Journal of Engineering Mathematics **34:** 21–44, 1998.
© 1998 *Kluwer Academic Publishers. Printed in the Netherlands.*

Computing flows on general two-dimensional nonsmooth staggered grids

P. WESSELING, A. SEGAL, C.G.M. KASSELS and H. BIJL
J.M. Burgers Centre, Delft University of Technology, Faculty of Information Technology and Systems, Mekelweg 4, 2628 CD Delft, The Netherlands

Received 18 June 1997; accepted in revised form 12 January 1998

Abstract. The classical staggered scheme for the incompressible Navier-Stokes equations is generalized from Cartesian grids to general boundary-fitted structured grids. The resulting discretization is coordinate-invariant. The unknowns are the pressure and the contravariant volume flux components. The grid can be strongly nonuniform and nonorthogonal. The smoothness properties of the coordinate mapping are carefully taken into account. As a result, the accuracy on rough grids is found to be at least as good as for typical finite element and nonstaggered finite volume schemes. Extension to compressible flows results in a scheme with Mach-uniform accuracy and efficiency for Mach numbers ranging from $M = 0$ to $M > 1$. Accurate discretization of two-equation turbulence models is also possible.

Keywords: Navier-Stokes; finite volume discretization; staggered grids; boundary-fitted coordinates.

1. Introduction

We think nobody will dispute that in Cartesian coordinates, computation of incompressible flows is best performed on the staggered grid proposed by Harlow and Welch [1]. In combination with the pressure correction method an efficient and accurate method to compute instationary flows is obtained. The method is also straightforward, provided spatial discretization precedes introduction of pressure correction, so that no artificial pressure boundary condition is required.

However, there is no such consensus when the domain is not rectangular. We cannot be complete in listing all possible approaches, and even less so in referring to the abundant literature. A first distinction may be made between structured and unstructured grids. In structured grids the number of cells that share an interior vertex is fixed. For unstructured grids there is no such restriction. Unstructured grids, which include finite element methods, will not be considered here. The general approach to handle complicated domains with structured grids is to use an unstructured decomposition of the domain into subdomains of simpler shape, with a structured grid inside each subdomain. We will consider only the case of a single subdomain, with a structured grid constructed by a boundary-fitted coordinate mapping.

In general coordinates, accurate discretization of differential operators on staggered grids is generally considered to be much more complicated (if not impossible) than on nonstaggered grids. As a consequence, nonstaggered (or collocated) discretization is much more widespread for the Navier-Stokes equations than staggered discretization, and prevails in commercial codes. An incomplete list of publications taking this route is [2], [3] (one-sided discretization of div $\boldsymbol{u}$ and grad p); [4]–[14] (using the pressure-weighted interpolation method of Rhie and Chow [12]); [15]–[24] (employing artificial compressibility). But for incompressible flows, a price has to be paid for the ease of handling general coordinates that nonstaggered discretization brings. In order to avoid spurious oscillations, regularizing terms must be added to the continuity equation. These terms may falsify transient behaviour, or make instationary computations more costly and complicated, or make extension to weakly compressible flow difficult, or are not suitable in the presence of strong body forces ([7]). Furthermore, good coupling conditions at subdomain boundaries in domain decomposition methods are harder to obtain. For these reasons, a relatively minor number of groups have sought to generalize the staggered scheme from Cartesian to generalized coordinates. Some publications in this direction are [25]–[34], and by our group (begging forgiveness for being complete this time) [35]–[50].

We think that on the question of whether nonstaggered or staggered grids are preferable the last word has not yet been said. Our purpose here is to show that the staggered scheme can be generalized from Cartesian to general coordinates while maintaining accuracy even on very nonuniform grids, provided the smoothness properties of the boundary-fitted coordinate mapping are carefully taken into account. If this is not done or if too much smoothness is implicitly assumed, the accuracy can become bad even on mildly nonsmooth grids. This experience has led many to think that on curvilinear grids, staggered discretization is inherently less accurate than nonstaggered discretization, but we intend to show that this is not so.

On nonstaggered grids it is convenient to discretize in physical space, and no reference is made to the coordinate mapping, so that its smoothness properties do not come into play, and no serious degradation of accuracy is observed as the grid becomes less smooth. Staggered discretization may also be carried out in physical space; this is done in [31]. We expect this method to behave satisfactory on nonsmooth grids, althoug this is not shown in [31]. But on staggered grids, discretization in physical space puts a heavy demand on geometric insight and pictorial representation, which is why we have developed an algebraic formulation. Furthermore, we think it desirable to bring out explicitly the role of the smoothness properties of the coordinate mapping. We will use tensor notation and derive a coordinate-invariant discretization in general coordinates. This approach can be extended to governing equations (in other fields) that contain tensors of rank higher than two. Discretization of such laws in physical space on staggered grids would seem hard to do.

If the mesh-size jumps, the local discretization error is of first order for vertex-centered and of zeroth order (which makes the scheme inconsistent in the maximum norm) for cell-centered schemes for the convection-diffusion equation ([51]). This has made some believe that grids need to be smooth for accurate results. But this is not so. In [51]–[54] it is shown that the global discretization error is second order on strongly nonuniform grids. These results for the convection-diffusion equation may be expected to carry over to the Navier-Stokes equations. This is fortunate, because it allows us to switch abruptly from a fine mesh in thin boundary or shear layers to a coarse mesh outside. For such grids, in [54]–[56] it is shown that the accuracy is uniform in the Reynolds number, for the convection-diffusion equation. In order to allow abrupt changes in mesh-size, we will assume the coordinate mapping to be merely piecewise differentiable.

We will start by discussing as far as necessary geometric aspects of coordinate transformations. A brief outline of nonstaggered discretization in physical space will be given. Next, a staggered discretization will be presented for the incompressible Navier-Stokes equations, which is accurate on general nonuniform grids. Then we will generalize this to the compressible case, obtaining a method with accuracy and efficiency uniform in the Mach number. Finally reference is made to numerical experiments including two-equation turbulence modeling.

2. Geometric quantities and their smoothness

For brevity we will only discuss the two-dimensional case, but the following considerations apply also in three dimensions. For deriving accurate discretizations with economic formulae it is useful to first establish some elementary geometrical properties related to coordinate transformations and structured grids, paying special attention to smoothness properties.

Let the physical domain Ω be topologically equivalent to the unit square G. In $\bar{G}$ we have Cartesian coordinates $\boldsymbol{\xi} = (\xi^1, \xi^2)$ and a uniform grid G_h, consisting of cell vertices located at $\boldsymbol{\xi}_j,\ j = (j_1, j_2)$:

$$G_h = \{\boldsymbol{\xi}_j : \xi_j^\alpha = j_\alpha \Delta\xi^\alpha,\ j_\alpha = 0, 1, \dots 1/\Delta\xi^\alpha,\ \alpha = 1, 2\}, \tag{2.1}$$

where $1/\Delta\xi^\alpha \in \mathbb{N}$. Greek indices will be used exclusively to refer to coordinate directions, and vice-versa. Unless stated otherwise, summation is implied exclusively over pairs consisting of one Greek superscript and one Greek subscript in terms and products. This summation convention does not apply to (2.1).

It is assumed that a boundary-fitted coordinate mapping is generated numerically, giving a mapping of G_h into $\bar{\Omega}$:

$$\boldsymbol{x}_j = \boldsymbol{x}_j(\boldsymbol{\xi}_j), \quad \boldsymbol{x} \in \bar{\Omega}, \quad \boldsymbol{\xi}_j \in G_h. \tag{2.2}$$

In order to provide a solid foundation for deriving accurate discretizations, we have to be precise about how the mapping (2.2) is extended to all of $\bar{G}$. To allow rough grids for reasons given in the preceding section, the mapping (2.2) is extended by bilinear interpolation. Let $\Omega_j \in \bar{\Omega}$ be a geometric figure to be further defined shortly with vertices

$$\boldsymbol{x}_{j-e_1-e_2},\ \ \boldsymbol{x}_{j+e_1-e_2},\ \ \boldsymbol{x}_{j-e_1+e_2},\ \ \boldsymbol{x}_{j+e_1+e_2},\ \ e_1 = (\tfrac{1}{2}, 0), \qquad e_2 = \left(0, \tfrac{1}{2}\right) \tag{2.3}$$

(since the vertices have integer indices, this implies that j_1 and j_2 are fractional). The image of Ω_j in $\bar{G}$ is a rectangle called G_j; G_j and Ω_j are called cells. Let $\boldsymbol{\xi}_0$ be some point in G_j. Bilinear interpolation in Ω_j gives the following relation between $\boldsymbol{x}$ and $\boldsymbol{\xi}$:

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{c}_\alpha(\xi^\alpha - \xi_0^\alpha) + \boldsymbol{c}_{12}(\xi^1 - \xi_0^1)(\xi^2 - \xi_0^2). \tag{2.4}$$

Bold Latin letters will always denote vectors in $\mathbb{R} \times \mathbb{R}$, whether they have subscripts or not. Since the mapping (2.4) is different in every cell, the definition (2.4) makes the mapping piecewise bilinear. The edges of G_j are straight and on every edge, ξ^1 or ξ^2 is constant. This makes the mapping (2.4) linear at the edges, so that the edges of Ω_j are also straight, as indicated in Figure 1. Each cell has its own mapping (2.4). The coefficients $\boldsymbol{x}_0, \boldsymbol{c}_\alpha$ and $\boldsymbol{c}_{12}$ follow from the requirement $\boldsymbol{x}(\boldsymbol{\xi}_m) = \boldsymbol{x}_m,\ m = 1, 2, 3, 4$ where the subscript m refers to the vertices enumerated in Figure 1, and are easily determined as follows. Choose $\boldsymbol{\xi}_0$ in the center of G_j, and define new variables s^α by

$$\xi^\alpha = \xi_0^\alpha + \tfrac{1}{2}\Delta\xi^\alpha s^\alpha \quad \text{(no summation)} \tag{2.5}$$

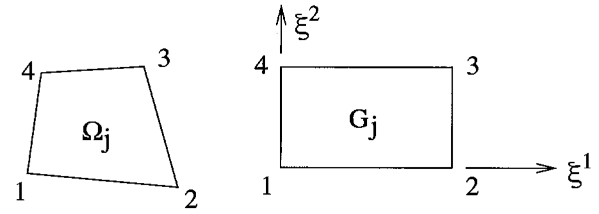

Figure 1. Grid cell in physical space (Ω_j) and in computational space (G_j).

so that (2.4) becomes

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{b}_\alpha s^\alpha + \boldsymbol{b}_{12} s^1 s^2 \tag{2.6}$$

with

$$\boldsymbol{b}_\alpha = \tfrac{1}{2}\Delta\xi^\alpha \boldsymbol{c}_\alpha, \qquad \boldsymbol{b}_{12} = \tfrac{1}{4}\Delta\xi^1\Delta\xi^2 \boldsymbol{c}_{12} \quad \text{(no summation)}. \tag{2.7}$$

By requiring $\boldsymbol{x}(\boldsymbol{\xi}_m) = \boldsymbol{x}_m,\ m = 1, 2, 3, 4$, we obtain the following system of equations:

$$Ay = r, \tag{2.8}$$

with $y = (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_{12}, \boldsymbol{x}_0)^T, \quad r = (\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4)^T$ and

$$A = \begin{pmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 \end{pmatrix}.$$

