

Stellingen

behorende bij het proefschrift

Robust multigrid methods for the steady and
unsteady incompressible Navier-Stokes equations in
general coordinates.

van

Kees Oosterlee

Delft, zomer 1993.

- Bij discretisatie van een tensorvergelijking worden allerlei extra eisen verkregen, waaraan een nauwkeurige discretisatie moet voldoen. Zo leidt de representatie in contravariante componenten van een vector \mathbf{u} met $V^\alpha = \sqrt{g}U^\alpha$ tot een nauwkeuriger discretisatie dan met U^α .
(Zie hoofdstuk 3 van dit proefschrift.)

- Het toepassen van de niet-lineaire multirooster oplosmethode op de incompressibele Navier-Stokes vergelijkingen in algemene coördinaten gediscrètiseerd op een uniform rooster leidt tot gelijkwaardige convergentie resultaten als in Cartesische coördinaten, mits de diverse onderdelen van de methode correct worden gekozen.
(Zie hoofdstuk 4 van dit proefschrift.)

- Onthullende testproblemen voor discretisaties in algemene coördinaten zijn vaak zeer eenvoudige stromingen gediscrètiseerd op een niet-uniform rooster.

- Het convergentie-onderzoek van numerieke methoden voor stromingen met hogere Reynolds getallen (> 1000) is niet interessant als niet eerst de discretisatie is aangepast, zodat numerieke diffusie gereduceerd wordt, door bijvoorbeeld de implementatie van een defect-correctie techniek.

- Op een vector computer kan slechts een eerste inzicht in de geschiktheid van oplosmethoden voor massief parallelle computers verkregen worden.

- Het is vaak zinvoller om een multirooster methode met een snelle (Krylov-subspace) oplosmethode te vergelijken dan met zijn single grid variant.

(Zie Sonneveld, Wesseling en de Zeeuw in *Multigrid methods for integral and differential equations*, editors: Holstein & Paddon, Clarendon, Oxford, 1985.)

- De Multiple Semi-coarsened Grids (MSG) multirooster methode met een gedempte punt-Jacobi smoother is een robuuste methode die nog $\mathcal{O}(\mathcal{N})$ is wanneer een F-cycle gebruikt wordt.

- Er zijn tegenwoordig zo veel vaktijdschriften dat het erg moeilijk is om alle literatuur op een bepaald vakgebied bij te houden.

- Met de oprichting van onderzoekscholen, zoals het J.M. Burgerscentrum, lijkt de basis voor geavanceerd tweede fase onderwijs voor AIO's/OIO's gelegd. Of dit onderwijs ook daadwerkelijk van de grond komt staat of valt met het enthousiasme van de hoogleraren en de AIO's.

- Het klimaat in het Elektrotechniek gebouw leent zich beter voor het kweken van tomaten dan voor het verrichten van werkzaamheden op het gebied van onderwijs en onderzoek.

- De overeenkomst tussen popmuziek en onderzoek in de stromingsleer is dat veel interessante fenomenen zich ver van de hoofdstroming (mainstream) afspelen.

- Proefschriften staan vol met leuke namen voor popgroepen.
 Hier bijvoorbeeld: "The General Coordinates",
 "Coarse Grid Correction (CGC)" of
 "The Low Frequency Error Components".

58 pages
3778921
TR diss 2259

Robust multigrid methods for the steady and
unsteady incompressible Navier-Stokes equations in
general coordinates

Robust multigrid methods for the steady and
unsteady incompressible Navier-Stokes equations in
general coordinates

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus,
Prof. ir. K.F. Wakker,
in het openbaar te verdedigen
ten overstaan van een commissie
aangewezen door het College van Dekanen
op 14 september 1993 te 16.00 uur

door

Cornelis Willebrordus Oosterlee,

geboren te Zoeterwoude,
wiskundig ingenieur.



Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. ir. P. Wesseling

Promotie commissie:

Prof. dr. W. Hackbusch
(Christian-Albrechts-Universität Kiel)

Prof. dr. P.W. Hemker
(CWI – Universiteit van Amsterdam)

Prof. dr. ir. P.J. Zandbergen
(Universiteit Twente)

Prof. dr. ir. F.T.M. Nieuwstadt
(TU Delft, Fac. WbMT)

Prof. dr. ir. J.W. Boerstool
(NLR – TU Delft, Fac. TWI)

Prof. dr. ir. A.W. Heemink
(TU Delft, Fac. TWI)

Voorwoord.

Voor u ligt mijn proefschrift met onderzoek waar ik vier jaar aan gewerkt heb. Ik heb dit werk echter niet alleen gedaan, vele mensen hebben geholpen en daarvoor wil ik hen hier bedanken.

In het bijzonder wilde ik mijn promotor Prof. dr. ir. Piet Wesseling hartelijk bedanken voor de aangename samenwerking en alle tijd die hij aan mijn onderzoek besteed heeft. Zijn enthousiasme en kennis werkten bijzonder inspirerend. Vooral de uurtjes "brain stormen" herinner ik mij als bijzonder nuttig. Ook ben ik hem erkentelijk voor de introductie bij wetenschappers in binnen- en buitenland.

Het promotiewerk heeft gedeeltelijk parallel gelopen aan het onderzoek in de zogenaamde ISNaS groep. Ik heb ervaren dat het enorm prettig is om met een groep te werken en te discussieren over het onderzoek.

Voor de reeds aanwezige pre- en postprocessing software faciliteiten ben ik Guus Segal zeer erkentelijk; dankzij zijn programma pakket *Sepran* had ik vrijwel geen omkijken naar de voor- en nabewerkings mogelijkheden voor de stromingsberekeningen.

Kees Kassels en William Kuppen dank ik hartelijk voor hun hulp bij het onderzoek en voor de gezellige tijd aan de TU.

Kees Vuik ben ik erkentelijk voor de samenwerking en de tips, vooral in het laatste jaar.

De overige leden van de ISNaS groep, te weten Jos, Peter, Marcel, Erik, Peter, Kees en Zeng Shi bedank ik voor de samenwerking.

Tatiana aite (voor hulp bij het tekst verwerken, en alle vriendelijkheid). Jac dank ik voor het tolereren van mijn veelvuldige aanwezigheid op zijn kamer, en voor de back-up, waarmee ik een deel van dit proefschrift zonder veel moeite terug kreeg.

Alle kamergenoten van de afgelopen jaren en alle AIO/OIO's van de 7de verdieping die voor een gezellige sfeer hebben gezorgd, met name Eric de Sturler en Hans Braker, ben ik daarvoor dank verschuldigd.

Van alle vrienden en kennissen ben ik vooral Bart erkentelijk voor het zorgen voor een dak boven mijn hoofd.

Mijn ouders en broer hebben mij altijd gestimuleerd en hebben veel belangstelling getoond. Juist zij zijn een zeer belangrijke factor geweest in het welslagen van mijn studie.

Tenslotte doet het mij enorm veel genoegen mijn vriendin Anasja te bedanken voor alle liefde en gezelligheid.

Contents

1	General introduction.	3
2	Finite volume discretization of physical conservation laws in general coordinates.	7
2.1	Introduction.	7
2.2	Some fundamentals of tensor analysis.	8
2.3	Finite volume discretization of the divergence theorem.	16
2.4	Finite volume discretization of a tensor equation.	23
2.5	Conclusions.	27
3	Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates.	29
3.1	Introduction.	29
3.2	The incompressible Navier-Stokes equations in general coordinates. .	32
3.3	Finite volume discretization.	33
3.4	Benchmark solutions.	43
3.5	Conclusions.	55
4	Multigrid schemes for steady incompressible Navier-Stokes equations in general coordinates.	61
4.1	Introduction.	61
4.2	Multigrid methods for steady incompressible Navier-Stokes equations.	62
4.2.1	Results with SCGS.	72
4.2.2	Results with the SCAL smoother.	81
4.3	Steady incompressible flow around objects.	102
4.4	Conclusions.	108
5	Multigrid schemes for the time-dependent incompressible Navier-Stokes equations in general coordinates.	113
5.1	Introduction.	113
5.2	The temporal discretization of the incompressible Navier-Stokes equations.	114
5.3	The parabolic / time-parallel multigrid method.	117
5.4	The multigrid waveform relaxation method.	120
5.5	Results.	122
5.6	Conclusions.	128
6	Concluding remarks, suggestions for future research.	131
A	An example of an unstable discretization.	133
B	A discretization based on method 4 of Chapter 2.	135

References	141
Samenvatting	151
Curriculum vitae	153

1 General introduction.

Flows and flow problems are everywhere around us. We meet air flows in the environment, in and around buildings, or around aircraft and airfoils; liquid flows, like water flows in rivers, tidal flows in seas and groundwater flows, or blood flow through veins in the body, etcetera. Flow processes also occur on a large scale in industry, like flows in engines and turbines, oil flows in pipes, or flows of substances like butter or plastics through pipes, and many other situations. People have been wanting to get a grip on flow phenomena for years, and therefore they try to simulate flow problems with experiments. However, physical experiments can be very costly; (expensive) scale models have to be made to simulate complex flow problems. It is interesting to compare experimental results with other results, so that the results obtained with a scale model can be better extrapolated to full scale. Another type of flow prediction can be found when the partial differential equations representing a flow problem are solved. For many problems the underlying partial differential equations are the so-called Navier-Stokes equations, derived in the nineteenth century. The nonlinear system of Navier-Stokes equations, that consists of conservation laws for basic flow quantities like mass, momentum and energy, is a general description of flows. These equations simplify for incompressible flows for which the specific mass of a fluid particle may be considered constant. The energy equations can then be decoupled from the mass and momentum equations. Furthermore, the mass conservation equation reduces for incompressible flows to a constraint for the momentum equations. Thus the incompressible Navier-Stokes equations are obtained, which will be investigated here.

