ISNaS - incompressible flow solver

Mathematical manual

Guus Segal
Hester Bijl
Kees Vuik
William Kuppen ?????
Marcel Zijlema
Charles Moulinec

version 2.1 29-07-1998

Contents

1 Introduction

In this manual we describe the mathematical techniques that are used in the ISNaS incompressible program. We do not give any derivation; for the mathematical theory we refer to the literature used.
This manual is meant for ISNaS developers only.

2 Some basic notations from tensor analysis

In the ISNaS incompressible code we are dealing with curvilinear boundary fitted grids. These grids are mapped (by an unknown transformation) onto a rectangular computational grid. Figure 2.1 gives a typical example of the mapping from physical (i.e. curvilinear) to computational grid. All computations are performed in the computational grid and hence the

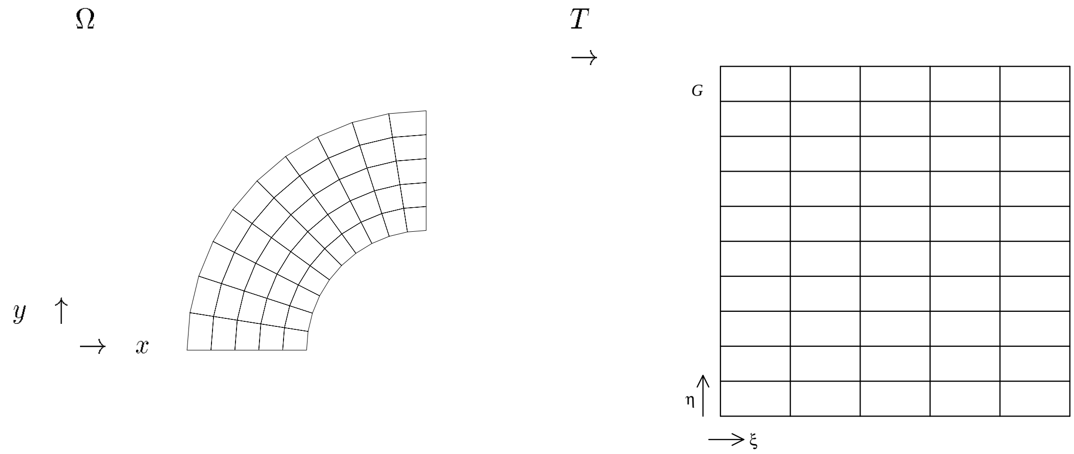


Figure 2.1: Boundary fitted co-ordinates and computational grid

differential equations are transformed from physical grid to computational grid. The resulting solution is transformed backwards.

In the sequel we shall use the following notations:

$$\begin{aligned} \boldsymbol{x} &= (x^1, x^2, x^3) \text{ is the Cartesian co-ordinate system,} \\ \boldsymbol{\xi} &= (\xi^1, \xi^2, \xi^3) \text{ is the general co-ordinate system,} \\ &\quad \text{i.e. the co-ordinate system corresponding to the computational grid.} \end{aligned}$$

The mapping T from Cartesian to computational domain is given by

$$T : x^\alpha = x^\alpha(\xi^1, \xi^2, \xi^3) \tag{2.1}$$

We assume that the Jacobian J:

$$J = |\frac{\partial x^\alpha}{\partial \xi^\beta}| \tag{2.2}$$

is unequal to zero.

We define the covariant base vector $\boldsymbol{a}_{(\alpha)}$ as the tangent vector to the surface $\boldsymbol{x}(\boldsymbol{\xi}^\alpha)$, hence

$$\boldsymbol{a}_{(\alpha)} = \frac{\partial \boldsymbol{x}}{\partial \xi^\alpha} \tag{2.3}$$

The subscript α is placed between parentheses to emphasize that $\boldsymbol{a}_{(\alpha)}$ is not a component but one of the three base vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3$.
Contravariant base vectors $\boldsymbol{a}^{(\alpha)}$ are defined as normal vectors to the $\boldsymbol{\xi}^\alpha$ = constant surfaces:

$$\boldsymbol{a}^{(\alpha)} = \text{grad } \xi^\alpha \tag{2.4}$$

It can be shown that

$$\boldsymbol{a}^{(\alpha)} = \frac{1}{\sqrt{g}} \boldsymbol{a}_{(\beta)} \wedge \boldsymbol{a}_{(\gamma)} \quad \text{for } \alpha, \beta, \gamma \text{ cyclic} \tag{2.5}$$

where $\wedge$ denotes the outer product.

The correspondence between vector and tensor notation for a rank one tensor is expressed by

$$\boldsymbol{u} = U^{\alpha} \boldsymbol{a}_{(\alpha)} = U_{\alpha} \boldsymbol{a}^{(\alpha)} \tag{2.6}$$

For a tensor of rank two the correspondence between the two notations is given by, for example in the case of a mixed tensor:

$$\boldsymbol{U} = U^{\alpha}_{\beta} \boldsymbol{a}_{(\alpha)} \boldsymbol{a}^{(\beta)} \tag{2.7}$$

The covariant and contravariant components of a vector $\boldsymbol{u}$ can be obtained from

$$U_{\alpha} = \boldsymbol{a}_{(\alpha)} \cdot \boldsymbol{u}, \quad U^{\alpha} = \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u} \tag{2.8}$$

For a rank two tensor we have for example

$$U^{\alpha}_{\beta} = \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{U} \cdot \boldsymbol{a}_{(\beta)} \tag{2.9}$$

The metric tensor The covariant and contravariant metric tensors $g_{\alpha\beta}$ and $g^{\alpha\beta}$ are defined as follows:

$$g_{\alpha\beta} = \boldsymbol{a}_{(\alpha)} \cdot \boldsymbol{a}_{(\beta)} \quad , \quad g^{\alpha\beta} = \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{a}^{(\beta)} \tag{2.10}$$

The name metric tensor is related to the fact that the length ds of a small line-segment satisfies

$$ds^2 = dx^{\alpha} dx^{\alpha} = g_{\alpha\beta} d\xi^{\alpha} d\xi^{\beta} \tag{2.11}$$

The determinant of $g_{\alpha\beta}$ is called g, and is given by

$$\sqrt{g} = \boldsymbol{a}_{(1)} \cdot (\boldsymbol{a}_{(2)} \wedge \boldsymbol{a}_{(3)}) \tag{2.12}$$

The two-dimensional version of (2.12) is given by

$$\sqrt{g} = a^1_{(1)} a^2_{(2)} - a^2_{(1)} a^1_{(2)} \tag{2.13}$$

By writing out the right-hand side one sees that

$$\sqrt{g} = J \tag{2.14}$$

The covariant derivative A covariant derivative is a tensor which reduces to a partial derivative of a vector field in Cartesian coordinates. For a scalar, the covariant derivative is the same as the partial derivative, and is denoted by

$$\mu_{,\alpha} = \frac{\partial \mu}{\partial \xi^{\alpha}} \tag{2.15}$$

The covariant derivative of a contravariant tensor of rank one is given by

$$U^{\alpha}_{,\beta} = \frac{\partial U^{\alpha}}{\partial \xi^{\beta}} + \left\{ {\alpha \atop \gamma\beta} \right\} U^{\gamma} \tag{2.16}$$

where $\left\{ {\alpha \atop \gamma\beta} \right\}$ is the so-called Christoffel symbol of the second kind given by

$$\left\{ {\alpha \atop \gamma\beta} \right\} = \boldsymbol{a}^{(\alpha)} \frac{\partial \boldsymbol{a}_{(\gamma)}}{\partial \xi^\beta} = \frac{\partial \xi^\alpha}{\partial x^\delta} \frac{\partial^2 x^\delta}{\partial \xi^\gamma \partial \xi^\beta} = \left\{ {\alpha \atop \beta\gamma} \right\} \tag{2.17}$$

It can be shown that

$$\left\{ {\alpha \atop \gamma\beta} \right\} = \frac{1}{2} g^{\alpha\delta} \left(\frac{\partial g_{\delta\gamma}}{\partial \xi^\beta} + \frac{\partial g_{\delta\beta}}{\partial \xi^\gamma} - \frac{\partial g_{\gamma\beta}}{\partial \xi^\delta} \right) \tag{2.18}$$

The covariant derivative of a covariant tensor of rank one is given by the expression:

$$U_{\alpha,\beta} = \frac{\partial U_\alpha}{\partial \xi^\beta} - \left\{ {\gamma \atop \alpha\beta} \right\} U_\gamma \tag{2.19}$$

It can be shown that

$$U^\alpha_{,\alpha} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} U^\alpha}{\partial \xi^\alpha} \tag{2.20}$$

The covariant derivative of a contravariant tensor of rank two is defined as follows:

$$T^{\alpha\beta}_{,\gamma} = \frac{\partial T^{\alpha\beta}}{\partial \xi^\gamma} + \left\{ {\alpha \atop \delta\gamma} \right\} T^{\delta\beta} + \left\{ {\beta \atop \delta\gamma} \right\} T^{\alpha\delta} \tag{2.21}$$

It can be shown that

$$T^{\alpha\beta}_{,\beta} = \frac{1}{\sqrt{g}} \frac{\delta \sqrt{g} T^{\alpha\beta}}{\partial \xi^\beta} + \left\{ {\alpha \atop \gamma\beta} \right\} T^{\gamma\beta} \tag{2.22}$$

and

$$g^{\alpha\beta}_{,\beta} = 0 \tag{2.23}$$

The covariant derivative of a scalar density (i.e. a relative scalar of weight 1) is defined as

$$\rho_{,\alpha} = \frac{\partial \rho}{\partial \xi^\alpha} - \rho \left\{ {\beta \atop \beta\alpha} \right\} \tag{2.24}$$

It can be shown that

$$\rho_{,\alpha} = \sqrt{g} \frac{\partial \rho / \sqrt{g}}{\partial \xi^\alpha} \tag{2.25}$$

Hence

$$\sqrt{g}_{,\alpha} = 0 \tag{2.26}$$

Another important identity is

$$\frac{\partial \sqrt{g} \boldsymbol{a}^{(\alpha)}}{\partial \xi^\alpha} = 0 \tag{2.27}$$

and is called the geometric identity. This can be derived as follows. Suppose $\boldsymbol{v}$ is constant. Then with the aid of (2.20) we have

$$0 = \operatorname{div} \boldsymbol{v} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{v}}{\partial \xi^\alpha} = \frac{1}{\sqrt{g}} \boldsymbol{v} \cdot \frac{\partial \sqrt{g} \boldsymbol{a}^{(\alpha)}}{\partial \xi^\alpha} \tag{2.28}$$

Since this holds for all constant $\boldsymbol{v}$, (2.27) follows.

The volume element The infinitesimal volume element $d\Omega$ in d dimensions is given by

$$d\Omega = \sqrt{g}d\xi^1 d\xi^2 ... d\xi^d \tag{2.29}$$

The divergence theorem in vector and tensor notation Let $V \subset \Omega$, and let S be the boundary of V. The divergence theorem says, in vector notation,

$$\int_\Omega div\, \boldsymbol{u} dV = \oint_\Gamma \boldsymbol{u} \cdot \boldsymbol{d\Gamma} \tag{2.30}$$

Here $\boldsymbol{d\Gamma}$ stands for the vector $\boldsymbol{n} d\Gamma$, with n the outward unit normal on Γ, and $d\Gamma$ the (physical) surface element. In tensor notation the divergence theorem is given by

$$\int_\Omega U^\alpha_{,\alpha} d\Omega = \oint_\Gamma U^\alpha d\Gamma_\alpha \tag{2.31}$$

For a derivation and further references see Van Kan *et al.* (1991).

3 Discretization of the geometrical quantities

Since we assume that the transformation T is not explicitly known, but only implicitly by the mapping of the co-ordinates of the vertices, it is necessary to discretize the geometrical quantities mentioned in Section 2. In the ISNaS incompressible code there are several ways of computing these quantities. The choice of which of these methods is used is defined by the input parameter geotype. In this section we shall treat the various possibilities as function of geotype. We distinguish between 2D and 3D.

3.1 2D-case

Geotype = 1

2D The following quantities are computed and stored in all points of the grid, i.e. the vertices, the centroids and the midside points:

$$\boldsymbol{a}_{(\alpha)}, g_{\alpha\beta}, \sqrt{g}, \left\{ {\alpha \atop \beta\gamma} \right\}$$

The quantities are computed in the following way:

Consider the p-cell with local numbering as shown in Figure 3.1. First $\boldsymbol{a}_{(1)}$ is computed in

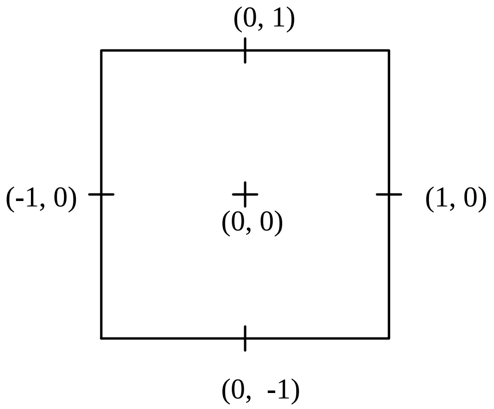


Figure 3.1: local numbering in P-cell

$(0, \pm 1)$ and $\boldsymbol{a}_{(2)}$ in $(\pm 1, 0)$ by:

$$\begin{aligned} \boldsymbol{a}_{(1)}(0, \pm 1) &= \boldsymbol{x}(1, \pm 1) - \boldsymbol{x}(-1, \pm 1) && (3.1) \\ \boldsymbol{a}_{(2)}(\pm 1, 0) &= \boldsymbol{x}(\pm 1, 1) - \boldsymbol{x}(\pm 1, -1) && (3.2) \end{aligned}$$

Next $\boldsymbol{a}_{(1)}$ and $\boldsymbol{a}_{(2)}$ are computed in all points where they are not available by linear or bilinear interpolation, using the fewest number of interpolation points.

Hence:

$$\begin{aligned} \boldsymbol{a}_{(1)}(0, 0) &= \frac{1}{2}\{\boldsymbol{a}_{(1)}(0, 1) + \boldsymbol{a}_{(1)}(0, -1)\} && (3.3) \\ \boldsymbol{a}_{(2)}(0, 0) &= \frac{1}{2}\{\boldsymbol{a}_{(2)}(1, 0) + \boldsymbol{a}_{(2)}(-1, 0)\} && (3.4) \end{aligned}$$

$$\begin{aligned}
\boldsymbol{a}_{(1)}(-1,0) &= \frac{1}{4}\{\boldsymbol{a}_{(1)}(0,1)+\boldsymbol{a}_{(1)}(0,-1)+\boldsymbol{a}_1(-2,1)+\boldsymbol{a}_1(-2,-1)\} && (3.5)\\
\boldsymbol{a}_{(2)}(0,-1) &= \frac{1}{4}\{\boldsymbol{a}_{(2)}(1,0)+\boldsymbol{a}_{(2)}(-1,0)+\boldsymbol{a}_2(-1,-2)+\boldsymbol{a}_2(1,-2)\} && (3.6)\\
\boldsymbol{a}_{(1)}(-1,-1) &= \frac{1}{2}\{\boldsymbol{a}_{(1)}(-1,0)+\boldsymbol{a}_{(1)}(-1,-2)\} && (3.7)\\
\boldsymbol{a}_{(2)}(-1,-1) &= \frac{1}{2}\{\boldsymbol{a}_{(2)}(0,-1)+\boldsymbol{a}_{(2)}(-2,-1)\} && (3.8)
\end{aligned}$$

etc.

From $\boldsymbol{a}_{(1)}$ and $\boldsymbol{a}_{(2)}$ we compute the $g_{\alpha\beta}$ in centroid by

$$g_{\alpha\beta} = \boldsymbol{a}_{(\alpha)} \cdot \boldsymbol{a}_{(\beta)} \qquad (3.9)$$

and $g_{\alpha\beta}$ in all other points is computed by linear or bilinear interpolation from these centroid points.
For example:

$$\begin{aligned}
g_{\alpha\beta}(-1,0) &= \frac{1}{2}\{g_{\alpha\beta}(-2,0)+g_{\alpha\beta}(0,0)\} && (3.10)\\
g_{\alpha\beta}(-1,-1) &= \frac{1}{4}\{g_{\alpha\beta}(0,0)+g_{\alpha\beta}(-2,0)+g_{\alpha\beta}(0,-2)+g_{\alpha\beta}(-2,-2)\} && (3.11)
\end{aligned}$$

etc.
Next $\sqrt{g}$ is computed in all points using the values of $g_{\alpha\beta}$ just computing by

$$\sqrt{g} = \sqrt{|det\ (g_{\alpha\beta})|} \qquad (3.12)$$

To compute $\{{\alpha \atop \beta\gamma}\}$ formula (2.18) is applied in all points of the elements.

So:

$$\begin{aligned}
\{{1 \atop 11}\} &= \frac{1}{2}g^{1\delta}(\frac{\partial g_{\delta 1}}{\partial \xi^1}+\frac{\partial g_{\delta 1}}{\partial \xi^1}-\frac{\partial g_{11}}{\partial \xi^\delta})\\
&= \frac{1}{2}g^{11}\frac{\partial g_{11}}{\partial \xi^1}+g^{12}(\frac{\partial g_{21}}{\partial \xi^1}-\frac{1}{2}\frac{\partial g_{11}}{\partial \xi^2})\\
&= \frac{1}{2}\frac{g_{22}}{g}\frac{\partial g_{11}}{\partial \xi^1}-\frac{g_{12}}{g}(\frac{\partial g_{21}}{\partial \xi^1}-\frac{1}{2}\frac{g_{11}}{\partial \xi^2}) && (3.13)
\end{aligned}$$

$$
\begin{aligned}
\left\{ {1 \atop 12} \right\} &= \frac{1}{2} g^{1\delta} \left(\frac{\partial g_{\delta 1}}{\partial \xi^2} + \frac{\partial g_{\delta 2}}{\partial \xi^1} - \frac{\partial g_{12}}{\partial \xi^\delta} \right) \\
&= \frac{1}{2} g^{11} \left(\frac{\partial g_{11}}{\partial \xi^2} + \frac{\partial g_{12}}{\partial \xi^1} - \frac{\partial g_{12}}{\partial \xi^1} \right) + \frac{1}{2} g^{12} \left(\frac{\partial g_{21}}{\partial \xi^2} + \frac{\partial g_{22}}{\partial \xi^1} - \frac{\partial g_{12}}{\partial \xi^2} \right) \\
&= \frac{1}{2} \frac{g_{22}}{g} \frac{\partial g_{11}}{\partial \xi^2} - \frac{1}{2} \frac{g_{12}}{g} \frac{\partial g_{22}}{\partial \xi^1}
\end{aligned} \tag{3.14}
$$

$$
\begin{aligned}
\left\{ {1 \atop 22} \right\} &= \frac{1}{2} g^{11} \left(\frac{\partial g_{12}}{\partial \xi^2} + \frac{\partial g_{21}}{\partial \xi^2} - \frac{\partial g_{22}}{\partial \xi^1} \right) + \frac{1}{2} g^{12} \left(\frac{\partial g_{22}}{\partial \xi^2} + \frac{\partial g_{22}}{\partial \xi^2} - \frac{\partial g_{22}}{\partial \xi^2} \right) \\
&= \frac{g_{22}}{g} \left(\frac{\partial g_{12}}{\partial \xi^2} - \frac{1}{2} \frac{\partial g_{22}}{\partial \xi^1} \right) - \frac{1}{2} \frac{g_{12}}{g} \frac{\partial g_{22}}{\partial \xi^2}
\end{aligned} \tag{3.15}
$$

$$
\begin{aligned}
\left\{ {2 \atop 12} \right\} &= \frac{1}{2} g^{21} \left(\frac{\partial g_{11}}{\partial \xi^2} + \frac{\partial g_{12}}{\partial \xi^1} - \frac{\partial g_{12}}{\partial \xi^1} \right) + \frac{1}{2} g^{22} \left(\frac{\partial g_{21}}{\partial \xi^2} + \frac{\partial g_{22}}{\partial \xi^1} - \frac{\partial g_{12}}{\partial \xi^2} \right) \\
&= -\frac{1}{2} \frac{g_{21}}{g} \frac{\partial g_{11}}{\partial \xi^2} + \frac{1}{2} \frac{g_{11}}{g} \frac{\partial g_{22}}{\partial \xi^1}
\end{aligned} \tag{3.16}
$$

$$
\begin{aligned}
\left\{ {2 \atop 11} \right\} &= \frac{1}{2} g^{21} \left(\frac{\partial g_{11}}{\partial \xi^1} + \frac{\partial g_{11}}{\partial \xi^1} - \frac{\partial g_{11}}{\partial \xi^1} \right) + \frac{1}{2} g^{22} \left(\frac{\partial g_{21}}{\partial \xi^1} + \frac{\partial g_{21}}{\partial \xi^1} - \frac{\partial g_{11}}{\partial \xi^2} \right) \\
&= -\frac{1}{2} \frac{g_{21}}{g} \frac{\partial g_{11}}{\partial \xi^1} + \frac{g_{11}}{g} \left(\frac{\partial g_{21}}{\partial \xi^1} - \frac{1}{2} \frac{\partial g_{11}}{\partial \xi^2} \right)
\end{aligned} \tag{3.17}
$$

$$
\begin{aligned}
\left\{ {2 \atop 22} \right\} &= \frac{1}{2} g^{21} \left(\frac{\partial g_{12}}{\partial \xi^2} + \frac{\partial g_{12}}{\partial \xi^2} - \frac{\partial g_{22}}{\partial \xi^1} \right) + \frac{1}{2} g^{22} \left(\frac{\partial g_{22}}{\partial \xi^2} + \frac{\partial g_{22}}{\partial \xi^2} - \frac{\partial g_{22}}{\partial \xi^2} \right) \\
&= -\frac{g_{21}}{g} \left(\frac{\partial g_{12}}{\partial \xi^2} - \frac{1}{2} \frac{\partial g_{22}}{\partial \xi^1} \right) + \frac{1}{2} \frac{g_{11}}{g} \frac{\partial g_{22}}{\partial \xi^2}
\end{aligned} \tag{3.18}
$$

In these expressions we have used that $g^{\alpha\beta}$ is the inverse of $g_{\alpha\beta}$, so

$$
\begin{bmatrix} g^{11} & g^{12} \\ g^{12} & g^{22} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} g_{22} & -g_{12} \\ -g_{12} & g_{11} \end{bmatrix} \tag{3.19}
$$

The derivatives $\partial \cdot / \partial \xi^\alpha$ are approximated by central differences using two neighbouring points.

Geotype = 2

The same quantities as for geotype = 1 are computed and stored in the same points. However, there are some minor differences, which result in a more accurate discretization of the differential equations.

The base vector $\boldsymbol{a}_{(\alpha)}$ are computed in exactly the same way as for geotype = 1, i.e. formulae (3.1) and (3.2) are applied.

The Jacobian $\sqrt{g}$ of the transformation in all points is computed from the base vectors in those points, using the expression:

$$
\sqrt{g}_{(\xi,\eta)} = |a^1_{(1)} a^2_{(2)} - a^2_{(1)} a^1_{(2)}|_{(\xi,\eta)} \tag{3.20}
$$

for all points.

In the same way $g_{\alpha\beta}$ is computed by (3.12) in all points.

With respect to the Christoffel symbols $\{ {\alpha \atop \beta\gamma} \}$ not only the interpolation is canceled but also formula (2.18) is replaced by formula (2.17). The base vectors $\boldsymbol{a}^{(\alpha)}$ are computed by inversion of $\boldsymbol{a}_{(\alpha)}$, i.e.

$$\boldsymbol{a}^{(1)} = \frac{1}{\sqrt{g}}(\boldsymbol{a}^2_{(2)},\ -\boldsymbol{a}^1_{(2)}),\quad \boldsymbol{a}^{(2)} = \frac{1}{\sqrt{g}}(-\boldsymbol{a}^2_{(1)},\ a^1_{(1)}) \tag{3.21}$$

The derivatives are again computed by central differences based on 2 neighbouring points.

The formulae derived for the geometrical quantities can all be computed for the internal region. However, at the boundary some extra kind of extrapolation is necessary. In the present version of the flow solver the extrapolation has been taken care of by the introduction of virtual cells and hence virtual co-ordinates. See Figure 3.2.

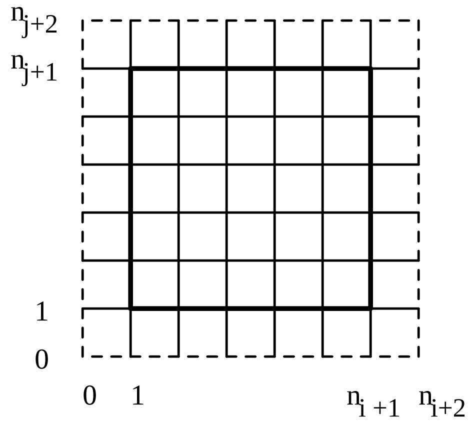


Figure 3.2: virtual cells surrounding the boundary of the region (computational space)

The co-ordinates of the virtual boundary are computed by linear extrapolation, for example

$$\boldsymbol{x}_{i,0} = 2\boldsymbol{x}_{i,1} - \boldsymbol{x}_{i,2} \tag{3.22}$$

The co-ordinates in the 4 vertex points are computed by taking the mean value of the linear extrapolation of the co-coordinates along the two virtual boundaries corresponding to this vertex.

For example

$$\boldsymbol{x}_{0,0} = \frac{1}{2}[(2x_{1,0} - x_{2,0}) + (2x_{0,1} - x_{0,2})] \tag{3.23}$$

The base vectors $\boldsymbol{a}_{(\alpha)}$ are computed in the centroids of all virtual cells and in the midside points of these cells. The metric tensor $g_{\alpha\beta}$ is computed in all non-virtual points as well as all virtual points that are not situated at the outer boundary of the virtual alls. The Christoffel symbols are only computed at the non-virtual points.

3.2 3D-case

The implementation here is only done for geotype = 2.

The covariant base vector $\boldsymbol{a}_{(\alpha)}$ is computed in the centre of the edges of a P-cell parallel to the ξ^{α}-axis, see Figure 3.3.

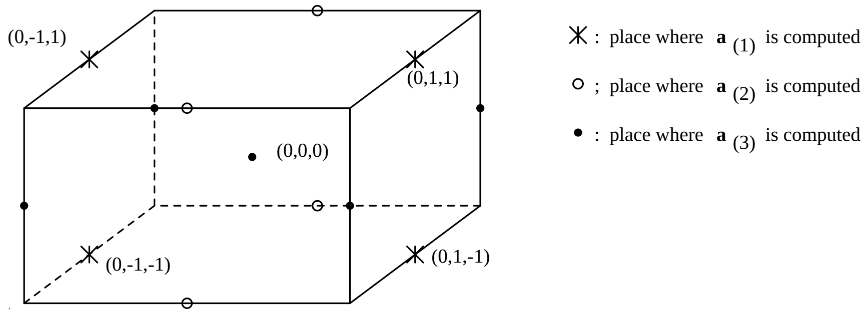


Figure 3.3: P-cell with local numbering and the places where $\mathbf{a}_{(1)}$, $\mathbf{a}_{(2)}$ and $\mathbf{a}_{(3)}$ are computed.

The $\boldsymbol{a}_{(1)}, \boldsymbol{a}_{(2)}$ and $\boldsymbol{a}_{(3)}$ are computed in the following way:

$$\boldsymbol{a}_{(1)}(0,i,j) = \boldsymbol{x}(1,i,j) - \boldsymbol{x}(-1,i,j) \tag{3.24}$$
$$\boldsymbol{a}_{(2)}(i,0,j) = \boldsymbol{x}(i,1,j) - \boldsymbol{x}(i,-1,j) \tag{3.25}$$
$$\boldsymbol{a}_{(3)}(i,j,0) = \boldsymbol{x}(i,j,1) - \boldsymbol{x}(i,j,-1) \tag{3.26}$$

where $i, j, \in \{-1, 1\}$.

Just as the 2D-case we compute $\boldsymbol{a}_{(1)}, \boldsymbol{a}_{(2)}$ and $\boldsymbol{a}_{(3)}$ in all grid points where they are not available by a linear interpolation, using the fewest number of interpolation points.
So:

$$\boldsymbol{a}_{(1)}(-1,-1,-1) = \frac{1}{2}\{\boldsymbol{a}_{(1)}(-2,-1,-1) + \boldsymbol{a}_{(1)}(0,-1,-1)\} \tag{3.27}$$
$$\begin{aligned}\boldsymbol{a}_{(1)}(-1,0,-1) = {} & \frac{1}{4}\{\boldsymbol{a}_{(1)}(-2,-1,-1) + \boldsymbol{a}_{(1)}(0,-1,-1) + \boldsymbol{a}_{(1)}(-2,1,-1) + \\ & \boldsymbol{a}_{(1)}(0,1,-1)\}\end{aligned} \tag{3.28}$$
$$\begin{aligned}\boldsymbol{a}_{(1)}(-1,-1,0) = {} & \frac{1}{4}\{\boldsymbol{a}_{(1)}(-2,-1,-1) + \boldsymbol{a}_{(1)}(0,-1,-1) + \boldsymbol{a}_{(1)}(-2,-1,1) + \\ & \boldsymbol{a}_{(1)}(0,-1,1)\}\end{aligned} \tag{3.29}$$
$$\begin{aligned}\boldsymbol{a}_{(1)}(-1,0,0) = {} & \frac{1}{8}\{\boldsymbol{a}_{(1)}(-2,-1,-1) + \boldsymbol{a}_{(1)}(0,-1,-1) + \boldsymbol{a}_{(1)}(-2,1,-1) + \boldsymbol{a}_{(1)}(0,1,-1) + \\ & \boldsymbol{a}_{(1)}(-2,-1,1) + \boldsymbol{a}_{(1)}(0,-1,1) + \boldsymbol{a}_{(1)}(-2,1,1) + \boldsymbol{a}_{(1)}(0,1,1)\}\end{aligned} \tag{3.30}$$

etc.

The geometrical quantity $\sqrt{g}$ is computed for all gridpoints from the covariant base vectors; using the expression:

$$\begin{aligned}\sqrt{g}_{(i,j,k)} = {} & (a^1_{(1)}a^2_{(2)}a^3_{(3)} - a^1_{(1)}a^3_{(2)}a^2_{(3)} + \\ & a^2_{(1)}a^3_{(2)}a^1_{(3)} - a^2_{(1)}a^1_{(2)}a^3_{(3)} + \\ & a^3_{(1)}a^1_{(2)}a^2_{(3)} - a^3_{(1)}a^2_{(2)}a^1_{(3)})_{(i,j,k)}.\end{aligned} \tag{3.31}$$

The metric tensors $g_{\alpha\beta}$ and $g^{\alpha\beta}$ are computed for all gridpoints by:

$$(g_{\alpha\beta})_{(i,j,k)} = (\boldsymbol{a}_{(\alpha)} \cdot \boldsymbol{a}_{(\beta)})_{(i,j,k)} \tag{3.32}$$

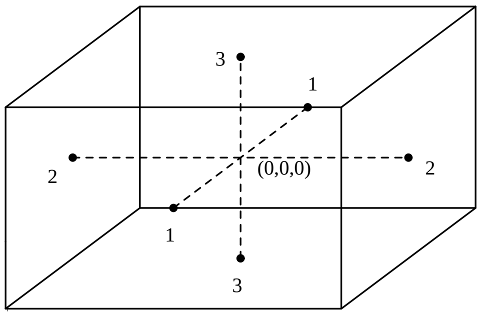
and

$$(g^{\alpha\beta})_{(i,j,k)} = \left(\frac{(-1)^{\alpha+\beta} det\ (G_{\alpha\beta})}{g}\right)_{(i,j,k)} \tag{3.33}$$

where

$$G_{\alpha\beta} = \underset{\alpha\beta}{\text{minor}} \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} . \tag{3.34}$$

Christoffel symbols are computed by formula (2.17) for the centers of the faces of a p-cell, $\{ {1 \atop 11} \}, \{ {1 \atop 12} \}, ..., \{ {1 \atop 33} \}$ for the front and back face $\{ {2 \atop 11} \}, \{ {2 \atop 12} \}, ..., \{ {2 \atop 33} \}$ for the right and left face and $\{ {3 \atop 11} \}, ..., \{ {3 \atop 33} \}$ for the upper and lower face, see Figure 3.4 [1]



1: $\{ {1 \atop 11} \}$, $\{ {1 \atop 12} \}, ..., \{ {1 \atop 33} \}$

2: $\{ {2 \atop 11} \}$, $\{ {2 \atop 12} \}, ..., \{ {2 \atop 33} \}$

3: $\{ {3 \atop 11} \}$, $\{ {3 \atop 12} \}, ..., \{ {3 \atop 33} \}$

Figure 3.4: Places in the p-cell where the Christoffel symbols are computed.

[1] In 3D we don't need the Christoffel symbols in all grid points, because we use another formula for the deviatoric stress tensor ((4.16) instead of (4.2) with (2.16)).

The contravariant base vectors in formula (2.17) are computed by (2.5).
Just as in the 2D-case we introduce virtual cells to compute the geometrical quantities at the boundaries. See Figure 3.5.

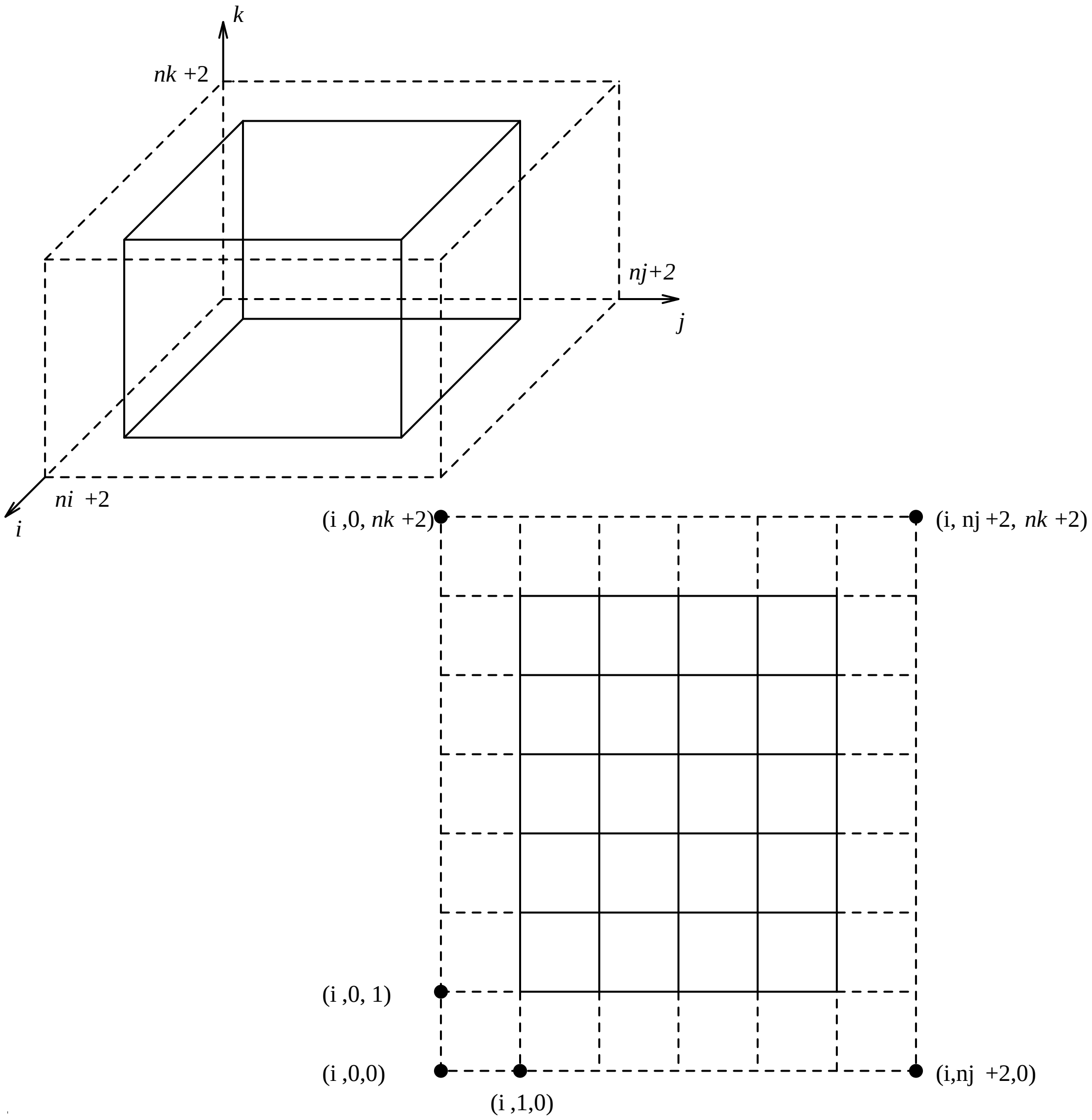


Figure 3.5: The virtual cells surrounding the boundary with a cross-section of the cube.