The columns of A are orthogonal. Premultiplication of (2.8) by A^T gives $y = \frac{1}{4}A^T r$ so that, defining $\boldsymbol{x}_{mn} \equiv \boldsymbol{x}_m + \boldsymbol{x}_n$

$$\begin{aligned} \boldsymbol{b}_1 &= \tfrac{1}{4}(\boldsymbol{x}_{34} - \boldsymbol{x}_{12}), & \boldsymbol{b}_2 &= \tfrac{1}{4}(\boldsymbol{x}_{23} - \boldsymbol{x}_{14}), \\ \boldsymbol{b}_{12} &= \tfrac{1}{4}(\boldsymbol{x}_{13} - \boldsymbol{x}_{24}), & \boldsymbol{x}_0 &= \tfrac{1}{4}(\boldsymbol{x}_{12} + \boldsymbol{x}_{34}). \end{aligned} \tag{2.9}$$

The coordinate mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ is assumed to be a bijection, so that we have for the Jacobian

$$J = \frac{\partial \boldsymbol{x}}{\partial \xi^1} \otimes \frac{\partial \boldsymbol{x}}{\partial \xi^2} \neq 0, \tag{2.10}$$

where the operator $\otimes$ is defined as

$$\boldsymbol{a} \otimes \boldsymbol{b} \equiv a^1 b^2 - a^2 b^1. \tag{2.11}$$

It is furthermore assumed that $J > 0$, i.e. the x^α and ξ^α coordinate systems are right-handed

By the definition of the outer product, in three dimensions the vector $\boldsymbol{a} \times \boldsymbol{b}$ has magnitude equal to twice the area of the triangle spanned by $\boldsymbol{a}$ and $\boldsymbol{b}$; hence, in two dimensions $\boldsymbol{a} \otimes \boldsymbol{b}$

with $\boldsymbol{a}$ and $\boldsymbol{b}$ ordered counterclockwise equals twice the area of the triangle spanned by $\boldsymbol{a}$ and $\boldsymbol{b}$. It follows that the cell area is, with the vertex enumeration of Figure 1,

$$|\Omega_j| = \tfrac{1}{2}(\boldsymbol{x}_3 - \boldsymbol{x}_1) \otimes (\boldsymbol{x}_4 - \boldsymbol{x}_2), \tag{2.12}$$

where we have taken advantage of the identities $\boldsymbol{a} \otimes \boldsymbol{a} = 0$ and $\boldsymbol{a} \otimes \boldsymbol{b} = -\boldsymbol{b} \otimes \boldsymbol{a}$. The vector $\boldsymbol{s}_{mn}$ normal to a cell edge $\boldsymbol{x}_m - \boldsymbol{x}_n$ with length $|\boldsymbol{x}_m - \boldsymbol{x}_n|$ will be called the cell edge vector. For complete specification, we add the requirement, assuming ξ^α = constant on $\boldsymbol{x}_m - \boldsymbol{x}_n$, that $\boldsymbol{s}_{mn}$ points into the cell where $\xi^\beta (\beta \neq \alpha)$ is largest. It follows that (cf. Figure 1)

$$\begin{aligned} \boldsymbol{s}_{12} &= \begin{pmatrix} x_1^2 - x_2^2 \\ x_2^1 - x_1^1 \end{pmatrix}, \qquad \boldsymbol{s}_{23} = \begin{pmatrix} x_3^2 - x_2^2 \\ x_2^1 - x_3^1 \end{pmatrix}, \\ \boldsymbol{s}_{34} &= \begin{pmatrix} x_4^2 - x_3^2 \\ x_3^1 - x_4^1 \end{pmatrix}, \qquad \boldsymbol{s}_{14} = \begin{pmatrix} x_4^2 - x_1^2 \\ x_1^1 - x_4^1 \end{pmatrix}. \end{aligned} \tag{2.13}$$

The covariant base vectors $\boldsymbol{a}_{(\alpha)}$ are defined by

$$\boldsymbol{a}_{(\alpha)} \equiv \frac{\partial \boldsymbol{x}}{\partial \xi^\alpha} \quad \text{or} \quad a^\beta_{(\alpha)} \equiv \frac{\partial x^\beta}{\partial \xi^\alpha}, \tag{2.14}$$

with $a^\beta_{(\alpha)}$ the Cartesian x^β-component of the vector $\boldsymbol{a}_{(\alpha)}$. We put parentheses around α to emphasize that no component is intended. Because $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ is piecewise bilinear, $\boldsymbol{a}_{(\alpha)}$ is piecewise continuous. In the cell interior and on edges ξ^β= constant, $\beta \neq \alpha$, $\boldsymbol{a}_{(\alpha)}$ is continuous, but $\boldsymbol{a}_{(\alpha)}$ is discontinuous at cell edges ξ^α = constant. For instance, $\boldsymbol{a}_{(1)}$ is discontinuous at the cell edge $\boldsymbol{x}_3 - \boldsymbol{x}_2$ (cf. Figure 1). We will need $\boldsymbol{a}_{(\alpha)}$ only in the cell center $\boldsymbol{x}_C = \frac{1}{4}(\boldsymbol{x}_1 + \boldsymbol{x}_2 + \boldsymbol{x}_3 + \boldsymbol{x}_4)$ and at the cell edge centers where $\boldsymbol{a}_{(\alpha)}$ is continuous. For mnemonic convenience, cell edge centers are indicated by subscripts N, W, S, E, where N stands for north etc., so that for instance $\boldsymbol{x}_N = \frac{1}{2}(\boldsymbol{x}_3 + \boldsymbol{x}_4)$. In $\boldsymbol{x}_C$ we have in (2.6) $s^1 = s^2 = 0$, in $\boldsymbol{x}_N$ we have $s^1 = 0,\ s^2 = 1$ etc. From (2.5), (2.6), (2.7), (2.9) and (2.14) it is easily seen that we have exactly

$$\begin{aligned} \boldsymbol{a}_{(1)C} &= (\boldsymbol{x}_E - \boldsymbol{x}_W)/\Delta\xi^1, \qquad \boldsymbol{a}_{(2)C} = (\boldsymbol{x}_N - \boldsymbol{x}_S)/\Delta\xi^2, \\ \boldsymbol{a}_{(1)N} &= (\boldsymbol{x}_3 - \boldsymbol{x}_4)/\Delta\xi^1, \qquad \boldsymbol{a}_{(1)S} = (\boldsymbol{x}_2 - \boldsymbol{x}_1)/\Delta\xi^1, \\ \boldsymbol{a}_{(2)E} &= (\boldsymbol{x}_3 - \boldsymbol{x}_2)/\Delta\xi^2, \qquad \boldsymbol{a}_{(2)W} = (\boldsymbol{x}_4 - \boldsymbol{x}_1)/\Delta\xi^2. \end{aligned} \tag{2.15}$$

The contravariant base vectors are defined by

$$\boldsymbol{a}^{(\alpha)} \equiv \nabla \xi^\alpha \quad \text{or} \quad a^{(\alpha)}_\beta \equiv \frac{\partial \xi^\alpha}{\partial x^\beta},$$

where $a^{(\alpha)}_\beta$ is the Cartesian x^β-component of the vector $\boldsymbol{a}^{(\alpha)}$. We have

$$\boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{a}_{(\beta)} = \delta^\alpha_\beta, \tag{2.16}$$

with δ^α_β the Kronecker delta. Solving (2.16) gives

$$\boldsymbol{a}^{(1)} = \frac{1}{\sqrt{g}} \begin{pmatrix} a^2_{(2)} \\ -a^1_{(2)} \end{pmatrix}, \qquad \boldsymbol{a}^{(2)} = \frac{1}{\sqrt{g}} \begin{pmatrix} -a^2_{(1)} \\ a^1_{(1)} \end{pmatrix}, \tag{2.17}$$

where $\sqrt{g}$ is the common designation in tensor analysis for the Jacobian J defined in (2.10), i.e.

$$\sqrt{g} \equiv \boldsymbol{a}_{(1)} \otimes \boldsymbol{a}_{(2)}. \tag{2.18}$$

From the smoothness properties of $\boldsymbol{a}_{(\alpha)}$ it follows that $\sqrt{g}$, $\boldsymbol{a}^{(\alpha)}$ and $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ are discontinuous at cell edges, except that $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ is continuous at cell edges of the type ξ^α = constant. From (2.17), (2.15) and (2.13) it follows that $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ is closely related to cell edge vectors:

$$(\sqrt{g}\boldsymbol{a}^{(1)})_E = \boldsymbol{s}_E/\Delta\xi^2, \tag{2.19}$$

where $\boldsymbol{s}_E = \boldsymbol{s}_{23}$, etc.

Not surprisingly, $\sqrt{g}$ is closely related to the cell area. We have, using (2.18), (2.15) and (2.12),

$$\begin{aligned} \sqrt{g_C}\Delta\xi^1\Delta\xi^2 &= (\boldsymbol{x}_E - \boldsymbol{x}_W) \otimes (\boldsymbol{x}_N - \boldsymbol{x}_S) \\ &= \tfrac{1}{4}(\boldsymbol{x}_2 + \boldsymbol{x}_3 - \boldsymbol{x}_1 - \boldsymbol{x}_4) \otimes (\boldsymbol{x}_3 + \boldsymbol{x}_4 - \boldsymbol{x}_1 - \boldsymbol{x}_2) = \Omega_j. \end{aligned}$$

3. Collocated discretization of the incompressible Navier-Stokes equations

Our major concern in this paper is with the effect of grid nonuniformity on discretization accuracy of staggered schemes. This brief section merely serves to illustrate that for collocated schemes, discretization can easily be done in the physical domain Ω, and that the smoothness properties of the coordinate mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ (which serves only to generate a grid in Ω) do not enter explicitly.

The incompressible Navier-Stokes equations can be written as (summation intended)

$$\frac{\partial q}{\partial t} + \partial f_\alpha/\partial x^\alpha = 0, \tag{3.1}$$

where

$$q \equiv \begin{pmatrix} u^1 \\ u^2 \\ 0 \end{pmatrix}, \qquad f_\alpha \equiv \begin{pmatrix} u^1u^\alpha + p\delta^1_\alpha - \mathrm{Re}^{-1}(\partial u^1/\partial x^\alpha + \partial u^\alpha/\partial x^1) \\ u^2u^\alpha + p\delta^2_\alpha - \mathrm{Re}^{-1}(\partial u^2/\partial x^\alpha + \partial u^\alpha/\partial x^2) \\ u^\alpha \end{pmatrix}.$$

Using the finite volume method and the divergence theorem, Equation (3.1) is integrated over the cell Ω_j (with vertices given in (2.3):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_j} q \,\mathrm{d}\Omega + \int_{\Gamma_j} f_\alpha n^\alpha \,\mathrm{d}\Gamma = 0. \tag{3.2}$$