These equations are too complex to solve analytically; they need to be solved with numerical techniques. With the introduction of computers of steadily increasing speed and memory size, it becomes possible to solve more and more complex problems numerically. The equations are discretized and the flow domain is split up in many small parts, in which the numerical solution of the discretized equations is sought. The relatively new research area Computational Fluid Dynamics (CFD) is still growing rapidly. An introductory book on discretization methods and numerical techniques for heat transfer and fluid flow equations is [72]. A general introduction in computational fluid dynamics for compressible and incompressible flows can be found in [40]. The numerical prediction of complex flows requires both accurate physical modelling and numerical accuracy. We will deal with the modelling of steady and unsteady incompressible flows in domains of general shape by means of a so-called finite volume discretization. For this purpose a coordinate invariant tensor formulation of the incompressible Navier-Stokes equations is used. Some tensor tools needed are presented in Chapter 2. It is not at all trivial to obtain accurate results for the equations derived in general coordinates. Some requirements, for example with respect to the choice of unknowns, should be satisfied. These are derived in Chapter 3.

Single-phase laminar Newtonian flow equations are investigated, since they form the

basis of all other fluid flow simulations. Real life industrial flows are much more complex. Often, multiphase or chemical reacting flows, non-Newtonian or turbulent flows in complex domains are encountered. The modelling of these physical phenomena often requires extra equations and approximations of the physical model on top of the basic equations. It is necessary to understand the behaviour, the possibilities and limitations of the underlying incompressible Navier-Stokes equations. These are introduced, discretized and investigated in Chapter 3.

In the second part of the thesis, fast and robust solution techniques to solve large systems of discretized equations are investigated. Specific for multigrid solution methods is the rapid convergence, with convergence rates bounded by a constant (much smaller than one), that does not depend on the mesh size. Problems with m unknowns can be solved in $O(m)$ operations. Multigrid methods were essentially discovered in the sixties, and started to draw attention as fast and efficient solvers for large systems of equations in the late seventies ([7], [33]). A wide range of areas where numerical models are used has been investigated since. Computational fluid dynamics is one of these areas. Recent results obtained in this field are collected in [118]. We chose to use the standard nonlinear multigrid method ([35], [8]) to solve the steady incompressible Navier-Stokes equations. This algorithm is presented for the stationary equations in general coordinates in Chapter 4, where several smoothing methods are investigated for robustness. In Chapter 5 three robust multigrid methods to solve the unsteady incompressible Navier-Stokes equations for arbitrarily shaped domains are compared. These algorithms for unsteady equations possess more or less the same sequential complexity, while they differ in efficiency on parallel computers. Here, the efficiency on one processor of a vector machine is compared to see whether the algorithms are also efficient on these machines.

The different steps of the numerical solution of the incompressible Navier-Stokes equations in general coordinates presented in this thesis are depicted in Figure 1.1.

Two important aspects of the work in this thesis are the following. First, investigation of the accuracy of the discretization of the incompressible Navier-Stokes equations in general domains with a finite volume method. Second, the search for robust fast multigrid solution methods to solve the resulting systems of discretized equations. Moreover, the combination of these two aspects is challenging.

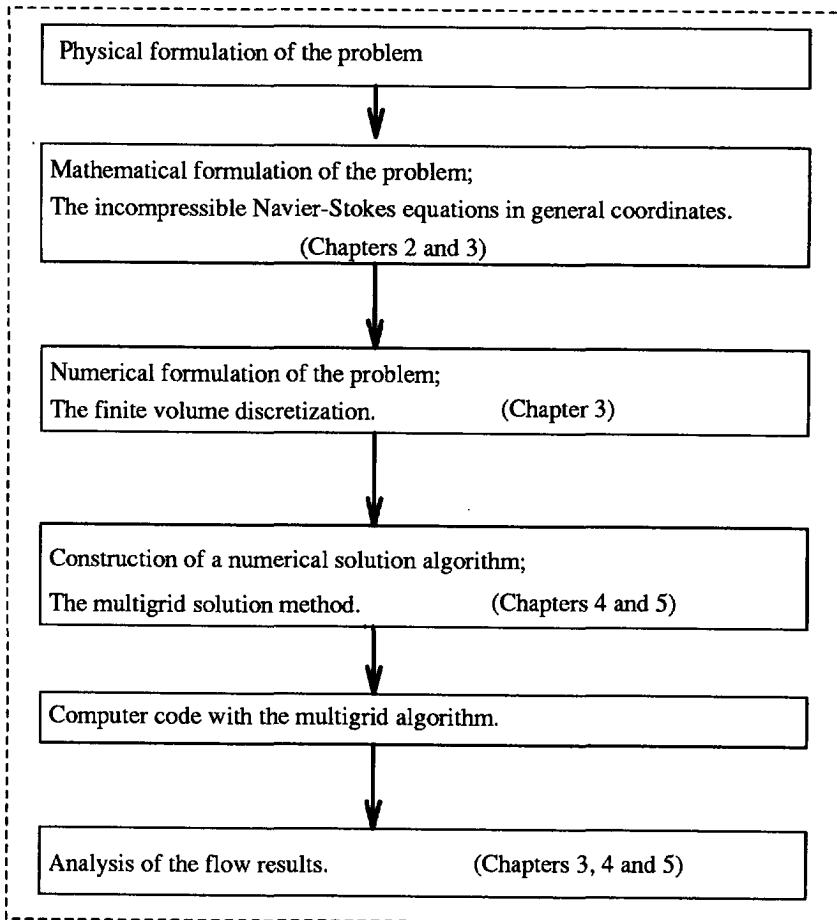


Figure 1.1: The standard approach for solving a CFD problem; the different analysis steps of the work presented in this thesis.

2 Finite volume discretization of physical conservation laws in general coordinates.

2.1 Introduction.

In order to calculate complex flows in arbitrary domains finite volume methods are quite popular nowadays. A finite volume discretization of the incompressible Navier-Stokes equations on a structured mesh results in structured operators with several blocks. Solution strategies exist to solve these types of problems efficiently. With finite volume methods, due to the rapidly growing computer speed and memory, the solution of three-dimensional complex flow problems in arbitrarily shaped domains comes within reach. However, when using a finite volume or a finite difference method it is not at all trivial to obtain accurate solutions of the incompressible Navier-Stokes equations in arbitrary domains. Several choices, each with their advantages and disadvantages, need to be made to obtain accurate discretizations. These choices are discussed throughout this thesis. The method adopted here is presented in full, and references to discretization methods adopting other choices are given.

Sections of this chapter are also presented in [113].

Boundary-fitted grids. A first choice to be made is that of the grid. Choosing a rectangular grid in the arbitrarily shaped domain means that a curved boundary must be approximated step-by-step, resulting in difficulties in the specification of boundary conditions. Therefore it is difficult to obtain accurate solutions in near-boundary regions, where often steep gradients occur. In [57] a rectangular coordinate system is used with a special treatment near the boundaries. An advantage is that the domain is discretized with Cartesian coordinates, which means that fairly simple well-known equations can be employed that do not contain terms due to curvature and non-orthogonality of coordinates. Another type of grid to be used in an arbitrary domain is a boundary-fitted grid, resulting from a change to curvilinear boundary-fitted coordinates. Then the boundary is a coordinate line, and the accurate specification of boundary conditions is much easier. Boundary-fitted grids can be generated by differential methods, for example by solving a Laplace or Poisson equation ([102]), or with algebraic methods ([27]). Boundary-fitted coordinates can be subdivided into two classes: orthogonal and non-orthogonal. When using orthogonal coordinates the transformed equations are still fairly simple. Successful discretization strategies using Cartesian coordinates can easily be extended to orthogonal curvilinear coordinates. The generation of orthogonal grids, however, can be very difficult in certain domains in two dimensions and sometimes impossible in three dimensions. Furthermore, orthogonal grids are not very flexible. There is not always sufficient control over the distribution of grid points. Non-orthogonal grids are easier to generate, especially in three-dimensional domains. In regions where extra fine resolution is required it is easy to concentrate many grid points. A disad-

vantage of non-orthogonal grids is that additional terms appear in the transformed equations, which require additional discretization approximations, that influence stability and accuracy of the solution, and possibly reduce the convergence rate of solution algorithms. An interesting development is the generation of non-orthogonal grids, which are orthogonal at the boundary ([94]). This facilitates the treatment of boundary conditions in general coordinates. In our research non-orthogonal grids are adopted; it is interesting to investigate advantages and limitations of this approach.