The co-ordinates of the virtual boundary are computed by a linear extrapolation, for example face $i = 0$

$$\boldsymbol{x}_{0,j,k} = 2\boldsymbol{x}_{1,j,k} - \boldsymbol{x}_{2,j,k} \ , \tag{3.35}$$

edge $i = 0$ and $j = 0$

$$\boldsymbol{x}_{0,0,k} = \frac{1}{2}[(2\boldsymbol{x}_{1,0,k} - \boldsymbol{x}_{2,0,k}) + (2\boldsymbol{x}_{0,1,k} - \boldsymbol{x}_{0,2,k})] \ , \tag{3.36}$$

vertex $i = 0,\ j = 0$ and $k = 0$

$$\boldsymbol{x}_{0,0,0} = \frac{1}{3}[(2\boldsymbol{x}_{1,0,0} - \boldsymbol{x}_{2,0,0}) + (2\boldsymbol{x}_{0,1,0} - \boldsymbol{x}_{0,2,0}) + (2\boldsymbol{x}_{0,0,1} - \boldsymbol{x}_{0,0,2})]\,. \tag{3.37}$$

4 Space discretization of the continuity and momentum equations

In this section we describe the space discretization of the continuity and momentum equations in the inner region. Special remarks are made concerning colocated grids. Discretizations due to the boundary conditions are treated in Section 8.

The equations describing the mean velocity field in incompressible turbulent flow follow from the momentum equations by the Reynolds decomposition of the instantaneous velocity field into a mean and a fluctuating part. The momentum equations in general co-ordinates read (see [36], formula 5.2)

$$\frac{\partial}{\partial t}(\rho U^\alpha) + (\rho U^\alpha U^\beta)_{,\beta} - R^{\alpha\beta}_{,\beta} + (g^{\alpha\beta}p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} = \rho f^\alpha \tag{4.1}$$

with $\tau^{\alpha\beta}$ the deviatoric stress tensor given by

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma}U^\beta_{,\gamma} + g^{\gamma\beta}U^\alpha_{,\gamma}). \tag{4.2}$$

Here, U^α is the contravariant mean velocity, ρ the density, p the mean pressure, f^α some external force per unit volume, μ dynamic viscosity and $R^{\alpha\beta} \equiv -\rho\overline{\widehat{U}^\alpha\widehat{U}^\beta}$ the turbulent stress tensor ($\widehat{U}^\alpha$ denotes contravariant velocity fluctuation), which has to be specified. This specification is accomplished by a two-equation eddy-viscosity turbulence model. This will be presented in Section 7. When calculating laminar flows, the turbulent stresses are set to zero.

In the present version all coefficients may depend on space, time and previous computed solutions. However, with respect to the density a correct implementation is only guaranteed for ρ is constant. Furthermore, the discretization presented below has been carried out as if the flow is assumed laminar.

The continuity equation reads (see [36], formula 5.1):

$$U^\alpha_{,\alpha} = 0 \tag{4.3}$$

As unknowns the fluxes $V^\alpha = \sqrt{g}U^\alpha$ are used.

Equations (4.1), (4.2) and (4.3) are discretized by a finite volume method.

We distinguish between the 2D and the 3D case.

4.1 2D-case

The discretization of the continuity equation is straightforward. We use a staggered grid arrangement as plotted in Figure 4.1.
The continuity equation is integrated over a so-called pressure-cell. This yields:

$$V^1|^{(1,0)}_{(-1,0)} + V^2|^{(0,1)}_{(0,-1)} = 0, \tag{4.4}$$

where the local numbering of Figure 3.1 is used.

With respect to the discretization of the momentum equations we distinguish between the

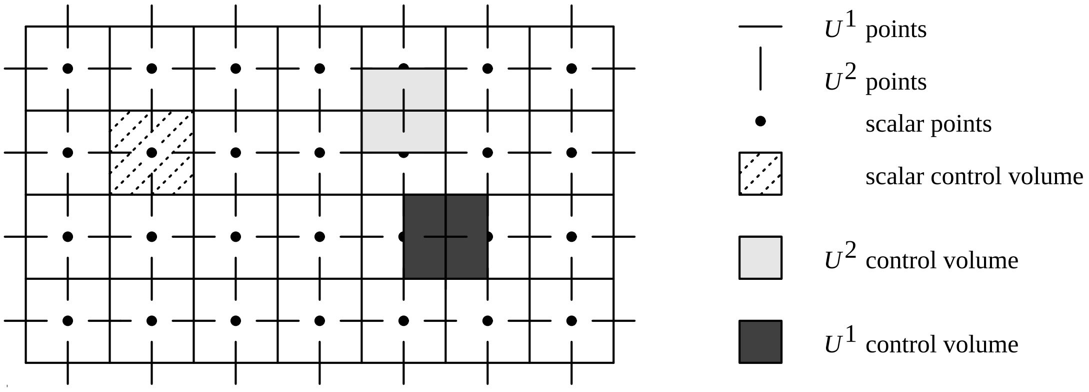


Figure 4.1: Arrangement of the unknowns for a staggered grid

time-derivative, the convection term, the pressure gradient, the deviatoric stress tensor and the right-hand-side term.

The discretization of the time-derivative is given by formula (5.35) of Van Kan *et al.* (1991)):

$$\frac{\partial}{\partial t}(\rho V^\alpha|_{(0,0)}), \tag{4.5}$$

where (0, 0) is the center of a V^α-cell.
The discretization of the right-hand-side term is given by formula (5.34) of that report:

$$\rho f^\alpha \sqrt{g}|_{(0,0)} \tag{4.6}$$

In order to solve the so-called no flow problem, the discretization of the right-hand side has slightly been improved by taking

$$\rho\sqrt{g}(a_1^{(\alpha)} f_1 + a_2^{(\alpha)} f_2)|_{(0,0)}$$

See [27].
The discretization of the convective terms requires a linearization. At this moment only one type of linearization is available, the Newton linearization given by

$$V^\alpha V^\beta \approx V^\alpha \bar{V}^\beta + \bar{V}^\alpha V^\beta - \bar{V}^\alpha \bar{V}^\beta \tag{4.7}$$

where V^α is taken at the new time level and $\bar{V}^\alpha$ at the preceding one.

Apart from the linearization, the discretization of the convective terms is given by formulae (5.8) and (5.9) of Van Kan *et al.* (1991):

V^1-cell:

$$\frac{\rho}{\sqrt{g}}(V^1)^2|_{(-1,0)}^{(1,0)} + \frac{\rho}{\sqrt{g}} V^1 V^2|_{(0,-1)}^{(0,1)} + \frac{\rho}{\sqrt{g}}\left\{ {1 \atop \gamma\beta} \right\} V^\gamma V^\beta|_{(0,0)} \tag{4.8}$$

V^2-cell:

$$\frac{\rho}{\sqrt{g}} V^2 V^1|_{(-1,0)}^{(1,0)} + \frac{\rho}{\sqrt{g}}(V^2)^2|_{(0,-1)}^{(0,1)} + \frac{\rho}{\sqrt{g}}\left\{ {2 \atop \gamma\beta} \right\} V^\gamma V^\beta|_{(0,0)} \tag{4.9}$$

Unknowns not present at points where they are required, are computed by linear interpolation using the least number of neighbouring points possible.

The discretization of the deviatoric stress tensor is carried out according to formulae (5.23) to (5.25) in Van Kan *et al.* (1991):

V^1-cell:

$$-\sqrt{g}\mu(g^{11}U^1_{,1}+g^{12}U^1_{,2})|^{(1,0)}_{(-1,0)}-\sqrt{g}\mu(g^{11}U^2_{,1}+g^{22}U^1_{,2})|^{(0,1)}_{(0,-1)}-\{{1 \atop \gamma\beta}\}\tau^{\gamma\beta}\sqrt{g}|_{(0,0)} \quad (4.10)$$

V^2-cell:

$$-\sqrt{g}\mu(g^{11}U^2_{,1}+g^{22}U^1_{,2})|^{(1,0)}_{(-1,0)}-\sqrt{g}2\mu(g^{12}U^2_{,1}+g^{22}U^2_{,2})|^{(0,1)}_{(0,-1)}-\{{2 \atop \gamma\beta}\}\tau^{\gamma\beta}\sqrt{g}|_{(0,0)} \quad (4.11)$$

with $U^\alpha_{,\beta}$ given by formula (2.22) and $\tau^{\alpha\beta}$ by formula (4.2).

The derivatives $\frac{\partial U^\alpha}{\partial \xi^\beta}$ are computed by central differences, hence

$$\frac{\partial U^\alpha}{\partial \xi^1}|_{(\xi,\eta)} = U^\alpha|_{(\xi+1,\eta)} - U^\alpha|_{(\xi-1,\eta)} \quad (4.12)$$

$$\frac{\partial U^\alpha}{\partial \xi^2}|_{(\xi,\eta)} = U^\alpha|_{(\xi,\eta+1)} - U^\alpha|_{(\xi,\eta-1)}. \quad (4.13)$$

where for (ξ,η) the local numbering is used.
The same type of interpolation is used as for the convective terms. U^α is replaced by $V^\alpha/\sqrt{g}$ in the points where U^α is evaluated, although a better method might be to replace $\sqrt{g}U^\alpha_{,\beta}$ by $V^\alpha_{,\beta}$, since $\sqrt{g}_{,\beta}=0$.

Finally, the discretization of the pressure gradient is carried by formula (3.14) in [26]:

$$(p_{(1,0)}-p_{(-1,0)})(g^{\alpha 1}\sqrt{g})_{(0,0)} + (p_{(0,1)}-p_{(0,-1)})(g^{\alpha 2}\sqrt{g})_{(0,0)} \quad (4.14)$$

4.2 3D-case

First we show in Figure 4.2 the staggered grid arrangements of the unknowns together with the control volumes.

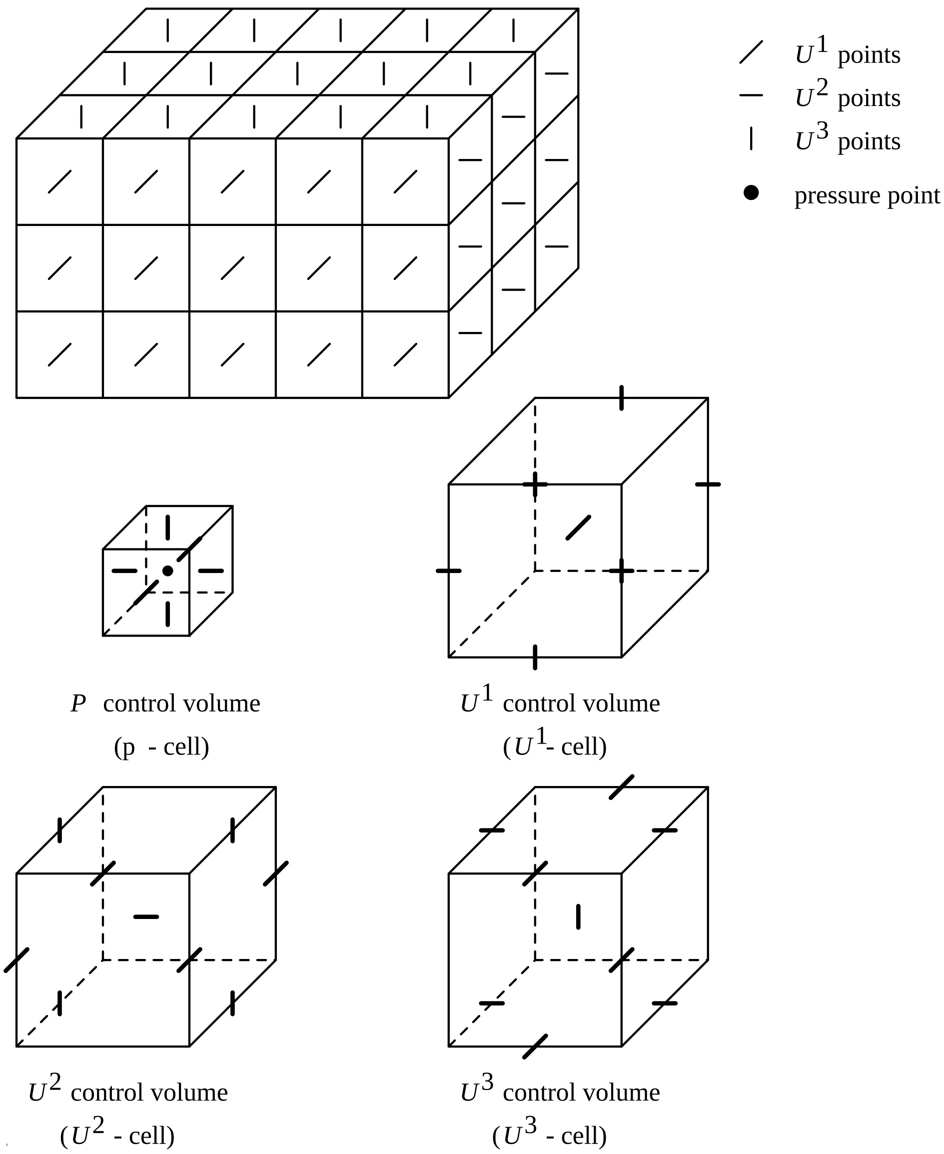


Figure 4.2: Arrangement of the unknowns for a staggered grid

If we integrate the continuity equation over a p-cell get

$$V^1|_{(-1,0,0)}^{(1,0,0)} + V^2|_{(0,-1,0)}^{(0,1,0)} + V^3|_{(0,0,-1)}^{(0,0,1)} = 0 \ , \tag{4.15}$$

this is a simple extension of formula (4.4).

Just as the 2D-case we splitup the momentum equation. The formulae (4.5) - (4.14) are all most the same in 3D. The main difference between 2D and 3D follows from the fact that not a discretization of formula (4.2) is used but from:

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma}\frac{\partial U^\beta}{\partial \xi^\gamma} + g^{\gamma\beta}\frac{\partial U^\alpha}{\partial \xi^\gamma} - \frac{\partial g^{\alpha\beta}}{\partial \xi^\gamma}U^\gamma) \ . \tag{4.16}$$

This formula (4.16) follows directly from equation (4.2), (2.16), (2.21) and $g^{\alpha\beta}_{,\gamma} = 0$.
The discretization of the time derivative is given by:

$$\frac{\partial}{\partial t}((\rho V^\alpha)|_{(0,0,0)}) \ , \tag{4.17}$$

where $(0,0,0)$ is the centre of a V^α-cell.
The discretization of the right-hand-side term is given by

$$(\rho f^\alpha \sqrt{g})|_{(0,0,0)} \ . \tag{4.18}$$

The discretization of the convective terms is given by a straight forward extension of formulae (8.7) and (8.8a)-(8.8b) of Van Kan *et al.* (1991):

$$\begin{aligned} V^\alpha - \text{ cell } : \quad & \\ & \frac{\rho}{\sqrt{g}}V^\alpha V^1|_{(-1,0,0)}^{(1,0,0)} + \frac{\rho}{\sqrt{g}}V^\alpha V^2|_{(0,-1,0)}^{(0,1,0)} + \frac{\rho}{\sqrt{g}}V^\alpha V^3|_{(0,0,-1)}^{(0,0,1)} + \\ & \frac{\rho}{\sqrt{g}}\left\{ {\alpha \atop \gamma\beta} \right\} V^\gamma V^\beta|_{(0,0,0)} \quad \text{for} \quad \alpha \in \{1,2,3\} \ . \end{aligned} \tag{4.19}$$

For all non-linear terms in (4.19) we used the Newton linearization given by

$$V^\alpha V^\beta \approx V^\alpha \bar{V}^\beta + \bar{V}^\alpha V^\beta - \bar{V}^\alpha \bar{V}^\beta \tag{4.20}$$

where V^α is taken at the new time level and $\bar{V}^\alpha$ at the old level.
Unknowns not present at points where they are required, are computed by linear interpolation using the fewest number of interpolation points.

The discretization of the deviatoric stress tensor is carried out according to:

$$\begin{aligned} V^\alpha - \text{ cell } : \quad & \\ & -\sqrt{g}\tau^{\alpha 1}|_{(-1,0,0)}^{(1,0,0)} - \sqrt{g}\tau^{\alpha 2}|_{(0,-1,0)}^{(0,1,0)} - \sqrt{g}\tau^{\alpha 3}|_{(0,0,-1)}^{(0,0,1)} \\ & -(\left\{ {\alpha \atop \gamma\beta} \right\}\tau^{\gamma\beta}\sqrt{g})|_{(0,0,0)} \ , \end{aligned} \tag{4.21}$$

with $\tau^{\alpha\beta}$ given by formula (4.16).
The derivatives $\frac{\partial U^\alpha}{\partial \xi^\beta}$ in (4.16) are computed by central differences, thus:

$$\frac{\partial U^\alpha}{\partial \xi^\beta}|_{(i,j,k)} = U^\alpha|_{(i,j,k)+\Delta_\beta} - U^\alpha|_{(i,j,k)-\Delta_\beta} \tag{4.22}$$

where

$$\Delta_\beta = (\frac{1}{2}|\beta-2|\,|\beta-3|,\ |\beta-1|\,|\beta-3|,\ \frac{1}{2}|\beta-1|\,|\beta-2|)\ . \tag{4.23}$$

U^α is replaced by $V^\alpha/\sqrt{g}$. We make here also the remark that it might be better to replace $\sqrt{g}U^\alpha_{,\beta}$ by $V^\alpha_{,\beta}$. So use

$$\sqrt{g}\tau^{\alpha\beta} = \mu(g^{\alpha\gamma}\frac{\partial V^\alpha}{\partial\xi^\gamma} + g^{\gamma\beta}\frac{\partial V^\alpha}{\partial\xi^\gamma} - \frac{\partial g^{\alpha\beta}}{\partial\xi^\gamma}V^\gamma) \tag{4.24}$$

instead of formula (4.16) in (4.21).
Finally the discretization of the pressure term is carried out by a generalization of formula (3.14) in [26]:

$V^\alpha -$ cell :

$$\begin{aligned}&(g^{\alpha 1}\sqrt{g})|_{(0,0,0)}(p_{(1,0,0)} - p_{(-1,0,0)}) + (\sqrt{g}^{\alpha 2}\sqrt{g})|_{(0,0,0)}(p_{(0,1,0)} - p_{(0,-1,0)}) + \\ &(g^{\alpha 3}\sqrt{g})|_{(0,0,0)}(p_{(0,0,1)} - p_{(0,0,-1)})\ .\end{aligned} \tag{4.25}$$

4.3 2D Cartesian colocated case

Contrary to the staggered case, where the fluxes are directly available at the cell faces, a special treatment has to be made on colocated grids to get the pressure equation. We first show what happens when the straightforward discretization is used and then we present the physical reconstruction of fluxes by Rhie and Chow [21].

First of all, the discretization of the momentum equation on a Cartesian grid is shortly reminded. We start from the continuous equation:

$$\frac{\partial(\rho u^\alpha)}{\partial t} + \frac{\partial(\rho u^\alpha u^\beta)}{\partial x^\beta} + \frac{\partial p}{\partial x^\alpha} - \frac{\partial}{\partial x^\beta}\mu(\frac{\partial u^\alpha}{\partial x^\beta} + \frac{\partial u^\beta}{\partial x^\alpha}) = \rho f^\alpha. \tag{4.26}$$

As the pressure-correction method is used to solve the velocity-pressure coupling (cf. Section 10.2), the discretization of the convective and diffusive terms is not considered here. If the temporal backward Euler scheme is used to discretize the time derivative, (4.26) reduces to:

$$\rho\frac{u^\alpha - u^{*\alpha}}{\Delta t} = -\frac{\partial(p^{n+1} - p^n)}{\partial x^\alpha}. \tag{4.27}$$

Assuming that $p' = p^{n+1} - p^n$, this is rewritten as:

$$u^\alpha = u^{*\alpha} - \frac{\Delta t}{\rho}\frac{\partial p'}{\partial x^\alpha}. \tag{4.28}$$

The grid contains rectangular cells with volume $\Delta x \times \Delta y$ where $\Delta x = \Delta x^1$ and $\Delta y = \Delta x^2$. Integrating the simplified equation (4.28) over the control volume $\Omega_{(0,0)}$ centered in $(0,0)$ gives:

$$u^\alpha_{(0,0)} = u^{*\alpha}_{(0,0)} - \frac{\Delta t}{\rho}\frac{1}{\Delta x\Delta y}\int_{\Omega_{(0,0)}}\frac{\partial p'}{\partial x^\alpha}dx^1dx^2. \tag{4.29}$$

The continuity equation is integrated over the same control volume:

$$\frac{u^1_{(1,0)} - u^1_{(-1,0)}}{\Delta x} + \frac{u^2_{(0,1)} - u^2_{(0,-1)}}{\Delta y} = 0. \tag{4.30}$$

The velocity components at the faces ($u^1_{(1,0)}$, for instance) are not directly available. The natural way to get them is to use interpolation from values known at the cell centers and to write, for the $(1,0)$-face:

$$u^1_{(1,0)} = \frac{u^1_{(0,0)} + u^1_{(2,0)}}{2}. \tag{4.31}$$

When (4.29) is applied to the $(2,0)$ cell, (4.31) reads:

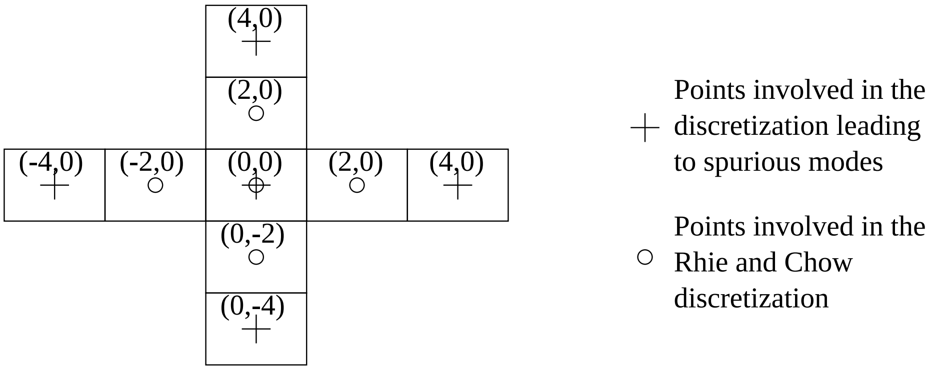


Figure 4.3: Pressure equation discretization

$$u^1_{(1,0)} = \frac{u^{*1}_{(0,0)} + u^{*1}_{(2,0)}}{2} - \frac{\Delta t}{2\rho}\frac{1}{\Delta x \Delta y}\left(\int_{\Omega_{(0,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 + \int_{\Omega_{(2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2\right). \tag{4.32}$$

Similarly, for the $(-1,0)$-face,

$$u^1_{(-1,0)} = \frac{u^{*1}_{(-2,0)} + u^{*1}_{(0,0)}}{2} - \frac{\Delta t}{2\rho}\frac{1}{\Delta x \Delta y}\left(\int_{\Omega_{(-2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 + \int_{\Omega_{(0,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2\right). \tag{4.33}$$

When the same process is followed for u^2, through the $(0,1)$ and $(0,-1)$ faces, the new expression of the continuity equation reads:

$$\begin{aligned}\frac{1}{\Delta x \Delta y}\Big(\frac{\Delta t}{2\rho \Delta x}\Big(\int_{\Omega_{(2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 - \int_{\Omega_{(-2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2\Big) +{} \\ \frac{\Delta t}{2\rho \Delta y}\Big(\int_{\Omega_{(0,2)}} \frac{\partial p'}{\partial x^2} dx^1 dx^2 - \int_{\Omega_{(0,-2)}} \frac{\partial p'}{\partial x^2} dx^1 dx^2\Big)\Big) ={} \\ \frac{u^{*1}_{(1,0)} - u^{*1}_{(-1,0)}}{\Delta x} + \frac{u^{*2}_{(0,1)} - u^{*2}_{(0,-1)}}{\Delta y}.\end{aligned} \tag{4.34}$$

Let us now consider $\frac{1}{\Delta x \Delta y}\int_{\Omega_{(2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2$. It is equal to:

$$\frac{1}{\Delta x \Delta y}\int_{\Omega_{(2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 = \frac{p'_{(3,0)} - p'_{(1,0)}}{\Delta x}, \tag{4.35}$$

or in other words, because $p'_{(3,0)} = \frac{p'_{(4,0)} + p'_{(2,0)}}{2}$ and $p'_{(1,0)} = \frac{p'_{(2,0)} + p'_{(0,0)}}{2}$:

$$\frac{1}{\Delta x \Delta y}\int_{\Omega_{(2,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 = \frac{p'_{(4,0)} - p'_{(0,0)}}{2\Delta x}. \tag{4.36}$$

Generalizing (4.36) to the other integrals involved in (4.34) gives:

$$\frac{\Delta t}{4\rho}\left(\frac{p'_{(4,0)} - 2p'_{(0,0)} + p'_{(-4,0)}}{(\Delta x)^2} + \frac{p'_{(0,4)} - 2p'_{(0,0)} + p'_{(0,-4)}}{(\Delta y)^2}\right) = \frac{u^{*1}_{(2,0)} - u^{*1}_{(-2,0)}}{2\Delta x} + \frac{u^{*2}_{(0,2)} - u^{*2}_{(0,-2)}}{2\Delta y}. \tag{4.37}$$

We can see clearly, that $p'_{(2,0)}$, $p'_{(-2,0)}$, $p'_{(0,2)}$ and $p'_{(0,-2)}$, which are the closest directly available neighbours of $p'_{(0,0)}$ are not present in (4.37) (see Figure 4.3). It means that spurious modes might appear in the pressure field. To avoid them and thus to build a more compact discretization of the Laplacian operator, Rhie and Chow have suggested not to use formula (4.32), but to generalize (4.29) to the cell faces. For the $(1,0)$-face, u^1 now reads:

$$u^1_{(1,0)} = u^{*1}_{(1,0)} - \frac{\Delta t}{\rho}\frac{1}{\Delta x \Delta y}\int_{\Omega_{(1,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2. \tag{4.38}$$

$u^{*1}_{(1,0)}$ is given by:

$$u^{*1}_{(1,0)} = \frac{u^{*1}_{(0,0)} + u^{*1}_{(2,0)}}{2} \tag{4.39}$$

and $\frac{1}{\Delta x \Delta y}\int_{\Omega_{(1,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2$ by:

$$\frac{1}{\Delta x \Delta y}\int_{\Omega_{(1,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2 = \frac{p'_{(2,0)} - p'_{(0,0)}}{\Delta x} \tag{4.40}$$

which leads to:

$$u^1_{(1,0)} = \frac{u^{*1}_{(0,0)} + u^{*1}_{(2,0)}}{2} - \frac{\Delta t}{\rho}\frac{p'_{(2,0)} - p'_{(0,0)}}{\Delta x} \tag{4.41}$$

The pressure equation is built by replacing the new expression of the velocity components at the faces by (4.41) in (4.30):

$$\frac{\Delta t}{\rho}\left(\frac{p'_{(2,0)} - 2p'_{(0,0)} + p'_{(-2,0)}}{(\Delta x)^2} + \frac{p'_{(0,2)} - 2p'_{(0,0)} + p'_{(0,-2)}}{(\Delta y)^2}\right) = \frac{u^{*1}_{(2,0)} - u^{*1}_{(-2,0)}}{2\Delta x} + \frac{u^{*2}_{(0,2)} - u^{*2}_{(0,-2)}}{2\Delta y}. \tag{4.42}$$

It is obvious that the contribution of the immediate neighbours of the discretization nodes is effective in this equation (see Figure 4.3). It can be noticed that the pressure operator is the same for staggered and colocated approaches when the pressure correction method is applied.

5 Space discretization of the transport equation

In this section we describe the space discretization of the transport equation in the inner region. Discretizations due to the boundary conditions are treated in Section 8.

5.1 Invariant finite volume discretization

The general transport equation in Cartesian co-ordinates reads:

$$\frac{\partial c^*\phi}{\partial t} + \boldsymbol{u}\cdot\nabla(c^*\phi) - \mathrm{div}\ (k\nabla\phi) + D\phi = f^*, \tag{5.1}$$

where ϕ stands for a physical scalar (e.g. temperature, concentration, turbulence quantities), c^* and D are general functions, k is a $d \times d$ symmetric diffusion matrix (d is the space dimension) and f^* is a source term. These functions may depend on other unknowns, like U^α and ϕ. Translated into general co-ordinates this equation becomes (see [36], formula 5.4):

$$\frac{\partial c^*\phi}{\partial t} + (c^*U^\alpha\phi)_{,\alpha} - (K^{\alpha\beta}\phi_{,\beta})_{,\alpha} + D\phi = f^*, \tag{5.2}$$

with $K^{\alpha\beta} = a_\gamma^{(\alpha)} a_\delta^{(\beta)} k^{\gamma\delta}$. Using (2.20), equation (5.2) can be written in a form which is suitable for the discretization by the finite volume method:

$$\frac{1}{\sqrt{g}}\frac{\partial\sqrt{g}Q^\alpha}{\partial\xi^\alpha} = f^* - D\phi - \frac{\partial c^*\phi}{\partial t} \tag{5.3}$$

with

$$Q^\alpha = c^*U^\alpha\phi - K^{\alpha\beta}\phi_{,\beta} \tag{5.4}$$

The transport equation (5.3) is integrated over a pressure cell with center (i,j,k) which yields

$$\begin{aligned}\int_{\Omega_{ijk}} \frac{1}{\sqrt{g}}\frac{\partial\sqrt{g}Q^\alpha}{\partial\xi^\alpha}d\Omega &= \int_{G_{ijk}} \frac{\partial\sqrt{g}Q^\alpha}{\partial\xi^\alpha}d\xi^1 d\xi^2 d\xi^3 \\ &\approx \sqrt{g}Q^1|_{(i-1/2,j,k)}^{(i+1/2,j,k)} + \sqrt{g}Q^2|_{(i,j-1/2,k)}^{(i,j+1/2,k)} + \sqrt{g}Q^3|_{(i,j,k-1/2)}^{(i,j,k+1/2)}\end{aligned} \tag{5.5}$$

whereas the right-hand side is integrated using the midpoint rule:

$$\int_{\Omega_{ijk}} (f^* - D\phi - \frac{\partial c^*\phi}{\partial t})d\Omega \approx \sqrt{g}(f^* - D\phi - \frac{\partial c^*\phi}{\partial t})|_{(i,j,k)} \tag{5.6}$$

Next, (5.4) is substituted in (5.5) which completes the discretization. Since the unknown ϕ is only given in the center of a cell, further approximation is needed. Central differences should be used because of second order accuracy. The problem here is, however, that such schemes tend to give rise to oscillations which are found to damage the stability of solutions of the two-equation models, since negative values of the turbulent quantities tend to be enlarged by nonlinearities and strong coupling between the model equations, which prevents the solutions to converge. This matter will be discussed in more detail in Section 5.3.

5.2 Invariant discretization on non-smooth grids

The discretization of the transport equation (5.1) as explained in the previous section is restricted to more or less smooth grids. Moreover, the diffusion coefficient $k^{\alpha\beta}$ often varies rapidly, so that $k^{\alpha\beta}$ has large jumps on cell faces. Hence, straightforward discretization of the diffusivity may results in large errors even on smooth grids. Here we shall consider a discretization which is exact for uniform and linear scalar fields, regardless the smoothness of the grid and of the diffusion coefficient. For this purpose, we need an expression for the partial derivative of any quantity Φ with respect to $\boldsymbol{x}$ in terms of general coordinates, namely

$$\frac{\partial \Phi}{\partial x^\beta} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\gamma} (\sqrt{g} a_\beta^{(\gamma)} \Phi) \tag{5.7}$$

This can be derived using the chain rule and (2.27). We start with the equation (5.1), viz.,

$$\frac{\partial c^* \phi}{\partial t} + \frac{\partial c^* u^\beta \phi}{\partial x^\beta} - \frac{\partial}{\partial x^\beta}(k^{\beta\gamma} \frac{\partial \phi}{\partial x^\gamma}) + D\phi = f^*, \tag{5.8}$$

and with (5.7) taken into account, the appropriate form of the transport equation becomes

$$\frac{\partial c^* \phi}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial c^* V^\alpha \phi}{\partial \xi^\alpha} - \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha} (\sqrt{g} a_\beta^{(\alpha)} k^{\beta\gamma} \frac{\partial \phi}{\partial x^\gamma}) + D\phi = f^* \tag{5.9}$$

Discretization of (5.9) is obtained by integration over a finite volume Ω_{ijk} with center (i, j, k). Hence, we have

$$\begin{aligned}\int_{\Omega_{ijk}} \frac{1}{\sqrt{g}} \frac{\partial c^* V^\alpha \phi}{\partial \xi^\alpha} d\Omega &= \int_{G_{ijk}} \frac{\partial c^* V^\alpha \phi}{\partial \xi^\alpha} d\xi^1 d\xi^2 d\xi^3 \\ &\approx c^* V^1 \phi|_{(i-1/2,j,k)}^{(i+1/2,j,k)} + c^* V^2 \phi|_{(i,j-1/2,k)}^{(i,j+1/2,k)} + c^* V^3 \phi|_{(i,j,k-1/2)}^{(i,j,k+1/2)}\end{aligned} \tag{5.10}$$

whereas the time derivative and the source terms are integrated using the midpoint rule:

$$\begin{aligned}&\int_{\Omega_{ijk}} \frac{\partial c^* \phi}{\partial t} d\Omega \approx \sqrt{g}_{(i,j,k)} \frac{\partial c^* \phi}{\partial t}|_{(i,j,k)}, \quad \int_{\Omega_{ijk}} D\phi d\Omega \approx \sqrt{g}_{(i,j,k)} D\phi|_{(i,j,k)}, \\ &\int_{\Omega_{ijk}} f^* d\Omega \approx \sqrt{g}_{(i,j,k)} f^*|_{(i,j,k)}\end{aligned} \tag{5.11}$$

So far, no difficulties arose because none of the quantities used in the above formulae are discontinuous. Furthermore, these discretizations are second order accurate and (5.10) is exact for constant ϕ. The cell-face values have to be approximated in terms of values of ϕ in neighbouring cell centers by means of interpolation. This will be discussed in the next section. The only point left to discuss is the approximation of the diffusion term. First, for the sake of easier manipulations, the diffusion term will hereafter be expressed as

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha} (\sqrt{g} a_\beta^{(\alpha)} k^{\beta\gamma} \frac{\partial \phi}{\partial x^\gamma}) = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha} (\sqrt{g} \boldsymbol{a}^{(\alpha)} \cdot k \nabla \phi) \tag{5.12}$$

Integration over Ω_{ijk} yields:

$$\begin{aligned}-\int_{G_{ijk}} \frac{\partial}{\partial \xi^\alpha} (\sqrt{g} \boldsymbol{a}^{(\alpha)} \cdot k \nabla \phi) d\xi^1 d\xi^2 d\xi^3 \quad &\approx \quad -\sqrt{g} \boldsymbol{a}^{(1)} \cdot k \nabla \phi|_{(i-1/2,j,k)}^{(i+1/2,j,k)} - \sqrt{g} \boldsymbol{a}^{(2)} \cdot k \nabla \phi|_{(i,j-1/2,k)}^{(i,j+1/2,k)} \\ &\quad -\sqrt{g} \boldsymbol{a}^{(3)} \cdot k \nabla \phi|_{(i,j,k-1/2)}^{(i,j,k+1/2)}\end{aligned} \tag{5.13}$$

The physical quantity $k\nabla\phi$ is everywhere continuous for arbitrary mappings. Hence, when k is discontinuous, $\nabla\phi$ is discontinuous. Thus, approximation of $\nabla\phi$ using central differences is inaccurate. However, using the integration-path method an accurate approximation of $k\nabla\phi$ at point $(i+1/2, j, k)$, for example, is obtained:

$$k\nabla\phi|_{(i+1/2,j,k)} = \phi|_{(i,j,k)}^{(i+1,j,k)}\boldsymbol{c}^{(1)} + \frac{1}{4}(\phi|_{(i,j-1,k)}^{(i,j+1,k)} + \phi|_{(i+1,j-1,k)}^{(i+1,j+1,k)})\boldsymbol{c}^{(2)} + \frac{1}{4}(\phi|_{(i,j,k-1)}^{(i,j,k+1)} + \phi|_{(i+1,j,k-1)}^{(i+1,j,k+1)})\boldsymbol{c}^{(3)} \tag{5.14}$$

where

$$\boldsymbol{c}^{(\alpha)} = \frac{1}{C}(\boldsymbol{c}_{(\beta)} \wedge \boldsymbol{c}_{(\gamma)}), \quad \alpha, \beta, \gamma \quad \text{cyclic} \tag{5.15}$$

where C is given by

$$C = \boldsymbol{c}_{(1)} \cdot (\boldsymbol{c}_{(2)} \wedge \boldsymbol{c}_{(3)}) \tag{5.16}$$

and

$$\boldsymbol{c}_{(1)} = \frac{1}{2}(\frac{\boldsymbol{a}_{(1)}}{k}|_{(i,j,k)} + \frac{\boldsymbol{a}_{(1)}}{k}|_{(i+1,j,k)}) \tag{5.17}$$

$$\begin{aligned} \boldsymbol{c}_{(2)} \quad = \quad & \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i,j-1,k)} + \frac{1}{4}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i,j,k)} + \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i,j+1,k)} + \\ & \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j-1,k)} + \frac{1}{4}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j,k)} + \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j+1,k)} \end{aligned} \tag{5.18}$$

$$\begin{aligned} \boldsymbol{c}_{(3)} \quad = \quad & \frac{1}{8}\frac{\boldsymbol{a}_{(3)}}{k}|_{(i,j,k-1)} + \frac{1}{4}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i,j,k)} + \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i,j,k+1)} + \\ & \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j,k-1)} + \frac{1}{4}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j,k)} + \frac{1}{8}\frac{\boldsymbol{a}_{(2)}}{k}|_{(i+1,j,k+1)} \end{aligned} \tag{5.19}$$

We hereby assume that $k^{\alpha\beta} = k\delta^{\alpha\beta}$. The two-dimensional version of (5.15) and (5.16) are given by

$$\boldsymbol{c}^{(1)} = \frac{1}{C}(c_{(2)}^2, -c_{(2)}^1)^\top, \quad \boldsymbol{c}^{(2)} = \frac{1}{C}(-c_{(1)}^2, c_{(1)}^1)^\top \tag{5.20}$$

with

$$C = c_{(2)}^2 c_{(1)}^1 - c_{(1)}^2 c_{(2)}^1 \tag{5.21}$$

Substitution of (5.14) in (5.13) gives a discretization which is exact when $\nabla\phi$ is constant, and hence exact for linear scalar fields. The other cell-face fluxes can be derived in exactly the similar way.

