Journal of Computational Physics **149,** 333–362 (1999)
Article ID jcph.1998.6156, available online at http://www.idealibrary.com on IDEAL®

Computing Flows on General Three-Dimensional Nonsmooth Staggered Grids

P. Wesseling, A. Segal, and C. G. M. Kassels

J.M. Burgers Centre, Delft University of Technology, Faculty of Technical Mathematics and Informatics, Mekelweg 4, 2628 CD Delft, The Netherlands
E-mail: p.wesseling@twi.tudelft.nl

Received September 8, 1998

The classical staggered scheme for the incompressible Navier–Stokes equations is generalized from Cartesian grids to general boundary-fitted structured grids in three dimensions. 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.

Key Words: Navier–Stokes; staggered grids; nonsmooth grids.

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 [33]. In combination with the pressure correction method [3, 6, 13–15, 22, 33, 36, 81, 83] 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.

0021-9991/99 $30.00

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 colocated) 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 [23, 25] (one-sided discretization of div $\boldsymbol{u}$ and grad p); [1, 2, 9, 19, 20, 29, 35, 40, 45, 46, 51, 52, 59, 61, 62, 96] (using the pressure-weighted interpolation method of Rhie and Chow [61]); and [4, 10, 12, 21, 34, 43, 44, 50, 49, 60, 63, 65, 64, 72, 77, 80] (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, make instationary computations more costly and complicated, or make extension to weakly compressible flow difficult; or they are not suitable in the presence of strong body forces [29]. 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 [16, 17, 37, 38, 41, 67–69, 7, 76], and by our group [5, 8, 53–57, 71, 87, 92–95, 98–103].

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 [67–69]. We expect this method to behave satisfactory on nonsmooth grids, although this is not shown in [67–69]. 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.

The methods using staggered grids in general coordinates proposed in the other publications mentioned above are likely to suffer from inaccuracy when the grid is nonsmooth. In [17, 41] and our own earlier work a coordinate-invariant form of the governing equations is discretized, with explicit use of Christoffel symbols, making it necessary that the coordinate mapping is twice continuously differentiable. Also in [37, 38] a set of invariant equations is discretized, but Christoffel symbols are avoided by the introduction in the viscous terms of the vorticity as an auxiliary variable. Nevertheless, second derivatives of the mapping still occur. Furthermore, the viscosity needs to be constant, precluding application to turbulent flows. In [16] finite volume integrals of an invariant formulation are simplified by assuming

the base vectors to be locally constant; this is likely to be inaccurate when the coordinate mapping is not smooth. In [7, 76] the scheme is not coordinate-invariant, and Christoffel symbols do not occur. Cartesian velocity components are used in the momentum equations and contravariant components are employed in the continuity equation. This method would seem to require less grid smoothness than the methods discussed before. The method is not directly applicable to nonstationary flows, because of approximations made in the pressure correction equation. Objections have been raised against the use of staggered Cartesian velocity components, because if the gridlines turn over an angle of 90°, as in a 90° bend for example, these components are parallel to the control volume faces instead of perpendicular, as in the original MAC scheme of [33]. But in [76] it is shown that the method continues to work well under these circumstances.

In the present work a time-accurate coordinate-invariant staggered scheme is presented for which the mapping merely needs to be piecewise differentiable, allowing abrupt changes of mesh size. 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 [48]. This has made some believe that grids need to be smooth for accurate results. But this is not so. In [58, 48, 88, 90] 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 [24, 66, 90] it is shown that the accuracy is uniform in the Reynolds number, for the convection-diffusion equation.

We will start by discussing as far as necessary geometric aspects of coordinate transformations. Next, a staggered discretization will be presented for the incompressible Navier–Stokes equations, which is accurate on general nonuniform grids. Finally, numerical experiments will be presented.

2. CELL VOLUMES AND CELL FACE AREAS IN BOUNDARY-FITTED GRIDS

Let the physical domain Ω be topologically equivalent to the unit cube G. In $\bar{G}$ we have Cartesian coordinates $\boldsymbol{\xi} = (\xi^1, \xi^2, \xi^3)$ and a uniform grid G_h consisting of grid points located at $\boldsymbol{\xi}_j$, $j = (j_1, j_2, j_3)$,

$$G_h = \{\boldsymbol{\xi}_j : \xi_j^\alpha = j_\alpha \Delta\xi^\alpha,\ j_\alpha = 0, 1, \ldots, 1/\Delta\xi^\alpha, \alpha = 1, 2, 3\}, \tag{2.1}$$

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

It is assumed that a boundary-fitted coordinate system is generated numerically, giving a one-to-one mapping

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

In order to obtain accurate discretizations we have to be precise about how the mapping (2.2) is extended to all of $\bar{\Omega}$ and $\bar{G}$. In order to allow rough grids for reasons given in the

preceding section, the mapping (2.2) is extended by trilinear interpolation. Let Ω_j be the grid cell with vertices

$$\begin{gathered}\boldsymbol{x}_{j\pm(e_1+e_2+e_3)}, \qquad \boldsymbol{x}_{j\pm(e_1-e_2+e_3)}, \qquad \boldsymbol{x}_{j\pm(e_1+e_2-e_3)}, \qquad \boldsymbol{x}_{j\pm(-e_1+e_2+e_3)}, \\ e_1 = \left(\frac{1}{2}, 0, 0\right), e_2 = \left(0, \frac{1}{2}, 0\right), e_3 = \left(0, 0, \frac{1}{2}\right).\end{gathered} \tag{2.3}$$

Since the vertices have integer indices, this implies that j_α is fractional. In the following, $\boldsymbol{x}_j$ will be a cell center, $\boldsymbol{x}_{j+e_\alpha}$ will be a cell face center, $x_{j+e_\alpha+e_\beta}$ $(\beta \neq \alpha)$ will be a cell edge center, and $\boldsymbol{x}_{j+e_1+e_2+e_3}$ will be a cell vertex. The image of Ω_j in G is a rectangular hexahedron called G_j; G_j and Ω_j are called cells.

Let $\boldsymbol{\xi}_0$ be some point in G_j. Then trilinear interpolation inside Ω_j gives the following relation between $\boldsymbol{x}$ and $\boldsymbol{\xi}$:

$$\begin{aligned}\boldsymbol{x} = \boldsymbol{x}_0 &+ \boldsymbol{c}_\alpha\left(\xi^\alpha - \xi_0^\alpha\right) + \boldsymbol{c}_{12}\left(\xi^1 - \xi_0^1\right)\left(\xi^2 - \xi_0^2\right) + \boldsymbol{c}_{23}\left(\xi^2 - \xi_0^2\right)\left(\xi^3 - \xi_0^3\right) \\ &+ \boldsymbol{c}_{13}\left(\xi^3 - \xi_0^3\right)\left(\xi^1 - \xi_0^1\right) + \boldsymbol{c}_{123}\left(\xi^1 - \xi_0^1\right)\left(\xi^2 - \xi_0^2\right)\left(\xi^3 - \xi_0^3\right).\end{aligned} \tag{2.4}$$

The coefficients $\boldsymbol{x}_0$ and $\boldsymbol{c}$ follow from the requirement that (2.2) holds in the vertices of G_j and need not be determined here. They differ per cell. In this way we obtain a piecewise trilinear mapping

$$\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi}), \qquad \boldsymbol{x} \in \bar{\Omega}, \boldsymbol{\xi} \in \bar{G}. \tag{2.5}$$

This mapping is assumed to be boundary-fitted, which means that the boundary $\partial\Omega$ is the image of ∂G. In other words, on every part of $\partial\Omega$ we have $\xi^\alpha = \text{constant}$ for some α.

It follows from (2.4) that the edges of the cell Ω_j are straight. Consider a face of Ω_j with $\xi^3 = \text{constant}$. In this face the mapping (2.4) becomes, choosing $\boldsymbol{\xi}_0$ in the corresponding face of G_j so that we have $\xi^3 = \xi_0^3$,

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{c}_1\left(\xi^1 - \xi_0^1\right) + \boldsymbol{c}_2\left(\xi^2 - \xi_0^2\right) + \boldsymbol{c}_{12}\left(\xi^1 - \xi_0^1\right)\left(\xi^2 - \xi_0^2\right), \tag{2.6}$$

with coefficients $\boldsymbol{x}_0$ and $\boldsymbol{c}$ in general different from those in (2.4), since $\boldsymbol{\xi}_0$ is changed. With $\xi^1 = \text{constant}$ or $\xi^2 = \text{constant}$, Eq. (2.6) describes straight lines, so that the cell face contains two families of straight lines and is therefore a doubly ruled surface. This means that $\partial\Omega$ is approximated by doubly ruled patches.

Taking $\xi^\alpha = \text{constant}$ and $\xi^\beta = \text{constant}$, $\alpha \neq \beta$, $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ gives us a curvilinear $\boldsymbol{\xi}$-coordinate system in Ω. It is assumed that $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ is one-to-one, i.e., we have for the Jacobian

$$J = \left\{\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}}\right\} = \frac{\partial \boldsymbol{x}}{\partial \xi^1} \cdot \left(\frac{\partial \boldsymbol{x}}{\partial \xi^2} \times \frac{\partial \boldsymbol{x}}{\partial \xi^3}\right) \neq 0. \tag{2.7}$$

Furthermore, it is assumed that the $\boldsymbol{\xi}$-coordinate system is right-handed, i.e.,

$$J > 0. \tag{2.8}$$

Figure 2.1 gives a picture of a (in this case piecewise bilinear) boundary-fitted coordinate mapping in two dimensions, showing the piecewise linear $\boldsymbol{\xi}$-coordinates. We will not discuss the two-dimensional case; it can be easily derived from the three-dimensional case.

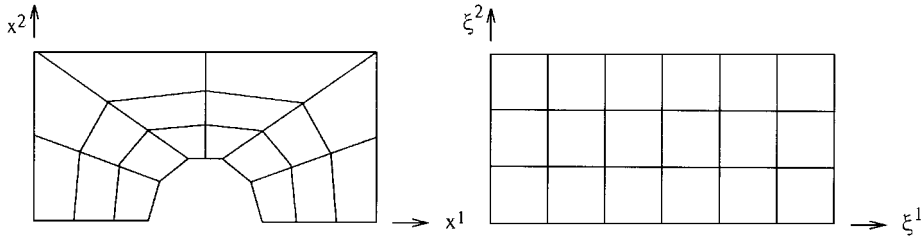

FIG. 2.1. Mapping of physical domain onto computational domain.

We will now discuss accurate and efficient computation of cell volumes and cell face areas and normals. This is of obvious importance for finite volume discretization, whether staggered or colocated. This elementary topic has of course been discussed before; see the excellent review of discretization in physical space by Vinokur [85] and references quoted there. But the topic still warrants attention, because many authors define a cell face to consist of two plane triangles. This is less satisfactory than the doubly ruled surface used here. The resulting formulae are not more economical, and a face can be divided into triangles in two ways, and adjacent cells have to fit, which complicates grid generation; furthermore, the normal is ill-defined. Therefore we feel that the approach in which the mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ is extended from the vertices to the whole domain by multilinear interpolation, or equivalently, in which cell faces are defined as doubly ruled surfaces, is to be preferred; the more so because the resulting expressions for areas and volumes are cheap, as will be seen.

We start with the cell volume. Figure 2.2 depicts a three-dimensional cell in $\boldsymbol{x}$- and $\boldsymbol{\xi}$-space, with numbered vertices. We have for the cell volume

$$|\Omega_j| = \int_{\Omega_j} d\Omega = \int_{\Omega_j} J \, d\xi^1 \, d\xi^2 \, d\xi^3. \tag{2.9}$$

Let us choose $\boldsymbol{\xi}_0$ in (2.4) in the center of the cell in G. Then the ranges of integration in this triple integral are $(\xi_0^\alpha - \frac{1}{2}\Delta\xi^\alpha, \xi_0^\alpha + \frac{1}{2}\Delta\xi^\alpha)$, $\alpha = 1, 2, 3$, with $\Delta\xi^\alpha$ the length of the edges in G. The following change of variables is convenient,

$$\xi^\alpha = \xi_0^\alpha + \frac{1}{2}\Delta\xi^\alpha s^\alpha \qquad \text{(no summation)}, \tag{2.10}$$

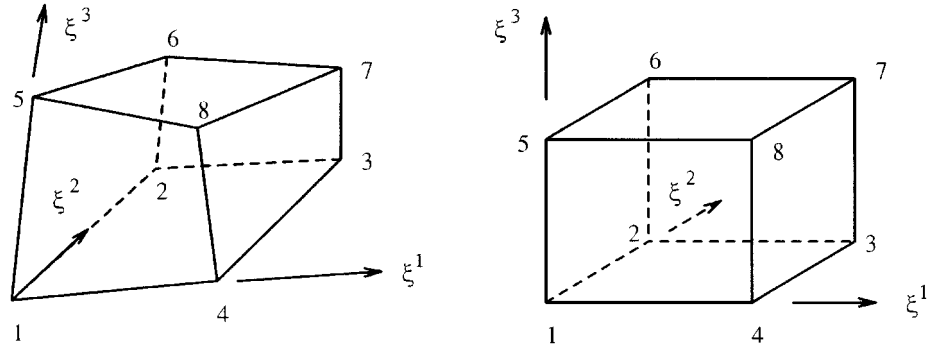

FIG. 2.2. Three-dimensional cell. Its image in G is a rectangular hexahedron.