2.2 Some fundamentals of tensor analysis.

In general coordinates it seems attractive to use coordinate invariant formulations and discretizations of physical laws, provided sufficient discretization accuracy can be obtained. The coordinate invariant formulations of the incompressible Navier-Stokes equations are written in tensor notation. Tensor notation proves indispensable for formulating physical conservation laws in general coordinates.

Scalars, vectors, tensors. Physical quantities that are specified by one real function of position are called scalars. Examples are pressure p and temperature T . Of course, actual values depend on the physical unit system chosen. Quantities specified by a magnitude and a direction in space are called vectors. The magnitude depends on the choice of units. A vector is specified by parameters called its components; in d dimensions there are d components. The components depend on the coordinate system used. A vector as physical entity is invariant with respect to changes of coordinates (see Figure 2.1). Coordinate-invariant entities such as scalars and vectors are called tensors. Their invariant nature leads to certain transformation laws for their values or components under changes of coordinates, which will be formulated below.

Invariance and tensors. The fundamentals of tensor analysis, especially in relation to continuum mechanics, are presented in textbooks such as [2] and [99]. For ease of reference a brief summary of the main tools needed in our research is presented here.

Let $\mathbf{x} = (x^1, x^2, \dots, x^d)$ and $\boldsymbol{\xi} = (\xi^1, \xi^2, \dots, \xi^d)$ be a Cartesian and a general coordinate system, respectively, with $d = 2$ or $d = 3$. Let $\Omega \subset R^d$, $G \subset R^d$. Consider a transformation $\Omega \rightarrow G$:

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}), \quad \mathbf{x} \in \Omega, \quad \boldsymbol{\xi} \in G \quad (2.1)$$

Hence the differentials of the coordinates dx^α and $d\xi^\beta$ are related by:

$$d\xi^\beta = \frac{\partial \xi^\beta}{\partial x^\alpha} dx^\alpha \quad (2.2)$$

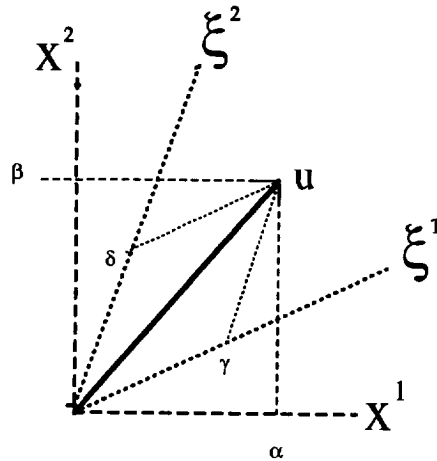


Figure 2.1: A vector represented in two different coordinate systems:
 $\mathbf{u} = \alpha \mathbf{e}_{(1)} + \beta \mathbf{e}_{(2)} = \gamma \mathbf{a}_{(1)} + \delta \mathbf{a}_{(2)}$, with $\mathbf{e}_{(\alpha)}$ and $\mathbf{a}_{(\alpha)}$ unit vectors along the coordinate axes.

Greek indices will have range $1, 2, \dots, d$. The summation convention will hold for Greek indices only. This convention implies that summation takes place over pairs of equal Greek indices in a single term or product, one of which must be a superscript and the other a subscript. Other repetitions of indices are not allowed in a term or product.

A transformation is called admissible if (2.1) is onto and

$$J = \left| \frac{\partial \mathbf{x}^\alpha}{\partial \xi^\beta} \right| = \begin{vmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^1} & \cdots & \frac{\partial x^d}{\partial \xi^1} \\ \vdots & \vdots & & \vdots \\ \frac{\partial x^1}{\partial \xi^d} & \frac{\partial x^2}{\partial \xi^d} & \cdots & \frac{\partial x^d}{\partial \xi^d} \end{vmatrix} \neq 0 \quad (2.3)$$

where J is the Jacobian of the transformation. This implies that J^{-1} exists and

$$J^{-1} \neq 0, \quad \forall \mathbf{x} \in \Omega \quad (2.4)$$

and that the mapping (2.1) is invertible, i.e. the mapping is one-to-one. The admissible transformations form a group. That is, if T_1, T_2 and T_3 are any admissible transformations, they satisfy

- (i) The identity mapping $\mathbf{x} = \boldsymbol{\xi}$ is admissible.
- (ii) $T_1 T_2$ is admissible.

(iii) T_1^{-1} is admissible.

(iv) $T_1(T_2T_3) = (T_1T_2)T_3$

Let there be given a set of m real continuous functions $f_i(\mathbf{x})$, $i = 1, 2, \dots, m$, $\mathbf{x} \in \Omega$. We associate with every admissible transformation T a transformation S :

$$S: f_i(\mathbf{x}) \rightarrow g_i(\boldsymbol{\xi}), \quad i = 1, 2, \dots, m, \quad \boldsymbol{\xi} \in G \quad (2.5)$$

with g_i real and continuous. Obviously, S will depend on T and will be said to be induced by T . Suppose that S as function of T satisfies the following conditions for every admissible T_1, T_2, T_3 and S_1, S_2, S_3 induced by T_1, T_2, T_3 respectively:

If T_1 is the identity transformation, then S_1 is the identity transformation. (2.6)

$$\text{If } T_3 = T_2T_1 \text{ then } S_3 = S_2S_1 \quad (2.7)$$

That is, the sets of admissible transformations and the set of transformations of type S are isomorphic. We shall then say that the set f_i represents the components of a tensor f in the \mathbf{x} -coordinate system, the tensor itself being the totality of sets of functions $f_i(\mathbf{x}), g_i(\boldsymbol{\xi})$, etc. In other words, tensors are induced transformations that are isomorphic to admissible transformations of coordinates. This is a mathematical expression of the notion that tensors are objects that are independent of the coordinate system in which they are described.

An example of such induced transformations is transformation by invariance:

$$f_i(\mathbf{x}) \rightarrow g_i(\boldsymbol{\xi}) = f_i\{\mathbf{x}(\boldsymbol{\xi})\} \quad (2.8)$$

Other such induced transformations, namely transformation by covariance and by contravariance will be introduced soon.

First, some notation is introduced. Vectors will be denoted by bold-faced lower case Roman letters, their Cartesian components by lower case Roman letters and their components in general coordinates by upper case Roman letters. For example, $\mathbf{u}, u_\alpha, u^\alpha, U_\alpha, U^\alpha$ all refer to the same vector. Strictly speaking, a quantity such as U^α is a vector component, but no harm is done in referring loosely to it as the vector U^α . We have $u_\alpha = u^\alpha$. The difference between U_α and U^α will become clear below.

A quantity Q^β is called a contravariant vector if under transformation from coordinates \mathbf{x} to $\boldsymbol{\xi}$ q^α transforms to Q^β according to

$$Q^\beta = \frac{\partial \xi^\beta}{\partial x^\alpha} q^\alpha \quad (2.9)$$

Hence, equation (2.2) shows that a line element is a contravariant vector. The velocity vector component $v^\alpha = dx^\alpha/dt$ is another example of a contravariant vector, because

$$V^\beta = \frac{d\xi^\beta}{dt} = \frac{\partial \xi^\beta}{\partial x^\alpha} \frac{dx^\alpha}{dt} = \frac{\partial \xi^\beta}{\partial x^\alpha} v^\alpha \quad (2.10)$$

A quantity Q_β is called a covariant vector if under coordinate transformation it transforms according to

$$Q_\beta = \frac{\partial x^\alpha}{\partial \xi^\beta} q_\alpha \quad (2.11)$$

A gradient of a scalar function is an example of a covariant vector, because

$$\frac{\partial p}{\partial \xi^\beta} = \frac{\partial x^\alpha}{\partial \xi^\beta} \frac{\partial p}{\partial x^\alpha} \quad (2.12)$$

In our research we only need tensors of rank zero (scalars), rank one (vectors) and of rank two (tensors with two indices). For a survey of general transformation laws we refer to the literature. We only give the following illustrative example: the relative mixed tensor $U_\beta^\alpha(\xi)$ of rank two and weight w satisfies the following transformation law:

$$U_\beta^\alpha = (\sqrt{g})^w \frac{\partial \xi^\alpha}{\partial x^\gamma} \frac{\partial x^\delta}{\partial \xi^\beta} u_\delta^\gamma \quad (2.13)$$

where $\sqrt{g} = J$ (specializing to right-handed coordinate systems we have $J > 0$, as will be shown shortly).