5.3 Approximation methods for convective flux

In this section we discuss several methods for expressing a cell-face value of a scalar in terms of surrounding nodal values. For clarity, the discussion relates to a uniform infinite staggered grid as depicted in Figure 5.1. For reasons of robustness and algorithmic simplicity, multidimensional central and upwind schemes are treated by one-dimensional decomposition in the normal direction for each cell face. Such schemes are called splitting schemes. Therefore we restrict ourselves to interpolation of the face values along one specific direction in G-space, taking for example the ξ^1-direction. We need to consider only the cell-face value $\phi_{i+1/2}$. The other face values will be treated in the same way.

Central difference scheme (CDS).

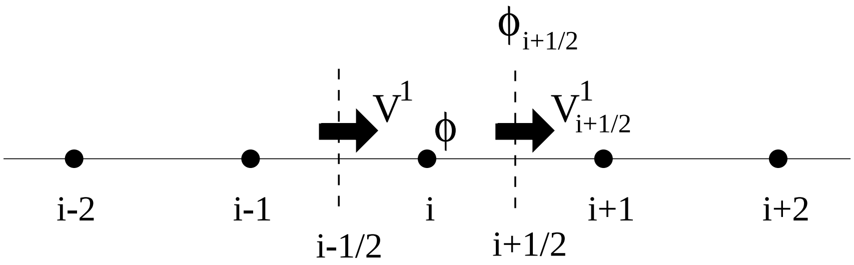


Figure 5.1: One-dimensional staggered grid showing the nodes involved in the evaluation of ϕ at cell-face $i+\frac{1}{2}$.

In this scheme, the face value $\phi_{i+1/2}$ is approximated by means of linear interpolation in the following way:

$$\phi_{i+1/2} = \frac{1}{2}(\phi_i + \phi_{i+1}) \tag{5.22}$$

resulting in second order accuracy. This scheme is obviously conservative and is not positive when a mesh-Péclet number defined as the ratio of the contributions to the convection and the diffusion exceeds a certain value. For example, when orthogonal grids are employed the scheme may produce wiggles when the mesh-Péclet number given by (see (5.5) and (5.4)):

$$Pe^1_{i+1/2} = \frac{\rho_{i+1/2}V^1_{i+1/2}}{2\sqrt{g}_{i+1/2}(K^{11})_{i+1/2}} \tag{5.23}$$

becomes greater than 1 in magnitude. In the case of skewed grids the expression for the mesh-Péclet number depends on how the mixed-derivative terms are approximated. This will be discussed in Section 5.4.

First order upwind difference scheme (UDS).

The face value $\phi_{i+1/2}$ is equal to the upstream node value and, for reasons of efficient vector coding, can be expressed as follows

$$\phi_{i+1/2} = \frac{1}{2}[1 + \mathrm{sign}(V^1_{i+1/2})]\phi_i + \frac{1}{2}[1 - \mathrm{sign}(V^1_{i+1/2})]\phi_{i+1} \tag{5.24}$$

This scheme is conservative, first order accurate, and unconditionally positive. However, when the flow direction is oblique to the grid lines this scheme produces numerical cross-flow diffusion. This may result in a large error in the solution.

Hybrid central/upwind difference scheme (HDS).

In an effort to combine the advantages of both central and upwind schemes Spalding [29] has proposed the hybrid central/upwind difference scheme. The approximation to $\phi_{i+1/2}$ is given by

$$\phi_{i+1/2} = [1 - s(Pe^1_{i+1/2})]\phi_C + s(Pe^1_{i+1/2})\phi_U \tag{5.25}$$

where ϕ_C is given by (5.22), ϕ_U by (5.24) and s is a switching function depending on the mesh-Péclet number Pe with $0 \le s(Pe) \le 1$. With the hybrid scheme the convection terms are approximated with central differences, unless $|Pe| > 1$ when a switch to the first order upwind scheme is applied:

$$s(Pe) = \begin{cases} 0, & |Pe| \le 1 \\ 1, & |Pe| > 1 \end{cases} \tag{5.26}$$

For convergence reasons, a smooth switching function may be used:

$$s(Pe) = 1 - \min(1, \frac{1}{|Pe|}) \tag{5.27}$$

This switching function is obtained by requiring that off-diagonal matrix elements involving convective and diffusive contributions are non-positive, which is sufficient for suppression of wiggles. A disadvantage of the hybrid scheme is that in convection-dominated flows first order upwind is employed everywhere, irrespective of whether spurious wiggles arise or not.

A number of higher order convection schemes have been embedded in the computational procedure. Also, a number of classes of flux-limited schemes based on the TVD methodology will be presented. All may conveniently be written in a canonical form involving a few scheme-related parameters. To this end, the face values $\phi_{i+1/2}$ are approximated by the first order upwind scheme, corrected by adding an appropriate anti-diffusive flux controlled by a limiter. That is, $\phi_{i+1/2}$ is approximated by

$$\phi_{i+1/2} = \begin{cases} \phi_i + \frac{1}{2}\Psi(r^+_{i+1/2})(\phi_i - \phi_{i-1}), & V^1_{i+1/2} > 0 \\ \\ \phi_{i+1} - \frac{1}{2}\Psi(r^-_{i+1/2})(\phi_{i+2} - \phi_{i+1}), & V^1_{i+1/2} < 0 \end{cases} \tag{5.28}$$

where

$$r^+_{i+1/2} = \frac{\phi_{i+1} - \phi_i}{\phi_i - \phi_{i-1}} \quad \text{and} \quad r^-_{i+1/2} = \frac{\phi_{i+1} - \phi_i}{\phi_{i+2} - \phi_{i+1}} \tag{5.29}$$

Formula (5.28) opens the possibility to incorporate arbitrary upwind biased schemes in an algorithmically simple way. The most interesting schemes are

$$\Psi_\kappa(r) = \tfrac{1}{2}(1+\kappa)r + \tfrac{1}{2}(1-\kappa) \qquad \text{Linear } \kappa\text{-scheme}$$

$$\Psi_\Phi(r) = \max[\,0,\, \min(\Phi r,\, 1),\, \min(r,\, \Phi)\,] \qquad \text{Sweby } \Phi\text{-limiter}$$

$$\Psi_\kappa(r) = \begin{cases} (r+|r|)\,\frac{-r^2+(3+\kappa)r-\kappa}{(1+r)^2}, & r \le 1,\quad -1 \le \kappa < 0 \\ \\ \frac{(2+\kappa)r-\kappa}{1+r}, & r \ge 1,\quad -1 \le \kappa < 0 \\ \\ (r+|r|)\,\frac{(1+\kappa)r+1-\kappa}{(1+r)^2}, & 0 \le \kappa \le 1 \end{cases} \qquad \text{R-}\kappa \text{ limiter}$$

$$\Psi_\kappa(r) = \begin{cases} r+|r|, & r \le \frac{1}{3} \\ \\ \frac{(8+9\kappa)r^2+(2-12\kappa)r+2+3\kappa}{3(1+r)^2}, & r \ge \frac{1}{3} \end{cases} \qquad \text{USR-}\kappa \text{ limiter}$$

$$\Psi_{\kappa,\mathrm{M}}(r) = \max[0,\, \min(M,\, \tfrac{1}{2}(1+\kappa)r + \tfrac{1}{2}(1-\kappa),\, 2r)] \qquad \text{PL-}\kappa \text{ limiter}$$

$$\Psi_{\kappa,\mathrm{M}}(r) = \max[0,\, \min(2r, \tfrac{2}{3}),\, \min(M,\, \tfrac{1}{2}(1+\kappa)r + \tfrac{1}{2}(1-\kappa),\, 2r)] \qquad \text{USPL-}\kappa \text{ limiter}$$

$$\begin{aligned} \Psi_\kappa(r) = \max[0,\, \min(2,\, &\tfrac{1}{2}(1+\kappa)r + \tfrac{1}{2}(1-\kappa), \\ &\tfrac{1}{2}(1-\kappa)r + \tfrac{1}{2}(1+\kappa),\, 2r)] \end{aligned} \qquad \text{Symmetric PL-}\kappa \text{ limiter}$$

where $-1 \le \kappa \le 1$, $1 \le \Phi \le 2$ and $1 \le M \le 4$. These classes of flux limiters bring together most limiters known in the literature. For example, $\Phi = 1$ and $\Phi = 2$ define the Minmod and Superbee, respectively. Details can be found in [31, 9]. The functions R-0 and R-$\frac{1}{2}$ are Van Leer [9] and ISNAS [43] limiters. The limited $\kappa = \frac{1}{3}$ scheme proposed by Koren [13] and SMART scheme [6] are identical to PL-$\frac{1}{3}$ with $M = 2$ and PL-$\frac{1}{2}$ with $M = 4$, respectively. Finally, by taking $\kappa = \frac{1}{2}$, $M = 2$ and $\kappa = 0$, $M = 2$ for symmetric PL-κ limiters the UMIST [16] and MUSCL [37] schemes are recovered. Apart from linear κ schemes, USR and USPL limiters, the above classes of limiters reduce locally to first order accuracy at physical extrema regardless of the order of accuracy in regions of monotonicity. If USR and USPL limiters are employed uniform second order accuracy is obtained ($\kappa \neq \frac{1}{3}$). For more details we refer to [44], where further extensive references can be found.

Generally, the function $\Psi(r)$ is nonlinear, and more than 2 immediate neighbouring nodal points per spatial direction may be involved in approximating the convective flux $\phi_{i+1/2}$. Difficulties for iterative solution methods can be circumvented by writing the flux $\phi_{i+1/2}$ in terms of a lower order approximation plus a correction term. This is known as the defect correction technique and was probably first used in the present context by Khosla and Rubin [12]. More specifically, the face value $\phi_{i+1/2}$ is written as

$$\phi_{i+1/2} = \phi^{\mathrm{LOS}}_{i+1/2} + (\phi^{\mathrm{HOS}}_{i+1/2} - \phi^{\mathrm{LOS}}_{i+1/2})^o \tag{5.30}$$

where $\phi^{\mathrm{LOS}}_{i+1/2}$ stands for the approximation by a lower order scheme, for example, first order upwind, and $\phi^{\mathrm{HOS}}_{i+1/2}$ is the higher-order approximation. The term in brackets is evaluated explicitly using the values from the previous time step, which is indicated by the superscript 'o'. It is typically small compared to the implicit part, so that its explicit treatment does not slow down convergence. This approach ensures diagonal dominance for the resulting algebraic equations, thus enhancing iterative rate of convergence while restoring higher-order accuracy at steady-state convergence. The limited anti-diffusive parts of (5.28) may be viewed as deferred corrections to the first order upwind approximation and hence can be put into the source term. Since the stencil associated with first order upwind is maintained, existing codes can easily be modified.

5.4 Diffusion flux approximation

In this section we shall focus on schemes for expressing cell-face derivatives (diffusive flux) in terms of neighbouring nodal values. Special attention will be paid to the mixed derivative terms arising from non-orthogonality of the coordinate system, particularly in view of the demand of positivity of turbulence quantities.

Finite volume discretization of the transport equation (5.3) yields equation (5.5). After substituting (5.4) in (5.5) for all cell faces, the following expression for the diffusive flux at cell face $(i+1/2, j, k)$, for example, results:

$$-\sqrt{g}K^{11}\frac{\partial \phi}{\partial \xi^1}|_{(i+1/2,j,k)} - \sqrt{g}K^{12}\frac{\partial \phi}{\partial \xi^2}|_{(i+1/2,j,k)} - \sqrt{g}K^{13}\frac{\partial \phi}{\partial \xi^3}|_{(i+1/2,j,k)} \tag{5.31}$$

For convenience we restrict ourselves to the cell-face $(i+\frac{1}{2}, j, k)$. The other faces are treated in the same way.

Remark: We assume the mapping $\boldsymbol{x}(\boldsymbol{\xi})$ is smooth, so that $\sqrt{g}$ and $K^{\alpha\beta}$ at cell faces may be approximated by bi- or trilinear interpolations in the obvious way.

If the grid is orthogonal, the last two terms of (5.31) vanish and the first part is approximated with central differences, as follows:

$$\frac{\partial \phi}{\partial \xi^1}|_{(i+1/2,j,k)} = \phi_{(i+1,j,k)} - \phi_{(i,j,k)} \tag{5.32}$$

and no further approximation is required. Clearly, this approximation is conservative and contributes to a positive scheme. The local truncation error is second order. However, when a non-orthogonal grid is used, the approximation of the mixed derivatives may cause spurious wiggles. For example, if we approximate $\partial\phi/\partial\xi^2|_{(i+1/2,j,k)}$ by central differences and bilinear (4-point) interpolation (required to express the ϕ-values at cell vertices in terms of nodal values) as follows:

$$\begin{aligned}\frac{\partial \phi}{\partial \xi^2}|_{(i+1/2,j,k)} &= \phi_{(i+1/2,j+1/2,k)} - \phi_{(i+1/2,j-1/2,k)} \\ &= \frac{1}{4}(\phi_{(i+1,j+1,k)} + \phi_{(i,j+1,k)} - \phi_{(i+1,j-1,k)} - \phi_{(i,j-1,k)})\end{aligned} \tag{5.33}$$

then the coefficients corresponding to $\phi_{(i,j-1,k)}$ and $\phi_{(i+1,j-1,k)}$ get the wrong sign, if $K^{12}_{(i+1/2,j,k)} > 0$. On the other hand, if $K^{12}_{(i+1/2,j,k)} < 0$, we get negative contributions to the coefficients of $\phi_{(i,j+1,k)}$ and $\phi_{(i+1,j+1,k)}$. This does not necessarily result in oscillatory solutions. In fact, these coefficients are usually small relative to the coefficients belonging to the normal derivative $\partial\phi/\partial\xi^1$, but, in some circumstances, for example when the grid is highly non-orthogonal, they become significant and wiggles may occur. Since this is undesirable, we shall consider three methods to address this difficulty.

Method (i).

The most obvious approach is to treat the mixed-derivative terms explicitly, i.e. they are evaluated at the previous time level and are incorporated in the source term. This method is most frequently used in several numerical procedures. It reduces the size of the stencil to a 5-point or 7-point one in 2D and 3D, respectively, and does not lower the order of accuracy in the stationary case. However, this method may cause serious deterioration in the convergence rate, particularly when the grid is highly non-orthogonal, and wiggles may still show up in the steady-state solution.

Method (ii).

In this method, proposed by Demirdžić [4], 2-point rather than 4-point interpolation is employed. For example, suppose that $K^{12}_{(i+1/2,j,k)} > 0$, then the following approximation is taken:

$$\begin{aligned}\frac{\partial \phi}{\partial \xi^2}|_{(i+1/2,j,k)} &= \phi_{(i+1/2,j+1/2,k)} - \phi_{(i+1/2,j-1/2,k)} \\ &= \frac{1}{2}(\phi_{(i,j,k)} + \phi_{(i+1,j+1,k)}) - \frac{1}{2}(\phi_{(i,j-1,k)} + \phi_{(i+1,j,k)})\end{aligned} \tag{5.34}$$

If $K^{12}_{(i+1/2,j,k)} < 0$, we have

$$\frac{\partial \phi}{\partial \xi^2}|_{(i+1/2,j,k)} = \frac{1}{2}(\phi_{(i,j+1,k)} + \phi_{(i+1,j,k)} - \phi_{(i,j,k)} - \phi_{(i+1,j-1,k)}) \tag{5.35}$$

This scheme is symmetric around $(i + 1/2, j, k)$ and thus second order accurate. Similar expressions follow for the last term of (5.31). The scheme is conservative, and produces unconditionally non-negative "corner" coefficients, i.e. those corresponding to $\phi_{(i+1,j-1,k)}$, $\phi_{(i+1,j+1,k)}$, $\phi_{(i+1,j,k-1)}$ and $\phi_{(i+1,j,k+1)}$. But it does not always guarantee positivity, since the contributions to the "principal" coefficients at points $(i, j-1, k)$, $(i+1, j, k)$, $(i, j+1, k)$, $(i, j, k-1)$ and $(i, j, k+1)$ can be either positive or negative. However, the K^{11} terms give contributions to these coefficients of the correct sign, which dominate if K^{12} is not too large.

Method (iii).

All schemes previously considered do not guarantee unconditional positivity of the scheme. In what follows, a new scheme that satisfies positivity will be described. It employs one-sided rather than central differences such that only non-negative coefficients are involved. The mixed derivative with respect to ξ^2 at $(i+1/2, j, k)$ is evaluated as follows:

$$\frac{\partial \phi}{\partial \xi^2}|_{(i+1/2,j,k)} = \begin{cases} \frac{1}{2}\phi_{(i+1,j+1,k)} + \frac{1}{2}\phi_{(i,j+1,k)} - \phi_{(i,j,k)}, & K^{12}_{(i+1/2,j,k)} > 0 \\ \phi_{(i,j,k)} - \frac{1}{2}\phi_{(i+1,j-1,k)} - \frac{1}{2}\phi_{(i,j-1,k)}, & K^{12}_{(i+1/2,j,k)} < 0 \end{cases} \tag{5.36}$$

Obviously, this decreases the accuracy to order one, but the scheme is unconditionally positive. Expressions similar to (5.36) follow for the last term of (5.31). Since the cell-face differences are uniquely defined at each cell face, the scheme is also conservative.

6 Space discretization of the continuity and momentum equations on colocated grids

In this section, we consider the discretization in the interior of the domain. We first present a decoupled approach to solve the Navier-Stokes equations. Then, we describe the momentum equation discretization which is based on the discretization of a transport equation in the staggered case. Adaptions are made for several terms and especially for the pressure gradient. Discretizations due to the boundary conditions are treated in Section 8.

6.1 Decoupled approach

In the colocated cell-center case (from now on, we call colocated case the colocated cell-center case), all the variables are located at the center of the cell. It means that the control volume is the cell itself for all the equations. Except for the pressure equation, the equations for ϕ, where ϕ is u^α or a scalar, have the following form:

$$\frac{\partial c\phi}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial cV^\beta\phi}{\partial\xi^\beta} - \frac{1}{\sqrt{g}}\frac{\partial}{\partial\xi^\beta}(\sqrt{g}a_\delta^{(\beta)}k^{\delta\gamma}\frac{\partial\phi}{\partial x^\gamma}) = g \tag{6.1}$$

c denotes ρ in u^α-equation; $k^{\delta\gamma}$ denotes μ in u^α-equation; g denotes an eventual force in u^α-equation and the second part of the contribution of the stress term which involves $\nabla^T u^\gamma \mid_{\gamma\neq\alpha}$. To take into account the structure of formula (6.1), the decoupled approach is chosen. It means that the terms of the left-hand side of (6.1) are discretized at the current time-step and thus put in the matrix wheras the source terms contained in g are discretized at the previous time-step and put in the right-hand side of the linear system.

6.2 Momentum equation

The discretization is based on the invariant discretization on non-smooth grids (see Section 5.1). The momentum equation is written in vector notation:

$$\frac{\partial\rho\mathbf{u}}{\partial t} + \frac{1}{\sqrt{g}}\frac{\partial\rho V^\gamma\mathbf{u}}{\partial\xi^\gamma} + \nabla p - \frac{1}{\sqrt{g}}\frac{\partial\sqrt{g}a_\beta^{(\gamma)}\tau^\beta}{\partial\xi^\gamma} = \rho\mathbf{f}, \tau^\beta = \mu(\nabla\mathbf{u} + \nabla^T\mathbf{u}). \tag{6.2}$$

The time derivative and the source term $\rho\mathbf{f}$ are discretized over $\Omega_{(0,0)}$ following (5.11), i.e. by the midpoint rule. We remind shortly their expression:

$$\int_{\Omega_{(0,0)}}\frac{\partial\rho\mathbf{u}}{\partial t}d\Omega \approx \sqrt{g}_{(0,0)}\frac{\partial\rho\mathbf{u}}{\partial t}\mid_{(0,0)}, \int_{\Omega_{(0,0)}}\rho\mathbf{f2}d\Omega \approx \sqrt{g}_{(0,0)}\rho\mathbf{f2}\mid_{(0,0)}. \tag{6.3}$$

All quantities are directly available because $\mathbf{u}$ and $\mathbf{f2}$ are known at the center of the cell.

Integration of the convective terms over $\Omega_{(0,0)}$ gives with the divergence theorem:

$$\int_{\Omega_{(0,0)}}\frac{1}{\sqrt{g}}\frac{\partial\rho V^\gamma\mathbf{u}}{\partial\xi^\gamma}d\Omega \approx \rho V^1\mathbf{u}\mid_{(-1,0)}^{(1,0)} + \rho V^2\mathbf{u}\mid_{(0,-1)}^{(0,1)}. \tag{6.4}$$

This expression needs further approximation, because the V^1 and V^2 mass fluxes at the faces of the control volume have to be interpolated. At this moment, linear interpolation is used.

A conservative second order central scheme is obtained with:

$$(V^1\mathbf{u})_{(1,0)} \approx \frac{1}{2}((V^1\mathbf{u})_{(2,0)} + (V^1\mathbf{u})_{(0,0)}). \tag{6.5}$$

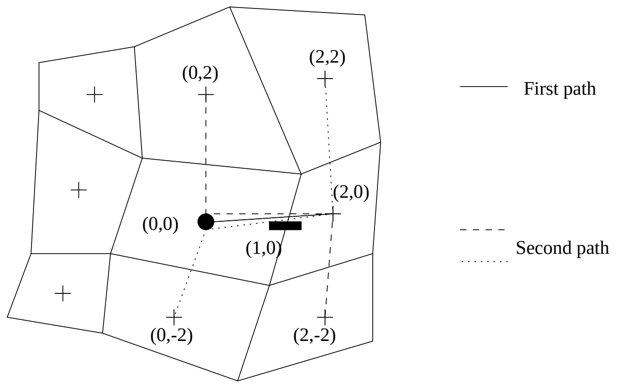


Figure 6.1: Cells involved in the discretization of the stress term

Integration of the stress term over the control volume $\Omega_{(0,0)}$ leads to

$$\int_{\Omega_{(0,0)}} \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} a_\beta^{(\gamma)} \tau^\beta}{\partial \xi^\gamma} d\Omega \approx \sqrt{g} a_\beta^{(1)} \tau^\beta \,|_{(-1,0)}^{(1,0)} + \sqrt{g} a_\beta^{(2)} \tau^\beta \,|_{(0,-1)}^{(0,1)} . \tag{6.6}$$

$(\sqrt{g}\mathbf{a}^{(1)})_{(1,0)}$ is known from the mapping. We rather evaluate ∇u^α than τ^β (see equation 6.2). The discretization of $\nabla^T u^\gamma \,|_{\gamma \neq \alpha}$ is easily deduced from the discretization of ∇u^α. The integration path method [33] is applied to approximate ∇u^α (see Figure 6.1). Chosing a path from $(0,0)$ to $(2,0)$, we have:

$$u^\alpha \,|_{(0,0)}^{(2,0)} = \int_{(0,0)}^{(2,0)} \nabla u^\alpha \cdot d\mathbf{x} \approx (\nabla u^\alpha)_{(1,0)} \cdot \int_{(0,0)}^{(2,0)} d\mathbf{x} \equiv (\nabla u^\alpha)_{(1,0)} \cdot \mathbf{c}_{(1)} \tag{6.7}$$

with

$$\mathbf{c}_{(1)} = \int_{(0,0)}^{(2,0)} d\mathbf{x} = \mathbf{x} \,|_{(0,0)}^{(2,0)} . \tag{6.8}$$

Another equation is necessary to determine ∇u^α. We choose a different path, or rather the average of two paths to take into account the possible non-orthogonality of the mesh, as follows:

$$\frac{1}{4}(u^\alpha \,|_{(0,-2)}^{(0,2)} + u^\alpha \,|_{(2,-2)}^{(2,2)}) = \frac{1}{4}(\int_{(0,-2)}^{(0,2)} + \int_{(2,-2)}^{(2,2)}) \nabla u^\alpha \cdot d\mathbf{x} \approx (\nabla u^\alpha)_{(1,0)} \cdot \mathbf{c}_{(2)}, \tag{6.9}$$

where

$$\mathbf{c}_{(2)} = \frac{1}{4}(\int_{(0,-2)}^{(0,2)} d\mathbf{x} + \int_{(2,-2)}^{(2,2)} d\mathbf{x}) = \frac{1}{4}(\mathbf{x} \,|_{(0,-2)}^{(0,2)} + \mathbf{x} \,|_{(2,-2)}^{(2,2)}). \tag{6.10}$$

The equations (6.7) and (6.9) lead to the following system:

$$\left\{ \begin{array}{l} (\nabla u^\alpha)_{(1,0)} \cdot \mathbf{c}_{(1)} = u^\alpha \,|_{(0,0)}^{(2,0)} \\ (\nabla u^\alpha)_{(1,0)} \cdot \mathbf{c}_{(2)} = \frac{1}{4}(u^\alpha \,|_{(0,-2)}^{(0,2)} + u^\alpha \,|_{(2,-2)}^{(2,2)}) \end{array} \right. . \tag{6.11}$$

Its solution is given by:

$$(\nabla u^{\alpha})_{(1,0)} = u^{\alpha} \,|_{(0,0)}^{(2,0)}\, \mathbf{c}^{(1)} + \frac{1}{4}(u^{\alpha} \,|_{(0,-2)}^{(0,2)} + u^{\alpha} \,|_{(2,-2)}^{(2,2)})\mathbf{c}^{(2)} \tag{6.12}$$

where

$$C = c^{2}_{(2)}c^{1}_{(1)} - c^{1}_{(2)}c^{2}_{(1)}, \mathbf{c}^{(1)} = \frac{1}{C}(c^{2}_{(2)}, -c^{1}_{(2)})^{T}, \mathbf{c}^{(2)} = \frac{1}{C}(-c^{2}_{(1)}, c^{1}_{(1)})^{T}. \tag{6.13}$$

The complete discretization of the stress tensor involves nine cells (Fig. 6.1).

To get an accurate discretization of the pressure gradient, the divergence theorem is applied to the α-th gradient component:

$$\int_{\Omega_{(0,0)}} \frac{1}{\sqrt{g}}(\nabla p)_{\alpha} d\Omega \approx \sqrt{g}a^{(1)}_{\alpha} p \,|^{(1,0)}_{(-1,0)} + \sqrt{g}a^{(2)}_{\alpha} p \,|^{(0,1)}_{(0,-1)} . \tag{6.14}$$

Expressing $p_{(1,0)}$ by linear interpolation does not lead to an accurate discretization over

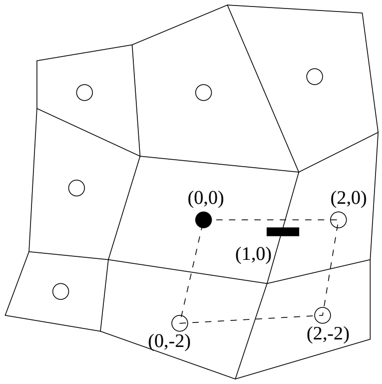


Figure 6.2: Quadrilateral of cell centers containing "$(1,0)$"

nonsmooth grids, because it implies that a five-point molecule is built. On the other hand, bilinear interpolation allows to calculate the pressure at the face accurately. Four cell centers surrounding the cell face center are necessary to build the bilinear function. The first step consists in searching in which quadrilateral of cell centers lays the "(1,0)"-point (Fig. 6.2). Then, the coefficients of the bilinear function have to be calculated. An accurate and quick way to get them is to use the Newton-Raphson method, for instance. Finally, the calculation of $p_{(1,0)}$ is straightforward. The generalization of this process to the other faces of the control volume leads to the use of a nine-point molecule.

6.3 2-D Pressure correction equation

We have seen that the pressure term in the momentum equation can be discretized by bilinear interpolation. This discretization of the pressure gradient is suitable for the momentum equation, but as usual over colocated grids, it allows spurious oscillations when it is used as the cornerstone of the pressure equation. Thus, we apply the Rhie and Chow process. After transformation, the continuous continuity equation reads:

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi^{\beta}}(\sqrt{g}a^{(\beta)}_{\alpha}u^{\alpha}) = 0. \tag{6.15}$$

Integration over the control volume $\Omega_{(0,0)}$ leads to:

$$\int_{\Omega_{(0,0)}} \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\beta} (\sqrt{g} a_\alpha^{(\beta)} u^\alpha) d\Omega \approx \sqrt{g} a_\alpha^{(1)} u^\alpha \mid_{(-1,0)}^{(1,0)} + \sqrt{g} a_\alpha^{(2)} u^\alpha \mid_{(0,-1)}^{(0,1)} = 0. \tag{6.16}$$

The same Cartesian approach consisting in defining an expression coming for the linearized momentum equation at the face of the control volume is made to build $u^\alpha_{(1,0)}$. Because it is inspired from the one followed in the Cartesian strategy, the expression:

$$u^1_{(1,0)} = u^{*1}_{(1,0)} - \frac{\Delta t}{\rho} \frac{1}{\Delta x \Delta y} \int_{\Omega_{(1,0)}} \frac{\partial p'}{\partial x^1} dx^1 dx^2. \tag{6.17}$$

has to be generalized for curvilinear grids. $u^{1*}_{(1,0)}$ is easily obtained by bilinear interpolation. The pressure gradient is required at the face center of the control volume. This configuration is the same as the one encountered in the discretization of the stress term. Thus, the integration path method still fits:

$$\nabla p_{(1,0)} = p \mid_{(0,0)}^{(2,0)} \mathbf{c}^{(1)} + \frac{1}{4} (p \mid_{(0,-2)}^{(0,2)} + p \mid_{(2,-2)}^{(2,2)}) \mathbf{c}^{(2)} \tag{6.18}$$

where $\mathbf{c}^{(1)}$ and $\mathbf{c}^{(2)}$ are the same as in the stress term (Eq. 6.13).
This formulation of the gradient which prevents spurious modes in the pressure field implies that the left-hand side of the pressure equation is the same for the staggered and colocated strategies if the pressure-correction method is used.
Concerning the right-hand side of the pressure equation, the difference between staggered and colocated lays on the use of the bilinear interpolation to get u^*.

7 Turbulence modeling and related numerical issues

7.1 Introduction

This section summarizes all turbulence models and the numerical implementation of these models, which feature in the current ISNaS package. For this, several issues are considered. Our goal is to develop an efficient and unconditionally positive scheme for turbulence equations, so that the non-negativity of the turbulence quantities (e.g., k and ε) is guaranteed.

7.2 Two-equation eddy-viscosity models

Two-equation eddy-viscosity models are based on approximate constitutive equations which predict the unknown Reynolds stress tensor $R^{\alpha\beta}$ that appears in the Reynolds-averaged Navier-Stokes equations (4.1). Through the introduction of a turbulent viscosity μ_t these models relate $R^{\alpha\beta}$ to mean flow variables. The most popular eddy-viscosity model is due to Boussinesq. He postulated that in analogy to molecular diffusion, the Reynolds stresses depend on the deformation rates of the mean flow as follows:

$$R^{\alpha\beta} = \mu_t(g^{\beta\gamma}U^{\alpha}_{,\gamma} + g^{\alpha\gamma}U^{\beta}_{,\gamma}) - \frac{2}{3}\rho k g^{\alpha\beta} \tag{7.1}$$

where k is the turbulent kinetic energy per unit mass, defined as

$$k = \frac{1}{2}g_{\alpha\beta}R^{\alpha\beta} \tag{7.2}$$

The term $\frac{2}{3}\rho k g^{\alpha\beta}$ has been added to ensure that the trace of equation (7.1) produces identical expressions on either side of the equality sign. Furthermore, this term can be absorbed by the pressure gradient term. We then have to replace static pressure by $p + \frac{2}{3}\rho k$. Expressions similar to (7.1) are employed for the turbulent fluxes of heat and mass which are counterparts of Fourier's and Fick's laws for the respective molecular fluxes. The turbulent diffusivities of heat and mass that result are then related directly to μ_t through constant turbulent Prandtl/Schmidt numbers, which are empirically determined and are of the order of unity. The eddy-viscosity μ_t is predicted from the solution of two semi-empirical transport equations for two turbulence quantities to be presented later.

Although the Boussinesq hypothesis works quite well for many flows (i.e. two-dimensional flows), it is too simplistic. More specifically, it assumes that turbulent diffusion is isotropic, so that primary shear stresses will be predicted well, but not secondary shear stresses and normal stresses. As a result the Boussinesq hypothesis may not be suitable for difficult flows involving strong three-dimensional effects. This weakness can potentially be removed using anisotropic eddy-viscosity models. The idea is to extend the stress-strain relationship (7.1) by adding nonlinear elements of the mean velocity gradient tensor. This leads to better approximations of the normal and shear stresses and therefore turbulence anisotropy structure.

Based on series-expansion arguments (details may be found in [7]), a general and coordinate-invariant quadratic relationship between stresses and strains can be written as

$$R^{\alpha\beta} = -\frac{2}{3}\rho k g^{\alpha\beta} + 2\mu_t S^{\alpha\beta} - \frac{4\mu_t^2}{\rho c_\mu k}[c_{\tau 1}(g^{\delta\beta}S^{\alpha\gamma}S_{\gamma\delta} - \frac{1}{3}S^{\delta\varepsilon}S_{\delta\varepsilon}g^{\alpha\beta})$$
$$+c_{\tau 2}(g^{\delta\beta}\Omega^{\alpha\gamma}S_{\gamma\delta} + g^{\delta\alpha}\Omega^{\beta\gamma}S_{\gamma\delta}) + c_{\tau 3}(g^{\delta\beta}\Omega^{\alpha\gamma}\Omega_{\beta\delta} - \frac{1}{3}\Omega^{\delta\varepsilon}\Omega_{\delta\varepsilon}g^{\alpha\beta})] \quad (7.3)$$

where $S^{\alpha\beta}$, $S_{\alpha\beta}$, $\Omega^{\alpha\beta}$ and $\Omega_{\alpha\beta}$ are the contravariant and covariant components of the mean rate of strain and rotation tensors, respectively, viz.,

$$S^{\alpha\beta} = \frac{1}{2}(g^{\beta\gamma}U^{\alpha}_{,\gamma} + g^{\alpha\gamma}U^{\beta}_{,\gamma}), \quad S_{\alpha\beta} = \frac{1}{2}(U_{\alpha,\beta} + U_{\beta,\alpha}) \quad (7.4)$$

$$\Omega^{\alpha\beta} = \frac{1}{2}(g^{\beta\gamma}U^{\alpha}_{,\gamma} - g^{\alpha\gamma}U^{\beta}_{,\gamma}), \quad \Omega_{\alpha\beta} = \frac{1}{2}(U_{\alpha,\beta} - U_{\beta,\alpha}) \quad (7.5)$$

Note that the quadratic products of strain and rotation tensors are grouped so that the resultant is symmetric in α and β. Approach has been to determine the closure coefficients c_μ, $c_{\tau 1}$, $c_{\tau 2}$ and $c_{\tau 3}$ so that agreement is achieved with a simple shear flow and with one other difficult class of flow. Several researchers have proposed anisotropic eddy-viscosity models (AEVM). These models in the current version of ISNaS include: the nonlinear AEVM of Speziale [30], the RNG based nonlinear AEVM of Rubinstein and Barton [22], the nonlinear variant of Nisizima and Yoshizawa arising from the Kraichnan's DIA theory [19] and the anisotropic eddy-viscosity closure of Myong and Kasagi [18]. Although the theoretical origins and derivations of these eddy-viscosity models differ greatly, they can be cast into a common mathematical form, like (7.3). They merely differ in respect of the numerical values of the closure constants, as given in Table 7.1. It should be noted that the Boussinesq hypothesis is

AEVM	c_μ	$c_{\tau 1}$	$c_{\tau 2}$	$c_{\tau 3}$
Speziale	0.09	0.1512	0.1512	0
Rubinstein-Barton	0.085	0.68	0.14	-0.56
Nisizima-Yoshizawa	0.09	-0.7881	0.1769	1.0675
Myong-Kasagi	0.09	0.275	0.2375	0.05

Table 7.1: Numerical values of closure constants in stress-strain relation.

obtained by setting $c_{\tau 1}$, $c_{\tau 2}$ and $c_{\tau 3}$ to zero.

