Applied Scientific Research **48**: 175–191, 1991.
© 1991 *Kluwer Academic Publishers. Printed in the Netherlands.*

The ISNaS incompressible Navier–Stokes solver: invariant discretization

A.E. MYNETT[1], P. WESSELING[2], A. SEGAL[2] & C.G.M. KASSELS[2]

[1]*Delft Hydraulics, Department of Research and Development, P.O. Box 177, 2600 MH Delft, The Netherlands;* [2]*Delft University of Technology, Department of Technical Mathematics and Informatics, P.O. Box 356, 2600 AJ Delft, The Netherlands*

Received 4 July 1990; accepted in revised form 3 November 1990

Abstract. The ISNaS-project aims at providing tools for computer aided design and engineering in aerodynamics and hydrodynamics by developing an Information System for the simulation of complex flows based on the Navier–Stokes equations. Major components of the project are the development of a method-shell and of accurate as well as robust solvers for both compressible and incompressible flows. For the incompressible case, guided by typical applications in the field of river and coastal hydrodynamics, a solution procedure is being developed that is capable of handling complicated geometries, including free surface effects, in particular for high-Reynolds number flow regimes. In the present paper the invariant discretization of the incompressible Navier-Stokes equations in general boundary-fitted coordinate systems is discussed. It is found to be important that certain rules are followed concerning the choice of unknowns and the approximation of the geometric quantities. This is illustrated by some preliminary results. Extensions to moving coordinate systems and time-varying computational grids are indicated.

1. Introduction

Before proceeding with the subject proper of this paper, a brief outline is given of the larger context of our research. Complex flow problems are encountered in many branches of engineering science. Although the basic formulation of fluid flow phenomena was already presented over a century ago by Navier and Stokes, the solution of the resulting equations could only be obtained for idealised conditions and simple geometries. However, with the advent of recent high speed (super)computers more realistic (complex) applications are coming within reach. This implies that more complicated geometries and more realistic flow conditions can be handled, at least in principle. This also implies, however, that more and more effort is required to formulate, program, test and maintain the software involved. This applies to all kinds of fluid flows and is of major concern.

In order to combine efforts in The Netherlands, the ISNaS-project was initiated by the National Aerospace Laboratory NLR, Delft Hydraulics, Delft University of Technology and Twente University, co-funded by the Dutch government. The ISNaS-project aims at providing tools for computer aided design and engineering by developing an Information System for flow simulation based on the Navier–Stokes equations. Details on background, scope and conceptual design are described in [12]. The two major components of the ISNaS-project are the development of a method-shell consisting of a method base, a method manager and an executive, and of accurate as well as robust solvers for both compressible and incompressible flows.

The purpose of the method shell is to provide tools for management and execution of computer programs. Different program components, supplemented with an extensive description of their functions and aspects of implementation, are stored in a method base system (MEBAS) which supports developers in composing application programs, ready for execution by end-users, by combining the appropriate program components. Implementation of component programs is handled by a method manager, enabling the user to access, compose and execute the methods in functional terms only. MEBAS is specifically designed for use in incremental software development and for support in algorithmic research. The executive ISNEX supports users in executing programs that are composed via MEBAS, on a distributed network of computer systems, keeping track of scheduling and data transfer. A description of the method base system and executive as tools for management and execution of software systems in computer aided engineering is given in [5].

The development of the compressible flow solver for aerodynamic applications is discussed in [2]. The present paper focusses on the incompressible flow solver. It has been decided to use a staggered finite volume discretization on a boundary-fitted grid, in a coordinate-invariant formulation. The purpose of this paper is to present this formulation. The choice between the various alternatives (finite volume or finite element, staggered or non-staggered grid, Cartesian or contravariant velocity components) is not clear-cut. It would lead too far to present here the considerations that have led to the present choice. Suffice it to say that the invariant discretization of physical conservation laws such as the Navier–Stokes equations is in itself a fundamental and interesting subject, which has not yet received much attention.

2. General curvilinear coordinates

In order to be able to handle fluid flow problems of engineering interest in domains of irregular geometry, an appropriate coordinate system has to be defined. General curvilinear non-orthogonal grids, suitable to deal with quite arbitrarily shaped flow domains and allowing adequate control of the computational grids, will be used in the ISNaS-project. A distinction can be made between methods which employ grid-oriented (contravariant) velocity components and methods which are based on Cartesian velocity components. The choice is not at all trivial and indeed strong arguments can be made for either of the two methods. Both formulations have received considerable attention in the ISNaS project. The present paper follows the contravariant approach.

Boundary-fitted computational grids are used, which implies that the boundaries of the solution domain coincide with grid lines. The physical domain is mapped onto a computational domain consisting of a number of rectangular blocks. In this way the accurate implementation of boundary conditions is substantially simplified and efficient use can be made of vector and parallel computers. Here we restrict ourselves to the one block case. It is assumed that admissible transformations $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ and $\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x})$ exist between the two domains, which implies that the Jacobians of the coordinate transformations do not vanish; $\mathbf{x}$ are Cartesian coordinates, $\boldsymbol{\xi}$ are boundary-conforming curvilinear coordinates.