One writes

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_j} q \,\mathrm{d}\Omega = |\Omega_j|\frac{\mathrm{d}q_j}{\mathrm{d}t}$$

and the value q_j is assigned to the cell center. The contour integral in (3.2) is approximated by, taking for instance the segment Γ_S of Γ

$$\int_{\Gamma_S} f_\alpha n^\alpha \,\mathrm{d}\Gamma = -f_{\alpha S} s_S^\alpha,$$

with $\boldsymbol{s}_S = \boldsymbol{s}_{12}$ the cell edge vector defined in (2.13). To complete the discretization, cell face values such as $f_{\alpha S}$ need to be approximated further and expressed in terms of the unknowns q_j located in the cell centers. How to do this for the convection terms has been the subject of much study, but does not concern us here. To outline the treatment of the viscous terms, we take as an example

$$\int_{\Gamma_E} \mathrm{Re}^{-1} \frac{\partial u^1}{\partial x^\alpha} n^\alpha \,\mathrm{d}\Gamma = \mathrm{Re}^{-1} (\nabla u^1 \cdot \boldsymbol{s})_E, \tag{3.3}$$

where ∇u^1 stands for $(\partial u^1/\partial x^1,\ \partial u^1/\partial x^2)^T$. On a general structured grid, this quantity can be expressed in terms of surrounding cell center values as follows. We write, using the convention $u^1|_j^k = u_k^1 - u_j^1$,

$$u^1 \Big|_j^{j+2e_1} = \int_{\boldsymbol{x}_j}^{\boldsymbol{x}_{j+2e_1}} \nabla u^1 \cdot \mathrm{d}\boldsymbol{x} \cong (\nabla u^1)_E \cdot \boldsymbol{c}_{(1)}, \qquad \boldsymbol{c}_{(1)} = \boldsymbol{x}_{j+2e_1} - \boldsymbol{x}_j,$$

where e_1 has been defined in (2.3). Similarly,

$$u^1 \Big|_{j-2e_2}^{j+2e_2} + u^1 \Big|_{j+2e_1-2e_2}^{j+2e_1+2e_2} = \left\{ \int_{\boldsymbol{x}_{j-2e_2}}^{\boldsymbol{x}_{j+2e_2}} + \int_{\boldsymbol{x}_{j+2e_1-2e_2}}^{\boldsymbol{x}_{j+2e_1+2e_2}} \right\} \nabla u^1 \cdot \mathrm{d}\boldsymbol{x} \cong (\nabla u^1)_E \cdot \boldsymbol{c}_{(2)},$$
$$\boldsymbol{c}_{(2)} = \boldsymbol{x} \Big|_{j-2e_2}^{j+2e_2} + \boldsymbol{x} \Big|_{j+2e_1-2e_2}^{j+2e_1+2e_2}.$$

We now have two equations for $(\nabla u^1)_E$, which we may denote by $C(\nabla u^1)_E = b$. The rows of C are $\boldsymbol{c}_{(1)}$ and $\boldsymbol{c}_{(2)}$. It is easy to see that the columns of C^{-1} are $\boldsymbol{c}^{(1)}$ and $\boldsymbol{c}^{(2)}$, defined by (cf. (2.17))

$$\boldsymbol{c}^{(1)} = \frac{1}{|C|} \begin{pmatrix} c_{(2)}^2 \\ -c_{(2)}^1 \end{pmatrix}, \qquad \boldsymbol{c}^{(2)} = \frac{1}{|C|} \begin{pmatrix} -c_{(1)}^2 \\ c_{(1)}^1 \end{pmatrix}, \qquad |C| = \boldsymbol{c}_{(1)} \otimes \boldsymbol{c}_{(2)}.$$

Hence

$$\left(\frac{\partial u^1}{\partial x^\alpha} \right)_E = c_\alpha^{(\beta)} b_\beta.$$

Collocated discretization on a grid generated by a general boundary-fitted coordinate mapping is seen to be not much more difficult than on a Cartesian grid. It is easily seen that all approximations are exact for $\boldsymbol{q}$ = constant, regardless of the sizes and shapes of the cells. Effects of grid nonuniformity are similar to those in the Cartesian case. This explains the preference shown by practitioners for collocated discretization when general structured grids are used.

As discussed in Section 1, collocated discretization of the incompressible Navier-Stokes equations has the disadvantage, that special measures must be taken to prevent spurious pressure oscillations. We will not discuss this further.

4. Staggered representation of the velocity vector field

We want to generalize the classical staggered Marker-And-Cell (MAC) scheme proposed in [1] from Cartesian to general coordinates. This means that we wish to compute the pressure in cell centers, and normal velocity components in cell edge centers. As noted in Section 2, the contravariant basis vector $\boldsymbol{a}^{(\alpha)}$ is perpendicular to curves ξ^α = constant, so at first sight the contravariant velocity components U^α, defined as $U^\alpha = \boldsymbol{u} \cdot \boldsymbol{a}^{(\alpha)}$, would seem to be suitable for representing the velocity field, and has been used as such in several publications. But we saw that $\boldsymbol{a}^{(\alpha)}$ is discontinuous at cell edges. As a consequence, the use of U^α in numerical schemes leads to bad accuracy on rough grids. However, $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ is continuous at cell edges where ξ^α = constant. Therefore the following coordinate-invariant staggered representation of the velocity field will be used (from now on denoting $\boldsymbol{x}_E$ by $\boldsymbol{x}_{j+e_1}$, etc.):

$$V^1_{j+e_1} = (\sqrt{g}\boldsymbol{a}^{(1)} \cdot \boldsymbol{u})_{j+e_1}, \qquad V^2_{j+e_2} = (\sqrt{g}\boldsymbol{a}^{(2)} \cdot \boldsymbol{u})_{j+e_2}, \tag{4.1}$$

and similarly for other cell faces. Since (using (2.19))

$$(\sqrt{g}\boldsymbol{a}^{(1)} \cdot \boldsymbol{u})_{j+e_1} = \boldsymbol{s}_E \cdot \boldsymbol{u}_E/\Delta\xi^2, \tag{4.2}$$

we see that $V^1_{j+e_1}\Delta\xi^2$ is an approximation to the volume flux through the cell edge with center at $\boldsymbol{x}_{j+e_1}$. Therefore V^1, V^2 will be called the volume flux components.

We will need to approximate V^α and $\boldsymbol{u}$ in various points in the grid; not only in the cell edge centers where the corresponding V^1 or V^2 are defined. This needs to be done carefully. A certain tedium is unavoidable. We have

$$V^\alpha = \sqrt{g}\boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u}, \qquad \boldsymbol{u} = \boldsymbol{a}_{(\alpha)} V^\alpha/\sqrt{g}. \tag{4.3}$$

From (2.17) and (2.18) we see that we can also write

$$V^1 = \boldsymbol{u} \otimes \boldsymbol{a}_{(2)}, \qquad V^2 = -\boldsymbol{u} \otimes \boldsymbol{a}_{(1)}. \tag{4.4}$$

We impose the following requirement on formulas for defining V^1 and V^2 in points other than their proper grid nodes: constant velocity fields $\boldsymbol{u}$ must be invariant under transformation to V^α-representation and back. More precisely, if $\boldsymbol{u}$ is given, and $V^1_{j+e_1}$, $V^2_{j+e_2}$ are computed with (4.4), and V^α is determined in some other point by some interpolation recipe, and $\boldsymbol{u}$ is recomputed with (4.3), then the original $\boldsymbol{u}$ must be recovered exactly in the special case that $\boldsymbol{u}$ = constant. We have found this invariance requirement essential for maintaining accuracy on rough grids.

Our definition of V^α in the cell center $\boldsymbol{x}_j$ is

$$V^1_j \equiv \tfrac{1}{2}(V^1_{j-e_1} + V^1_{j+e_1}), \qquad V^2_j \equiv \tfrac{1}{2}(V^2_{j-e_2} + V^2_{j+e_2}). \tag{4.5}$$

We show that the above invariance requirement is met. Note that from (2.15) it follows that

$$\boldsymbol{a}_{(1)j} = \tfrac{1}{2}(\boldsymbol{a}_{(1)j-e_2} + \boldsymbol{a}_{(1)j+e_2}), \qquad \boldsymbol{a}_{(2)j} = \tfrac{1}{2}(\boldsymbol{a}_{(2)j-e_1} + \boldsymbol{a}_{(2)j+e_1}). \tag{4.6}$$

(this is a consequence of the bilinearity of the coordinate mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$). Suppose $\boldsymbol{u} =$ constant. Then (4.5), (4.4) and (4.6) give

$$\begin{aligned} V_j^1 &= \tfrac{1}{2}\boldsymbol{u} \otimes (\boldsymbol{a}_{(2)j-e_1} + \boldsymbol{a}_{(2)j+e_1}) = \boldsymbol{u} \otimes \boldsymbol{a}_{(2)j}, \\ V_j^2 &= -\tfrac{1}{2}\boldsymbol{u} \otimes (\boldsymbol{a}_{(1)j-e_2} + \boldsymbol{a}_{(1)j+e_2}) = -\boldsymbol{u} \otimes \boldsymbol{a}_{(1)j}. \end{aligned} \tag{4.7}$$

Computing $\boldsymbol{u}_j$ from (4.7) using (4.3) gives

$$\boldsymbol{u}_j = \frac{1}{\sqrt{g_j}}\{\boldsymbol{a}_{(1)j}(\boldsymbol{u} \otimes \boldsymbol{a}_{(2)j}) - \boldsymbol{a}_{(2)j}(\boldsymbol{u} \otimes \boldsymbol{a}_{(1)j})\}. \tag{4.8}$$

By using (2.11), we easily see that this gives

$$\boldsymbol{u}_j = \frac{1}{\sqrt{g_j}}\boldsymbol{u}(\boldsymbol{a}_{(1)j} \otimes \boldsymbol{a}_{(2)j}) = \boldsymbol{u}, \tag{4.9}$$

which shows that our invariance requirement is met.

Next, we consider the cell-edge center $\boldsymbol{x}_{j+e_1}$. There we define $V^2_{j+e_1}$ as the average between the values at the nearest V^2-nodes:

$$V^2_{j+e_1} \equiv \tfrac{1}{4}(V^2_{j-e_2} + V^2_{j+e_2} + V^2_{j+2e_1-e_2} + V^2_{j+2e_1+e_2}). \tag{4.10}$$

In order to apply (4.3) and (4.4) we need to give $\sqrt{g}$ and $\boldsymbol{a}_{(1)}$ at $\boldsymbol{x}_{j+e_1}$, where they are discontinuous. We define $\boldsymbol{a}_{(1)}$ similar to V^2:

$$\boldsymbol{a}_{(1)j+e_1} \equiv \tfrac{1}{4}(\boldsymbol{a}_{(1)j-e_2} + \boldsymbol{a}_{(1)j+e_2} + \boldsymbol{a}_{(1)j+2e_1-e_2} + \boldsymbol{a}_{(1)j+2e_1+e_2}), \tag{4.11}$$

$$\sqrt{g_{j+e_1}} \equiv (\boldsymbol{a}_{(1)} \otimes \boldsymbol{a}_{(2)})_{j+e_1}. \tag{4.12}$$

In a similar way as before, we check whether the invariance condition is satisfied. If $\boldsymbol{u} =$ constant, (4.10), (4.4) and (4.11) give $V^2_{j+e_1} = -\boldsymbol{u} \otimes \boldsymbol{a}_{(1)j+e_1}$ and of course, by definition, $V^1_{j+e_1} = \boldsymbol{u} \otimes \boldsymbol{a}_{(2)j+e_1}$. Computing $\boldsymbol{u}_{j+e_1}$ from $V^\alpha_{j+e_1}$ gives (4.8) and (4.9) with j replaced by $j + e_1$, again showing that the invariance requirement is met.

Finally, we have to specify quantities at cell vertices $\boldsymbol{x}_{j+e_1+e_2}$. We define

$$\begin{aligned} &V^1_{j+e_1+e_2} \equiv \tfrac{1}{2}(V^1_{j+e_1} + V^1_{j+e_1+2e_2}), \qquad V^2_{j+e_1+e_2} \equiv \tfrac{1}{2}(V^2_{j+e_2} + V^2_{j+2e_1+e_2}), \\ &\boldsymbol{a}_{(1)j+e_1+e_2} \equiv \tfrac{1}{2}(\boldsymbol{a}_{(1)j+e_2} + \boldsymbol{a}_{(1)j+2e_1+e_2}), \\ &\boldsymbol{a}_{(2)j+e_1+e_2} \equiv \tfrac{1}{2}(\boldsymbol{a}_{(2)j+e_1} + \boldsymbol{a}_{(2)j+e_1+2e_2}), \\ &\sqrt{g_{j+e_1+e_2}} \equiv (\boldsymbol{a}_{(1)} \otimes \boldsymbol{a}_{(2)})_{j+e_1+e_2}. \end{aligned}$$

In the same way as before it is easily shown that the invariant requirement is met.