The tensor is called absolute if $w = 0$. The superscript α is called a contravariant index, and transforms according to the contravariant transformation law. The subscript β is called a covariant index, and transforms according to the covariant transformation law. Another example is the transformation for a relative scalar of weight w :

$$\rho(\xi) = (\sqrt{g})^w \rho(\mathbf{x}) \quad (2.14)$$

It is not difficult to show that all of the above transformation laws are isomorphic to the set of admissible transformations and hence correspond to tensors; we will not do this here.

Base vectors. Contravariant base vectors $\mathbf{a}^{(\alpha)}$ are defined as normal vectors to the $\xi^\alpha = \text{constant}$ surfaces:

$$\mathbf{a}^{(\alpha)} = \text{grad } \xi^\alpha \quad (2.15)$$

The subscript α is placed between parentheses to emphasize that $\mathbf{a}^{(\alpha)}$ is not a component but one of the d contravariant base vectors.

Covariant base vectors $\mathbf{a}_{(\alpha)}$ are defined as tangent vectors to the coordinate lines ξ^α :

$$\mathbf{a}_{(\alpha)} = \frac{\partial \mathbf{x}}{\partial \xi^\alpha} \quad (2.16)$$

Only right-handed coordinate systems are used. This means that for $d = 3$

$$\mathbf{a}_{(\alpha)} \cdot (\mathbf{a}_{(\beta)} \wedge \mathbf{a}_{(\gamma)}) > 0, \quad \alpha, \beta, \gamma \text{ cyclic} \quad (2.17)$$

Comparison with (2.3) shows that this implies

$$J = \mathbf{a}_{(1)} \cdot (\mathbf{a}_{(2)} \wedge \mathbf{a}_{(3)}) > 0 \quad (2.18)$$

The two-dimensional version of (2.17) is obtained by defining $x^3 = \xi^3$, $x^\alpha = x^\alpha(\xi^1, \xi^2)$, $\alpha = 1, 2$. This gives

$$a_{(\alpha)}^3 = 0, \quad \alpha = 1, 2; \quad \mathbf{a}_{(3)} = (0, 0, 1) \quad (2.19)$$

so that (2.18) gives again

$$J = \begin{vmatrix} \frac{\partial x^1}{\partial \xi^1} & \frac{\partial x^2}{\partial \xi^1} \\ \frac{\partial x^1}{\partial \xi^2} & \frac{\partial x^2}{\partial \xi^2} \end{vmatrix} \neq 0 \quad (2.20)$$

It can be shown that

$$\mathbf{a}^{(\alpha)} = \frac{1}{\sqrt{g}}(\mathbf{a}_{(\beta)} \wedge \mathbf{a}_{(\gamma)}) \quad \text{for } \alpha, \beta, \gamma \text{ cyclic} \quad (2.21)$$

(for $d = 3$) and

$$\mathbf{a}_{(\alpha)} \cdot \mathbf{a}^{(\beta)} = \delta_\alpha^\beta \quad (2.22)$$

with δ the Kronecker delta.

The two-dimensional version of (2.21) is obtained by using (2.19). This gives

$$\mathbf{a}^{(1)} = \frac{1}{\sqrt{g}}(a_{(2)}^2, -a_{(2)}^1), \quad \mathbf{a}^{(2)} = \frac{1}{\sqrt{g}}(-a_{(1)}^2, a_{(1)}^1) \quad (2.23)$$

The Cartesian components of the base vectors are given by

$$a_{\beta}^{(\alpha)} = \frac{\delta \xi^\alpha}{\delta x^\beta}, \quad a_{(\alpha)}^\beta = \frac{\delta x^\beta}{\delta \xi^\alpha} \quad (2.24)$$

Figure 2.2 shows a sketch of the (x^1, x^2) -plane with two coordinate lines of a general coordinate system (ξ^1, ξ^2) . Furthermore, co- and contravariant base vectors are shown.

Relation between vector and tensor notation. According to (2.9) and (2.11) we may write

$$U^\alpha = \mathbf{a}^{(\alpha)} \cdot \mathbf{u}, \quad U_\alpha = \mathbf{a}_{(\alpha)} \cdot \mathbf{u} \quad (2.25)$$

Using (2.22) it follows that

$$\mathbf{u} = U^\alpha \mathbf{a}_{(\alpha)} = U_\alpha \mathbf{a}^{(\alpha)} \quad (2.26)$$

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

$$g_{\alpha\beta} = \mathbf{a}_{(\alpha)} \cdot \mathbf{a}_{(\beta)} \quad ; \quad g^{\alpha\beta} = \mathbf{a}^{(\alpha)} \cdot \mathbf{a}^{(\beta)} \quad (2.27)$$

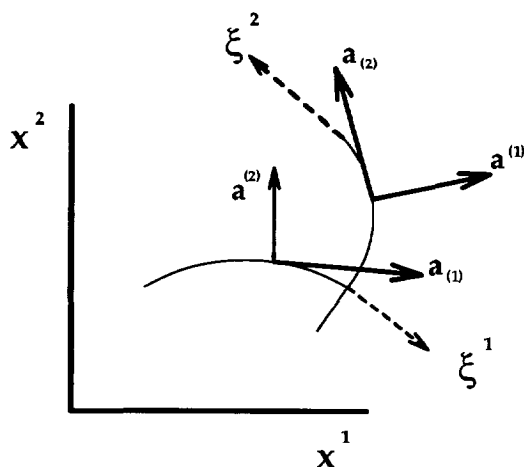


Figure 2.2: The Cartesian domain with general ξ^α -coordinate lines and co- and contravariant base vectors.

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

$$ds^2 = \delta_{\alpha\beta} dx^\alpha dx^\beta = g_{\alpha\beta} d\xi^\alpha d\xi^\beta = g^{\alpha\beta} d\xi_\alpha d\xi_\beta \quad (2.28)$$

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

$$g = J^2 \quad (2.29)$$

For the representation of a volume we find

$$dx^1 dx^2 \dots dx^d = \sqrt{g} d\xi^1 d\xi^2 \dots d\xi^d \quad (2.30)$$

The metric tensor can be used for raising and lowering indices, as illustrated in the following example

$$U_\beta^\alpha = g^{\alpha\gamma} U_{\gamma\beta} \quad (2.31)$$

Note that the notation U_β^α is ambiguous if $U_{\alpha\beta} \neq U_{\beta\alpha}$ in which case (not occurring here) one writes U_{β}^{α} .

The covariant derivative. A partial derivative $\partial/\partial\xi^\alpha$ does not necessarily transform according to the rules (2.6) and (2.7) and hence does not need to result in a tensor. For example, the quantity $\partial p/\partial\xi^\beta$ is a covariant tensor of rank one (2.12), but the second derivative is not a second order tensor:

$$\begin{aligned} \frac{\partial}{\partial\xi^\gamma} \left(\frac{\partial p}{\partial\xi^\beta} \right) &= \frac{\partial}{\partial\xi^\gamma} \left(\frac{\partial x^\alpha}{\partial\xi^\beta} \frac{\partial p}{\partial x^\alpha} \right) = \\ &= \frac{\partial^2 p}{\partial x^\alpha \partial x^\delta} \frac{\partial x^\delta}{\partial\xi^\gamma} \frac{\partial x^\alpha}{\partial\xi^\beta} + \frac{\partial p}{\partial x^\alpha} \frac{\partial^2 x^\alpha}{\partial\xi^\beta \partial\xi^\gamma} \end{aligned} \quad (2.32)$$

The operation that if applied to a tensor results in a tensor and that corresponds to partial differentiation when the mapping is the identity $\mathbf{x} = \boldsymbol{\xi}$ is called covariant differentiation.

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

$$\rho_{,\alpha} \equiv \frac{\partial \rho}{\partial \xi^\alpha} \quad (2.33)$$

The covariant derivative of a scalar density (for example $\sqrt{g}\rho$, i.e. a relative scalar of weight one) is defined as

$$(\sqrt{g}\rho)_{,\alpha} = \frac{\partial \sqrt{g}\rho}{\partial \xi^\alpha} - \{ \begin{smallmatrix} \beta \\ \beta\alpha \end{smallmatrix} \} \sqrt{g}\rho \quad (2.34)$$

Here $\{ \begin{smallmatrix} \alpha \\ \beta\gamma \end{smallmatrix} \}$ is the so-called Christoffel symbol of the second kind, defined by

$$\{ \begin{smallmatrix} \alpha \\ \beta\gamma \end{smallmatrix} \} = \{ \begin{smallmatrix} \alpha \\ \gamma\beta \end{smallmatrix} \} = \mathbf{a}^{(\alpha)} \cdot \frac{\partial \mathbf{a}_{(\gamma)}}{\partial \xi^\beta} = \frac{\partial \xi^\alpha}{\partial x^\delta} \frac{\partial^2 x^\delta}{\partial \xi^\gamma \partial \xi^\beta} \quad (2.35)$$

It can be shown that

$$(\sqrt{g}\rho)_{,\alpha} = \sqrt{g} \frac{\partial \rho}{\partial \xi^\alpha} \quad (2.36)$$