The turbulent viscosity μ_t must be specified by the two-equation model. Two-equation models are turbulence models in which the evolution of both the velocity and length scales characterizing turbulent motion is obtain by solving semi-empirical partial differential equations. At the moment four two-equation models are dealt with: the standard k-ε model of Launder and Spalding [15], the RNG based k-ε variant of Yakhot *et al.* [42], the extended k-ε model of Chen and Kim [3] and Wilcox's k-ω closure [40].

If the k-ε modeling framework is employed, the turbulent viscosity is related to k and ε, the dissipation rate of k, through the following semi-empirical expression:

$$\mu_t = \rho c_\mu \frac{k^2}{\varepsilon} \quad (7.6)$$

where k and ε are determined from the coordinate-invariant semi-empirical transport equations

$$\frac{\partial \rho k}{\partial t} + (\rho U^\alpha k)_{,\alpha} - ((\mu + \frac{\mu_t}{\sigma_k})g^{\alpha\beta}k_{,\beta})_{,\alpha} = P_k - \rho\varepsilon \quad (7.7)$$

$$\frac{\partial \rho\varepsilon}{\partial t} + (\rho U^\alpha \varepsilon)_{,\alpha} - ((\mu + \frac{\mu_t}{\sigma_\varepsilon}) g^{\alpha\beta} \varepsilon_{,\beta})_{,\alpha} = \frac{\varepsilon}{k}(c^*_{\varepsilon 1} P_k - c_{\varepsilon 2}\rho\varepsilon) \tag{7.8}$$

where P_k is the production rate of turbulent energy given by

$$P_k = g_{\alpha\gamma} R^{\alpha\beta} U^\gamma_{,\beta} \tag{7.9}$$

Different variants of the above model arise from the different approaches to determining the model coefficients c_μ, σ_k, σ_ε, $c^*_{\varepsilon 1}$ and $c_{\varepsilon 2}$. The coefficient $c^*_{\varepsilon 1}$ is made a function of the equilibrium state of turbulence which is characterizes by the ratio of times scales of turbulence and mean strain, denoted as $\eta = Sk/\varepsilon$. Here $S = (2S^{\alpha\beta}S_{\alpha\beta})^{\frac{1}{2}}$ is the magnitude of the mean rate of strain. It should be noted that if the production rate of turbulent energy (7.9) is modeled using the Boussinesq hypothesis, then we obtain a simple expression for η, namely $\eta = \sqrt{P_k/\rho c_\mu \varepsilon}$. In the above three k-ε models $c^*_{\varepsilon 1}$ is obtained as follows:

$$c^*_{\varepsilon 1} = \begin{cases} c_{\varepsilon 1}, & \text{standard model} \\ c_{\varepsilon 1} - \frac{\eta(1-\eta/\eta_0)}{1+\gamma\eta^3}, & \text{RNG model} \\ c_{\varepsilon 1} + c_{\varepsilon 3} c_\mu \eta^2, & \text{extended model} \end{cases} \tag{7.10}$$

where $\gamma = 0.012$ and $\eta_0 = 4.38$. The numerical values used in the three k-ε variants for the closure constants are tabulated below. The extended model in ISNaS employs slightly revised

k-ε variant	c_μ	σ_k	σ_ε	$c_{\varepsilon 1}$	$c_{\varepsilon 2}$	$c_{\varepsilon 3}$
Standard	0.09	1.0	1.3	1.44	1.92	-
RNG	0.085	0.7179	0.7179	1.42	1.68	-
Extended	0.09	0.75	1.15	1.35	1.9	0.05

Table 7.2: Numerical values of closure constants in k-ε closure.

values for coefficients $c_{\varepsilon 1}$ and $c_{\varepsilon 3}$. Chen and Kim [3] recommended $c_{\varepsilon 1} = 1.15$ and $c_{\varepsilon 3} = 0.25$ which produce significantly wrong solutions over a wide range of flows. In [8], it was reported that this model give consistently better results, when $c_{\varepsilon 1} = 1.35$ and $c_{\varepsilon 3} = 0.05$.

The three k-ε variants are of the high-Reynolds-number type and the viscosity-affected near wall region is resolved with a low-Reynolds-number model according to Lam and Bremhorst [14]. An alternative and still widely employed approach is the use of so-called wall functions. This issue will be presented in Section 8. When the low-Reynolds-number k-ε model is used the values of the closure constants c_μ, $c_{\varepsilon 1}$, $c_{\varepsilon 2}$, σ_k and σ_ε remain the same and the viscous damping functions are introduced into the constants, as follows:

$$c_\mu \leftarrow f_\mu c_\mu, \quad c_{\varepsilon 1} \leftarrow f_1 c_{\varepsilon 1}, \quad c_{\varepsilon 2} \leftarrow f_2 c_{\varepsilon 2} \tag{7.11}$$

The damping functions are chosen according to the model proposed by Lam and Bremhorst [14]:

$$f_\mu = (1 - e^{-0.0165 R_y})^2 (1 + \frac{20.5}{Re_{\mathrm{T}}}) \tag{7.12}$$

$$f_1 = 1 + (\frac{0.05}{f_\mu})^3 \tag{7.13}$$

$$f_2 = 1 - e^{-Re_T^2} \tag{7.14}$$

with

$$R_y = \frac{\rho\sqrt{k}Y}{\mu}, \quad Re_{\mathrm{T}} = \frac{\rho k^2}{\varepsilon\mu} \tag{7.15}$$

the local and turbulent Reynolds numbers, respectively. Furthermore, Y is the normal distance to the solid boundary. Boundary conditions for the momentum, k and ε equations are, respectively,

$$\boldsymbol{u} = 0, \quad k = 0, \quad \varepsilon = \nu\frac{\partial^2 k}{\partial Y^2}|_{Y=0} \tag{7.16}$$

For computational expediency, we choose

$$\frac{\partial\varepsilon}{\partial Y} = 0 \quad \text{at} \quad Y = 0 \tag{7.17}$$

as the boundary condition for the dissipation rate, as suggested by Lam and Bremhorst [14]. The disadvantage of the low-Reynolds-number models is that they impose a very fine mesh normal to the wall, which can be prohibitive when dealing with large 3D applications.

Many proposals for the k-ω model have been made (for a review, see [41]). The version devised by Wilcox [40] is perhaps the most popular one and will be presented here. Rather than solving for the dissipation rate of turbulent energy, the second variable considered here is the specific dissipation rate, i.e. the rate of dissipation per unit kinetic energy. This variable is denoted as ω ($\equiv \varepsilon/k$). On the basis of a simple dimensional analysis, the eddy-viscosity is taken to be the quotient of the turbulent energy and the specific dissipation rate, thus:

$$\mu_t = \rho\frac{k}{\omega} \tag{7.18}$$

The two turbulent parameters obey the following modeled transport equations:

$$\frac{\partial\rho k}{\partial t} + (\rho U^\alpha k)_{,\alpha} - ((\mu + \sigma^*\mu_t)g^{\alpha\beta}k_{,\beta})_{,\alpha} = P_k - \beta^*\rho k\omega \tag{7.19}$$

$$\frac{\partial\rho\omega}{\partial t} + (\rho U^\alpha \omega)_{,\alpha} - ((\mu + \sigma\mu_t)g^{\alpha\beta}\omega_{,\beta})_{,\alpha} = \alpha\frac{\omega}{k}P_k - \beta\rho\omega^2 \tag{7.20}$$

where P_k is given by (7.9). The closure constants employed are as follows:

$$\alpha = \frac{5}{9}, \quad \beta = \frac{3}{40}, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{1}{2} \tag{7.21}$$

The k-ω model is a low-Reynolds-number closure which means that it can be integrated through the viscous sub-layer without requiring a near-wall model. Hence, standard boundary conditions must be employed at a solid wall, i.e. for the momentum equations noslip conditions are imposed on the boundary and k is simply zero on the wall. Due to the singular behaviour of ω at the wall, a special boundary condition for ω must be used, which is given by

$$\lim_{Y\to 0} \omega = \frac{N_\omega \nu}{Y^2} \tag{7.22}$$

where

$$N_\omega = \begin{cases} 6/\beta, & \text{without viscous corrections} \\ 2/\beta^*, & \text{with viscous corrections.} \end{cases} \tag{7.23}$$

The singularity at $Y = 0$ does not allow this boundary condition to be imposed on the wall. The numerical treatment of this singularity will be presented in the next section.

For stagnation flows, the so-called Kato-Launder modification [10], which replaces the strain in the production of turbulent energy term by the vorticity, has been implemented. Using the Boussinesq eddy-viscosity approximation, we obtain

$$P_k = \mu_t S^2 \tag{7.24}$$

In a stagnation flow, the very high levels of S produce excessive levels of turbulent energy whereas deformation near stagnation point is nearly irrotational. Defining the magnitude of the mean rotation as $\Omega = (2\Omega^{\alpha\beta}\Omega_{\alpha\beta})^{\frac{1}{2}}$ and replacing (7.24) by

$$P_k = \mu_t S\Omega \tag{7.25}$$

leads to a marked reduction in energy production near the stagnation point, while having no effect in a simple shear flow [10]. In [2] a hybrid form is proposed in which (7.24) and Kato-Launder correction (7.25) are averaged:

$$P_k = \mu_t S((1-\alpha)S + \alpha\Omega) \tag{7.26}$$

with $0 \leq \alpha \leq 1$ a weight factor. This hybrid model is particularly used for stagnation flows. In that case the weight factor is chosen to be $\alpha = 0.85$, as recommended by [2].

7.3 Numerical aspects of two-equation eddy-viscosity modeling

7.3.1 Space discretization of the stress-strain model

The anisotropic eddy-viscosity model contains nonlinear elements. Difficulties for iterative solution methods can be circumvented by writing the stress term $R^{\alpha\beta}$ in terms of the implicit Boussinesq formulation plus an explicit quadratic term. This can be regarded as the well-known defect correction technique. Hence, we define the extra turbulent stress $Q^{\alpha\beta}$ for the nonlinear model as the difference between the total stress $R^{\alpha\beta}$ and the linear model contribution,

$$Q^{\alpha\beta} = R^{\alpha\beta} + \frac{2}{3}\rho k g^{\alpha\beta} - 2\mu_t S^{\alpha\beta} \tag{7.27}$$

which is explicitly calculated at every time step, using the values from the previous time step. The linear model is treated implicitly, which means that the laminar viscosity μ is increased by the turbulent viscosity μ_t in the deviatoric stress term (4.2). The right-hand side of the momentum equations is updated with the extra turbulent stress, as follows

$$\rho f^\alpha \leftarrow \rho f^\alpha + Q^{\alpha\beta}_{,\beta} \tag{7.28}$$

Taken $\alpha = 1$ as an example, the finite volume discretization results in

$$\begin{aligned}
&\int_{\Omega_{i+1/2,j,k}} (\rho f^1 + \frac{1}{\sqrt{g}}\frac{\partial\sqrt{g}Q^{1\beta}}{\partial\xi^\beta} + \{{1 \atop \beta\gamma}\}Q^{\beta\gamma})d\Omega = \int_{\Omega_{i+1/2,j,k}} (\rho f^1 + \{{1 \atop \beta\gamma}\}Q^{\beta\gamma})d\Omega + \\
&\int_{G_{i+1/2,j,k}} \frac{\partial\sqrt{g}Q^{1\beta}}{\partial\xi^\beta} d\xi^1 d\xi^2 d\xi^3 \approx \sqrt{g}(\rho f^1 + \{{1 \atop \beta\gamma}\}Q^{\beta\gamma})|_{(i+1/2,j,k)} + \\
&\sqrt{g}Q^{11}|_{(i,j,k)}^{(i+1,j,k)} + \sqrt{g}Q^{12}|_{(i+1/2,j-1/2,k)}^{(i+1/2,j+1/2,k)} + \sqrt{g}Q^{13}|_{(i+1/2,j,k-1/2)}^{(i+1/2,j,k+1/2)}
\end{aligned} \tag{7.29}$$

The discretization is completed by substituting (7.27) in (7.29), followed by central differences and bilinear interpolations. Due to the covariant derivatives of the contravariant velocity components in (7.27) extra Christoffel symbols are introduced in the discretized right-hand side of the momentum equations. At this moment, the above discretization is only carried out in the inner region.

7.3.2 Space discretization of the production term in two-equation models

One of the source terms in turbulence model equations is the production rate of turbulent energy given by (7.9). The discretization of this term is carried out at center (i, j, k) with central differences and bilinear interpolations in which the fewest number of neighbouring nodal points are taken. Since we use $V^\alpha = \sqrt{g}U^\alpha$ as unknowns, the covariant derivative of the contravariant velocity components must be expressed in terms of flux components. We have

$$U^\alpha_{,\beta} = \frac{1}{\sqrt{g}}\left(\frac{\partial V^\alpha}{\partial \xi^\beta} - \left\{ {\gamma \atop \gamma\beta} \right\} V^\alpha + \left\{ {\alpha \atop \beta\gamma} \right\} V^\gamma\right) \tag{7.30}$$

The partial derivative of the flux component can be approximated by central differences. The same interpolation rules as for the momentum equations are applied. All geometrical quantities are evaluated at the centre of a scalar cell. Closest to a boundary, some derivatives $\partial V^\alpha / \partial \xi^\beta$ also contain virtual fluxes. These virtual quantities are expressed in internal fluxes by using linear extrapolation. In two-dimensional case, for example, at lower boundary we get:

$$V^1_{i,-2} = 2V^1_{i,0} - V^1_{i,2} \tag{7.31}$$

The above discretization is not well-suited when a non-smooth grid is employed. An approach to discretize the production term in case of non-smooth grids is to integrate the Cartesian expression of P_k over a finite volume Ω_{ijk} so that no Christoffel symbols or metric tensors occur in the formulation. At this moment we restrict ourselves to the Boussinesq hypothesis for the modeling of the production of turbulent energy. From (7.24) it follows that the Cartesian expression for P_k is given by

$$P_k = \mu_t\left(\frac{\partial u_\alpha}{\partial x_\beta} + \frac{\partial u_\beta}{\partial x_\alpha}\right)\frac{\partial u_\alpha}{\partial x_\beta} \tag{7.32}$$

The remain task is to discretize the partial derivatives of the Cartesian velocity components with respect to $\boldsymbol{x}$ at point (i, j, k). This can be done with the integration-path method. Here a "quick" approach is given. To approximate the x^β-derivative of u^α this derivative has to be expressed in terms of the derivative with respect to $\boldsymbol{\xi}$. Using the chain rule, one gets:

$$\frac{\partial u^\alpha}{\partial x^\beta} = \frac{\partial \xi^\gamma}{\partial x^\beta}\frac{\partial u^\alpha}{\partial \xi^\gamma} = a^{(\gamma)}_\beta \frac{\partial u^\alpha}{\partial \xi^\gamma} \tag{7.33}$$

The approximation of (7.33) at point (i, j, k) leads to

$$\frac{\partial u^\alpha}{\partial x^\beta}\Big|_{(i,j,k)} \approx a^{(\gamma)}_\beta\Big|_{(i,j,k)} \Delta_\gamma u^\alpha\Big|_{(i,j,k)} \tag{7.34}$$

Here $\Delta_\gamma u^\alpha|_{(i,j,k)}$ represents the difference in u^α, across the cell enclosing point (i, j, k), in ξ^γ direction. The differences are evaluated as:

$$\begin{aligned}
\Delta_1 u^\alpha|_{(i,j,k)} &= u^\alpha_{(i+1/2,j,k)} - u^\alpha_{(i-1/2,j,k)}, \\
\Delta_2 u^\alpha|_{(i,j,k)} &= u^\alpha_{(i,j+1/2,k)} - u^\alpha_{(i,j-1/2,k)}, \\
\Delta_3 u^\alpha|_{(i,j,k)} &= u^\alpha_{(i,j,k+1/2)} - u^\alpha_{(i,j,k-1/2)}
\end{aligned} \tag{7.35}$$

The final expression of the approximation for

$$\nabla u^\alpha = \left(\frac{\partial u^\alpha}{\partial x^1}, \frac{\partial u^\alpha}{\partial x^2}, \frac{\partial u^\alpha}{\partial x^3} \right)^\top \tag{7.36}$$

becomes

$$\nabla u^\alpha|_{(i,j,k)} \approx \boldsymbol{a}^{(1)}|_{(i,j,k)} u^\alpha|^{(i+1/2,j,k)}_{(i-1/2,j,k)} + \boldsymbol{a}^{(2)}|_{(i,j,k)} u^\alpha|^{(i,j+1/2,k)}_{(i,j-1/2,k)} + \boldsymbol{a}^{(3)}|_{(i,j,k)} u^\alpha|^{(i,j,k+1/2)}_{(i,j,k-1/2)} \tag{7.37}$$

The velocity components on cell faces are to be obtained with

$$\boldsymbol{u} = \frac{\boldsymbol{a}_{(\alpha)} V^\alpha}{\sqrt{g}} \tag{7.38}$$

Since $\boldsymbol{u}$ is only given at center of cell faces, discontinuous geometric quantities and fluxes have to be replaced by suitable definitions such that (7.38) is exact for constant $\boldsymbol{u}$ on arbitrary grids. An example:

$$\boldsymbol{u}_{(i,j+1/2,k)} = \frac{\boldsymbol{a}_{(1)} V^1 + \boldsymbol{a}_{(2)} V^2 + \boldsymbol{a}_{(3)} V^3}{\sqrt{g}}\Big|_{(i,j+1/2,k)} \tag{7.39}$$

with

$$\boldsymbol{a}_{(2)}|_{(i,j+1/2,k)} \equiv \frac{1}{4}(\boldsymbol{a}_{(2)}|_{(i-1/2,j,k)} + \boldsymbol{a}_{(2)}|_{(i-1/2,j+1,k)} + \boldsymbol{a}_{(2)}|_{(i+1/2,j,k)} + \boldsymbol{a}_{(2)}|_{(i+1/2,j+1,k)}) \tag{7.40}$$

$$V^1_{(i,j+1/2,k)} \equiv \frac{1}{4}(V^1_{(i-1/2,j,k)} + V^1_{(i-1/2,j+1,k)} + V^1_{(i+1/2,j,k)} + V^1_{(i+1/2,j+1,k)}) \tag{7.41}$$

7.3.3 2D implementation of low-Reynolds-number modeling

When implementing the Lam-Bremhorst model, a serious aspect to consider is the near-wall behaviour of the damping functions. For example, consider the damping function

$$f_\mu = (1 - e^{-0.0165 R_y})^2 \, (1 + \frac{20.5}{Re_{\mathrm{T}}}) \tag{7.42}$$

Approaching the wall, desired asymptotic behaviour depends upon accurate values of this function. Using Taylor series expansions, this damping function can be written as

$$f_\mu = \begin{cases} (1 - e^{-0.0165 R_y})^2 \, (1 + \frac{20.5}{Re_{\mathrm{T}}}), & R_y > 1.0 \\ \\ (0.0165 R_y)^2 \, (1 + \frac{20.5}{Re_{\mathrm{T}}}), & R_y \leq 1.0 \end{cases} \tag{7.43}$$

Furthermore, the limiting form of f_μ should be asymptotically consistent with the near-wall behaviour of time-averaged properties. For example, it is well known that the asymptotic variation of turbulent energy and dissipation rate near the wall are

$$k \sim \frac{1}{2} Y^2, \quad \varepsilon \sim \nu \quad \text{as} \quad Y \to 0 \tag{7.44}$$

Hence, we have

$$R_y \sim \frac{\sqrt{2} Y^2}{2\nu}, \quad Re_{\mathrm{T}} \sim \frac{Y^4}{4\nu^2} \quad \text{as} \quad Y \to 0 \tag{7.45}$$

Substituting in (7.42) gives

$$\lim_{Y \to 0} f_\mu = 0.01116225 \tag{7.46}$$

Because the order of the leading term of Re_{T} is larger than that of R_y, it sometimes happen, numerically, that f_μ is relatively large even when $Y \to 0$. Therefore, if $Re_{\mathrm{T}} < 1$ and $f_\mu > 1$, f_μ should be set to its limit value. In order to avoid very small (or negative) values of k in regions in which μ_t becomes very small, the implementation of the damping function f_2 is slightly modified: f_2 should not be smaller than 0.01.

The calculation of the normal distance from a grid point in the inner region to the wall, i.e. Y, will now be described. We consider the following situation as depicted in Figure 7.1. The distance of a cell-center point P from a boundary surface can be found as the scalar

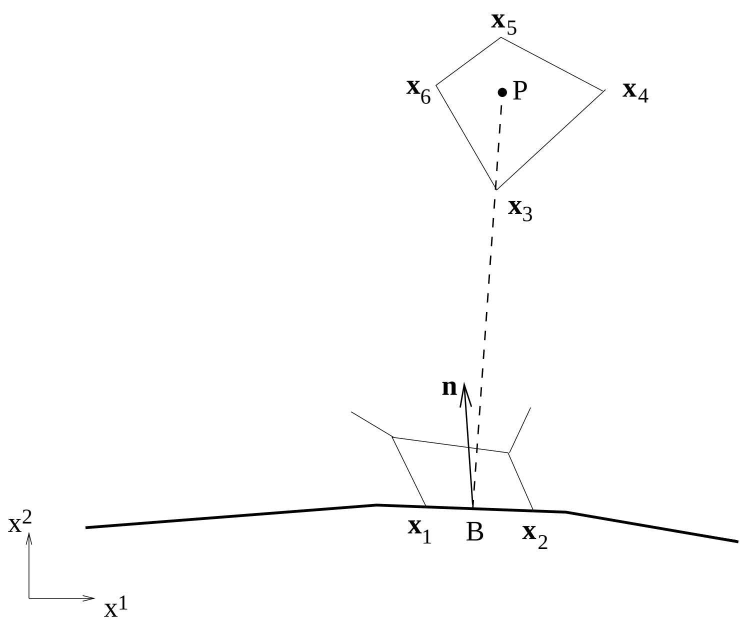


Figure 7.1: Calculation of normal distance between node P and boundary point B

product of a vector connecting a boundary point B and P and the unit normal vector $\boldsymbol{n}$:

$$Y_{\mathrm{P}} = \boldsymbol{n} \cdot \boldsymbol{BP} = \boldsymbol{n} \cdot (\boldsymbol{x}_{\mathrm{P}} - \boldsymbol{x}_{\mathrm{B}}) \tag{7.47}$$

The index numbers of the cell containing P and the near-wall volume including B are identical. The co-ordinates of B and P are obtained from the co-ordinates of cell vertices by linear interpolations, hence

$$\boldsymbol{x}_{\mathrm{B}} = \frac{1}{2}(\boldsymbol{x}_1 + \boldsymbol{x}_2), \quad \boldsymbol{x}_{\mathrm{P}} = \frac{1}{4}(\boldsymbol{x}_3 + \boldsymbol{x}_4 + \boldsymbol{x}_5 + \boldsymbol{x}_6) \tag{7.48}$$

The unit normal vector is computed as follows:

$$\boldsymbol{n} = (-t_2, t_1)^\top, \quad \boldsymbol{t} = \frac{\boldsymbol{x}_2 - \boldsymbol{x}_1}{||\boldsymbol{x}_2 - \boldsymbol{x}_1||} \tag{7.49}$$

In the presence of several walls - the usual case - Y_{P} is, in some sense, a weighted average of distances to all points on the solid boundary seen by the point at which Y is to be computed.

There is no consensus, however, on how Y should be computed in general. A simple approach, adopted in complicated geometrical domains, is to take Y as the distance to the nearest wall.

In the case of k-ω model, standard boundary conditions must be employed at a solid wall, i.e. for the momentum equations noslip conditions are imposed on the boundary, whereas for turbulent energy k a homogeneous Dirichlet condition holds. However, in order to avoid non-positive values of k, $k = 10^{-6}$ may be taken as boundary condition on the wall. In order to impose condition (7.22), the equation for ω is solved up to the second grid point closest to the wall, applying the condition

$$\omega = \frac{N_\omega \nu}{Y_{\mathrm{P}}^2}, \qquad Y_{\mathrm{P}}^+ < 5.0 \tag{7.50}$$

at the first grid point P above the surface. Here, N_ω is given by (7.23), $Y^+ = \rho u_\tau Y/\mu$ is the nondimensional distance from the wall and $u_\tau = \sqrt{\tau_w/\rho}$ is the friction velocity with τ_w the wall shear stress.

7.3.4 Implementation of the positive scheme for two-equation models

The turbulence model equations, as discussed in the previous section, are transport equations. Hence, for the implementation of the two-equation models the equation (5.2) will be considered. The evaluation of the functions c^*, $K^{\alpha\beta}$, D and f^* for each turbulence equations must result in a positive scheme, which means that the solution of turbulence quantities assumes non-negative. This positivity consideration is urged because of our desire to develop efficient and robust method for turbulent transport equations. See also [44]. It should be mentioned that in case of space discretization the positivity of the turbulence quantities is guaranteed via TVD constraints on the convection scheme, as has been discussed in Section 5.3.

The turbulence source term has a considerable impact on the time integration. It turns out that we are compelled to use standard Newton linearization in order to avoid the possibility of negative solution of turbulence quantities. As an example, we consider the equation for turbulent kinetic energy in the high-Reynolds-number standard k-ε model where the source term is

$$P_k - \rho\varepsilon = 2\rho c_\mu \frac{k^2}{\varepsilon} G - \rho\varepsilon \tag{7.51}$$

with $G = S^{\alpha\beta} S_{\alpha\beta}$. The treatment of the quadratic term in the production rate of turbulent energy, due to the use of an anisotropy model, will be discussed later. The eddy-viscosity and consequently the production term will be frozen at time level n, whereas the dissipation term will be treated implicitly and linearized as follows:

$$-\rho\varepsilon^{n+1} \approx -\rho^2 c_\mu \frac{(k^{n+1})^2}{\mu_t^n} \approx -\frac{2\rho^2 c_\mu k^n}{\mu_t^n} k^{n+1} + \rho^2 c_\mu \frac{(k^n)^2}{\mu_t^n} = \rho\varepsilon^n - 2\rho\frac{\varepsilon^n}{k^n} k^{n+1} \tag{7.52}$$

where n denotes the preceding time level and $n+1$ the new time level. The functions D and f^* becomes

$$D = 2\rho\frac{\varepsilon^n}{k^n}, \quad f^* = P_k^n + \rho\varepsilon^n \tag{7.53}$$

The other functions are given by

$$c^* = \rho, \quad K^{\alpha\beta} = g^{\alpha\beta}(\mu + \frac{\mu_t^n}{\sigma_k}) \tag{7.54}$$

The same holds for the equation for dissipation rate. We have

$$c^* = \rho, \quad K^{\alpha\beta} = g^{\alpha\beta}(\mu + \frac{\mu_t^n}{\sigma_\varepsilon}), \quad D = 2\rho c_{\varepsilon 2}\frac{\varepsilon^n}{k^n}, \quad f^* = c_{\varepsilon 1}\frac{\varepsilon^n}{k^n}P_k^n + \rho c_{\varepsilon 2}\frac{(\varepsilon^n)^2}{k^n} \tag{7.55}$$

An analogous procedure is followed in respect of the k-ω model.

So far, we have considered the standard k-ε model. The extra term in the RNG dissipation rate equation should also be treated appropriately. Consider the term

$$-\frac{\eta(1-\eta/\eta_0)}{1+\gamma\eta^3}\,\frac{\varepsilon}{k}\,P_k \tag{7.56}$$

then by replacing the functions D and f^*, respectively, with

$$D + \frac{\eta P_k^n}{k^n(1+\gamma\eta^3)}, \quad f^* + \frac{\eta^2\varepsilon^n P_k^n}{\eta_0 k^n(1+\gamma\eta^3)} \tag{7.57}$$

and $\eta = \sqrt{P_k^n/\rho c_\mu \varepsilon^n}$, it can be verified that the right-hand side of the discrete equation for ε and the coefficient of ε^{n+1} consist of positive contributions.

Physically, the production rate of turbulent energy is always non-negative. The problem, however, is that, when employing an anisotropic model, there is no guarantee that P_k is non-negative numerically. Hence, special measures are taken to ensure that at no stage of the time stepping the production rate assume negative values. By virtue of (7.9), (7.3) and (7.27) one sees that

$$P_k = 2\mu_t S^{\alpha\beta}S_{\alpha\beta} + g_{\alpha\gamma}Q^{\alpha\beta}U^\gamma_{,\beta} \tag{7.58}$$

The first term in the right-hand side of (7.58) is always non-negative and hence, can be contributed to f^*, as explained before. The second term would normally be expected to contribute to f^*. However, it must only do so if it is non-negative, otherwise it must be allocated to D. This can be done in the following way: this negative term is first divided by the value of k available from the previous time level and then added to D. Algebraically, this is implemented through,

$$f^* := f^* + \max(0, g_{\alpha\gamma}Q^{\alpha\beta}U^\gamma_{,\beta})$$

$$D := D - \frac{\min(0, g_{\alpha\gamma}Q^{\alpha\beta}U^\gamma_{,\beta})}{k^n} \tag{7.59}$$

An analogous procedure is followed in respect of the dissipation rate equation.

8 Implementation of the boundary conditions

In the present version of the ISNaS incompressible program, the following types of boundary conditions have been implemented.

Boundary conditions for the momentum equations:

Type 1: Velocity prescribed (Dirichlet boundary condition)

Type 2: Stress prescribed (Natural boundary condition)

Type 3: Normal stress and tangential velocity given (Semi natural flow)

Type 4: Tangential stress and normal velocity given (Slip boundary condition and also symmetry condition)

Boundary conditions for the transport equation are:

Type 1: Scalar ϕ prescribed (Dirichlet boundary condition)

Type 2: $\sigma\phi + (k\nabla\phi) \cdot \boldsymbol{n}$ prescribed (Robbins boundary condition)

In the next paragraphs we consider the various boundary conditions separately.

In 2D the notion normal and tangential vector have been defined in a somewhat strange way. For a user the normal vector is defined as the outward normal in the case of a counterclockwise direction and as the inward normal in the case of a clockwise direction. The tangential vector is defined in the direction of the outer boundary. In the program, however, the normal and tangential vector are always defined in the 1 or 2 direction in the computational grid. Hence at the boundary $\xi^2 = 0$, the normal direction is the ξ^2 direction and tangential direction the ξ^1 direction. At the boundary $\xi^1 = nx$, the normal direction is ξ^1 and the tangential direction ξ^2 etc.
In the sequel the internal definition will be used.

8.1 Prescribed velocities

8.1.1 2D implementation

In 2D, prescribed velocity given means in the present version $\boldsymbol{u} \cdot \boldsymbol{n}$ and $\boldsymbol{u} \cdot \boldsymbol{t}$ given. These quantities are transformed to contravariant components using the formulae (6.1) and (6.4) from Van Kan *et al.* (1991):

$$U^n = \sqrt{g^{nn}}\boldsymbol{u} \cdot \boldsymbol{n} \tag{8.1}$$

$$U^t = \frac{1}{g_{tt}}(\sqrt{g_{tt}}\boldsymbol{u} \cdot \boldsymbol{t} - g_{nt}U^n) \tag{8.2}$$

Here the definitions of $\boldsymbol{n}$ and $\boldsymbol{t}$ as described above are meant both for the physical components as for the computational components.

The given normal velocity component (in computational space) is implemented by explicitly prescribing the velocity unknown at the boundary. In the program this is implemented by setting the corresponding main diagonal element equal to 1 and the off-diagonal elements in the corresponding rows to 0. The right-hand-side component corresponding to this unknown is made equal to the unknown itself.

The given velocity component U^t is implemented in the following way:

The matrix is built for all unknowns including all the "tangential" unknowns. The rows corresponding to the "tangential" unknowns closest to the boundary (see Figure 8.1) contain elements referring to virtual pressures and virtual "tangential" velocity unknowns. The virtual

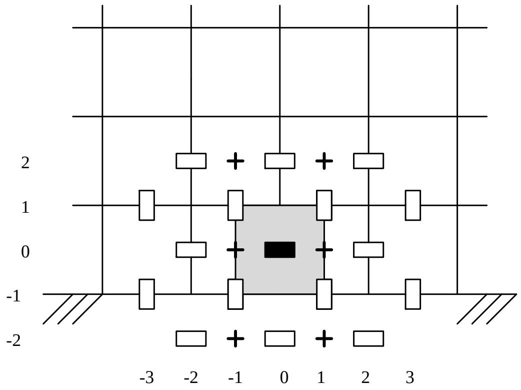


Figure 8.1: A "tangential" velocity cell at the boundary

quantities are expressed in internal unknowns and prescribed velocity components at the boundary using linear extrapolation. For example for the lower boundary (Figure 8.1) we get:

$$p_{i,-2} = 2p_{i,0} - p_{i,2} \tag{8.3}$$

$$V^1_{i,-2} = 2V^1_{i,-1} - V^1_{i,0}, \tag{8.4}$$

where $V^1_{i,-1}$ is the value of $\sqrt{g}U^t$ at the boundary point. The coefficient in the matrix corresponding to the virtual unknown multiplied by the expression (8.3) or (8.4) is transported to the right-hand side or the other matrix terms.

8.1.2 3D implementation

We give here two ways to describe the velocities in the physical domain:

(i) with Cartesian velocity components,

(ii) with normal and tangential components.

In the present version, only the first approach is implemented.

(i) If the Cartesian velocity components are prescribed we can compute the contravariant components in the following way:

$$U^\alpha = \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u} \tag{8.5}$$

(ii) Normal and tangential components are prescribed on the plane where ξ^n is constant. The scalar products $\boldsymbol{u} \cdot \boldsymbol{n}$, $\boldsymbol{u} \cdot \boldsymbol{\tau}_1$ and $\boldsymbol{u} \cdot \boldsymbol{\tau}_2$ with the tangential vectors $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are given by the user. See Figure 8.2.

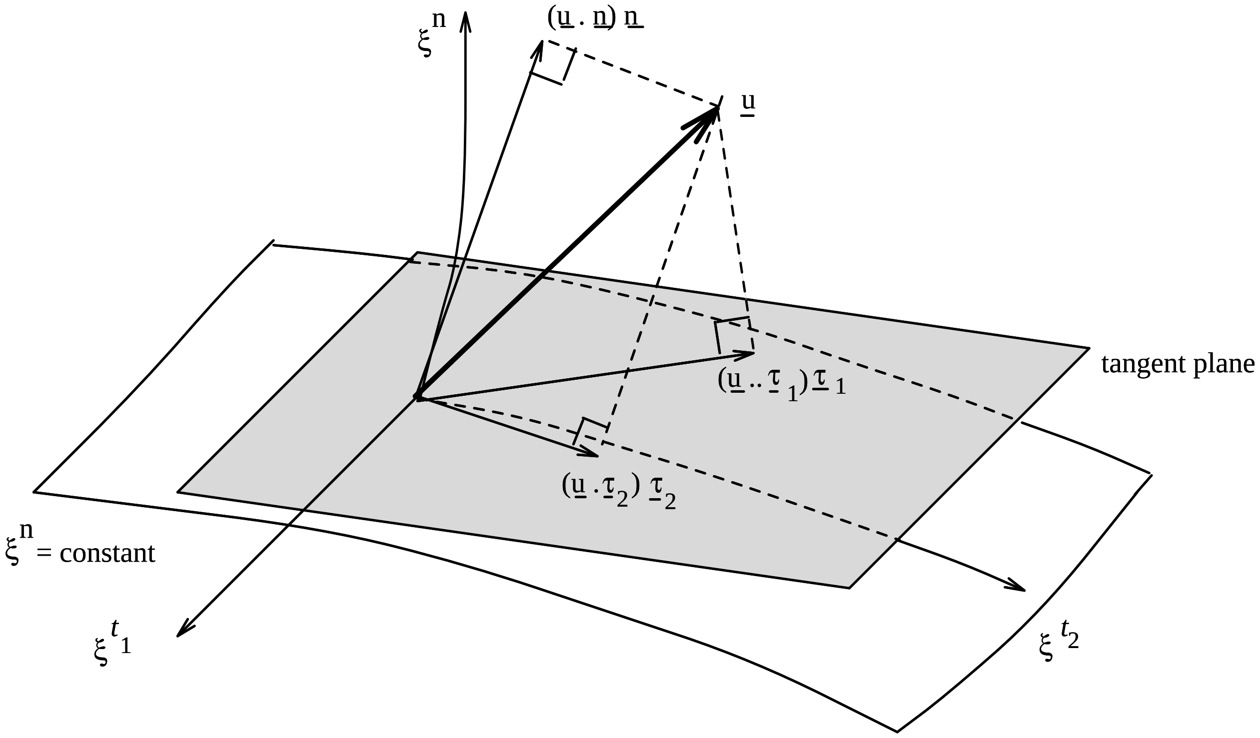


Figure 8.2: The scalar products $\underline{u} \cdot \underline{n}$, $\underline{u} \cdot \underline{\tau}_1$ and $\underline{u} \cdot \underline{\tau}_2$ and tangential vectors $\underline{\tau}_1$ and $\underline{\tau}_2$ are prescribed. Here is $\|\underline{n}\| = \|\underline{\tau}_1\| = \|\underline{\tau}_2\| = 1$.