It will be found later that this way of coping with nonsmoothness of the coordinate mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, together with certain procedures to be followed in the next section, ensures accuracy on rough grids.

5. Coordinate-invariant discretization of the Navier-Stokes equations

Finite volume integration of the continuity equation $\operatorname{div} \boldsymbol{u} = 0$ over Ω_j gives, if we use (4.1) and (4.2)

$$\begin{aligned} 0 &= \int_{\Omega_j} \operatorname{div} \boldsymbol{u}\, \mathrm{d}\Omega = \int_{\Gamma_j} \boldsymbol{u} \cdot \mathbf{n}\, \mathrm{d}\Gamma = (\boldsymbol{u} \cdot \boldsymbol{s}) \left\{ \Big|_{j-e_1}^{j+e_1} + \Big|_{j-e_2}^{j+e_2} \right\} \\ &= \Delta\xi^2 V^1 \Big|_{j-e_1}^{j+e_1} + \Delta\xi^1 V^2 \Big|_{j-e_2}^{j+e_2} . \end{aligned} \tag{5.1}$$

For robustness, the discretization scheme should be coordinate invariant. This is the case for (5.1), because it contains a contravariant representation of the velocity field.

A straightforward way to proceed would be to discretize the momentum equations written in coordinate invariant form, using tensor analysis. However, in this formulation the so-called Christoffel symbols occur. These involve second derivatives of the mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$. Because the mapping is piecewise bilinear the Christoffel symbols are ‘infinite’ at cell edges. Approximation of the Christoffel symbols by straightforward finite differences gives reasonable results on smooth grids only. This is perhaps what has led to a widespread belief that staggered discretization is inaccurate in general coordinates. In order to avoid the difficulty with the Christoffel symbols we first transform only the independent variables, obtaining a form that is not coordinate-invariant, which is discretized and used as a stepping stone to arrive at a coordinate-invariant discretization.

The derivative of some quantity φ transforms according to

$$\frac{\partial\varphi}{\partial x^\alpha} = a_\alpha^{(\beta)} \frac{\partial\varphi}{\partial\xi^\beta}. \tag{5.2}$$

By using the identity

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

we can rewrite this as

$$\frac{\partial\varphi}{\partial x^\alpha} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial\xi^\beta}(\sqrt{g} a_\alpha^{(\beta)} \varphi). \tag{5.3}$$

The momentum equations can be written as

$$\frac{\partial\boldsymbol{u}}{\partial t} + \frac{\partial}{\partial x^\alpha}(u^\alpha \boldsymbol{u}) = -\nabla p + \mathrm{Re}^{-1} \frac{\partial\boldsymbol{e}^{(\alpha)}}{\partial x^\alpha}, \tag{5.4}$$

where

$$\boldsymbol{e}^{(\alpha)} \equiv \begin{pmatrix} \dfrac{\partial u^1}{\partial x^\alpha} + \dfrac{\partial u^\alpha}{\partial x^1} \\ \dfrac{\partial u^2}{\partial x^\alpha} + \dfrac{\partial u^\alpha}{\partial x^2} \end{pmatrix}.$$

By applying (5.3), we can rewrite Equation (5.4) as

$$N(\boldsymbol{u}, p) \equiv \frac{\partial\boldsymbol{u}}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial\xi^\alpha}(\boldsymbol{u} V^\alpha) + \nabla p - \mathrm{Re}^{-1} \frac{1}{\sqrt{g}} \frac{\partial}{\partial\xi^\alpha}(\sqrt{g} a_\beta^{(\alpha)} \boldsymbol{e}^{(\beta)}) = 0. \tag{5.5}$$

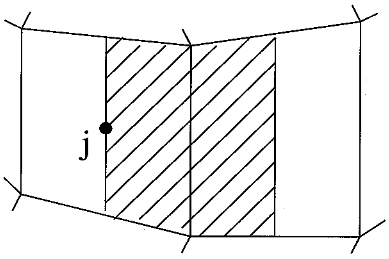

Figure 2. Shifted finite volume Ω_{j+e_1}.

This equation is integrated over the shifted finite volume Ω_{j+e_1} depicted in Figure 2. Treating each term successively, we obtain

$$\int_{\Omega_{j+e_1}} \frac{\partial \boldsymbol{u}}{\partial t}\, \mathrm{d}\Omega = |\Omega_{j+e_1}|\, \mathrm{d}\boldsymbol{u}_{j+e_1}/\mathrm{d}t,$$

$$\int_{\Omega_{j+e_1}} \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha}(\boldsymbol{u}V^\alpha)\, \mathrm{d}\Omega = \int_{G_{j+e_1}} \frac{\partial}{\partial \xi^\alpha}(\boldsymbol{u}V^\alpha)\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2. \tag{5.6}$$

The two parts of $|\Omega_{j+e_1}|$ are evaluated exactly similar to (2.12). We have

$$\boldsymbol{I}_{11} \equiv \int_{G_{j+e_1}} \frac{\partial}{\partial \xi^1}(\boldsymbol{u}V^1)\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2 = \int_{\xi^2_{j_2-1/2}}^{\xi^2_{j_2+1/2}} (\boldsymbol{u}V^1)\Big|_{j_1}^{j_1+1}\, \mathrm{d}\xi^2,$$

where the fact that $\boldsymbol{u}V^1$ is continuous in G_{j+e_1} has been used. This integral is approximated by the midpoint rule:

$$\boldsymbol{I}_{11} \cong \Delta\xi^2(\boldsymbol{u}V^1)\Big|_j^{j+2e_1}.$$

This is approximated further by using

$$V_j^1 \cong \tfrac{1}{2}(V_{j-e_1}^1 + V_{j+e_1}^1). \tag{5.7}$$

Similarly,

$$\boldsymbol{I}_{12} \equiv \int_{G_{j+e_1}} \frac{\partial}{\partial \xi^2}(\boldsymbol{u}V^2)\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2 \cong \Delta\xi^1(\boldsymbol{u}V^2)\Big|_{j+e_1-e_2}^{j+e_1+e_2}. \tag{5.8}$$

This is further approximated using $V_{j+e_1+e_2}^2 \cong \frac{1}{2}(V_{j+e_2}^2 + V_{j+2e_1+e_2}^2)$. Integration of the pressure term is done as follows:

$$I_{13} \equiv \int_{\Omega_{j+e_1}} \nabla p\, \mathrm{d}\Omega = \nabla p_{j+e_1}|\Omega_{j+e_1}|.$$

The term ∇p_{j+e_1} is expressed in terms of surrounding nodal values in the same way as done for ∇u^1 in (3.3), which results in

$$\nabla p_{j+e_1} \cong p\Big|_j^{j+2e_1}\boldsymbol{c}^{(1)} + \left\{p\Big|_{j-2e_2}^{j+2e_2} + p\Big|_{j+2e_1-2e_2}^{j+2e_1+2e_2}\right\}\boldsymbol{c}^{(2)},$$

$$\boldsymbol{c}^{(1)} = \frac{1}{C}\begin{pmatrix} c_{(2)}^2 \\ -c_{(2)}^1 \end{pmatrix}, \qquad \boldsymbol{c}^{(2)} = \frac{1}{C}\begin{pmatrix} -c_{(1)}^2 \\ c_{(1)}^1 \end{pmatrix}, \qquad C = \boldsymbol{c}_{(1)} \otimes \boldsymbol{c}_{(2)}, \tag{5.9}$$

$$\boldsymbol{c}_{(1)} = \boldsymbol{x}\Big|_j^{j+2e_1}, \qquad \boldsymbol{c}_{(2)} = \boldsymbol{x}\Big|_{j-2e_2}^{j+2e_2} + \boldsymbol{x}\Big|_{j+2e_1-2e_2}^{j+2e_1+2e_2}.$$

Integration of the viscous term over Ω_{j+e_1} gives the following two contributions:

$$\boldsymbol{I}_{14} \equiv \mathrm{Re}^{-1} \int_{\xi^2_{j_2-1/2}}^{\xi^2_{j_2+1/2}} (\sqrt{g} a^{(1)}_{\beta} \boldsymbol{e}^{(\beta)}) \Big|_{j_1}^{j_1+1} \, \mathrm{d}\xi^2. \tag{5.10}$$

In (5.10), $\sqrt{g}\boldsymbol{a}^{(1)}$ is constant. We write

$$\boldsymbol{I}_{14} \cong \mathrm{Re}^{-1} \Delta\xi^2 (\sqrt{g} a^{(1)}_{\beta} \boldsymbol{e}^{(\beta)}) \Big|_{j}^{j+2e_1}.$$

The second contribution is

$$\boldsymbol{I}_{15} \equiv \mathrm{Re}^{-1} \int_{\xi^1_{j_1}}^{\xi^1_{j_1+1}} (\sqrt{g} a^{(2)}_{\beta} \boldsymbol{e}^{(\beta)}) \Big|_{j_2-1/2}^{j_2+1/2} \, \mathrm{d}\xi^1. \tag{5.11}$$

In (5.11), $\sqrt{g}\boldsymbol{a}^{(2)}$ is piecewise constant. We make the following approximation:

$$\boldsymbol{I}_{15} \cong \mathrm{Re}^{-1} \Delta\xi^1 (\sqrt{g} a^{(2)}_{\beta} \boldsymbol{e}^{(\beta)}) \Big|_{j+e_1-e_2}^{j+e_1+e_2},$$

where we define

$$(\sqrt{g}\boldsymbol{a}^{(2)})_{j+e_1\pm e_2} \equiv \tfrac{1}{2}\{(\sqrt{g}\boldsymbol{a}^{(2)})_{j\pm e_2} + (\sqrt{g}\boldsymbol{a}^{(2)})_{j+2e_1\pm e_2}\}.$$

In $\boldsymbol{I}_{14}$ and $\boldsymbol{I}_{15}$, $\boldsymbol{e}^{(\beta)}$ has to be further approximated, which requires discretizations of derivatives of $\boldsymbol{u}$. We start with $\boldsymbol{I}_{14}$ and write, using (5.2),

$$\left(\frac{\partial \boldsymbol{u}}{\partial x^{\beta}}\right)_j = \left(a^{(\alpha)}_{\beta} \frac{\partial \boldsymbol{u}}{\partial \xi^{\alpha}}\right)_j,$$

which is approximated by

$$\left(a^{(1)}_{\beta} \frac{\partial \boldsymbol{u}}{\partial \xi^1}\right)_j \cong \frac{1}{\Delta\xi^1} (a^{(1)}_{\beta})_j \boldsymbol{u} \Big|_{j-e_1}^{j+e_1}, \qquad \left(a^{(2)}_{\beta} \frac{\partial \boldsymbol{u}}{\partial \xi^2}\right)_j \cong \frac{1}{\Delta\xi^2} (a^{(2)}_{\beta})_j \boldsymbol{u} \Big|_{j-e_2}^{j+e_2}.$$

The same procedure cannot be followed for $\boldsymbol{I}_{15}$, because in this case we are at a cell vertex, where the geometric quantities are discontinuous. Instead, we proceed in a similar way as for ∇p, and write

$$u^{\alpha} \Big|_{j+e_2}^{j+2e_1+e_2} = \int_{\mathbf{x}_{j+e_2}}^{\mathbf{x}_{j+2e_1+e_2}} \nabla u^{\alpha} \cdot \mathrm{d}\boldsymbol{x} \cong \nabla u^{\alpha}_{j+e_1+e_2} \cdot \boldsymbol{c}_{(1)}, \qquad \boldsymbol{c}_{(1)} \equiv \boldsymbol{x} \Big|_{j+e_2}^{j+2e_1+e_2}, \tag{5.12}$$

$$u^{\alpha} \Big|_{j+e_1}^{j+e_1+2e_2} \cong \nabla u^{\alpha}_{j+e_1+e_2} \cdot \boldsymbol{c}_{(2)}, \qquad \boldsymbol{c}_{(2)} \equiv \boldsymbol{x} \Big|_{j+e_1}^{j+e_1+2e_2}, \tag{5.13}$$

with solution

$$\nabla u^{\alpha}_{j+e_1+e_2} = u^{\alpha} \Big|_{j+e_2}^{j+2e_1+e_2} \boldsymbol{c}^{(1)} + u^{\alpha} \Big|_{j+e_1}^{j+e_1+2e_2} \boldsymbol{c}^{(2)}, \tag{5.14}$$

with $\boldsymbol{c}^{(\alpha)}$ expressed in terms of $\boldsymbol{c}_{(\alpha)}$ by (5.9).