Covariant base vectors $\mathbf{a}_{(\alpha)}$ are defined as tangent vectors to the curvilinear coordinate lines $\xi^\alpha = \text{constant}$, i.e.

$$\mathbf{a}_{(\alpha)} = \frac{\partial \mathbf{x}}{\partial \xi^\alpha}. \tag{2.1}$$

Contravariant base vectors $\mathbf{a}^{(\alpha)}$ are defined as normal vectors to the surfaces on which ξ^α is constant, i.e.

$$\mathbf{a}^{(\alpha)} = \frac{\partial \xi^\alpha}{\partial \mathbf{x}}. \tag{2.2}$$

We have $\mathbf{a}_{(\alpha)} \cdot \mathbf{a}^{(\beta)} = \delta^\beta_\alpha$ with δ the Kronecker delta. The covariant and contravariant metric tensors $g_{\alpha\beta}$ and $g^{\alpha\beta}$ are defined by

$$g_{\alpha\beta} = \mathbf{a}_{(\alpha)} \cdot \mathbf{a}_{(\beta)}; \qquad g^{\alpha\beta} = \mathbf{a}^{(\alpha)} \cdot \mathbf{a}^{(\beta)}. \tag{2.3}$$

The determinant of the covariant metric tensor $g_{\alpha\beta}$ is denoted by g; $\sqrt{g}$ equals the Jacobian of the transformation, given by

$$J = \sqrt{g} = \mathbf{a}_{(1)} \cdot (\mathbf{a}_{(2)} \wedge \mathbf{a}_{(3)}). \tag{2.4}$$

Tensor notation proves indispensable for formulating physical conservation laws in general coordinates. An introduction to tensor analysis can be found in for example [1], [6], [7], [10]. For completeness we summarize some basic facts. Tensors are induced transformations that are isomorphic to transformations of coordinates, i.e. tensors are mathematical objects that are independent of the coordinate system. We will only consider tensors of rank zero (scalars), one (vectors) or two. An example is a mixed relative tensor Q^α_β of weight w and rank two with Cartesian components q^γ_δ, satisfying the following transformation law:

$$Q^\alpha_\beta = (\sqrt{g})^w a^{(\alpha)}_\gamma a^\delta_{(\beta)} q^\gamma_\delta, \tag{2.5}$$

where Q^α_β are the components in the curvilinear coordinate system. Relative tensors of weight zero are called absolute; relative tensors of weight one are called densities.

A covariant derivative is a tensor which reduces to a partial derivative of a vector field in Cartesian coordinates. For an absolute scalar σ, the covariant derivative is identical to the partial derivative, and is denoted by

$$\sigma_{,\alpha} = \frac{\partial\sigma}{\partial\xi^{\alpha}}. \tag{2.6}$$

The fluid density ρ is a tensor of weight 1 (a density). This means that ρ is not invariant under coordinate transformation, but $\sqrt{g}\,\rho$ is. Its covariant derivative is given by

$$\rho_{,\alpha} = \frac{\partial\rho}{\partial\xi^{\alpha}} - \left\{\begin{matrix}\beta\\ \beta\alpha\end{matrix}\right\}\rho = \sqrt{g}\,\frac{\partial}{\partial\xi^{\alpha}}\left(\frac{1}{\sqrt{g}}\,\rho\right), \tag{2.7}$$

where $\left\{{}^{\alpha}_{\gamma\beta}\right\}$ represents the Christoffel symbol of the second kind, defined by

$$\left\{\begin{matrix}\alpha\\ \gamma\beta\end{matrix}\right\} = \mathbf{a}^{(\alpha)}\cdot\frac{\partial\mathbf{a}_{(\gamma)}}{\partial\xi^{\beta}} = \frac{\partial\xi^{\alpha}}{\partial x^{\delta}}\,\frac{\partial^{2}x^{\delta}}{\partial\xi^{\gamma}\partial\xi^{\beta}}. \tag{2.8}$$

The covariant derivative of a contravariant tensor of rank 1 is defined by

$$Q^{\alpha}_{,\beta} = \frac{\partial Q^{\alpha}}{\partial\xi^{\beta}} + \left\{\begin{matrix}\alpha\\ \gamma\beta\end{matrix}\right\}Q^{\gamma}. \tag{2.9}$$

The covariant derivative of a contravariant tensor of rank 2 is defined by

$$Q^{\alpha\beta}_{,\gamma} = \frac{\partial Q^{\alpha\beta}}{\partial\xi^{\gamma}} + \left\{\begin{matrix}\alpha\\ \delta\gamma\end{matrix}\right\}Q^{\delta\beta} + \left\{\begin{matrix}\beta\\ \delta\gamma\end{matrix}\right\}Q^{\alpha\delta}. \tag{2.10}$$

It can be shown that

$$Q^{\alpha\beta}_{,\beta} = \frac{1}{\sqrt{g}}\,\frac{\partial\sqrt{g}\,Q^{\alpha\beta}}{\partial\xi^{\beta}} + \left\{\begin{matrix}\alpha\\ \gamma\beta\end{matrix}\right\}Q^{\gamma\beta}. \tag{2.11}$$

In tensor notation the divergence theorem is given by

$$\int_{\Omega} Q^{\alpha}_{,\alpha}\,d\Omega = \oint_{S} Q^{\alpha}\,dS_{\alpha}, \tag{2.12}$$

where $d\Omega$ is the infinitesimal volume element given by

$$d\Omega = \sqrt{g}\,d\xi^{1}d\xi^{2}\cdots d\xi^{d}, \tag{2.13}$$

d being the number of spatial dimensions, and dS_{α} represents the (physical) surface element. A fundamental geometric identity can be derived by applying the divergence theorem to a constant vector field which gives

$$\int_{\Omega} Q^{\alpha}_{,\alpha} d\Omega = \oint_{S} Q^{\alpha} dS_{\alpha} = q^{\beta} \oint_{S} a^{(\alpha)}_{\beta} dS_{\alpha} = 0. \tag{2.14}$$

Since q^{β} is arbitrary, this leads to the following geometric identity:

$$\oint_{S} a^{(\alpha)}_{\beta} dS_{\alpha} = 0. \tag{2.15}$$

It is important to satisfy this identity also numerically, in order to prevent conservation errors in the numerical solution.