Hence:

$$\sqrt{g}_{,\alpha} = 0 \quad (2.37)$$

The covariant derivative of an absolute contravariant tensor of rank one is defined by

$$U_{,\beta}^\alpha \equiv \frac{\partial U^\alpha}{\partial \xi^\beta} + \{ \begin{smallmatrix} \alpha \\ \gamma\beta \end{smallmatrix} \} U^\gamma \quad (2.38)$$

The covariant derivative for a relative contravariant tensor of weight one ((2.13) with $w = 1$), for example $\sqrt{g}U^\alpha$, is defined by

$$(\sqrt{g}U^\alpha)_{,\beta} = \frac{\partial \sqrt{g}U^\alpha}{\partial \xi^\beta} + \{ \begin{smallmatrix} \alpha \\ \gamma\beta \end{smallmatrix} \} \sqrt{g}U^\gamma - \{ \begin{smallmatrix} \beta \\ \beta\gamma \end{smallmatrix} \} \sqrt{g}U^\alpha \quad (2.39)$$

and can be found from (2.38), (2.37) and (2.34) with $\rho \equiv 1$ by substitution of $\sqrt{g}U^\alpha$. The covariant derivative of an absolute covariant tensor of rank one is defined by

$$U_{\alpha,\beta} \equiv \frac{\partial U_\alpha}{\partial \xi^\beta} - \{ \begin{smallmatrix} \gamma \\ \alpha\beta \end{smallmatrix} \} U_\gamma \quad (2.40)$$

It can be shown that

$$\operatorname{div} \mathbf{u} = U_{,\alpha}^{\alpha} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} U^{\alpha}}{\partial \xi^{\alpha}} \quad (2.41)$$

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

$$T_{,\gamma}^{\alpha\beta} \equiv \frac{\partial T^{\alpha\beta}}{\partial \xi^{\gamma}} + \left\{ \begin{matrix} \alpha \\ \gamma\delta \end{matrix} \right\} T^{\delta\beta} + \left\{ \begin{matrix} \beta \\ \gamma\delta \end{matrix} \right\} T^{\alpha\delta} \quad (2.42)$$

It can be shown that

$$T_{,\beta}^{\alpha\beta} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g} T^{\alpha\beta}}{\partial \xi^{\beta}} + \left\{ \begin{matrix} \alpha \\ \gamma\beta \end{matrix} \right\} T^{\gamma\beta} \quad (2.43)$$

Ricci's lemma. A powerful feature of tensor analysis is, that if a tensor equation holds in one coordinate system, then it holds in every coordinate system obtained from the first by an admissible transformation. This is known as Ricci's lemma. In other words, tensor laws are invariant with respect to changes in coordinates. The following important identity follows immediately

$$g_{,\gamma}^{\alpha\beta} = 0 \quad (2.44)$$

since it obviously holds in Cartesian coordinates (take $\mathbf{x} = \boldsymbol{\xi}$). Ricci's lemma enables us to write down physical laws immediately in arbitrary coordinates, by writing their Cartesian formulation in tensor notation.

A geometric identity. Suppose \mathbf{v} is constant. Then with (2.41) is found:

$$0 = \operatorname{div} \mathbf{v} = \frac{1}{\sqrt{g}} \frac{\partial (\sqrt{g} \mathbf{a}^{(\alpha)} \cdot \mathbf{v})}{\partial \xi^{\alpha}} = \frac{1}{\sqrt{g}} \mathbf{v} \cdot \frac{\partial \sqrt{g} \mathbf{a}^{(\alpha)}}{\partial \xi^{\alpha}} \quad (2.45)$$

for all constant \mathbf{v} . Hence, we have the following geometric identity:

$$\frac{\partial \sqrt{g} \mathbf{a}^{(\alpha)}}{\partial \xi^{\alpha}} = 0 \quad (2.46)$$

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

$$\int_V \operatorname{div} \mathbf{u} dV = \oint_S \mathbf{u} \cdot d\mathbf{S} \quad (2.47)$$

Here \mathbf{dS} stands for the vector $\mathbf{n}dS$, with \mathbf{n} the outward unit normal vector on S , and dS the (physical) surface element. In tensor notation the divergence theorem is given by (using Ricci's lemma):

$$\int_V U_{,\alpha}^\alpha dV = \oint_S U^\alpha dS_\alpha \quad (2.48)$$

Integral version of geometric identity. An integral version of the geometric identity (2.46) is obtained by applying (2.48) to a constant vector field \mathbf{v}

$$0 = \int_V V_{,\alpha}^\alpha dV = \oint_S V^\alpha dS_\alpha = v^\beta \oint_S a_\beta^{(\alpha)} dS_\alpha \quad (2.49)$$

for all constant \mathbf{v} . This gives the following integral version of the geometric identity (2.46):

$$\oint_S a_\beta^{(\alpha)} dS_\alpha = 0 \quad (2.50)$$

2.3 Finite volume discretization of the divergence theorem.

Boundary-fitted coordinates and computational grid. From now on $d = 2$, the extension to $d = 3$ is more or less straightforward. The physical domain Ω is transformed to a rectangle G with the mapping (2.1). Ω is assumed to be topologically equivalent to G , so that admissible mappings from Ω to G exist. The boundary $\partial\Omega$ of Ω consists of four segments, each of which is the image of a side of G . This means that along each segment of $\partial\Omega$ we have $\xi^\alpha = \text{constant}$, for $\alpha = 1$ or $\alpha = 2$. That is why the ξ^α -coordinates are called boundary-fitted. On G a computational grid G_h is defined; see Figure 2.3. The grid G_h is defined as follows:

$$G_h = \{ \xi_{i-1/2, j-1/2} : \xi_{i-1/2}^1 = (i-1)\Delta\xi^1, \quad i = 1, 2, \dots, \bar{i} + 1, \quad (2.51) \\ \xi_{j-1/2}^2 = (j-1)\Delta\xi^2, \quad j = 1, 2, \dots, \bar{j} + 1, \quad \Delta\xi^1 = L_1/\bar{i}, \quad \Delta\xi^2 = L_2/\bar{j} \}$$

with L_α the size of G_h in the ξ^α -direction. Hence there are i cells in the ξ^1 -direction and j cells in the ξ^2 -direction. The cell with center at $\xi_{ij} = ((i-\frac{1}{2})\Delta\xi^1, (j-\frac{1}{2})\Delta\xi^2)$ is called C_{ij} . Figure 2.4 shows how points belonging to C_{ij} are indexed.

The mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ is specified more precisely as follows. A numerical grid generation procedure is assumed to generate $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ for $\boldsymbol{\xi}$ in G_h . The mapping is extended to all of G (and Ω) by bilinear interpolation. As a consequence, $\partial\Omega$ is approximated by linear segments. Furthermore, $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$ is continuous and infinitely differentiable, except on the boundaries of the cells C_{ij} , where the mapping is continuous but certain derivatives do not exist. For example, in $\xi_{i-\frac{1}{2}, j}$, $\mathbf{a}_{(2)}$ exists, but $\mathbf{a}_{(1)}$ does not. Inside a cell the mapping is given by

$$\mathbf{x} = \mathbf{c}_{(0)} + \mathbf{c}_{(1)}\xi^1 + \mathbf{c}_{(2)}\xi^2 + \mathbf{c}_{(3)}\xi^1\xi^2 \quad (2.52)$$

for certain constant vectors $\mathbf{c}_{(0)}, \dots, \mathbf{c}_{(3)}$. Each cell C_{ij} in G has an image V_{ij} in Ω . The boundary of a cell V_{ij} is called S_{ij} . The connections between the vertices of a cell are called the faces of a cell. Because the mapping is piecewise bilinear, the faces are straight in G , and ξ^1 or ξ^2 is constant along a face, the faces of V_{ij} are straight.

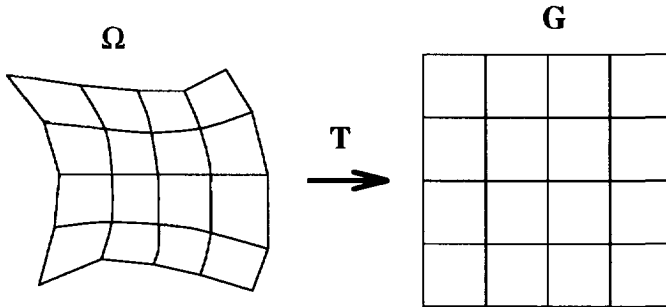


Figure 2.3: Boundary-fitted coordinates and computational grid.

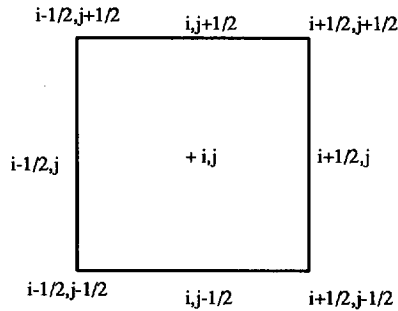


Figure 2.4: Numbering of points in a cell.

