partial \rho u_i}{\partial t} + (\rho u_i u_j)_{,j} + p_{,i} = 0, \tag{2.39}$$

$$\frac{\partial \rho E}{\partial t} + (\rho u_i H)_{,i} = 0, \tag{2.40}$$

$$p = \frac{\gamma - 1}{\gamma}\rho h, \tag{2.41}$$

where:

$$E = e + \frac{1}{2}u_i u_i, \quad H = h + \frac{1}{2}u_i u_i, \tag{2.42}$$

and:

$$h = e + p/\rho, \tag{2.43}$$

Equations (2.38) to (2.40) are called the Euler equations. In fluid mechanics it is very common to write the governing equations in dimensionless form. This is done in the next section.

2.6 Dimensionless form of the governing equations

In this section a dimensionless form of the equations will be discussed. We choose the following primitive variables: ρ (density), $\mathbf{m}$ (momentum), h (enthalpy). The velocity $\mathbf{u}$ equals: $\mathbf{m}/\rho$.

We can build a model with a small scale to simulate a model with a larger scale. By using the fact that laws of nature do not depend on physical units defined by convention, we can say something about the similarity between the two cases.

The governing equations can be made dimensionless by using units that are not conventional (such as the mks system) but are characteristic for the problem at hand. If we have a flow in a pipe then the characteristic unit for length would for example be the radius of the pipe. We will need four units for the length, density, enthalpy and momentum from which all other reference values will follow.

The four independent characteristic units are chosen as follows: L_r, ρ_r, h_r, m_r. This gives the following dimensionless quantities, indicated by a tilde:

$$\tilde{\mathbf{x}} = \frac{\mathbf{x}}{L_r}, \quad \tilde{\rho} = \frac{\rho}{\rho_r}, \quad \tilde{h} = \frac{h}{h_r}, \quad \tilde{\mathbf{m}} = \frac{\mathbf{m}}{m_r}. \tag{2.44}$$

Velocity and time can be scaled as:

$$\tilde{\mathbf{u}} = \frac{\mathbf{u}}{u_r}, \quad \tilde{t} = \frac{t}{t_r}, \tag{2.45}$$

with:

$$u_r = \frac{m_r}{\rho_r}, \quad t_r = \frac{L_r}{u_r} = \frac{L_r \rho_r}{m_r}. \tag{2.46}$$

In this report only compressible flows are considered. A flow is considered to be incompressible if:

$$M \ll 1, \tag{2.47}$$

where $M = \frac{u}{c}$ is the Mach number (the ratio between the velocity of the flow U and the speed of sound c). For the speed of sound we have the following important relationship:

$$c^2 = (\gamma - 1)h. \tag{2.48}$$

In practice one assumes a flow to be compressible if $M \geq 0.2$. As we will see, the Mach number will play an important role in the nondimensionalization process.

Mass conservation

For the mass conservation equation we have equation (2.3).
Using:

$$\frac{\partial}{\partial t} = \frac{m_r}{L_r \rho_r} \frac{\partial}{\partial \tilde{t}}, \quad \frac{\partial}{\partial x_i} = \frac{1}{L_r} \frac{\partial}{\partial \tilde{x}_i}, \tag{2.49}$$

we see that the scaled mass conservation equation is invariant with respect to the scaling.

The equation of state

Using (2.37) we can deduce the following expression for the reference pressure:

$$p_r = \frac{\gamma - 1}{\gamma} \rho_r h_r, \tag{2.50}$$

The dimensionless pressure is:

$$\tilde{p} = \frac{p}{p_r}. \tag{2.51}$$

The dimensionless equation of state becomes (tildes have been omitted):

$$p = \rho h. \tag{2.52}$$

Momentum equation

The momentum equation is given by (2.17). Inserting (2.44) and (2.48) in (2.50) leads to:

$$p_r = \gamma^{-1} \rho_r c_r^2, \tag{2.53}$$

where c_r represents the reference speed of sound. Using the scaling for the viscosity $\mu = \tilde{\mu}\mu_r$ we get (omitting the tildes):

$$\frac{\partial m_i}{\partial t} + (\rho^{-1} m_i m_j)_{,j} - \frac{\mu_r}{L_r m_r}\{2\mu(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} = -\frac{\rho_r^2 c_r^2}{\gamma m_r^2} p_{,i}. \qquad (2.54)$$

Using the definition of the *Mach number* and the *Reynolds number*:

$$M_r = \frac{m_r}{\rho_r c_r}, \quad Re_r = \frac{L_r m_r}{\mu_r}, \qquad (2.55)$$

we get:

$$\frac{\partial m_i}{\partial t} + (\rho^{-1} m_i m_j)_{,j} - \frac{1}{Re_r}\{2\mu(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} = -\frac{1}{M_r^2 \gamma} p_{,i}. \qquad (2.56)$$

Since both sides of (2.56) are finite, we see that a problem arises when $M_r \downarrow 0$. In order to keep the right-hand side of (2.56) of order unity, the pressure becomes independent of position $\mathbf{x}$ in the incompressible limit. This is not physical. However, since we restrict ourselves to compressible flows with $M_r \geq 0.2$, the singularity in $M_r \downarrow 0$ is avoided.

There is another dimensionless parameter that characterizes the flow, namely the Reynolds number. A very high Reynolds number indicates that convection dominates the flow, on the other hand if the Reynolds number is small we have a strongly viscous flow.

Energy equation

Expanding the terms of (2.28), using (2.32) and omitting the tildes yields:

$$\frac{m_r h_r}{L_r}\left[\gamma^{-1}\frac{\partial(\rho h)}{\partial t} + (m_i h)_{,i} + \frac{1}{2}\frac{m_r^2}{h_r \rho_r^2}\left\{\frac{\partial(\rho m_i m_i)}{\partial t} + (m_i u_j u_j)_{,i}\right\}\right]$$
$$-\frac{m_r h_r}{L_r}\left[\frac{k_r}{m_r L_r c_p}(k h_{,i})_{,i} + \frac{m_r \mu_r}{\rho_r^2 L_r h_r}\{2\mu u_i(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j}\right] = 0. \qquad (2.57)$$

It is common practice to introduce the dimensionless *Prandtl number*:

$$Pr = \frac{c_p \mu_r}{k_r}. \qquad (2.58)$$

The Prandtl number is the ratio between viscosity and heat conduction. Its value depends on the medium. For air it is approximately constant (≈ 0.72). Substituting (2.44), (2.48) and (2.58) into relation (2.57) and using the definition of M_r leads to:

$$\gamma^{-1}\frac{\partial(\rho h)}{\partial t} + \frac{1}{2}M_r^2(\gamma - 1)\left\{\frac{\partial(\rho^{-1} m_i m_i)}{\partial t} + (m_i m_j u_j)_{,i}\right\} + \qquad (2.59)$$
$$+(u_i h)_{,i} - (\gamma - 1)\frac{M_r^2}{Re_r}\{2\mu u_i(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} - \frac{1}{Re_r Pr}(k h_{,i})_{,i} = 0.$$

System of equations

The resulting system of dimensionless Navier-Stokes equations in primary variables is:

$$\frac{\partial \rho}{\partial t} + m_{i,i} = 0, \tag{2.60}$$

$$\frac{\partial m_i}{\partial t} + (u_i m_j)_{,i} + \frac{1}{Re_r}\{2\mu(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} = -\frac{1}{\gamma M_r^2} p_{,i}, \tag{2.61}$$

$$\gamma^{-1}\frac{\partial(\rho h)}{\partial t} + \frac{1}{2}M_r^2(\gamma-1)\Big\{\frac{\partial(\rho^{-1}m_i m_i)}{\partial t} + (\rho^{-2}m_i m_j m_j)_{,i}\Big\} \tag{2.62}$$
$$+(m_i h)_{,i} - (\gamma-1)\frac{M_r^2}{Re_r}\{2\mu u_i(e_{ij} - \frac{1}{3}\Delta\delta_{ij})\}_{,j} - \frac{1}{Re_r Pr}(kh_{,i})_{,i} = 0.$$

$$p = \rho h. \tag{2.63}$$

In order to have a unique solution of these equations it is necessary to have sufficient initial and boundary conditions. In the next section we shall consider the types of boundary conditions that are commonly used.

2.7 Boundary conditions

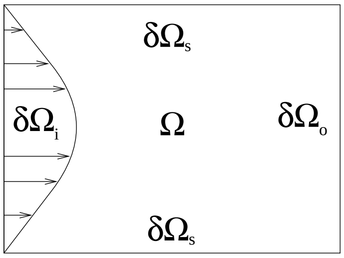

Figure 2.1: Boundaries for a channel

The Euler equations are given by (2.38) to (2.40). We have the independent variables $\mathbf{m}$, ρ and h. For a flow in a channel the situation will be the following: we have an inflow boundary, written as: $\partial\Omega_i$, two solid walls $\partial\Omega_s$ and the outflow boundary $\partial\Omega_o$. In order to get a unique solution we need boundary conditions. The Euler equations are hyperbolic equations. Therefore the number of boundary conditions depends on the number of incoming characteristics. For supersonic flows, where $M > 1$ other boundary conditions are needed than for subsonic flows, because the number of incoming characteristics is different. Only the subsonic case is considered in this report.

One can prove (see [3]) that the number of boundary conditions for a subsonic flow in two dimensions is given by: three conditions on the inlet and one on the outlet and for a supersonic flow by: four conditions on the inlet and none on the outlet. For subsonic flows in a channel the following choice is possible:

$$\begin{aligned} h_i(t) &= \mathbf{given}, \quad \mathbf{m}_i(t) = \mathbf{given}, \\ p_o(t) &= \mathbf{given}, \\ u_s^n &= \mathbf{u}_s \cdot \mathbf{n}_s = 0. \end{aligned} \tag{2.64}$$

We see clearly three boundary conditions on the inlet and one on the outlet.

The fourth condition implies that there is no flow through a solid wall. If we solve the Navier-Stokes equations, theoretically the situation is much more simple. The Navier-Stokes equations are parabolic or elliptic (in the stationary case). Hence on each boundary exactly two boundary conditions for the velocity and one for the enthalpy is required.

However: the flows we consider are convection dominant. As a consequence large parts of the flow behave more like Euler than like Navier-Stokes. The numerical scheme does not feel much influence of the relatively small viscosity and behaves as if the equations were hyperbolic. Especially for the outflow conditions this means that formally we have to prescribe 2 velocities, but in the case of Euler only one boundary for the momentum equations is required (the pressure). As a consequence prescribing the velocity of the outflow might result in unrealistic wiggles since a kind of artificial obstacle is introduced, unless we know the exact velocity corresponding to the numerical scheme. For that reason one always tries to use boundary conditions that disturb the flow as little as possible. For example at the outflow one sets:

$$\sigma^{nt} = 0, \quad \sigma^{nn} = -p, \tag{2.65}$$

where σ^{nt} and σ^{nn} are the components of the normal stress. For a flow with viscosity we also have the no-slip condition at the solid walls, i.e. the velocity vector equals zero:

$$\mathbf{u}_s = \mathbf{0}, \tag{2.66}$$

At the solid wall we need one more boundary condition. At the solid wall and the outflow boundary we give a homogeneous Neumann condition for the enthalpy in normal direction. This gives us the following boundary conditions for Navier-Stokes:

$$
\begin{aligned}
h_i(t) &= \textbf{given}, \quad \mathbf{m}_i(t) = \textbf{given}, \\
\left.\frac{\partial h}{\partial \mathbf{n}}\right|_o &= 0, \quad \sigma_o^{nt} = 0, \quad \sigma_o^{nn} = -p, && (2.67) \\
\mathbf{u}_s &= \mathbf{0}, \quad \left.\frac{\partial h}{\partial \mathbf{n}}\right|_s = 0. && (2.68)
\end{aligned}
$$

The compressible Navier-Stokes are in general much too complicated to be solved analytically. For that reason it is necessary to use a numerical method. In our case we have decided to discretize the equations by a finite volume method (FVM).

Chapter 3

Discretization of the Navier-Stokes equations

In order to discretize the differential equations the finite volume method (FVM) will be used. This will be described in Section 3.1. In Section 3.2 some notations are introduced. The FVM will be applied to the equations of motion. In Section 3.3 a short introduction is given. Sections 3.6, 3.7 and 3.8 deal with the discretization of respectively the momentum, continuity and enthalpy equation.

3.1 Finite volume method

The finite volume method is a general discretization method for partial differential equations that can be applied if the equations are written in conservative form; i.e. as a divergence of a vector. An example is the Poisson equation, written as:

$$\nabla \cdot \nabla u = f, \tag{3.1}$$

where u is the velocity in one dimension and f a function. In the finite volume method we start by subdividing the region into subregions, for example in $\Re^2$: triangles or quadrilaterals. In this report we will limit ourselves to quadrilaterals only.

Next we choose special points per subregion where we want to put the unknown quantities. The points are for example the vertices of the subregion or the centroids of the quadrilaterals.

Next we have to define the equations in order to be able to solve the unknowns.

An important step in the FVM is the choice of a control volume. This volume is chosen such that for each unknown we have a control volume.

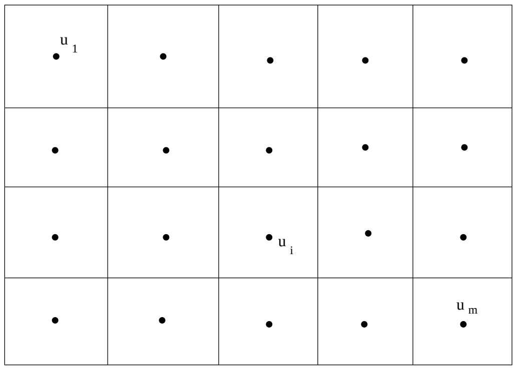

Figure 3.1: A structured grid with quadrilaterals

The unknowns are positioned in the centers of the control volumes. Therefore we take the control volume the same as the grid cell. After choosing the control volume the left- and right-hand side of equation (3.1) are integrated over the volumes Ω_i, with $i = 1 \dots m$ where m is the number of control volumes:

$$\int_{\Omega_i} \nabla \cdot \nabla u d\Omega = \int_{\Omega_i} f d\Omega. \tag{3.2}$$

Let the boundary of control volume Ω_i be $\partial\Omega_i$. Application of the Gauss theorem to the left-hand side gives:

$$\int_{\partial\Omega_i} \mathbf{n} \cdot \nabla u d\Gamma = \int_{\Omega_i} f d\Omega. \tag{3.3}$$

Here $\mathbf{n}$ is the outward unit normal on Γ. Let the faces $\partial\Omega_i$ consist of straight parts $\partial\Omega_{i_1}, \partial\Omega_{i_2} \dots \partial\Omega_{i_4}$ with lengths: $h_{i_1} h_{i_2} \dots h_{i_4}$. The integral in the left-hand side can be approximated by using the midpoint rule:

$$\int_{\partial\Omega_i} \mathbf{n} \cdot \nabla u \, d\Gamma \approx \sum_{k=1}^{4} \mathbf{n}_{i_k} \cdot \nabla u(\mathbf{x}_{i_k}) h_{i_k}, \tag{3.4}$$

where $\mathbf{x}_{i_k}$ is the midpoint of the face $\partial\Omega_{i_k}$. The right-hand side is approximated by a midpoint rule:

$$\int_{\Omega_i} f d\Omega \approx f(\mathbf{x}_i)\Omega_i, \tag{3.5}$$

where $\mathbf{x}_i$ is the center of cell i. This leads to the following equations for the FVM for all i:

$$\sum_{k=1}^{n} \mathbf{n}_{i_k} \cdot \nabla u(\mathbf{x}_{i_k}) h_{i_k} = f(\mathbf{x}_i)\Omega_i. \tag{3.6}$$

For two volumes lying next to each other we see that the outward normals $\mathbf{n}$ on the common face point in opposite directions. The contributions at the faces of the cells cancel out each other. The absolute contribution is the normal component of the total flux through the face.

$\nabla u \cdot \mathbf{n}$ is the normal derivative $\partial u / \partial \mathbf{n}$. This can be approximated by a numerical scheme, for example central differentiation.

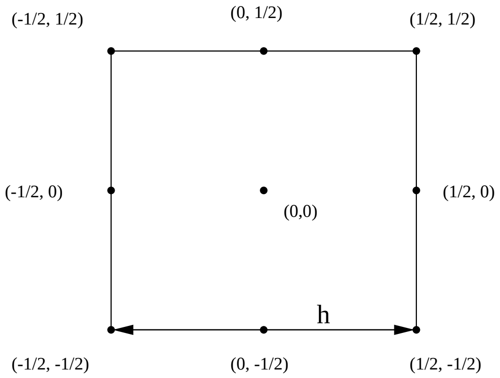

Figure 3.2: A primary cell

Application of this procedure to equation (3.1) to the case of square control volumes with $h_{i_k} = h$, see Figure 3.1, leads to the following discretization for internal cells:

$$h\left[\left(\frac{\partial u}{\partial \mathbf{n}}\right)_{0,1/2} + \left(\frac{\partial u}{\partial \mathbf{n}}\right)_{-1/2,0} + \left(\frac{\partial u}{\partial \mathbf{n}}\right)_{0,-1/2} + \left(\frac{\partial u}{\partial \mathbf{n}}\right)_{1/2,0}\right] = h^2 f_{0,0}. \quad (3.7)$$

In Cartesian coordinates we get:

$$h\left[\left(\frac{\partial u}{\partial y}\right)_{0,1/2} - \left(\frac{\partial u}{\partial x}\right)_{-1/2,0} - \left(\frac{\partial u}{\partial y}\right)_{0,-1/2} + \left(\frac{\partial u}{\partial x}\right)_{1/2,0}\right] = h^2 f_{0,0}. \quad (3.8)$$

Note that the unknowns are positioned at the coordinates (mh, nh) with m and n integers. Central differentiation leads to:

$$u_{0,1} + u_{-1,0} + u_{0,-1} + u_{1,0} - 4u_{0,0} = h^2 f_{0,0}. \quad (3.9)$$

Here we recognize the standard finite difference formula in Poisson's equation. Before we apply the finite volume method to the Navier-Stokes, a couple of notations will be introduced in the next section.