so that (2.9) becomes

$$|\Omega_j| = \frac{1}{8}\Delta\xi^1\Delta\xi^2\Delta\xi^3 \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} J\, ds^1\, ds^2\, ds^3. \tag{2.11}$$

According to (2.4) we have

$$\begin{aligned}
\frac{\partial \boldsymbol{x}}{\partial \xi^1} &= \boldsymbol{c}_1 + \boldsymbol{d}_{12}s^2 + \boldsymbol{d}_{13}s^3 + \boldsymbol{d}_{123}s^2s^3,\\
\frac{\partial \boldsymbol{x}}{\partial \xi^2} &= \boldsymbol{c}_2 + \boldsymbol{d}_{21}s^1 + \boldsymbol{d}_{23}s^3 + \boldsymbol{d}_{231}s^3s^1,\\
\frac{\partial \boldsymbol{x}}{\partial \xi^3} &= \boldsymbol{c}_3 + \boldsymbol{d}_{31}s^1 + \boldsymbol{d}_{32}s^2 + \boldsymbol{d}_{312}s^1s^2,
\end{aligned} \tag{2.12}$$

where

$$\begin{aligned}
\boldsymbol{d}_{12} &= \frac{1}{2}\Delta\xi^2\boldsymbol{c}_{12}, \qquad \boldsymbol{d}_{13} = \frac{1}{2}\Delta\xi^3\boldsymbol{c}_{13}, \qquad \boldsymbol{d}_{123} = \frac{1}{4}\Delta\xi^2\Delta\xi^3\boldsymbol{c}_{123},\\
\boldsymbol{d}_{21} &= \frac{1}{2}\Delta\xi^1\boldsymbol{c}_{12}, \qquad \boldsymbol{d}_{23} = \frac{1}{2}\Delta\xi^3\boldsymbol{c}_{23}, \qquad \boldsymbol{d}_{231} = \frac{1}{4}\Delta\xi^3\Delta\xi^1\boldsymbol{c}_{123},\\
\boldsymbol{d}_{31} &= \frac{1}{2}\Delta\xi^1\boldsymbol{c}_{13}, \qquad \boldsymbol{d}_{32} = \frac{1}{2}\Delta\xi^2\boldsymbol{c}_{23}, \qquad \boldsymbol{d}_{312} = \frac{1}{4}\Delta\xi^1\Delta\xi^2\boldsymbol{c}_{123}.
\end{aligned} \tag{2.13}$$

Because of symmetry (which is why $\boldsymbol{\xi}_0$ was chosen in the center), terms that contain odd powers of s^α do not contribute. Neglecting these terms, Eq. (2.7) gives

$$\begin{aligned}
J \sim\ & \boldsymbol{c}_1\cdot(\boldsymbol{c}_2\times\boldsymbol{c}_3) + s^1s^1\boldsymbol{c}_1\cdot(\boldsymbol{d}_{21}\times\boldsymbol{d}_{31}) + s^2s^2\boldsymbol{d}_{12}\cdot(\boldsymbol{c}_2\times\boldsymbol{d}_{32})\\
& + s^3s^3\boldsymbol{d}_{13}\cdot(\boldsymbol{d}_{23}\times\boldsymbol{c}_3) + s^1s^1s^2s^2s^3s^3\boldsymbol{d}_{123}\cdot(\boldsymbol{d}_{231}\times\boldsymbol{d}_{312}),
\end{aligned} \tag{2.14}$$

where the symbol $\sim$ indicates that terms have been neglected. Because the $\boldsymbol{d}_{\alpha\beta\gamma}$ vectors are parallel, the last term is zero. Integration gives, using (2.13),

$$\begin{aligned}
|\Omega_j| = \Delta\xi^1\Delta\xi^2\Delta\xi^3\Big\{ & \boldsymbol{c}_1\cdot(\boldsymbol{c}_2\times\boldsymbol{c}_3) + \frac{2}{3}\Delta\xi^1\Delta\xi^1\boldsymbol{c}_1\cdot(\boldsymbol{c}_{12}\times\boldsymbol{c}_{13})\\
& + \frac{2}{3}\Delta\xi^2\Delta\xi^2\boldsymbol{c}_{12}\cdot(\boldsymbol{c}_2\times\boldsymbol{c}_{23}) + \frac{2}{3}\Delta\xi^3\Delta\xi^3\boldsymbol{c}_{13}\cdot(\boldsymbol{c}_{23}\times\boldsymbol{c}_3)\Big\}.
\end{aligned} \tag{2.15}$$

The mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$ is given in the cell vertices only. We want to rewrite (2.15) in terms of this information. Expressed in the local $\boldsymbol{s}$-coordinates, defined in (2.10), we have

$$\boldsymbol{x} = \boldsymbol{x}_0 + \boldsymbol{b}_\alpha s^\alpha + \boldsymbol{b}_{12}s^1s^2 + \boldsymbol{b}_{13}s^1s^3 + \boldsymbol{b}_{23}s^2s^3 + \boldsymbol{b}_{123}s^1s^2s^3, \tag{2.16}$$

with (no summation)

$$\begin{aligned}
\boldsymbol{b}_\alpha &= \frac{1}{2}\Delta\xi^\alpha\boldsymbol{c}_\alpha, \qquad \boldsymbol{b}_{\alpha\beta} = \frac{1}{4}\Delta\xi^\alpha\Delta\xi^\beta\boldsymbol{c}_{\alpha\beta},\\
\boldsymbol{b}_{123} &= \frac{1}{8}\Delta\xi^1\Delta\xi^2\Delta\xi^3\boldsymbol{c}_{123}.
\end{aligned} \tag{2.17}$$

Requiring $\boldsymbol{x} = \boldsymbol{x}_m$, $m = 1, 2, \ldots, 8$ (see Fig. 2.2) gives the following system of equations,

$$Ay = r, \tag{2.18}$$

with $y = (\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3, \boldsymbol{b}_{12}, \boldsymbol{b}_{13}, \boldsymbol{b}_{23}, \boldsymbol{b}_{123}, \boldsymbol{x}_0)^T$, $r = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_8)^T$, and

$$A = \begin{pmatrix} -1 & -1 & -1 & 1 & 1 & 1 & -1 & 1 \\ -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 \\ 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix}. \tag{2.19}$$

The columns of A are orthogonal. Premultiplication of (2.18) by A^T gives

$$y = \frac{1}{8} A^T r \tag{2.20}$$

so that

$$\begin{aligned} \boldsymbol{b}_1 &= \frac{1}{8}(\boldsymbol{x}_{3478} - \boldsymbol{x}_{1256}), & \boldsymbol{b}_2 &= \frac{1}{8}(\boldsymbol{x}_{2367} - \boldsymbol{x}_{1458}), \\ \boldsymbol{b}_3 &= \frac{1}{8}(\boldsymbol{x}_{5678} - \boldsymbol{x}_{1234}), & \boldsymbol{b}_{12} &= \frac{1}{8}(\boldsymbol{x}_{1357} - \boldsymbol{x}_{2468}), \\ \boldsymbol{b}_{13} &= \frac{1}{8}(\boldsymbol{x}_{1278} - \boldsymbol{x}_{3456}), & \boldsymbol{b}_{23} &= \frac{1}{8}(\boldsymbol{x}_{1467} - \boldsymbol{x}_{2358}), \\ \boldsymbol{b}_{123} &= \frac{1}{8}(\boldsymbol{x}_{2457} - \boldsymbol{x}_{1368}), \end{aligned} \tag{2.21}$$

where

$$\boldsymbol{x}_{jkmn} \equiv \boldsymbol{x}_j + \boldsymbol{x}_k + \boldsymbol{x}_m + \boldsymbol{x}_n. \tag{2.22}$$

Substitution of (2.17) in (2.15) gives

$$|\Omega_j| = 8\boldsymbol{b}_1 \cdot (\boldsymbol{b}_2 \times \boldsymbol{b}_3) + \frac{8}{3}\{\boldsymbol{b}_1 \cdot (\boldsymbol{b}_{12} \times \boldsymbol{b}_{13}) + \boldsymbol{b}_{12} \cdot (\boldsymbol{b}_2 \times \boldsymbol{b}_{23}) + \boldsymbol{b}_{13} \cdot (\boldsymbol{b}_{23} \times \boldsymbol{b}_3)\}. \tag{2.23}$$

This formula is exact. Before giving a more efficient expression we first give a formula for the area of a cell face multiplied by its outward unit normal.

Consider the cell face 1234 (cf. Fig. 2.2). In this cell face the coordinate mapping can be written in the form given by (2.6). The vector normal to the face in a point $\boldsymbol{x}$ in the positive ξ^3-direction with length equal to the area of the parallelogram spanned by two infinitesimal displacement vectors $d\boldsymbol{x}_{(1)}$ and $d\boldsymbol{x}_{(2)}$ is given by $d\boldsymbol{x}_{(1)} \times d\boldsymbol{x}_{(2)}$, so that the average normal vector with length equal to the area of the face 1432 (which we call the cell face vector) is given by

$$\boldsymbol{s}_{1432} = \int_{1432} \boldsymbol{n}\, d\Gamma = \int_{1432} d\boldsymbol{x}_{(1)} \times d\boldsymbol{x}_{(2)}. \tag{2.24}$$

The indices are put in parentheses to emphasize that no components are intended. Introducing s^α defined by (2.10) gives

$$d\boldsymbol{x}_{(\alpha)} = \frac{\partial \boldsymbol{x}}{\partial \xi^\alpha}\, d\xi^\alpha \qquad \text{(no summation)}, \tag{2.25}$$

with

$$\frac{\partial \boldsymbol{x}}{\partial \xi^1} = \boldsymbol{c}_1 + \boldsymbol{d}_{12} s^2, \qquad \frac{\partial \boldsymbol{x}}{\partial \xi^2} = \boldsymbol{c}_2 + \boldsymbol{d}_{21} s^1, \tag{2.26}$$

where $\boldsymbol{d}_{12}$ and $\boldsymbol{d}_{21}$ are defined by (2.13). Substitution in (2.24) gives

$$\boldsymbol{s}_{1432} = \frac{1}{4}\Delta\xi^1\Delta\xi^2 \int_{-1}^{1}\int_{-1}^{1} \frac{\partial \boldsymbol{x}}{\partial \xi^1} \times \frac{\partial \boldsymbol{x}}{\partial \xi^2}\, ds^1\, ds^2 = \Delta\xi^1\Delta\xi^2 \boldsymbol{c}_1 \times \boldsymbol{c}_2. \tag{2.27}$$

The vectors $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are easily determined as follows. The transformation (2.6) can be rewritten as

$$\boldsymbol{x} = \boldsymbol{x}_0 + \frac{1}{2}\Delta\xi^1 \boldsymbol{c}_1 s^1 + \frac{1}{2}\Delta\xi^2 \boldsymbol{c}_2 s^2 + \frac{1}{4}\Delta\xi^1\Delta\xi^2 \boldsymbol{c}_{12} s^1 s^2. \tag{2.28}$$

Requiring $\boldsymbol{x} = \boldsymbol{x}_m$, $m = 1, 2, 3, 4$, gives a system of equations that is easily solved in the same way as (2.18). The result is

$$\boldsymbol{c}_1 = \frac{1}{2\Delta\xi^1}(\boldsymbol{x}_{34} - \boldsymbol{x}_{12}), \qquad \boldsymbol{c}_2 = \frac{1}{2\Delta\xi^2}(\boldsymbol{x}_{23} - \boldsymbol{x}_{14}), \tag{2.29}$$

where $\boldsymbol{x}_{jm} \equiv \boldsymbol{x}_j + \boldsymbol{x}_m$. Using $\boldsymbol{a} \times \boldsymbol{a} = 0$ this gives the efficient form

$$\boldsymbol{s}_{1432} = \frac{1}{2}(\boldsymbol{x}_4 - \boldsymbol{x}_2) \times (\boldsymbol{x}_3 - \boldsymbol{x}_1), \tag{2.30}$$

which is identical to the formula for a plane quadrilateral. This equivalence holds only for doubly ruled surfaces with straight edges. The general formula for a cell face vector is

$$\boldsymbol{s}_{mnpq} = \frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{x}_q) \times (\boldsymbol{x}_p - \boldsymbol{x}_m) \tag{2.31}$$

assuming the cell vertices are numbered such that the diagonals are mp and nq and that the vector has a positive component in a direction of increasing ξ^α (cf. Fig. 2.2).

It may be shown that

$$\boldsymbol{s}_{mnpq} // \boldsymbol{n}_0, \tag{2.32}$$

with $\boldsymbol{n}_0$ the normal in the cell face center $\boldsymbol{x}_0$. It is easily seen from (2.6) that

$$\boldsymbol{x}_0 = \frac{1}{4}(\boldsymbol{x}_m + \boldsymbol{x}_n + \boldsymbol{x}_p + \boldsymbol{x}_q). \tag{2.33}$$

Let Γ_j be the surface of the cell Ω_j. Since Γ_j is closed we must have

$$\int_{\Gamma_j} \boldsymbol{n}\, d\Gamma = 0, \tag{2.34}$$

with $\boldsymbol{n}$ the unit outward normal. This is called the geometric identity. For uniform flow to be an exact solution of discretization schemes it is necessary that the geometric identity is satisfied exactly. Since our cell faces are smooth $\boldsymbol{n}$ is defined unambiguously, and it is easily shown that

$$\int_{\Gamma_j} \boldsymbol{n}\, d\Gamma = \sum_{\Gamma_j} \boldsymbol{s}_{mnpq} = 0, \tag{2.35}$$

where summation takes place over the cell faces constituting Γ_j. We see that the geometric identity is satisfied.