Discrete approximations of geometric quantities. Due to the piecewise bilinear mapping all geometric quantities can be calculated exactly inside each cell. However, some geometric quantities are discontinuous on cell-faces. As already mentioned, covariant base vectors $\mathbf{a}_{(\alpha)}$ are discontinuous on faces $\xi^\alpha = \text{constant}$ (these faces will be called α -faces), \sqrt{g} is discontinuous on all faces. The computation of $\mathbf{a}_{(\alpha)}$ can be done in a straightforward way exactly by means of finite differences in the centers of the ξ^α -faces, as follows,

$$a_{(1)}^\alpha{}_{i,j+1/2} = (\delta x^\alpha / \delta \xi^1)_{i,j+1/2}, \quad a_{(2)}^\alpha{}_{i+1/2,j} = (\delta x^\alpha / \delta \xi^2)_{i+1/2,j} \quad (2.53)$$

where $\delta x^\alpha_{i,j+1/2}$ and $\delta \xi^1_{i,j+1/2}$ are the increments of x^α and ξ^1 in the positive ξ^1 -direction along the face with center $(i, j+1/2)$, and similarly for $(i+1/2, j)$. Because

the mapping is piecewise bilinear, $\mathbf{a}_{(\alpha)}$ can be obtained exactly inside a cell by interpolation, for example

$$\mathbf{a}_{(1);i,j} = \frac{1}{2} (\mathbf{a}_{(1);i,j+1/2} + \mathbf{a}_{(1);i,j-1/2}) \quad (2.54)$$

$$\mathbf{a}_{(2);i,j} = \frac{1}{2} (\mathbf{a}_{(2);i+1/2,j} + \mathbf{a}_{(2);i-1/2,j}) \quad (2.55)$$

On faces where $\mathbf{a}_{(\alpha)}$ is not defined it is also approximated by interpolation, but the accuracy will depend on the smoothness of the mapping $\mathbf{x} = \mathbf{x}(\boldsymbol{\xi})$. For example,

$$\mathbf{a}_{(1);i+\frac{1}{2},j} \approx \bar{\mathbf{a}}_{(1);i+\frac{1}{2},j} = \frac{1}{4} \{ \mathbf{a}_{(1);i,j+1/2} + \mathbf{a}_{(1);i+1,j+1/2} + \mathbf{a}_{(1);i+1,j-1/2} + \mathbf{a}_{(1);i,j-1/2} \} \quad (2.56)$$

The Jacobian \sqrt{g} is computed inside a cell and on all faces with (2.20) from $\mathbf{a}_{(\alpha)}$. So in $(i+1/2, j)$ \sqrt{g} is computed as follows:

$$\sqrt{g}_{i+1/2,j} = a_{(2);i+1/2,j}^2 \bar{a}_{(1);i+1/2,j}^1 - a_{(2);i+1/2,j}^1 \bar{a}_{(1);i+1/2,j}^2 \quad (2.57)$$

Contravariant base vectors $\mathbf{a}^{(\alpha)}$ are also computed from the interpolated covariant base vectors with (2.23):

$$\begin{aligned} \mathbf{a}_{i+1/2,j}^{(1)} &\approx \frac{1}{\sqrt{g}} (\delta x^2 / \delta \xi^2, -\delta x^1 / \delta \xi^2)_{i+1/2,j}, \quad (= \frac{1}{\sqrt{g}} (a_{(2)}^2, -a_{(2)}^1)_{i+1/2,j}) \\ \mathbf{a}_{i,j+1/2}^{(2)} &\approx \frac{1}{\sqrt{g}} (-\delta x^2 / \delta \xi^1, \delta x^1 / \delta \xi^1)_{i,j+1/2} \end{aligned} \quad (2.58)$$

From (2.53), (2.54), (2.55), (2.57) and (2.58) it follows that (2.22) is satisfied. We will see shortly that the geometric identity (2.50) is also satisfied. The restriction to a smooth mapping can sometimes be an undesirable limitation. Therefore research is being done ([116]) to avoid the interpolations (2.56) and to derive approximations that are exact for piecewise bilinear mappings, for example by using appropriate one-sided limits at places where a quantity is discontinuous. Results of this approach are not yet known, and we will not discuss this approach here.

Discrete approximations of the divergence theorem. Let us apply the divergence theorem to the finite volumes V_{ij} . Let $\delta \mathbf{S}^{(\alpha)}$ be the vector normal to an α -face in positive ξ^α -direction. (The superscript in brackets does not refer to a contravariant index, but specifies an α -face. So $\delta S_\beta^{(\alpha)}$ is a covariant vector.) Its Cartesian components are given by

$$\delta \mathbf{S}^{(1)} = (\delta x^2, -\delta x^1), \quad \delta \mathbf{S}^{(2)} = (-\delta x^2, \delta x^1) \quad (2.59)$$

where δx^α is the increment of x^α along the cell face in the positive ξ^α -direction. Hence, the length of $\delta \mathbf{S}^{(\alpha)}$ is the length of the cell face. We have

$$\delta S^{(1)} = \sqrt{g} \mathbf{a}^{(1)} \delta \xi^2, \quad \delta S^{(2)} = \sqrt{g} \mathbf{a}^{(2)} \delta \xi^1 \quad (2.60)$$

The surface integral in (2.47) and (2.48) has to be approximated numerically. This numerical approximation can be done in several ways. Because we have in mind the discretization of the Navier-Stokes equations on a staggered grid arrangement of variables, it is assumed that an appropriate contravariant velocity component or the mass flux is available in the centers of the faces.

Method 1. We start from the vector formulation (2.47). Assume that the mass fluxes V^α , defined as

$$V^\alpha = \sqrt{g} U^\alpha \quad (2.61)$$

are available in the centers of the α -faces. V^α is a relative tensor of weight one. The following approximation is made for the integral along AB with midpoint M (see Figure 2.5):

$$\int_A^B \mathbf{u} \cdot d\mathbf{S} \approx \mathbf{u}_M \cdot \delta \mathbf{S}_M^{(2)} \quad (2.62)$$

Hence (using (2.60) and (2.61)):

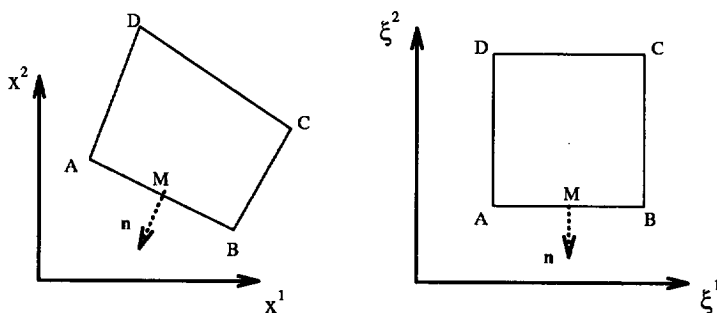


Figure 2.5: A cell in Cartesian and computational domain with indices.

$$\int_A^B \mathbf{u} \cdot d\mathbf{S} \approx -V_{i,j-1/2}^2 \delta \xi^1 \quad (2.63)$$

Treating the other faces in similar fashion we obtain

$$\oint_S \mathbf{u} \cdot d\mathbf{S} \approx (-V_{i,j-1/2}^2 + V_{i,j+1/2}^2) \delta \xi^1 + (V_{i+1/2,j}^1 - V_{i-1/2,j}^1) \delta \xi^2 \quad (2.64)$$

Now suppose that \mathbf{u} is constant. This implies that

$$\oint_S \mathbf{u} \cdot d\mathbf{S} = 0 \quad (2.65)$$

It is desirable that the numerical approximation (2.64) also results in (2.65). This is indeed the case if we compute $\mathbf{a}^{(\alpha)}$ by (2.57) and (2.58). From (2.23) and (2.25) we have

$$U^1 = \frac{1}{\sqrt{g}} \left(a_{(2)}^2 u^1 - a_{(2)}^1 u^2 \right), \quad U^2 = \frac{1}{\sqrt{g}} \left(-a_{(1)}^2 u^1 + a_{(1)}^1 u^2 \right) \quad (2.66)$$

Writing (2.64) as

$$\oint_S \mathbf{u} \cdot d\mathbf{S} \approx V^2 |_{i,j-1/2}^{i,j+1/2} \delta\xi^1 + V^1 |_{i-1/2,j}^{i+1/2,j} \delta\xi^2 \quad (2.67)$$

and substitution of (2.61) and (2.66) gives

$$\begin{aligned} \oint_S \mathbf{u} \cdot d\mathbf{S} &\approx u^1 \{ a_{(2)}^2 |_{i-1/2,j}^{i+1/2,j} \delta\xi^2 - a_{(1)}^2 |_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \} \\ &+ u^2 \{ -a_{(2)}^1 |_{i-1/2,j}^{i+1/2,j} \delta\xi^2 + a_{(1)}^1 |_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \} \end{aligned} \quad (2.68)$$

With $\mathbf{a}_{(\alpha)}$ computed according to (2.53) equation (2.68) gives (2.65) exactly ! This implies that our approximations of the base vectors satisfy the geometric identity (2.50).