The governing equations are the incompressible Navier–Stokes equations. Using contravariant tensor components for the fluid density, velocities, viscous stresses and pressure, the coordinate-invariant formulation of the governing equations becomes

$$U^{\alpha}_{,\alpha} = 0 \tag{2.16}$$

and

$$\frac{\partial}{\partial t}(\rho U^{\alpha}) + (\rho U^{\alpha} U^{\beta})_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} = \rho f^{\alpha}, \tag{2.17}$$

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

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma} U^{\beta}_{,\gamma} + g^{\gamma\beta} U^{\alpha}_{,\gamma}), \tag{2.18}$$

with μ the (laminar or turbulent) viscosity and ρ the fluid density. The convection-diffusion equation for a scalar T is given by

$$c^* \frac{\partial T}{\partial t} + (U^{\alpha} T)_{,\alpha} - (K^{\alpha\beta} T_{,\beta})_{,\alpha} + D^* T = f^*, \tag{2.19}$$

where c^*, $K^{\alpha\beta}$, D^* and f^* are given functions.

3. Invariant finite volume discretization

Invariant (i.e. invariant with respect to change of coordinates) formulations and discretizations of physical conservation laws are complicated at first sight, and costly. Therefore most formulations are in special coordinate systems, or not completely invariant. For example, Cartesian vector components are used as scalar unknowns. The accuracy and stability of such approaches is sufficient for practice only if restrictions are placed on the mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$. Both from a practical and a fundamental point of view it seems attractive to develop general invariant discretizations of invariant formulations of physical conservation laws. To the authors' knowledge, the only publication where this is done for the incompressible Navier–Stokes equations is [4], using the Gibbs notation. We will adhere to standard tensor notation. A useful general discussion of invariant formulations and discretizations of conservation laws is given in [11].

For simplicity we restrict ourselves to the two-dimensional case. In order to obtain a stable discretization without introducing artificial stability terms, a staggered grid is used. For a comparison between the staggered and the non-staggered approach, see [3]. Figure 3.1 shows part of the staggered grid in the physical plane. This is the image under the mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ of a uniform grid in the $\boldsymbol{\xi}$-plane. The pressure is computed in cell centers (●); U^{α} is computed in the centers of cell faces connecting vertices with equal values of ξ^{α} (U^1 in (→) and U^2 in (↑)). The symbols → and ↑ are used for typographical convenience only and are not meant to indicate the direction of a vector. It is assumed that the only geometric information available is the value of $\mathbf{x}(\boldsymbol{\xi})$ in the cell vertices. The other geometric quantities ($\sqrt{g}, \mathbf{a}_{(\alpha)}$ etc.) have to be deduced carefully. For accuracy reasons, the following requirements should be met:

(i) When representing a constant velocity field $\mathbf{u}$ on the staggered grid in terms of its contravariant components U^{α}, and recomputing $\mathbf{u}$ from U^{α}, the original vector field $\mathbf{u}$ should be recovered exactly.

(ii) The geometric identity (2.15) should be satisfied exactly for all cells.

These requirements can be met if:

(i) Instead of the velocities U^{α}, the fluxes $V^{\alpha} = \sqrt{g}\, U^{\alpha}$ are used as primary unknowns.

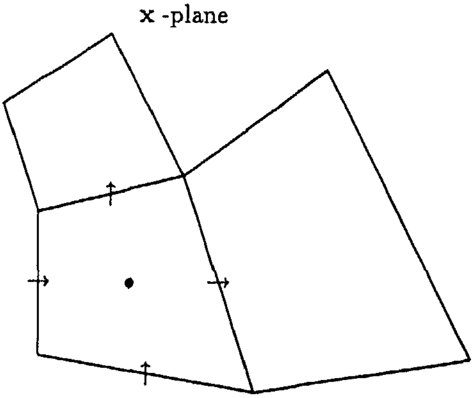

Fig. 3.1. Staggered curvilinear grid.

(ii) The geometric quantities are computed as follows. The base vectors $\mathbf{a}_{(\alpha)}$ are computed according to

$$a^{\beta}_{(1)} = \frac{\delta x^{\beta}}{\delta \xi^{(1)}}, \qquad a^{\beta}_{(2)} = \frac{\delta x^{\beta}}{\delta \xi^{2}} \tag{3.1}$$

in the U^2- and U^1-points respectively. Here δ implies taking differences between the points where $\mathbf{x}(\xi)$ is given (i.e. the cell vertices) in the obvious way. Furthermore,

$$\sqrt{g} = a^{1}_{(1)}a^{2}_{(2)} - a^{2}_{(1)}a^{1}_{(2)} \tag{3.2}$$

taking averages where required. Finally,

$$\mathbf{a}^{(1)} = \frac{1}{\sqrt{g}}(a^{2}_{(2)}, -a^{1}_{(2)}), \qquad \mathbf{a}^{(2)} = \frac{1}{\sqrt{g}}(-a^{2}_{(1)}, a^{1}_{(1)}) \tag{3.3}$$

again taking averages where required.