We can now write down a more economical formula for the cell volume. The cell face vectors are needed in the discretization. It would be nice if they could be reused to compute cell face volumes. It would not be unexpected if the volume is related to face area times face distance. The vector $2\boldsymbol{b}_\alpha$ (defined in (2.21)) connects face centers. We rewrite (2.23) as

$$\begin{aligned} |\Omega_j| = \frac{8}{3}\{&\boldsymbol{b}_1 \cdot (\boldsymbol{b}_2 \times \boldsymbol{b}_3 + \boldsymbol{b}_{12} \times \boldsymbol{b}_{13}) + \boldsymbol{b}_2 \cdot (\boldsymbol{b}_3 \times \boldsymbol{b}_1 + \boldsymbol{b}_{23} \times \boldsymbol{b}_{12}) \\ &+ \boldsymbol{b}_3 \cdot (\boldsymbol{b}_1 \times \boldsymbol{b}_2 + \boldsymbol{b}_{13} \times \boldsymbol{b}_{23})\}. \end{aligned} \tag{2.36}$$

After some manipulation we find

$$\boldsymbol{b}_2 \times \boldsymbol{b}_3 = \frac{1}{32}(\boldsymbol{x}_{23} \times \boldsymbol{x}_{67} + \boldsymbol{x}_{67} \times \boldsymbol{x}_{58} + \boldsymbol{x}_{14} \times \boldsymbol{x}_{23} + \boldsymbol{x}_{58} \times \boldsymbol{x}_{14}), \tag{2.37}$$

and

$$\boldsymbol{b}_{12} \times \boldsymbol{b}_{13} = \frac{1}{32}(\boldsymbol{x}_{17} \times \boldsymbol{x}_{28} + \boldsymbol{x}_{35} \times \boldsymbol{x}_{17} + \boldsymbol{x}_{28} \times \boldsymbol{x}_{46} + \boldsymbol{x}_{46} \times \boldsymbol{x}_{35}), \tag{2.38}$$

so that

$$\boldsymbol{b}_1 \cdot (\boldsymbol{b}_2 \times \boldsymbol{b}_3 + \boldsymbol{b}_{12} \times \boldsymbol{b}_{13}) = \frac{1}{8}\boldsymbol{b}_1 \cdot (\boldsymbol{s}_{1265} + \boldsymbol{s}_{4378}). \tag{2.39}$$

Not unexpectedly, this is the average area of two opposing faces times their distance divided by 8. The other two terms in (2.36) can be handled in the same way, resulting in the efficient formula

$$|\Omega_j| = \frac{1}{3}\{\boldsymbol{b}_1 \cdot (\boldsymbol{s}_{1265} + \boldsymbol{s}_{4378}) + \boldsymbol{b}_2 \cdot (\boldsymbol{s}_{1584} + \boldsymbol{s}_{2673}) + \boldsymbol{b}_3 \cdot (\boldsymbol{s}_{1432} + \boldsymbol{s}_{8765})\}. \tag{2.40}$$

Since $\boldsymbol{s}_{mnpq}$ is needed anyway, as noted before, this is the most efficient formula for the cell volume. Equation (2.40) consists of parts that can be reused for adjacent volumes. For example, $\boldsymbol{b}_1 = \frac{1}{8}(\boldsymbol{x}_{3478} - \boldsymbol{x}_{1256})$, and $\boldsymbol{x}_{3478}$ can be used both in Ω_j and Ω_{j+2e_1}. The same holds for $\boldsymbol{s}_{4378}$, and the other vectors in (2.40) can also be used more than once. We have not seen (2.40) published elsewhere.

3. GEOMETRIC QUANTITIES AND THEIR SMOOTHNESS PROPERTIES

For the reasons given in Section 1 we will work in the transformed domain G to derive a staggered discretization. Physical laws in general coordinates are best formulated in coordinate-invariant form using tensor analysis. As a preparation for this we introduce a few fundamental geometric quantities related to the coordinate mapping $\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{\xi})$ introduced in Section 2, and take a careful look at their smoothness properties. The summation convention applies.

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

$$\boldsymbol{a}_{(\alpha)}=\frac{\partial \boldsymbol{x}}{\partial \xi^{\alpha}} \qquad \text{or} \qquad a^{\beta}_{(\alpha)}=\frac{\partial x^{\beta}}{\partial \xi^{\alpha}}, \tag{3.1}$$

with $a^{\beta}_{(\alpha)}$ the Cartesian x^{β}-component of the vector $\boldsymbol{a}_{(\alpha)}$. Because $\boldsymbol{x}=\boldsymbol{x}(\boldsymbol{\xi})$ is piecewise multilinear, $\boldsymbol{a}_{(\alpha)}$ is piecewise continuous and does not exist everywhere. For $\boldsymbol{a}_{(\alpha)}$ we have Eq. (2.12). We will need $\boldsymbol{a}_{(\alpha)}$ only in cell centers and cell face centers. For mnemonic convenience we indicate these points by subscripts C, N, R, F, etc., with C for center, N for north, R for rear, F for front, etc. Hence, with the vertex numbering of Fig. 2.2,

$$\boldsymbol{x}_C=\frac{1}{8}\sum_{1}^{8}\boldsymbol{x}_m, \qquad \boldsymbol{x}_F=\frac{1}{4}(\boldsymbol{x}_1+\boldsymbol{x}_4+\boldsymbol{x}_5+\boldsymbol{x}_8), \tag{3.2}$$

etc. It follows from (2.33) that $\boldsymbol{x}_N, \ldots$ are in the doubly ruled cell faces. For economy we need to express $\boldsymbol{a}_{(\alpha)}$ in the points $C, N, \ldots$ in terms of cell vertex locations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_8$ in an efficient way. In C we have in (2.16), $s^1=s^2=s^3=0$; in N we have $(s^1, s^2, s^3)=(0, 0, 1)$, etc. After some algebra we find from (2.12), (2.13), (2.17), and (2.21)

$$\begin{aligned}
&\boldsymbol{a}_{(1)C}=(\boldsymbol{x}_E-\boldsymbol{x}_W)/\Delta\xi^1, \qquad \boldsymbol{a}_{(2)C}=(\boldsymbol{x}_R-\boldsymbol{x}_F)/\Delta\xi^2, \qquad \boldsymbol{a}_{(3)C}=(\boldsymbol{x}_N-\boldsymbol{x}_S)/\Delta\xi^3,\\
&\boldsymbol{a}_{(2)W}=\frac{1}{2\Delta\xi^2}(\boldsymbol{x}_{26}-\boldsymbol{x}_{15}), \qquad \boldsymbol{a}_{(3)W}=\frac{1}{2\Delta\xi^3}(\boldsymbol{x}_{56}-\boldsymbol{x}_{12}),\\
&\boldsymbol{a}_{(1)F}=\frac{1}{2\Delta\xi^1}(\boldsymbol{x}_{48}-\boldsymbol{x}_{15}), \qquad \boldsymbol{a}_{(3)F}=\frac{1}{2\Delta\xi^3}(\boldsymbol{x}_{58}-\boldsymbol{x}_{14}),\\
&\boldsymbol{a}_{(1)S}=\frac{1}{2\Delta\xi^1}(\boldsymbol{x}_{34}-\boldsymbol{x}_{12}), \qquad \boldsymbol{a}_{(2)S}=\frac{1}{2\Delta\xi^2}(\boldsymbol{x}_{23}-\boldsymbol{x}_{14}),
\end{aligned} \tag{3.3}$$

etc., where we have used the notation $\boldsymbol{x}_{mn}=\boldsymbol{x}_m+\boldsymbol{x}_n$. At a cell face $\xi^{\alpha}=\text{constant}$ $\boldsymbol{a}_{(\alpha)}$ is discontinuous and will not be used, but $\boldsymbol{a}_{(\beta)}$, $\beta\neq\alpha$ is continuous at such a face.

The contravariant base vectors are defined by

$$\boldsymbol{a}^{(\alpha)}=\nabla\xi^{\alpha} \qquad \text{or} \qquad a^{(\alpha)}_{\beta}=\frac{\partial \xi^{\alpha}}{\partial x^{\beta}}. \tag{3.4}$$

We have

$$\boldsymbol{a}^{(\alpha)}\cdot\boldsymbol{a}_{(\beta)}=\delta^{\alpha}_{\beta}, \tag{3.5}$$

with δ^{α}_{β} the Kronecker delta. Solving (3.5) gives

$$\boldsymbol{a}^{(\alpha)}=\frac{1}{\sqrt{g}}\left(\boldsymbol{a}_{(\beta)}\times\boldsymbol{a}_{(\gamma)}\right), \qquad \alpha, \beta, \gamma \text{ cyclic}, \tag{3.6}$$

where $\sqrt{g}$ is the common designation in tensor analysis for the Jacobian J defined in (2.7), i.e.,

$$\sqrt{g} = \boldsymbol{a}_{(\alpha)} \cdot \big(\boldsymbol{a}_{(\beta)} \times \boldsymbol{a}_{(\gamma)}\big), \qquad \alpha, \beta, \gamma \text{ cyclic.} \tag{3.7}$$

From the smoothness properties of $\boldsymbol{a}_{(\alpha)}$ it follows that $\sqrt{g}$ and $\boldsymbol{a}^{(\alpha)}$ are discontinuous at cell faces. But $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ turns out to be continuous at cell faces of the type $\xi^\alpha = \text{constant}$. After some algebra we find that in the cell face centers where $\sqrt{g}\boldsymbol{a}^\alpha$ is continuous, $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ equals the corresponding cell face vector times a scaling factor:

$$\begin{gathered}\big(\sqrt{g}\boldsymbol{a}^{(1)}\big)_{W,E} = \frac{1}{\Delta\xi^2\Delta\xi^3}\boldsymbol{s}_{W,E}, \qquad \big(\sqrt{g}\boldsymbol{a}^{(2)}\big)_{R,F} = \frac{1}{\Delta\xi^1\Delta\xi^3}\boldsymbol{s}_{R,F}, \\ \big(\sqrt{g}\boldsymbol{a}^{(3)}\big)_{N,S} = \frac{1}{\Delta\xi^1\Delta\xi^2}\boldsymbol{s}_{N,S}.\end{gathered} \tag{3.8}$$

4. STAGGERED REPRESENTATION OF THE VELOCITY FIELD

We want to generalize the classical staggered Marker-and-Cell (MAC) scheme proposed by Harlow and Welch [33] from Cartesian to general coordinates. This means that we wish to compute the pressure in cell centers, and normal velocity components in cell face centers. It follows from (3.8) that the contravariant base vector $\boldsymbol{a}_{(\alpha)}$ is perpendicular to faces of the type $\xi^\alpha = \text{constant}$, so at first sight the contravariant velocity components U^α, defined as

$$U^\alpha = \boldsymbol{u} \cdot \boldsymbol{a}^{(\alpha)} \tag{4.1}$$

would seem to be suitable for representing the velocity field $\boldsymbol{u}$. But we saw in the preceding section that $\boldsymbol{a}^{(\alpha)}$ is discontinuous at cell faces. As a consequence, the use of U^α leads to bad accuracy on rough grids. But as we saw $\sqrt{g}\boldsymbol{a}^{(\alpha)}$ is continuous at cell faces where $\xi^\alpha = \text{constant}$. Therefore the following coordinate-invariant staggered representation of the velocity field $\boldsymbol{u}$ will be employed (from now on denoting $\boldsymbol{x}_E$ by $\boldsymbol{x}_{j+e_1}$, etc.):

$$V^\alpha_{j+e_\alpha} = \big(\sqrt{g}\boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u}\big)_{j+e_\alpha} \qquad \text{(no summation)}. \tag{4.2}$$

From (3.8) it follows that we may also write

$$V^\alpha_{j+e_\alpha} = (\boldsymbol{s} \cdot \boldsymbol{u})_{j+e_\alpha} \big/ \Delta\xi^\beta\Delta\xi^\gamma, \qquad \alpha, \beta, \gamma \text{ cyclic.} \tag{4.3}$$

This shows that $V^\alpha\Delta\xi^\beta\Delta\xi^\gamma$ approximates the volume flux through the cell face. Therefore V^α will be called the volume flux components. Comparing (4.2) and (4.3) and using (2.30) we have the efficient formula (cf. Fig. 2.2)

$$\big(\sqrt{g}\boldsymbol{a}^{(1)}\big)_{j+e_1} = \frac{1}{2}(\boldsymbol{x}_7 - \boldsymbol{x}_4) \times (\boldsymbol{x}_8 - \boldsymbol{x}_9)/\Delta\xi^2\Delta\xi^3, \tag{4.4}$$

etc.