3.2 Notations

The cells, i.e. quadrilaterals, will be ordered in a grid. We consider Figure 3.3. Let the midpoint of face K defined by e_K and its length by l_K. The

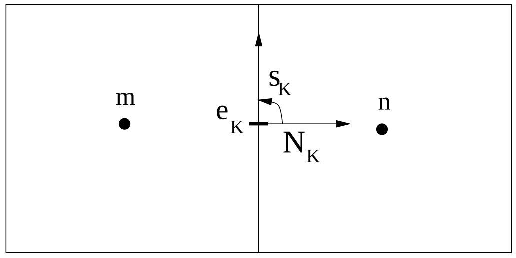

Figure 3.3: Notations

unique (unity) normal is done in the following way: $\mathbf{N}_K$ points from cell m to cell n if $m < n$, and points from cell n to cell m if $m > n$. This means that $\mathbf{N}_K$ always points to the cell with the higher number. The advantage of this choice is that it can be trivially extended to three dimensions. The tangential vector $\mathbf{s}_K$ is obtained by rotating $\mathbf{N}_K$ over $\pi/2$ counter-clockwise.

With these definitions we now will consider the discretization of the equations of motion: the momentum equation, the energy equation and the conservation of mass. In order to discretize the system of equations we have to pay attention to the following questions:

- Where will we place the unknowns?
- How is the time stepping performed?
- What discretization will be used and which interpolations are to be performed?

We will use the so-called staggered grid arrangement, which is explained in the next section.

3.3 Staggered grid arrangement

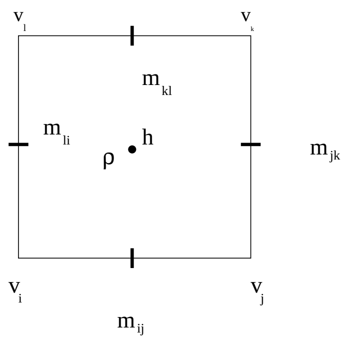

Figure 3.4: A cell of a staggered grid

For the discretization of the Euler and Navier-Stokes equations we shall use a so-called staggered grid arrangement, see Figure 3.4. That means that the scalar unknowns such as enthalpy, pressure and density will be positioned in the centers of the cells. The vector quantities, i.e. velocity $\mathbf{u}$ and $\mathbf{m}$ will be stored in a special way. On the midpoint of each face the normal component of this vector quantity is stored in the midpoint. Our primary variables are: $\mathbf{m}$, ρ, h.

3.4 Time stepping

For time discretization we will use Euler-backward. If we have the equation:

$$\frac{\partial u}{\partial t} = f(t, u(t)), \tag{3.10}$$

then Euler-backward says:

$$\frac{u^{n+1} - u^n}{\Delta t} = f(t^{n+1}, u^{n+1}). \tag{3.11}$$

Hence we have an implicit scheme. The advantage of implicit schemes is that they are more stable than explicit schemes, thus we can take larger time steps. Suppose that the function in the right-hand side is linear, then we have an equation that can be solved directly. In the case of a non-linear right-hand side we must linearize the equation and possibly iterate. The Navier-Stokes equations are non-linear so they have to be linearized first. Per time step we solve the equations of motion independently. The discretization of the equations is performed in the following order:

1. The momentum equation will be discretized with the aid of m^n, u^n, ρ^n and p^n. The new momentum m^{n+1} will be solved from the resulting system (Section 3.6),

2. The continuity equation is used to obtain ρ^{n+1} from m^{n+1} and ρ^n (Section 3.7),

3. The velocity u^{n+1} is calculated by $\frac{m^{n+1}}{\rho^{n+1}}$,

4. the new enthalpy h^{n+1} is solved from the energy equation (in Section 3.8), using h^n and p^n and the computed m^{n+1}, u^{n+1} and ρ^{n+1},

5. the new pressure follows directly from the new density and the enthalpy with aid of the equation of state, ie. $p^{n+1} = \rho^{n+1} h^{n+1}$.

In order to calculate the new momentum and enthalpy we have to build a linear system for each of them. For example the new enthalpy $\mathbf{h}^{n+1}$ is obtained from the following system:

$$\mathbf{A}_h \mathbf{h}^{n+1} = \mathbf{b}, \tag{3.12}$$

where $\mathbf{A}_h$ is the matrix for the enthalpy and $\mathbf{b}$ the right-hand side.

Time stepping is performed until a stationary solution has been reached. For example the enthalpy has reached a stationary solution when the following stopping criterion is met:

$$\|\mathbf{h}^{n+1} - \mathbf{h}^n\|_2 \leq \frac{1-\lambda}{\lambda}(\epsilon_{rel}\|\mathbf{h}^{n+1}\|_2 + \epsilon_{abs}), \tag{3.13}$$

where:

$$\lambda = \frac{\|\mathbf{h}^{n+1} - \mathbf{h}^n\|_2}{\|\mathbf{h}^n - \mathbf{h}^{n-1}\|_2}. \tag{3.14}$$

The two values ϵ_{rel} and ϵ_{abs} are respectively the relative and the absolute accuracy (see [5]).

The new momentum and enthalpy are calculated by using an implicit time integration, however, the calculation of the new density is performed an explicit time integration. Performing explicit calculations has the advantage that it is easier to implement, the disadvantage is that it is not possible to take the time step independent from the mesh width. The time step should meet a condition, namely the CFL-condition (see [4]), which guarantees that each grid point lies within the domain of dependence of the relevant grid points at the previous time level. For a one-dimensional problem we have the following requirement:

$$\frac{(u+c)\Delta t}{\Delta x} \leq 1, \tag{3.15}$$

where u is the velocity of the flow, c is the speed of sound, Δt the time step and for Δx is the smallest distance between two nodal points in the grid. In a two-dimensional grid with quadrilaterals we take for Δx the length of the smallest face.

Since all calculations are performed in dimensionless quantities, it is useful to write the CFL-condition in terms of dimensionless primary variables. With the aid of (2.48), (2.44) and (2.45) we arrive at:

$$\frac{(\rho^{-1}m + M_r^{-1}\sqrt{h})\Delta t}{\Delta x} \leq 1, \tag{3.16}$$

where ρ, m, h, Δt and Δx are dimensionless.

3.5 The upwind scheme and central differences

In the discretization of the equations of motion we are making use of two important discretization schemes, namely central differences and first order upwind discretization. In this section these schemes are explained briefly with the aid of an example.

Consider the following one dimensional differential equation:

$$u\frac{d\Phi}{dx} - \nu\frac{d^2\Phi}{dx^2} = 0, \tag{3.17}$$

with propriate boundary conditions. Here u is the velocity and ν is the viscosity. The first and second derivative for Φ in point i can be obtained by taking into account the values of both of its neighbors, Φ_{i-1} and Φ_{i+1}. This is called central differences. Discretization on a uniform grid with stepsizes h leads to:

$$u\frac{\Phi_{i+1} - \Phi_{i-1}}{2h} - \nu\frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{h^2} = 0, \tag{3.18}$$

It can be shown that this discretization may lead to spurious wiggles when the mesh-Peclet number $Pe = \frac{uh}{2\nu}$ is larger than one. Since we consider usually convection dominated flows, where $u \gg \nu$, the stepsize has to be unacceptably small to meet this no-oscillations criterion. It can be shown that the use of upwind interpolation for the convection term automatically prevents the pressure of spurious wiggles. First order upwind discretization leads to:

$$u\frac{d\Phi}{dx} \approx \begin{cases} \mathrm{u}\frac{\Phi_{i+1}-\Phi_i}{h} & \text{if } u < 0, \\ \mathrm{u}\frac{\Phi_i-\Phi_{i-1}}{h} & \text{if } u \geq 0. \end{cases} \tag{3.19}$$

Although there are much more sophisticated upwind interpolations than the first order upwind, this will be the only scheme to be discussed.

A useful interpretation of (3.19) is that only information coming from the flow direction (upwind) is taken into account. This idea can be applied to the momentum and enthalpy equation.

3.6 Discretization of the momentum equation

In this section we consider with the discretization of the momentum equations. After a short introduction the discretization of the terms will be given.

To derive an equation for the normal momentum component m_K on the face $\mathbf{e}_K$ we project equation (2.61) on the uniquely defined normal $\mathbf{N}_K$ (see Section 3.2) on that face, denoting $m_K = \mathbf{m}_K \cdot \mathbf{N}_K$. For the momentum m_K we take the control volume Ω_K, as is sketched in Figure 3.5, with boundary

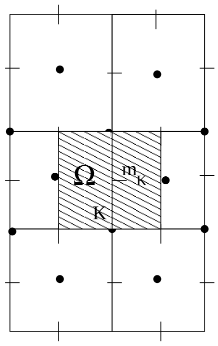

Figure 3.5: A control volume for the momentum

$\partial\Omega_K$. Integrating (2.61) over Ω_k gives:

$$\int_{\Omega_k} \frac{\partial m^\alpha}{\partial t} d\Omega + \int_{\Omega_k} (u^\beta m^\alpha)_{,\beta} d\Omega - \frac{1}{Re_r} \int_{\Omega_k} t^{\alpha\beta}_{,\beta} d\Omega = -\frac{1}{M_r^2\gamma} \int_{\Omega_k} p_{,\alpha} d\Omega, \tag{3.20}$$

with $t^{\alpha\beta} = 2\mu(e^{\alpha\beta} - \frac{1}{3}\Delta\delta^{\alpha\beta})$. Here we use Greek superscripts for the components and Greek subscripts for the partial derivatives. For example:

$$t^{\alpha\beta}_{,\beta} = t_{ij,j}. \tag{3.21}$$

Application of the Gauss theorem gives:

$$\int_{\Omega_k} \frac{\partial m^\alpha}{\partial t} d\Omega + \int_{\partial\Omega_k} u^\beta m^\alpha n^\beta d\Gamma - \frac{1}{Re_r} \int_{\partial\Omega_k} t^{\alpha\beta} n^\beta d\Gamma = -\frac{1}{M_r^2\gamma} \int_{\Omega_k} p_{,\alpha} d\Omega, \tag{3.22}$$

where n^β is the outward unit normal of Ω_K. Taking the inner product with the uniquely defined normal N_K^α gives:

$$\int_{\Omega_k} \frac{\partial m^\alpha}{\partial t} N_K^\alpha d\Omega + \int_{\partial\Omega_k} u^\beta m^\alpha n^\beta N_K^\alpha d\Gamma - \frac{1}{Re_r} \int_{\partial\Omega_k} t^{\alpha\beta} n^\beta N_K^\alpha d\Gamma = -\frac{1}{M_r^2\gamma} \int_{\Omega_k} p_{,\alpha} N_K^\alpha d\Omega. \tag{3.23}$$

The contour integrals are taken counterclockwise. The system consists of four terms, which will be discussed in the following subsections.

3.6.1 The inertia term

The first term on the left-hand side is the inertia term and is approximated as:

$$\frac{d}{dt} \int_{\Omega_K} m^\alpha N_K^\alpha d\Omega \approx \Omega_K \frac{dm_K}{dt}. \tag{3.24}$$

The time derivative is approximated in the following way:

$$\frac{dm_K}{dt} \approx \frac{m_K^{n+1} - m_K^n}{\Delta t}, \tag{3.25}$$

where Δt is the time step.

3.6.2 The pressure term

The pressure term is given by:

$$-\frac{1}{\gamma M_r^2} \int_{\Omega_K} N_K^\alpha p_{,\alpha} d\Omega = -\frac{1}{\gamma M_r^2} \int_{\Omega_K} \nabla p \cdot \mathbf{N}_K d\Omega. \tag{3.26}$$

This term can be evaluated by the so-called path-integral formulation, see Figure 3.6. The basic idea is that first an approximation for the gradient ∇p is made. Then the inner product with respect to $\mathbf{N}_K$ is taken. The path-integral method is based on the following a midpoint rule, namely:

$$\int_{\mathbf{x}_2}^{\mathbf{x}_1} \nabla p \cdot d\mathbf{s} \approx \nabla p_K \cdot (\mathbf{x}_2 - \mathbf{x}_1). \tag{3.27}$$

Here ∇p_K is an approximation of the pressure gradient at $\mathbf{e}_K$. The unknowns are the elements for the gradient of p_K. In two dimensions we need one more equation to solve for ∇p_K.

We can also take the path from p_3 to p_6 or from p_4 to p_5, provided the path crosses the face by $\mathbf{e}_K$ yielding:

$$p_6 - p_3 \approx \nabla p_K \cdot (\mathbf{x}_6 - \mathbf{x}_3). \tag{3.28}$$

and:

$$p_5 - p_4 \approx \nabla p_K \cdot (\mathbf{x}_5 - \mathbf{x}_4). \tag{3.29}$$

From Figure 3.6 it is clear that there is no reason to prefer one of them, hence we take the average of both. It is clear that the common contribution of the path from p_1 to p_2 cancels. In conclusion: the gradient ∇p_K is solved from the following system:

$$\begin{aligned} p_2 - p_1 &\approx \nabla p_K \cdot (\mathbf{x}_2 - \mathbf{x}_1), \\ p_6 + p_5 - p_4 - p_3 &\approx \nabla p_K \cdot (\mathbf{x}_6 + \mathbf{x}_5 - \mathbf{x}_4 - \mathbf{x}_3). \end{aligned} \tag{3.30}$$

By using the midpoint rule we find for the second term in the right-hand side:

$$-\frac{1}{\gamma M_r^2} \int_{\Omega_K} \nabla p \cdot \mathbf{N}_K d\Omega \approx -\frac{1}{\gamma M_r^2} \Omega_K \nabla p_K \cdot \mathbf{N}_K. \tag{3.31}$$

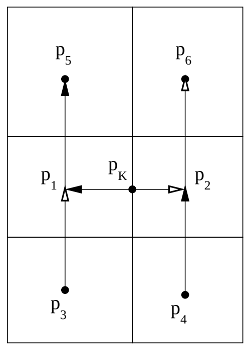

Figure 3.6: The paths from p_3 to p_6 and from p_4 to p_5

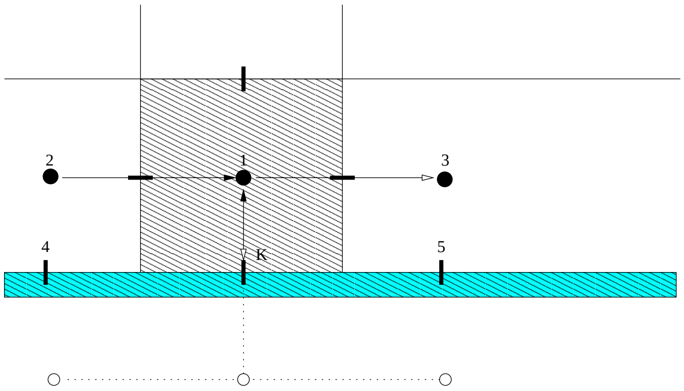

Figure 3.7: Paths on the boundary

We now consider the situation in the neighborhood of the boundaries, where the pressure is not given.
For cells with at least one boundary face complications are met in the path integral formulation (see Figure 3.7). Let us consider the pressure in K: we do not have the pressure on the other side of the boundary (marked by open dots). We take the path from 1 to 3, but for the other direction we use one-sided differences.
The same problem is found for the pressure gradient in point K in Figure 3.8. We take the paths from 1 to 3, from 2 to 4 (one-sided differences) and from 1 to 2. The same method is applied to the pressure gradient in K'. Points 5 and 6 are not used for the pressure term.

3.6.3 The convection term

For the convection term the following discretization will be used (See Figure 3.10):

$$\int_{\partial\Omega_{11}} (m^\alpha N_{11}^\alpha)(u^\beta n^\beta) d\Gamma \approx \sum_e \tilde{l}_e u_e (\mathbf{m}_e \cdot \mathbf{N}_{11}),$$

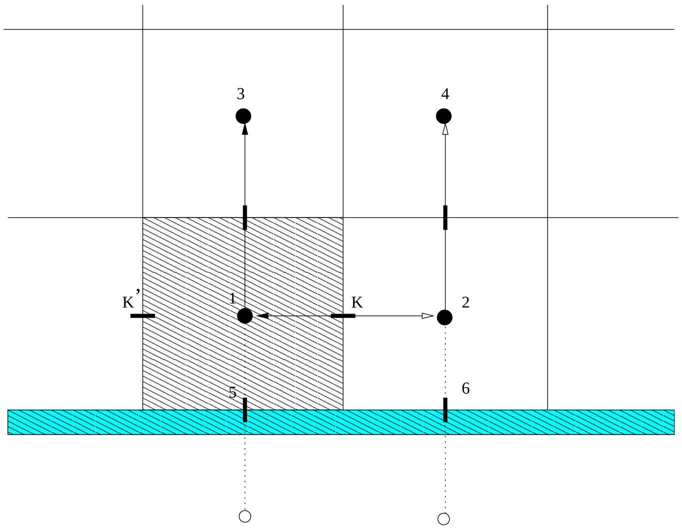

Figure 3.8: Paths on the boundary

where e stands for the integration points on the ad hoc defined faces $12, 13, 14$ and 15 of the control volume Ω_{11} and $\tilde{l}_e = (\mathbf{n}_e \cdot \mathbf{N}_e) l_e$.