For convenience we introduce the local cell coordinates given by Fig. 3.2, which shows part of the computational grid in the ξ-plane. Integration of the incompressibility constraint over a pressure cell with center at (0, 0) gives

$$\int_{\Omega} U^{\alpha}_{,\alpha} d\Omega = \oint_{S} U^{\alpha} dS_{\alpha}. \tag{3.4}$$

Let $\delta\mathbf{S}^{(\alpha)}$ be the vector with direction along the outward normal to a face with a U^{α} point as center, and with length equal to the length of that face. Hence

$$\delta\mathbf{S}^{(\alpha)} = \sqrt{g}\,\mathbf{a}^{(\alpha)}\delta\xi^{\beta}, \quad \alpha, \beta \text{ cyclic}. \tag{3.5}$$

Computing $\mathbf{a}^{(\alpha)}$ according to (3.1), (3.3) implies that effectively in the $\mathbf{x}$-plane the faces of the cells are taken to be straight (hence they are curved in the ξ-plane; they have

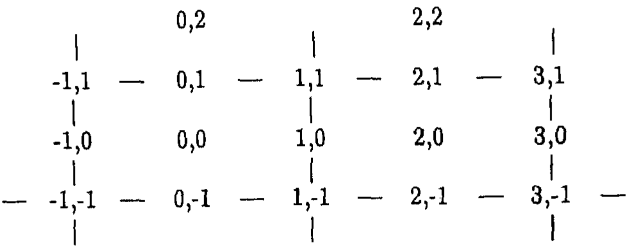

Fig. 3.2. Local cell coordinates.

been drawn straight in Fig. 3.2 merely for convenience). To explain the discretization of the integral along S in (3.4) it suffices to consider the integral along the face with vertices (1, −1) and (1, 1). We have

$$\int_{1,-1}^{1,1} U^\alpha dS_\alpha \cong (U^\alpha \delta S_\alpha^{(1)})|_{1,0}. \tag{3.6}$$

We have $\delta S_\alpha^{(1)} = a_{(\alpha)}^\beta \sqrt{g}\, a_\beta^{(1)} \delta\xi^2 = \sqrt{g}\, \delta\xi^2$ since $a_{(\alpha)}^\beta a_\beta^{(\gamma)} = \delta_\alpha^\gamma$. Hence

$$\int_{1,-1}^{1,1} U^\alpha dS_\alpha \cong V_{(1,0)}^\alpha \delta\xi^2$$

and

$$\oint_S U^\alpha dS_\alpha \cong V^1|_{-1,0}^{1,0} \delta\xi^2 + V^2|_{0,-1}^{0,1} \delta\xi^1. \tag{3.7}$$

Let $\mathbf{u}$ be a constant vector field. Substituting $V^\alpha = \sqrt{g}\, a_\beta^{(\alpha)} u^\beta$ and using (3.1) and (3.2) one finds that

$$V^1|_{-1,0}^{1,0} \delta\xi^2 + V^2|_{0,-1}^{0,1} \delta\xi^1 = 0, \tag{3.8}$$

so that requirement (ii) is satisfied. Requirement (i) is verified as follows. Let $\mathbf{w}$ be a constant vector field. Its representation in terms of V^α on the staggered grid is $V^\alpha = \sqrt{g}\, a_\beta^{(\alpha)} w^\beta$. Hence, using (3.3),

$$V^1 = a_{(2)}^2 w^1 - a_{(2)}^1 w^2, \qquad V^2 = -a_{(1)}^2 w^1 + a_{(1)}^1 w^2. \tag{3.9}$$

Now recompute the Cartesian components u^α from (3.9) in the cell vertices:

$$u^\alpha \Big|_{1,1} = \frac{1}{2\sqrt{g}} \Big|_{1,1} \left\{ \sum_1 (a_{(1)}^\alpha V^1) + \sum_2 (a_{(2)}^\alpha V^2) \right\}, \tag{3.10}$$

where Σ_1 indicates summation over grid points (1,0) and (1,2), and Σ_2 indicates summation over (0, 1) and (2, 1). Substitution of (3.9) in (3.10), and evaluation of $\sqrt{g}$ according to (3.2) results in

$$u^\alpha|_{1,1} = w^\alpha|_{1,1}. \tag{3.11}$$

We also have (3.11) in cell centers (0, 0). Hence, requirement (i) is satisfied. If U^α is used as primary unknown instead of V^α (3.11) would not hold exactly, which is why the use of V^α is to be preferred.

As a preliminary to the discretization of the momentum equation we first discuss the discretization of a general conservation law of the form

$$T^{\alpha\beta}_{,\beta} = f^\alpha. \tag{3.12}$$

This equation is to be integrated over finite volumes. On the staggered grid used here, integration takes place over cells with vertices in U^2-points and center in a U^1-point for $\alpha = 1$, and vice-versa for $\alpha = 2$. Taking a cell with center at (1,0) as an example, equations (2.11) and (3.12) and partial integration gives

$$\begin{aligned}\int_\Omega T^{1\beta}_{,\beta} d\Omega &= \int_\Omega \frac{\partial\sqrt{g}\,T^{1\beta}}{\partial\xi^\beta} d\xi^1 d\xi^2 + \int_\Omega \begin{Bmatrix} 1 \\ \gamma\beta \end{Bmatrix} T^{\gamma\beta}\sqrt{g}\,d\xi^1 d\xi^2 \\ &\cong (\sqrt{g}\,T^{11})|^{2,0}_{0,0}\delta\xi^2 + (\sqrt{g}\,T^{12})|^{1,1}_{1,-1}\delta\xi^1 + \left(\sqrt{g}\begin{Bmatrix} 1 \\ \gamma\beta \end{Bmatrix} T^\gamma\beta|_{1,0}\right)\delta\xi^1\delta\xi^2.\end{aligned} \tag{3.13}$$