In a similar way, Equation (5.5) is integrated over the finite volume Ω_{j+e_2}, which is shifted in the ξ^2-direction. This results in the following formulas:

$$\int_{\Omega_{j+e_2}} \frac{\partial \boldsymbol{u}}{\partial t}\, \mathrm{d}\Omega = |\Omega_{j+e_2}|\, \mathrm{d}\boldsymbol{u}_{j+e_2}/\mathrm{d}t, \qquad \int_{\Omega_{j+e_2}} \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha}(\boldsymbol{u}V^\alpha)\, \mathrm{d}\Omega = \boldsymbol{I}_{21} + \boldsymbol{I}_{22},$$

$$\boldsymbol{I}_{21} \cong \Delta\xi^2(\boldsymbol{u}V^1)\Big|_{j-e_1+e_2}^{j+e_1+e_2}, \qquad \boldsymbol{I}_{22} \equiv \tfrac{1}{2}\Delta\xi^1(\boldsymbol{u}V^2)\Big|_{j-e_2}^{j+3e_2},$$

$$\boldsymbol{I}_{23} \equiv \int_{\Omega_{j+e_2}} \nabla p\, \mathrm{d}\Omega = \nabla p_{j+e_2}|\Omega_{j+e_2}|,$$

$$\nabla p_{j+e_2} \cong p\Big|_{j}^{j+2e_2}\boldsymbol{c}^{(1)} + \left\{p\Big|_{j-2e_1}^{j+2e_1} + p\Big|_{j-2e_1+2e_2}^{j+2e_1+2e_2}\right\}\boldsymbol{c}^{(2)},$$

with $\boldsymbol{c}^{(\alpha)}$ according to (5.9), with

$$\boldsymbol{c}_{(1)} = \boldsymbol{x}\Big|_{j}^{j+2e_2}, \qquad \boldsymbol{c}_{(2)} = \boldsymbol{x}\Big|_{j-2e_1}^{j+2e_1} + \boldsymbol{x}\Big|_{j-2e_1+2e_2}^{j+2e_1+2e_2},$$

$$\mathrm{Re}^{-1}\int_{\Omega_{j+e_2}} \frac{1}{\sqrt{g}}\frac{\partial}{\partial\xi^\alpha}(\sqrt{g}a_\beta^{(\alpha)}\boldsymbol{e}^{(\beta)})\, \mathrm{d}\Omega = \boldsymbol{I}_{24} + \boldsymbol{I}_{25},$$

$$\boldsymbol{I}_{24} = \mathrm{Re}^{-1}\int_{\xi^2_{j_2}}^{\xi^2_{j_2+1}}(\sqrt{g}a_\beta^{(1)}\boldsymbol{e}^{(\beta)})\Big|_{j_1-1/2}^{j_1+1/2}\mathrm{d}\xi^2 \cong \mathrm{Re}^{-1}\Delta\xi^2(\sqrt{g}a_\beta^{(1)}\boldsymbol{e}^{(\beta)})\Big|_{j-e_1+e_2}^{j+e_1+e_2}, \tag{5.15}$$

$$\boldsymbol{I}_{25} = \mathrm{Re}^{-1}\int_{\xi^1_{j_1-1/2}}^{\xi^1_{j_1+1/2}}(\sqrt{g}a_\beta^{(2)}\boldsymbol{e}^{(\beta)}\, \mathrm{d}\xi^1\Big|_{j_2}^{j_2+1}\mathrm{d}\xi^1 \cong \mathrm{Re}^{-1}\Delta\xi^1(\sqrt{g}a_\beta^{(2)}\boldsymbol{e}^{(\beta)})\Big|_{j}^{j+2e_2}. \tag{5.16}$$

In (5.15) we define

$$(\sqrt{g}\boldsymbol{a}^{(1)})_{j\pm e_1+e_2} \equiv \tfrac{1}{2}\{(\sqrt{g}\boldsymbol{a}^{(2)})_{j\pm e_1} + (\sqrt{g}\boldsymbol{a}^{(2)})_{j\pm e_1+2e_e}\}.$$

Furthermore, in (5.15) and (5.16) $\boldsymbol{e}^{(\beta)}$ is approximated by the same method as for $\boldsymbol{I}_{14}$ and $\boldsymbol{I}_{15}$.

We may regard this finite volume discretization of (5.5), i.e.

$$\int_{\Omega_{j+e_1}} N(\boldsymbol{u}, p)\, \mathrm{d}\Omega = 0 \quad \text{and} \quad \int_{\Omega_{j+e_2}} N(\boldsymbol{u}, p)\, \mathrm{d}\Omega = 0, \tag{5.17}$$

with the integrals approximated in the way described above, as semi-discretized evolution equations for $\boldsymbol{u}$ at the cell-edge centers (by semi-discretization we mean discretization in space but not in time). This is not what we want, because we wish to generalize the classical staggered scheme of [1] from Cartesian to general coordinates, which implies that we want evolution equations for the normal velocity components at cell edge centers, or preferably for the volume flux components $V^1_{j+e_1}$ and $V^2_{j+e_2}$, as argued in Section 4. With the use of (4.4), we achieve this by replacing (5.17) by

$$-\boldsymbol{a}_{(2)j+e_1} \otimes \int_{\Omega_{j+e_1}} N(\boldsymbol{u}, p)\, \mathrm{d}\Omega = 0, \qquad \boldsymbol{a}_{(1)j+e_2} \otimes \int_{\Omega_{j+e_2}} N(\boldsymbol{u}, p)\, \mathrm{d}\Omega = 0.$$

We obtain, using (5.6) and (4.4),

$$-\boldsymbol{a}_{(2)j+e_1} \otimes \int_{\Omega_{j+e_1}} \frac{\partial \boldsymbol{u}}{\partial t}\, \mathrm{d}\Omega = |\Omega_{j+e_1}|\, \mathrm{d}V^1_{j+e_1}/\mathrm{d}t,$$

which is indeed what we want. Furthermore, using (5.7),

$$-\boldsymbol{a}_{(2)j+e_1} \otimes \boldsymbol{I}_{11} \cong -\tfrac{1}{2}\Delta\xi^2 \boldsymbol{a}_{(2)j+e_1} \otimes (\boldsymbol{u}V^1)\Big|^{j+3e_1}_{j-e_1}. \tag{5.18}$$

In order to obtain a closed system of equations involving only the unknowns V^α and p, and to obtain a fully coordinate-invariant formulation, $\boldsymbol{u}$ in (5.18) is expressed in terms of V^α in the way described in Section 4. That is, we write, for example, $\boldsymbol{u}_{j-e_1} = (\boldsymbol{a}_{(\alpha)}V^\alpha/\sqrt{g})_{j-e_1}$ and define $\boldsymbol{a}_{(\alpha)j-e_1}$, $V^2_{j-e_1}$ and $\sqrt{g}_{j-e_1}$ according to (4.10)–(4.12). The other contributions to the finite volume integral over Ω_{j+e_1} are handled similarly. For the integral over Ω_{j+e_2} we have

$$\boldsymbol{a}_{(1)j+e_2} \otimes \int_{\Omega_{j+e_2}} \frac{\partial \boldsymbol{u}}{\partial t}\, \mathrm{d}\Omega = |\Omega_{j+e_2}|\, \mathrm{d}V^2_{j+e_2}/\mathrm{d}t,$$

which is again precisely what we want. There is no need to describe the procedure for the remaining terms.

What we obtain in this way is a fully coordinate-invariant discretization, that in the case of the identity mapping $\boldsymbol{x} = \boldsymbol{\xi}$ reduces to the classical Cartesian staggered grid discretization of [1]. Furthermore, the discretization error is zero for $\boldsymbol{u}$ and ∇p constant, regardless of smoothness and nonorthogonality of the grid. It would be too tedious to show this here, but verification by numerical experiment is easy.

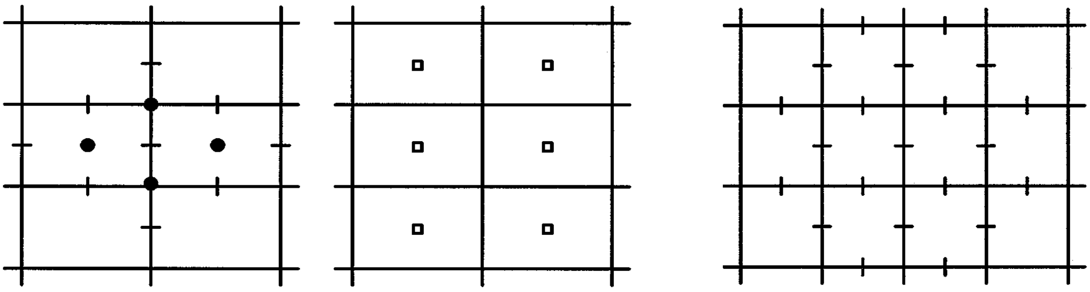

Figure 3. Stencils of inertia, pressure and viscous terms: ● : **u**, – : V^1, | : V^2, □ : p.

Figure 3 shows at which points of the grid (in the $\boldsymbol{\xi}$-plane) $\boldsymbol{u}$, V^α and p are used in the finite volume discretization for a V^1-cell of the inertia, pressure and viscous terms. V^α-points needed to express $\boldsymbol{u}$ and $\boldsymbol{e}^{(\beta)}$ in terms of V^α are included. The stencil of the viscous term contains 9 V^1-points. This is not more than expected, since mixed derivatives have to be approximated if the coordinates are nonorthogonal. In addition, the viscous stencil contains 12 V^2-points. In view of the fact that second order differential operators have to be approximated in the four V^2-points surrounding the central V^1-point, this seems a reasonable number.