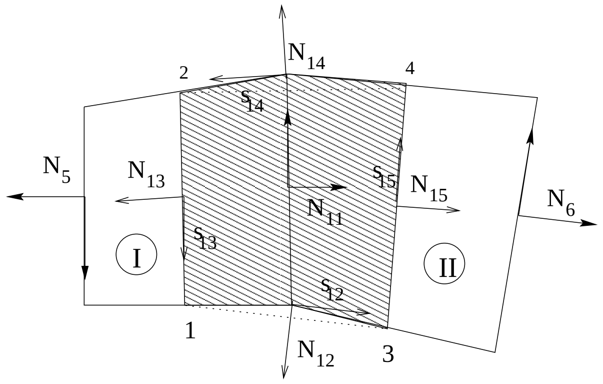

Figure 3.9: The normals on the control volume faces

We mark that the normals $\mathbf{N}_e$ are situated on vertices or cell centers instead of the midpoints of the faces. Consequently they cannot be defined as is described in Section 3.2. In order to define the normals we take the tangential vectors of faces 12 to 15 counter-clockwise (see Figure 3.9). The normals $\mathbf{N}_{12}, \mathbf{N}_{13}, \mathbf{N}_{14}$ and $\mathbf{N}_{15}$ are then obtained by rotating the tangential vectors $\mathbf{s}_{12}$, $\mathbf{s}_{13}$, $\mathbf{s}_{14}$ and $\mathbf{s}_{15}$ over $-\pi/2$ and normalization. Mark that they are always pointing outward, so $\mathbf{n}_e \cdot \mathbf{N}_e = 1$ always. This leads to the following

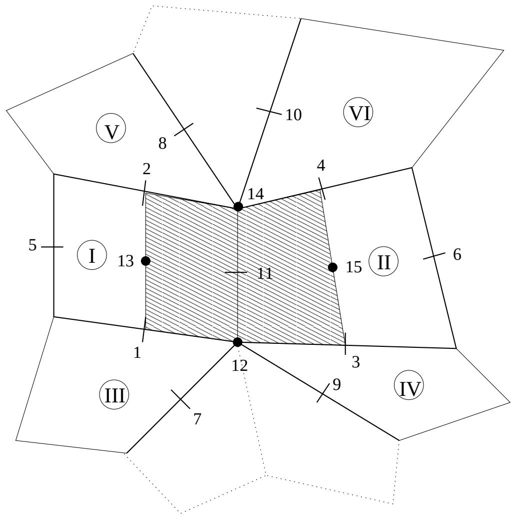

Figure 3.10: The stencil for the momentum

simplification for the convection term:

$$\sum_e u_e^n(\mathbf{m}_e^{n+1} \cdot \mathbf{N}_{11})\tilde{l}_e = \sum_e u_e^n(\mathbf{m}_e^{n+1} \cdot \mathbf{N}_{11})l_e. \tag{3.32}$$

Another problem we are faced with is that the momentum and the velocity are not available in the integration points 12 to 15. Hence we must use an interpolation. An extra complicating factor is that both quantities are vectors and we have only one of the components available per point. For the momentum vector $\mathbf{m}$ we write $\mathbf{m} = (m, \tilde{m})$. Here m represents the normal momentum component and $\tilde{m}$ the tangential component. Note that the values for m inside the control volumes are the unknowns. Momentum components m and $\tilde{m}$ on the faces follow from a reconstruction procedure, in which we want to express $\mathbf{m}$ in neighboring unknowns.

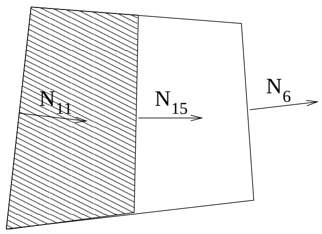

Figure 3.11: situation 1

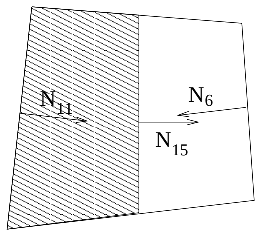

Figure 3.12: situation 2

We have two types of situations:

1. Points 13 and 15 are situated in the centers of respectively cell I and cell II.

2. Points 12 and 14 are situated in vertices.

These two cases require different treatments, which will be discussed subsequently.

1. In integration points 13 and 15 we calculate the normal momentum component by taking an interpolation of neighboring unknowns. Let us consider the momentum in 15. A simple approximation is to assume that the sides of the quadrilaterals are more or less parallel. An easy approximation for m_{15} is to take the average of m_6 and m_{11}. However, since we are dealing with non-scalars we have to take into account the direction of the normal in the contribution with respect to the normal in the integration point. This is evaluated by taking the inner product of the two normals. For example, $\mathbf{N}_6 \cdot \mathbf{N}_{15} > 0$ if $\mathbf{N}_6$ points approximately in the same direction as $\mathbf{N}_{15}$ (see Figure 3.11) and $\mathbf{N}_6 \cdot \mathbf{N}_{15} < 0$ if it points in approximately the opposite direction (see Figure 3.12). If the direction of $\mathbf{N}_6$ is approximately the same as the direction of $\mathbf{N}_{15}$ then the contribution is added, otherwise it is subtracted. The same reasoning is applied for m_{11}. We then get the following interpolation for m_{15}:

$$m_{15} = \frac{1}{2}\sigma(\mathbf{N}_{11} \cdot \mathbf{N}_{15})m_{11} + \frac{1}{2}\sigma(\mathbf{N}_6 \cdot \mathbf{N}_{15})m_6, \tag{3.33}$$

   where:

$$\sigma(\mathbf{N}_6 \cdot \mathbf{N}_{15}) = \mathrm{sign}(\mathbf{N}_6 \cdot \mathbf{N}_{15}). \tag{3.34}$$

   This is called central differences (See section 3.5) First order upwind interpolation leads to the following:

$$m_{15} = \begin{cases} \sigma(\mathbf{N}_{11} \cdot \mathbf{N}_{15})m_{11}, & u_{15}^n > 0, \\ \sigma(\mathbf{N}_6 \cdot \mathbf{N}_{15})m_6, & u_{15}^n < 0. \end{cases} \tag{3.35}$$

   For the tangential component we should consider the directions of the normals $\mathbf{N}_3$, $\mathbf{N}_4$ and the tangential vector $\mathbf{s}_{15}$ (see Figure 3.13). We then find:

$$\tilde{m}_{15} = \frac{1}{2}\sigma(\mathbf{N}_3 \cdot \mathbf{s}_{15})m_3 + \frac{1}{2}\sigma(\mathbf{N}_4 \cdot \mathbf{s}_{15})m_4, \tag{3.36}$$

   The momenta m_{13} and $\tilde{m}_{13}$ are evaluated in the same way.

2. The normal momentum component m_{12} is evaluated by means of an interpolation of m_1 and m_3:

$$m_{12} = \frac{1}{2}\sigma(\mathbf{N}_1 \cdot \mathbf{N}_{12})m_1 + \frac{1}{2}\sigma(\mathbf{N}_3 \cdot \mathbf{N}_{12})m_3. \tag{3.37}$$

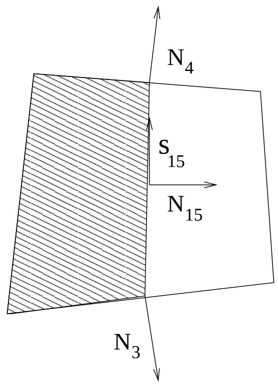

Figure 3.13: interpolation of the tangential momentum component in 15

For the tangential momentum component $\tilde{m}_{12}$ we are faced with the problem that we might have more than four cells round integration point 12. The choice is more or less arbitrary, but in practice it is easiest to make an interpolation with the normal momentum components in 7, 9 and 11. For the tangential component $\tilde{m}_{12}$ the normal momentum component m_{11} gives a contribution and the other contribution is the average of momenta m_7 and m_9, so:

$$\begin{aligned} \tilde{m}_{12} &= \frac{1}{2}\sigma(\mathbf{s}_{12} \cdot \mathbf{N}_{11})m_{11} + \\ &+ \frac{\sigma(\mathbf{s}_{12} \cdot \mathbf{N}_7)m_7 + \sigma(\mathbf{s}_{12} \cdot \mathbf{N}_9)m_9}{4}. \end{aligned} \tag{3.38}$$

No upwind discretization is applied to points 12 and 14 in both directions, only central differences. If the grid is structured around integration point 12 then both numbers 7 and 9 refer to the same face and consequently the contribution in these points are added to the same matrix element. When cells III and IV refer to the same cell it means that only three cells are situated round vertex 12 and consequently faces 7 and 9 do not exist. In this situation it is easiest to apply one-sided differences in order to obtain $\tilde{m}_{12}$. The momentum components m_{14} and $\tilde{m}_{14}$ are calculated in a similar way.

In the neighborhood of a boundary it can occur that one or more faces in the stencil are situated on the boundary and consequently cells III, IV, V or VI in Figure 3.10 are missing. We then encounter complications in integration points 12 and 14, however the treatment of integration points 13 and 15 remains unchanged. Let us consider integration point 12.

We should consider two situations:

1. either cell III or IV cell is missing,

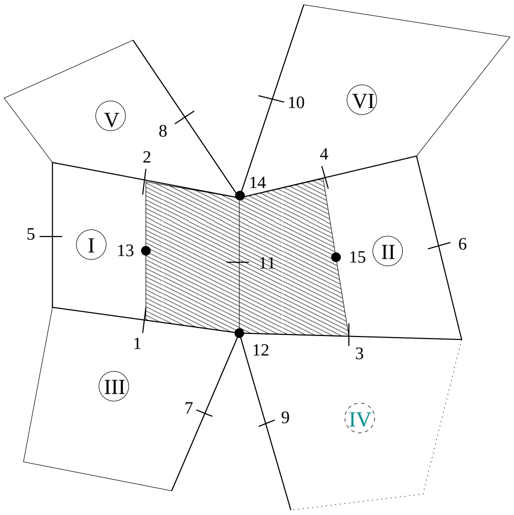

Figure 3.14: A stencil with one missing cell

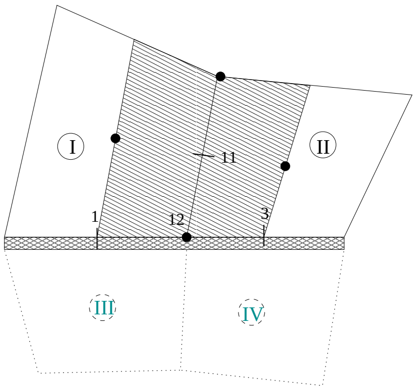

Figure 3.15: Two missing cells at the boundary

2. both cells III and IV are missing.

We first consider the case where only cell IV is missing (see again Figure 3.14). The normal momentum component, m_{12}, is calculated as in (3.37). For the tangential component, $\tilde{m}_{12}$, however, we only let the normal momentum components m_{11} and m_7 contribute in the interpolation given by (3.38), leading to:

$$\tilde{m}_{12} = \frac{1}{2}\sigma(\mathbf{s}_{12}\cdot\mathbf{N}_{11})m_{11} + \frac{1}{2}\sigma(\mathbf{s}_{12}\cdot\mathbf{N}_7)m_7. \quad (3.39)$$

In the second situation both cells III and IV, are missing (see Figure 3.15). Again m_{12} follows from the momenta m_1 and m_3 by (3.37). For the tangential component $\tilde{m}_{12}$ we then have two types of situations:

1. the momentum $\tilde{m}_{12}$ is given.

2. the tangential momentum component in 12 is *not* given and follows from m_{11} (one-sided difference):

$$\tilde{m}_{12} = \sigma(\mathbf{s}_{12} \cdot \mathbf{N}_{11}) m_{11}. \qquad (3.40)$$

At the boundary we have two typical situations (see Section 2.7):

- The momentum vector is given,
- the momentum vector is not given.

In the first and situation nothing has to be done in order to calculate the normal momentum component.

In the second situation the normal component of the momentum is not given. We then need to discretize the momentum equation in order to calculate the momentum at the boundary face.

The control volume is now one half of cell I (see Figure 3.16). A problem we are faced with is that integration points 8 and 9 are neither situated on vertices nor on positions where we have unknowns. This makes the interpolations differently from the ones which are usually performed. In order to simplify the calculation of the convection term and the pressure term points 8 and 9 are moved. We have the possibility to move 8 and 9 to either positions 1 and 2 or 8^* and 9^*. It is more practical to move the integration points to positions 8^* and 9^*, since in this case no enlargement of the stencil for convection is needed. The subscript * is omitted in the rest of this section.

The normal component m_6 is an unknown in the stencil and therefore no interpolations have to be performed. The momentum components in integration point 7 are interpolated as in the usual way. However, since the integration points 8 and 9 are moved to the boundary, a different treatment is required for $\mathbf{m}_8$ and $\mathbf{m}_9$.

For $\mathbf{m}_8$ the following calculations are performed:

- the normal momentum component m_8 is calculated by a one-sided difference:

$$m_8 = \sigma(\mathbf{N}_8 \cdot \mathbf{N}_1) m_1. \qquad (3.41)$$

- the tangential component follows from m_4 and m_6 (central differences):

$$\tilde{m}_8 = \frac{1}{2}\sigma(\mathbf{s}_8 \cdot \mathbf{N}_4) m_4 + \frac{1}{2}\sigma(\mathbf{s}_8 \cdot \mathbf{N}_6) m_6. \qquad (3.42)$$

  If cell II is non-existent then we also take a one-sided difference for $\tilde{m}_8$:

$$\tilde{m}_8 = \sigma(\mathbf{s}_8 \cdot \mathbf{N}_6) m_6. \qquad (3.43)$$

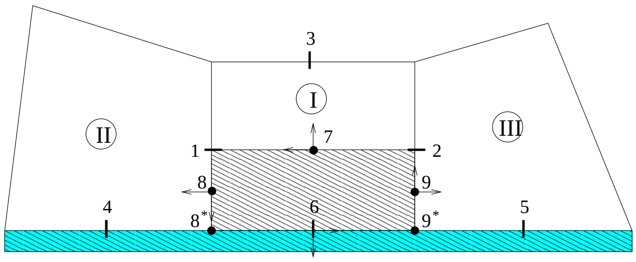

Figure 3.16: Two missing cells at the boundary

For the momentum components in integration point 9 we have the same types of interpolations.

3.6.4 The stress tensor

The stress tensor consists of partial derivatives of velocities, positioned at the faces of the cells. We use the control volume defined in Figure 3.5. The contour integral over the stress tensor will be approximated in the following way:

$$\begin{aligned} & -\frac{1}{Re_r}\int_{\partial\Omega_k} t^{\alpha\beta} n^\beta N_K^\alpha d\Gamma \approx \\ \approx \quad & -\frac{1}{Re_r}\sum_e \mu_e \left[\{(u^\alpha_{,\beta})_e + (u^\beta_{,\alpha})_e\} N_K^\alpha n_e^\beta - \frac{2}{3}(u^\alpha_{,\alpha})_e (N_K^\beta n_e^\beta)\right] l_e \quad (3.44) \end{aligned}$$

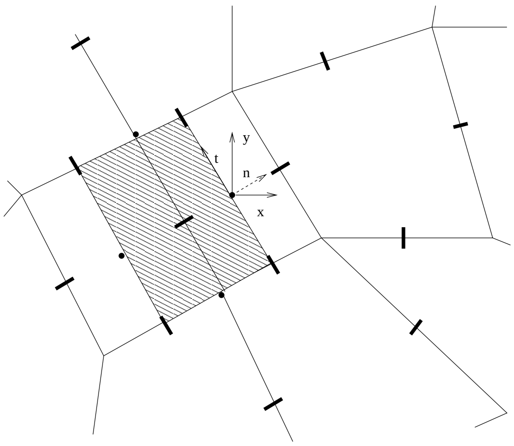

Figure 3.17: The Cartesian components of the velocity and normal components at the faces.

In formula (3.44) μ_e is the viscosity on face e of the control volume. In general it depends on the local properties of the fluid. The problem is how to evaluate the $t^{\alpha\beta}$ at the faces of the control volume. We are working with normal components of the velocities and momenta, but if one takes a gradient then one takes the partial derivatives with respect to the Cartesian system. Hence we have x and y components instead of normal and tangential components. Generally these components do not point into the same direction as the normal and tangential components. By adding the first component of the gradient of the first velocity component and the second component of the second velocity component we find the divergence $u^{\alpha}_{,\alpha}$.

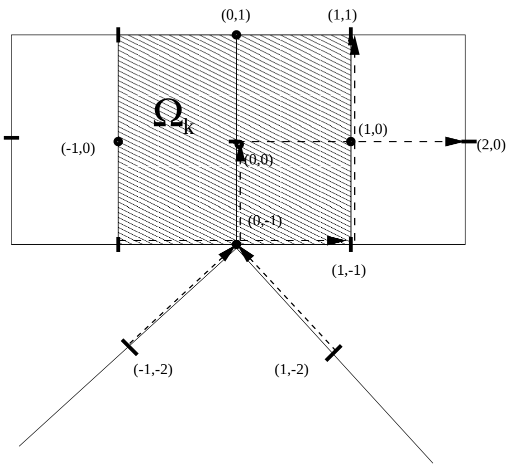

Figure 3.18: Two paths for the gradient of **u** in (1,0)

We take the faces as is illustrated in Figure 3.18. With the path integral formulation the derivatives of the vector **u** in for example (1,0) can be expressed in neighboring u's by taking the path from (0,0) to (2,0) and from (1,-1) to (1,1), leading to respectively:

$$\mathbf{u}_{(2,0)} - \mathbf{u}_{(0,0)} = \int_{(0,0)}^{(2,0)} \nabla \mathbf{u}_{(1,0)} \cdot d\mathbf{x} \approx \nabla \mathbf{u}_{(1,0)} \cdot (\mathbf{x}_{(2,0)} - \mathbf{x}_{(0,0)}) \qquad (3.45)$$

and:

$$\mathbf{u}_{(1,1)} - \mathbf{u}_{(1,-1)} = \int_{(1,-1)}^{(1,1)} \nabla \mathbf{u}_{(1,0)} \cdot d\mathbf{x} \approx \nabla \mathbf{u}_{(1,0)} \cdot (\mathbf{x}_{(1,1)} - \mathbf{x}_{(1,-1)}) \quad (3.46)$$

For the velocity gradient in integration point (0,-1) we take the path from (-1,-1) to (1-1), and the average of the path from (-1,-2) to (0,0) and the path from (1,-2) to (0,0).

Hence we have two equations and per unknown component of **u** two unknown elements of the gradient. We now have the gradient of **u** expressed in surrounding **u**.