We will need to approximate V^α and $\boldsymbol{u}$ not only in the cell face centers, but in other points as well. The general relation between V^α and $\boldsymbol{u}$ is (using (3.5))

$$V^\alpha = \sqrt{g}\boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u}, \qquad \boldsymbol{u} = \boldsymbol{a}_{(\alpha)} V^\alpha/\sqrt{g}. \tag{4.5}$$

Finding $\boldsymbol{u}$ requires evaluation of V^α in points other than its proper grid nodes. Because of lack of smoothness of the geometric quantities, evaluation of V^α in other points than

its proper grid nodes and evaluation of $\boldsymbol{u}$ need to be done carefully. A certain tedium is unavoidable. We impose the following accuracy requirement on formulas for defining V^α in other points than its proper grid nodes $\boldsymbol{x}_{j+e_\alpha}$: constant velocity fields $\boldsymbol{u}$ must be invariant under transformation to V^α-representation and back. More precisely, if $\boldsymbol{u}$ is given, and $V^\alpha_{j+e_\alpha}$ is computed with (4.5), and V^α is determined in another point by some interpolation recipe, and $\boldsymbol{u}$ is computed in this point with (4.5), then the original $\boldsymbol{u}$ is recovered exactly in the special case that $\boldsymbol{u}=$ constant. Furthermore, in the case of the identity mapping $\boldsymbol{x}=\boldsymbol{\xi}$, we will have multilinear interpolation. We have found these accuracy requirements to be essential for maintaining accuracy on rough grids.

In the cell centers $\boldsymbol{x}_j$ we define

$$V^\alpha_j \equiv \frac{1}{2}\left(V^\alpha_{j-e_\alpha} + V^\alpha_{j+e_\alpha}\right) \qquad \text{(no summation)}, \tag{4.6}$$

$$\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_j \equiv \frac{1}{2}\left\{\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j-e_\alpha} + \left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha}\right\} \qquad \text{(no summation)}, \tag{4.7}$$

$$\left(\frac{1}{\sqrt{g}}\boldsymbol{a}_{(\alpha)}\right)_j \equiv \left\{\frac{\sqrt{g}\boldsymbol{a}^{(\beta)}\times\sqrt{g}\boldsymbol{a}^{(\gamma)}}{\sqrt{g}\boldsymbol{a}^{(1)}\cdot\left(\sqrt{g}\boldsymbol{a}^{(2)}\times\sqrt{g}\boldsymbol{a}^{(3)}\right)}\right\}_j, \qquad \alpha,\beta,\gamma \text{ cyclic}. \tag{4.8}$$

We show that the above invariance requirement is met. Suppose $\boldsymbol{u}=$ constant. Then (4.6), (4.5), and (4.7) give

$$V^\alpha_j = \left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_j \cdot \boldsymbol{u}. \tag{4.9}$$

Recomputing $\boldsymbol{u}_j$ from V^α_j gives, using (4.5) and (4.8),

$$\boldsymbol{u}_j = \left(\frac{1}{\sqrt{g}}\boldsymbol{a}_{(\alpha)}\right)_j \left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_j \boldsymbol{u} = \boldsymbol{u}, \tag{4.10}$$

which we wanted to show.

In the cell face centers $\boldsymbol{x}_{j+e_\alpha}$ we define for $\beta\neq\alpha$,

$$V^\beta_{j+e_\alpha} \equiv \frac{1}{4}\left(V^\beta_{j+e_\beta} + V^\beta_{j-e_\beta} + V^\beta_{j+2e_\alpha+e_\beta} + V^\beta_{j+2e_\alpha-e_\beta}\right), \tag{4.11}$$

$$\left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+e_\alpha} \equiv \frac{1}{4}\left\{\left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+e_\beta} + \left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j-e_\beta} + \left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+2e_\alpha+e_\beta} + \left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+2e_\alpha-e_\beta}\right\} \tag{4.12}$$

(no summation), and $(\boldsymbol{a}_{(\alpha)}/\sqrt{g})_{j+e_\alpha}$ is related to $(\sqrt{g}\boldsymbol{a}^{(\beta)})_{j+e_\alpha}$ by replacing j by $j+e_\alpha$ in (4.8). In a similar way as before it is easily seen that the invariance requirement is satisfied.

In the cell edge centers $\boldsymbol{x}_{j+e_\alpha+e_\beta}$, $\beta\neq\alpha$ we define (no summation)

$$\begin{aligned}
V^\alpha_{j+e_\alpha+e_\beta} &\equiv \frac{1}{2}\left(V^\alpha_{j+e_\alpha} + V^\alpha_{j+e_\alpha+2e_\beta}\right),\\
\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha+e_\beta} &\equiv \frac{1}{2}\left\{\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha} + \left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha+2e_\beta}\right\},\\
V^\beta_{j+e_\alpha+e_\beta} &\equiv \frac{1}{2}\left(V^\beta_{j+e_\beta} + V^\beta_{j+2e_\alpha+e_\beta}\right),\\
\left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+e_\alpha+e_\beta} &\equiv \frac{1}{2}\left\{\left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+e_\beta} + \left(\sqrt{g}\boldsymbol{a}^{(\beta)}\right)_{j+2e_\alpha+e_\beta}\right\},
\end{aligned} \tag{4.13}$$

and for $\gamma \neq \alpha, \beta$

$$
\begin{aligned}
V^{\gamma}_{j+e_\alpha+e_\beta} &\equiv \frac{1}{8}\big(V^{\gamma}_{j-e_\gamma} + V^{\gamma}_{j+e_\gamma} + V^{\gamma}_{j-e_\gamma+2e_\alpha} + V^{\gamma}_{j+e_\gamma+2e_\alpha} + V^{\gamma}_{j-e_\gamma+2e_\beta} \\
&\quad + V^{\gamma}_{j+e_\gamma+2e_\beta} + V^{\gamma}_{j-e_\gamma+2e_\alpha+2e_\beta} + V^{\gamma}_{j+e_\gamma+2e_\alpha+2e_\beta}\big), \\
\big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j+e_\alpha+e_\beta} &\equiv \frac{1}{8}\big\{\big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j-e_\gamma} + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j+e_\gamma} + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j-e_\gamma+2e_\alpha} \\
&\quad + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j+e_\gamma+2e_\alpha} + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j-e_\gamma+2e_\beta} + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j+e_\gamma+2e_\beta} \\
&\quad + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j-e_\gamma+2e_\alpha+2e_\beta} + \big(\sqrt{g}\boldsymbol{a}^{(\gamma)}\big)_{j+e_\gamma+2e_\alpha+2e_\beta}\big\}.
\end{aligned}
\tag{4.14}
$$

As before, $(\boldsymbol{a}_{(\gamma)}/\sqrt{g})_{j+e_\alpha+e_\beta}$ is related to $(\sqrt{g}\boldsymbol{a}^{(\gamma)})_{j+e_\alpha+e_\beta}$ by replacing j by $j+e_\alpha+e_\beta$ in (4.8). Again, it is easily seen that the invariance requirement is satisfied.

Furthermore, $\boldsymbol{u}$ is required at the cell vertices $\boldsymbol{x}_{j+e_1+e_2+e_3}$. The way to proceed is clear from the preceding cases. We define ($\alpha \neq \beta, \alpha \neq \gamma, \beta \neq \gamma$, no summation)

$$
V^{\alpha}_{j+e_1+e_2+e_3} \equiv \frac{1}{4}\big(V^{\alpha}_{j+e_\alpha} + V^{\alpha}_{j+e_\alpha+2e_\beta} + V^{\alpha}_{j+e_\alpha+2e_\gamma} + V^{\alpha}_{j+e_\alpha+2e_\beta+2e_\gamma}\big), \tag{4.15}
$$

$$
\begin{aligned}
\big(\sqrt{g}\boldsymbol{a}^{(\alpha)}\big)_{j+e_1+e_2+e_3} &\equiv \frac{1}{4}\big\{\big(\sqrt{g}\boldsymbol{a}^{(\alpha)}\big)_{j+e_\alpha} + \big(\sqrt{g}\boldsymbol{a}^{(\alpha)}\big)_{j+e_\alpha+2e_\beta} \\
&\quad + \big(\sqrt{g}\boldsymbol{a}^{(\alpha)}\big)_{j+e_\alpha+2e_\gamma} + \big(\sqrt{g}\boldsymbol{a}^{(\alpha)}\big)_{j+e_\alpha+2e_\beta+2e_\gamma}\big\}
\end{aligned}
\tag{4.16}
$$

and $(\boldsymbol{a}_{(\alpha)}/\sqrt{g})$ is related to $(\sqrt{g}\boldsymbol{a}^{(\alpha)})$ in $\boldsymbol{x}_{j+e_1+e_2+e_3}$ by (4.8) with appropriate grid point indices.

5. COORDINATE-INVARIANT DISCRETIZATION OF THE NAVIER–STOKES EQUATIONS

Finite volume integration of the continuity equation div $\boldsymbol{u} = 0$ over Ω_j gives

$$
\begin{aligned}
0 &= \int_{\Omega_j} \operatorname{div} \boldsymbol{u}\, d\Omega = \int_{\Gamma_j} \boldsymbol{u}\cdot\boldsymbol{n}\, d\Gamma = \sum_{\alpha=1}^{3} (\boldsymbol{u}\cdot\boldsymbol{s})|_{j-e_\alpha}^{j+e_\alpha} \\
&= \sum_{\alpha=1}^{3} \Delta\xi^{\beta}\Delta\xi^{\gamma} V^{\alpha}|_{j-e_\alpha}^{j+e_\alpha}, \qquad \alpha, \beta, \gamma \text{ cyclic},
\end{aligned}
\tag{5.1}
$$

using the convention $\boldsymbol{u}|_k^j = \boldsymbol{u}_j - \boldsymbol{u}_k$. 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

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

Using the identity

$$\frac{\partial}{\partial \xi^\alpha}\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right) = 0, \tag{5.3}$$

this can be rewritten as

$$\frac{\partial \varphi}{\partial x^\alpha} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\beta}\left(\sqrt{g} a_\alpha^{(\beta)} \varphi\right). \tag{5.4}$$

The momentum equations can be written as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial}{\partial x^\alpha}(u^\alpha \boldsymbol{u}) = -\nabla p + \frac{\partial \nu \boldsymbol{e}^{(\alpha)}}{\partial x^\alpha}, \tag{5.5}$$

where

$$\boldsymbol{e}^{(\alpha)} \equiv \begin{pmatrix} \frac{\partial u^1}{\partial x^\alpha} + \frac{\partial u^\alpha}{\partial x^1} \\ \frac{\partial u^2}{\partial x^\alpha} + \frac{\partial u^\alpha}{\partial x^2} \\ \frac{\partial u^3}{\partial x^\alpha} + \frac{\partial u^\alpha}{\partial x^3} \end{pmatrix}. \tag{5.6}$$

Applying (5.4), Eq. (5.5) can be rewritten as

$$N(\boldsymbol{u}, p) \equiv \frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha}(\boldsymbol{u} V^\alpha) + \nabla p - \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha}\left(\sqrt{g} a_\beta^{(\alpha)} \nu \boldsymbol{e}^{(\beta)}\right) = 0. \tag{5.7}$$

This equation is integrated over the shifted finite volume Ω_{j+e_1} depicted in Fig. 5.1. Treating each term successively, we obtain

$$\int_{\Omega_{j+e_1}} \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega = \left|\Omega_{j+e_1}\right| d\boldsymbol{u}_{j+e_1}/dt, \tag{5.8}$$

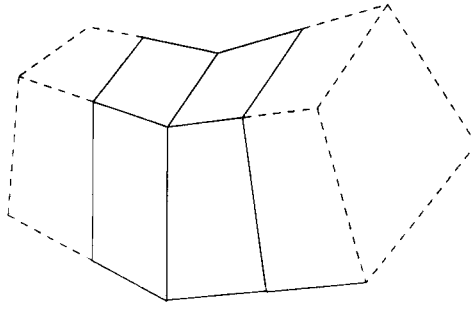

FIG. 5.1. Shifted finite volume Ω_{j+e_1}.

$$\int\limits_{\Omega_{j+e_1}} \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^\alpha} (\boldsymbol{u} V^\alpha)\, d\Omega = \int\limits_{G_{j+e_1}} \frac{\partial}{\partial \xi^\alpha} (\boldsymbol{u} V^\alpha)\, d\xi^1\, d\xi^2\, d\xi^3. \tag{5.9}$$

We have

$$\boldsymbol{I}_{11} \equiv \int\limits_{G_{j+e_1}} \frac{\partial}{\partial \xi^1} (\boldsymbol{u} V^1)\, d\xi^1\, d\xi^2\, d\xi^3 = \int\limits_{\xi^2_{j_2-1/2}}^{\xi^2_{j_2+1/2}} \int\limits_{\xi^3_{j_3-1/2}}^{\xi^3_{j_3+1/2}} (\boldsymbol{u} V^1)\big|_{j_1}^{j_1+1}\, d\xi^2\, d\xi^3, \tag{5.10}$$

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:

$$\boldsymbol{I}_{11} \cong \Delta\xi^2 \Delta\xi^3 (\boldsymbol{u} V^1)\big|_j^{j+2e_1}. \tag{5.11}$$

This is approximated further by using

$$V^1_j \cong \frac{1}{2}\big(V^1_{j-e_1} + V^1_{j+e_1}\big). \tag{5.12}$$

Similarly,

$$\boldsymbol{I}_{12} \equiv \int\limits_{G_{j+e_1}} \frac{\partial}{\partial \xi^2} (\boldsymbol{u} V^2)\, d\xi^1\, d\xi^2\, d\xi^3 \cong \Delta\xi^1 \Delta\xi^3 (\boldsymbol{u} V^2)\big|_{j+e_1-e_2}^{j+e_1+e_2}. \tag{5.13}$$