From $\boldsymbol{u} \cdot \boldsymbol{n}$, $\boldsymbol{u} \cdot \boldsymbol{\tau}_1$, $\boldsymbol{u} \cdot \boldsymbol{\tau}_2$, $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ we can compute U^n, U^{t_1} and U^{t_2} by:

$$U^n = sign\ (\boldsymbol{a}^{(n)} \cdot \boldsymbol{n})\sqrt{g^{nn}}\boldsymbol{u} \cdot \boldsymbol{n} \tag{8.6}$$

$$\begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{\tau}_1 \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_2 \end{bmatrix} - U^n \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} \right\} \tag{8.7}$$

where

$$\alpha_{i1} = \frac{\begin{vmatrix} \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_1)} & g_{t_1t_2} \\ \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_2)} & g_{t_2t_2} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}}, \tag{8.8a}$$

$$\alpha_{i2} = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_1)} \\ g_{t_2t_1} & \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}}. \tag{8.8b}$$

Formula (8.7) makes no sense if the tangential vectors $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ are linear dependent, so they have to be linear independent.

The given U^n, U^{t_1} and U^{t_2} are implemented in almost the same way as in the 2D-case.

8.2 Stresses prescribed

8.2.1 2D implementation

In 2D stresses prescribed implies that normal and tangential stress components at the boundary are prescribed. Let S^{nn} and S^{nt} be the normal resp. tangential physical stress at the boundary, where the normal and tangential vector are defined as in 5.1.1.

From S^{nn} and S^{nt} we can compute σ^{nn} and σ^{nt} by

$$\begin{aligned} \sigma^{nn} &= g^{nn}S^{nn}, && (8.9) \\ \sigma^{nt} &= (\sqrt{g^{nn}g_{tt}}S^{nt} - g_{nt}\sigma^{nn})/g_{tt}, && (8.10) \end{aligned}$$

where $\sigma^{\alpha\beta}$ is defined by

$$\sigma^{\alpha\beta} = -g^{\alpha\beta}p + \tau^{\alpha\beta}. \qquad (8.11)$$

An important remark is that in this formulation pressure and deviatoric stress tensor can not be separated, hence the discretization of both must be the same at the boundary. For that reason the discretization of the pressure at the boundary will be different from the one in the inner region.

Since no velocities are prescribed, it is necessary to consider finite volume cells around each velocity unknown, including the "normal" velocity points at the boundary.

Let us first consider the "tangential" boundary cell as sketched in Figure 8.1. The discretization of the convective terms, the right-hand side and the time derivative are exactly the same as for the inner cells, with the exception that virtual (tangential) velocities are eliminated by linear extrapolation as in formula (8.4).

The stress tensor (deviatoric part and pressure together) is discretized by:

$$\sqrt{g}\sigma^{11}|_{(-1,0)}^{(1,0)} + \sqrt{g}\sigma^{12}|_{(0,-1)}^{(0,1)} + \left\{ {1 \atop \gamma\beta} \right\}\sigma^{\gamma\beta}\sqrt{g}|_{(0,0)} \qquad (8.12)$$

In this expression $\sigma^{12}|_{(0,-1)}$ is given by formula (8.4). All other terms are treated in the usual way (except of course for the pressure).

With respect to the normal velocity unknown at the boundary a half cell is defined as in Figure 8.3. The discretization of the convective terms plus the stress tensor at the boundary is given by formula (6.14) from Van Kan *et al.* (1991):

$$\frac{1}{2}\sqrt{g}T^{21}|_{(-1,1)}^{(1,1)} + \sqrt{g}T^{22}|_{(0,0)}^{(0,1)} + \frac{1}{2}\sqrt{g}\left\{ {2 \atop \gamma\beta} \right\}T^{\gamma\beta}|_{(0,0)}, \qquad (8.13)$$

where

$$T^{\alpha\beta} = \rho U^{\alpha}U^{\beta} - \sigma^{\alpha\beta} \qquad (8.14)$$

The discretization of the right-hand side gives

$$\frac{1}{2}\rho f^{\alpha}\sqrt{g}|_{(0,0)} \qquad (8.15)$$

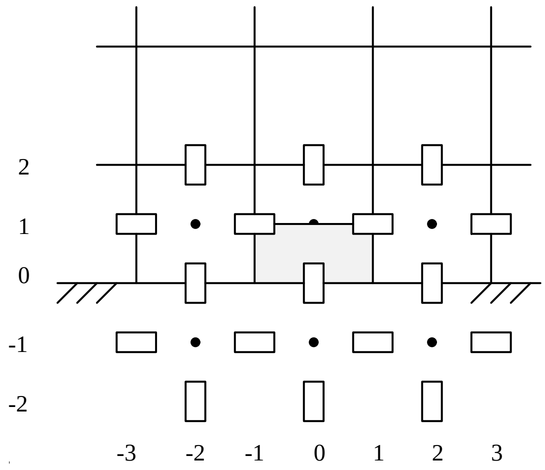


Figure 8.3: A "normal" velocity half cell at the boundary

and of the time-derivative:

$$\frac{1}{2}\frac{\partial}{\partial t}(\rho V^\alpha)|_{(0,0)} \tag{8.16}$$

The discretization of the convective terms is derived from (8.13) by substitution of

$$T^{\alpha\beta} = \rho U^\alpha U^\beta \tag{8.17}$$

and the approximation

$$V^1_{0,0} = \frac{1}{2}(V^1_{1,1} + V^1_{-1,1}) \tag{8.18}$$

The discretization of the stress tensor at the boundary is given by formulae (6.14), (6.15) of Van Kan *et al.* (1991):

$$RHS - \sqrt{g}\sigma^{22}|_{(0,1)} - \frac{1}{2}\sqrt{g}\left\{ \begin{matrix} 2 \\ 11 \end{matrix} \right\}\sigma^{11}|_{(0,0)}, \tag{8.19}$$

where RHS is defined by

$$\begin{aligned} RHS \;\; = \;\; & -\frac{1}{2}\sqrt{g}\sigma^{21}|_{(1,0)} + \frac{1}{2}\sqrt{g}\sigma^{21}|_{(-1,0)} + \sqrt{g}\sigma^{22}|_{(0,0)} \\ & -\frac{1}{2}\sqrt{g}\left\{ \begin{matrix} 2 \\ 22 \end{matrix} \right\}\sigma^{22}|_{(0,0)} - \sqrt{g}\left\{ \begin{matrix} 2 \\ 12 \end{matrix} \right\}\sigma^{12}|_{(0,0)} \end{aligned} \tag{8.20}$$

The evaluation of $\sigma^{11}|_{(0,0)}$ introduces extra difficulties.
Following Van Kan *et al.* (1991), page 76, we use $p_{0,1}$ instead of $p_{0,0}$.
Furthermore $\frac{\partial U^1}{\partial \xi^2}|_{(0,0)}$ is computed at the preceding time-level, and $\frac{\partial U^1}{\partial \xi^1}|_{(0,0)}$ replaced by $\frac{\partial U^1}{\partial \xi^1}|_{(0,1)}$. Virtual velocities are not used. To compute $\frac{\partial U^1}{\partial \xi^2}|_{(0,0)}$ at the preceding time level, U^1 at the boundary is computed by linear extrapolation from inside, using two points.

8.2.2 3D implementation

The normal and tangential stress components at the boundary are prescribed. Let S^{nn} be the normal stress component at the boundary and $S^{n\tau}$ the tangential stress component in the

$\boldsymbol{\tau}$ direction (see Figure 8.4). So:

$$\boldsymbol{S}_{\xi^n=const} = S^{nn}\boldsymbol{n} + S^{n\tau}\boldsymbol{\tau} \tag{8.21}$$

where $\boldsymbol{n}$ is an unit normal vector and $\boldsymbol{\tau}$ is an unit tangential vector. From $S^{nn}\boldsymbol{n}$ and $S^{n\tau}\boldsymbol{\tau}$ we compute σ^{nn}, σ^{nt_1} and σ^{nt_2} the stresses in the computational domain. Just asin the 2D-case

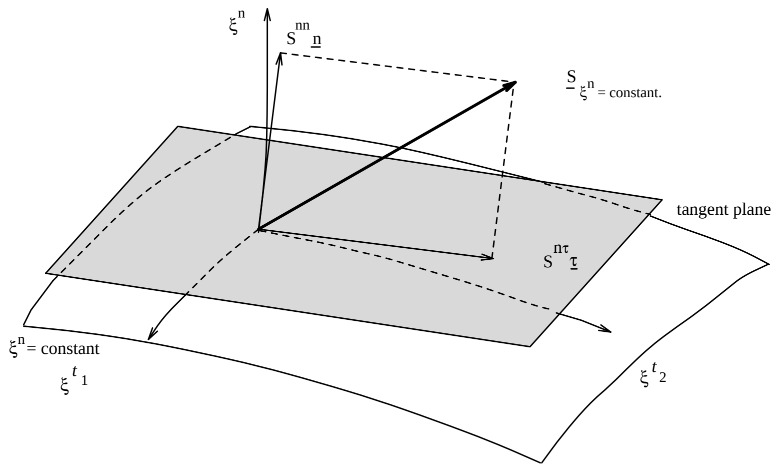


Figure 8.4: The normal and tangential stress in the physical domain at the boundary $\xi^n = constant$.

is:

$$\sigma^{nn} = g^{nn}S^{nn}\ , \tag{8.22}$$

σ^{nt_1} and σ^{nt_2} are computed by:

$$\begin{bmatrix} \sigma^{nt_1} \\ \sigma^{nt_2} \end{bmatrix} = sign\ (\boldsymbol{a}^{(n)} \cdot \boldsymbol{n})\sqrt{g^{nn}}S^{n\tau} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} - \sigma^{nn} \begin{bmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{bmatrix}^{-1} \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} \tag{8.23}$$

where

$$\alpha_1 = \frac{\begin{vmatrix} \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} & g_{t_1t_2} \\ \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} & g_{t_2t_2} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}}\ .; \tag{8.24a}$$

$$\alpha_2 = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} \\ g_{t_2t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} \tag{8.24b}$$

The stress $\sigma^{\alpha\beta}$ is defined by $\sigma^{\alpha\beta} = -g^{\alpha\beta}p + \tau^{\alpha\beta}$. At the boundary it is impossible to separate the pressure from te deviatoric stress tensor $\tau^{\alpha\beta}$. So the discretization of the pressure at the boundary will be different from the one in the inner region.

We have to consider three different cells closest to the boundary two "tangential" and one "normal" velocity cell.

Let us first consider the "tangential" cells. The two "tangential" cells closest to the boundary are considered differently from the ones in the inner region, because the stencil contain virtual unknowns (see Figure 8.5). The discretization of the convective terms, the right-hand side

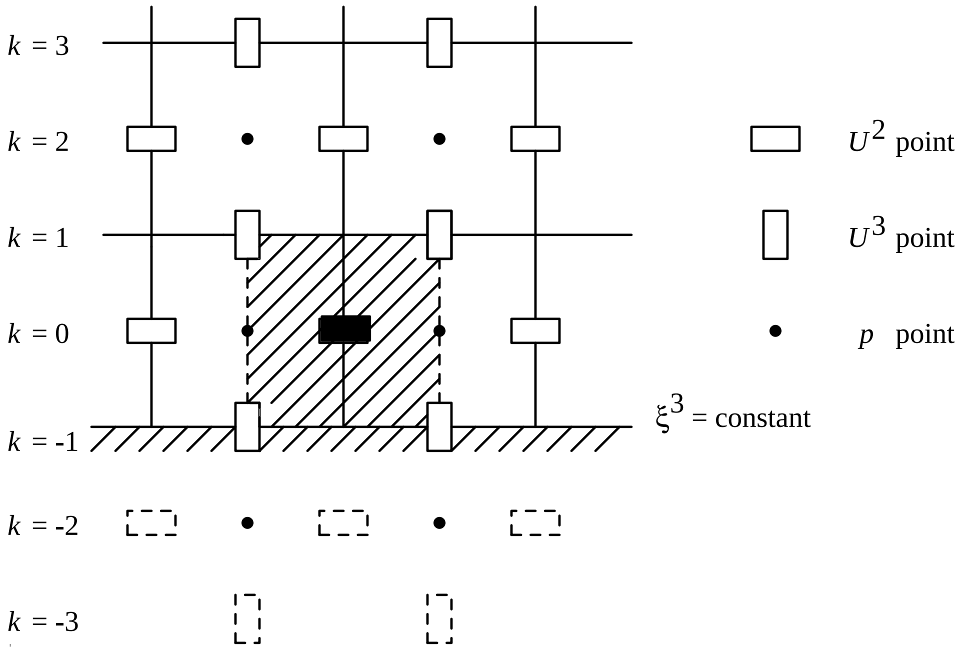


Figure 8.5: A cross-section ($\xi^1 = constant$) over an U^2-cell closest to the bottom of the region.

and the time derivative are the same as for the inner cells, with the exception that virtual velocities are eliminated by linear extrapolation. For example for the bottom boundary (see Figure 8.5) we get:

$$V^1_{i,j,-2} = 2V^1_{i,j,0} - V^1_{i,j,2} \tag{8.25}$$

$$V^2_{i,j,-2} = 2V^2_{i,j,0} - V^2_{i,j,2} \tag{8.26}$$

$$V^3_{i,j,-3} = 2V^3_{i,j,-1} - V^3_{i,j,1} \tag{8.27}$$

The stress tensor $\sigma^{\alpha\beta}$ is discretized in the following way for the V^α-cell:

$$\begin{aligned} & -\sqrt{g}\sigma^{\alpha 1}|^{(1,0,0)}_{(-1,0,0)} - \sqrt{g}\sigma^{\alpha 2}|^{(0,1,0)}_{(0,-1,0)} - \sqrt{g}\sigma^{\alpha 3}|^{(0,0,1)}_{(0,0,-1)} \\ & -\sqrt{g}\left\{ {\alpha \atop \gamma\beta} \right\}\sigma^{\gamma\beta}|_{(0,0,0)} \quad \text{for} \quad \alpha = 1, 2. \end{aligned} \tag{8.28}$$

Term $\sigma^{\alpha 3}|_{(0,0,-1)}$ is given by formula (8.23), if we are concerned with the bottom boundary. All other terms are treated in the usual way.

Since no normal velocity components are prescribed at the boundary we have to consider a finite volume around a "normal" velocity point at the boundary (see Figure 8.6). We will now consider the discretization for a "normal" velocity cell at the bottom boundary.
The discretization of the time-derivative gives:

$$\frac{1}{2}\frac{\partial}{\partial t}(\rho V^3)|_{(0,0,0)} \tag{8.29}$$

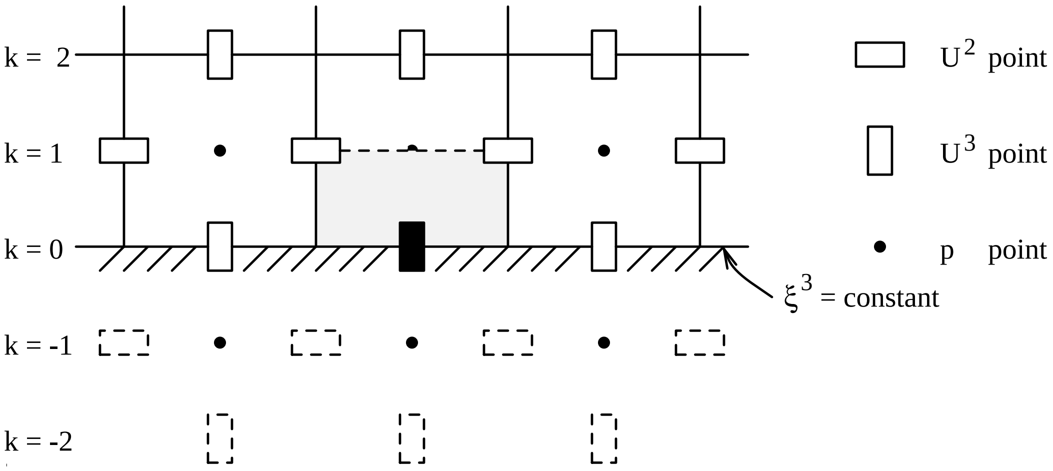


Figure 8.6: A cross-section ($\xi^1 = constant$) over an U^3-cell at the bottom.

and of the right-hand side:

$$\frac{1}{2}(\rho f^3 \sqrt{g})|_{(0,0,0)} \ . \tag{8.30}$$

The discretization of the convective terms is given by:

$$\begin{aligned} &\frac{1}{2}\frac{\rho}{\sqrt{g}} V^3 V^1 |^{(1,0,0)}_{(-1,0,0)} + \frac{1}{2}\frac{\rho}{\sqrt{g}} V^3 V^2 |^{(0,1,0)}_{(0,-1,0)} + \frac{\rho}{\sqrt{g}} V^3 V^3 |^{(0,0,1)}_{(0,0,0)} \\ &+\frac{1}{2}\frac{\rho}{\sqrt{g}} \{ {}^{\ 3}_{\gamma\beta} \} V^\gamma V^\beta |_{(0,0,0)} \end{aligned} \tag{8.31}$$

and the approximation:

$$V^1_{i,j,0} = V^1_{i,j,1} \tag{8.32}$$
$$V^2_{i,j,0} = V^2_{i,j,1} \ . \tag{8.33}$$

If V^1 or V^2 are not present at $(i, j, 1)$ then they are approximated by:

$$V^1_{i,j,1} = \frac{1}{2}(V^1_{i-1,j,1} + V^1_{i+1,j,1}) \tag{8.34}$$
$$V^2_{i,j,1} = \frac{1}{2}(V^2_{i,j-1,1} + V^2_{i,j+1,1}) \ . \tag{8.35}$$

The discretization of the stress tensor at the boundary is given by the following formula:

$$\begin{aligned} &-\frac{1}{2}\sqrt{g}\sigma^{31}|^{(1,0,0)}_{(-1,0,0)} - \frac{1}{2}\sqrt{g}\sigma^{32}|^{(0,1,0)}_{(0,-1,0)} - \sqrt{g}\sigma^{33}|^{(0,0,1)}_{(0,0,0)} \\ &-\frac{1}{2}\sqrt{g}\{ {}^{\ 3}_{\gamma\beta} \}\sigma^{\gamma\beta}|_{(0,0,0)} \end{aligned} \tag{8.36}$$

or

$$\begin{aligned} RHS \ &- \ \sqrt{g}\sigma^{33}|_{(0,0,1)} - \frac{1}{2}\sqrt{g}\{ {}^{3}_{11} \}\sigma^{11}|_{(0,0,0)} \\ &- \ \sqrt{g}\{ {}^{3}_{12} \}\sigma^{12}|_{(0,0,0)} - \frac{1}{2}\sqrt{g}\{ {}^{3}_{22} \}\sigma^{22}|_{(0,0,0)} \end{aligned} \tag{8.37}$$

where RHS is given by:

$$\begin{aligned} RHS &= -\frac{1}{2}\sqrt{g}\sigma^{31}|_{(-1,0,0)}^{(1,0,0)} - \frac{1}{2}\sqrt{g}\sigma^{32}|_{(0,-1,0)}^{(0,1,0)} + \sqrt{g}\sigma^{33}|_{(0,0,0)} \\ & -\sqrt{g}\{ {3 \atop 13} \}\sigma^{13}|_{(0,0,0)} - \sqrt{g}\{ {3 \atop 23} \}\sigma^{23}|_{(0,0,0)} - \frac{1}{2}\sqrt{g}\{ {3 \atop 33} \}\sigma^{33}|_{(0,0,0)} \,. \end{aligned} \tag{8.38}$$

The evaluation of $\sigma^{11}|_{(0,0,0)}$, $\sigma^{12}_{(0,0,0)}$ and $\sigma^{22}_{(0,0,0)}$ introduces some difficulties. First we need the pressure in point $(0,0,0)$, because we have to split up σ^{11}, σ^{12} and σ^{22}. Instead of $p_{0,0,0}$ we use $p_{0,0,1}$ just as in the 2D-case.
Secondly we need $\frac{\partial U^1}{\partial \xi^\alpha}$ and $\frac{\partial U^2}{\partial \xi^\alpha}$ at (0,0,0) for $\alpha = 1,2,3$. The derivatives $\frac{\partial U^1}{\partial \xi^1}|_{(0,0,0)}$, $\frac{\partial U^1}{\partial \xi^2}|_{(0,0,0)}$, $\frac{\partial U^2}{\partial \xi^1}|_{(0,0,0)}$ and $\frac{\partial U^2}{\partial \xi^2}|_{(0,0,0)}$ are replaced respectively by $\frac{\partial U^1}{\partial \xi^1}|_{(0,0,1)}$, $\frac{\partial U^1}{\partial \xi^2}|_{(0,0,1)}$, $\frac{\partial U^2}{\partial \xi^1}|_{(0,0,1)}$ and $\frac{\partial U^2}{\partial \xi^2}|_{(0,0,1)}$. In Van Kan *et al.* (1991), page 34, there are three strategies mentioned to compute $\frac{\partial U^1}{\partial \xi^3}|_{(0,0,0)}$ and $\frac{\partial U^2}{\partial \xi^3}|_{(0,0,0)}$. It seems reasonable if we use the same strategy as used in the 2D-case. That is, the derivatives $\frac{\partial U^1}{\partial \xi^3}|_{(0,0,0)}$ and $\frac{\partial U^2}{\partial \xi^3}|_{(0,0,0)}$ are computed at the preceding time level, U^1 and U^2 at the boundary are computed by linear extrapolation, using two points, so:

$$\frac{\partial U^\alpha}{\partial \xi^3}|_{(0,0,0)} = U^\alpha|_{(0,0,3)} - U^\alpha|_{(0,0,1)} \tag{8.39}$$

(at the preceding time level) for $\alpha = 1, 2$.

8.3 Semi-natural outflow condition

8.3.1 2D implementation

With semi-natural outflow condition we mean tangential velocity and normal stress prescribed at the boundary, i.e.

$$\boldsymbol{u} \cdot \boldsymbol{t} \text{ and } S^{nn} \text{ given.} \tag{8.40}$$

Although $\boldsymbol{u} \cdot \boldsymbol{t}$ prescribed does in general not imply U^t prescribed we assume that instead of (8.40) the following boundary condition is given:

$$U^t \text{ and } S^{nn} \text{ given} \tag{8.41}$$

where S^{nn} is related to σ^{nn} by (8.9).

Boundary condition (8.41) influences both the tangential velocity cell as sketched in Figure 8.1, as the normal velocity half-cell sketched in Figure 8.3.

With respect to the tangential cell, the molecule is built in the same way as for the inner cells. The only difference is that virtual velocity components and virtual pressures are eliminated by linear extrapolation, i.e. by applying formulae (8.3) and (8.4).

The normal velocity half cell gives rise to the following discretization of the stress tensor (see [36], formula (6.19)):

$$\begin{aligned} & -\frac{1}{2}\sqrt{g}\sigma^{21}|_{(-1,0)}^{(1,0)} - \sqrt{g}\sigma^{22}|_{(0,1)} - \frac{1}{2}\sqrt{g}(\{ {2 \atop 11} \}\sigma^{11} + 2\{ {2 \atop 12} \}\sigma^{12})|_{(0,0)} \\ & +\sqrt{g}\sigma^{22}|_{(0,0)} - \frac{1}{2}\sqrt{g}\{ {2 \atop 22} \}\sigma^{22}|_{(0,0)}, \end{aligned} \tag{8.42}$$

where $\sigma^{22}|_{(0,0)}$ is given by (8.9). Virtual velocities are eliminated by linear extrapolation using formula (8.3). It must be remarked that the first term with respect to the pressure is evaluated in the points (1,1) and (-1,1) instead of (1,0) resp. (-1,0).
The convective terms are evaluated by expanding

$$\frac{1}{2}\rho\sqrt{g}U^2U^\beta|_{(-1,0)}^{(1,0)} + \rho\sqrt{g}U^2U^\beta|_{(0,0)}^{(0,1)} + \frac{1}{2}\rho\sqrt{g}\{{}_{\gamma\beta}^{2}\}U^\gamma U^\beta|_{(0,0)} \tag{8.43}$$

using the standard inter- and extrapolations.

8.3.2 3D implementation

The tangential velocity and normal stress are prescribed at the boundary, i.e.:

$$\boldsymbol{u}\cdot\boldsymbol{\tau}_1,\ \boldsymbol{u}\cdot\boldsymbol{\tau}_2,\ \boldsymbol{\tau}_1,\ \boldsymbol{\tau}_2 \quad \text{and} \quad S^{nn} \quad \text{are given .} \tag{8.44}$$

From equation (8.7) it is clear that we can calculate U^{t_1} and U^{t_2} as g_{nt_1} and g_{nt_2} are zero, otherwise we have to make an assumption about U^{t_1} and U^{t_2}. In the remainder of this section we assume that

$$U^{t_1},\ U^{t_2} \quad \text{and} \quad S^{nn} \quad \text{are given ,} \tag{8.45}$$

at the boundary.
From S^{nn} and equation (8.22) follows the stress σ^{nn} at the boundary in the computational domain.

Boundary condition (8.45) influences the two "tangential" velocity cells (see Figure 8.5) and the "normal" velocity half-cell (see Figure 8.6).

The U^1 and U^2 "tangential" cells (bottom boundary) are built in a similar way as the inner cells. The only difference is that virtual velocity components and pressures are eliminated by linear extrapolation. For example for the bottom boundary we get:

$$\begin{aligned} V^\alpha_{i,j,-2} &= 2V^\alpha_{i,j,-1} - V^\alpha_{i,j,0} \quad \text{for} \quad \alpha = 1,2 && (8.46)\\ V^3_{i,j,-3} &= 2V^3_{i,j,-1} - V^3_{i,j,1} && (8.47)\\ p_{i,j,-2} &= 2p_{i,j,0} - p_{i,j,2}\ . && (8.48) \end{aligned}$$

The normal velocity half-cell at the bottom boundary. The discretization of the convective term gives

$$\begin{aligned} &\frac{1}{2}\frac{\rho}{\sqrt{g}}V^3V^1|_{(-1,0,0)}^{(1,0,0)} + \frac{1}{2}\frac{\rho}{\sqrt{g}}V^3V^2|_{(0,-1,0)}^{(0,1,0)} + \frac{\rho}{\sqrt{g}}V^3V^3|_{(0,0,0)}^{(0,0,1)} \\ &+\frac{1}{2}\frac{\rho}{\sqrt{g}}\{{}_{\gamma\beta}^{3}\}V^\gamma V^\beta|_{(0,0,0)}\ . \end{aligned} \tag{8.49}$$

Terms with the factor V^3V^3 are the only terms where we need a linearization procedure, since V^1 and V^2 are given at the boundary.
We use the following discretization of the stress tensor:

$$\begin{aligned} &- \frac{1}{2}\sqrt{g}\sigma^{31}|_{(-1,0,0)}^{(1,0,0)} - \frac{1}{2}\sqrt{g}\sigma^{32}|_{(0,-1,0)}^{(0,1,0)} - \sqrt{g}\sigma^{33}|_{(0,0,0)}^{(0,0,1)} \\ &- \frac{1}{2}\sqrt{g}\{{}_{\gamma\beta}^{3}\}\sigma^{\gamma\beta}|_{(0,0,0)}\ , \end{aligned} \tag{8.50}$$

where $\sigma^{33}|_{(0,0,0)}$ is given by $(g^{33}S^{33})|_{(0,0,0)}$. So the right-hand side gets the contribution:

$$(-1+\frac{1}{2}\{\begin{smallmatrix}3\\33\end{smallmatrix}\})\sqrt{g}\sigma^{33}|_{(0,0,0)}\,. \tag{8.51}$$

The virtual velocities introduced by formula (8.51) are eliminated by linear extrapolation. Just as in 2D is the pressure evaluated in the points (1,0,1), (-1,0,1), (0,1,1) and (0,-1,1) instead of (1,0,0), (-1,0,0), (0,1,0) and (0,-1,0).

8.4 Slip boundary condition

8.4.1 2D implementation

The slip boundary condition is equivalent to tangential stress as well as normal velocity component given. The treatment of this type of boundary conditions is the subject of [25]. All formulae in this section are copied from that report.
The normal velocity given implies U^n given by the relation (8.1). The tangential stress given, however, does not automatically imply that a component of the contravariant stress tensor is prescribed. In fact it only implies a linear combination between stress components at the boundary through the relation

$$\sqrt{g^{nn}g_{tt}}S^{nt}=g_{nt}\sigma^{nn}+g_{tt}\sigma^{nt} \tag{8.52}$$

Since the normal velocity component is given no normal half cells are introduced.

The discretization of the convective terms implies the evaluation of virtual velocities by linear extrapolation. The discretization of the stress tensor is given by

$$-\sqrt{g}\sigma^{11}|_{(-1,0)}^{(1,0)}-\sqrt{g}\sigma^{12}|_{(0,-1)}^{(0,1)}-\{\begin{smallmatrix}1\\\gamma\beta\end{smallmatrix}\}\sigma^{\gamma\beta}\sqrt{g}|_{(0,0)} \tag{8.53}$$

The linear extrapolation formula for the velocities is given by

$$V^1_{i,-2}=2V^1_{i,0}-V^1_{i,2} \tag{8.54}$$

The term $(\sqrt{g}\sigma^{12})_{(0,-1)}$ is equal to:

$$\sqrt{g}\sigma^{12}|_{(0,-1)}=S^{nt}|_{(0,-1)}-\sqrt{g}\frac{g_{21}}{g_{11}}\sigma^{22}|_{(0,-1)}, \tag{8.55}$$

where the first term is given and the second term involves virtual velocities and pressures that can be eliminated by (8.54) resp. (8.3).

8.4.2 3D implementation

In this condition the normal velocity $\boldsymbol{u}\cdot\boldsymbol{n}$ and tangential stress $S^{n\tau}$ with $\boldsymbol{\tau}$ are prescribed. So we don't need "normal" velocity half-cells. We have only to consider the two "tangential" cells. For the remainder of this section is the slip boundary condition given on the bottom boundary.

The discretization of the convective terms is given by:

$$
\begin{aligned}
&\frac{\rho}{\sqrt{g}} V^\alpha V^1 |_{(-1,0,0)}^{(1,0,0)} + \frac{\rho}{\sqrt{g}} V^\alpha V^2 |_{(0,-1,0)}^{(0,1,0)} + \frac{\rho}{\sqrt{g}} V^\alpha V^3 |_{(0,0,-1)}^{(0,0,1)} \\
&+ \frac{\rho}{\sqrt{g}} \left\{ {\alpha \atop \gamma\beta} \right\} V^\gamma V^\beta |_{(0,0,0)} \quad \text{for} \quad \alpha = 1, 2 \,. \qquad (8.56)
\end{aligned}
$$

Since $V^3 = \sqrt{g} U^3$ is prescribed at the bottom boundary we do not have to use a linear approximation for $V^\alpha V^3|_{(0,0,-1)}$ (see [25], formula (3.9)):

$$
V^\alpha V^3|_{(0,0,-1)} = \frac{1}{2}(3V^\alpha|_{(0,0,0)} - V^\alpha|_{(0,0,2)}) V^3|_{(0,0,-1)} \quad \text{for} \quad \alpha = 1, 2 \,. \qquad (8.57)
$$

The discretization of the stress tensor is given by:

$$
\begin{aligned}
&-\sqrt{g}\sigma^{\alpha 1}|_{(-1,0,0)}^{(1,0,0)} - \sqrt{g}\sigma^{\alpha 2}|_{(0,-1,0)}^{(0,1,0)} - \sqrt{g}\sigma^{\alpha 3}|_{(0,0,-1)}^{(0,0,1)} \\
&-\sqrt{g}\left\{ {\alpha \atop \gamma\beta} \right\} \sigma^{\gamma\beta}|_{(0,0,0)} \quad \text{for} \quad \alpha = 1, 2 \,. \qquad (8.58)
\end{aligned}
$$

The virtual velocities introduced by formula (8.58) are eliminated by linear extrapolation, i.e. using formula (8.25), (8.26) and (8.27).

The term $\sigma^{\alpha 3}|_{(0,0,-1)}$ in (8.58) is for $\alpha = 1$ (U^1-cell) equal to:

$$
\sigma^{13}|_{(0,0,-1)} = \left(sign\ (\boldsymbol{a}^{(3)} \cdot \boldsymbol{n}) \sqrt{g^{33}} S^{n\tau} \alpha_1 - \sigma^{33} \boldsymbol{e}_1^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1} \begin{bmatrix} g_{31} \\ g_{32} \end{bmatrix} \right) |_{(0,0,-1)} \qquad (8.59)
$$

where $\boldsymbol{e}_1^T = [1 \;\; 0]$, so:

$$
\sigma^{13}|_{(0,0,-1)} = \left(sign\ (\boldsymbol{a}^{(3)} \cdot \boldsymbol{n}) \sqrt{g^{33}} S^{n\tau} \alpha_1 - \sigma^{33} \frac{g_{22}g_{31} - g_{12}g_{32}}{g_{11}g_{22} - g_{12}g_{21}} \right) |_{(0,0,-1)} \,. \qquad (8.60)
$$

Term $\sigma^{\alpha 3}|_{(0,0,-1)}$ in (8.58) is for $\alpha = 2$ equal to:

$$
\sigma^{23}|_{(0,0,-1)} = \left(sign\ (\boldsymbol{a}^{(3)} \cdot \boldsymbol{n}) \sqrt{g^{33}} S^{n\tau} \alpha_2 - \sigma^{33} \boldsymbol{e}_2^T \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}^{-1} \begin{bmatrix} g_{31} \\ g_{32} \end{bmatrix} \right) |_{(0,0,-1)} \qquad (8.61)
$$

where $\boldsymbol{e}_2^T = [0 \;\; 1]$, so:

$$
\sigma^{23}|_{(0,0,-1)} = \left(sign\ (\boldsymbol{a}^{(3)} \cdot \boldsymbol{n}) \sqrt{g^{33}} S^{n\tau} \alpha_2 - \sigma^{33} \frac{g_{11}g_{32} - g_{21}g_{31}}{g_{11}g_{22} - g_{21}g_{12}} \right) |_{(0,0,-1)} \,. \qquad (8.62)
$$

The factors α_1 and α_2 are calculated with formula (8.24a)-(8.24b).
The $\sigma^{33}|_{(0,0,-1)}$ in (8.60) and (8.62) involves virtual velocities and pressures that can be eliminated by (8.25)-(8.27) and (8.48).

Before treating the boundary conditions for the scalar equations we shall first consider the special cases where we have a transition of one type of a boundary condition to another as well the case of a corner of the region.

8.5 Transition of types of boundary conditions

8.5.1 2D implementation

Just as in the preceding cases we restrict ourselves to the lower boundary in the computational domain. Let at the vertex point S we have two types of boundary conditions (see Figure 8.7).