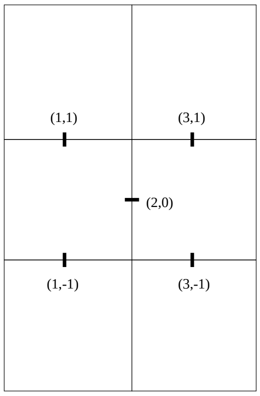

Figure 3.19: Interpolation of the velocity

The problem is that we do not have $\mathbf{u}$, only a normal component per momentum as unknown. The velocity $\mathbf{u}$ can be expressed in the normal and tangential components. The tangential components can be reconstructed from surrounding normal components with the same method as in the previous section, i.e. by averaging the neighbors. We take $\mathbf{u}_{(2,0)}$ as an example. The velocities on (2,-1) and (2,1) are not given. For that reason we average the normal components of four velocities closest to it, thus:

$$\begin{aligned} \tilde{u}_{(2,0)} &\approx \frac{1}{4}\{(\mathbf{s}_{(2,0)} \cdot \mathbf{N}_{(1,1)})u_{(1,1)} + (\mathbf{s}_{(2,0)} \cdot \mathbf{N}_{(1,-1)})u_{(1,-1)} \\ &+ (\mathbf{s}_{(2,0)} \cdot \mathbf{N}_{(3,1)})u_{(3,1)} + (\mathbf{s}_{(2,0)} \cdot \mathbf{N}_{(3,-1)})u_{(3,-1)}\}. \end{aligned} \tag{3.47}$$

This will only work properly for quadrilaterals with parallel faces as remarked before. Instead of the normal and the tangential component we

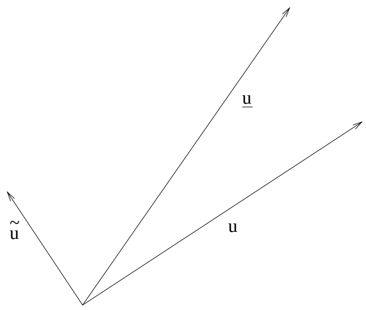

Figure 3.20: The normal and tangential component of $\mathbf{u}$

want to have the x component and y component of $\mathbf{u}$. In the situation of Figure 3.20 the x component and y component of $\mathbf{u}$, written as u_x and u_y, are calculated from the normal and the tangential component of the momentum. Both the unity normal $\mathbf{N}$ and the unity tangential vector $\mathbf{s}$ consist of x and y components:

$$\mathbf{N} = (N_x, N_y), \quad \mathbf{s} = (s_x, s_y). \tag{3.48}$$

The velocity $\mathbf{u}$ is now written as:

$$\mathbf{u} = (u_x, u_y) = u\mathbf{N} + \tilde{u}\mathbf{s}. \tag{3.49}$$

This yields the following for the x and y components:

$$u_x = uN_x + \tilde{u}s_x, \tag{3.50}$$
$$u_y = uN_y + \tilde{u}s_y. \tag{3.51}$$

The other gradients at the boundary of the control volume Ω_K are dealt with in the same way. However: we do not have the velocities as primary

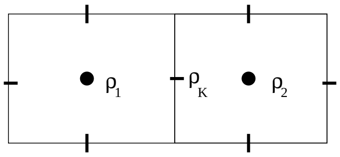

Figure 3.21: Interpolation of the density

variables, only the normal components of the momenta. The new velocity u_K can be expressed as $\frac{m_K}{\rho_K}$. We do not know the new density ρ_K, but we can linearize around the old density, ρ_K^n. The density on the faces is calculated by averaging the pressures in the centers of the primary cells, for example as in Figure 3.21:

$$\rho_K^n \approx \frac{\rho_1^n + \rho_2^n}{2} \tag{3.52}$$

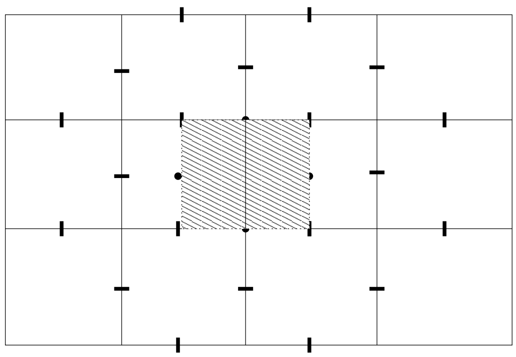

Figure 3.22: A stencil with 21 unknowns

Hence we get a stencil with 21 unknown momenta in total, marked by a stroke in Figure 3.22. This means that we have 21 non-zero elements per row, while for Euler flows, in which the viscosity is absent, there are only 9 non-zeroes on a structured grid. This means that one needs more memory to store the matrix.

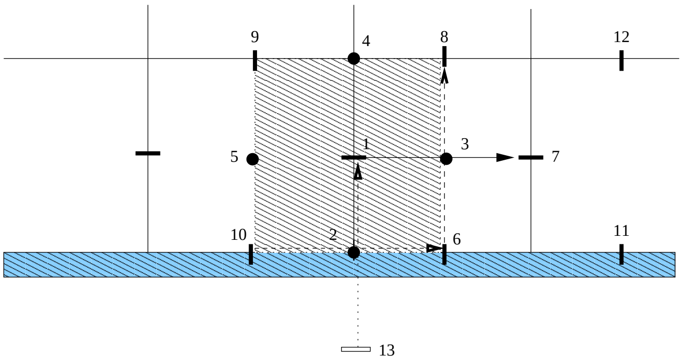

Figure 3.23: Paths at the boundary

In the neighborhood of the boundaries we meet the same problems for the path integral method as is sketched for the pressure term.

Take for example the momentum $\mathbf{m}_1$ in Figure 3.23. The path from 13 to 1 would cross the boundary. We do not have any interpolation points outside the grid. Therefore we take one-sided difference here, in other words, we take the path from 2 to 1. The second path goes from 10 to 6 where the normal component of the momenta in these points are prescribed. For the momenta in the points 3, 4 and 5 we do not meet this complication. In 3 for example we can safely take the path from 6 to 8 and from 1 to 7, with $\mathbf{m}_6$ given. The tangential momentum component $\tilde{m}_7$ is calculated by:

$$\tilde{m}_7 = \frac{m_6 + m_8 + m_{11} + m_{12}}{2}, \tag{3.53}$$

where both m_6 and m_{11} are given.

On the outflow the normal stress is prescribed instead of the momentum. This can be a good motivation for working with contour integrals: For face 6 in Figure 3.16 equation 3.23 can be rewritten as:

$$\int_{\Omega_6} \frac{\partial m^\alpha}{\partial t} N_6^\alpha d\Omega + \int_{\partial\Omega_6} u^\alpha m^\beta n^\alpha N_6^\beta d\Gamma = \frac{1}{Re_r} \int_{\partial\Omega_6} \sigma^{\alpha\beta} n^\beta N_6^\alpha d\Gamma.$$

For the term in the right hand side we use the following discretization:

$$\int_{\partial\Omega_6} \sigma^{\alpha\beta} n^\beta N_6^\alpha d\Gamma \approx \sum_e \sigma_e^{\alpha\beta} n_e^\beta N_6^\alpha l_e, \tag{3.54}$$

where e represents integration points 6 to 9. Integration points 8 and 9 are moved to the boundary. Since $\sigma_6^{nn} = \sigma_6^{\alpha\beta} n_6^{\alpha} n_6^{\beta}$ we see that σ_6^{nn} is given. The same holds for σ_8^{nn} and σ_9^{nn}. The stress σ^{nn} in integration point 7 follows from σ_6^{nn} and σ_3^{nn}.

The numerical stencil of this contour integral is smaller than the stencil of the path integral method and as we have seen in the beginning of this section the numerical treatment of the boundary conditions will be more complicated if we work with path integrals.

3.7 Discretization of the mass conservation equation

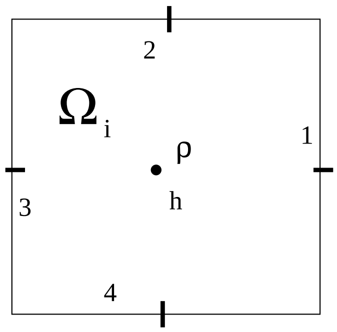

Figure 3.24: A control volume for the continuity equation

In this section we deal with the discretization of the continuity equation or mass conservation equation. The control volume Ω_i for the continuity equation is the primary cell i with ρ in the center and the momenta m_e on the faces e. In Figure 3.24 $e \in \{1, 2, 3, 4\}$. Integrating over the control volume Ω_i yields:

$$\int_{\Omega_i} \frac{\partial \rho}{\partial t} d\Omega + \int_{\Omega_i} m^{\alpha}_{,\alpha} d\Omega = 0. \tag{3.55}$$

Applying the Gauss-theorem to the second term yields:

$$\int_{\Omega_i} \frac{\partial \rho}{\partial t} d\Omega + \int_{\partial\Omega_i} m^{\alpha} n^{\alpha} d\Gamma = 0. \tag{3.56}$$

Using the midpoint-rule on both terms gives:

$$\Omega_i \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + \sum_e m_e^{n+1} l_e N_e^{\alpha} n_e^{\alpha} = 0, \tag{3.57}$$

From this equation we find the new ρ which will be used in the enthalpy equation.

3.8 Discretization of the energy equation

In this section the discretization of the energy equation will be discussed. The control volume for the enthalpy is the same as the control volume for the density, see Figure 3.24. m^{n+1} and ρ^{n+1} are already known and will be used here. From the energy equation we will calculate the new enthalpy h. Integrating (2.62) over a control volume Ω_i gives:

$$\begin{aligned}\gamma^{-1}\int_{\Omega_i}\frac{\partial(\rho h)}{\partial t}d\Omega+\frac{1}{2}M_r^2(\gamma-1)\int_{\Omega_i}\frac{\partial(\rho^{-1}m^\alpha m^\alpha)}{\partial t}d\Omega+\\+\frac{1}{2}M_r^2(\gamma-1)\int_{\Omega_i}(\rho^{-2}m^\alpha m^\beta m^\beta)_{,\alpha}d\Omega+\int_{\Omega_i}(m^\alpha h)_{,\alpha}d\Omega-\\-(\gamma-1)\frac{M_r^2}{Re_r}\int_{\Omega_i}\{u^\alpha t^{\alpha\beta}\}_{,\beta}d\Omega+\int_{\Omega_i}\frac{1}{Re_rPr}(kh_{,\alpha})_{,\alpha}d\Omega=0.\end{aligned} \tag{3.58}$$

Applying the Gauss-theorem yields:

$$\begin{aligned}\gamma^{-1}\int_{\Omega_i}\frac{\partial(\rho h)}{\partial t}d\Omega+\frac{1}{2}M_r^2(\gamma-1)\int_{\Omega_i}\frac{\partial(\rho^{-1}m^\alpha m^\alpha)}{\partial t}d\Omega+\\+\frac{1}{2}M_r^2(\gamma-1)\int_{\partial\Omega_i}(\rho^{-2}m^\alpha m^\beta m^\beta)n^\alpha d\Gamma+\int_{\partial\Omega_i}(m^\alpha h)n^\alpha d\Gamma-\\-(\gamma-1)\frac{M_r^2}{Re_r}\int_{\partial\Omega_i}\{u^\alpha t^{\alpha\beta}\}n^\beta d\Gamma+\int_{\partial\Omega_i}\frac{1}{Re_rPr}(kh_{,\alpha})n^\alpha d\Gamma=0.\end{aligned} \tag{3.59}$$

The the terms in the first row are the first and the second time derivative. In the second row we find the terms which describe the convection, respectively the kinetic energy convection and the enthalpy convection. The first term of the last row is the viscous term and the last one is the diffusive term.

3.8.1 The first time derivative

Let Δt be the time step. Then the first time derivative is approximated by:

$$\int_{\Omega_i}\frac{\partial(\rho h)}{\partial t}d\Omega\approx\Omega_i\left(\frac{\rho_i^{n+1}h_i^{n+1}-\rho_i^nh_i^n}{\Delta t}\right) \tag{3.60}$$

3.8.2 Second time derivative

the second term in the first line can be approximated by:

$$\int_{\Omega_i}\frac{\partial(\rho^{-1}m^\alpha m^\alpha)}{\partial t}d\Omega\approx\Omega_i\frac{(\rho_i^{-1})^{n+1}(m_i^\alpha m_i^\alpha)^{n+1}-(\rho_i^{-1})^n(m_i^\alpha m_i^\alpha)^n}{\Delta t} \tag{3.61}$$

We already know $\mathbf{m}$ from the momentum equation in Section 3.6. The tangential momentum components $\tilde{m}$ and normal momentum components m in the centers of the cells are calculated by using a reconstruction method described in Section 3.6.3.

3.8.3 Kinetic energy convection

The first term in the second line of (3.59) represents the so-called kinetic energy convection term. By applying the midpoint rule to this term we find the following approximation:

$$\int_{\partial\Omega_i} (\rho^{-2} m^\alpha m^\beta m^\beta) n^\alpha d\Gamma \approx \sum_e m_e \frac{(\mathbf{m}\cdot\mathbf{m})_e}{\rho_e^2} l_e \mathbf{n}_e \cdot \mathbf{N}_e. \tag{3.62}$$

Both momentum and density are taken at the new time level. To get ρ at the faces e the interpolation method in Section 3.6.4 can be used. No complications are found at the boundary since we only are using already calculated variables.

3.8.4 Enthalpy convection

The second term in the second line of (3.59) is called the enthalpy convection term. The enthalpy convection term is discretized as follows:

$$\int_{\partial\Omega_i} (m^\alpha h) n^\alpha d\Gamma \approx \sum_e m_e h_e l_e \mathbf{N}_e \cdot \mathbf{n}_e, \tag{3.63}$$

where e are the faces 1 to 4 in Figure 3.25. However, we only know the

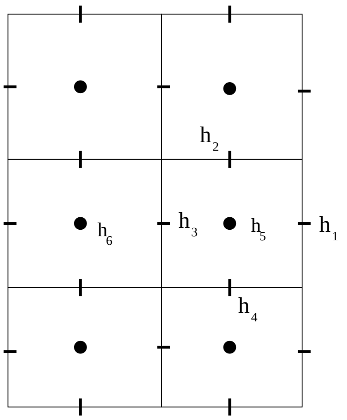

Figure 3.25: Interpolation of the enthalpy

enthalpy in the centers of the primary cells, but we do not have it on the faces of the primary cells. For the central scheme we take the average of the neighbors. The enthalpy h_3, for example, is approximated by:

$$h_3 \approx \frac{h_5 + h_6}{2} \tag{3.64}$$

Application of upwind discretization gives the following for h_3 in Figure 3.25:

If the direction of the flow is from cell 5 to cell 6 then:

$$h_3 = h_5. \tag{3.65}$$

On the other hand, if the flow is from cell 6 to cell 5 then:

$$h_3 = h_6. \tag{3.66}$$

For the boundary cells some adaptations have to be made. We have either a Dirichlet or a Neumann boundary condition for the enthalpy.

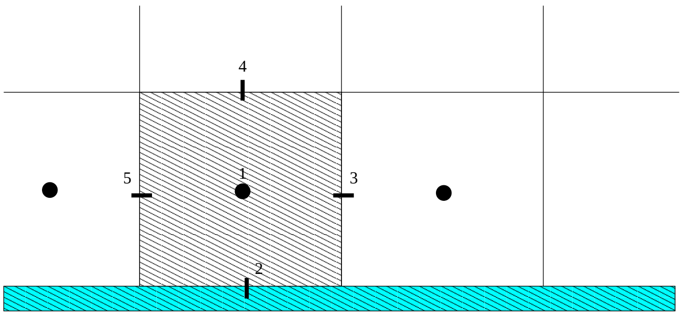

Figure 3.26: The control volume of the enthalpy at the boundary

In Figure 3.26 we need to know the enthalpy in points 2, 3, 4 and 5. The enthalpy in integration points 3, 4 and 5 is determined in the usual way. For h_2 we have either a Dirichlet or a Neumann condition. In case of a homogeneous Neumann condition it means that we apply a one-side difference, so we take: $h_2 = h_1$. If a Dirichlet condition is given, then nothing has to be done.

3.8.5 The viscous term

The first term in the third row of (3.59), the so-called viscous term, is one of the most difficult terms. The viscous term is neglected in order to save time.

3.8.6 The diffusive term

The diffusive term can be approximated in the following way:

$$\int_{\partial\Omega_i} (kh_{,\alpha})n^\alpha d\Gamma \approx \sum_e k_e(h_{,\alpha})_e n_e^\alpha l_e. \tag{3.67}$$

The gradient $(h_{,\alpha})_e$ of the enthalpy can be approximated with the aid of the path integral method as discussed in Section 3.6.2. or cells with at least one boundary face complications are met in the path integral formulation (See Figure 3.27).

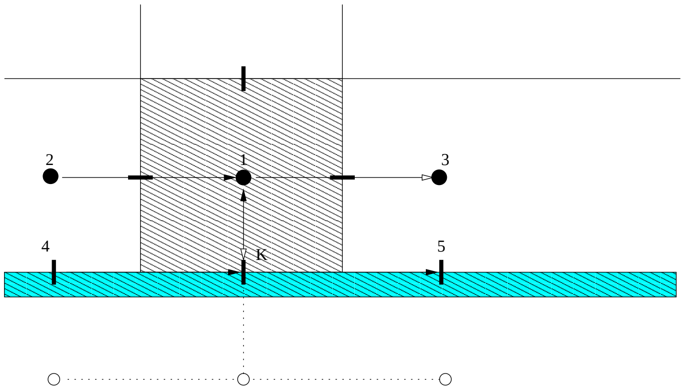

Figure 3.27: Paths on the boundary

Let us consider the enthalpy for cell 1 on face K. If we have a homogeneous Neumann condition then the gradient of the path from K to 1 is given. On the other hand: if we have a Dirichlet condition then the path that would have been outside the flow is moved to the boundary (where we know the enthalpy) so we take the path from 2 to 5, from 4 to 3 and from 1 to K.

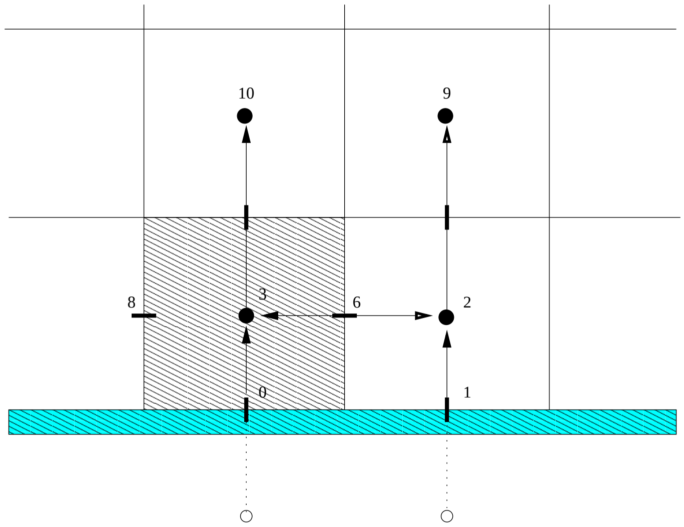

Figure 3.28: Paths on the boundary

The same problem is found for the enthalpy gradient in point K in Figure 3.28. If a Dirichlet condition is given then we let the paths through 1 and 2 stop on the boundary. This yields a path from 5 to 4 and a path from 6 to 3. The same method is applied for the enthalpy gradient in K'. If however a homogeneous Neumann condition is given then we apply one-sided differences exactly in the way it is described in Section 3.6.2 (see Figure 3.8).