This is further approximated using

$$V^2_{j+e_1+e_2} \cong \frac{1}{2}\big(V^2_{j+e_2} + V^2_{j+2e_1+e_2}\big). \tag{5.14}$$

Analogously,

$$\boldsymbol{I}_{13} \equiv \int\limits_{G_{j+e_1}} \frac{\partial}{\partial \xi^3} (\boldsymbol{u} V^3)\, d\xi^1\, d\xi^2\, d\xi^3 \cong \Delta\xi^1 \Delta\xi^2 (\boldsymbol{u} V^3)\big|_{j+e_1-e_3}^{j+e_1+e_3}, \tag{5.15}$$

$$(\boldsymbol{u} V^3)_{j+e_1+e_3} \cong \frac{1}{4}\big(\boldsymbol{u}_{j+e_1} + \boldsymbol{u}_{j+e_1+2e_3}\big)\big(V^3_{j+e_3} + V^3_{j+2e_1+e_3}\big). \tag{5.16}$$

Integration of the pressure term is done as

$$\boldsymbol{I}_{14} \equiv \int\limits_{\Omega_{j+e_1}} \nabla p\, d\Omega \cong \nabla p_{j+e_1} \big|\Omega_{j+e_1}\big|, \tag{5.17}$$

where we have used smoothness of ∇p. The term ∇p_{j+e_1} is expressed in terms of surrounding nodal values in the following way. We write, using smoothness of ∇p,

$$b_1 \equiv p\big|_j^{j+2e_1} = \int\limits_{\boldsymbol{x}_j}^{\boldsymbol{x}_{j+2e_1}} \nabla p \cdot d\boldsymbol{x} \cong \nabla p_{j+e_1} \cdot \boldsymbol{c}_{(1)}, \qquad \boldsymbol{c}_{(1)} \equiv \boldsymbol{x}\big|_j^{j+2e_1}. \tag{5.18}$$

Similarly,

$$b_2 \equiv p|_{j-2e_2}^{j+2e_2} + p|_{j+2e_1-2e_2}^{j+2e_1+2e_2} = \left\{ \int_{\boldsymbol{x}_{j-2e_2}}^{\boldsymbol{x}_{j+2e_2}} + \int_{\boldsymbol{x}_{j+2e_1-2e_2}}^{\boldsymbol{x}_{j+2e_1+2e_2}} \right\} \nabla p \cdot d\boldsymbol{x} \cong \nabla p_{j+e_1} \cdot \boldsymbol{c}_{(2)}, \quad (5.19)$$

$$\boldsymbol{c}_{(2)} \equiv \boldsymbol{x}|_{j-2e_2}^{j+2e_2} + \boldsymbol{x}|_{j+2e_1-2e_2}^{j+2e_1+2e_2}. \quad (5.20)$$

Using the third direction we obtain similarly

$$b_3 \equiv p|_{j-2e_3}^{j+2e_3} + p|_{j+2e_1-2e_3}^{j+2e_1+2e_3} \cong \nabla p_{j+e_1} \cdot \boldsymbol{c}_{(3)}, \quad (5.21)$$

$$\boldsymbol{c}_{(3)} \equiv \boldsymbol{x}|_{j-2e_3}^{j+2e_3} + \boldsymbol{x}|_{j+2e_1-2e_3}^{j+2e_1+2e_3}. \quad (5.22)$$

We now have three equations expressing ∇p in terms of surrounding nodal values, without any assumption about the smoothness of the coordinate mapping. The system is solved for ∇p by defining

$$\boldsymbol{c}^{(\alpha)} \equiv \boldsymbol{c}_{(\beta)} \times \boldsymbol{c}_{\gamma}/C, \qquad C \equiv \boldsymbol{c}_{(1)} \cdot \big(\boldsymbol{c}_{(2)} \times \boldsymbol{c}_{(3)}\big). \quad (5.23)$$

Then the solution of the above three equations for ∇p is

$$\frac{\partial p}{\partial x_\alpha} \cong c_\alpha^{(\beta)} b_\beta. \quad (5.24)$$

Integration of the viscous term over Ω_{j+e_1} gives three contributions. The first is

$$\begin{aligned} \boldsymbol{I}_{15} &\equiv \int_{G_{j+e_1}} \frac{\partial}{\partial \xi_1} \big(\sqrt{g} a_\beta^{(1)} \nu \boldsymbol{e}^{(\beta)}\big)\, d\xi^1\, d\xi^2\, d\xi^3 \\ &= \int_{\xi^2_{j_2-1/2}}^{\xi^2_{j_2+1/2}} \int_{\xi^3_{j_3-1/2}}^{\xi^3_{j_3+1/2}} \big(\sqrt{g} a_\beta^{(1)} \nu \boldsymbol{e}^{(\beta)}\big)\big|_{j_1}^{j_1+1} d\xi^2\, d\xi^3. \end{aligned} \quad (5.25)$$

In (5.25), $\sqrt{g} a_\beta^{(1)}$ is constant. We write

$$\boldsymbol{I}_{15} \cong \Delta\xi^2 \Delta\xi^3 \big(\sqrt{g} a_\beta^{(1)} \nu \boldsymbol{e}^{(\beta)}\big)\big|_{j}^{j+2e_1}. \quad (5.26)$$

The second contribution is

$$\boldsymbol{I}_{16} \equiv \int_{\xi^1_{j_1}}^{\xi^1_{j_1+1}} \int_{\xi^3_{j_3-1/2}}^{\xi^3_{j_3+1/2}} \big(\sqrt{g} a_\beta^{(2)} \nu \boldsymbol{e}^{(\beta)}\big)\big|_{j_2-1/2}^{j_2+1/2} d\xi^1\, d\xi^3. \quad (5.27)$$

In (5.27), $\sqrt{g}\boldsymbol{a}^{(2)}$ is piecewise constant. We make the following approximation,

$$\boldsymbol{I}_{16} \cong \Delta\xi^1 \Delta\xi^3 \big(\sqrt{g} a_\beta^{(2)} \nu \boldsymbol{e}^{(\beta)}\big)\big|_{j+e_1-e_2}^{j+e_1+e_2}, \quad (5.28)$$

where we define

$$\left(\sqrt{g}\boldsymbol{a}^{(2)}\right)_{j+e_1\pm e_2} \equiv \frac{1}{2}\left\{\left(\sqrt{g}\boldsymbol{a}^{(2)}\right)_{j\pm e_2} + \left(\sqrt{g}\boldsymbol{a}^{(2)}\right)_{j+2e_1\pm e_2}\right\}. \tag{5.29}$$

The third contribution is $\boldsymbol{I}_{17}$, which is handled just like $\boldsymbol{I}_{16}$,

$$\begin{aligned}\boldsymbol{I}_{17} &\equiv \int_{\xi^1_{j_1}}^{\xi^1_{j_1+1}} \int_{\xi^2_{j_2-1/2}}^{\xi^2_{j_2+1/2}} \left(\sqrt{g}a^{(3)}_\beta \nu \boldsymbol{e}^{(\beta)}\right)\Big|_{j_3-1/2}^{j_3+1/2} d\xi^1\, d\xi^2 \\ &\cong \Delta\xi^1\Delta\xi^2\left(\sqrt{g}a^{(3)}_\beta \nu \boldsymbol{e}^{(\beta)}\right)\Big|_{j+e_1-e_3}^{j+e_1+e_3},\end{aligned} \tag{5.30}$$

where

$$\left(\sqrt{g}\boldsymbol{a}^{(3)}\right)_{j+e_1\pm e_3} \equiv \frac{1}{2}\left\{\left(\sqrt{g}\boldsymbol{a}^{(3)}\right)_{j\pm e_3} + \left(\sqrt{g}\boldsymbol{a}^{(3)}\right)_{j+2e_1\pm e_3}\right\}. \tag{5.31}$$

In $\boldsymbol{I}_{15}$, $\boldsymbol{I}_{16}$, and $\boldsymbol{I}_{17}$, $\boldsymbol{e}^{(\beta)}$ has to be approximated, requiring discretizations of derivatives of $\boldsymbol{u}$. We start with $\boldsymbol{I}_{15}$ and write, using (5.2),

$$\left(\frac{\partial \boldsymbol{u}}{\partial x^\beta}\right)_j = \left(a^{(\alpha)}_\beta \frac{\partial \boldsymbol{u}}{\partial \xi^\alpha}\right)_j. \tag{5.32}$$

In a cell-center we are not hampered by nonsmoothness of the mapping $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$, and the following straightforward approximation can be used:

$$\left(\frac{\partial \boldsymbol{u}}{\partial x^\beta}\right)_j \cong \sum_{\alpha=1}^{3} \frac{1}{\Delta\xi^\alpha}\left(a^{(\alpha)}_\beta\right)_j \boldsymbol{u}\Big|_{j-e_\alpha}^{j+e_\alpha}. \tag{5.33}$$

The same procedure cannot be followed for $\boldsymbol{I}_{16}$ and $\boldsymbol{I}_{17}$, because we are at a cell edges, where the geometric quantities are discontinuous. Instead, we proceed in a similar way as for ∇p. An approximation for $\partial u^\alpha/\partial x^\beta$ in $\boldsymbol{x}_{j+e_1+e_2}$, for example, is derived by writing (no summation)

$$\begin{aligned} b_\gamma \equiv u^\alpha\Big|_{j+e_1+e_2-e_\gamma}^{j+e_1+e_2+e_\gamma} &= \int_{\boldsymbol{x}_{j+e_1+e_2-e_\gamma}}^{\boldsymbol{x}_{j+e_1+e_2+e_\gamma}} \nabla u^\alpha \cdot d\boldsymbol{x} \cong \nabla u^\alpha_{j+e_1+e_2}\cdot \boldsymbol{c}_{(\gamma)}, \\ \boldsymbol{c}_{(\gamma)} &\equiv \boldsymbol{x}\Big|_{j+e_1+e_2-e_\gamma}^{j+e_1+e_2+e_\gamma}, \qquad \gamma = 12, 3. \end{aligned} \tag{5.34}$$

This system of three equations for ∇u^α is solved in similar manner as the system for ∇p. We define $\boldsymbol{c}^{(\alpha)}$ by (5.23), taking $\boldsymbol{c}_{(\alpha)}$ from (5.34), and find

$$\left(\frac{\partial u^\alpha}{\partial x^\beta}\right)_{j+e_1+e_2} \cong c^{(\gamma)}_\beta b_\gamma. \tag{5.35}$$

In $c^{(\gamma)}_\beta$ values of $\boldsymbol{u}$ in cell face centers, cell edge centers and cell vertices occur; these are expanded in terms of V^α by means of (4.5), (4.8), and (4.12)–(4.16).

The momentum equation (5.7) is integrated also over shifted finite volumes Ω_{j+e_2} and Ω_{j+e_3} and discretized in a completely similar manner. We may consider the resulting finite volume discretization of (5.7), briefly denoted as

$$\int_{\Omega_{j+e_\alpha}} N(\boldsymbol{u},\, p)\, d\Omega = 0 \tag{5.36}$$

with the integral approximated in the way indicated above, as semi-discretized evolution equations for $\boldsymbol{u}$ at the cell face 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 staggered scheme of Harlow and Welch [33] from Cartesian to general grids, which implies that we want evolution equations for the volume flux components at the cell face centers. Using (4.5), this is achieved by replacing (5.36) by

$$\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha} \cdot \int_{\Omega_{j+e_\alpha}} N(\boldsymbol{u},\, p)\, d\Omega = 0 \qquad \text{(no summation)}, \tag{5.37}$$

with the integral approximated in the way described above. This gives for the various terms in $N(\boldsymbol{u},\, p)$ the following results. From (5.8) and (4.5) it follows that

$$\left(\sqrt{g}\boldsymbol{a}^{(\alpha)}\right)_{j+e_\alpha} \cdot \int_{\Omega_{j+e_\alpha}} \frac{\partial \boldsymbol{u}}{\partial t}\, d\Omega = \left|\Omega_{j+e_\alpha}\right| dV^\alpha_{j+e_\alpha}/dt \qquad \text{(no summation)}, \tag{5.38}$$

which is precisely what we want. Furthermore.

$$\left(\sqrt{g}\boldsymbol{a}^{(1)}\right)_{j+e_1} \cdot \boldsymbol{I}_{11} \cong \Delta\xi^2 \Delta\xi^3 \left(\sqrt{g}\boldsymbol{a}^{(1)}\right)_{j+e_1} \cdot (\boldsymbol{u}V^1)\big|_j^{j+2e_1}. \tag{5.39}$$

The equivalent of the second order central scheme for the convection term is obtained if we express $\boldsymbol{u}$ in terms of V^α in the way described in Section 4. The first order upwind scheme is obtained if we shift the flux in the usual way; for example, if $V^1_j > 0$ then $(\boldsymbol{u}V^1)_j$ is replaced by $(\boldsymbol{u}V^1)_{j-e_1}$. After this shift $\boldsymbol{u}$ is expressed in terms of V^α. A way to implement second order upwind-biased flux-limited schemes is to substitute in (5.39) componentwise, assuming $V^1_j > 0$, in the notation of [79]:

$$u_j = u_{j-e_1} + \frac{1}{2}\psi(r_j)\left(u_{j+e_1} - u_{j-e_1}\right),$$

$$r_j = \left(u_{j-e_1} - u_{j-3e_1}\right)\big/\left(u_{j+e_1} - u_{j-e_1}\right).$$

When solving, the nonlinear term can lag behind, so that defect correction is applied to the first order upwind scheme, which is common practice for flux-limited schemes.