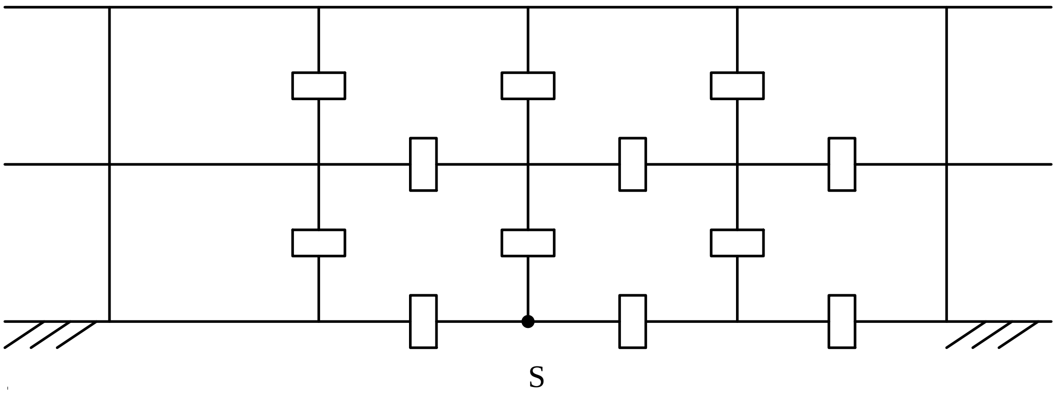


Figure 8.7: transition point S (vertex of cell). At the left of point S the type of boundary condition differs from the one on the right.

We shall only consider the boundary condition at the left of point S but in relation to the boundary condition at the right. In all cases the most restrictive boundary conditions, i.e. the one that influences the velocity most directly will be applied. Let us first consider Dirichlet boundary conditions at the left of S. Since Dirichlet boundary conditions are the most restrictive, it is assumed that also in point S the velocity is prescribed. All points left of S are treated in the usual way. The only special treatment is required for the tangential cell just above point S (Figure 8.8).

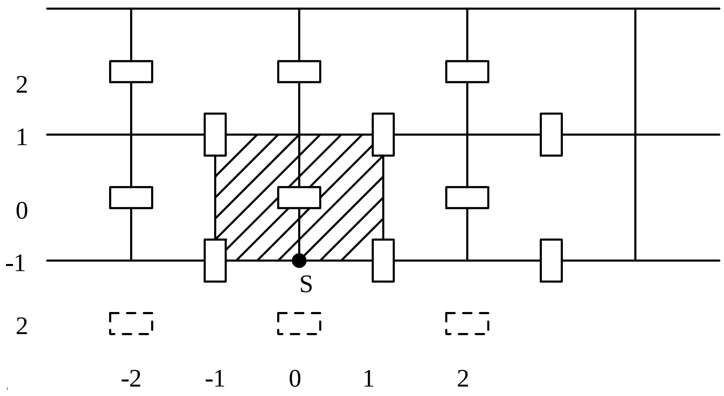


Figure 8.8: Tangential cell just above transition point S.

The molecule corresponding to $V^1_{(0,0)}$ contains 3 virtual points, 2 of which can be eliminated by linear extrapolation using the boundary conditions.
The only special point is $V^1_{(2,-2)}$. If at the right side of S we have a boundary condition of type 3 this point may be treated in the usual way. However, if boundary conditions of type

2 or 4 are prescribed $V^1_{(2,-2)}$ must be eliminated by the linear extrapolation:

$$V^1_{(2,-2)} = 2V^1_{(2,0)} - V^1_{(2,2)} \tag{8.63}$$

Now suppose that we have a boundary condition of type 2 at the left side of S and a boundary condition of type 1, 3 or 4 at the right side of S. It is sufficient to consider the tangential cell sketched in Figure 8.8 and the normal half cell left of point S sketched in Figure 8.9.

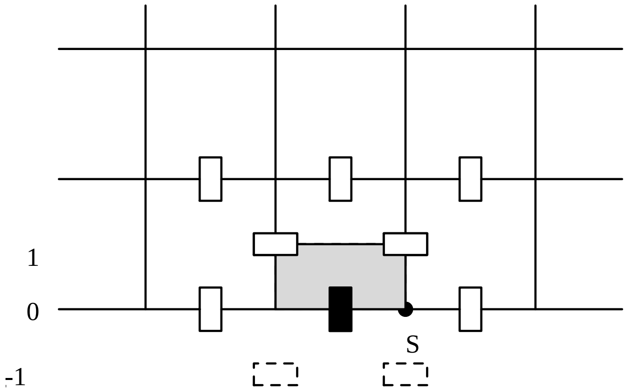


Figure 8.9: Normal half cell left of the transition point S

All other cells are treated in the usual way. Let us first consider the tangential cell of Figure 8.8. If at the right side of point S we have a boundary condition of type 1 or type 3, the tangential velocity in point S is given and the cell is treated as if corresponding to the right side. In the case of boundary conditions of type 4 the cell may be treated in the usual way.
With respect to the normal half cell we can proceed as usual. Since no virtual velocities appear no special treatment is necessary.

In the case that we have a boundary condition of type 3 at the left side of S we also distinguish between the tangential cell of Figure 8.8 and the normal half cell of Figure 8.9. With respect to the tangential cell the procedure described in 5.3.1 can be applied without any restriction. With respect to the normal half cell we may also proceed in the standard way, i.e. apply formula (8.42). This procedure leads to virtual unknowns which may be eliminated in the usual way. There is no need to use extra information if velocities are given at the right of point S.

Finally with respect to boundary conditions of type 4 it is sufficient to consider the tangential cell of Figure 8.8. If at the right side of point S boundary conditions of type 1 or type 3 are given (i.e. u_t prescribed) these boundary conditions prevail. In the case of boundary condition of type 2 at the right side of S, no special action is necessary.

8.5.2 3D implementation

Just as in the 2D case we restrict ourselves to only one boundary, the bottom boundary in the computational domain. First we assume that only one boundary condition type is prescribed on a "bottom boundary"-face of a p-cell, see Figure 8.10.
It is clear that the only thing that has to be treated very carefully is the extrapolations of

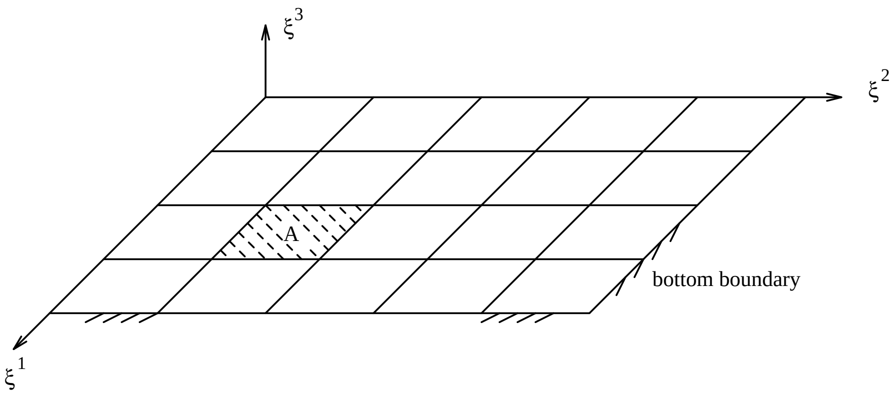


Figure 8.10: There is only one boundary condition type prescribed on A, the "bottom boundary"-face of a p-cell.

the virtual points, see Figure 8.11.
For the extrapolation of for instance $V^2_{(1,1,-2)}$ in a U^1-cell (see Figure 8.11(a)) we have to consider the boundary conditions in the two "bottom boundary"-faces A and B. If boundary condition type 1 (Dirichlet) or type 3 (semi-natural outflow) is prescribed in one of the two "bottom boundary"-faces then the following extrapolation is used:

$$V^2_{(1,1,-2)} = 2V^2_{(1,1,-1)} - V^2_{(1,1,0)} \,, \tag{8.64}$$

otherwise

$$V^2_{(1,1,-2)} = 2V^2_{(1,1,0)} - V^2_{(1,1,2)} \,. \tag{8.65}$$

It is clear that a similar procedure can be used for all virtual velocities in the "tangential" cells and "normal" half cell.

8.6 Treatment of boundary conditions at the corners of the region

With respect to the corners of the region, it is necessary to consider the boundary conditions carefully, because unknowns may be not present anymore. Let us investigate the four boundary conditions in this special case.

8.6.1 2D

For simplicity we restrict ourselves to the case of a boundary condition at the right side of the lower boundary in the computational region.

In the case of Dirichlet boundary conditions (type 1), no points at the right of the left boundary appear, hence no special precautions are necessary.

In the case of boundary conditions of type 2 we have to distinguish between "tangential" cells and normal half cells. Only tangential cells of tangential velocities not lying on another boundary are considered. As a consequence the last tangential cell is at a distance 1 from the boundary and no special treatment is necessary.
With respect to the normal half cell sketched in Figure 8.12 we have to be more careful.
The discretization of the convective terms using formulae (8.13), (8.17) and (8.18) introduces

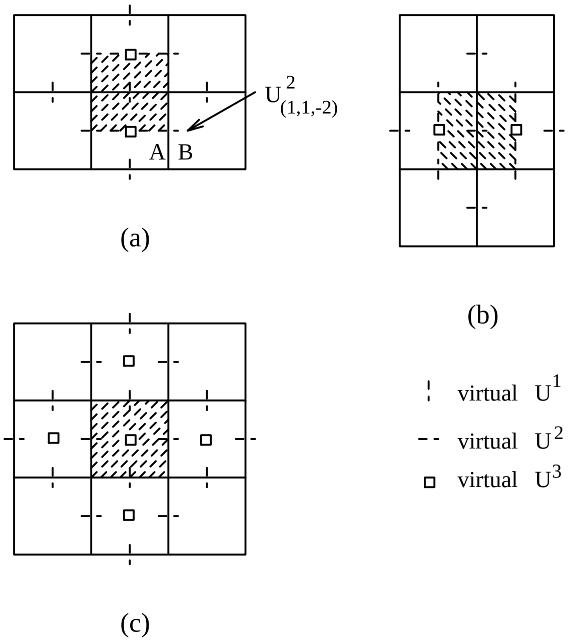


Figure 8.11: A cross-section of the bottom boundary and an (a) U^1-cell, (b) U^2-cell and (c) U^3-cell, with the positions of the virtual unknowns.

virtual velocities in the points (2,0) and (2,2). These virtual velocities are eliminated in the standard way by linear extrapolation using the value of V^2 at the right boundary if available and otherwise using the values $V^2_{(0,i)}$ and $V^2_{(-2,i)}$. Hence even if V^2 is given at the right boundary, we still use the interpolated values. This approach simplifies the treatment of the boundary conditions.

The stress tensor in this cell as treated in formulae (8.19), (8.20) does not introduce virtual unknowns at the right of the right boundary. Hence this part does not require a special treatment.

With respect to boundary conditions of type 3 and type 4 the standard procedure may be

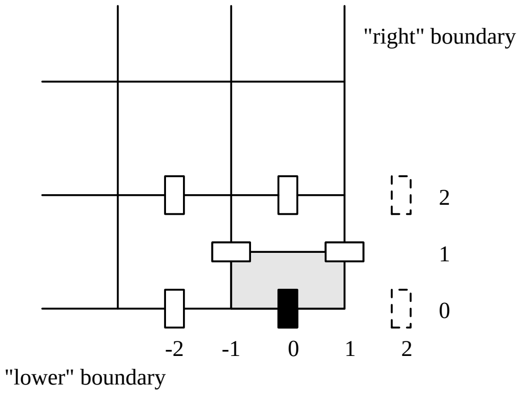


Figure 8.12: "normal" half-cell at the intersection of "lower" and right boundary.

followed, provided virtual velocities are eliminated in the usual way. This is the case both for the tangential cells and the normal half cells.

8.6.2 3D

In the 3D-case we have to consider two kinds of corners, edges and vertices.

Edges We restrict ourselves to the case of boundary conditions at the edge left under ($\xi^2 = 0$ and $\xi^3 = 0$) in the computational domain. There are $4^2 = 16$ possible combinations to prescribe the boundary conditions, but only $\begin{pmatrix} 4+2-1 \\ 2 \end{pmatrix} = 10$ are really different, see Table 8.1.

combination	boundary condition type	
	left boundary ($\xi^2 = 0$)	bottom boundary ($\xi^3 = 0$)
(i)	1	1
(ii)	1	2
(iii)	1	3
(iv)	1	4
(v)	2	2
(vi)	2	3
(vii)	2	4
(viii)	3	3
(ix)	3	4
(x)	4	4

Table 8.1: Combinations of the boundary conditions for the edge left under ($\xi^2 = 0$ and $\xi^3 = 0$.)

Combination (i)
Since the normal velocities are given at the boundaries we have only to consider the "tangential" U^1-cell, see Figure 8.13.

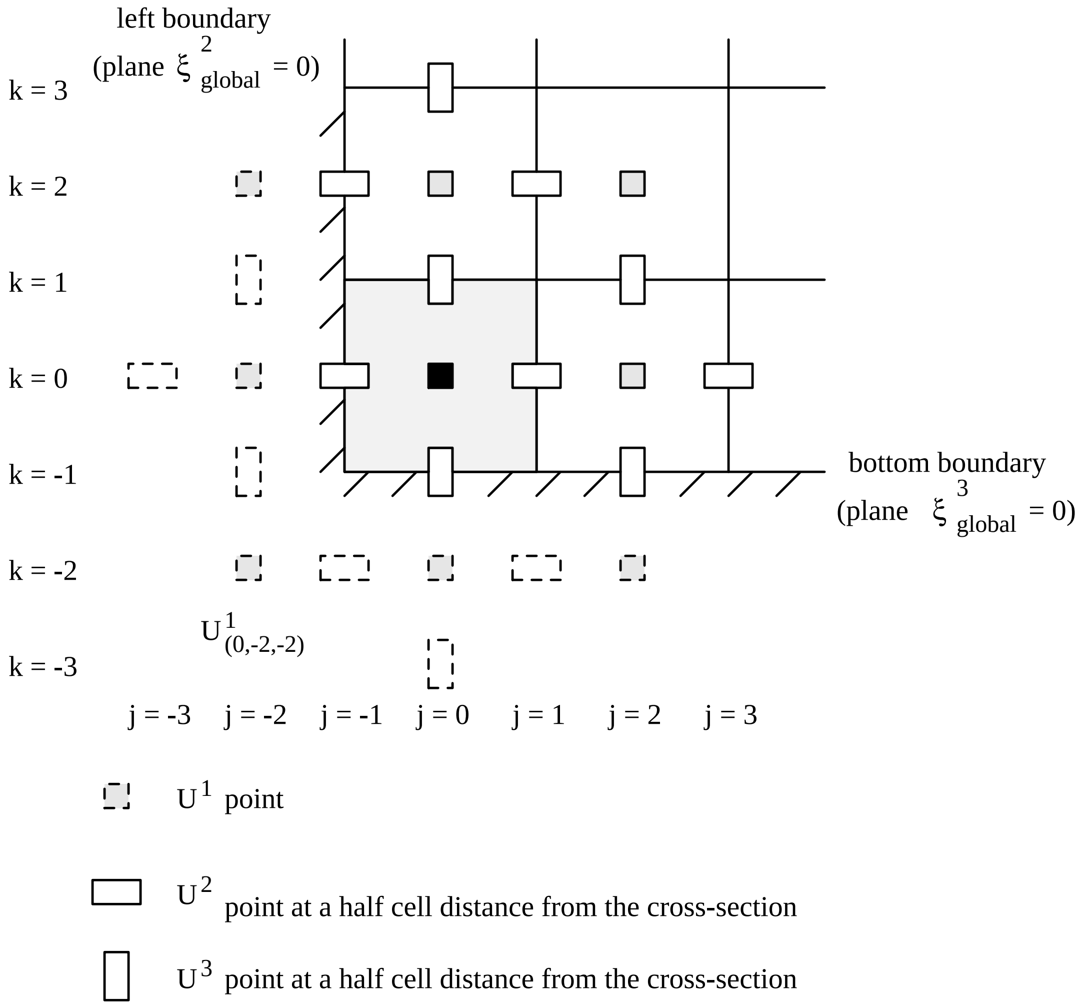


Figure 8.13: Cross-section over the "tangential" cell near the edge left under.

The virtual velocities are eliminated in the usual way [1] one exception that is for $V^1_{(0,-2,-2)}$. This virtual quantity can be extrapolated in the following way:

$$V^1_{(0,-2,-2)} = 4V^1_{(0,-1,-1)} - 2V^1_{(0,-1,0)} - 2V^1_{(0,0,-1)} + V^1_{(0,0,0)} \tag{8.66}$$

[1]

$$\begin{aligned}
V^\alpha_{(i,j,-2)} &= 2V^\alpha_{(i,j,-1)} - V^\alpha_{(i,j,0)} \quad \text{for} \quad \alpha = 1,2 \quad \text{and} \quad j \neq -2 \\
V^\alpha_{(i,-2,k)} &= 2V^\alpha_{(i,-1,k)} - V^\alpha_{(i,0,k)} \quad \text{for} \quad \alpha = 1,3 \quad \text{and} \quad k \neq -2 \\
V^2_{(i,-3,0)} &= 2V^2_{(i,-1,0)} - V^2_{(i,1,0)} \\
V^3_{(i,0,-3)} &= 2V^3_{(i,0,-1)} - V^3_{(i,0,1)} \quad .
\end{aligned}$$

or

$$V^1_{(0,-2,-2)} = 2V^1_{(0,-1,-1)} - V^1_{(0,0,0)} \,. \tag{8.67}$$

Both equations have the same order of accuracy.

Combination (ii)
Here we have to consider the "tangential" cell and the "normal" half-cell at the bottom boundary, see Figures 8.13 and 8.14.

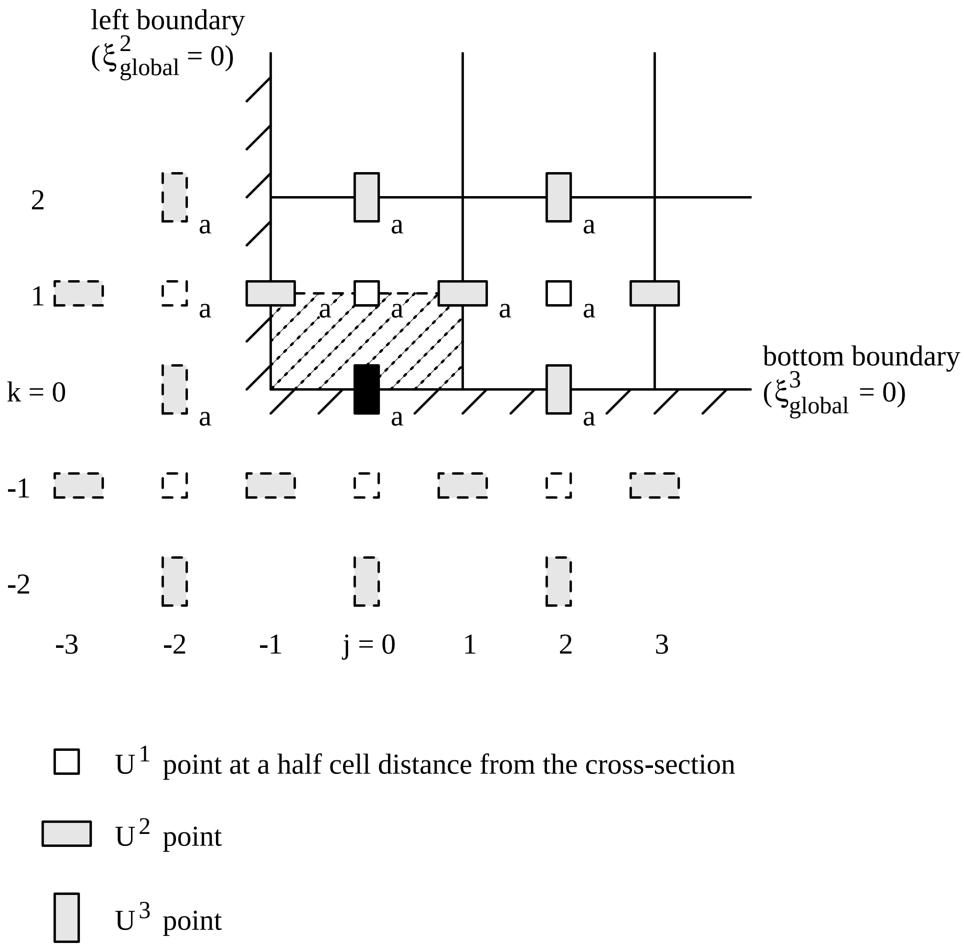


Figure 8.14: Cross-section over a "normal" U^3 half-cell.

"Tangential" U^1-cell.
The approach is here al most the same as the one prescribed in paragraph 8.2.2. Although there are more virtual unknowns they can be eliminated in the usual way. The $V^1_{(0,-2,-2)}$ velocity forms an exception in this case. We can only use equation (8.67) for the elimination of $V^1_{(0,-2,-2)}$, since V^1 is not prescribed at the bottom boundary.

"Normal" U^3 half-cell.
Only the velocities marked an a in Figure 8.14 appear in the discretization, since σ^{31}, σ^{32} and σ^{33} respectively U^1, U^2 and U^3 are given at the bottom boundary respectively left boundary. The approach is almost identical to the one given in paragraph 8.2.2, only the terms

$\frac{1}{2}\frac{\rho}{\sqrt{g}}V^3V^2|_{(0,-1,0)}$ and $\frac{\partial U^1}{\partial \xi^2}|_{(0,0,-1)}$ are treated in a different way. The first term is completely known since V^2 and V^3 are prescribed at the left boundary. So this term can be transported to the right-hand side.

The derivative $\frac{\partial U^1}{\partial \xi^2}|_{(0,0,1)} = \frac{1}{2}(U^1_{(0,2,1)} - U^1_{(0,-2,1)})$, introduced by the stress tensor can be computed using the standard elimination rules, so:

$$\frac{\partial U^1}{\partial \xi^2}|_{(0,0,1)} = \frac{1}{2}(U^1_{(0,2,1)} - (2U^1_{(0,-1,1)} - U^1_{(0,0,1)}) \tag{8.68}$$

where

$$U^1_{(0,i,j)} = \frac{1}{2}(U^1_{(-1,i,j)} + U^1_{(1,i,j)}) \; . \tag{8.69}$$

Combination (iii)

The U^1, U^2 and σ^{33} are given at the bottom boundary and U^1, U^2 and U^3 at the left boundary, so it is necessary to consider besides the "tangential" cell the "normal" U^3 half-cell.
The "tangential" cell can be treated in almost the same way as the "tangential" cell in combination (i), see also paragraph 8.2.2.
The discretization for the "normal" U^3 half-cell is almost given in paragraph 8.2.2. But now there are more virtual unknowns and some of them: $V^1_{(0,-2,-1)}, V^2_{(0,-3,-1)}$ and $V^3_{(0,-2,-2)}$ can not be eliminated in the usual way. The virtual unknowns $V^1_{(0,-2,-1)}$ and $V^1_{(0,-3,-1)}$ and $V^3_{(0,-2,-2)}$ (see Figure 8.14) can be eliminated by using one of the following equations:

$$V^1_{(i,-2,-1)} = 4V^1_{(i,-1,0)} - 2V^1_{(i,0,0)} - 2V^1_{(i,-1,1)} + V^1_{(i,0,1)} \tag{8.70}$$

for $i = -1$ or 1,

$$V^2_{(0,-3,-1)} = 4V^2_{(0,-1,0)} - 2V^2_{(0,1,0)} - 2V^2_{(0,-1,1)} + V^2_{(0,1,1)} \; , \tag{8.71}$$

and

$$V^3_{(0,-2,-2)} = 4V^3_{(0,0,0)} - 2V^3_{(0,2,0)} - 2V^3_{(0,0,2)} + V^3_{(0,2,2)} \; . \tag{8.72}$$

Combination (iv)

Here we have only to consider the "tangential" cell, since the "normal" velocities are given (U^2 at the left boundary and U^3 at the bottom boundary). For the treatment of the "tangential" cell, we refer to paragraph 8.4.2. All virtual unknowns (see Figure 8.13) exept of $V^1_{(0,-2,-2)}$ are eliminated in the usual way. For $V^1_{(0,-2,-2)}$ we can use formula (8.67).

Combination (v)

The stresses σ^{21}, σ^{22} and σ^{23} are prescribed at the left boundary and σ^{31}, σ^{32} and σ^{33} at the botom boundary. So we have to consider two "normal' half-cells and the "tangential" U^1-cell. See paragraph 8.2.2, for the treatment of the "tangential" U^1 cell (Figure 8.15) and the "normal" half-cells (Figure 8.14). However, in this case there are some differences:
Firstly, there are more virtual unknowns, but they caus no extra problems, since they can be eliminated by using the standard rules.
Secondly, the discretization of the stress tensor $\sigma^{\alpha\beta}$ for the "tangential" U^1-cell produces a difference. In formula (8.28) not only $\sigma^{13}|_{(0,0,-1)}$ is given, but also $\sigma^{12}_{(0,0,-1)}$.

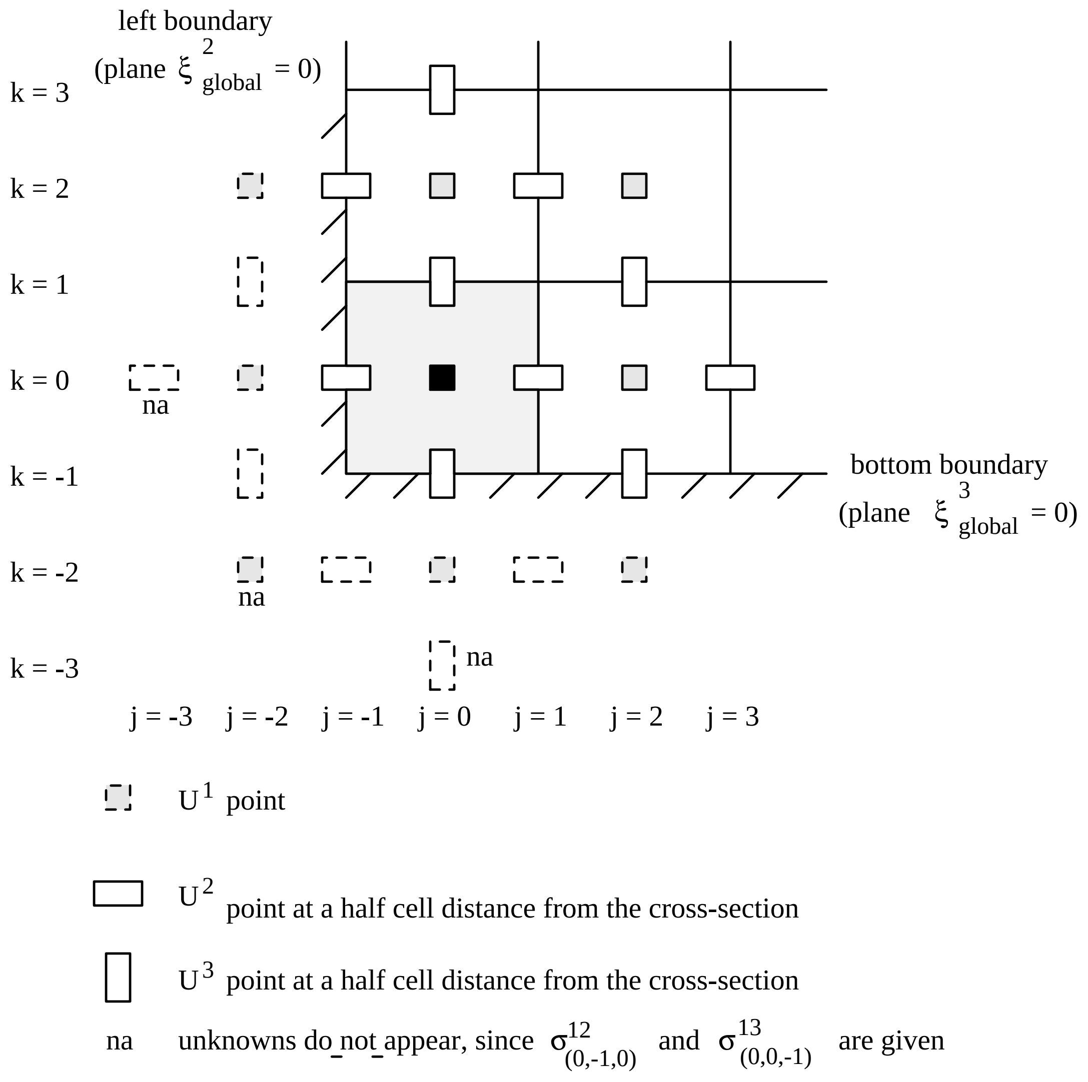


Figure 8.15: Cross-section over the "tangential" U^1-cell.

8.7 Boundary conditions for the transport equation

The boundary conditions for the transport equation are much easier to implement than the boundary conditions for the velocity components.

In the case of Dirichlet boundary conditions it is sufficient to use linear extrapolation to eliminate virtual scalars. So for example in Figure 8.16 we use the following formulae:

$$\phi_{.,0} = 2\phi_{.,1/2} - \phi_{.,1} \tag{8.73}$$

for normal boundary points and

$$\phi_{0,0} = \frac{1}{2}(2\phi_{1,0} - \phi_{2,0}) + \frac{1}{2}(2\phi_{0,1} - \phi_{0,2}) \tag{8.74}$$

for the corner points. In the case of a Robbins boundary condition we follow Van Kan *et al.*

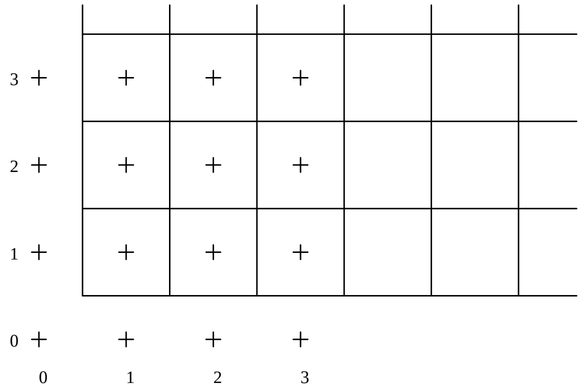


Figure 8.16: Cells for scalar quantities and corresponding virtual points.

(1991). This means that the Robbins boundary condition

$$k^{\alpha\beta}\phi_{,\beta}n_{\alpha} = b - \sigma\phi \tag{8.75}$$

is substituted in the diffusive term:

$$\begin{aligned} \int_{\Omega} -(k^{\alpha\beta}\phi_{,\beta})_{,\alpha} d\Omega &= -\int_{\Gamma} k^{\alpha\beta}\phi_{,\beta}n_{\alpha} d\Gamma \\ &= -\int_{\Gamma\backslash\Gamma_b} k^{\alpha\beta}\phi_{,\beta}n_{\alpha} d\Gamma - \int_{\Gamma_b} (b - \sigma\phi) d\Gamma \ , \end{aligned} \tag{8.76}$$

where Γ_b is the boundary at which the robbins boundary condition is given. The approximation of the first integral in the right-hand side of (8.76) is the same as for the inner region. The evaluation of the last term of the right-hand side of (8.76) on, for example, the left boundary of the domain is as follows:

$$\int_{\Gamma_b} (b - \sigma\phi) d\Gamma \approx \sqrt{g}\sqrt{g^{11}}(b - \sigma\phi) \tag{8.77}$$

In the case of non-smooth grids, the following approximation could be better used:

$$\begin{aligned} \int_{\Gamma_b} (b - \sigma\phi) d\Gamma &\approx \sqrt{(\sqrt{g}a_1^{(1)})^2 + (\sqrt{g}a_2^{(1)})^2}\ (b - \sigma\phi) \\ &= \sqrt{(a^2_{(2)})^2 + (a^1_{(2)})^2}\ (b - \sigma\phi) \end{aligned} \tag{8.78}$$

Virtual scalars ϕ are eliminated in the usual way. Similar expressions may be found for the three-dimensional case.

8.8 The wall function method

It is a well-known fact that integrating of the k-ε type models through the near-wall region and applying the no-slip condition yields unsatisfactory results. A way to overcome this deficiency is to introduce damping effects, resulting in a low-Reynolds-number form of these models, as outlined in the previous section. An alternative and still widely employed approach is the use of so-called wall functions, which model the near-wall region. Wall functions use empirical laws to circumvent the inability of the k-ε model to predict a logarithmic velocity profile near a wall. With these laws it is possible to express the mean velocity parallel to the wall and turbulence quantities outside the viscous sublayer in terms of the distance to the wall and wall conditions such as wall shear stress, pressure gradient and wall heat transfer. Hence, the wall functions can be used to provide near-wall boundary conditions for the momentum and turbulence transport equations, rather than conditions at the wall itself, so that the viscous sublayer does not have to be resolved and the need for a very fine mesh is circumvented. This method is proposed by Launder and Spalding [15].

The wall function method can be summarized as follows. The near-wall flow is modeled as a steady Couette flow. Experimental and dimensional analysis shows that the wall shear stress $\boldsymbol{\tau}_w$ is related to the mean velocity parallel to the wall through the so-called logarithmic law of the wall:

$$\boldsymbol{\tau}_w = \frac{\rho c_\mu^{1/4} \kappa \sqrt{k_{\mathrm{P}}}}{\ln(E Y_{\mathrm{P}}^+)} \boldsymbol{u}_{\mathrm{P}}^t \tag{8.79}$$

where the wall-coordinate Y^+ is given by

$$Y^+ = \frac{\rho c_\mu^{1/4} Y \sqrt{k}}{\mu} \tag{8.80}$$

Here, $\boldsymbol{u}^t$ is the tangential velocity vector, κ is the Von Kármán constant (≈ 0.4) and E is a roughness parameter, approximately equal to 9.0 for a smooth wall. The subscript P refers to the center of a cell adjacent to the wall. The location of the cell center away from the wall must be such that $Y_{\mathrm{P}}^+ > 11.3$ for the wall law (8.79) to be valid. Otherwise, it is calculated from the viscous sublayer profile:

$$\boldsymbol{\tau}_w = \frac{\mu}{Y_{\mathrm{P}}} \boldsymbol{u}_{\mathrm{P}}^t \tag{8.81}$$

These relations are accurate only for two-dimensional near-wall turbulent flows where local equilibrium prevails, but we shall use them also in more general circumstances, for lack of anything better of comparable simplicity.

The wall shear stress $\boldsymbol{\tau}_w$ can be employed as a boundary condition for the momentum equations, as follows:

$$S^{nt} = -|\boldsymbol{\tau}_w|, \quad \boldsymbol{t} = \frac{\boldsymbol{\tau}_w}{|\boldsymbol{\tau}_w|} \tag{8.82}$$

so that the tangential stress $S^{nt}\boldsymbol{t}$ is prescribed. The second condition is assumed to be

$$\boldsymbol{u} \cdot \boldsymbol{n} = 0 \tag{8.83}$$

The vector $\boldsymbol{u}^t$ can be obtained by subtracting the normal vector $\boldsymbol{u}^n$ from the velocity vector $\boldsymbol{u}$:

$$\boldsymbol{u}^t = \boldsymbol{u} - \boldsymbol{u}^n \tag{8.84}$$

with

$$\boldsymbol{u} = U^\alpha \boldsymbol{a}_{(\alpha)} \quad \text{and} \quad \boldsymbol{u}^n = (\boldsymbol{u} \cdot \boldsymbol{n})\boldsymbol{n} = \frac{U^n}{g^{nn}} \boldsymbol{a}^{(n)} \tag{8.85}$$

The contravariant velocity components U^α and U^n at cell centers are calculated by linear interpolations using the neighbouring points. The distance of a near-wall node P from a boundary surface can be found as the scalar product of a vector connecting a boundary point B and P and the unit normal vector $\boldsymbol{n}$ (see Figure 8.17):

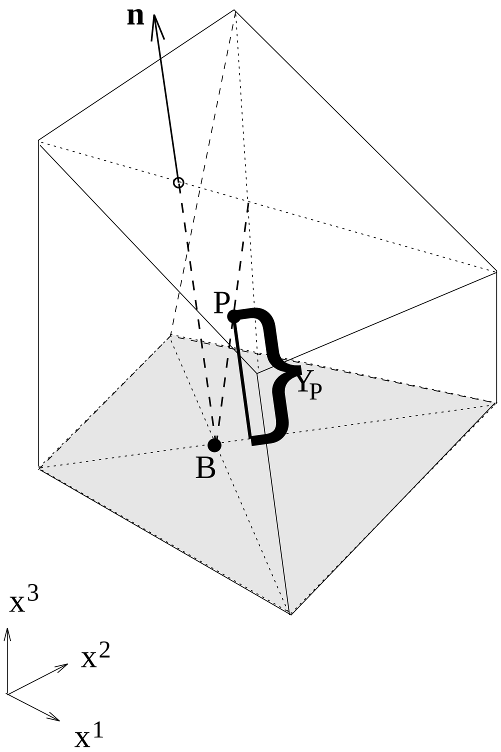


Figure 8.17: Calculation of normal distance Y_p between node P and boundary surface.

$$\mathrm{Y_P} = \boldsymbol{BP} \cdot \boldsymbol{n} = \frac{\boldsymbol{BP} \cdot \boldsymbol{a}_{\mathrm{B}}^{(n)}}{\sqrt{g_{\mathrm{B}}^{nn}}} \tag{8.86}$$

The coordinates of B and P are obtained from the coordinates of cell vertices by linear interpolations.