Now that we dealt with the discretization the next logical step is the implementation of the governing equations. This is discussed in the next chapter.

Chapter 4

Description of the implementation

In the previous chapter we considered the discretization of the governing equations. The subject of this chapter is the implementation. However, since we do not have much time, some restrictions are made, which is done in section 4.1.

In the Section 4.2 attention is paid to some fundamental arrays, which form an extension to the arrays which already have been implemented and are needed in almost every part of the implementation. In Section 4.3 a description of the matrix building is presented.

4.1 Restrictions

Since the amount of time available is very limited we shall introduce some restrictions in this section. We have only implemented the time-dependent terms and the convective terms but neglect the diffusive and viscous terms. This means that we have implemented the Euler equations instead of the full Navier-Stokes equations.

Secondly we only consider subsonic flows. In flows in which the Mach number is larger than one, usually shocks appear. These are hard to deal with numerically.

For incompressible flows our nondimensionalization method is not valid any more, so we only consider flows with the following property for the Mach number:

$$0.2 < M < 1.0. \tag{4.1}$$

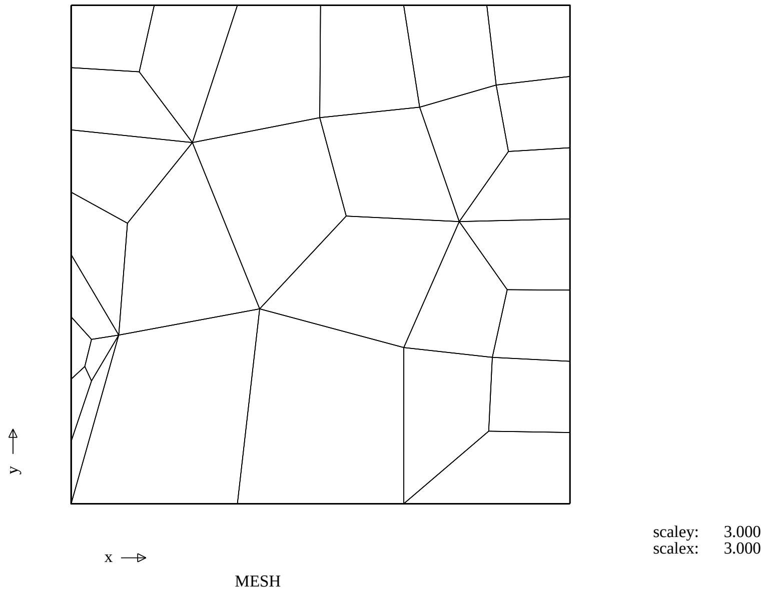

Figure 4.1: An unstructured grid generated

Furthermore in the discretization central differences are applied by taking the average of two neighboring unknowns. For the convection terms we optionally apply first order upwind discretization. These methods will only perform well on grids with more or less rectangular elements, and not on grids with varying angles and areas, such as the grid in Figure 4.1. We will not consider sophisticated reconstruction methods in order to keep the interpolations as simple as possible but at the same time we try to work as accurate as possible.

We now have made our restrictions. In the next section the implementation is discussed.

4.2 The datastructure

In this section a short description of the datastructure is presented. We are making use of a mesh generator, which generates a mesh consisting of quadrilaterals and returns the coordinates of the vertices and per cell the faces. The staggering of the primary variables requires more information compared to collocated placement, since some variables are positioned on the faces and others in the centers of the cells. Per face we want to know what its neighboring cells are. Hence a face based array is needed in which the adjacent cells of each face in the mesh is stored. For each cell we want to know what its faces are. There we need a cell-based array in which the sequence numbers of the faces are stored. The coordinates of the vertices follow directly from the mesh generation and are stored in an array.

Let the number of cells be E and the number of vertices be V. The number of vertices lying on the boundaries is called V_b and the number of internal vertices is denoted as V_i with the following relationship:

$$V = V_i + V_b. \tag{4.2}$$

In practice we might have to do with flows around obstacles. In the mesh these obstacles will appear as internal boundaries. In this report we will call them "holes". Let H be the number of holes. The number of faces is written as F. The faces consist of F_i internal faces and F_b boundary faces, with:

$$F = F_i + F_b. \tag{4.3}$$

Internal faces are situated inside the mesh, the boundary faces on the boundaries of the surface. In the implementation we need to know the amount of internal faces, which is not known in advance. We want to express this in the number of faces and cells. Between cells and faces we have the following relationship (see [2]):

$$E = \frac{1}{4}(F_b + 2F_i). \tag{4.4}$$

By combination of (4.4) and (4.3) we get:

$$F_i = 4E - F. \tag{4.5}$$

This formula is used in the implementation. Between the vertices and cells we have the following relationship:

$$E = \frac{1}{2}(V_b + 2V_i) + H - 1. \tag{4.6}$$

On the boundaries the number of faces is equal to the number of vertices leading to:

$$F_b = V_b. \tag{4.7}$$

Combining (4.4), (4.6) and (4.7) leads to:

$$E + V = F + 1 - H. \tag{4.8}$$

This relationship is used when we test the arrays containing relevant information on the cells and faces. The following three arrays are discussed:

- an array containing the coordinates of the vertices,
- an array containing the information for the cells,
- an array containing the information for the faces.

Coordinates array

The coordinates array, called *coor*, contains the Cartesian coordinates of all vertices. Since we are working in two dimensions, the coordinates are stored in a (2 x V) array of real numbers, denoted by *coor(1:2, 1:V)*.

The cell array

We now consider the cell-based array. For each cell we have to know what the faces are. For quadrilaterals each cell has four faces, so we need a (4 x E) integer array *cellfv(1:4,1:E)*. The faces are stored in counterclockwise order.

The face array

Given a face number we want to know what the sequence numbers of the adjacent cells are and the corresponding position of the face in array $cellfv$ for each of the adjacent cells. We also want to know what the sequence numbers of the vertices are. This is stored in a (6 x F) integer array *face(1:6,1:F)*.

Other quantities that can be determined from these arrays are:

1. the length l_e of the faces,
2. the midpoints of the faces. The midpoint on face e is found by taking the average of the coordinates of the vertices of the face,
3. the inner product of the outward normal with the unique normal per face $(\mathbf{N}_e \cdot \mathbf{n}_e)$.
4. the area, Ω, of a cell,
5. the coordinates of the cell-centroids, found by taking the average of the vertex coordinates of the cell.
6. boundary cells will be treated differently from internal cells. For the cells we need a reordering array. A reordering array contains the same elements as the original array, only in a different order. In the reordering array of the cells all internal cells are stored first. In the second part of the array all the boundary cells are stored.
7. For the face-based arrays we also need a reordering array. We have three types of faces:
    - boundary faces, which are situated on the boundary,
    - quasi-internal faces, where at least one of the four adjacent faces is a boundary face (see Figure 4.2)

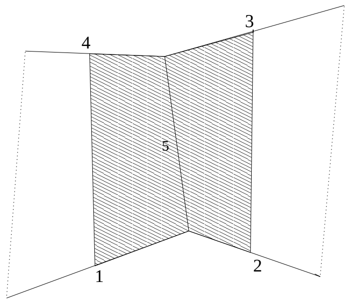

Figure 4.2: Four adjacent faces to the current face 5

- all the other faces are called real internal faces.

The definition of the quasi-internal faces is required for the treatment of the convection term (see Section 3.6.3).

The information provided in this section is used in the rest of the implementation, which consists of the building process of the matrices. This is discussed in the next section.

4.3 Construction of the matrices

We are solving for the following primary variables from the Euler equations:

- the new momentum m,
- the new density ρ,
- the new enthalpy h.

The new density is explicitly solved from already calculated quantities.

For the time stepping we are performing implicit Euler. The variables m and h are solved implicitly: we have to build a matrix for the normal momentum components and enthalpy.

The Dirichlet boundary conditions are incorporated in the matrix and the right-hand side. If the normal component of the momentum in i is given, then a one is put in the i-th position of the diagonal of the matrix, while the rest of the $i-th$ row is filled with zeroes. The condition is put in the $i-th$ position of the right-hand side.

Storing complete matrices leads to memory problems when the matrices are large, which usually is the case. We therefore need to store the matrices in an economic way. In this section is indicated how the matrices are built.

The way in which the matrix is stored is chosen such that it requires a minimal amount of memory. Furthermore matrix-vector multiplication must be carried out efficiently, since this is the most time-consuming part of an iterative linear solver.

The principle is that only essential non-zero elements in the matrix, will be stored. Essential non-zero elements are matrix elements that are usually not equal to zero due to the structure of the grid and the discretization molecules used. The disadvantage of this storage is that it is time-consuming to find an arbitrary element A_{ij} in the matrix.

The collection of the essential non-zero elements form the structure of the matrix.

The construction of the matrix is subdivided into two phases:

1. the structure of the matrix is determined,

2. the matrix elements are filled with the correct numbers.

Section 4.4 deals with the structure of the matrices. The computation of the matrix elements is treated in Sections 4.5 and 4.6.

4.4 The structure of the matrices

First we consider the structure of the matrix. The structure of the matrices is stored in two integer arrays: S_h and S_m, where the subscript h stands for the enthalpy and m for the momentum. The lengths of these arrays are respectively l_{S_h} and l_{S_m}. The arrays S_h and S_m contain the positions of the essential non-zero elements of the matrix. The real matrix elements are stored in respectively array $A_h(l_{A_h})$ and $A_m(l_{A_m})$, where l_{A_h} and l_{A_m} are the lengths of respectively A_h and A_m.

In some occasions essential non-zero elements turn out to be equal to zero. Consider upwind discretization for example. Using upwind interpolation (see Section 3.5) we find that the direction of the flow determines which unknowns are taken into account for interpolation. Since we do not know the direction of the flow, we store all the unknowns which are in the stencil as if central difference were applied.

Let us take the momentum matrix as example for the matrix construction (the enthalpy matrix is built in the same way). If face i under consideration has unknown j in its stencil then the element A_m^{ij} is an essential non-zero element. When A_m^{ij} is an essential non-zero element then also A_m^{ji} is an essential non-zero element, so we have a symmetric connectivity. In integer array S_m we are storing, per row i, all column numbers $j < i$ for which the element A_{ij}^m is non-zero. However, the matrix itself is non-symmetric in general, so all essential non-zero elements have to be stored in A_m. For S_h and A_h the same reasoning holds. Array S_m has the following contents:

- In positions $1 \leq e \leq F+1$ of array S_m (where F is the number of faces) we store the total number of non-zero elements $\{A_{ij} : 1 \leq j < i \leq e-1\}$. We have $S_m(1) = 0$. Hence element $S_m(F+1)$ contains the total number of non-zero elements in the lower triangular part of matrix $\mathbf{A}$, hence the total length l_{S_m} of array S_m is given by: $l_{S_m} = F + 1 + S_m(F+1)$.
- In the last $S_m(F+1)$ positions of the array we are storing the column numbers per row in the following way: the column numbers of all lower triangular non-zero elements are stored subsequently from the starting position of the row. Furthermore the preconditioner of the linear solver requires that the column numbers are stored in increasing order. The position next to the position of the last column number of a certain row is the starting position of the next row.

The real array A_m consists of the following subarrays:

- In array A_m^d with length F we store the diagonal elements $\{A_{ij} : i = j\}$,
- array A_m^l of length $S_m(F+1)$ contains the lower triangular elements $\{A_{ij} : i > j\}$, stored per row,
- the last subarray, A_m^u, with the same length as A^l the upper triangular matrix with elements $\{A_{ij} : i < j\}$ stored per column.

The lengths of the arrays A_m and A_h are given by respectively:

$$l_{A_m} = F + 2 * S_m(F+1), \tag{4.9}$$
$$l_{A_h} = E + 2 * S_h(E+1). \tag{4.10}$$

With this information it is possible to find the position of matrix element A_{ij} in array S_m and inserting the matrix element at the correct position in array A_m. The exact position of the matrix elements in array S_m is found as follows:

1. The starting position for row i is given by: $p_i = F + 1 + S_m(i)$.
2. The number of elements in the lower triangular part of row i is given by $S_m(i+1) - S_m(i)$. Given a row i and column j, knowing that A_{ij} is a non-zero matrix element, there is exactly one integer k with $1 \leq k \leq S_m(i+1) - S_m(i)$ such that $j = S_m(p_i + k)$. Now we know the exact position of element A_{ij} in array S_m.

With the aid of array S_m we can fill the array A_m. Given a matrix element A_{ij} and an array S_m three cases should be considered:

1. $i = j$ (diagonal elements): element A_{ii} is stored at position i of real array A^d_m,
2. $i > j$ (lower triangular elements): the relative position k of for the element A_{ij} in row i is determined as above. This matrix element is now stored at position $S_m(i) + k$ in A^l_m.
3. $i < j$ (upper triangular elements): for the matrix element A_{ji} the value k is determined. The upper triangular element A_{ij} is now stored in position $S_m(j) + k$ in A^u_m.

Arrays S_h and A_h are constructed in the same way.

Consider the following matrix-vector product:

$$b_i = [\mathbf{Ax}]_i = \sum_{j=1}^{n} \left\{ a_{ij}x_j : a_{ij} \neq 0 \right\} = a_{ii}x_i + \\ + \sum_{j<i}^{n} \left\{ a^l_{ij}x_j : a^l_{ij} \neq 0 \right\} + \sum_{j>i}^{n} \left\{ a^u_{ij}x_j : a^u_{ij} \neq 0 \right\}, \tag{4.11}$$

where $\mathbf{A}^l$ is the lower triangular matrix and $\mathbf{A}^u$ the upper triangular matrix. For the upper triangular part the elements are stored per column, while we are multiplying per row. This problem can be avoided by performing calculations per component b_i. The matrix vector product is calculated in the following way:

for all i
 $b_i = a_{ii}x_i$
 $b_i = b_i + \sum_{j \in J} a^l_{ij}x_j$
 for $j \in J$
  $b_j = b_j + a^u_{ji}x_i$
 end
end

where:

$$J = \{j < i : a_{ij} \neq 0\}. \tag{4.12}$$

The product $a_{ii}x_i$ is a trivial operation, however, the computation of $a^l_{ij}x_j$ and $a^u_{ji}x_i$ requires some explanation:

- the loop $b_i = \sum_{j \in J} a^l_{ij} x_j$ is executed in the following way:

$$\begin{aligned} &\textit{for } k := 1(1)n \textit{ do} \\ &\quad j := S_m(p_i + k - 1) \\ &\quad a^l_{ij} := A^l_m(p^l_m + k - 1) \\ &\quad b_i := b_i + a^l_{ij} x_j, \end{aligned}$$

  where p_i and p^l_m are the beginning positions of row i in respectively arrays S_m and A^l_m, n is the number of non-zero elements in row i and k is the column number of the non-zero matrix element.

- for the computation of $a^u_{ji} x_i$ we do the following:

$$\begin{aligned} &\textit{for } k := 1(1)n \textit{ do} \\ &\quad j := S_m(p_i + k - 1) \\ &\quad a^u_{ji} := A^u_m(p^u_m + k - 1) \\ &\quad b_j := b_j + a^u_{ji} x_i, \end{aligned}$$

where p^u_m is the starting position of column i in A^u_m. Of crucial importance is the determination of the stencil of unknowns. For the matrix structure we both consider convection and diffusion. Later on, in the computation of the matrix elements, we drop diffusion. For the matrix structure diffusion is taken into account because it does not require additional work.

Determination of the stencil for momentum

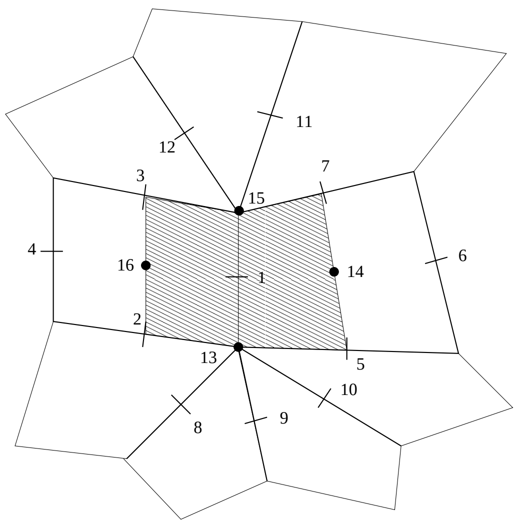

Figure 4.3: essential non-zero elements for the momentum matrix

First we determine the stencil for the momentum. All faces 1 to 12 in Figure 4.3 are considered to be essential non-zero elements. The control volume for the normal momentum component positioned on the face is shaded. The faces where essential non-zero elements are positioned are found in the following way:

- we determine adjacent cells 1 and 2 of face 1,
- from the cell we get the adjacent faces 2 to 7,
- for face 1 we determine the vertices 1 and 2,
- for vertices 1 and 2 we determine the rest of the adjacent faces.

Determination of the stencil for enthalpy

Again we can have the situation that we have more than four cells meeting each other in a vertex (see Figure 4.4). We determine the essential non-zero

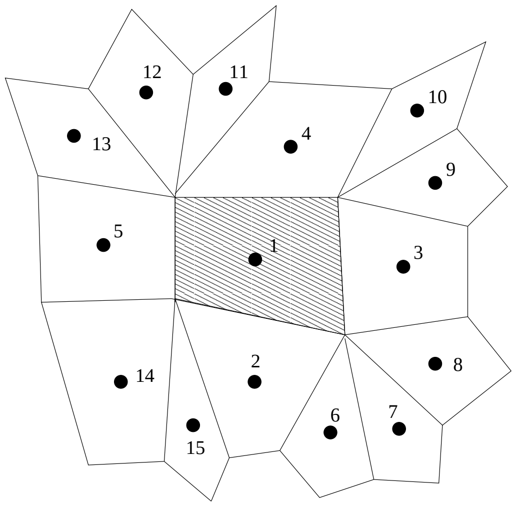

Figure 4.4: essential non-zero elements for the enthalpy matrix

elements for the enthalpy matrix in the following way:

- the faces of cell 1 are determined,
- we determine the adjacent cells 2 to 5 of the faces of cell 1,
- we determine the vertices 1 to 4 of cell 1,
- we determine the rest of the adjacent cells to vertices 1 to 4. These cells are only used for the diffusion term, which is skipped in this report.