There is no need to discuss the remaining terms. In this way a coordinate-invariant discretization is obtained, that in the case of the identity mapping $\boldsymbol{x} = \boldsymbol{\xi}$ is reduced to the classical Cartesian staggered discretization of Harlow and Welch [33]. Furthermore, the discretization error is zero for $\boldsymbol{u}$ and ∇p constant, regardless of roughness and nonorthogonality of the grid. It would be too tedious to show this here, but verification by numerical experiment is easy.

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

$$\left\{ \begin{array}{c} \alpha \\ \beta\gamma \end{array} \right\} \equiv \boldsymbol{a}^{(\alpha)} \cdot \frac{\partial \boldsymbol{a}_{(\beta)}}{\partial \xi^{\gamma}}. \tag{5.40}$$

In (5.39), for example, if $\boldsymbol{u}$ is expressed in terms of V^{α}, products of $\boldsymbol{a}^{(\alpha)}$ in one point and $\boldsymbol{a}_{(\beta)}$ in neighbouring points are hidden, similar to what one would get by discretizing (5.40) (assuming differentiability of $\boldsymbol{a}_{(\beta)}$). Furthermore, contra- and covariant base vectors are implicitly present in $\boldsymbol{I}_{15}$, because $\boldsymbol{c}_{(\alpha)}$ in (5.18), (5.20), and (5.22) is related to $\boldsymbol{a}_{(\alpha)}$, and $\boldsymbol{c}^{(\alpha)}$ in (5.23) 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 (5.40).

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

6. SOLUTION METHODS

Putting all unknows $V^{\alpha}_{j+e_\alpha}$ (no summation) 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 structure

$$\begin{aligned} \frac{du}{dt} + N(u) + Gp &= f(t), \\ Du &= g(t). \end{aligned} \tag{6.1}$$

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 $t^n = n\tau$,

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

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

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

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

$$\begin{aligned} u^n - u^{n-1} + \theta\tau N(u^n) + (1-\theta)\tau N(u^{n-1}) + \tau G p^{n-1/2} &= \theta f(t^n) + (1-\theta) f(t^{n-1}) \\ Du^n &= g(t^n) \end{aligned} \tag{6.4}$$

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

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

$$u^n - u^* + \tau G(p^{n-1/2} - p^{n-3/2}) = 0 \tag{6.6}$$

$$Du^n = g(t^n). \tag{6.7}$$

Equation (6.5) 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 (6.6) and using (6.7), resulting in

$$DG\delta p = \frac{1}{\tau}(Du^* - g(t^n)), \qquad p^{n-1/2} = p^{n-3/2} + \delta p \tag{6.8}$$

After δp has been computed, u^n follows from (6.6). Because DG is a discrete Laplacian, (6.8) 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 [26–28].

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 (6.5)–(6.7) is equivalent to (6.2), and this remains true if $p^{n-3/2}$ is neglected in (6.5) and (6.6). But in the implicit case this does not hold, and inclusion of a sufficiently accurate first guess for $p^{n-1/2}$ in (6.5), 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

$$\frac{dg(t)}{dt} + DN(u(t)) + DGp(t) = Df(t). \tag{6.9}$$

After putting $t = 0$ and solving $p(0)$ from (6.9), we put $p^{-1/2} = p(0)$ in (6.5).

In the Cartesian case, convergence of the method is studied theoretically in [36] and references quoted there. The pressure correction method has been formulated and studied

in [3, 13–15, 22, 33, 36, 6, 81, 83]. 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 (6.9) for $t = t^n$ to find p^n with the same order of temporal accuracy as u^n.

For stability of (6.5)–(6.7) it seems necessary that (6.5) is stable. Since (6.5) 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 [91]. It is conjectured, supported by numerical evidence, that stability of (6.5) is sufficient for stability of (6.5)–(6.7).

In the explicit case, on orthogonal grids, the bulk of the computing time goes to solving δp from (6.8), 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 [78]. On general grids these methods are not applicable. In general, the matrix of (6.8) is not symmetric, and when the coordinates are strongly non-orthogonal it is not an M-matrix. But multigrid [31, 89, 97] and preconditioned Krylov subspace methods [70, 86, 87, 97] 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 [89], 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 [89]. We found that on general grids the discretization that we presented is sufficiently complicated to make the work required to update the matrix A in (6.5) not neglible, though still smaller than the work required to compute the pressure correction.

In the implicit case, (6.5) 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 derivate 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 [89] the same methods as before are found to be eligible. Navier–Stokes applications are shown in [89, 86, 87, 97].

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 colocated finite volume methods using Rhie–Chow interpolation [61] and finite element methods, we approximate a simple exact solution on a rough grid. We start with a simple two-dimensional example, so that the coordinate mapping satisfies $x^\alpha = x^\alpha(\xi^1, \xi^2)$ $(\alpha = 1, 2)$, $x^3 = \xi^3$. The

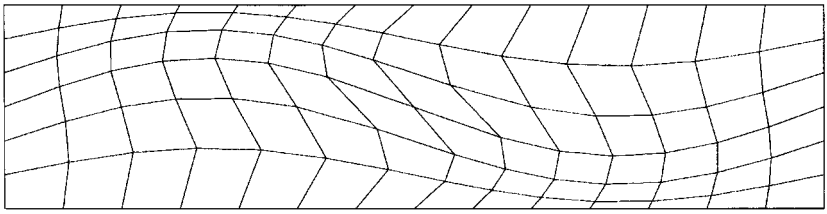

FIG. 7.1. Grid for Poiseuille flow.

exact solution is Poiseuille flow:

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

The grid, shown in Fig. 7.1, is chosen rough deliberately. Figures 7.2–7.4 give stream-lines 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 colocated discretization with Rhie–Chow interpolation. Figure 7.5 gives an even wilder grid, meant to investigate the effect of sudden refinement and derefinement. Results are shown in Figs. 7.6 and 7.7. We do not have results for the colocated 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. Figure 7.8 shows a grid that is also distorted in the ξ^3-direction. The exact solution is chosen as before. Figure 7.9 shows the intersections (isobars) of constant pressure surfaces with several planes $x^2 = \text{constant}$ and $x^3 = \text{constant}$ (the x^1-axis points in the longitudinal direction). These lines should be straight and uniformly spaced. We see that this is the case to a satisfactory extent, given the extreme roughness of the grid.

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. Examples involving physically more interesting flows are referred

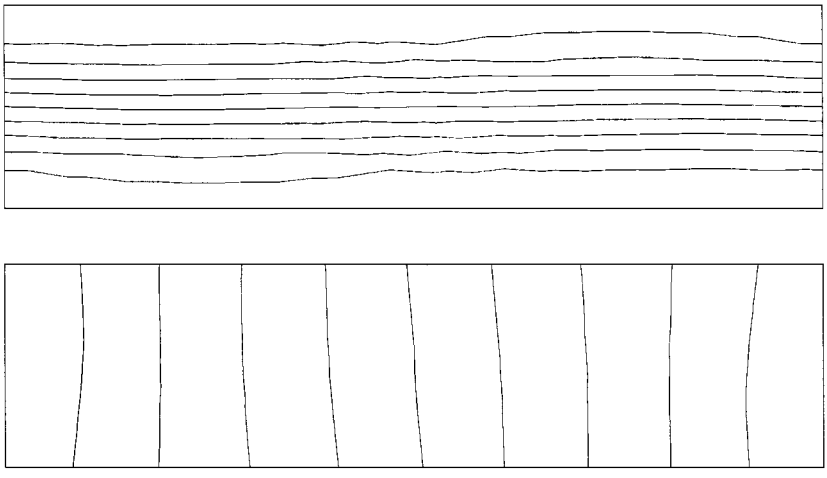

FIG. 7.2. Streamlines and isobars for staggered discretization.

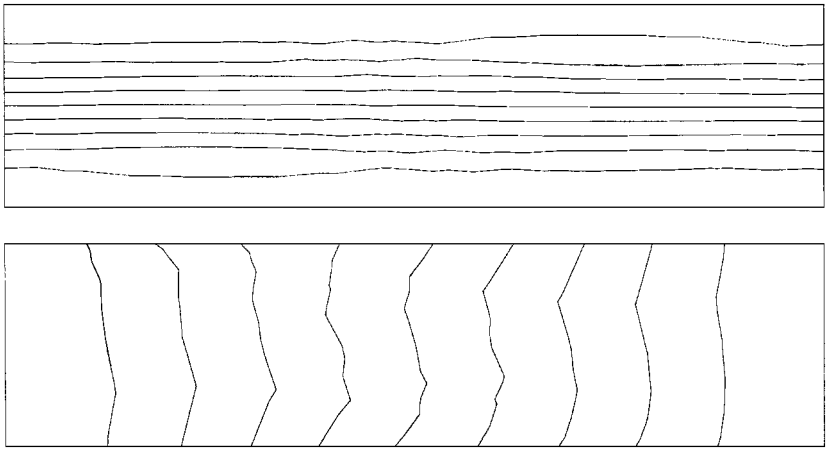

FIG. 7.3. Streamlines and isobars for finite element method.

to below. Satisfactory results were obtained on more or less smooth grids using an earlier version of our code making explicit use of Christoffel symbols. With the version described above results are at least as good.

Extension of the present approach to the case of nonstationary compressible flow is described in [5], resulting in a unified method for compressible and incompressible flow, with approximately Mach-independent accuracy and efficiency. For the development of such Mach-uniform methods, staggered schemes have an advantage over colocated schemes, that we will now briefly discuss. The disadvantage of greater difficulty in handling geometrically complicated domains remains, of course, a difficulty that is not unsurmountable, as we have shown above. The limit $M \downarrow 0$ of the Euler and Navier–Stokes equations is singular [42, 47]. Classical methods to compute compressible flows break down as $M \downarrow 0$. Measures can be taken to decrease to a certain extent the lowest value of M for which reasonable results can be obtained [11, 30, 82, 84] (until $M \cong 0.1$), but these measures usually falsify transient behaviour and are therefore limited to stationary flows, and $M = 0$ cannot be reached. A

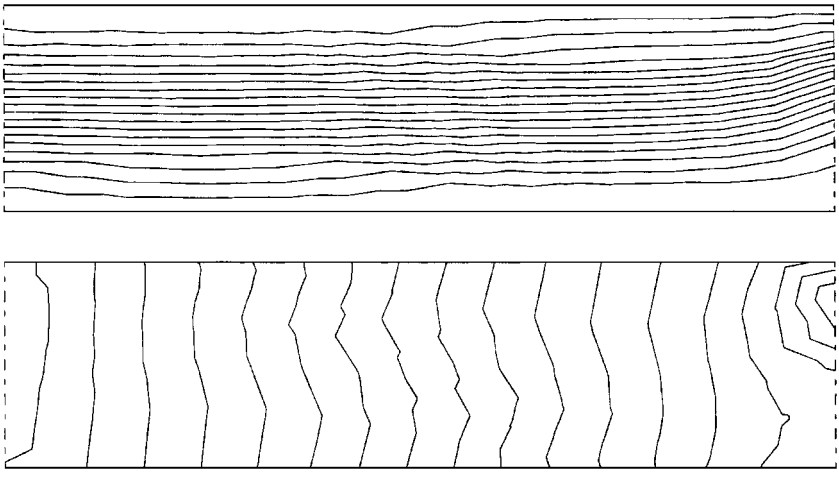

FIG. 7.4. Streamlines and isobars for colocated discretization.

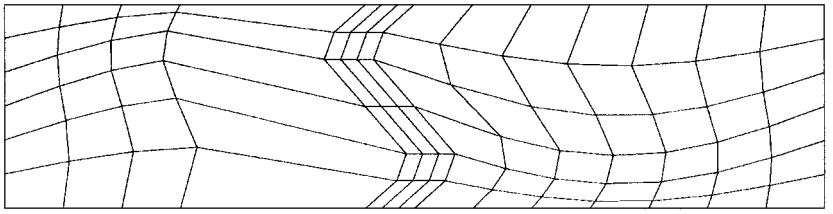

FIG. 7.5. Grid for Poiseuille flow.

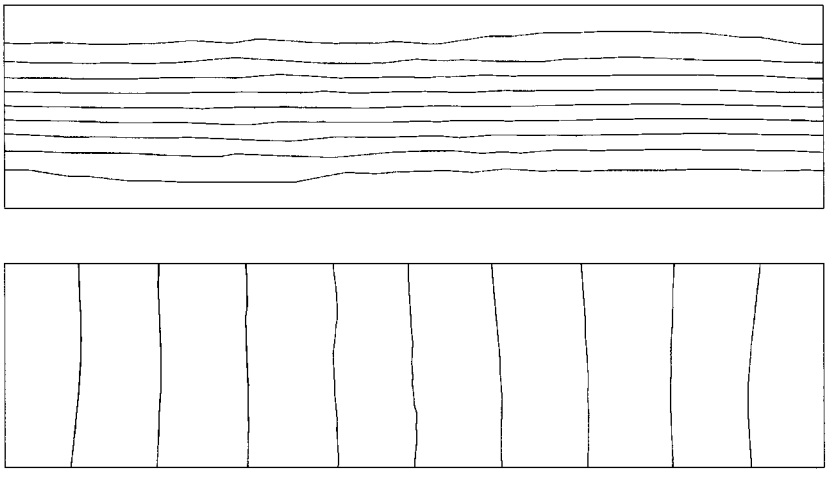

FIG. 7.6. Streamlines and isobars for staggered discretization.