Method 2. We start from the tensor formulation (2.48). We assume that we have the contravariant components U^α available in the centers of the α -faces. The covariant components $\delta S_\beta^{(\alpha)}$ are derived from (2.60) and (2.11) and one obtains:

$$\delta S_1^{(1)} \approx \sqrt{g} \delta\xi^2, \quad \delta S_2^{(1)} \approx 0, \quad \delta S_1^{(2)} \approx 0, \quad \delta S_2^{(2)} \approx \sqrt{g} \delta\xi^1 \quad (2.69)$$

The following approximation is made for the integral along AB with midpoint M (see Figure 2.5):

$$\int_A^B U^\alpha dS_\alpha \approx U^\alpha(M) \delta S_\alpha^{(2)}(M) \approx -(\sqrt{g} U^2)_{i,j-1/2} \delta\xi^1 \quad (2.70)$$

Treating the other faces in similar fashion we obtain

$$\oint_S U^\alpha dS_\alpha \approx (\sqrt{g} U^1)_{i-1/2,j}^{i+1/2,j} \delta\xi^2 + (\sqrt{g} U^2)_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \quad (2.71)$$

Using (2.61) one sees that this is exactly the same as (2.64).

Method 3. Straightforward integration in the ξ -plane using (2.41) gives

$$\int_\Omega U_{,\alpha}^\alpha d\Omega = \int_\Omega \frac{\partial \sqrt{g} U^\alpha}{\partial \xi^\alpha} d\xi^1 d\xi^2 \quad (2.72)$$

Partial integration results again in (2.71). We see that, apart from the change of variables (2.61), the three methods give the same discretization. Other discretizations are obtained by locating the geometric quantities in other points. But this will not be pursued further, because with the present choices we have the following desirable features: that the integral version of the geometric identity (2.50) holds for finite volumes, that $\int_S \mathbf{u} dS = 0$ for \mathbf{u} constant, and that constant vector fields are preserved when changing coordinates, as will be shown below.

Computation of Cartesian components. After a flow computation has been completed using staggered contravariant flux components V^α or velocity components U^α as unknowns it may be required, for example for output purposes, to compute the Cartesian velocity vector \mathbf{u} . This is not quite straightforward, because we do not have all components V^α or U^α available in the same point. Furthermore, it has been assumed that the base vectors $\mathbf{a}_{(\alpha)}$ are available componentwise only in the centers of the faces, according to (2.53). First, assume that U^α is used. We have

$$u^\alpha = a_{(\beta)}^\alpha U^\beta \quad (2.73)$$

We will evaluate \mathbf{u} in the vertices of the cells, as follows

$$u_{i+1/2, j+1/2}^\alpha = \frac{1}{4} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 U^1 + \Sigma_1 a_{(2)}^\alpha \Sigma_2 U^2 \} \quad (2.74)$$

where Σ_1 indicates summation over the gridpoints $(i + \frac{1}{2}, j)$ and $(i + \frac{1}{2}, j + 1)$ and Σ_2 indicates summation over the gridpoints $(i, j + \frac{1}{2})$ and $(i + 1, j + \frac{1}{2})$. If U^α is the contravariant representation of a constant vector field \mathbf{w} then it would be nice if we would get exactly

$$u_{i+1/2, j+1/2}^\alpha = w^\alpha \quad (2.75)$$

Substitution of (2.66) gives

$$\begin{aligned} u_{i+1/2, j+1/2}^\alpha &= \frac{1}{4} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 \frac{1}{\sqrt{g}} a_{(2)}^2 - \Sigma_1 a_{(2)}^\alpha \Sigma_2 \frac{1}{\sqrt{g}} a_{(1)}^2 \} w^1 \\ &\quad - \frac{1}{4} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 \frac{1}{\sqrt{g}} a_{(2)}^1 - \Sigma_1 a_{(2)}^\alpha \Sigma_2 \frac{1}{\sqrt{g}} a_{(1)}^1 \} w^2 \end{aligned} \quad (2.76)$$

Hence, in general $u_{i+1/2, j+1/2}^\alpha \neq w_{i+1/2, j+1/2}^\alpha$!

Next, assume that V^α is used. We have according to (2.61) and (2.66)

$$V^1 = a_{(2)}^2 w^1 - a_{(2)}^1 w^2, \quad V^2 = -a_{(1)}^2 w^1 + a_{(1)}^1 w^2 \quad (2.77)$$

Hence

$$\begin{aligned}
u_{i+1/2,j+1/2}^\alpha &= \frac{1}{4\sqrt{g}_{i+1/2,j+1/2}} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 V^1 + \Sigma_1 a_{(2)}^\alpha \Sigma_2 V^2 \} \\
&= \frac{1}{4\sqrt{g}_{i+1/2,j+1/2}} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 a_{(2)}^2 - \Sigma_1 a_{(2)}^\alpha \Sigma_2 a_{(1)}^2 \} w^1 \\
&\quad - \frac{1}{4\sqrt{g}_{i+1/2,j+1/2}} \{ \Sigma_2 a_{(1)}^\alpha \Sigma_1 a_{(2)}^1 - \Sigma_1 a_{(2)}^\alpha \Sigma_2 a_{(1)}^1 \} w^2 \quad (2.78)
\end{aligned}$$

If $\sqrt{g}_{i+1/2,j+1/2}$ is computed according to

$$\sqrt{g}_{i+1/2,j+1/2} = \frac{1}{4} \{ \Sigma_2 a_{(1)}^1 \Sigma_1 a_{(2)}^2 - \Sigma_1 a_{(2)}^1 \Sigma_2 a_{(1)}^2 \} \quad (2.79)$$

then (2.79) gives $u_{i+1/2,j+1/2}^\alpha = w_{i+1/2,j+1/2}^\alpha$.

Next, \mathbf{u} will be evaluated in the cell centers, again with V^α as unknowns, as follows:

$$u_{i,j}^\alpha = \frac{1}{4\sqrt{g}_{i,j}} \{ \Sigma_4 a_{(1)}^\alpha \Sigma_3 V^1 + \Sigma_3 a_{(2)}^\alpha \Sigma_4 V^2 \} \quad (2.80)$$

where Σ_3 indicates summation over the gridpoints $(i + \frac{1}{2}, j)$ and $(i - \frac{1}{2}, j)$ and Σ_4 indicates summation over the gridpoints $(i, j + \frac{1}{2})$ and $(i, j - \frac{1}{2})$. If $\frac{1}{\sqrt{g}} V^\alpha$ is the contravariant representation of a constant vector field \mathbf{w} we obtain

$$\begin{aligned}
u_{i,j}^\alpha &= \frac{1}{4\sqrt{g}_{i,j}} \{ \Sigma_4 a_{(1)}^\alpha \Sigma_3 a_{(2)}^2 - \Sigma_3 a_{(2)}^\alpha \Sigma_4 a_{(1)}^2 \} w^1 \\
&\quad - \frac{1}{4\sqrt{g}_{i,j}} \{ \Sigma_4 a_{(1)}^\alpha \Sigma_3 a_{(2)}^1 - \Sigma_3 a_{(2)}^\alpha \Sigma_4 a_{(1)}^1 \} w^2 \quad (2.81)
\end{aligned}$$

Hence, if $\sqrt{g}_{i,j}$ is computed according to

$$\sqrt{g}_{i,j} = \frac{1}{4} \{ \Sigma_4 a_{(1)}^1 \Sigma_3 a_{(2)}^2 - \Sigma_3 a_{(2)}^1 \Sigma_4 a_{(1)}^2 \} \quad (2.82)$$

then (2.81) gives $u_{i,j}^\alpha = w_{i,j}^\alpha$. We may conclude that constant vector fields are preserved under coordinate changes if the following conditions are satisfied:

- (i) Physical vector fields are evaluated at the *vertices* of the cells.
- (ii) \sqrt{g} is computed in the vertices according to (2.79).
- (iii) V^α is chosen to represent the vector field.

or if the following conditions are satisfied:

- (i) Physical vector fields are evaluated at the *centers* of the cells.

(ii) \sqrt{g} is computed in the centers according to (2.82).

(iii) V^α is chosen to represent the vector field.

The preservation property just discussed also holds in three dimensions, although the derivation is not straightforward. We will not go into this here.

Example 2.3.1. The difference between U^α and V^α .