Remark: this method for the calculation of the normal wall distance is particularly meant for three dimensions. For the two-dimensional case, we refer to Section 7.3.3.

To ensure an accurate numerical representation of near-wall effects on the turbulent energy, special care is needed in evaluating the source terms in wall-adjacent cells. Let us consider the production term of the equation for turbulent energy k. Because the near-wall flow is modeled as steady Couette flow, the dominant contribution to the production is

$$P_k = \boldsymbol{\tau}_w \cdot \frac{\partial \boldsymbol{u}^t}{\partial Y} \tag{8.87}$$

Following Launder and Spalding [15], we assume that the local value of production at wall-adjacent cell center can be best obtained by averaging it over half of the near-wall cell:

$$\overline{P}_k = \frac{1}{Y_{\mathrm{P}}}\int_0^{Y_{\mathrm{P}}} \boldsymbol{\tau}_w \cdot \frac{\partial \boldsymbol{u}^t}{\partial Y}\,dY = \boldsymbol{\tau}_w \cdot \frac{\boldsymbol{u}^t_{\mathrm{P}}}{Y_{\mathrm{P}}} \tag{8.88}$$

assuming that $\boldsymbol{\tau}_w$ is constant across the near-wall cell. The dissipation rate of k in near-wall cells must be handled analogously. To evaluate the dissipation rate in the logarithmic layer, we take

$$\varepsilon = \frac{c_\mu^{3/4} k^{3/2}}{\kappa Y} \tag{8.89}$$

assuming local equilibrium, consistent with the use of the logarithmic law of the wall. Within the viscous sublayer we adopt the following expression:

$$\varepsilon = \rho c_\mu \frac{k^2}{\mu} \tag{8.90}$$

With (8.89) and (8.90) we can compute the average of the dissipation rate over half of the near-wall cell:

$$\overline{\varepsilon} = \frac{1}{Y_{\mathrm{P}}}\int_0^{Y_{\mathrm{P}}} \varepsilon dY = \begin{cases} c_\mu^{3/4} k_{\mathrm{P}}^{3/2} \frac{Y_{\mathrm{P}}^+}{Y_{\mathrm{P}}}, & Y_{\mathrm{P}}^+ < 11.3 \\ c_\mu^{3/4} k_{\mathrm{P}}^{3/2} \frac{\ln(E Y_{\mathrm{P}}^+)}{\kappa Y_{\mathrm{P}}}, & Y_{\mathrm{P}}^+ > 11.3 \end{cases} \tag{8.91}$$

Here, we assume that the variation of turbulent energy across the near-wall cell is negligible. The expressions (8.88) and (8.91) replace P_k and ε, respectively, which are source terms in the standard form of the equation for turbulent energy (7.7). Finally, the flux of turbulent energy through the wall is set to zero and, instead of solving the equation for ε, the value of ε at the first grid point away from the wall is determined from (8.89).

An important advantage of wall functions is that they allow inclusion of empirical information for special cases, such as wall roughness, pressure gradients and mass and heat transfer. Here, we shall discuss wall functions applicable to a rough wall. We consider a turbulent flow over rough surfaces. Let h_{R} denote the average height of roughness elements. We assume that the roughness has no influence on the flow except near the wall, i.e. $h_{\mathrm{R}}/L \ll 1$ where L is the characteristic length of the flow geometry. Following Tennekes and Lumley [32], the law of the wall for a rough wall is given by

$$\frac{|\boldsymbol{u}^t|}{|\boldsymbol{u}_\tau|} = \frac{1}{\kappa}\ln(\frac{Y}{h_{\mathrm{R}}}) + f(h_{\mathrm{R}}^+) \tag{8.92}$$

Here, $\boldsymbol{\tau}_w = \rho|\boldsymbol{u}_\tau|\boldsymbol{u}_\tau$ and h_{R}^+ is the roughness Reynolds number defined as

$$h_{\mathrm{R}}^+ = \frac{\rho|\boldsymbol{u}_\tau|h_{\mathrm{R}}}{\mu} \tag{8.93}$$

A generalized form of the wall law (8.92) can be found with the aid of a one-equation model in which the transport equation is provided for the turbulent energy k:

$$\frac{|\boldsymbol{u}^t|\, c_\mu^{1/4}\sqrt{k}}{|\boldsymbol{\tau}_w|/\rho} = \frac{1}{\kappa}\ln(\frac{Y}{h_{\mathrm{R}}}) + f(h_{\mathrm{R}}^+) \quad \text{and} \quad h_{\mathrm{R}}^+ = \frac{\rho c_\mu^{1/4}\sqrt{k}h_{\mathrm{R}}}{\mu} \tag{8.94}$$

This law of the wall (8.94) is preferable because it allows non-equilibrium effects on the turbulent energy k. If the roughness elements are submerged in the viscous sublayer then the turbulence will not be affected by the roughness. In other words, the wall can be considered as smooth. Thus, in the limit $h_{\mathrm{R}}^{+} \to 0$, we should have

$$f(h_{\mathrm{R}}^{+}) \to \frac{1}{\kappa}\ln(h_{\mathrm{R}}^{+}) + 5.5 \quad \text{as } h_{\mathrm{R}}^{+} \to 0 \tag{8.95}$$

On the other hand, Nikuradse (see [24]) found by means of experiments that when the surface is very rough, i.e. for large values of h_{R}^{+}, the function $f(h_{\mathrm{R}}^{+})$ becomes independent of h_{R}^{+}, viz.,

$$f(h_{\mathrm{R}}^{+}) \to 8.5 \quad \text{if } h_{\mathrm{R}}^{+} > 30 \tag{8.96}$$

If the roughness elements are submerged in the buffer layer (a transition region between the viscous sublayer and the log layer), i.e. $5 < h_{\mathrm{R}}^{+} < 30$, then the function f depends on the roughness Reynolds number. However, in many engineering calculations, the buffer layer is ignored. Hence, the location of the edge of viscous sublayer and log layer is taken equal to $Y^{+} = 11.3$, which value is obtained by simply linking the linear velocity profile in the viscous sublayer to the logarithmic velocity profile in the log layer. Thus, for $h_{\mathrm{R}}^{+} < 11.3$, the wall is considered to be smooth, otherwise the wall is rough. The generalized law of the wall for rough walls then becomes:

$$\rho \boldsymbol{u}^{t}\, c_{\mu}^{1/4}\sqrt{k} = \frac{\boldsymbol{\tau}_{w}}{\kappa}\ln(E_{r}\frac{Y}{h_{\mathrm{R}}}), \quad h_{\mathrm{R}}^{+} > 11.3 \tag{8.97}$$

where $E_r \approx 30$. From this wall law the wall shear stress can be computed, which can be used as boundary condition for the momentum equations (see (8.82)).

The rapid variation of turbulence quantities also necessitates special measures in evaluating the production and dissipation rates of turbulent kinetic energy near the rough wall. The average production and dissipation rates used in the near-wall cells have the following form:

$$\overline{P}_{k} = \boldsymbol{\tau}_{w} \cdot \frac{\boldsymbol{u}^{t}}{Y} \quad \text{and} \quad \overline{\varepsilon} = c_{\mu}^{3/4} k^{3/2} \frac{\ln(E_{r}Y/h_{\mathrm{R}})}{\kappa Y} \tag{8.98}$$

9 Time-discretization

9.1 Introduction

After application of the space discretization of momentum and transport equations and the implementation of the boundary conditions as described in Section 7, the discretized equations in the time-domain read:

$$M\frac{d\boldsymbol{V}}{dt} + S(\boldsymbol{V}, \boldsymbol{\phi}_1, .., \boldsymbol{\phi}_{\mathrm{N}}) + G\boldsymbol{p} = \boldsymbol{F} \tag{9.1}$$

$$D\boldsymbol{V} = \boldsymbol{0} \tag{9.2}$$

$$M\frac{d\boldsymbol{\phi}_i}{dt} + T_i(\boldsymbol{V}, \boldsymbol{\phi}_1, .., \boldsymbol{\phi}_{\mathrm{N}}) = \boldsymbol{S}_i + \boldsymbol{S}_{dc,i}, \quad \forall i \in \{1, 2, .., \mathrm{N}\} \tag{9.3}$$

where $\boldsymbol{V}$ and $\boldsymbol{p}$ denote algebraic vectors containing the velocity and pressure unknowns in grid points, $\boldsymbol{\phi}_i$ is the i^{th} discrete scalar grid function and the total number of scalar unknowns is given by N. For two-equation turbulence models in the absence of other physical effects, N = 2. Furthermore, M is the diagonal matrix containing the value of ρ in the centroids on the diagonal, D and G are the discretized divergence and gradient operators, S represents the space discretization of the convection and viscous stress tensors and T_i is an operator involving the discretization of convection and diffusion of the i^{th} scalar. In fact this term may also be non-linear, but in our program it is treated as if it is linear. The vector $\boldsymbol{F}$ contains the volume forces and boundary values of the velocities and $\boldsymbol{S}_i$ represents the source term with respect to $\boldsymbol{\phi}_i$, which is generally a function of $\boldsymbol{V}$ and $\boldsymbol{\phi}_j$, $\forall j \in \{1, .., \mathrm{N}\}$ and the boundary conditions. The extra source term $\boldsymbol{S}_{dc,i}$ result from the anti-diffusive parts as deferred corrections to the first order upwind approximation.

The time discretization is performed with a standard technique for the solution of ordinary differential equations. At this moment only one type of time-solver is present: the so-called θ method, i.e. a linear combination of the forward and backward Euler schemes.

Generalized θ method ??????? (Jos, Guus)

9.2 The θ-method

In this section we restrict ourselves to the time discretization of the momentum equations. The treatment of the transport equations will be done in exactly the same manner. Furthermore, for brevity the arguments $\boldsymbol{\phi}_1, .., \boldsymbol{\phi}_{\mathrm{N}}$ will be dropped from the operator S. Application of the θ-method to (9.1), (9.2) gives

$$\begin{aligned} M\frac{\boldsymbol{V}^{n+1} - \boldsymbol{V}^n}{\Delta t} + \theta S(\boldsymbol{V}^{n+1}) \quad &+ \quad (1-\theta)S(\boldsymbol{V}^n) + \theta G\boldsymbol{p}^{n+1} + (1-\theta)G\boldsymbol{p}^n \\ &= \quad \theta\boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n \qquad (9.4) \\ D\boldsymbol{V}^{n+1} \quad &= \quad \boldsymbol{0} \qquad (9.5) \end{aligned}$$

where θ lies between zero and unity, n denotes the preceding time level, $n+1$ the new time level and Δt is the time step. For $\theta = 0$ and $\theta = 1$ we obtain the first order explicit and implicit Euler schemes, respectively, and for $\theta = \frac{1}{2}$ we have the second order Crank-Nicolson scheme. The θ-method is unconditionally stable for $0.5 \leq \theta \leq 1$. In the range $0 \leq \theta < 0.5$ a time-step restriction is necessary. At this moment $\theta < 0.5$ has not been tested, except $\theta = 0$.

To solve (9.4), (9.5) it is necessary to linearize the term $S(\boldsymbol{V}^{n+1})$. In ISNaS, the convective terms are linearized by a Newton linearization as given in formula (4.7). Coefficients that depend on the solution, like for example the viscosity, are evaluated at the preceding time-level.

Practical implementation:

Instead of solving (9.4), (9.5) immediately, we introduce an intermediate level $n + \theta$ by:

$$\begin{aligned} \boldsymbol{V}^{n+\theta} &= \theta \boldsymbol{V}^{n+1} + (1-\theta)\boldsymbol{V}^{n} \\ \boldsymbol{p}^{n+\theta} &= \theta \boldsymbol{p}^{n+1} + (1-\theta)\boldsymbol{p}^{n} \\ \boldsymbol{F}^{n+\theta} &= \theta \boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^{n} \end{aligned} \tag{9.6}$$

If we assume that $S(\boldsymbol{V})$ is linearized, i.e. can be written as $S(\boldsymbol{V}^{n+1}) \simeq A(\boldsymbol{V}^n) + B(\boldsymbol{V}^n)\boldsymbol{V}^{n+1}$, then (9.4) reduces to:

$$\begin{aligned} M\frac{\boldsymbol{V}^{n+1} - \boldsymbol{V}^{n}}{\Delta t} &+ \theta B(\boldsymbol{V}^n)\boldsymbol{V}^{n+1} + (1-\theta)B(\boldsymbol{V}^n)\boldsymbol{V}^n + \theta G\boldsymbol{p}^{n+1} + (1-\theta)G\boldsymbol{p}^n \\ &+ A(\boldsymbol{V}^n) = \theta \boldsymbol{F}^{n+1} + (1-\theta)\boldsymbol{F}^n \end{aligned} \tag{9.7}$$

Substitution of (9.6) into (9.7) and (9.5) gives

$$M\frac{\boldsymbol{V}^{n+\theta} - \boldsymbol{V}^{n}}{\theta \Delta t} + B(\boldsymbol{V}^n)\boldsymbol{V}^{n+\theta} + G\boldsymbol{p}^{n+\theta} = \boldsymbol{F}^{n+\theta} - A(\boldsymbol{V}^n) \tag{9.8}$$

$$D\boldsymbol{V}^{n+\theta} = \boldsymbol{0} \tag{9.9}$$

From (9.6) it then follows that:

$$\begin{aligned} \boldsymbol{V}^{n+1} &= \frac{1}{\theta}(\boldsymbol{V}^{n+\theta} - (1-\theta)\boldsymbol{V}^{n}) \\ \boldsymbol{p}^{n+1} &= \frac{1}{\theta}(\boldsymbol{p}^{n+\theta} - (1-\theta)\boldsymbol{p}^{n}) \end{aligned} \tag{9.10}$$

Once the momentum equations have been solved for t^{n+1}, each of the transport equations for $\boldsymbol{\phi}_i$ is solved for one time step. Exactly the same methodology as described above is employed. Quantities already computed, like the velocity are substituted in these equations, thus improving the stability. However, it should be noted that the source term $\boldsymbol{S}_{dc,i}$ contains deferred corrections which are evaluated explicitly, so that no contribution to coefficients of the system to be solved for the new time level is involved.

9.3 The solution algorithm

The spatial and temporal discretizations yield a set of coupled algebraic equations for the velocity, pressure and scalar quantities. One should have some idea about how this coupled set of equations is going to be solved. The choice affects the coupling of the various unknowns. Basically, the velocity and pressure fields are coupled through the continuity equation. For turbulent flows, the momentum equations are coupled to turbulence transport equations through the eddy-viscosity. Also a strong coupling exists between the turbulence equations.

In principle, there are two approaches for the solution of the coupled set of discretized equations. The first one is to solve all equations simultaneously at each grid point. By contrast, an uncoupled solution technique proceeds sequentially through the equations by treating the other variables as known until the converged solution of the coupled set of equations is obtained. The coupled solution method requires a very large computer memory, but may have better rate of convergence and numerical stability than the uncoupled one. Nevertheless, solving all equations simultaneously may be so complicated that coupled solution procedures are difficult to use. It may then be preferable to employ uncoupled methods or a blended form of both strategies.

Here, the following overall solution algorithm will be used. For each time step, the process start by guessing the variables V^α, p and $\phi_i, \forall i \in \{1, .., N\}$, either initially or from the previous time level. Note that the guessed velocity must satisfy the incompressibility constraint. Then the continuity equation and the coupled momentum equations are solved using the non-updated eddy-viscosity, if applicable. To ensure a divergence-free velocity field the pressure correction scheme as will be outlined in Section 10 is used. Note that the linear momentum and pressure correction equations are solved sequentially. Because of the nonlinearities this loop ($V^\alpha \to p$) may be repeated until a converged nonlinear result is obtained, but one Newton iteration in each time step is sufficient. Finally, the transport equations and then turbulence equations are solved in a decoupled way using the updated mean flow quantities and non-updated eddy-viscosity, if applicable. The transport equations are solved in the sequence given by their index number, whereas the equation for ε or ω is solved after k. It may be necessary to repeat this loop ($k \to \varepsilon$ or ω) in each time step until convergence is reached. In addition, the outer loop ($V^\alpha \to p \to \phi_1 \to \,..\, \to \phi_N \to k \to \varepsilon$ or ω), which contains three inner loops ($V^\alpha \to p$), ($\phi_1 \to \,..\, \to \phi_N$) and ($k \to \varepsilon$ or ω) coupled via turbulent viscosity, may be repeated until all variables at time level $n+1$ converge. However, at this moment one inner and one outer iteration cycle per time step suffice.

10 Pressure correction

10.1 Introduction

An essential difficulty in the solution of the coupled momentum equations and continuity equation (9.1), (9.2) or its (time discretized form (for example (9.8), (9.9), is the absence of the pressure in the continuity equation. If we consider the system of equations as one large system of linear equations to be solved, this means that in the part corresponding to the continuity equations we have zeros at the main diagonal. Formally equations (9.8), (9.9) may be written as:

$$\begin{bmatrix} S & G \\ D & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^{n+\theta} \\ \boldsymbol{p}^{n+\theta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}_1^{n+\theta} \\ \boldsymbol{F}_2^{n+\theta} \end{bmatrix} , \tag{10.1}$$

where $\boldsymbol{F}_2^{n+\theta}$ is only non-zero if non-zero Dirichlet boundary conditions for the velocity are prescribed.

The solution of systems of equations of the form (10.1) is in general more difficult for a linear solver than the solution of equations arising from the discretization of standard convection-diffusion equations. There are several ways to solve this problem. One of the possible ways is to perturb the continuity equation. This leads to methods like the penalty method or Uzawa iterations. An alternative way to solve the problem is formed by projection methods. In these methods first the pressure at the new level is estimated, for example by the old pressure, and then the momentum equations are solved yielding an intermediate velocity field. By projecting this velocity onto the space of divergence-free vector fields a new velocity and pressure may be computed. An important representant of this class is the so-called pressure-correction method, which will be treated in 10.2.

10.2 The pressure-correction method

The pressure-correction method as implemented in the ISNaS incompressible code is the one described in Van Kan *et al.* (1991). Starting point is the θ-method formulated by (9.4), (9.5) or the variant (9.8), (9.9).

Following Van Kan *et al.* (1991) we define an intermediate velocity $\boldsymbol{V}^*$ by:

$$M\frac{\boldsymbol{V}^* - \boldsymbol{V}^n}{\theta\Delta t} + B(\boldsymbol{V}^n)\boldsymbol{V}^* + G\boldsymbol{p}^n = \boldsymbol{F}^{n+\theta} - A(\boldsymbol{V}^n) \tag{10.2}$$

$\boldsymbol{V}^*$ must be such that the boundary conditions at $t = t^n + \theta\Delta t$ are satisfied. In the case of prescribed normal velocities this means that the corresponding rows in the matrix G contain zeros.

Subtraction of (10.2) from (9.8) gives

$$M\frac{\boldsymbol{V}^{n+\theta} - \boldsymbol{V}^*}{\theta\Delta t} = -G(\boldsymbol{p}^{n+\theta} - \boldsymbol{p}^n) , \tag{10.3}$$

where the term $B(\boldsymbol{V}^n)(\boldsymbol{V}^{n+\theta} - \boldsymbol{V}^*)$ has been neglected.
Application of (9.9) to (10.3) gives

$$D\boldsymbol{V}^* = \theta\Delta t D M^{-1} G(\boldsymbol{p}^{n+\theta} - \boldsymbol{p}^n) , \tag{10.4}$$

which is a Laplacian-type equation for the pressure correction. Once $\boldsymbol{p}^{n+\theta}$ has been computed $\boldsymbol{V}^{n+\theta}$ follows from (10.3):

$$\boldsymbol{V}^{n+\theta} = \boldsymbol{V}^{*} - \theta \Delta t M^{-1} G(\boldsymbol{p}^{n+\theta} - \boldsymbol{p}^{n}) \tag{10.5}$$

Remark: the matrix $DM^{-1}G$ is in general non-symmetrical.

11 The linear solver

11.1 Introduction

The discretization of the incompressible Navier-Stokes equations in general curvilinear coordinates is described in the foregoing sections. The space discretization consists of a finite volume technique on a structured grid. The motivation for these choices is that we want to solve large two and three dimensional problems. In these problems it is important to obtain fast iterative methods to solve the discretized equations. This is easier using a finite volume technique instead of a finite element technique. Finally the structured grid enables us to develop a good implementation of the methods on vector computers.

The linear systems to be solved are [38, 39]:

the momentum equations

$$M^{n+1}u^{n+1} = f^{n+1} \quad , \quad u^{n+1} = \begin{pmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \end{pmatrix} ,$$

the pressure equation

$$P\Delta p^{n+1} = g^{n+1} \quad , \quad \Delta p^{n+1} = p^{n+1} - p^n ,$$

and eventually one or more transport equations:

transport equations

$$\begin{aligned} C_1^{n+1} \quad c_1^{n+1} &= d_1^{n+1} , \\ &\vdots \\ C_k^{n+1} \quad c_k^{n+1} &= d_k^{n+1} . \end{aligned}$$

Suppose n_i is the number of grid points in the x_i-direction, where we take $n_3 = 1$ for a 2-dimensional problem. The pressure and transport matrices have $n_1 \cdot n_2 \cdot n_3$ rows and columns. The dimension of the momentum matrix is $2 \cdot n_1 \cdot n_2$ in 2-dimensional problems and $3 \cdot n_1 \cdot n_2 \cdot n_3$ in 3-dimensional problems.

For the structure of the matrices in 2-dimensions we refer to Vuik (1992) and Vuik (1993). In the 3-dimensional case the nonzero structure is symmetric for all matrices. In 3 dimensions the structure of the pressure equation is given in Figure 11.1.
Note that the nonzero structure is symmetric. The momentum matrix can be partitioned in the following form:

$$\begin{pmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} .$$

The structure of M_{ii}, $i = 1, 2, 3$ is the same as for the pressure equations. The off-diagonal blocks contain 16 non zero diagonals. The non zero structure of the momentum matrix is non symmetric. To illustrate this we give M_{12} and M_{21} in Figures 11.2 and 11.3 and note the non zero structure of M_{12} is not equal to the non zero structure of M_{21}^T.

In the following table we summarize the number of non zero elements in some matrices.
In three dimensions the momentum matrix is much larger than the pressure matrix. The

	2D	3D
pressure matrix	$9 \cdot n_1 \cdot n_2$	$19 \cdot n_1 \cdot n_2 \cdot n_3$
momentum matrix	$13 \cdot 2 \cdot n_1 \cdot n_2$	$51 \cdot 3 \cdot n_1 \cdot n_2 \cdot n_3$

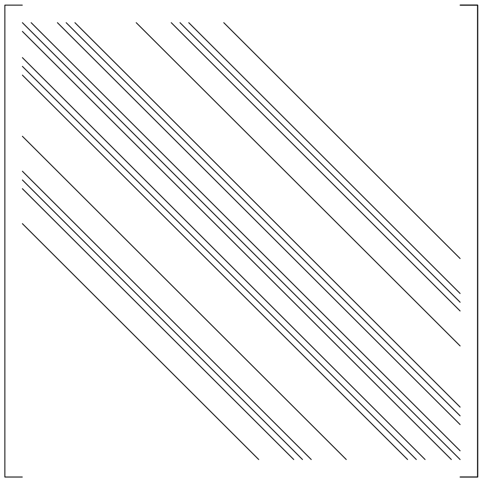


Figure 11.1: The pressure matrix P

ratio in 2D is equal to $\frac{13 \cdot 2}{9} = 3$ whereas the ration in 3D is equal to $\frac{51 \cdot 3}{19} = 8$.

So in 3D a momentum matrix times vector is 8 times as expensive as a pressure matrix times vector.

The momentum matrix and the transport matrix depend on the time t. In many problems the pressure matrix is independent of the time. However, this property of the pressure matrix is not used in the current implementation.

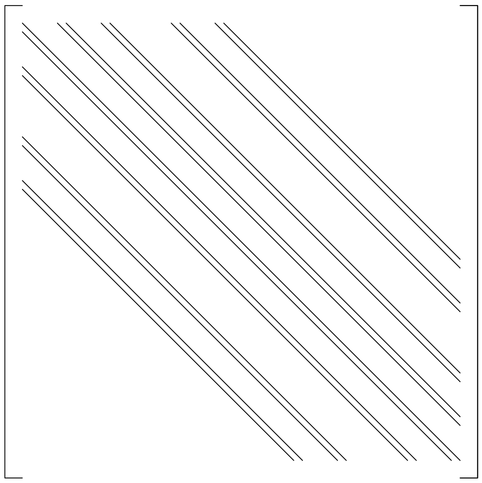


Figure 11.2: The momentum-matrix M_{12}

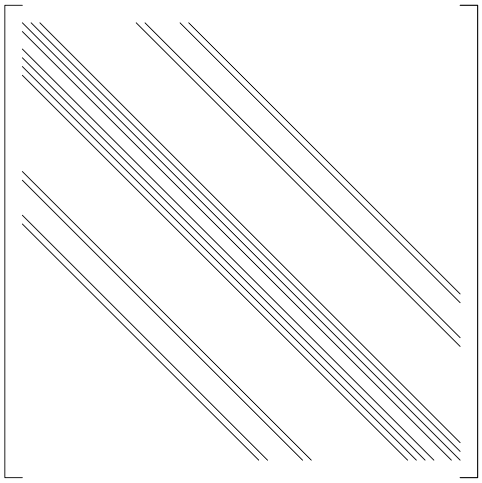


Figure 11.3: The momentum-matrix M_{21}

11.2 Survey of iterative methods

The systems given in Section 11.1 are solved with iterative methods of CG-type. All the methods used in ISNaS can be applied to unsymmetric matrices. The methods used in ISNaS are:

LSQR
This is a stable implementation of CG applied to the normal equations [20].

CGS
CGS is an iterative method based on the Bi-Lanczos algorithm [28].

GMRES
An iterative method, which computes an approximation with a minimal residual [23].

GMRESR
A method based on GMRES, but in general cheaper with respect to work and memory [35]. In Table 11.2 we summarize the properties of the iterative methods. This table only gives

properties	bad $\longleftarrow$		good $\longrightarrow$	
memory	GMRES	GMRESR	CGS	LSQR
robustness	CGS	GMRES	GMRESR	LSQR
CPU-time	LSQR	GMRES	CGS	GMRESR

Table 11.2: Properties of the iterative methods

an indication of the properties. So in many experiments the results agree with Table 11.2. However, for specific problems the results may be different.

Stopping criteria

For iterative methods it is necessary to specify a stopping criterion. In general the norm of the residual: $\|r_k\|_2 = \|b - Ax_k\|_2$ is easy to obtain. So all our stopping criteria are based on $\|r_k\|_2$. For the different equations we recommend different stopping criteria. For the details we refer to Vuik (1992), p. 8 for the momentum equations, Vuik (1992), p.13 for the pressure equation, and Vuik (1992), p. 15 for a transport equation.

Starting vector

Finally we have to choose a starting vector for the iterative methods. Since we solve the systems for every timestep, the solution of the foregoing timestep is in general a good starting vector. For the details we refer to Vuik (1992), p. 6, 7 for the momentum equations, Vuik (1992), p. 13 for the pressure equation, and Vuik (1992), p. 15 for a transport equation.

11.3 Preconditioning

In many applications, iterative methods are combined with a preconditioner [17]. It is a well known fact that a good preconditioner is very important in order to obtain fast iterative methods. The preconditioners used in ISNaS are based on incomplete LU decompositions. In such a preconditioner, one constructs a lower triangular matrix L and an upper triangular matrix U, where L and U have a prescribed nonzero pattern, and LU is a good approximation of A. The iterative methods can be applied to

$$U^{-1}L^{-1}Ax = U^{-1}L^{-1}b \,, \tag{11.1}$$

$$AU^{-1}L^{-1}y = b \,, \tag{11.2}$$

or

$$L^{-1}AU^{-1}y = L^{-1}b \,. \tag{11.3}$$

We call equation (11.1) a preconditioned system and equation (11.2) a postconditioned system. The final equation is only used in combination with the Eisenstat implementation [5]. In general, the convergence behaviour of a Krylov type iterative method depends on the eigenvalue distribution of the matrix. In the three equations given above the eigenvalues of the product-matrices are the same. So the convergence behaviour is approximately the same when we use (11.1), (11.2), or (11.3). A small advantage of a postconditioned system is that the norm of a residual is not influenced by the matrices L and U (compare [38], p. 12, 13).

Below we give a short description of the preconditioners used in ISNaS.

Diagonal scaling

A diagonal preconditioner is obtained by choosing $L = I$ and $U = diag\ (A)$. This is a cheap preconditioner with respect to memory and can be used in combination with vector and parallel computers. For most problems the gain in the number of iterations is small.

ILUD

For this preconditioner we construct $LD^{-1}U$ as an approximation of A. To obtain L, D, and U we use the following rules [34]:

- $diag\ (L) = diag\ (U) = D$;

- the off-diagonal parts of L and U are equal to the corresponding parts of A;

- $diag\ (LD^{-1}U) = diag\ (A)$.

The third rule can be replaced by the following:

$$\text{rowsum } (LD^{-1}U) = \text{ rowsum } (A) \quad \text{for every row },$$

which leads to the MILU preconditioning. We always use an average of ILUD and MILUD. This preconditioner is also cheap with respect to memory. It costs two extra vectors, one for D and the other one for D^{-1}. Using the Eisenstat implementation we are able to save one matrix vector product per iteration. In the ISNaS program ILUD preconditioner means application of the iterative method to (11.3) and not to (11.1), which is done in the other preconditioners. Multiplication with L^{-1} and U^{-1} leads to recurrencies. So these parts do not run in vector speed on a vector computer.

ILU
This preconditioner is only used for the pressure and transport equations. The matrices L and U are constructed such that LU approximates A and satisfies the following rules:

- $diag\ (L) = I$;

- the structure of L and U is comparable to the structure of A;

- if $a_{ij} \neq 0$ then $(LU)_{ij} = a_{ij}$.

Again the last rule for $i = j$ can be repaced by

- rowsum (LU) = rowsum (A),

which leads to MILU. We always use an average of ILU and MILU. The convergence behaviour of an iterative method combined with MILU is in general beter than a combination with MILUD. A disadvantage is that extra memory space is needed to store L and U. The amount of extra memory is the same as the amount of memory to store A. Furthermore it is impossible to save a matrix vector product per iteration (compare the Eisenstat implementation). From our experiments we conclude that if the memory space is available than it is better to use MILU.

Memory space
During the solution of the pressure or transport equation the memory space of the momentum matrix is available. For this reason we always use the MILU preconditioner to solve the pressure and transport equations, and the MILUD preconditioner for the momentum equations.

Vectorization
Due to the recurrencies, the multiplication of L^{-1} or U^{-1} for MILUD or MILU runs in scalar speed on a vector computer. In the ISNaS program the loops are rewritten in such a way that they run in vectorspeed [1]. Note that the rewritten loops use indirect adressing and are much shorter than the original loops. On the Convex C3840 this leads to good results.

11.4 Concluding remarks

Not all the combinations described in the foregoing sections are implemented. In the User Manual all the implemented combinations are summarized.

12 Post-processing

The ISNaS incompressible program computes the fluxes V^α in the midside points of the cells and the scalars in the centroids. For post-processing purposes these quantities are needed in the vertices of the cells. For that reason it is necessary to interpolate (or at the boundary extrapolate) the computed values to the vertex points. Numerical examples have shown that a straighforward interpolation in the computational space is not accurate enough. For that reason a weighted approach, taking into account the distances in physical space, is necessary. In the following sections we consider the interpolation and backtransformation applied both for scalar quantities and for the fluxes.

12.1 Interpolation of scalars in 2D

The scalar unknowns are positioned in the centroids of the cells. In order to interpolate these values to the vertices a weighted mean value of the four surrounding cells is used. Figure 12.1 sketches a typical example. In this figure point i,j is the vertex in which the interpolated

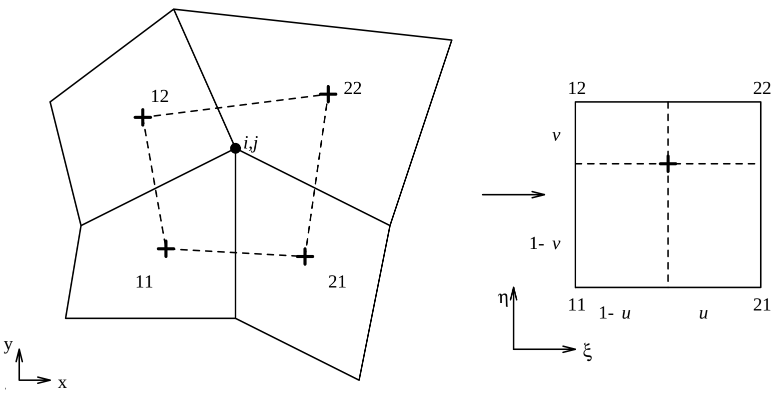


Figure 12.1: Vertex i,j with four surrounding cells and mapping of quadrilateral formed by centroids on to a square.

values must be computed. This point is part of 4 cells with centroids 11, 21, 12 and 22. In order to compute the interpolated value, the quadrilateral spanned by the 4 centroids is mapped onto a unit square (0, 1) × (0,1) by a bilinear mapping as is usual in finite elements. So

$$\begin{bmatrix} x \\ y \end{bmatrix} = \sum_{i,j=1}^{2} \lambda_i(\xi)\lambda_j(\eta) \begin{bmatrix} x \\ y \end{bmatrix}_{ij}, \tag{12.1}$$

with $\lambda_1(\xi) = 1 - \xi, \quad \lambda_2(\xi) = \xi$

The value of the scalar in (x, y) is computed by

$$\text{Scalar } (x,y) = \sum_{i,j=1}^{2} \lambda_i(\xi)\lambda_j(\eta) \text{ scalar } (x_{ij}, y_{ij}) \tag{12.2}$$

To evaluate (12.2) it is necessary to know the value of (ξ, η) in point (x, y). This value can be computed from (12.1) by solving this system of non-linear equation with a Newton-Raphson

method.
Define

$$\begin{bmatrix} F_1(\xi,\eta) \\ F_2(\xi,\eta) \end{bmatrix} = \sum_{i,j=1}^{2} \lambda_i(\xi)\lambda_j(\eta) \begin{bmatrix} x \\ y \end{bmatrix}_{ij} - \begin{bmatrix} x \\ y \end{bmatrix} \tag{12.3}$$

The Newton-Raphson method can be written as:

$$(\xi,\eta)^1 = (1/2, 1/2)$$

$$\begin{bmatrix} \partial F_1/\partial\xi & \partial F_1/\partial\eta \\ \partial F_2/\partial\xi & \partial F_2/\partial\eta \end{bmatrix}^n (\begin{bmatrix} \xi \\ \eta \end{bmatrix}^{n+1} - \begin{bmatrix} \xi \\ \eta \end{bmatrix}^n = - \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}^n \quad n = 1, 2, ... \tag{12.4}$$

Since Newton is a fast converging process, the maximal number of iterations is restricted to 5. At this moment the iterations is stopped if $\|\boldsymbol{\xi}^{n+1} - \boldsymbol{\xi}^n\| < 0.001$.

From (12.3) it follows that:

$$\begin{aligned} \boldsymbol{F}(\xi,\eta) &= (1-\xi)(1-\eta)\boldsymbol{x}_{11} + \xi\eta\boldsymbol{x}_{21} + (1-\xi)\eta\boldsymbol{x}_{12} - \boldsymbol{x} \\ &= \boldsymbol{x}_{11} + (\boldsymbol{x}_{11} + \boldsymbol{x}_{22} - \boldsymbol{x}_{21} - \boldsymbol{x}_{12})\xi\eta + (\boldsymbol{x}_{21} - \boldsymbol{x}_{11})\xi + (\boldsymbol{x}_{12} - \boldsymbol{x}_{11})\eta - \boldsymbol{x} \end{aligned} \tag{12.5}$$

and

$$\frac{\partial \boldsymbol{F}}{\partial \xi} = (\boldsymbol{x}_{11} + \boldsymbol{x}_{22} - \boldsymbol{x}_{21} - \boldsymbol{x}_{12})\eta + \boldsymbol{x}_{21} - \boldsymbol{x}_{11} \tag{12.6}$$

$$\frac{\partial \boldsymbol{F}}{\partial \eta} = (\boldsymbol{x}_{11} + \boldsymbol{x}_{22} - \boldsymbol{x}_{21} - \boldsymbol{x}_{12})\xi + \boldsymbol{x}_{12} - \boldsymbol{x}_{11} \tag{12.7}$$

With respect to the boundary points it is not longer possible to use an interpolation. In that case an extrapolation is used. Figure 12.2 shows the four points that are used to compute the value at an under boundary.