This method of discretization we will call variant 1. Christoffel symbols occur. These are computed in the obvious way by computing $\mathbf{a}^{(\alpha)}$ and $\mathbf{a}_{(\alpha)}$ in the way described before, and taking differences. The Christoffel symbols involve second derivatives of the mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$, which may not be approximated accurately by the method just described (nevertheless, as will be seen, results with variant 1 look promising). The Christoffel symbols can be avoided by the following more subtle approach, which we will call variant 2. Variant 2 is equivalent to the discretization presented in [4]. We contract (3.12) with linearly independent vector fields $\mathbf{w}^{(\gamma)}$, $\gamma = 1, 2$, before finite volume integration. In order to prevent the occurrence of Christoffel symbols, $\mathbf{w}^{(\gamma)}$ should be constant. In order to obtain stable discretizations in general, $\mathbf{w}^{(\gamma)}$ should be chosen differently in each cell. We define $\mathbf{w}^{(\gamma)}$ in the center of the finite volume under consideration by

$$W^{(\gamma)}_\alpha = \delta^{(\gamma)}_\alpha, \tag{3.14}$$

with δ the Kronecker delta, and extend the definition of $\mathbf{w}^\gamma$ over the whole domain by imposing $\mathbf{w}^\gamma$ = constant. Integration takes place over cells with vertices in the U^2-points and center in a U^1-point with $\gamma = 1$, and over cells with vertices in U^1-points and center in a U^2-point with $\gamma = 2$. This gives, taking a cell with center at (1,0) as an example,

$$\begin{aligned}\int_\Omega W^{(1)}_\alpha T^{\alpha\beta}_{,\beta} d\Omega &= \int_\Omega (W^{(1)}_\alpha T^{\alpha\beta})_{,\beta} d\Omega \\ &\cong (\sqrt{g}\,W^{(1)}_\alpha T^{\alpha 1})|^{2,0}_{0,0}\delta\xi^2 + (\sqrt{g}\,W^{(1)}_\alpha T^{\alpha 2})|^{1,1}_{1,-1}\delta\xi^1.\end{aligned} \tag{3.15}$$

We have

$$W_{\alpha}^{(1)}|_{0,0} = (a_{(\alpha)}^{\beta} w_{\beta}^{(1)})|_{0,0} = a_{(\alpha)}^{\beta}|_{0,0} w_{\beta}^{(1)}|_{1,0}$$
$$= a_{(\alpha)}^{\beta}|_{0,0}(a_{\beta}^{(\gamma)} W_{\gamma}^{(1)})|_{1,0} = a_{(\alpha)}^{\beta}|_{(0,0)} a_{\beta}^{(1)}|_{1,0}. \tag{3.16}$$

Note that Christoffel symbols do not appear, and that the mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ is differentiated only once to obtain $\mathbf{a}_{(\alpha)}$ and $\mathbf{a}^{(\gamma)}$.

Variants 1 and 2 are applied as follows to the momentum equation (2.17), abbreviated as

$$\frac{\partial \rho U^{\alpha}}{\partial t} + T^{\alpha\beta}_{,\beta} = \rho f^{\alpha}. \tag{3.17}$$

Integration over a finite volume of $T^{\alpha\beta}_{,\beta}$ is done according to (3.13), or to (3.15) and (3.16). Next, $T^{\alpha\beta}$ has to be approximated. The convection and pressure parts are easily obtained, averaging where necessary. In the viscous part covariant derivatives occur. These are handled as follows. If suffices to discuss variant 2. As an example we take again a cell with center at (1, 0), and consider the term $(\sqrt{g}\, W_{\alpha}^{(1)} \mu g^{\alpha\gamma} U^{1}_{,\gamma})|_{0,0}$. Let now Ω be the pressure cell with center in (0, 0). We write

$$(\sqrt{g}\, W_{\alpha}^{(1)} \mu g^{\alpha\gamma} U^{1}_{,\gamma})|_{0,0} \cong \frac{\mu_{0,0}}{|\Omega|} \int_{\Omega} (\sqrt{g}\, W_{\alpha}^{(1)} g^{\alpha\gamma} U^{1})_{,\gamma} d\Omega$$
$$= \frac{\mu_{0,0}}{|\Omega|} \oint_{S} W_{\alpha}^{(1)} g^{\alpha\gamma} V^{1} dS_{\gamma}, \tag{3.18}$$

where we have used the fact that $\sqrt{g}_{,\gamma} = 0$ and $g^{\alpha\gamma}_{,\gamma} = 0$. The remaining contour integral is discretized in a similar way as before. Figure 3.3 shows for variant 2 the structure of the resulting stencils; (a), (b), (c) and (d) refer to the continuity equation and the contributions of the inertia, pressure and viscous terms, respectively. The stencils for the U^2-momentum equation are obtained from those for the U^1-equation by rotation over 90°. The total number of variables linked together in a momentum equation is 23. Variant 1 results in slightly smaller stencils, not given here.

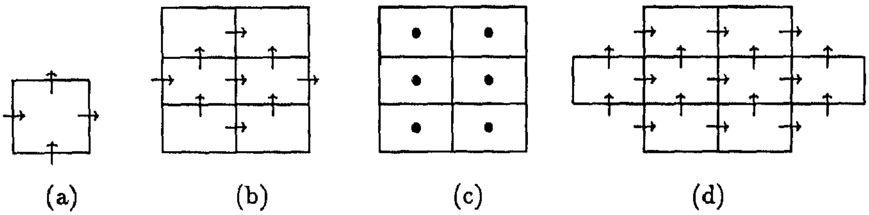

Fig. 3.3. Stencil structure.