We now only have the structure of the matrices, i.e. we know which elements are usually not equal to zero. However we do not know the values to be inserted in the matrix yet. These values should follow from the discretization of the set of equation that has been used. This will be discussed in the next section.

4.5 The matrix elements for the enthalpy

In the previous section the structure of the matrices is discussed. Now the non-zero elements have to be filled. Two real arrays have to be constructed for respectively the momentum matrix and the enthalpy matrix, containing the matrix elements. In this subsection we will deal with the calculation of the elements in the enthalpy matrix. Diffusion will be neglected from now on, although the implementation for the matrix structure can handle it.

We distinguish between two types of cells, namely internal cells and boundary cells. These should be treated differently in the implementation. We make a loop over all internal cells and per cell we build the row in the matrix and the component in the right-hand side of the system. The same is performed for the boundary cells.

The internal cells

In the implementation we shall apply the routines for the general convection-diffusion equation to the enthalpy equation, which is a special case of the first one. The general convection-diffusion equation is given by:

$$\frac{\partial \rho a h}{\partial t} + \nabla \cdot (b\mathbf{m}h) - \nabla \cdot (c\nabla h) + dh = f + Q, \tag{4.13}$$

where a, b, c and d are coefficients which are usually time and/or place dependent. Furthermore there are two source terms f and Q: f is a given function of time and/or place, while Q is a source term following from the choice of the primary variables in the energy equation and is also a function of partial derivatives. The difference between f and Q is made for programming purposes: the term f is determined in a standard routine and can only be place or time dependent, but for term Q a new subroutine has to be implemented. For the enthalpy equation (2.62) without viscosity and diffusion we have:

$$\begin{aligned}\gamma^{-1}\frac{\partial \rho h}{\partial t} + \nabla \cdot (\mathbf{m}h) &= -\frac{1}{2}M_r^2(\gamma-1)\frac{\partial(\rho^{-1}\mathbf{m}\cdot\mathbf{m})}{\partial t} - \\ &\quad -\frac{1}{2}M_r^2(\gamma-1)\nabla\cdot(\rho^{-2}\mathbf{m}(\mathbf{m}\cdot\mathbf{m})). \end{aligned}\tag{4.14}$$

Substitution of (4.14) in (4.13) gives the following coefficients:

$$a = \frac{1}{\gamma}, \quad b = 1, \quad c = d = 0, \tag{4.15}$$

and:

$$f = 0, \quad Q = -\frac{1}{2}M_r^2(\gamma - 1)\left\{\frac{\partial(\rho^{-1}\mathbf{m}\cdot\mathbf{m})}{\partial t} + \nabla\cdot(\rho^{-2}\mathbf{m}(\mathbf{m}\cdot\mathbf{m}))\right\}. \tag{4.16}$$

According to Section 3.8 we have the following discretization for the enthalpy equation in position i:

$$\begin{aligned}
&\gamma^{-1}\Omega_i\left(\frac{\rho_i^{n+1}h_i^{n+1} - \rho_i^n h_i^n}{\Delta t}\right) + \sum_e m_e^{n+1}h_e^{n+1}\tilde{l}_e = \\
&-\frac{1}{2}M_r^2(\gamma-1)\left\{\Omega_i\frac{(\rho^{-1})_i^{n+1}(\mathbf{m}_i^{n+1}\cdot\mathbf{m}_i^{n+1}) - (\rho^{-1})_i^n(\mathbf{m}_i^n\cdot\mathbf{m}_i^n)}{\Delta t}\right\} - \\
&-\frac{1}{2}M_r^2(\gamma-1)\sum_e m_e\left(\frac{\mathbf{m}_e^{n+1}\cdot\mathbf{m}_e^{n+1}}{(\rho^2)_e^{n+1}}\right)\tilde{l}_e,
\end{aligned} \tag{4.17}$$

where e are the faces of the control volume and:

$$\tilde{l}_e = l_e\mathbf{N}_e\cdot\mathbf{n}_e. \tag{4.18}$$

In the implementation we are doing a loop over all cells. The discretized version of the general convection-diffusion is given by:

$$\begin{aligned}
&\Omega_i\frac{\rho_i^{n+1}a_i^{n+1}h_i^{n+1}}{\Delta t} + \sum_e b_e^{n+1}m_e^{n+1}h_e^{n+1}\tilde{l}_e - \sum_e c_e^{n+1}(\nabla h_e^{n+1}\cdot\mathbf{n})_e\tilde{l}_e + \\
&\qquad + \Omega_i d_i^{n+1} = \Omega_i\frac{\rho_i^n a_i^n h_i^n}{\Delta t} + \Omega_i f_i^{n+1} + Q^{n,n+1}.
\end{aligned} \tag{4.19}$$

with:

$$a_i^n = a_i^{n+1} = \frac{1}{\gamma}, \quad b_e^{n+1} = 1, \quad c_e^{n+1} = d_i^{n+1} = 0. \tag{4.20}$$

Furthermore we find the following for the source terms:

$$f_i^{n+1} = 0, \tag{4.21}$$

and:

$$\begin{aligned}
Q^{n,n+1} &= -\frac{1}{2}M_r^2(\gamma-1)\sum_e m_e\left(\frac{\mathbf{m}_e^{n+1}\cdot\mathbf{m}_e^{n+1}}{(\rho^2)_e^{n+1}}\right)\tilde{l}_e - \\
&-\frac{1}{2}M_r^2(\gamma-1)\left\{\Omega_i\frac{(\rho^{-1})_i^{n+1}(\mathbf{m}_i^{n+1}\cdot\mathbf{m}_i^{n+1}) - (\rho^{-1})_i^n(\mathbf{m}_i^n\cdot\mathbf{m}_i^n)}{\Delta t}\right\}
\end{aligned} \tag{4.22}$$

In order to compute the matrix elements we first have to gather all geometrical information. We need the following information (see Figure 4.5):

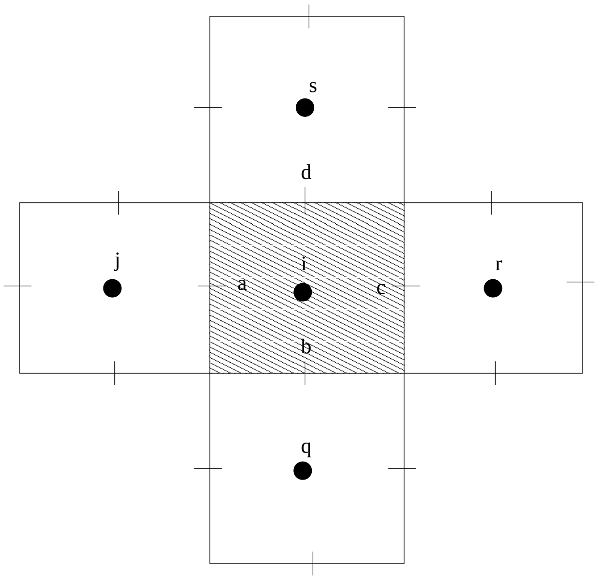

Figure 4.5: the current cell and adjacent faces/cells

1. the area of the control volume Ω_i,
2. the numbers of the faces of Ω_i. Furthermore we need the value $\tilde{l}_e$ which follows from:
   - the length l_e of the faces ,
   - the inner product of the normals $\mathbf{N}_e \cdot \mathbf{n}_e$. This inner product is not calculated, but determined using the convention for the cell numbers (see Section 4.2).
3. we need the following information with respect to the primary variables:
   (a) per face we need to know the following:
      - on the faces e of Ω_i we need $\mathbf{m}_e^{n+1}$ (calculated from the normal and tangential momentum component exactly as in Section 3.6). On the other faces we need m_e^{n+1} for the kinetic energy convection (the second term of 4.17),
      - the density ρ_e^{n+1}.

   (b) the following information in the center of the cell is required:
      - the momenta $\mathbf{m}_i^n$ and $\mathbf{m}_i^{n+1}$,
      - the densities ρ_i^n and ρ_i^{n+1},
      - the old enthalpy h_i^n.

In the kinetic energy convection we need the new density on the faces e. This is calculated by taking the average of the densities in the adjacent cell centers.

We need both the old and the new momenta in the center of the control volume, which follow from the average of the normal momentum components on the faces.

For the enthalpy convection term the new level enthalpy has to be determined on the faces with the aid of the interpolations described in Section 3.8.4. Suppose that a and b are the adjacent cells of a face f under consideration. Then h_f is calculated as follows:

$$h_f = h_a + \alpha_f(h_b - h_a), \tag{4.23}$$

where α_f is the interpolation factor and we take $a < b$.

If we apply central differences and take the average of the enthalpy in the neighboring cells a and b then we find from (4.23) that $\alpha_f = \frac{1}{2}$. In case of first order upwind interpolation we have the following coefficients:

$$\begin{aligned} \alpha_f = 0, &\quad m_f(\mathbf{N}_f \cdot \mathbf{n}_f) > 0, \\ \alpha_f = 1, &\quad m_f(\mathbf{N}_f \cdot \mathbf{n}_f) < 0. \end{aligned}$$

With this information we can construct the matrix and the right hand side in the system for the enthalpy equation. In the right-hand side we have the terms which are already known.

The matrix is filled in the following way (see Figure 4.5):

- For the diagonal elements for internal cells i we have:

$$A_{ii} = \frac{\Omega_i \rho_i^{n+1}}{\gamma \Delta t} + \sum_{e \in \{a,b,c,d\}} \tilde{\alpha}_e^{n+1} m_e^{n+1} \tilde{l}_e, \tag{4.24}$$

  with:

$$\tilde{\alpha}_e^{n+1} = \alpha_e^{n+1} \quad , \quad \textit{if } (\mathbf{N}_e \cdot \mathbf{n}_e) = 1, \tag{4.25}$$

$$\tilde{\alpha}_e^{n+1} = 1 - \alpha_e^{n+1} \quad , \quad \textit{if } (\mathbf{N}_e \cdot \mathbf{n}_e) = -1. \tag{4.26}$$

- Furthermore the following off-diagonal elements are found:

$$A_{ij} = (1 - \tilde{\alpha}_a^{n+1}) m_a^{n+1} \tilde{l}_a, \tag{4.27}$$

$$A_{iq} = (1 - \tilde{\alpha}_b^{n+1}) m_b^{n+1} \tilde{l}_b, \tag{4.28}$$

$$A_{ir} = (1 - \tilde{\alpha}_c^{n+1}) m_c^{n+1} \tilde{l}_c, \tag{4.29}$$

$$A_{is} = (1 - \tilde{\alpha}_d^{n+1}) m_d^{n+1} \tilde{l}_d. \tag{4.30}$$

- all the other elements in row i are equal to zero.

The $i-th$ element of the right hand side is as follows:

$$b_i = -\frac{1}{2}M_r^2(\gamma-1)\left\{\Omega_i\frac{(\rho^{-1})_i^{n+1}(\mathbf{m}_i^{n+1}\cdot\mathbf{m}_i^{n+1})-(\rho^{-1})_i^n(\mathbf{m}_i^n\cdot\mathbf{m}_i^n)}{\Delta t}\right\}+ \\ \Omega_i\frac{\rho_i^n a_i^n h_i^n}{\Delta t}-\frac{1}{2}M_r^2(\gamma-1)\sum_e m_e\left(\frac{\mathbf{m}_e^{n+1}\cdot\mathbf{m}_e^{n+1}}{(\rho^2)_e^{n+1}}\right)\tilde{l}_e \tag{4.31}$$

The inner product $(\mathbf{m}\cdot\mathbf{m})^{n+1}$ is calculated in the same way as is described in Section 3.6.4. In short the following algorithm is executed:

For all internal cells do:

1. determine the geometry
2. determine $\tilde{\alpha}$,
3. calculate the coefficients b_e and c_e,
4. build the right-hand side component,
5. build the matrix row.

The boundary cells

The boundary cells get almost the same treatment as the internal cells. However, a couple of changes are found in the matrix elements.

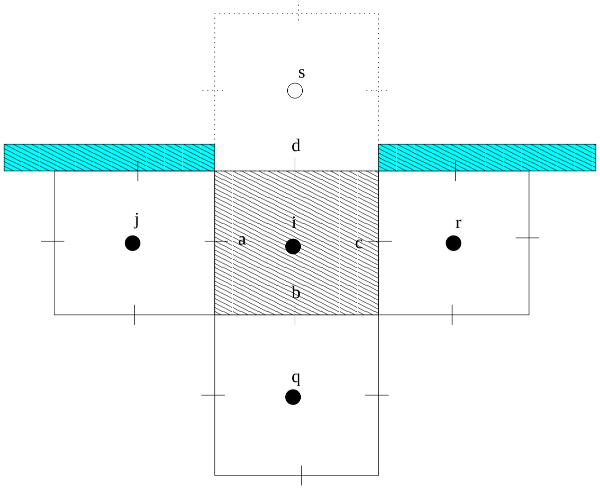

Figure 4.6: current cell and adjacent cells/faces at the boundary

A possible situation is sketched in Figure 4.6. The fact that cells are situated on the boundary leads to one or more absent cells of the stencil and therefore some adaptations have to be made to the procedure here described. On the boundary either a Dirichlet boundary condition or nothing is given. For the Dirichlet conditions we have the following changes in the matrix elements with respect to the matrix elements for the internal cells:

- element A_{is} does not exist, since cell s does not exist,
- for the diagonal elements we get:

$$A_{ii} = \frac{\Omega_i \rho_i^{n+1}}{\gamma \Delta t} + \sum_{e \in \{a,b,c\}} \tilde{\alpha}_e^{n+1} m_e^{n+1} \tilde{l}_e. \tag{4.32}$$

  Mark that the contribution in face d does not appear in the matrix, since enthalpy is given on the boundary.

- the value $-m_d^{n+1} h_d^{n+1} \tilde{l}_d$ is added to the $i-th$ component of the right-hand side, b_i in expression (4.31).

When we have a homogeneous Neumann condition then we get another situation. The enthalpy h_d in Figure 4.6 is unknown and must follow from h_i and h_s. However, since cell s does not exist, we can only use the enthalpy in the cell center of i. This leads to the following changes in the matrix elements:

- the off-diagonal element A_{is} does not exist,
- for the diagonal elements we have:

$$A_{ii} = \frac{\Omega_i \rho_i^{n+1}}{\gamma \Delta t} + \sum_{e \in \{a,b,c\}} \tilde{\alpha}_e^{n+1} m_e^{n+1} \tilde{l}_e + m_d^{n+1} \tilde{l}_d, \tag{4.33}$$

- the element b_i does not change.

Of course situations are possible in which we have two or three boundary faces with different types of boundary conditions in one cell, leading to similar changes in the matrix and/or the right hand-side.

4.6 The matrix elements for the momentum

In this section a description of the implementation of the matrix elements for the momentum equation is presented.

The faces can be subdivided into three groups (see Section 4.2): the internal, quasi-internal and boundary faces. These faces are to be treated differently in the implementation.

According to Section 3.6 the following types of boundary conditions for the momentum equation can occur in case of Euler:

1. the momentum vector at the boundary is given,
2. the normal component of the momentum is given, but not the tangential component,
3. only the pressure is given.

Real internal faces

The integral representation for the momentum equation (2.61), given by (3.23) is discretized as follows:

$$\Omega_{11}\frac{m_{11}^{n+1}}{\Delta t}+\sum_e u_e^n(\mathbf{m}_e^{n+1}\cdot\mathbf{N}_{11})\tilde{l}_e=\Omega_{11}\frac{m_{11}^n}{\Delta t}-\frac{1}{M_r^2}\Omega_{11}\nabla p_{11}^n\cdot\mathbf{N}_{11}, \quad (4.34)$$

where e stands for the integration points on the ad hoc defined faces $12, 13, 14$ and 15 of the control volume Ω_{11} and $\tilde{l}_e=(\mathbf{n}_e\cdot\mathbf{N}_e)l_e$ (see Figure 3.10).

The numbering of the cells faces in the stencil has been chosen in the way it is shown in Figure 3.10 in order to make the implementation for the convection term as easy as possible. Mark that it differs from the determination of the stencil for the momentum in Section 4.4. By convention cell I is the cell with the lower number and II the cell with the higher number.

Vectors $\mathbf{n}_e$ and $\mathbf{N}_e$ are respectively the outward and the unique normals on faces e. If both numbers III and IV refer to the same cell then consequently faces 7 and 9 are absent. These positions in the stencil are then put to zero.

We need the following information:

1. the momentum on face 11 at the old time level, m_{11}^n,
2. the pressures p_I^n, p_{II}^n, p_{III}^n, p_{IV}^n, p_V^n and p_{VI}^n,
3. the length l of faces 1 to 4,
4. the areas Ω_I and Ω_{II} of cells I and II.
5. the normal component of the old velocity vector, u^n on faces 1 to 11 in the stencil,
6. the normals $\mathbf{N}_1\ldots\mathbf{N}_{11}$,

7. the coordinates of the vertices of cells I and II.

If faces 7 and 9 are absent, then u_7^n and u_9^n are set to zero.

The area of the control volume for face 11 is approximated by:

$$\Omega_{11} \approx \frac{\Omega_I + \Omega_{II}}{2}. \tag{4.35}$$

Next we have to calculate the lengths of the faces of the control volume. The lengths of faces 12 and 14 are given by:

$$l_{12} = \frac{l_1 + l_3}{2}, \quad l_{14} = \frac{l_2 + l_4}{2}. \tag{4.36}$$

The lengths l_{13} and l_{15} follow from the coordinates of the midpoints of the adjacent faces 1 to 4, which are calculated with the aid of the coordinates of the cell vertices (see Section 3.2).