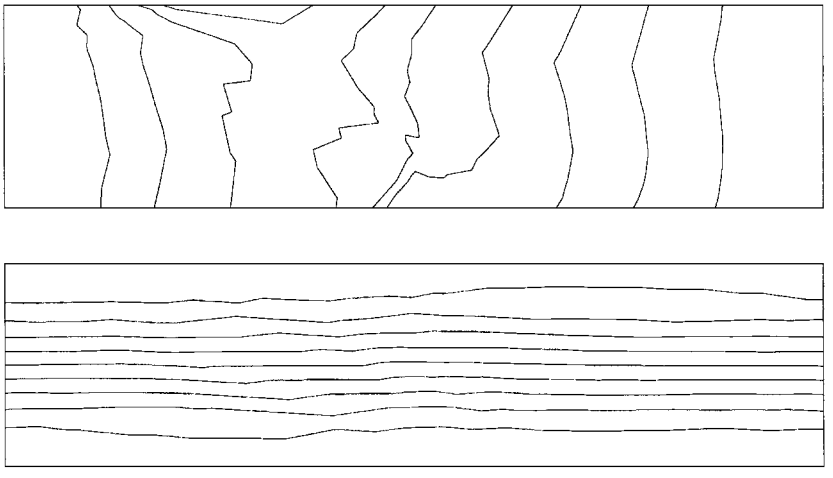

FIG. 7.7. Streamlines and isobars for finite element method.

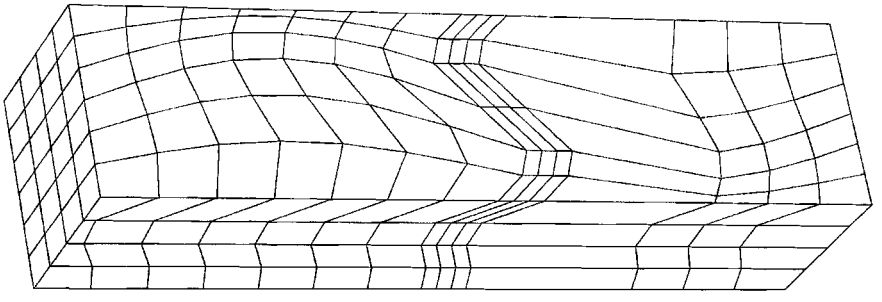

FIG. 7.8. Three-dimensional grid for Poiseuille flow.

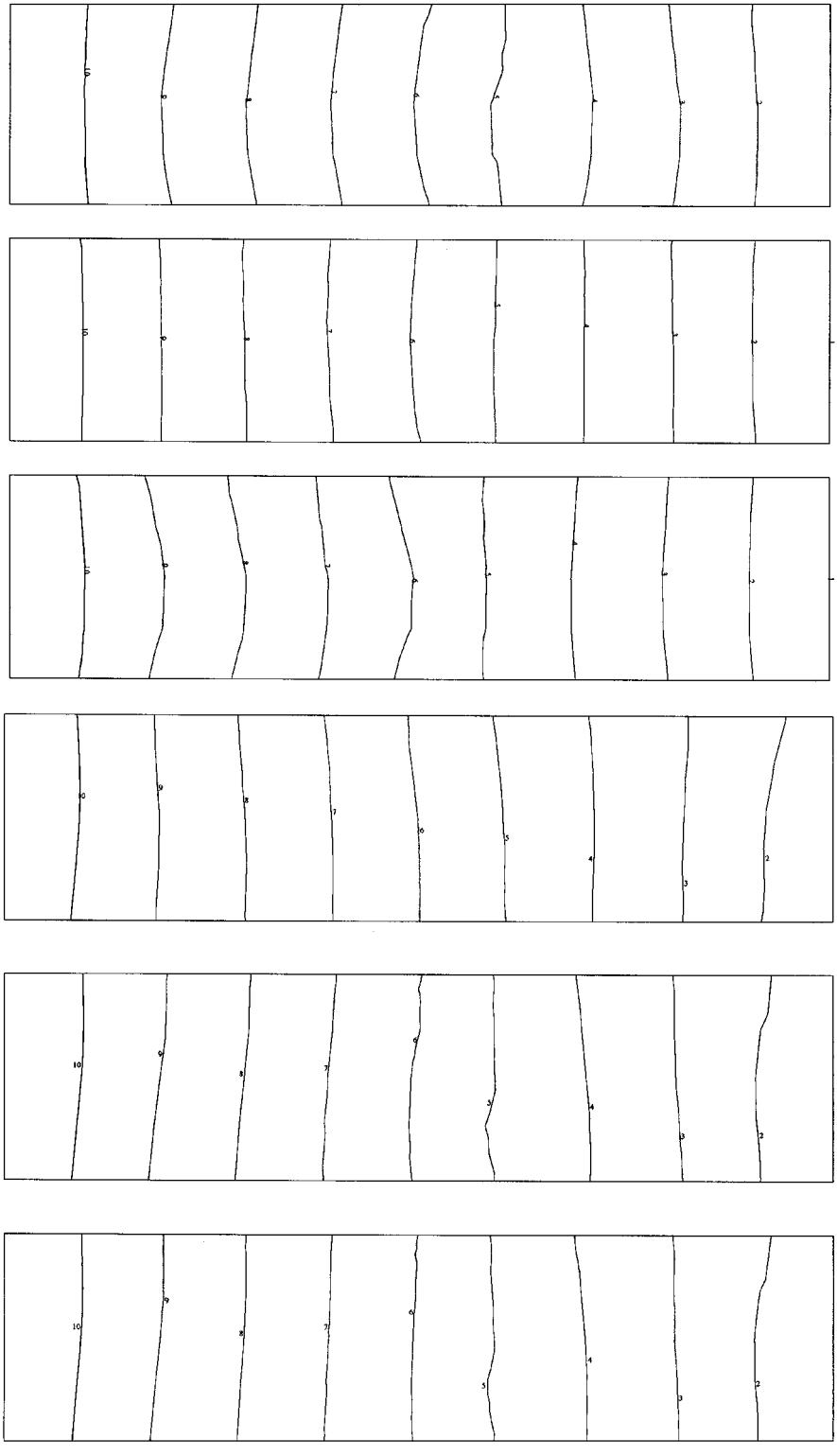

FIG. 7.9. Top to bottom, isobars at $x^2 = 1/4,\ 1/2,\ 3/4$ and $x^3 = 1/4,\ 1/2,\ 3/4$.

method that tends to an established method for incompressible flows as $M \downarrow 0$ 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 colocated scheme in [18] and for a staggered scheme in [32, 39, 73–75]. Colocated schemes add, either implicitly or explicitly, an artificial regularizing term to the mass conservation equation; for an explicit expression of this term generated by the Rhie–Chow velocity interpolation method [61], see [51]. 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 nonstationary flows.

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 [98–103]. 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 [99, 103].

8. CONCLUDING REMARKS

We have shown how the classical staggered scheme for the incompressible Navier–Stokes equations 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 typical colocated 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. The resulting scheme works satisfactorily from $M = 0$ to $M > 1$.

ACKNOWLEDGMENT

The authors are indebted to R. Agtersloot for providing the results of Fig. 7.4.

REFERENCES

1. S. Acharya and F. H. Moukalled, Improvements to incompressible flow calculation on a nonstaggered curvilinear grid, *Numer. Heat Transfer B* **15**, 131 (1989).
2. S. W. Armfield, Ellipticity, accuracy and convergence of the discrete Navier–Stokes equations, *J. Comput. Phys.* **114**, 176 (1994).
3. J. B. Bell, P. Colella, and H. M. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 257 (1989).
4. A. A. Belov, L. Martinelli, and A. Jameson, A novel fully implicit multigrid driven algorithm for unsteady incompressible flow calculations, in *Computational Fluid Dynamics '94*, edited by S. Wagner, E. H. Hirschel, J. Périaux, and R. Piva (Wiley, Chichester, 1994), pp. 662–670.
5. H. Bijl and P. Wesseling, A unified method for computing incompressible and compressible flows in boundary-fitted coordinates, *J. Comput. Phys.* **141**, 153 (1998).
6. J. Blair Perot, An analysis of the fractional step method, *J. Comput. Phys.* **108**, 51 (1993).

7. M. Braaten and W. Shyy, A study of recirculating flow computation using body-fitted coordinates: Consistency aspects and mesh skewness, *Numer. Heat Transfer* **9**, 559 (1986).
8. E. Brakkee, C. Vuik, and P. Wesseling, Domain decomposition for the incompressible Navier–Stokes equations: Solving subdomain problems accurately and inaccurately, *Internat. J. Numer. Methods Fluids* **26**, 1217 (1998).
9. A. D. Burns, I. P. Jones, J. R. Kightley, and N. S. Wilkes, The implementation of a finite difference method for predicting incompressible flows in complex geometries, in *Num. Meth. in Lam. and Turb. Fl.*, edited by C. Taylor, W. G. Habashi, and M. M. Hafez (Pineridge, Swansea, 1987), Vol. 5, pp. 339–350.
10. J. L. C. Chang and D. Kwak, *On the Method of Pseudo Compressibility for Numerically Solving Incompressible Flows*, AIAA Paper 84-0252, 1984.
11. Y.-H. Choi and C. L. Merkle, The application of preconditioning in viscous flows, *J. Comput. Phys.* **105**, 207 (1993).
12. A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).
13. A. J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comp.* **22**, 745 (1968).
14. A. J. Chorin, On the convergence of discrete approximations to the Navier–Stokes equations, *Math. Comp.* **23**, 342 (1969).
15. A. J. Chorin, Numerical solution of incompressible flow problems, in *Studies in Numerical Analysis 2*, edited by J. M. Ortega and W. C. Rheinbold (SIAM, Philadelphia, 1970), pp. 64–71.
16. L. Davidson and P. Hedberg, Mathematical derivation of a finite volume formulation for laminar flow in complex geometries, *Internat. J. Numer. Methods Fluids* **9**, 531 (1989).
17. I. Demirdžić, A. D. Gosman, R. I. Issa, and M. Perić, A calculation procedure for turbulent flow in complex geometries, *Comput. & Fluids* **15**, 251 (1987).
18. I. Demirdžić, Z. Lilek, and M. Perić, A collocated finite volume method for predicting flows at all speeds, *Internat. J. Numer. Methods Fluids* **16**, 1029 (1993).
19. I. Demirdžić and M. Perić, Finite volume method for prediction of fluid flow in arbitrary shaped domains with moving boundaries, *Internat. J. Numer. Methods Fluids* **10**, 771 (1990).
20. G. B. Deng, Numerical simulation of incompressible turbulent appendage-flat plate junction flows, in *Numerical Methods in Laminar and Turbulent Flows*, edited by C. Taylor, W. G. Habashi, and M. M. Hafez (Pineridge, Swansea, 1989), Vol. 6, Part I, pp. 793–803.
21. D. Drikakis and M. Schäfer, Comparison between a pressure correction and an artificial compressibility/characteristic based method in parallel incompressible fluid flow computations, in *Computational Fluid Dynamics '94*, edited by S. Wagner, E. H. Hirschel, J. Périaux, and R. Piva (Wiley, Chichester, 1994), pp. 619–626.
22. J. K. Dukowicz and A. S. Dvinsky, Approximate factorization as a high order splitting for the implicit incompressible flow equations, *J. Comput. Phys.* **102**, 336 (1992).
23. J. H. Ellison, C. A. Hall, and T. A. Porsching, An unconditionallystable convergent finite difference method for Navier–Stokes problems on curved domains, *SIAM J. Numer. Anal.* **24**, 1233 (1987).
24. P. A. Farrell, J. J. Miller, E. O'Riordan, and G. I. Shishkin, A uniformly convergent finite difference scheme for a singularly perturbed semilinear equation, *SIAM J. Numer. Anal.* **33**, 1135 (1996).
25. L. Fuchs and H. S. Zhao, Solution of three-dimensional viscous incompressible flows by a multigrid method, *Internat. J. Numer. Methods Fluids* **4**, 539 (1984).
26. M. P. Gresho and R. L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Fluids* **7**, 1111 (1987).
27. P. M. Gresho, Some current CFD issues relevant to the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Engrg.* **87**, 201 (1991).
28. P. M. Gresho, Some interesting issues in incompressible fluid dynamics, both in the continuum and in numerical simulation, *Adv. Appl. Mech.* **28**, 45 (1992).
29. C.-Y. Gu, Computations of flows with large body forces, in *Numerical Methods in Laminar and Turbulent Flow*, edited by C. Taylor, J. H. Chin, and G. M. Homsky (Pineridge, Swansea, 1991), Vol. 7, Part 2, pp. 1568–1578.