Discretization in time can be done by any of the usual methods. We have

$$\int_\Omega W^{(\gamma)}_\alpha U^\alpha d\Omega \cong (\sqrt{g}\, W^{(\gamma)}_\alpha U^\alpha)_C \delta\xi^1 \delta\xi^2 = V^\gamma_C \delta\xi^1 \delta\xi^2, \tag{3.19}$$

where C is the center of Ω. A possible time discretization is

$$V^{\gamma,n+1}_C - V^{\gamma,n}_C = \frac{\delta t}{\delta\xi^1 \delta\xi^2} \int_\Omega \{W^{(\gamma}_\alpha(\Theta T^{\alpha\beta,n+1}_{,\beta} + (1-\Theta) T^{\alpha\beta,n}_{,\beta} \\ + \rho(\Theta f^{\alpha,n+1} + (1-\Theta) f^{\alpha,n})\} d\Omega. \tag{3.20}$$

For stability we take $\Theta \geqslant 1/2$.

Details about the implementation of boundary conditions are not given here. We restrict ourselves to noting that if the grid is orthogonal at the boundary, the situation is similar to the Cartesian case. Since it is possible to generate boundary-orthogonal grids in two or three dimensions, this may be the most attractive way to proceed.

The resulting nonlinear system of equations is solved approximately by the well-known pressure-correction method ([13]), as follows. We abandon tensor notation. Let π be the vector containing all pressure unknowns, and let σ contain all V^α unknowns. The algebraic system can be written as

$$\sigma^{n+1} = \sigma^n + N(\sigma^{n+1}) + \Theta D_1 \pi^{n+1} + f^n, \qquad D_2 \sigma^{n+1} = 0, \tag{3.21}$$

where N is a nonlinear operator, D_1 and D_2 are linear operators, and f^n is a known source term. The pressure correction method is used. First σ^* is determined from

$$\sigma^* = \sigma^n + N(\sigma^*) + \Theta D_1 \pi^n + f^n. \tag{3.22}$$

To determine σ^* one Newton iteration is carried out with σ^n as initial guess. The resulting linear system is solved with a conjugate gradient type iterative method. Next, $\delta\pi^{n+1}$ and σ^{n+1} are determined from

$$\sigma^{n+1} = \sigma^* + \Theta D_1 \delta\pi^{n+1}, \qquad D_2 \sigma^{n+1} = 0. \tag{3.23}$$

This gives

$$D_2 D_1 \delta\pi^{n+1} = -\frac{1}{\Theta} D_2 \sigma^*. \tag{3.24}$$

This equation is also solved by a conjugate gradient method. Finally one puts

$$\pi^{n+1} = \pi^n + \delta\pi^{n+1}. \tag{3.25}$$

4. Time-varying computational grids

In the previous sections, curvilinear boundary-conforming structured grids were considered, that are stationary in time. In this section the implications of a moving coordinate system and time-varying three-dimensional computational grids for grid generation and discretization are indicated.

Since in general it is not possible to map an arbitrary three-dimensional domain onto a single cube, a given domain is first divided into subdomains, each of which is topologically equivalent to a cube. This is known as domain decomposition. In this way, quite complicated geometries can be handled and smooth computational grids can be constructed. For a survey of domain decomposition, see [8], [9]. Within each subdomain a structured computational grid is generated. Details on the foundation and application of numerical grid generation can be found in [10].

In many hydrodynamic applications the free fluid surface plays an important role. Sometimes changes in the fluid surface are induced by the flow itself, e.g. by the Bernoulli-effect for flow over a weir, sometimes a moving fluid surface is caused by external forces, e.g. in case of wind-generated waves. Since the free surface boundary conditions have to be specified at the free surface position – which may be part of the solution – special treatment is required to handle free surface flow problems.

One way to model a moving fluid surface is to use so-called adaptive computational grids where the surface motion is either prescribed or governed by the fluid flow. In both cases it is necessary to re-compute the positions of the grid points at various time intervals. Moreover, an adaptive moving mesh method will introduce additional terms in the transformed equations of motion due to the time-varying grid point locations. The relative importance of these terms depends on the length and time scales of the physical problem at hand. It is assumed here that they have to be included.

The governing equations expressing conservation of mass and momentum in a stationary reference frame are given by (2.16) and (2.17). In a moving reference frame, the grid point locations vary in time and hence Christoffel symbols related to time-varying grid point locations appear. To account for the time coordinate we introduce the four-vector (ξ^0, ξ^α) where the index 0 refers to the time coordinate $\xi^0 = t$. The additional Christoffel symbols related to time derivatives can be derived from the definition (2.8) which gives

$$\begin{Bmatrix} 0 \\ \gamma\beta \end{Bmatrix} = \frac{\partial t}{\partial x^\delta} \frac{\partial^2 x^\delta}{\partial \xi^\gamma \partial \xi^\beta} = 0, \tag{4.1}$$

$$\begin{Bmatrix} \alpha \\ \gamma 0 \end{Bmatrix} = \begin{Bmatrix} \alpha \\ 0\gamma \end{Bmatrix} = \frac{\partial \xi^\alpha}{\partial \xi^\delta} \frac{\partial^2 x^\delta}{\partial \xi^\gamma \partial t} = \frac{\partial \xi^\alpha}{\partial x^\delta} \frac{\partial \dot{x}^\delta}{\partial \xi^\gamma} = \dot{X}^\delta, \tag{4.2}$$

$$\begin{Bmatrix} \alpha \\ 00 \end{Bmatrix} = \frac{\partial \xi^\alpha}{\partial x^\delta} \frac{\partial^2 x^\delta}{\partial t^2} = \frac{\partial \xi^\alpha}{\partial x^\delta} \ddot{x}^\delta = \ddot{X}^\delta, \tag{4.3}$$

where $\dot{x}^\delta$ and $\ddot{x}^\delta$ are the Cartesian grid point velocities and accelerations, $\dot{X}^\delta$ and $\ddot{X}^\delta$ being the associated contravariant components. The contravariant grid point velocity is readily obtained by invoking

$$\mathbf{a}_{(0)} \cdot \mathbf{a}^{(\alpha)} = \frac{\partial \xi^\alpha}{\partial t} + \frac{\partial x^\delta}{\partial t} \frac{\partial \xi^\alpha}{\partial x^\delta} = 0, \tag{4.4}$$

which gives

$$\frac{\partial \xi^\alpha}{\partial t} = -\dot{x}^\delta \frac{\partial \xi^\alpha}{\partial x^\delta} = -\dot{X}^\delta. \tag{4.5}$$