In order to get the momenta in the integration points the interpolations described in Section 3.6.3, are performed. From this we find the normal and tangential components of these momenta. The normals in the convection term, however, are given in cartesian coordinates, so we have to transform the momenta $\mathbf{m} = (m, \tilde{m})$ into $\mathbf{m} = (m_x, m_y)$. This procedure has been described in Section 3.6.4.

The second difficult term in the momentum equation is the pressure term (second term in right-hand side of (4.34). We use the path integral method to obtain the pressure gradient on face 11, which has been described in Section 3.6.2. In order to build the matrix elements the following algorithm is executed:

For all internal faces do:

1. determine the stencil and all local quantities,
2. build the right-hand side component,
3. build the matrix row.

Quasi-internal faces

In the neighborhood of a boundary it can occur that one or more of the cells III, IV, V and VI in Figure 3.10 are missing. Face 11 is then called a quasi-internal face. The quasi-internal faces must be given a different treatment in the determination of the stencil and calculation of the convection term. The stencil is determined in the way it is done for the real internal faces with a few exceptions. When a cell is non-existent, for example cell IV in Figure 3.14, the following changes are done:

- the element in the cell-array referring to cell IV is given the value $E+1$, where E is the number of cells. Furthermore pressure p_{IV} is set to zero,

- face 9 will not give a contribution to the interpolation. Position 9 in the stencil array and m_9 are therefore set to 0,

It might happen that also cell III is missing (see Figure 3.15), we then perform the same procedure for cell III and face 7. When the tangential momentum component in 12 is given, then we also get an extra contribution in the right-hand side.

The same holds for cells V and VI.

Boundary faces

At the boundary faces we have two typical situations (see Sections 2.7, 3.6.3 and 3.6.4):

- The normal component of the momentum is given,

- Only the pressure is given.

In the first situation a one is put in the diagonal of the matrix, the rest of the row is filled with zeroes. The Dirichlet condition is put at the appropriate position in the right-hand side.

In the second situation the normal component of the momentum is not given. We then use the discretization for the momentum described in Section 3.6.3.

A boundary face has only one adjacent cell. According to the convention made in Section 4.6, cell II in Figure 3.14 will be missing and so will cells III and V. This new situation is sketched in Figure 3.16, where cells and faces are renumbered.

The control volume is the shaded part of cell I, with faces 6, 7, 8 and 9, of which the last 3 faces are defined locally.

The area of the control volume is approximated by:

$$\Omega_6 \approx \frac{1}{2}\Omega_I. \tag{4.37}$$

The lengths of faces 8 and 9 follow from:

$$l_8 = \frac{1}{2}l_1, \quad l_9 = \frac{1}{2}l_2. \tag{4.38}$$

If cell II does not exist then face 4 is non-existent as well and the same steps are performed as described for the non-existent cells in the stencil of the quasi-internal faces. The same is done for cell III and face 5.

With the two matrices and right-hand sides in this chapter we can solve for the primary variables m and h in the same order as described in Section 3.4. In the next sections the testing of the discretization of the equations is discussed.

Chapter 5

Results

The implementation described in the previous chapter has been tested in the following ways:

- separate tests have been discussed for respectively the enthalpy equation (Section 5.1) and the momentum equation (Section 5.2),
- the coupled system of equations also has been tested. This is the subject of Section 5.3.

In each section we find a description of the test procedures, the results and a discussion.

5.1 Testing of the enthalpy equation

In this section we describe the test procedure that is performed to check the implementation of the enthalpy equation. All other variables (density, momentum) are given.

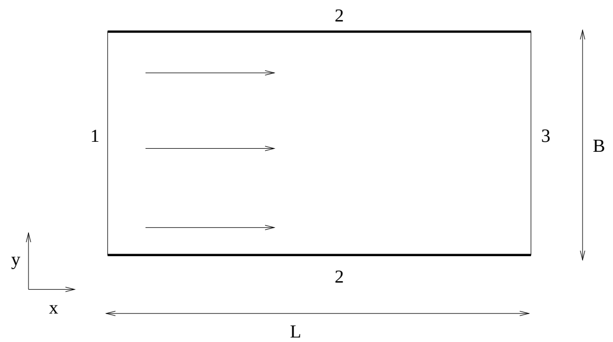

Figure 5.1: The test channel

To test the enthalpy equation we take a channel as in Figure 5.1. Boundary 1 is the inflow boundary, boundaries 2 are solid walls and boundary 3 is the outflow boundary. In our test model the initial conditions are the following:

$$\begin{cases} h^0 = 1, \\ \rho^0 = 1, \\ m_x^0 = m_x(x,y) \\ m_y^0 = 0, \end{cases} \tag{5.1}$$

where $m_x(x,y)$ is a given function.

Furthermore we have the following boundary condition on boundary 1:

$$h = h_1(x,y,t). \tag{5.2}$$

We take the following dimensions for the channel:

$$\begin{cases} L = 4, \\ B = 3, \end{cases} \tag{5.3}$$

where L is the length and B is the width. On boundaries 2 and 3 no enthalpy is prescribed.

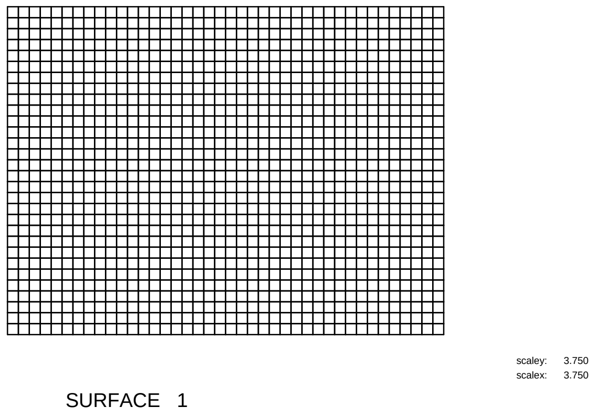

Figure 5.2: A structured 40 x 30 grid

While testing the enthalpy we vary both functions $m_x(x,y)$ and $h_1(x,y,t)$ and take different meshes, namely the one given by Figure 4.1 and a structured grid of 40 x 30 squares (see Figure 5.2). Time steps used were 0.2, 0.02 and 0.002. We test both central differences and first order upwind interpolations. Both meshes are used for all tests.

Test 1

In test 1 we take $m_x^0(x,y) = 1$ and $h_1(x,y) = 1$. Right after the first time step we arrive at the stationary solution $h = h(x,y,t) = 1$, since there is no need for the enthalpy in the channel to adapt to the situation on the inflow.

Test 2

In the second test we take time step $\tau = 0.2$ and give the following Dirichlet condition at the inflow:

$$h_1(x,y,t) = \begin{cases} 1 + \frac{1}{2}t, & 0 < t < 2, \\ 2, & t \geq 2, \end{cases} \tag{5.4}$$

$$m_x^0(x,y) = 1. \tag{5.5}$$

The enthalpy should converge to $h = 2$ over the channel. The time for a particle in the flow to move from the inflow to the outflow boundary is given by:

$$t = \frac{L\rho^0}{m_x^0} = 4.0. \tag{5.6}$$

This means that a stationary situation has to be reached at approximately $t = 2.0 + 4.0 = 6.0$. At time $t = 5.8$ a stationary solution was found for central differences. Using first order upwind interpolations a stationary solution was obtained at $t = 6.0$.

In this interim test the enthalpy was tested for predefined momentum and density and the implementation performed well for these tests. In the next section the tests for the discretization of the momentum equation are discussed.

5.2 Testing of the momentum equation

A couple of test problems are run to check the implementation of the momentum equation. The density and enthalpy are taken constant in time and place during the tests. For all tests we take a surface as in Figure 5.1 with the dimensions $L = 4$, $B = 3$ and set the following initial conditions:

$$\begin{cases} p^0 & = 1, \\ \rho^0 & = 1, \\ m_x^0 & = 1, \\ m_y^0 & = 0. \end{cases} \tag{5.7}$$

For all tests in this section we take the Mach number equal to 0.5 and the time step τ equal to 0.01. We have the following boundary conditions:

$$\begin{cases} \partial\Omega_1 : & \mathbf{m} = (m_x, 0), \quad h = 1, \\ \partial\Omega_2 : & m = 0, \\ \partial\Omega_3 : & p = 1. \end{cases} \tag{5.8}$$

Test 1

In test 1 we describe a trivial problem on the 40 x 30 grid given by Figure 5.2. For m_x in (5.8) we take:

$$m_x = 1. \tag{5.9}$$

The test has been run for upwind interpolation and central differences. Since the boundary conditions match the initial conditions, the new solution must be equal to the old one and therefore exact. After the first time step the stationary solution $\mathbf{m} = (1, 0)$ was reached.

Test 2

In this test we vary the momentum at the inflow:

$$m_x = \begin{cases} 1 + \frac{1}{2}t, & t < 2, \\ 2, & t \geq 2. \end{cases} \tag{5.10}$$

Central differences did not lead to a stationary solution. However, the momentum equation performed well with upwind interpolations. According to (5.10) it takes 2 time units for the momentum at the inflow to arrive at the maximum momentum in x-direction. The time for the particles with a velocity equal to 2 to travel from the inflow to the outflow is another 2 time units. This means that a stationary solution has to be reached approximately at time $t = 4$. At time $t = 3.7$ the momentum was converged to the stationary solution: $\mathbf{m} = (2, 0)$.

Test 3

We now want to know how the discretization of the momentum equation performs on grids with grid elements which are not rectangular. For this purpose we use the mesh shown Figure 5.3 for the problem given in test 1. After 1 time step we expect to find the stationary solution $\mathbf{m} = (1, 0)$, which is exactly equal to the initial conditions. Applying this problem definition and mesh to the numerical scheme here discussed did not lead to any stationary solution at all.

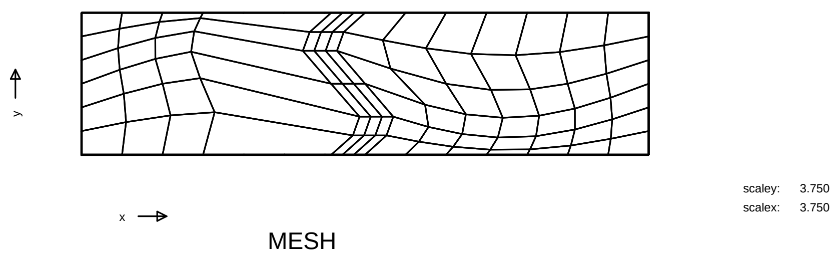

Figure 5.3: A structured grid with non-parallel faces

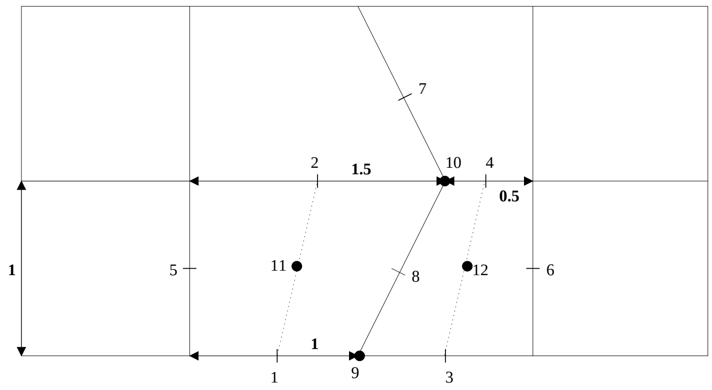

Figure 5.4: A grid with two non-parallel faces

To find out what cause of the problem is, we consider a less complicated grid, shown in Figure 5.4, where compared to a rectangular grid only one vertex is moved with 0.5 length units to the right. In this almost trivial grid it is possible to find out where exactly the problems occur. Again, we take the trivial problem given in test 1. For the calculations performed for this example upwind interpolations are used. For face 8 we now will consider the discretization. In the grid we find the following values for the lengths:

$$l_i = \begin{cases} 1, & i \in \{1,3,5,6,9,10\} \\ \frac{3}{2}, & i \in \{2\} \\ \frac{1}{2}, & i \in \{4\} \\ \frac{1}{2}\sqrt{5}, & i \in \{8\} \\ \frac{1}{4}\sqrt{17}, & i \in \{11,12\} \end{cases} \tag{5.11}$$

and the following momenta and velocities (the initial conditions):

$$m_i = u_i = \begin{cases} 0, & i \in \{1,\ldots,4\} \\ 1, & i \in \{5,6\} \\ \frac{2}{5}\sqrt{5}, & i \in \{8\} \end{cases} \tag{5.12}$$

The discretization of the momentum equation with constant pressure is given by:

$$\Omega_8 \frac{m_8^{n+1} - m_8^n}{\tau} + \sum_e u_e^n (\mathbf{m}_e^{n+1} \cdot \mathbf{N}_8) l_e = 0, \tag{5.13}$$

where $e \in \{9, 10, 11, 12\}$. Since the velocity and momentum are zero on face 1 to 4, integration points 9 and 10 do not give any contribution to the convection term, only points 11 and 12 do. The normals in 5, 6, 8, 11 and 12 are given by:

$$\mathbf{N}_i = \begin{cases} (1, 0), & i \in \{5, 6\} \\ (\frac{2}{5}\sqrt{5}, -\frac{1}{5}\sqrt{5}), & i \in \{8\} \\ (\frac{4}{17}\sqrt{17}, -\frac{1}{17}\sqrt{17}), & i \in \{11, 12\} \end{cases} \tag{5.14}$$

We expect $\mathbf{m}^{n+1}$ to be equal to $\mathbf{m}^n$ in the domain. Substitution of $\mathbf{m}^{n+1} = \mathbf{m}^n$ into (5.13) gives:

$$\sum_e u_e^n (\mathbf{m}_e^n \cdot \mathbf{N}_8) l_e = 0, \tag{5.15}$$

where we use the fact that the normals on the control volume are pointing outward. If the statement above does not hold, then the discretization method is not exact for this problem on this mesh. We write (see Section 3.6.4):

$$\mathbf{m}_e^n = (m_x, m_y)_e = (n_x m + t_x \tilde{m}; n_y m + t_y \tilde{m})_e. \tag{5.16}$$

The tangential momentum component $\tilde{m}_{11}$ follows from an interpolation with m_1 and m_2 and is equal to zero since $m_1 = m_2 = 0$. In the same way we find that $\tilde{m}_{12} = 0$. This yields:

$$\sum_e u_e^n (\mathbf{m}_e^n \cdot \mathbf{N}_i) l_e = u_{11} m_{11} (\mathbf{N}_{11} \cdot \mathbf{N}_8) + u_{12} m_{12} (\mathbf{N}_{12} \cdot \mathbf{N}_8). \tag{5.17}$$

Application of upwind interpolation leads to:

$$\begin{cases} m_{11}^n & = -m_5^n, \\ m_{12}^n & = m_8^n, \\ u_{11}^n & = -u_5^n, \\ u_{12}^n & = u_8^n. \end{cases} \tag{5.18}$$

Substitution of (5.11), (5.14) and (5.18) into (5.15) gives:

$$\sum_e u_e^n (\mathbf{m}_e^n \cdot \mathbf{N}_8) l_e \approx -0.179, \tag{5.19}$$

and we see that the method is not exact. We expected to find a result equal to 0, instead we see that the convection term is of order $O(10^{-1})$, while the solution is of order $O(1)$. However, when we do the same test for Deft, we find that Deft is exact in this case, so Deft performs better on the grid given by Figure 5.4 for trivial problem and and it performs better than the method presented in this report. In Deft one interpolates the momentum multiplied by the length for the convection term. This is only a small adaptation to our existing implementation. Performing the same interpolations we find:

$$\sum_e u_e^n (\mathbf{m}_e^n \cdot \mathbf{N}_8) l_e = 0. \tag{5.20}$$

Our conclusion is that the numerical scheme is not exact on grids containing grid elements with sharp and large angles, even for trivial problems, but that it is exact on this grid if we take into account the lengths of the faces.

However, in general faces 5 and 6 are not parallel to each other. This gives complications in the next example.

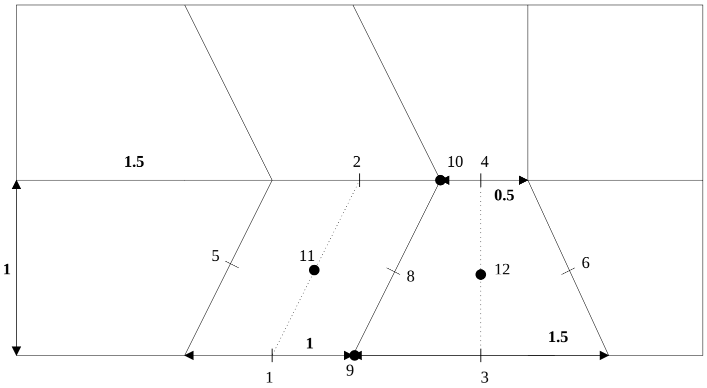

Figure 5.5: A grid with two non-parallel faces

Consider the mesh in Figure 5.5 for the same problem. Neither integration points 9 and 10, nor the tangential components of the momenta 11 and 12 give a contribution to the convection.

The following normals are given:

$$\mathbf{N}_i = \begin{cases} (\frac{2}{5}\sqrt{5}, -\frac{1}{5}\sqrt{5}), & i \in \{5,8,11\} \\ (1,0), & i \in \{12\} \\ (\frac{2}{5}\sqrt{5}, \frac{1}{5}\sqrt{5}), & i \in \{6\} \end{cases} \tag{5.21}$$

For faces 1, 6, 8, 11 and 12 we find the following lengths:

$$l_i = \begin{cases} \frac{1}{2}\sqrt{5}, & i \in \{5,6,8,11\} \\ 1, & i \in \{12\} \end{cases}, \tag{5.22}$$

and the following velocities/momenta for faces 5, 6, 8:

$$m_i = u_i = \frac{2}{5}\sqrt{5}, \quad i \in \{5,6,8\} \tag{5.23}$$

Application of upwind leads to:

$$\begin{cases} m_{11}^n l_{11} & = -m_5^n l_5, \\ m_{12}^n l_{12} & = m_8^n l_8, \\ u_{11}^n & = -u_5^n, \\ u_{12}^n & = u_8^n. \end{cases} \tag{5.24}$$

Equations (5.21), (5.22) and (5.24) into (5.15) yields:

$$\sum_e u_e^n (\mathbf{m}_e^n \cdot \mathbf{N}_8) l_e = -0.2, \tag{5.25}$$

with $e \in \{9, 10, 11, 12\}$. It can be concluded that the adaptation did not lead to an exact solution for the trivial problem on this mesh. For this trivial problem and a grid with a constant cell size in y direction the tangential components of the momenta are equal to zero and we have:

$$\textit{convection term} = \sum_e u_e^n m_e^n l_e (\mathbf{N}_e \cdot \mathbf{N}_8), \tag{5.26}$$

and we mark that we have an inner product $\mathbf{N}_e$ and $\mathbf{N}_8$. In the first example the opposite faces of the control volume are parallel. The inner product will be constant if the faces of the control volume are parallel, and the convection term turns out to be exact if we interpolate the momentum multiplied by the length. In general the parallellity between these faces is *not* guaranteed.