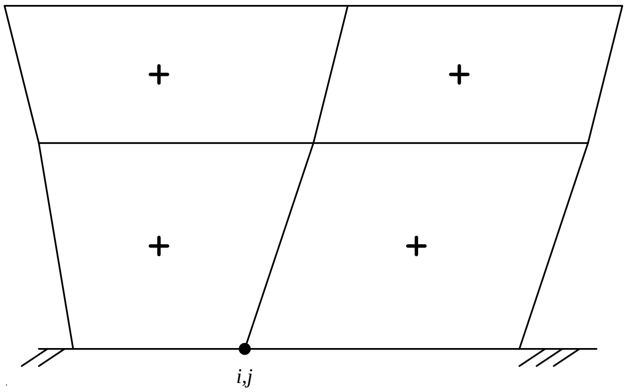


Figure 12.2: Cells that are used to extrapolate the scalar value at the under boundary i, j.

12.2 Interpolation of the velocity in 2D

The interpolation of the velocity is performed in three steps.

In the first step the Cartesian velocity is computed in the cell centre. First the fluxes V^1 and V^2 are averaged according to

$$V^1_{(0,0)} = (V^1_{(1,0)} + V^1_{(-1,0)})/2 \,, \tag{12.8}$$
$$V^2_{(0,0)} = (V^2_{(0,1)} + V^2_{(0,-1)})/2 \,, \tag{12.9}$$

see Figure 12.3 for the notations. Next the fluxes are transformed to Cartesian velocity

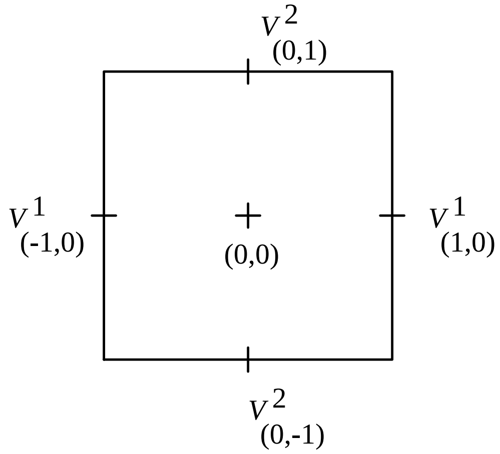


Figure 12.3: Cell with fluxes and centroid

components using:

$$\begin{aligned} V^\alpha &= \sqrt{g} U^\alpha \,, \\ \boldsymbol{u} &= U^\alpha \boldsymbol{a}_\alpha \,, \end{aligned}$$

hence

$$\boldsymbol{u} = (V^1 \boldsymbol{a}_{(1)} + V^2 \boldsymbol{a}_{(2)})/\sqrt{g} \tag{12.10}$$

In the second step each of the components is interpolated to the vertices by exactly the procedure described for scalars in 12.1.

Finally at the boundary essential boundary conditions, if present, are substituted in order to avoid unnecessary interpolation erors.

12.3 Computation of the stream function

A special scalar that is computed, is the stream function ψ. Since in fact ISNaS incompressible computes the fluxes, the steam function computation is straightforward.

At present we assume $\psi = 0$ at the vertex point (1,1). The values in the other vertices are computed by summation:

For $j := 1(1)nj$ do
    $\psi_{1,j+1} := \psi_{1,j} + V^1_{1,j}$

        $i := 1(1)ni$ do
            $\psi_{i+1,j+1} := \psi_{i,j+1} + V^2_{i,j+1}$

References

1. C.C. Ashcraft and R.G. Grimes. On vectorizing incomplete factorization and SSOR preconditioners. *SIAM J. Sci. Stat. Comput.*, 9:122–151, 1988.

2. W.L. Chen, F.S. Lien, and M.A. Leschziner. Computational modelling of turbulent flow in turbomachine passage with low-Re two-equation models. In S. Wagner, E.H. Hirschel, J. Périaux, and R. Piva, editors, *Computational Fluid Dynamics '94*, pages 517–524, John Wiley and Sons, Chicester, 1994.

3. Y.-S. Chen and S.-W. Kim. Computations of turbulent flows using an extended k-ε turbulence closure model. Report NASA CR-179204, NASA-Marshall Space Flight Center, Alabama, USA, 1987.

4. I.A. Demirdžić. *A finite volume method for computation of fluid flow in complex geometries.* PhD thesis, University of London, 1982.

5. S.C. Eisenstat. Efficient implementation of a class of preconditioned conjugate gradient methods. *SIAM J. Sci. Stat. Comput.*, 2:1–4, 1981.

6. P.H. Gaskell and K.C. Lau. Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm. *Int. J. Numer. Methods in Fluids*, 8:617–641, 1988.

7. T.B. Gatski and C.G. Speziale. On explicit algebraic stress models for complex turbulent flows. *J. Fluid Mech.*, 254:59–78, 1993.

8. V. Haroutunian. Simulation of vortex shedding past a square prism using three two-equation turbulence models. In *Sixth Int. Symp. on CFD*, volume 1, pages 408–414, Lake Tahoe, Nevada, 1995. A collection of technical papers.

9. C. Hirsch. *Numerical Computation of Internal and External Flows. Vol.1: Fundamentals of Numerical Discretization.* John Wiley, Chichester, 1988.

10. M. Kato and B.E. Launder. The modelling of turbulent flow around stationary and vibrating square cylinders. In *Proc. Ninth Symposium on Turbulent Shear Flows*, volume 9, page 10.4.1, Kyoto, Japan, 1993.

11. C.D. Kay. *Schaum's outline of theory and problems of tensor calculus.* McGraw-Hill, New York, 1988.

12. P. K. Khosla and S. G. Rubin. A diagonally dominant second-order accurate implicit scheme. *Comput. Fluids*, 2:207–209, 1974.

13. B. Koren. A robust upwind discretization method for advection, diffusion and source terms. In C.B. Vreugdenhil and B. Koren, editors, *Numerical methods for advection-diffusion problems*, pages 117–137, Vieweg. Braunschweig, Wiesbaden, 1993. Notes on Numerical Fluid Mechanics 45.

14. C.K.G. Lam and K. Bremhorst. A modified form of the k-ε model for predicting wall turbulence. *ASME J. Fluids Engng.*, 103:456–460, 1981.

15. B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Comp. Methods Appl. Mech. Eng.*, 3:269–289, 1974.

16. F.S. Lien and M.A. Leschziner. Upstream monotonic interpolation for scalar transport with application to complex turbulent flows. *Int. J. Num. Meth. Fluids*, 19:527–548, 1994.

17. J.A. Meijerink and H.A. Van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.

18. H.K. Myong and N. Kasagi. unknown. *J. Fluids Engng.*, 112:521, 1990.

19. S. Nisizima and A. Yoshizawa. Turbulent channel and Couette flows using an anisotropic k-ε model. *AIAA J.*, 25:414–420, 1987.

20. C.C. Paige and M.A. Saunders. LSQR: an algorithm for sparse linear equations and sparse least square problem. *ACM Trans. Math. Softw.*, 8:44–71, 1982.

21. C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.*, 21:1525–1532, 1983.

22. R. Rubenstein and J.M. Barton. Non-linear Reynolds stress models and the renormalisation group. *Phys. Fluids A*, 2:1472, 1990.

23. Y. Saad and M.H. Schultz. GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.

24. H. Schlichting. *Boundary layer theory.* McGraw Hill, New York, 1969.

25. A. Segal. The treatment of slip boundary conditions for the incompressible navier-stokes equations in general co-ordinates. Report 91-22, Faculty of Mathematics and Informatics, Delft University of Technology, Delft, 1991.

26. A. Segal and K. Kassels. Some 2d test examples for the isnas incompressible code. Report 91-44, Faculty of Mathematics and Informatics, Delft University of Technology, Delft, 1991.

27. Guus Segal and Kees Kassels. Improvements of the discretization of the incompressible Navier-Stokes equations in general coordinates. Report 96-81, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, 1996.

28. P. Sonneveld. CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 10:36–52, 1989.

29. D.B. Spalding. A novel finite difference formulation for differential expressions involving both first and second derivatives. *Int. J. Numer. Methods in Engineering*, 4:551–559, 1972.

30. C.G. Speziale. On nonlinear k-l and k-ε models of turbulence. *J. Fluid Mech.*, 178:459–475, 1987.

31. P.K. Sweby. High resolution schemes using flux-limiters for hyperbolic conservation laws. *SIAM J. Num. Anal.*, 21:995–1011, 1984.

32. H. Tennekes and J.L. Lumley. *A First Course in Turbulence.* MIT Press, Cambridge, Massachussets, 1982.

33. P. van Beek, R.R.P. van Nooyen, and P. Wesseling. Accurate discretization on non-uniform curvilinear staggered grids. *J. Comp. Phys.*, 117:364–367, 1995.

34. H.A. Van der Vorst. Iterative solution method for certain sparse linear systems with a non-symmetric matrix arising from PDE-problems. *J. Comput. Phys.*, 44:1–19, 1981.

35. H.A. van der Vorst and C. Vuik. GMRESR: a family of nested GMRES methods. *Num. Lin. Alg. Appl.*, 1:369–386, 1994.

36. J.J.I.M. Van Kan, C.W. Oosterlee, A. Segal, and P. Wesseling. Discretization of the incompressible Navier-Stokes equations in general coordinates using contravariant velocity components. Report 91-09, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, 1991.

37. B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32:101–136, 1979.

38. C. Vuik. Termination criteria for GMRES-like methods to solve the discretizet incompressible navier-stokes equations. Report 92-50, Faculty of Technical Mathematics and Informatics, Delft University of Technology, Delft, 1992.

39. C. Vuik. Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. *Int. J. Num. Meth. Fluids*, 16:507–523, 1993.

40. D.C. Wilcox. Reassessment of the scale determining equation for advanced turbulence models. *AIAA J.*, 26:1299–1310, 1988.

41. D.C. Wilcox. A half century historical review of the k-ω model. AIAA Paper 91-0615, 1991.

42. V. Yakhot, S.A. Orszag, S. Thangam, T.B. Gatski, and C.G. Speziale. Development of turbulence models for shear flows by a double expansion technique. *Phys. Fluids A*, 4:1510–1520, 1992.

43. M. Zijlema. On the construction of a third-order accurate TVD scheme with application to turbulent flows in general domains. *Int. J. Numer. Meth. Fluids*, 1995. To appear.

44. M. Zijlema. *Computational modeling of turbulent flows in general domains.* PhD thesis, Delft University of Technology, The Netherlands, April 1996.

Appendices

In these appendices we prove equations (8.6), (8.7), (8.22) and (8.23).

A Proof of (8.6) and (8.7)

We have to prove (see Figure A.1):

$$U^n = sign(\boldsymbol{a}^{(n)} \cdot \boldsymbol{n})\sqrt{g^{nn}}\, \boldsymbol{u} \cdot \boldsymbol{n} \tag{A.1}$$

and

$$\begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{\tau}_1 \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_2 \end{bmatrix} - U^n \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} \right\} \tag{A.2}$$

where

$$\alpha_{i1} = \frac{\begin{vmatrix} \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_1)} & g_{t_1t_2} \\ \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_2)} & g_{t_2t_2} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} \tag{A.3}$$

and

$$\alpha_{i2} = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_1)} \\ g_{t_2t_1} & \boldsymbol{\tau}_i \cdot \boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} \tag{A.4}$$

First (A.1):

$$\boldsymbol{u} \cdot \boldsymbol{n} = \boldsymbol{u} \cdot \frac{\boldsymbol{a}^{(n)}}{\|\boldsymbol{a}^{(n)}\|} = \frac{U^n}{\sqrt{g^{nn}}}$$

so

$$U^n = \sqrt{g^{nn}}\, \boldsymbol{u} \cdot \boldsymbol{n} \; .$$

This formula is true if $\boldsymbol{a}^{(n)}$ and $\boldsymbol{n}$ have the same direction else we have to use:

$$U^n = -\sqrt{g^{nn}}\, \boldsymbol{u} \cdot \boldsymbol{n} \; .$$

$\square$

Formula (A.2):
The tangential vector $\boldsymbol{\tau}_i$, given by the user can be decomposed in the following way:

$$\boldsymbol{\tau}_i = \alpha_{i1}\boldsymbol{a}_{(t_1)} + \alpha_{i2}\boldsymbol{a}_{(t_2)} \; , \tag{A.5}$$

see Figure A.2.

The calculation of α_{ij}:
From Figure A.2 it is clear that

$$\begin{aligned} \boldsymbol{a}_{(t_1)} \cdot \alpha_{i2}\boldsymbol{a}_{(t_2)} &= \boldsymbol{a}_{(t_1)} \cdot (\boldsymbol{\tau}_i - \alpha_{i1}\boldsymbol{a}_{(t_1)}) \\ \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_1)}\alpha_{i1} + \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_2)}\alpha_{i2} &= \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{\tau}_i \end{aligned} \tag{A.6}$$

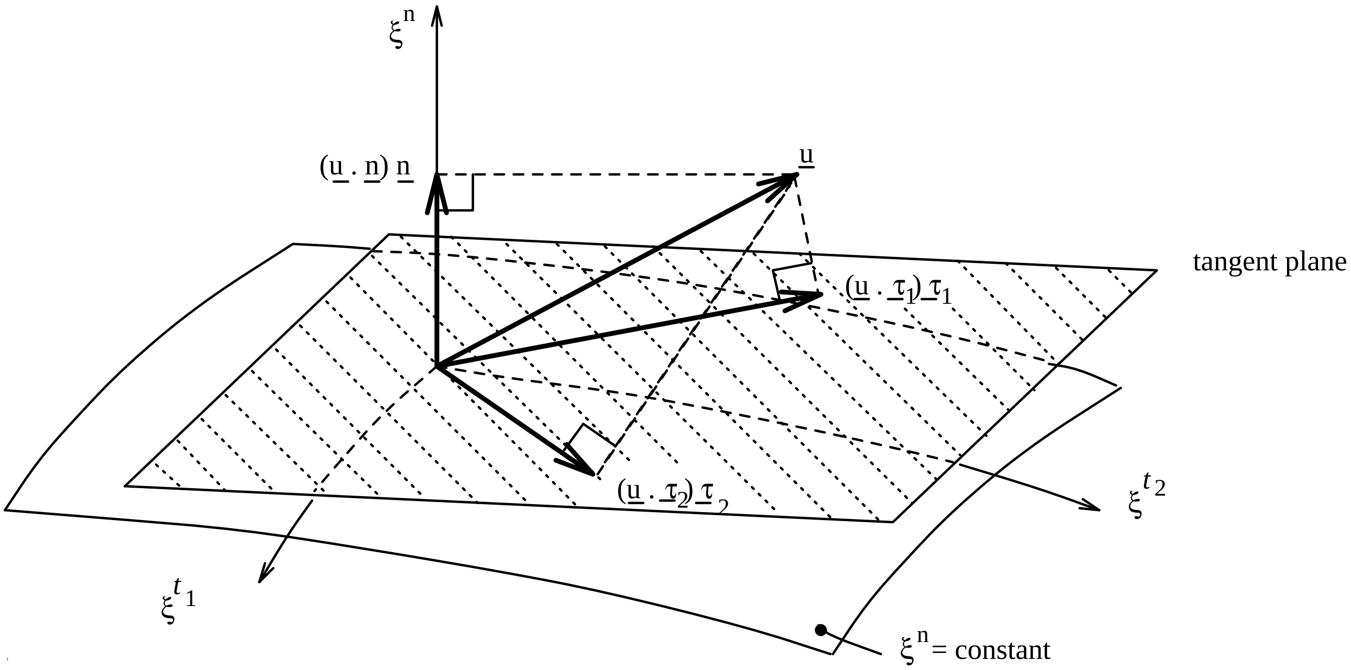


Figure A.1: Normal and tangential velocity components are given by the user ($\|\underline{n}\| = \|\underline{\tau}_1\| = \|\underline{\tau}_2\| = 1$)

and

$$\begin{aligned} \boldsymbol{a}_{(t_2)} \cdot \alpha_{i1}\boldsymbol{a}_{(t_1)} = \boldsymbol{a}_{(t_2)} \cdot (\boldsymbol{\tau}_i - \alpha_{i2}\boldsymbol{a}_{(t_2)}) \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_1)}\alpha_{i1} + \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_2)}\alpha_{i2} = \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{\tau}_i \ . \end{aligned} \tag{A.7}$$

Frome (A.6), (A.7) and Cramer's rule we obtain.

$$\alpha_{i1} = \frac{\begin{vmatrix} \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{\tau}_i & \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_2)} \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{\tau}_i & \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_2)} \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_2)} \end{vmatrix}} = \frac{\begin{vmatrix} \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{\tau}_i & g_{t_1t_2} \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{\tau}_i & g_{t_2t_2} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} \tag{A.8}$$

and

$$\alpha_{i2} = \frac{\begin{vmatrix} \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{\tau}_i \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{\tau}_i \end{vmatrix}}{\begin{vmatrix} \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{a}_{(t_2)} \\ \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_1)} & \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{a}_{(t_2)} \end{vmatrix}} = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{a}_{(t_1)} \cdot \boldsymbol{\tau}_i \\ g_{t_2t_1} & \boldsymbol{a}_{(t_2)} \cdot \boldsymbol{\tau}_i \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} \tag{A.9}$$

Back to formula (A.5):

$$\begin{aligned} \boldsymbol{u} \cdot \boldsymbol{\tau}_i &= \boldsymbol{u} \cdot (\alpha_{i1}\boldsymbol{a}_{(t_1)} + \alpha_{i2}\boldsymbol{a}_{(t_2)}) \\ &= \alpha_{i1}\boldsymbol{u} \cdot g_{pt_1}\boldsymbol{a}^{(p)} + \alpha_{i2}\boldsymbol{u} \cdot g_{pt_2}\boldsymbol{a}^{(p)} \\ &= (\alpha_{i1}g_{pt_1} + \alpha_{i2}g_{pt_2})U^p \ , \end{aligned} \tag{A.10}$$

so:

$$\boldsymbol{u} \cdot \boldsymbol{\tau}_1 = (\alpha_{11}g_{t_1t_1} + \alpha_{12}g_{t_1t_2})U^{t_1} + (\alpha_{11}g_{t_2t_1} + \alpha_{12}g_{t_2t_1})U^{t_2} +$$

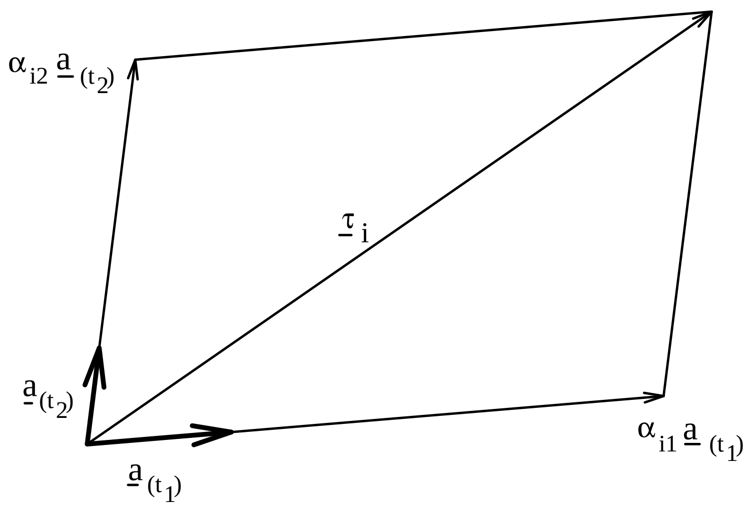


Figure A.2: Decomposition of $\underline{\tau}_i$.

$$
\begin{aligned}
& (\alpha_{11} g_{nt_1} + \alpha_{12} g_{nt_2}) U^n \;, && \text{(A.11)} \\
\boldsymbol{u} \cdot \boldsymbol{\tau}_2 \;=\; & (\alpha_{21} g_{t_1 t_1} + \alpha_{22} g_{t_1 t_2}) U^{t_1} + (\alpha_{21} g_{t_2 t_1} + \alpha_{22} g_{t_2 t_2}) U^{t_2} + \\
& (\alpha_{21} g_{nt_1} + \alpha_{22} g_{nt_2}) U^n \;. && \text{(A.12)}
\end{aligned}
$$

Formula (A.11) and (A.12) in matrix notation gives:

$$
\begin{bmatrix} \alpha_{11} g_{t_1 t_1} + \alpha_{12} g_{t_1 t_2} & \alpha_{11} g_{t_2 t_1} + \alpha_{12} g_{t_2 t_2} \\ \alpha_{21} g_{t_1 t_1} + \alpha_{22} g_{t_1 t_2} & \alpha_{21} g_{t_2 t_1} + \alpha_{22} g_{t_2 t_2} \end{bmatrix} \begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{\tau}_1 - (\alpha_{11} g_{nt_1} + \alpha_{12} g_{nt_2}) U^n \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_2 - (\alpha_{21} g_{nt_1} + \alpha_{22} g_{nt_2}) U^n \end{bmatrix}
$$

or:

$$
\begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} g_{t_1 t_1} & g_{t_1 t_2} \\ g_{t_2 t_1} & g_{t_2 t_2} \end{bmatrix} \begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{\tau}_1 \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_2 \end{bmatrix} - U^n \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix}
$$

Hence:

$$\begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{\tau}_1 \\ \boldsymbol{u} \cdot \boldsymbol{\tau}_2 \end{bmatrix} - U^n \begin{bmatrix} g_{nt_1} \\ g_{nt_1} \end{bmatrix} \right\}$$ [1]

where α_{ij} is given by (A.8) and (A.9). □

[1] This formula can be modified in the following way:

$$\begin{bmatrix} U^{t_1} \\ U^{t_2} \end{bmatrix} = \begin{bmatrix} g_{t_2t_2} & g_{t_1t_2} \\ -g_{t_2t_1} & g_{t_1t_1} \end{bmatrix} \left\{ \begin{bmatrix} \beta_{11} & \beta_{12} \\ \beta_{21} & \beta_{22} \end{bmatrix}^{-1} \begin{bmatrix} \underline{u} \cdot \underline{\tau_1} \\ \underline{u} \cdot \underline{\tau_2} \end{bmatrix} - \frac{U^n}{g_{t_1t_1}g_{t_2t_2} - g_{t_1t_2}g_{t_2t_1}} \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} \right\}$$

where

$$\beta_{i1} = \begin{vmatrix} \underline{\tau_i} \cdot \underline{a}_{(t_1)} & g_{t_1t_2} \\ \underline{\tau_i} \cdot \underline{a}_{(t_2)} & g_{t_2t_2} \end{vmatrix}$$

and

$$\beta_{12} = \begin{vmatrix} g_{t_1t_1} & \underline{\tau_i} \cdot \underline{a}_{(t_1)} \\ g_{t_2t_1} & \underline{\tau_i} \cdot \underline{a}_{(t_2)} \end{vmatrix}$$

B Proof of (8.22) and (8.23)

We have to prove (see Figure B.1):

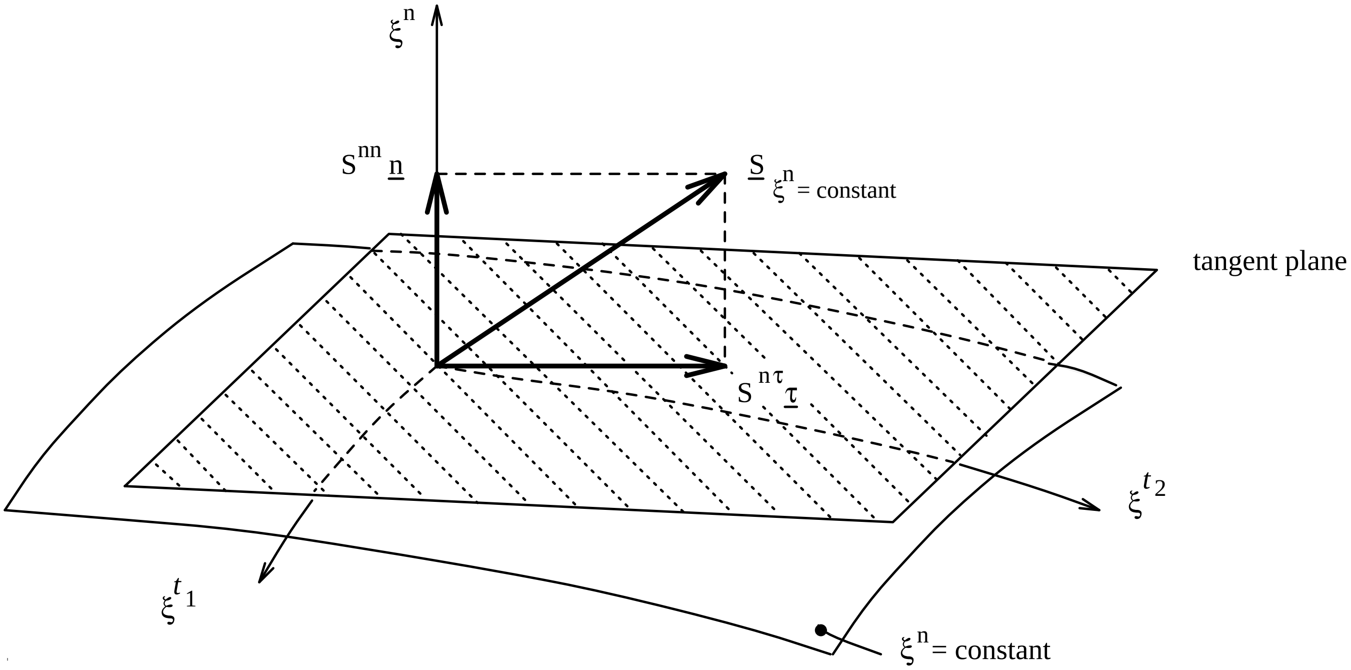


Figure B.1: The normal and tangential stress in the physical domain at the boundary $\xi^n = constant$.

$$\sigma^{nn} = g^{nn} S^{nn} \tag{B.1}$$

and

$$\begin{bmatrix} \sigma_{nt_1} \\ \sigma_{nt_2} \end{bmatrix} = sign(\boldsymbol{a}^{(n)} \cdot \boldsymbol{n}) \cdot \sqrt{g^{nn}} S^{n\tau} \begin{bmatrix} \alpha_1 \\ \alpha_2 \end{bmatrix} - \sigma^{nn} \begin{bmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{bmatrix}^{-1} \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} {}^2 \tag{B.2}$$

where

$$\alpha_1 = \frac{\begin{vmatrix} \boldsymbol{\tau} \cdot \boldsymbol{a}_{(t_1)} & g_{t_1t_2} \\ \boldsymbol{\tau} \cdot \boldsymbol{a}_{(t_2)} & g_{t_2t_3} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} , \tag{B.3}$$

---

[2] This formula can also be modified in the following way:

$$\begin{bmatrix} \sigma^{nt_1} \\ \sigma^{nt_2} \end{bmatrix} = \frac{1}{g_{t_1t_1} \cdot g_{t_2t_2} - g_{t_1t_2} \cdot g_{t_2t_1}} \left\{ sign(a^{(n)} \cdot \underline{n}) \sqrt{g^{nn}} S^{n\tau} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} - \sigma^{nn} \begin{bmatrix} g_{t_2t_2} & -g_{t_1t_2} \\ -g_{t_2t_1} & g_{t_1t_1} \end{bmatrix} \begin{bmatrix} g_{nt_1} \\ g_{nt_2} \end{bmatrix} \right\}$$

where

$$\beta_1 = \begin{vmatrix} \underline{\tau} \cdot \underline{a}_{(t_1)} & g_{t_1t_2} \\ \underline{\tau} \cdot \underline{a}_{(t_2)} & g_{t_2t_2} \end{vmatrix}$$

and

$$\beta_2 = \begin{vmatrix} g_{t_1t_1} & \underline{\tau} \cdot \underline{a}_{(t_1)} \\ g_{t_2t_1} & \underline{\tau} \cdot \underline{a}_{(t_2)} \end{vmatrix} .$$

$$\alpha_2 = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} \\ g_{t_2t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} \\ g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} . \tag{B.4}$$

We start with formula (3.17) from [11]:

$$\boldsymbol{S}_{\xi^n=constant} = \bar{\sigma}^{ij}(\boldsymbol{n}\cdot\boldsymbol{e}_i)\boldsymbol{e}_j \tag{B.5}$$

where $\bar{\sigma}^{ij}$ is stress tensor in the physical domain.
From $\underline{S}_{\xi^n=constant} = S^{nn}\boldsymbol{n} + S^{n\tau}\boldsymbol{\tau}$ and (B.5) we get:

$$\bar{\sigma}^{ij}(\boldsymbol{n}\cdot\boldsymbol{e}_i)\boldsymbol{e}_j = S^{nn}\boldsymbol{n} + S^{n\tau}\boldsymbol{\tau} \tag{B.6}$$

so

$$(\bar{\sigma}^{ij}(\boldsymbol{n}\cdot\boldsymbol{e}_j)\boldsymbol{e}_j)\cdot g^{(n)} = (S^{nn}\boldsymbol{n} + S^{n\tau}\boldsymbol{\tau})\cdot\boldsymbol{a}^{(n)} = S^{nn}\boldsymbol{n}\cdot\boldsymbol{a}^{(n)} . \tag{B.7}$$

It should be noticed that S^{nn} and $S^{n\tau}$ are not tensors.
The normal vector $\boldsymbol{n}$ pointing in the outside direction of the domain is equal to: $\frac{1}{\|\boldsymbol{a}^{(n)}\|}\boldsymbol{a}^{(n)}$ if $\boldsymbol{a}^{(n)}$ is pointing in the outside direction and $-\frac{1}{\|a^{(n)}\|}\boldsymbol{a}^{(n)}$ otherwise. Formally we can write:

$$\boldsymbol{n} = \frac{sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})}{\|\boldsymbol{a}^{(n)}\|}\boldsymbol{a}^{(n)} . \tag{B.8}$$

From (B.7) and (B.8) we get:

$$\begin{aligned} &(\bar{\sigma}^{ij}(\boldsymbol{a}^{(n)}\cdot\boldsymbol{e}_i)\boldsymbol{e}_j)\cdot\boldsymbol{a}^{(n)} = S^{nn}\boldsymbol{a}^{(n)}\cdot\boldsymbol{a}^{(n)} \\ &\bar{\sigma}^{ij}(\boldsymbol{a}^{(n)})_i(\boldsymbol{a}^{(n)})_j = S^{nn}g^{nn} \\ &\bar{\sigma}^{ij}\frac{\partial\xi^n}{\partial x^i}\frac{\partial\xi^n}{\partial x^j} = g^{nn}S^{nn} \end{aligned}$$

so

$$\sigma^{nn} = g^{nn}S^{nn} .$$

$\square$

Formula (B.2):
The tangential vector $\boldsymbol{\tau}$ is just as in the previous proof equal to:

$$\boldsymbol{\tau} = \alpha_1\boldsymbol{a}_{(t_1)} + \alpha_2\boldsymbol{a}_{(t_2)} \tag{B.9}$$

where

$$\alpha_1 = \frac{\begin{vmatrix} \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} & g_{t_1t_2} \\ \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} & g_{t_2t_1} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}}$$

and

$$\alpha_2 = \frac{\begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} \\ g_{t_2t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} \end{vmatrix}}{\begin{vmatrix} g_{t_1t_1} & g_{t_1t_2} \\ g_{t_2t_1} & g_{t_2t_2} \end{vmatrix}} .$$

From (B.7), (B.8) and (B.9) it follows that:

$$
\begin{aligned}
&(\bar{\sigma}^{ij}(\frac{sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})}{\|\boldsymbol{a}^{(n)}\|}\boldsymbol{a}^{(n)}\cdot\boldsymbol{e}_i)\boldsymbol{e}_j)\cdot\boldsymbol{a}_{t_l)} = S^{n\tau}(\alpha_1\boldsymbol{a}_{(t_1)}+\alpha_2\boldsymbol{a}_{(t_2)})\cdot\boldsymbol{a}_{(t_l)}\\
&sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\bar{\sigma}^{ij}(\boldsymbol{a}^{(n)})_i(\boldsymbol{a}_{(t_l)})_j = \|\boldsymbol{a}^{(n)}\|S^{n\tau}(\alpha_1\boldsymbol{a}_{(t_1)}\cdot\boldsymbol{a}_{(t_l)}+\alpha_2\boldsymbol{a}_{(t_2)}\cdot\boldsymbol{a}_{t_l)})\\
&sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\bar{\sigma}^{ij}(\boldsymbol{a}^{(n)})_i g_{pt_l}(\boldsymbol{a}^{(p)})_j = \sqrt{g^{nn}}S^{n\tau}(\alpha_1 g_{t_1t_l}+\alpha_2 g_{t_2t_l})\\
&sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})g_{pt_l}\bar{\sigma}^{ij}\frac{\partial\xi^n}{\partial x^i}\frac{\partial\xi^p}{\partial x^j} = \sqrt{g^{nn}}S^{n\tau}(\alpha_1 g_{t_1t_1}+\alpha_2 g_{t_2t_l})
\end{aligned}
$$

so:

$$
sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})g_{pt_l}\sigma^{np} = \sqrt{g^{nn}}S^{n\tau}(\alpha_1 g_{t_1t_l}+\alpha_2 g_{t_2t_l})\ , \tag{B.10}
$$

where $\sigma^{\alpha\beta}$ is the stress tensor in the computational domain. For $l = 1, 2$ we get:

$$
sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\{g_{t_1t_1}\sigma^{nt_1}+g_{t_2t_1}\sigma^{nt_2}+g_{nt_1}\sigma^{nn}\} = \sqrt{g^{nn}}S^{n\tau}(\alpha_1 g_{t_1t_1}+\alpha_2 g_{t_2t_1})
$$

and

$$
sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\{g_{t_1t_2}\sigma^{nt_1}+g_{t_2t_2}\sigma^{nt_2}+g_{nt_2}\sigma^{nn}\} = \sqrt{g^{nn}}S^{n\tau}(\alpha_1 g_{t_1t_2}+\alpha_2 gt_2t_2)
$$

or:

$$
\begin{bmatrix} g_{t_1t_1} & g_{t_2t_1}\\ g_{t_1t_2} & g_{t_2t_2}\end{bmatrix}\begin{bmatrix}\sigma^{nt_1}\\ \sigma^{nt_2}\end{bmatrix}+\sigma^{nn}\begin{bmatrix} g_{nt_1}\\ g_{nt_2}\end{bmatrix} = sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\sqrt{g^{nn}}S^{n\tau}\begin{bmatrix} g_{t_1t_1} & g_{t_2t_1}\\ g_{t_1t_2} & g_{t_2t_2}\end{bmatrix}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}
$$

so:

$$
\begin{bmatrix}\sigma^{nt_1}\\ \sigma^{nt_2}\end{bmatrix} = sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\sqrt{g^{nn}}S^{n\tau}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}-\sigma^{nn}\begin{bmatrix} g_{t_1t_1} & g_{t_2t_1}\\ g_{t_1t_2} & g_{t_2t_2}\end{bmatrix}^{-1}\begin{bmatrix} g_{nt_1}\\ g_{nt_2}\end{bmatrix}
$$

hence:

$$
\begin{bmatrix}\sigma^{nt_1}\\ \sigma^{nt_2}\end{bmatrix} = sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\sqrt{g^{nn}}S^{n\tau}\begin{bmatrix}\alpha_1\\ \alpha_2\end{bmatrix}-\sigma^{nn}\begin{bmatrix} g_{t_1t_1} & g_{t_1t_2}\\ g_{t_2t_1} & g_{t_2t_2}\end{bmatrix}^{-1}\begin{bmatrix} g_{nt_1}\\ g_{nt_2}\end{bmatrix}\ . \tag{B.11}
$$

By using Cramers rule in formula (B.11) we get:

$$
\begin{bmatrix}\sigma^{nt_1}\\ \sigma^{nt_2}\end{bmatrix} = \frac{1}{g_{t_1t_1}g_{t_2t_1}-g_{t_1t_2}g_{t_2t_1}}\left\{sign(\boldsymbol{a}^{(n)}\cdot\boldsymbol{n})\sqrt{g^{nn}}S^{n\tau}\begin{bmatrix}\beta_1\\ \beta_2\end{bmatrix}-\sigma^{nn}\begin{bmatrix} g_{t_2t_2} & -g_{t_1t_2}\\ -g_{t_2t_1} & g_{t_1t_1}\end{bmatrix}\begin{bmatrix} g_{nt_1}\\ g_{nt_2}\end{bmatrix}\right\} \tag{B.12}
$$

where

$$
\beta_1 = \begin{vmatrix}\boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)} & g_{t_1t_2}\\ \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)} & g_{t_2t_2}\end{vmatrix} \tag{B.13}
$$

and

$$
\beta_2 = \begin{vmatrix} g_{t_1t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_1)}\\ g_{t_2t_1} & \boldsymbol{\tau}\cdot\boldsymbol{a}_{(t_2)}\end{vmatrix}\ . \tag{B.14}
$$

□