The fundamental metric identity given by (2.15) in a stationary reference frame can also be extended to a moving coordinate system by applying the divergence theorem to a constant vector field

$$\int_\Omega Q^0_{,0} d\Omega + \int_\Omega Q^\alpha_{,\alpha} d\Omega = 0, \tag{4.6}$$

which, by virtue of the divergence theorem, gives

$$\int_\Omega Q^0_{,0} d\Omega + \oint_S Q^\alpha dS_\alpha = 0. \tag{4.7}$$

Analogous to (2.14) the expression

$$q^0 \left\{ \frac{\partial \Omega}{\partial t} - \oint_S \dot{x}^\alpha dS_\alpha \right\} + q^\beta \oint_S a^{(\alpha)}_\beta dS_\alpha = 0 \tag{4.8}$$

should hold for an arbitrary four-vector (q^0, q^β) which implies the following geometric identities in a time varying coordinate system:

$$\frac{\partial \Omega}{\partial t} = \oint_S \dot{x}^\alpha dS_\alpha; \qquad \oint_S a^{(\alpha)}_\beta dS_\alpha = 0. \tag{4.9}$$

Expressions (4.1), (4.2), (4.3) and (4.9) can be used to formulate the governing equations in a moving curvilinear coordinate system. If the contravariant velocities relative to the moving curvilinear coordinate system are used as unknowns, the governing equations take the following form

$$U^\alpha_{,\alpha} + \dot{X}^\alpha_{,\alpha} = 0, \tag{4.10}$$

$$\frac{\partial}{\partial t}(\rho U^\alpha) + (\rho U^\alpha U^\beta)_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau^{\alpha\beta}_{,\beta} + \left\{ \begin{matrix} \alpha \\ 00 \end{matrix} \right\} \rho + 2 \left\{ \begin{matrix} \alpha \\ 0\beta \end{matrix} \right\} \rho U^\beta = \rho f^\alpha, \tag{4.11}$$

where grid point velocities and grid point accelerations appear because of (4.2) and (4.3). The need to compute the grid point accelerations can be avoided by using, instead of the contravariant velocities U^α relative to the moving curvilinear coordinate system, the contravariant components $\hat{U}^\alpha$ relative to the stationary grid, given by

$$\hat{U}^\alpha = U^\alpha + \dot{X}^\alpha. \tag{4.12}$$

The incompressibility constraint is again given by

$$\hat{U}^\alpha_{,\alpha} = 0, \tag{4.13}$$

but the momentum equations now become

$$\begin{aligned}\frac{1}{\sqrt{g}}\frac{\partial}{\partial t}(\sqrt{g}\,\rho U^\alpha) &+ \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi^\alpha}(\sqrt{g}\,\rho \hat{U}^\alpha(\hat{U}^\beta - \dot{X}^\beta) - \sqrt{g}\,\hat{\tau}^{\alpha\beta}) \\ &+ \begin{Bmatrix}\alpha\\ \gamma\beta\end{Bmatrix}(\rho\hat{U}^\gamma(\hat{U}^\beta - \dot{X}^\beta) - \hat{\tau}^{\gamma\beta}) + \frac{1}{\sqrt{g}}\frac{\partial}{\partial \xi^\alpha}(\sqrt{g}\,\rho) + \begin{Bmatrix}\alpha\\ \gamma\beta\end{Bmatrix}p \\ &+ 3\begin{Bmatrix}\alpha\\ 0\beta\end{Bmatrix}\rho(\hat{U}^\beta - \dot{X}^\beta) = \rho f^\alpha,\end{aligned} \tag{4.14}$$

where the deviatoric stress tensor is given by

$$\hat{\tau}^{\alpha\beta} = \mu(\hat{g}^{\alpha\gamma}\hat{U}^\beta_{,\gamma} + \hat{g}^{\gamma\beta}\hat{U}^\alpha_{,\gamma}) \tag{4.15}$$

and the metric tensor by

$$\hat{g}^{\alpha\beta} = g^{\alpha\beta} - \dot{X}^\alpha\dot{X}^\beta, \tag{4.16}$$

and only locations and velocities of the grid points appear, and no accelerations. These equations can be used as a starting point for the numerical discretization in very much the same way as described above for a stationary computational grid. However, the moving grid point positions and velocities in the interior of the computational domain have to be computed by a grid generator from the prescribed or computed values along the boundaries. This implies a direct coupling between the grid generator and the flow solver.