One may wonder how it is possible that a coordinate-invariant discretization has been obtained without encountering Christoffel symbols. This may, to a certain extent, be elucidated as follows. The Christoffel symbols are defined by

$$\left\{ \begin{array}{c} \alpha \\ \beta\gamma \end{array} \right\} \equiv \boldsymbol{a}^{(\alpha)} \cdot \frac{\partial \boldsymbol{a}_{(\beta)}}{\partial \xi^{\gamma}}. \tag{5.19}$$

In (5.17), more particularly, for example, in the contribution $\boldsymbol{a}_{(2)j+e_1} \otimes \boldsymbol{I}_{14}$, products of $\boldsymbol{a}_{(2)}$ in one point and $\boldsymbol{a}^{(\alpha)}$ in neighbouring points are hidden, similar to what one would get by discretizing (5.19) (assuming differentiability of $\boldsymbol{a}_{(\beta)}$). Contravariant base vectors $\boldsymbol{a}^{(\alpha)}$ are implicitly present in $\boldsymbol{I}_{14}$, because $\boldsymbol{c}_{(\alpha)}$ in (5.12), (5.13) is related to $\boldsymbol{a}_{(\alpha)}$, and $\boldsymbol{c}^{(\alpha)}$ in (5.14) to $\boldsymbol{a}^{(\alpha)}$. Another way of looking at our circumvention of Christoffel symbols is to note that finite volume integration precedes transformation to invariant form, so that integrals of the Christoffel symbols are required, removing the derivative in (5.19).

The spatial discretization is completed by implementing the boundary conditions. This presents no particular problem, and will not be discussed here.

6. Solution methods

If we put all unknows $V^1_{j+e_1}$ and $V^2_{j+e_2}$ in some order in an algebraic vector u and all unknowns p_j in an algebraic vector p, the semi-discretized incompressible Navier-Stokes equations go over in a differential-algebraic system of the following structure:

$$\frac{\mathrm{d}u}{\mathrm{d}t} + N(u) + Gp = f(t), \qquad Du = g(t). \tag{6.1}$$

Here N is a nonlinear algebraic operator arising from the discretization of the inertia and viscous terms, G and D are linear algebraic operators corresponding to the gradient and divergence operators, and f and g are known terms, arising from the boundary conditions.

Temporal discretization methods carry over directly from the Cartesian to the general coordinates case, and result in systems of the following general form, assuming a constant timestep τ and a superscript n to indicate time level $t^n = n\tau$:

$$A(u^n) + \tau G p^{n-1/2} = r^n, \qquad Du^n = g(t^n), \tag{6.2}$$

where r^n is known from previous time steps and the boundary conditions. For explicit methods, the nonlinear algebraic operator A is the identity. For example, the second order Adams–Bashforth method applied to (6.1) gives

$$\begin{aligned} &u^n - u^{n-1} + \tfrac{1}{2}\tau\{3N(u^{n-1}) - N(u^{n-2}) + G(3p^{n-1} - p^{n-2})\} \\ &\quad = \tfrac{1}{2}\tau\{3f(t^{n-1}) - f(t^{n-2})\}, \\ &Du^n = g(t^n). \end{aligned} \tag{6.3}$$

In order to avoid an overdetermined system, u^n and p^{n-1} must be determined simultaneously. The solution for u^n is not affected if we define $p^{n-1/2} = \frac{3}{2}p^{n-1} - \frac{1}{2}p^{n-2}$, which results in a system of the form (6.2). The formulation (6.2) brings out more clearly than (6.3) the fact that the pressure acts as a Lagrangian multiplier guaranteeing satisfaction of the continuity equation. As a second example, application of the θ-method to (6.1) gives

$$\begin{aligned}&u^n - u^{n-1} + \theta\tau N(u^n) + (1-\theta)\tau N(u^{n-1}) + \tau G p^{n-1/2} = \theta f(t^n) + (1-\theta) f(t^{n-1}),\\ &Du^n = g(t^n),\end{aligned}$$

which is again of the form (6.2).

For computing time-dependent solutions of (6.2), pressure correction is the method of choice. With the pressure-correction method, (6.2) is not solved as it stands, but first a prediction u^* is made that does not satisfy the continuity equation. Then a correction is computed involving the pressure such that continuity is satisfied. The advantage of this is that u^n and $p^{n-1/2}$ are solved for separately. The pressure correction method is given by

$$A(u^*) + \tau G p^{n-3/2} = r^n, \tag{6.4}$$

$$u^n - u^* + \tau G(p^{n-1/2} - p^{n-3/2}) = 0, \tag{6.5}$$

$$Du^n = g(t^n). \tag{6.6}$$

Equation (6.4) more or less amounts to solving discretized convection-diffusion equations for the predicted velocity components. Next, $p^{n-1/2}$ is computed by applying D to (6.5) and using (6.6), resulting in

$$DG\delta p = \frac{1}{\tau}(Du^* - g(t^n)), \qquad p^{n-1/2} = p^{n-3/2} + \delta p. \tag{6.7}$$

After δp has been computed, u^n follows from (6.5). Because DG is a discrete Laplacian, (6.7) is frequently called the pressure Poisson equation. Note that no boundary conditions need to be invoked for δp, which is fortunate, because no such conditions are given in general. The boundary conditions have already been incorporated in D, G, g and r^n; the operator DG works exclusively on pressure values in grid points in the interior of the domain. The issue of boundary conditions for the pressure Poisson equation (which does not arise with the approach followed here) is discussed extensively in [57], [58], [59].

Even if the method is explicit $(A = I)$, we still have to solve an implicit system for δp. This is a consequence of the differential-algebraic nature of (6.1). By elimination of u^* it is easily seen that in the explicit case the pressure correction method (6.4)–(6.6) is equivalent to (6.2), and this remains true if $p^{n-3/2}$ is neglected in (6.4) and (6.5). But in the implicit case this does not hold, and inclusion of a sufficiently accurate first guess for $p^{n-1/2}$ in (6.4), such as $p^{n-3/2}$, seems to be necessary to obtain full temporal accuracy for u^n. This may make it necessary to compute the initial pressure at the starting step $(n = 1)$, to be substituted for $p^{-1/2}$. This may be done as follows. Application of D to (6.1) gives

$$\frac{\mathrm{d}g(t)}{\mathrm{d}t} + DN(u(t)) + DGp(t) = Df(t). \tag{6.8}$$

After putting $t = 0$ and solving $p(0)$ from (6.8), we put $p^{-1/2} = p(0)$ in (6.4).

In the Cartesian case, convergence of the method is studied theoretically in [60] and references quoted there. The pressure correction method has been formulated and studied in [1], [61]–[68]. Indications are that the temporal accuracy of u^n is the same as the order of accuracy of the underlying time stepping method, but that the accuracy of $p^{n-1/2}$ is $O(\tau)$, irrespective of the temporal discretization used. If one desires, a pressure field with improved accuracy can be obtained after u^n has been computed, by using (6.8) for $t = t^n$ to find p^n with the same order of temporal accuracy as u^n.

For stability of (6.4)–(6.6) it seems necessary that (6.4) is stable. Since (6.4) is very close to a system of convection-diffusion equations for V^1 and V^2, application of Fourier analysis to show von Neumann stability is relatively easy ([69]). We think, supported by numerical evidence, that stability of (6.4) is sufficient for stability of (6.4)–(6.6).

In the explicit case, the bulk of the computing time goes to solving δp from (6.7), so it pays to do this efficiently. On uniform grids in orthogonal coordinates, fast Poisson solvers based on fast Fourier transformation and/or cyclic reduction are in widespread use. A survey of these methods is given in [70]. On general grids these methods are not applicable. In general, the matrix of (6.7) is not symmetric, and when the coordinates are strongly nonorthogonal it is not an M-matrix. But multigrid [71], [72], [73] and preconditioned Krylov subspace methods [74], [41], [42], [73] work fine. For robustness, the smoother or preconditioner must be able to cope with high cell aspect ratios. This, and the influence of mixed derivatives is modeled by the rotated anisotropic diffusion equation of Chapter 7 in [72], where efficient and robust smoothers for this problem are identified; these may also be expected to be effective preconditioners for Krylov subspace methods. These methods are of line relaxation or incomplete LU type, namely damped alternating Jacobi or zebra relaxation, alternating line Gauss–Seidel and various ILU variants; for details see [72].

In the implicit case, (6.4) also has to be solved iteratively. The nonlinear character is taken care of by some outer iteration or a prediction of nonlinear coefficients by extrapolation from previous time levels, so for the present discussion A is assumed linear. With central discretization of convection and a practical Reynolds number A will not be an M-matrix, but for τ small enough the main diagonal is enhanced sufficiently by the time derivative to put iterative methods in business. Otherwise, the system may be preconditioned by a sufficiently upwind-biased scheme (defect correction). In nonorthogonal coordinates mixed derivatives can be significant, which causes ADI and other fractional step methods to loose much of their lustre. Again, Krylov subspace and multigrid methods may be used. The convection-diffusion equation can serve as a testbed for identifying robust and efficient smoothers and preconditioners. In [72] the same methods as before are found to be eligible. Navier-Stokes applications are shown in [72], [41], [42], [73].

7. Applications and extensions

In order to illustrate that the generalized coordinates staggered discretization described before is at least as accurate as the discretization methods that are mostly used at present in codes to compute Navier-Stokes solutions in complicated domains, namely collocated finite volume methods using Rhie-Chow interpolation ([12]) and finite element methods, we approximate a simple exact solution on a rough grid. The exact solution is Poiseuille flow:

$$u^1 = x^2(1-x^2), \qquad u^2 = 0, \qquad p = -2\mathrm{Re}^{-1}x^1, \qquad \mathrm{Re} = 1.$$

The grid, shown in Figure 4, is chosen rough deliberately. Figures 5, 6 and 7 give streamlines and isobars for the staggered discretization, a finite-element code using $Q_1 - P_0$ elements (quadrilaterals with bilinear basis functions for the velocity and constant basis functions for the pressure) and a commercial code using collocated discretization with Rhie-Chow interpolation.

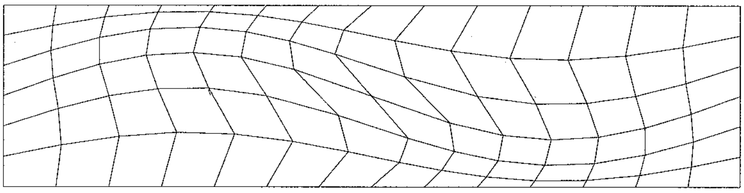

Figure 4. Grid for Poiseuille flow.

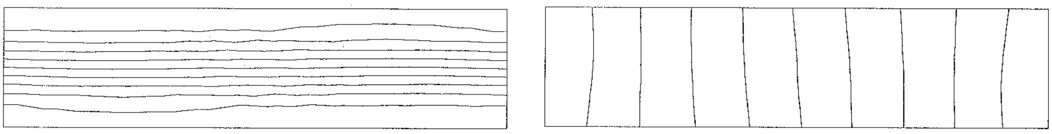

Figure 5. Streamlines and isobars for staggered discretization.

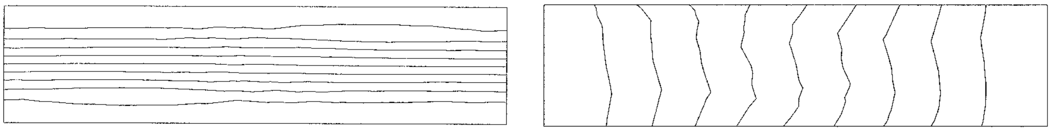

Figure 6. Streamlines and isobars for finite element method.

Figure 8 gives an even wilder grid, meant to investigate the effect of sudden refinement and derefinement. Results are shown in Figures 9 and 10. We do not have results for the collocated code for this case. The streamlines should be straight and the isobars straight and equally spaced. Clearly, the staggered discretization is more accurate than the other two methods. This illustrates that staggered schemes are not inherently inaccurate on general grids. On the contrary, they can be quite accurate, provided the smoothness properties of the boundary-fitted coordinate mapping are carefully taken into account. This can be done in the way described above.