30. J. Guerra and B. Gustafsson, A numerical method for incompressible and compressible flow problems with smooth solutions, *J. Comput. Phys.* **63**, 377 (1986).
31. W. Hackbusch, *Multi-grid Methods and Applications* (Springer-Verlag, Berlin, 1985).
32. F. H. Harlow and A. A. Amsden, A numerical fluid dynamics calculation method for all flow speeds, *J. Comput. Phys.* **8**, 197 (1971).
33. F. H. Harlow and J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface, *Phys. Fluids* **8**, 2182 (1965).
34. P.-M. Hartwich and C.-H. Hsu, High-resolution upwind schemes for the three-dimensional incompressible Navier–Stokes equations, *AIAA J.* **26**, 1321 (1988).
35. Y.-H. Ho and B. Lakshminarayana, Computation of unsteady viscous flow using a pressure-based algorithm, *AIAA J.* **31**, 2232 (1993).
36. T. Y. Hou and B. T. R. Wetton, Second-order convergence of a projection scheme for the incompressible Navier–Stokes equations with boundaries, *SIAM J. Numer. Anal.* **30**, 609 (1993).
37. T. Ikohagi and B. R. Shin, Finite difference schemes for steady incompressible Navier–Stokes equations in general curvilinear coordinates, *Comput. & Fluids* **19**, 479 (1991).
38. T. Ikohagi, B. R. Shin, and H. Daiguji, Application of an implicit time-marching scheme to a three-dimensional incompressible flow problem in curvilinear coordinate systems, *Comput. & Fluids* **21**, 163 (1992).
39. K. C. Karki and S. V. Patankar, Pressure based calculation procedure for viscous flows at all speed in arbitrary configurations, *AIAA J.* **27**, 1167 (1989).
40. M. H. Kobayashi and J. C. F. Pereira, Numerical comparison of momentum interpolation methods and pressure-velocity algorithms using non-staggered grids, *Comm. Appl. Numer. Methods* **7**, 173 (1991).
41. S. Koshizuka, Y. Oka, and S. Kondo, A staggered differencing technique on boundary-fitted curvilinear grids for incompressible flows along curvilinear or slant walls, *Comput. Mech.* **7**, 123 (1990).
42. H. O. Kreiss, J. Lorenz, and M. J. Naughton, Convergence of the solutions of the compressible to the solutions of the incompressible Navier–Stokes equations, *Adv. Appl. Math.* **12**, 187 (1991).
43. D. Kwak, J. I. C. Chang, S. P. Shanks, and S. R. Chakravarthy, A three-dimensional incompressible Navier–Stokes flow solver using primitive variables, *AIAA J.* **24**, 390 (1986).
44. D. Kwak and J. L. C. Chang, A computational method for viscous incompressible flows, in *Advances in Computer Methods for Partial Differential Equations V*, edited by R. Vichnevetsky and R. S. Stepleman (IMACS, New Brunswick, 1984), pp. 277–288.
45. B. L. Lapworth, Examination of pressure oscillations arising in the computation of cascade flow using a boundary-fitted co-ordinate system, *Internat. J. Numer. Methods Fluids* **8**, 387 (1988).
46. F. S. Lien and M. A. Leschziner, Multigrid acceleration for recirculating laminar and turbulent flows computed with a non-orthogonal, collocated finite-volume scheme, *Comput. Methods Appl. Mech. Engrg.* **118**, 351 (1994).
47. A. Majda, Compressible fluid flow and systems of conservation laws in several space variables, *Applied Mathematical Sciences* (Springer-Verlag, New York, 1984), Vol. 53.
48. T. A. Manteuffel and A. B. White, Jr., The numerical solution of second-order boundary value problems on nonuniform meshes, *Math. Comp.* **47**, 511 (1986).
49. C. L. Merkle and M. Athvale, *Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility*, AIAA Paper 87-1137, 1987.
50. C. L. Merkle and P. Y. L. Tsai, *Application of Runge–Kutta Schemes to Incompressible Flow*, AIAA Paper 86-0553, 1986.
51. T. F. Miller and F. W. Schmidt, Use of a pressure-weighted interpolation method for the solution of the incompressible Navier–Stokes equations on a nonstaggered grid system, *Numer. Heat Transfer* **14**, 213 (1988).
52. F. Moukalled and S. Acharya, A local adaptive grid procedure for incompressible flows with multigridding and equidistribution concepts, *Internat. J. Numer. Methods Fluids* **13**, 1085 (1991).
53. A. E. Mynett, P. Wesseling, A. Segal, and C. G. M. Kassels, The ISNaS incompressible Navier–Stokes solver: Invariant discretization, *Appl. Sci. Res.* **48**, 175 (1991).

54. C. W. Oosterlee and P. Wesseling, Multigrid schemes for time-dependent incompressible Navier–Stokes equations, *Impact Comput. Sci. Engrg.* **5**, 153 (1993).
55. C. W. Oosterlee and P. Wesseling, A robust multigrid method for a discretization of the incompressible Navier–Stokes equations in general coordinates, *Impact Comput. Sci. Engrg.* **5**, 128 (1993).
56. C. W. Oosterlee and P. Wesseling, Steady incompressible flow around objects in general coordinates with a multigrid solution method, *Numer. Methods Partial Differential Equations* **10**, 295 (1994).
57. C. W. Oosterlee, P. Wesseling, A. Segal, and E. Brakkee, Benchmark solutions for the incompressible Navier–Stokes equations in general co-ordinates on staggered grids, *Internat. J. Numer. Methods Fluids* **17**, 301 (1993).
58. P. A. Forsyth, Jr., and P. H. Sammon, Quadratic convergence for cell-centered grids, *Appl. Numer. Math.* **4**, 377 (1988).
59. M. Perić, R. Kessler, and G. Scheuerer, Comparison of finite-volume numerical methods with staggered and collocated grids, *Comput. & Fluids* **16**, 389 (1988).
60. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow* (Springer-Verlag, Berlin, 1985).
61. C. M. Rhie and W. L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation, *AIAA J.* **21**, 1525 (1983).
62. W. Rodi, S. Majumdar, and B. Schönung, Finite volume methods for two-dimensional incompressible flows with complex boundaries, *Comput. Methods Appl. Mech. Engrg.* **75**, 369 (1989).
63. S. E. Rogers and D. Kwak, Upwind differencing scheme for the time-accurate incompressible Navier–Stokes equations, *AIAA J.* **28**, 253 (1990).
64. S. E. Rogers, D. Kwak, and U. Kaul, On the accuracy of the pseudocompressibility method in solving the incompressible Navier–Stokes equations, *Appl. Math. Modelling* **11**, 35 (1987).
65. S. E. Rogers, D. Kwak, and C. Kiris, Steady and unsteady solutions of the incompressible Navier–Stokes equations, *AIAA J.* **29**, 603 (1991).
66. H.-G. Roos, M. Stynes, and L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations* (Springer-Verlag, Berlin, 1996).
67. M. Rosenfeld, Validation of numerical simulation of incompressible pulsatile flow in a constricted channel, *Comput. & Fluids* **22**, 139 (1993).
68. M. Rosenfeld and D. Kwak, Time-dependent solution of viscous incompressible flows in moving coordinates, *Internat. J. Numer. Methods Fluids* **13**, 1311 (1991).
69. M. Rosenfeld, D. Kwak, and M. Vinokur, A fractional step solution method for the unsteady incompressible Navier–Stokes equations in generalized coordinate systems, *J. Comput. Phys.* **94**, 102 (1991).
70. Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving non-symmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
71. A. Segal, P. Wesseling, J. van Kan, C. W. Oosterlee, and K. Kassels, Invariant discretization of the incompressible Navier–Stokes equations in boundary fitted co-ordinates, *Internat. J. Numer. Methods Fluids* **15**, 411 (1992).
72. C. Sheng, L. K. Taylor, and D. L. Whitfield, Multigrid algorithm for three-dimensional high-Reynolds number turbulent flow, *AIAA J.* **33**, 2073 (1995).
73. W. Shyy and M. E. Braaten, *Adaptive Grid Computation for Inviscid Compressible Flows Using a Pressure Correction Method*, AIAA Paper 88-3566-CP, 1988.
74. W. Shyy, M.-H. Chen, and C.-S. Sun, Pressure-based multigrid algorithm for flow at all speeds, *AIAA J.* **30**, 2660 (1992).
75. W. Shyy, S. S. Tong, and S. M. Correa, Numerical recirculating flow calculation using a body fitted coordinate system, *Numer. Heat Transfer* **8**, 99 (1985).
76. W. Shyy and T. C. Vu, On the adoption of velocity variable and grid system for fluid flow computation in curvilinear coordinates, *J. Comput. Phys.* **92**, 82 (1991).
77. W. Y. Soh and J. W. Goodrich, Unsteady solution of incompressible Navier–Stokes equations, *J. Comput. Phys.* **79**, 113 (1988).
78. P. N. Swarztrauber, Fast Poisson solvers, in *Studies in Numerical Analysis*, edited by G. H. Golub (Mathematical Society of America, 1984), Vol. 24, pp. 319–370.

79. P. K. Sweby, High resolution schemes using flux-limiters for hyperbolic conservation laws, *SIAM J. Numer. Anal.* **21**, 995 (1984).
80. P. Tamamidis, G. Zhang, and D. N. Assanis, Comparison of pressure-based and artificial compressibility methods for solving 3D steady incompressible flows, *J. Comput. Phys.* **124**, 1 (1996).
81. R. Teman, *Navier–Stokes Equations: Theory and Numerical Analysis* (North-Holland, Amsterdam, 1977).
82. E. Turkel, Review of preconditioning techniques for fluid dynamics, *Appl. Numer. Math.* **12**, 257 (1993).
83. J. J. I. M. Van Kan, A second-order accurate pressure correction method for viscous incompressible flow, *SIAM J. Sci. Stat. Comput.* **7**, 870 (1986).
84. B. van Leer, W.-T. Lee, and P. L. Roe, *Characteristic Time-Stepping or Local Preconditioning of the Euler Equations*, AIAA Paper 91-1552, 1991.
85. M. Vinokur, An analysis of finite difference and finite-volume formulations of conservation laws, *J. Comput. Phys.* **81**, 1 (1989).
86. C. Vuik, Solution of the discretized incompressible Navier–Stokes equations with the GMRES method, *Internat. J. Numer. Methods Fluids* **16**, 507 (1993).
87. C. Vuik, Fast iterative solvers for the discretized incompressible Navier–Stokes equations, *Internat. J. Numer. Methods Fluids* **22**, 195 (1996).
88. A. Weiser and M. F. Wheeler, On convergence of block-centered finite differences for elliptic problems, *SIAM J. Numer. Anal.* **25**, 351 (1988).
89. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, Chichester, 1992).
90. P. Wesseling, Uniform convergence of discretization error for a singular perturbation problem, *Numer. Methods Partial Differential Equations* **12**, 657 (1996).
91. P. Wesseling, Von Neumann stability conditions for the convection-diffusion equation, *IMA J. Numer. Anal.* **16**, 583 (1996).
92. P. Wesseling, C. G. M. Kassels, C. W. Oosterlee, A. Segal, C. Vuik, S. Zeng, and M. Zijlema, Computing incompressible flows in general domains, in *Numerical Methods for the Navier–Stokes Equations*, edited by F.-K. Hebeker, R. Rannacher, and G. Wittum (Vieweg, Braunschweig, 1994), pp. 298–314.
93. P. Wesseling, A. Segal, J. J. I. M. van Kan, C. W. Oosterlee, and C. G. M. Kassels, Finite volume discretization of the incompressible Navier–Stokes equations in general coordinates on staggered grids, *Comput. Fluid Dynam. J.* **1**, 27 (1992).
94. P. Wesseling, P. van Beek, and R. R. P. van Nooyen, Aspects of non-smoothness in flow computations, in *Computational Methods in Water Resources X*, edited by A. Peters, G. Wittum, B. Herrling, U. Meissner, C. A. Brebbia, W. G. Gray, and G. F. Pinder (Kluwer Academic, Dordrecht, 1994), pp. 1263–1271.
95. P. Wesseling, M. Zijlema, A. Segal, and C. G. M. Kassels, Computation of turbulent flow in general domains, *Math. Comput. Simulation* **44**, 369 (1997).
96. Y. Zang, R. L. Street, and J. R. Koseff, A non-staggered grid, fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates, *J. Comput. Phys.* **114**, 18 (1994).
97. S. Zeng, C. Vuik, and P. Wesseling, Numerical solution of the incompressible Navier–Stokes equations by Krylov subspace and multigrid methods, *Adv. Comput. Math.* **4**, 27 (1995).
98. M. Zijlema, *Computational Modeling of Turbulent Flows in General Domains*, Ph.D. thesis, Delft University of Technology, The Netherlands, April 1996.
99. M. Zijlema, On the construction of a third-order accurate monotone convection scheme with application to turbulent flow in general coordinates, *Internat. J. Numer. Methods Fluids* **22**, 619 (1996).
100. M. Zijlema, A. Segal, and P. Wesseling, Finite volume computation of incompressible turbulent flows in general coordinates on staggered grids, *Internat. J. Numer. Methods Fluids* **20**, 621 (1995).
101. 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, *Comput. & Fluids* **24**, 209 (1995).
102. M. Zijlema and P. Wesseling, On accurate discretization of turbulence transport equations in general co-ordinates, in *Numerical Methods in Laminar and Turbulent Flow*, edited by C. Taylor and P. Durbetaki (Pineridge, Swansea, 1995), Vol. 9, Part 1, pp. 34–45.
103. M. Zijlema and P. Wesseling, Higher-order flux-limiting schemes for the finite volume computation of incompressible flow, *Internat. J. Comput. Fluid Dynam.* **9**, 89 (1998).