Mark that the treatment of a the convection term is not correct: we assume parallellity of the faces when we interpolate the normal components of the momentum with central differences or upwind and we only make interpolation with the normal components of the momentum. However still we are taking into account both momentum components in the integration points and take the inner product with the normal of the current face instead of the normals in the integration points. This can lead to mathematical inconsistency.
We have two options:

1. we only take into account the normal components of the momenta in both the interpolation and discretization. We then get the following discretization for the convection term:

$$\textit{convection term} = \sum_e u_e^n m_e^n l_e, \tag{5.27}$$

   It is easy to see that this yields an exact result for both examples. However, it might not perform well for non-trivial problems on non-trivial grids.

2. another option would be the following kind of reconstruction (see Figure 5.6): both components of the momentum in 1 follow from normal components m_2 to m_7. This will not be discussed in this report.

In the following section a test for the coupled system of equations is treated.

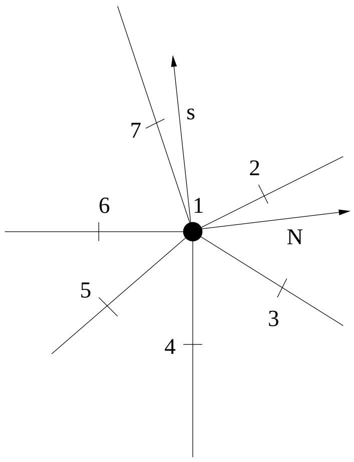

Figure 5.6: Reconstruction in integration point 1

5.3 A flow through a bend

In this section we consider a flow through a bend. For this purpose we use the mesh in Figure 5.7. Mark that the mesh is structured and that the cell faces are more or less parallel to each other. The lower horizontal boundary is the inflow and the upper vertical wall is the outflow. The bent boundaries are solid walls.

The problem is defined as follows:

For the initial conditions we take:

$$\begin{cases} \rho = 1, \\ p = 1, \\ \mathbf{m} = (0,0). \end{cases} \tag{5.28}$$

On the inflow we set:

$$m_y(x,y,t) = \begin{cases} \frac{1}{2}t, & t < 2, \\ 1, & t \geq 2, \end{cases} \tag{5.29}$$

and:

$$m_x = 0, \quad h = 1. \tag{5.30}$$

At the solid walls we have a free-slip condition and on the outflow the pressure is given, namely:

$$p = 1. \tag{5.31}$$

The Mach number is taken equal to 0.5.

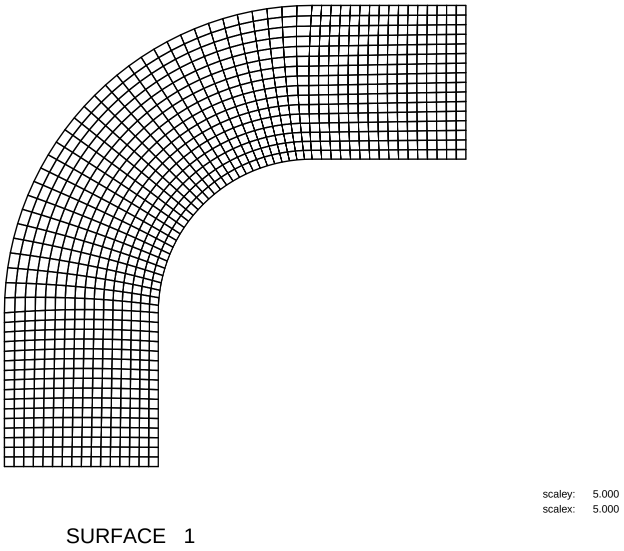

Figure 5.7: A grid for the bent surface

The test is run using first order upwind interpolations. The time step is taken equal to $\tau = 0.005$ and satisfies the CFL-condition given by (3.16). The end solution was reached at time $t = 21.320$. The results are shown in figures 5.8 to 5.13. They were compared to the results found for the same problem and mesh by Deft and the results were practically equal to each other (in Figure 5.14 the solution for the same problem by Deft is shown. Non-dimensionalization has not been performed for the tests with Deft. The reference value for the pressure is $p_r = 0.857 * 10^5$.

Discussion about the results

The irregularities compared to the same problem on a straight channel are found in the inside and outside bend.

For the pressure we mark that the pressure on the left-hand wall is higher than the pressure at the right-hand wall. This has a physical explanation. The flow coming from the inflow tends to move straight on, but the left-hand wall forces the flow to change direction and this gives the extra pressure.

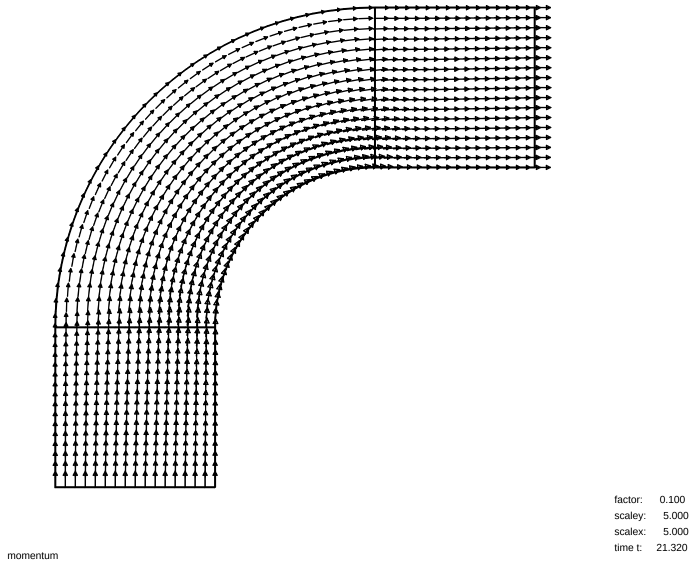

Figure 5.8: The stationary solution for the momentum

The flow is pressed together and this causes a higher density in the environment of the left-hand wall (see Figure 5.11). Since the pressure increases (see Figure 5.9) the pressure gradient in the momentum equation will make the flow slow down. Just after the bend we see that the horizontal momentum at the outside bend is less than 1 (see Figure 5.12).

The flow also tends to move from the right-hand wall. This is shown in the upper straight part of the surface in Figure 5.13 and this explains the lower pressure in the neighborhood of this wall. Since the particles in the flow have a larger freedom of movement we find that the density is also lower and that the momentum in the beginning of the straight part after the bend is larger than 1 in the neighborhood of the inside bend.

In the next and final chapter an overview is given of the activities that have been done and which ones remain to be done in the future in order to develop a method with quadrilaterals that performs well on a larger class of grids for a larger scala of problems.

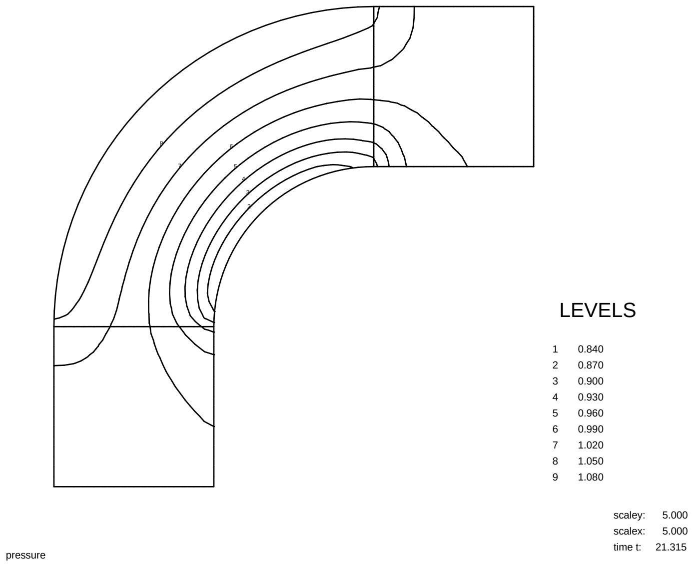

Figure 5.9: The stationary solution for the pressure

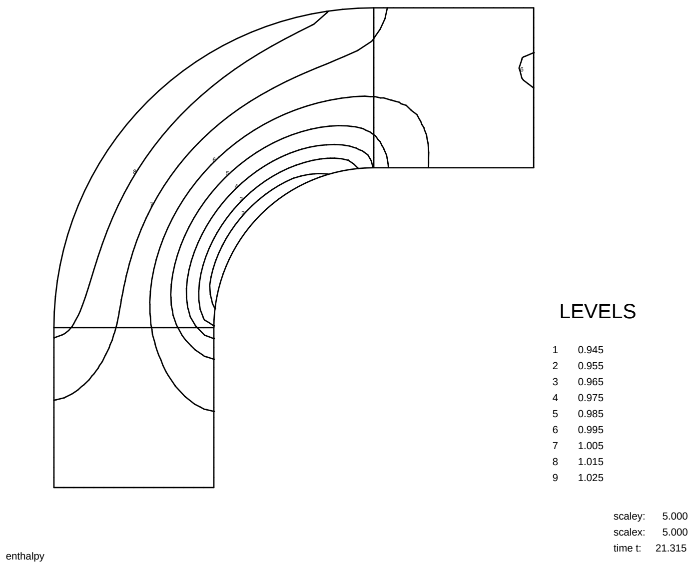

Figure 5.10: The stationary solution for the enthalpy

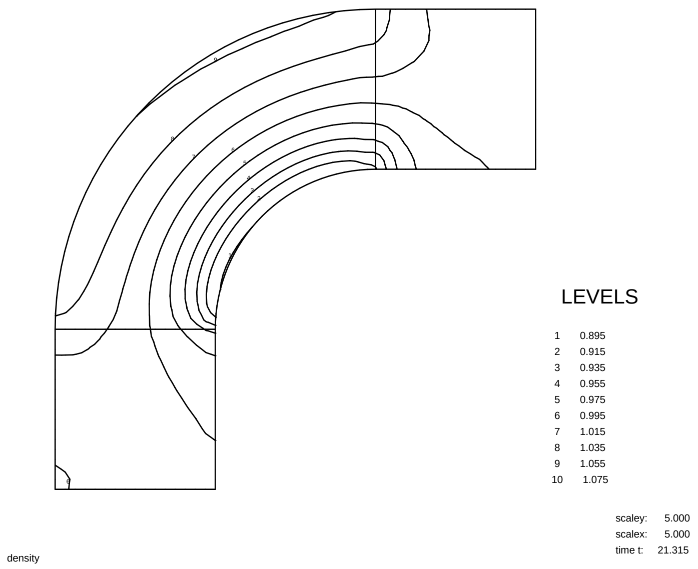

Figure 5.11: The stationary solution for the density

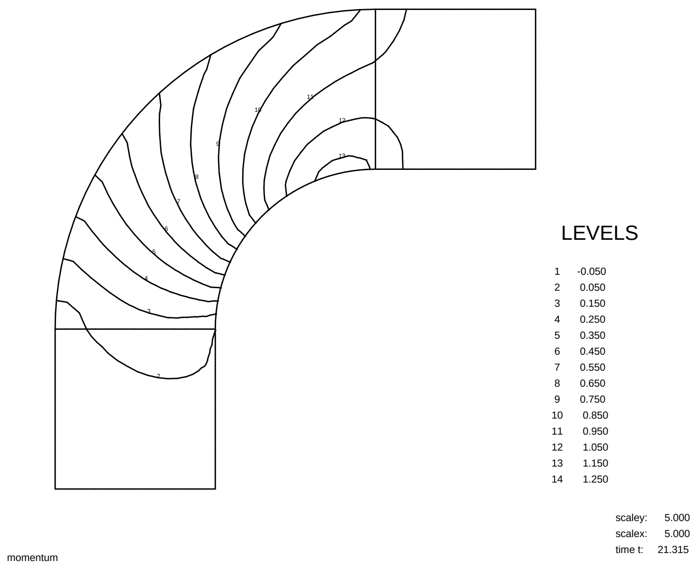

Figure 5.12: The stationary solution for the momentum in horizontal direction

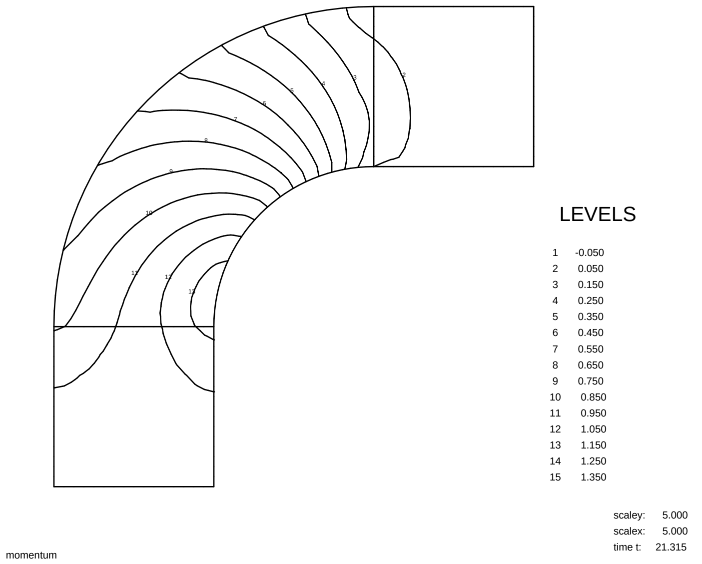

Figure 5.13: The stationary solution for the momentum in vertical direction

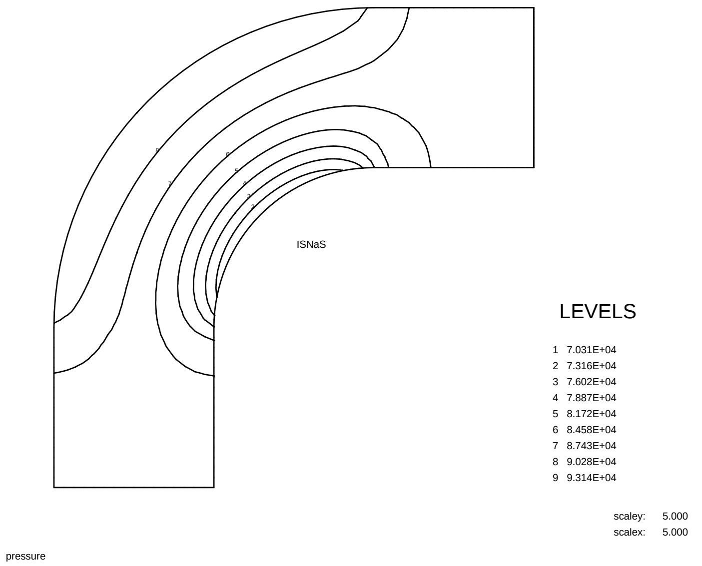

Figure 5.14: The stationary solution for the pressure by Deft

Chapter 6

Conclusions

In this report a discretization for the Navier-Stokes equations on unstructured staggered grids with quadrilaterals is described.

For that purpose the Navier-Stokes equations have been written into a dimensionless form. The nondimensionalization method used however is only suitable for compressible flows.

The dimensionless equations have been discretized with the aid of the finite volume method. Timestepping has been performed using implicit Euler. For convection we make use of both central differences and first order upwind interpolations. For the gradients in the pressure, diffusion and viscous term for the momentum equation the path integral method has been used. In order to save time the viscous term for the enthalpy equation has been neglected.

Because of limited time the implementation of the discretization only includes the Euler equations.

The equations have been tested separately with easy examples and for a more realistic problem, namely a flow through a bend. Shocks have been avoided by only considering subsonic flows.

The results obtained for the coupled system of Euler equations using this implementation have been compared to the results for the same problem using Deft and found to be correct.

However, the implementation did not perform well for the seperate tests of the momentum equation using non-rectangular grids. In the interpolations for the convection terms, the problem arises that our method requires parallellity of the opposite cell faces in both directions. In the integral for convection we are making use of both momentum components in the integration points. In the interpolation however, only the one direction is taken into account. This method is not exact for trivial problems on almost trivial grids, with two or three non-parallel faces. The inplementation, however, performs well for the problem of the flow through the bend on a grid where opposite cell faces are more or less parallel to each other. In order to deal with this problem we have to perform a more sophisticated reconstruction.

After this step one can consider implementation of the viscous terms, which require some changes in the matrix structures described in Section 4.4, and the diffusion term.

For incompressible flows the nondimensionalization method has to be adapted.

When these problem discussed have been solved then one can consider coupling the method for quadrilaterals to the existing method for triangles.

Bibliography

[1] G.K. Bachelor, *An Introduction to Fluid Dynamics.* Cambridge University Press, Cambridge, UK, 1967.

[2] Ewing et al., *Rules governing the numbers of nodes and elements in a finite element mesh*, International Journal for Numerical Methods in Engineering, vol 2, 597-601, 1970

[3] C. Hirsch, *Numerical Computation of Internal and External Flows. Vol.2: Computational Methods for Inviscid and Viscous Flows.* John Wiley, Chichester, 1990.

[4] J. van Kan, A. Segal, *Numerieke methoden voor partiële differentiaalvergelijkingen*, Delft University, Delft, 1993

[5] A. Segal, *ISNaS Mathematical Manual*, Delft University, Delft University, Delft, 1996.

[6] Ivo Wenneker et al., *Computation of compressible and incompressible flows by a staggered finite volume scheme on unstructured grids*, Delft University, Delft, 1998.

[7] Ivo Wenneker, *An unstructured staggered scheme for the Navier-Stokes equations.* Delft University, Delft, 1999.

[8] P. Wesseling, *Principles of Computational Fluid Dynamics.* Delft University, Delft, 1998. (unpublished)