We will now discuss extension to compressible flows such that accuracy and efficiency are uniform in the Mach number M as M ↓ 0. The limit M ↓ 0 of the Euler and Navier-Stokes equation is singular ([75], [76]). Classical methods to compute compressible flows break down as M ↓ 0. Measures can be taken to decrease to a certain extent the lowest value of M for which reasonable results can be obtained ([77]–[80]) (until $M \cong 0{\cdot}1$), but these measures usually falsify transient behaviour and are therefore limited to stationary flows, and $M = 0$ cannot be reached. A method that tends to an established method for incompressible flows as M ↓ 0 obviously does not break down for small Mach numbers. A way to obtain methods with Mach-uniform accuracy and efficiency is therefore to generalize methods designed for the incompressible case to the compressible case. For the stationary Navier-Stokes equations this has been done with a collocated scheme in [81] and for a staggered scheme in [82]–[85]. Collocated schemes add, either implicitly or explicitly, an artificial regularizing term to the mass

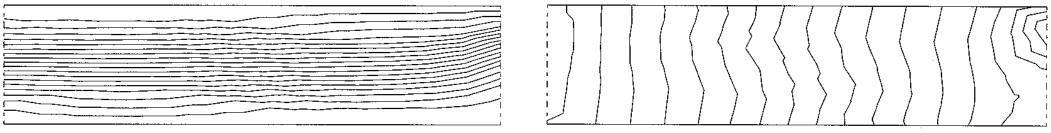

Figure 7. Streamlines and isobars for collocated discretization.

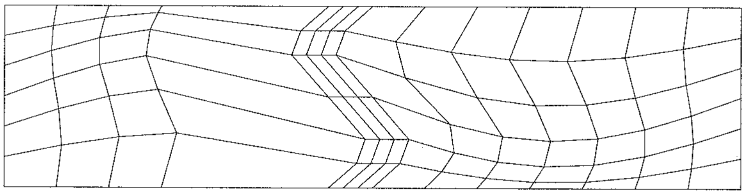

Figure 8. Grid for Poiseuille flow.

conservation equation; for an explicit expression of this term generated by the Rhie-Chow velocity interpolation method [12], see [10]. Introduction of weak compressibility entails a small physical modification of the mass conservation equation, which may be dominated by the artificial regularizing term. This is not the case for staggered schemes, which seems to be an advantage, especially for instationary flows.

The difficulties associated with M ↓ 0 already fully manifest themselves for the Euler equations, to which we restrict ourselves here. A generalization of the staggered scheme and pressure correction method described before to the instationary compressible Euler equations is given in [86]. The M ↓ 0 singularity is removed by a proper choice of units. Except for the pressure, these are the usual inflow or stagnation conditions, indicated by a subscript ∞ or 0, respectively. The choice for the dimensionless pressure is crucial:

$$\tilde{p} = \frac{p - p_1}{\rho_0 w_\infty^2},$$

where p_1 is the outlet pressure (subsonic inflow) or the inlet pressure (supersonic inflow), and w is the velocity magnitude. The primary (dimensionless) unknowns are (deleting tildes) $p,\ u^\alpha$ and enthalpy h. The density ρ follows from the dimensionless equation of state, which is easily found to be:

$$\rho = \rho(p,h) = \frac{\gamma M_\infty^2}{1 + \frac{\gamma-1}{2} M_\infty^2} \frac{p}{h} + \left[p_v \left\{ \left(1 + \frac{\gamma - 1}{2} M_\infty^2 \right)^{\frac{-\gamma}{\gamma-1}} - 1 \right\} + 1 \right] \frac{1}{h},$$

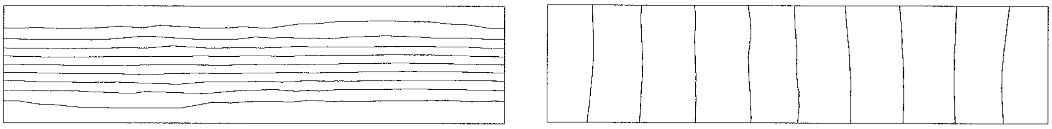

Figure 9. Streamlines and isobars for staggered discretization.

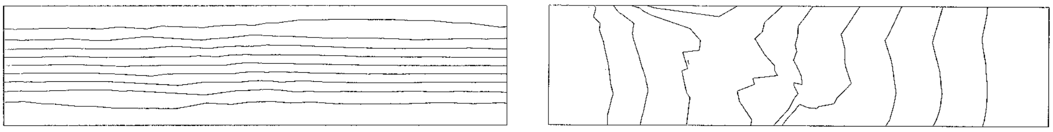

Figure 10. Streamlines and isobars for finite element method.

where$p_v = (p_1^* - p_0^*)/\,(p_\infty^* - p_0^*)$, denoting dimensional quantities by an asterisk. The Euler equations are invariant under this nondimensionalization. With pressure as primary unknown, the mass conservation equation is given by

$$\left(\frac{\partial\rho}{\partial p}\right)_h \frac{\partial p}{\partial t} + \left(\frac{\partial\rho}{\partial h}\right)_p \frac{\partial h}{\partial t} + \frac{\partial\rho u^\alpha}{\partial x^\alpha} = 0.$$

Note that as $\mathrm{M}_\infty \downarrow 0$ there is no singularity in the equations of motion and the equation of state. The equations are discretized on a staggered grid in general coordinates in the way described before. Time-stepping takes place with the pressure-correction method as described in [86]. In subsonic flow central discretization can be used. As for standard methods for compressible flows, in order to enforce thermodynamic irreversibility, some form of upwind-biased scheme is to be used when $\mathrm{M} > 1$, with, in addition, an upwind-biased evaluation of the density. The resulting method performs well on the full range of Mach numbers from zero until real gas effects set in. Examples are given in [86].

Extension to turbulent flow is straightforward. Two-equation turbulence models, such as the $k - \varepsilon$ and $k - \omega$ models, can be discretized on general staggered grids using the principles outlined before ([46]–[50]). Turbulence quantities such as k, ε and ω are located in the same grid points as the pressure. For higher order accuracy while maintaining positivity of k and ε, higher order upwind biased schemes with flux limiting can be extended to general staggered grids [47]. Robustness with respect to grid nonuniformity and nonorthogonality is illustrated for a number of turbulent flows in [48]–[50].

8. Concluding remarks

We have shown how the classical staggered scheme for the incompressible Navier-Stokes equation can be generalized from Cartesian to coordinate invariant form on general strongly nonuniform and nonorthogonal grids in such a way that the accuracy is at least as good as a typical finite element method and a collocated scheme using Rhie-Chow velocity interpolation. The accuracy is maintained when two-equation turbulence modeling is included. Extension of the usual solution techniques to general structured grids is straightforward. Because on staggered grids no artificial regularization is needed for the mass conservation equation, accurate extension to weakly compressible instationary flows is possible, provided the equations are made dimensionless in such a way that the $\mathrm{M} \downarrow 0$ singularity is removed from the equations. The resulting scheme is found to work satisfactorily for all Mach numbers from M = 0 to $\mathrm{M} > 1$. The basic principles discussed extend to the three-dimensional case, but more technicalities are involved. The three-dimensional case will be discussed elsewhere.

Acknowledgement

The authors are indebted to R. Agtersloot (Delft Hydraulics) for providing the results of Figure 7. H. Bijl was supported by the Dutch Technology Foundation (STW).

References

1. F. Harlow and J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids* 8 (1965) 2182–2189.
2. J. Ellison, C. Hall and T. Porsching, An unconditionally stable convergent finite difference method for Navier-Stokes problems on curved domains. *SIAM J. Num. Anal.* 24 (1987) 1233–1248.
3. L. Fuchs and H. Zhao, Solution of three-dimensional viscous incompressible flows by a multigrid method. *Int. J. Num. Meth. Fluids* 4 (1987) 539–555.
4. S. Armfield, Ellipticity, accuracy and convergence of the discrete Navier-Stokes equations. *J. Comp. Phys.* 114 (1994) 176–184.
5. A. Burns, I. Jones, J. Kightley and N. Wilkes, The implementation of a finite difference method for predicting incompressible flows in complex geometries. In: C. Taylor, W. Habashi and M. Hafez (eds.), *Num. Meth. in Lam. and Turb. Fl., vol.5*. Swansea: Pineridge Press (1987) pp. 339–350.
6. I. Demirdžić and M. Perić, Finite volume method for prediction of fluid flow in arbitrary shaped domains with moving boundaries. *Int. J. Num. Meth. Fluids* 10 (1990) 771–790.
7. C.-Y. Gu, Computations of flows with large body forces. In: C. Taylor, J. Chin and G. Homsky (eds.), *Numerical Methods in Laminar and Turbulent Flow, Vol. 7, Part 2*. Swansea: Pineridge Press (1991) pp. 1568–1578.
8. Y.-H. Ho and B. Lakshminarayana, Computation of unsteady viscous flow using a pressure-based algorithm. *AIAA J.* 31 (1993) 2232–2240.
9. M. Kobayashi and J. Pereira, Numerical comparison of momentum interpolation methods and pressure-velocity algorithms using nonstaggered grids. *Comm. Appl. Num. Meth.* 7 (1991) 173–186.
10. T. Miller and F. Schmidt, Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a nonstaggered grid system. *Num. Heat Transfer* 14 (1988) 213–233.
11. M. Perić, R. Kessler and G. Scheuerer, Comparison of finite-volume numerical methods with staggered and collocated grids. *Computers and Fluids* 16 (1988) 389–403.
12. C. Rhie and W. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.* 21 (1983) 1525–1532.
13. W. Rodi, S. Majumdar and B. Schönung, Finite volume methods for two-dimensional incompressible flows with complex boundaries. *Comp. Meth. Appl. Mech. Eng.* 75 (1989) 369–392.
14. Y. Zang, R. Street and J. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates. *J. Comp. Phys.* 114 (1994) 18–33.
15. A. Belov, L. Martinelli and A. Jameson, A novel fully implicit multigrid driven algorithm for unsteady incompressible flow calculations. In: S. Wagner, E. Hirschel, J. Périaux and R. Piva (eds.), *Computational Fluid Dynamics '94*. Chichester: Wiley (1994) pp. 662–670.
16. J. Chang and D. Kwak, On the method of pseudo compressibility for numerically solving incompressible flows. AIAA Paper 84-0252 (1984).
17. A. Chorin, A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.* 2 (1967) 12–26.
18. D. Drikakis and M. Schäfer, Comparison between a pressure correction and an artificial compressibility/characteristic based method in parallel incompressible fluid flow computations. In: S. Wagner, E. Hirschel, J. Périaux and R. Piva (eds.), *Computational Fluid Dynamics '94*. Chichester: Wiley (1994) pp. 619–626.
19. P.-M. Hartwich and C.-H. Hsu, High-resolution upwind schemes for the three-dimensional incompressible Navier-Stokes equations. *AIAA J.* 26 (1988) 1321–1328.
20. D. Kwak, J. Chang, S. Shanks and S. Chakravarthy, A three-dimensional incompressible Navier-Stokes flow solver using primitive variables. *AIAA J.* 24 (1986) 390–396.
21. R. Peyret and T. Taylor, *Computational Methods for Fluid Flow*. Berlin: Springer (1985) 358pp.