5. Preliminary results

In this section some preliminary results will be presented obtained with variant 1. The domain and the grid are given in Fig. 5.1. We have an L-shaped channel. This geometry has been chosen because it leads to rather abrupt variations in grid

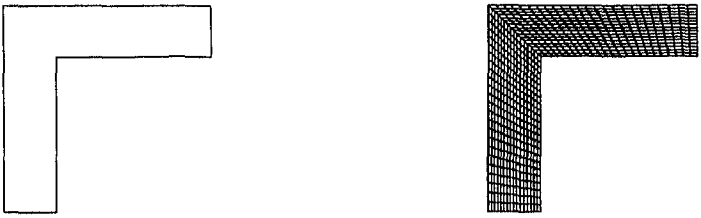

Fig. 5.1. Domain and grid.

geometry and solution, thus presenting a severe test for discretization methods. First, a uniform flow field is prescribed at the boundaries in an arbitrary direction. The exact solution is a uniform flow field. The exact solution is reproduced exactly by the numerical method. This is a nice feature, which many general coordinate codes lack. Next, a parabolic velocity profile is prescribed at inflow and the shear stress component and tangential velocity component are put zero at outflow. The resulting streamlines are given in Fig. 5.2. The Reynolds number is based on the width of the channel. This solution shows good agreement with results obtained with a standard finite element code. Figure 5.2 also gives results with a modification of variant 1, in which, contrary to the recommendations of Section 3, U^α are used as unknowns instead of V^α. On this coarse mesh a significantly different solution is obtained which deviates appreciably from finite element results. It may be concluded that the recommendations of Section 3 should be followed, and that variant 1 provides a robust and accurate discretization on coarse and non-uniform grids. Results for variant 2 are not yet available.

6. Concluding remarks

A brief outline has been given of the ISNaS-project, which aims at providing tools for computer aided design and engineering by developing an information system for flow simulation based on the Navier–Stokes equations. The main topic discussed is the discretization for incompressible flow.

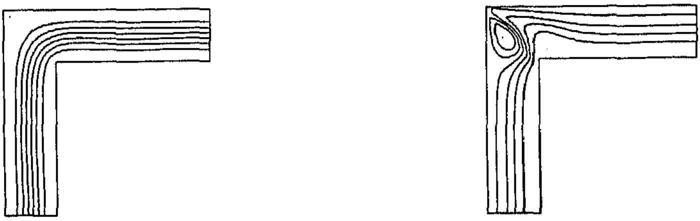

Fig. 5.2. Streamline patterns obtained with variant 1; Re = 10. (a): V^α unknowns; (b): U^α unknowns.

The physical domain is mapped onto a computational domain consisting of a number of rectangular, uniformly gridded blocks. Only the single block case is treated. In order to formulate the physical conservation laws in general coordinates, tensor notation is used. An invariant finite volume discretization is derived for the governing equations describing the physical conservation laws. Particular attention is given to the way in which the geometric quantities are deduced from the discrete geometric information in the cell vertices. The contravariant flux-components rather than velocity components are used as primary unknowns. Christoffel symbols appear in the conservation equations. They involve second derivatives of the mapping, which might lead to inaccuracies. Two variants have been presented. In variant 1 the Christoffel symbols are approximated in a straightforward manner. In variant 2 the occurrence of Christoffel symbols is avoided by contracting the governing equations. Preliminary results obtained with variant 1 show that a robust discretization is obtained on coarse and non-uniform grids. For stability reasons a staggered grid arrangement is used for the velocity components and the fluid pressure. Research is under way to investigate the applicability of collocated methods as well.

Time discretization is presented and an outline is given of the solution strategy based on the pressure correction method. The resulting systems are solved with a conjugate gradient type iterative method.

It is shown that time-varying computational grid can be handled in very much the same way, thus enabling extension of the solution method to adaptive computational grids as encountered in hydrodynamic applications involving a moving free fluid surface.

Acknowledgement

This research is partially funded by the Ministries of Education and Sciences and of Transportation and Public Works of The Netherlands (Directorate General of Science Policy).

References

1. Aris, R., *Vectors, tensors and the basic equations of fluid mechanics.* Prentice-Hall, Inc., Englewood Cliffs, N.J. (1962).
2. Brandsma, F.J. and Kuerten, J.G.M., The ISNaS compressible Navier–Stokes solver; first results for single airfoils. To appear in the *Proceedings of the 12th International Conference on Numerical Methods in Fluid Dynamics,* Oxford (1990).
3. Perić, M., Kessler, R. and Scheuerer, G., Comparison of finite-volume numerical methods with staggered and colocated grids. *Comp. & Fluids* 16 (1988) 389–403.
4. Rosenfeld, M. and Kwak, D. and Vinokur, M., A solution method for the unsteady and incompressible Navier–Stokes equations in generalized coordinate systems. AIAA Paper AIAA-88-0718 (1988).

5. Schuurman, J.J. and Loeve, W., Method Base System and Executive; tools for management and execution of software systems for CAE. In: *CAPE '89 – Third International Conference on Computer Applications in Production and Engineering,* Tokyo, Japan. Elsevier Science Publishers (1989) pp. 759–774.
6. Sedov, L.I., *A course in continuum mechanics*, Vol. I. *Basic equations and analytical techniques*. Wolters-Noordhoff Publishing, Groningen, The Netherlands (1964).
7. Sokolnikoff, I.S., *Tensor analysis*. John Wiley & Sons, Englewood Cliffs, N.J. (1964).
8. Steger, J.L. and Benek, J.A., On the use of composite grid schemes in computational aerodynamics. *Comp. Methods in Appl. Mech. and Eng.* 64 (1987) 301–320.
9. Thompson, J.F. and Steger, J.L., Three dimensional grid generation for complex configurations – recent progress. AGARDograph No. 309, AGARD, Neuilly-sur-Seine, France (1988).
10. Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W., *Numerical grid generation, foundations and applications*. North-Holland, Amsterdam (1985).
11. Vinokur, M., An analysis of finite-difference and finite-volume formulations of conservation laws. *J. Comp. Phys.* 81 (1989) 1–52.
12. Vogels, M.E.S. and Loeve, W., Development of ISNaS; an information system for flow simulation in design. In: *CAPE '89 – Third International Conference on Computer Applications in Production and Engineering,* Tokyo, Japan. Elsevier Science Publishers B.V. (1989) pp. 545–556.
13. Harlow, F.H. and Welch, J.E., Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids* 8 (1965) 2182–2189.