22. S. Rogers, D. Kwak and C. Kiris, Steady and unsteady solutions of the incompressible Navier-Sokes equations. *AIAA J.* 29, (1991) 603–610.
23. W. Soh and J. Goodrich, Unsteady solution of incompressible Navier-Stokes equations. *J. Comp. Phys.* 79 (1988) 113–134.
24. P. Tamamidis, G. Zhang and D. Assanis, Comparison of pressure-based and artificial compressibility methods for solving 3D steady incompressible flows. *J. Comp. Phys.* 124 (1996) 1–13.
25. L. Davidson and P. Hedberg, Mathematical derivation of a finite volume formulation for laminar flow in complex geometries. *Int. J. Num. Meth. Fluids* 9 (1989) 531–540.
26. T. Ikohagi and B. Shin, Finite difference schemes for steady incompressible Navier-Stokes equations in general curvilinear coordinates. *Computers and Fluids* 19 (1991) 479–488.
27. T. Ikohagi, B. Shin and H. Daiguji, Application of an implicit time-marching scheme to a three-dimensional incompressible flow problem in curvilinear coordinate systems. *Computers and Fluids* 21 (1992) 163–175.
28. S. Koshizuka, Y. Oka and S. Kondo, A staggered differencing technique on boundary-fitted curvilinear grids for incompressible flows along curvilinear or slant walls. *Computational Mechanics* 7 (1990) 123–136.
29. M. Rosenfeld, Validation of numerical simulation of incompressible pulsatile flow in a constricted channel. *Computers and Fluids* 22 (1993) 139–156.
30. M. Rosenfeld and D. Kwak, Time-dependent solution of viscous incompressible flows in moving coordinates. *Int. J. Num. Meth. Fluids* 13 (1991) 1311–1328.
31. M. Rosenfeld, D. Kwak and M. Vinokur, A fractional step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems. *J. Comp. Phys.* 94 (1991) 102–137.
32. W. Shyy and C.-S. Sun, Development of a pressure-correction/staggered-grid based multigrid solver for incompressible recirculating flows. *Computers and Fluids* 22 (1992) 51–76.
33. W. Shyy and T. Vu, On the adoption of velocity variable and grid system for fluid flow computation in curvilinear coordinates. *J. Comp. Phys.* 92 (1991) 82–105.
34. M. Thompson and J. Ferziger, An adaptive multigrid technique for the incompressible Navier-Stokes equations. *J. Comp. Phys.* 82 (1989) 94–121.
35. A. Mynett, P. Wesseling, A. Segal and C. Kassels, The ISNaS incompressible Navier-Stokes solver: invariant discretization. *Applied Scientific Research* 48 (1991) 175–191.
36. C. Oosterlee and P. Wesseling, Multigrid schemes for time-dependent incompressible Navier-Stokes equations. *Impact Comp. Science Eng.* 5 (1993) 153–175.
37. C. Oosterlee and P. Wesseling, A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates. *Impact Comp. Science Eng.* 5 (1993) 128–151.
38. C. Oosterlee and P. Wesseling, Steady incompressible flow around objects in general coordinates with a multigrid solution method. *Num. Meth. Part. Diff. Eqs.* 10 (1994) 295–308.
39. C. Oosterlee, P. Wesseling, A. Segal and E. Brakkee, Benchmark solutions for the incompressible Navier-Stokes equations in general co-ordinates on staggered grids. *Int. J. Num. Meth. Fluids* 17 (1993) 301–321.
40. A. Segal, P. Wesseling, J. van Kan, C. Oosterlee and K. Kassels, Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates. *Int. J. Num. Meth. Fluids* 15 (1992) 411–426.
41. C. Vuik, Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. *Int. J. Num. Meth. Fluids* 16 (1993) 507–523.
42. C. Vuik, Fast iterative solvers for the discretized incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids* 22 (1996) 195–210.
43. P. Wesseling, C. Kassels, C. Oosterlee, A. Segal, C. Vuik, S. Zeng and M. Zijlema, Computing incompressible flows in general domains. In: F.-K. Hebeker, R. Rannacher and G. Wittum (eds.), *Numerical methods for the Navier-Stokes equations.* Braunschweig: Vieweg (1994) pp. 298–314.
44. P. Wesseling, A. Segal, J. van Kan, C. Oosterlee and C. Kassels, Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates on staggered grids. *Comp. Fluid Dynamics Journal* 1 (1992) 27–33.
45. P. Wesseling, P. van Beek and R. van Nooyen, Aspects of non-smoothness in flow computations. In: A. Peters, G. Wittum, B. Herrling, U. Meissner, C. Brebbia, W. Gray and G. Pinder (eds.), *Computational Methods in Water Resources X*. Dordrecht: Kluwer (1994) pp. 1263–1271.
46. M. Zijlema, *Computational modeling of turbulent flows in general domains*. PhD thesis, Delft University of Technology, The Netherlands, April (1996).
47. M. Zijlema, On the construction of a third-order accurate monotone convection scheme with application to turbulent flow in general coordinates. *Int. J. Num. Meth. Fluids* 22 (1996) 619–641.

48. M. Zijlema, A. Segal and P. Wesseling, Finite volume computation of incompressible turbulent flows in general coordinates on staggered grids. *Int. J. Num. Meth. Fluids* 20 (1995) 621–640.
49. M. Zijlema, A. Segal and P. Wesseling, Invariant discretization of the k-ε model in general co-ordinates for prediction of turbulent flow in complicated geometries. *Computers and Fluids* 24 (1995) 209–225.
50. M. Zijlema and P. Wesseling, On accurate discretization of turbulence transport equations in general coordinates. In: C. Taylor and P. Durbetaki (eds.), *Numerical Methods in Laminar and Turbulent Flow, Vol.9, Part 1*. Swansea: Pineridge Press (1995) pp. 34–45.
51. T. Manteuffel and A. White, Jr., The numerical solution of second-order boundary value problems on nonuniform meshes. *Math. Comp.* 47 (1986) 511–535.
52. P.A. Forsyth, Jr. and P. Sammon, Quadratic convergence for cell-centered grids. *Appl. Num. Math.* 4 (1988) 377–394.
53. A. Weiser and M. Wheeler, On convergence of block-centered finite differences for elliptic problems. *SIAM J. Num. Anal.* 25 (1988) 351–375.
54. P. Wesseling, Uniform convergence of discretization error for a singular perturbation problem. *Num. Meth. Part. Diff. Eqs.* 12 (1996) 657–671.
55. P. Farrell, J. Miller, E. O'Riordan and G. Shishkin, A uniformly convergent finite difference scheme for a singularly perturbed semilinear equation. *SIAM J. Num. Anal.* 33 (1996) 1135–1149.
56. H.-G. Roos, M. Stynes and L. Tobiska, *Numerical methods for singularly perturbed differential equations*. Berlin: Springer (1996).
57. M. Gresho and R. Sani, On pressure boundary conditions for the incompressible Navier-Stokes equations. *Int. J. Num. Meth. Fluids* 7 (1987) 1111–1145.
58. P. Gresho, Some current CFD issues relevant to the incompressible Navier-Stokes equations. *Comp. Meth. Appl. Mech. Eng.* 87 (1991) 201–252.
59. P. Gresho, Some interesting issues in incompressible fluid dynamics, both in the continuum and in numerical simulation. *Advances in Applied Mechanics* 28 (1992) 45–140.
60. T. Hou and B. Wetton, Second-order convergence of a projection scheme for the incompressible Navier-Stokes equations with boundaries. *SIAM J. Num. Anal.* 30 (1993) 609–629.
61. J. Bell, P. Colella and H. Glaz, A second-order projection method for the incompressible Navier-Stokes equations. *J. Comp. Phys.* 85 (1989) 257–283.
62. A. Chorin, Numerical solution of the Navier-Stokes equations. *Math. Comp.* 22 (1968) 745–762.
63. A. Chorin, On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comp.* 23 (1969) 342–353.
64. A. Chorin, Numerical solution of incompressible flow problems. In: J. Ortega and W. Rheinbold (eds.), *Studies in Numerical Analysis 2*. Philadelphia: SIAM (1970) pp. 64–71.
65. J. Dukowicz and A. Dvinsky, Approximate factorization as a high order splitting for the implicit incompressible flow equations. *J. Comp. Phys.* 102 (1992) 336–347.
66. J. Blair Perot, An analysis of the fractional step method. *J. Comp. Phys.* 108 (1993) 51–58.
67. R. Temam, *Navier-Stokes equations; theory and numerical analysis*. Amsterdam: North-Holland (1977).
68. J. Van Kan, A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* 7 (1986) 870–891.
69. P. Wesseling, Von Neumann stability conditions for the convection-diffusion equation. *IMA J. Num. Anal.* 16 (1996) 583–598.
70. P. Swarztrauber, Fast Poisson solvers. In: G. Golub (ed.), *Studies in Numerical Analysis*. The Mathematical Society of America (1984) pp. 319–370. Studies in Mathematics Vol. 24.
71. W. Hackbusch, *Multi-grid methods and applications.* Berlin: Springer (1985).
72. P. Wesseling, *An introduction to multigrid methods.* Chichester: Wiley (1992).
73. S. Zeng, C. Vuik and P. Wesseling, Numerical solution of the incompressible Navier-Stokes equations by Krylov subspace and multigrid methods. *Advances in Computational Mathematics* 4 (1995) 27–50.
74. Y. Saad and M. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.* 7 (1986) 856–869.
75. H. Kreiss, J. Lorenz and M. Naughton, Convergence of the solutions of the compressible to the solutions of the incompressible Navier-Stokes equations. *Adv. Appl. Math.* 12 (1991) 187–214.
76. A. Majda, *Compressible fluid flow and systems of conservation laws in several space variables*, vol. 53 of *Applied Mathematical Sciences*. New York: Springer (1984).

77. Y.-H. Choi and C. Merkle, The application of preconditioning in viscous flows. *J. Comp. Phys.* 105 (1993) 207–223.
78. J. Guerra and B. Gustafsson, A numerical method for incompressible and compressible flow problems with smooth solutions. *J. Comp. Phys.* 63 (1986) 377–397.
79. E. Turkel, Review of preconditioning techniques for fluid dynamics. *Appl. Num. Math.* 12 (1993) 257–284.
80. B. van Leer, W.-T. Lee and P. Roe, Characteristic time-stepping or local preconditioning of the Euler equations. AIAA Paper 91-1552 (1991).
81. I. Demirdžić, Z. Lilek and M. Perić, A collocated finite volume method for predicting flows at all speeds. *Int. J. Num. Meth. Fluids* 16 (1993) 1029–1050.
82. K. Karki and S. Patankar, Pressure based calculation procedure for viscous flows at all speed in arbitrary configurations. *AIAA J.* 27 (1989) 1167–1174.
83. W. Shyy, S. Tong and S. Correa, Numerical recirculating flow calculation using a body fitted coordinate system. *Numerical Heat Transfer* 8 (1985) 99–113.
84. W. Shyy and M. Braaten, Adaptive grid computation for inviscid compressible flows using a pressure correction method. AIAA Paper 88-3566-CP (1988).
85. W. Shyy, M.-H. Chen and C.-S. Sun, Pressure-based multigrid algorithm for flow at all speeds. *AIAA J.* 30 (1992) 2660–2669.
86. H. Bijl and P. Wesseling, A numerical method for the computation of compressible flows with low Mach number regions. In: J.-A. Désidéri, C. Hirsch, P. L. Tallec, M. Pandolfi and J. Périaux (eds.), *Computational Fluid Dynamics ’96*. Chichester: Wiley (1996) pp. 206 – 212.
87. S. Eidelman, P. Colella and R. Shreeve, Application of the Godunov method and its second-order extension to cascade flow modelling. *AIAA J.* 22 (1984) 1609–1615.