Given an L-shaped domain into which a grid with a kink (Figure 2.6) is generated. We look at the transformation from the Cartesian to the computational domain and back. The input is $\mathbf{u} = (1, 0)^T$. The vectors in Figure (2.7), which do not represent exactly the vector $(1, 0)^T$ at the kink in the grid lines are the vectors \mathbf{u} after the following transformations:

$$\mathbf{u} \rightarrow U^\alpha \rightarrow \mathbf{u}.$$

The vectors $(1, 0)^T$ show \mathbf{u} after the following transformations:

$$\mathbf{u} \rightarrow V^\alpha \rightarrow \mathbf{u}.$$

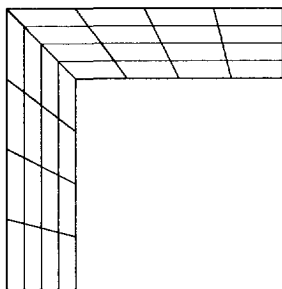


Figure 2.6: Grid for L-shaped region.

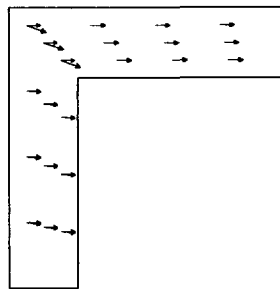


Figure 2.7: Vector $(1, 0)^T$ after transformation with different variables presenting the vector field.

We see that non-smoothness of the grid can cause significant inaccuracies in the contravariant representation of a constant vector field, if U^α is used to represent the field.

2.4 Finite volume discretization of a tensor equation.

As will be seen later, the Navier-Stokes momentum equations are of the following type:

$$T_{,\beta}^{\alpha\beta} = f^\alpha \quad (2.83)$$

For discretization (2.83) is integrated over finite volumes that will be specified later. As before, their faces satisfy $\xi^\alpha = \text{constant}$. We will discuss four methods to obtain a finite volume discretization of (2.83).

Method 1. Using (2.43) and partial integration we obtain

$$\begin{aligned} \int_{\Omega} T_{,\beta}^{\alpha\beta} d\Omega &= \int_{\Omega} \frac{\partial \sqrt{g} T^{\alpha\beta}}{\partial \xi^\beta} d\xi^1 d\xi^2 + \int_{\Omega} \left\{ \begin{matrix} \alpha \\ \gamma\beta \end{matrix} \right\} T^{\gamma\beta} d\Omega \\ &\approx \sqrt{g} T^{\alpha 1} \Big|_{i-\frac{1}{2},j}^{i+\frac{1}{2},j} \delta \xi^2 + (\sqrt{g} T^{\alpha 2}) \Big|_{i,j-\frac{1}{2}}^{i,j+\frac{1}{2}} \delta \xi^1 \\ &+ \int_{\Omega} \left\{ \begin{matrix} \alpha \\ \gamma\beta \end{matrix} \right\} T^{\gamma\beta} d\Omega \end{aligned} \quad (2.84)$$

Christoffel symbols have to be computed, which implies that the mapping $\mathbf{x}(\boldsymbol{\xi})$ has to be differentiated twice. This may result in inaccuracies when the mapping is non-smooth. Due to the persistence of the volume integrals, this formulation is called not conservative.

In the following methods (2.83) is contracted with two linearly independent vector fields $\mathbf{w}^{(\gamma)}$, $\gamma = 1, 2$. We have, using the chain rule and Gauss' Theorem:

$$\int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega = \oint_S W_{\alpha}^{(\gamma)} T^{\alpha\beta} dS_{\beta} - \int_{\Omega} W_{\alpha,\beta}^{(\gamma)} T^{\alpha\beta} d\Omega \quad (2.85)$$

The integral over S can be discretized with the methods of the preceding section. Different discretizations result, depending on the choice that is made for $\mathbf{w}^{(\gamma)}$.

Method 2. Tensor $W_{\alpha}^{(\gamma)}$ is given by

$$W_{\alpha}^{(\gamma)} = \delta_{\alpha}^{(\gamma)} \quad (2.86)$$

with δ the Kronecker delta. Figure 2.8(i) illustrates the corresponding vector field $\mathbf{w}^{(\gamma)}$. We have

$$W_{\alpha,\beta}^{(\gamma)} = -\left\{ \begin{matrix} \gamma \\ \alpha\beta \end{matrix} \right\} \quad (2.87)$$

as follows from (2.40). This gives

$$\int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega = \oint_S W_{\alpha}^{(\gamma)} T^{\alpha\beta} dS_{\beta} + \int_{\Omega} \left\{ \begin{matrix} \gamma \\ \alpha\beta \end{matrix} \right\} T^{\alpha\beta} d\Omega \quad (2.88)$$

Discretizing the integral over S according to (2.64) and using (2.86) in (2.88) gives

$$\begin{aligned} \int_{\Omega} T_{,\beta}^{\gamma\beta} d\Omega &= (\sqrt{g} T^{\gamma 1}) \Big|_{i-1/2,j}^{i+1/2,j} \delta \xi^2 + (\sqrt{g} T^{\gamma 2}) \Big|_{i,j-1/2}^{i,j+1/2} \delta \xi^1 \\ &+ \int_{\Omega} \left\{ \begin{matrix} \gamma \\ \alpha\beta \end{matrix} \right\} T^{\alpha\beta} d\Omega \end{aligned} \quad (2.89)$$

Hence method 2 is identical to method 1.

Method 3. Choose $\mathbf{w}^{(\gamma)}$ to be a constant vector field (Figure 2.8(ii)). As a consequence the derivative $W_{\alpha,\beta}^{(\gamma)}$ is zero and the volume integral vanishes, and (2.85) reduces to

$$\int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega = \oint_S W_{\alpha}^{(\gamma)} T^{\alpha\beta} dS_{\beta} \quad (2.90)$$

Choosing $w_{\beta}^{(\gamma)} = \delta_{\beta}^{(\gamma)}$ (2.90) can be written as:

$$\int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega = \oint_S a_{(\alpha)}^{\gamma} T^{\alpha\beta} dS_{\beta} \quad (2.91)$$

Discretizing the integral over S according to (2.64) gives

$$\begin{aligned} \int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega &= \sqrt{g} a_{(\alpha)}^{\gamma} T^{\alpha_1} |_{i-1/2,j}^{i+1/2,j} \delta\xi^2 \\ &+ \sqrt{g} a_{(\alpha)}^{\gamma} T^{\alpha_2} |_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \end{aligned} \quad (2.92)$$

This approximation is unstable under certain circumstances. The reason is that by the premultiplication by the vector $\mathbf{w}^{(\gamma)}$ each of the equations (2.83) is projected on a fixed direction which is not related to the orientation of the gridlines. When the coordinate transformation consists of rotation of the axes over 90° an unstable discretization results, as illustrated in Appendix A. This type of discretization may also be obtained by merely transforming the independent variables, while keeping the original Cartesian tensor components as unknowns (Cartesian approach). The advantage is the absence of Christoffel symbols.

Method 4. Choose $\mathbf{w}^{(\gamma)}$ constant (over the whole space) but different for each control volume, as follows (see Figure 2.8(iii)):

$$\mathbf{w}^{(\gamma)} = \text{constant}, \quad W_{\alpha}^{(\gamma)} |_{i,j} = \delta_{\alpha}^{(\gamma)} \quad (2.93)$$

Hence $W_{\alpha,\beta}^{(\gamma)} = 0$ and we obtain

$$\begin{aligned} \int_{\Omega} W_{\alpha}^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega &= \oint_S W_{\alpha}^{(\gamma)} T^{\alpha\beta} dS_{\beta} \\ &\approx (\sqrt{g} W_{\alpha}^{(\gamma)} T^{\alpha_1}) |_{i-1/2,j}^{i+1/2,j} \delta\xi^2 + \\ &(\sqrt{g} W_{\alpha}^{(\gamma)} T^{\alpha_2}) |_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \end{aligned} \quad (2.94)$$

This method is equivalent to the discretization described in terms of Gibbs' vector notation by Rosenfeld et al. ([88]). This approach combines advantages of the Cartesian and the contravariant discretization. Christoffel symbols are absent.

However, the directions of projection of the equations is given by the contravariant base vectors at the cell centers. So these directions are related to the orientation of the gridlines and for that reason the instability related to method 3 does not occur. When the coordinate transformation merely consists of a rotation of the axes the standard Cartesian staggered grid discretization is recovered, which is known to be stable. Because the volume integral vanishes this approach sometimes is called conservative, but this terminology is misleading here, because the flux over a given face is not the same for two adjoining cells, since each cell has its own $\mathbf{w}^{(\gamma)}$. The evaluation of $W_\alpha^{(\gamma)}$ at the cell faces is straightforward using the standard formulas (2.25) and (2.26):

$$w_\alpha^{(\gamma)} = \text{constant} = (a_\alpha^{(\varepsilon)} W_\varepsilon^{(\gamma)})_{i,j} = a_\alpha^{(\gamma)}{}_{i,j} \quad (2.95)$$

$$W_{\alpha \ i+1/2,j}^{(\gamma)} = a_{(\alpha) \ i+1/2,j}^\beta a_{\beta \ i,j}^{(\gamma)} \quad (2.96)$$

so that

$$(\sqrt{g} W_\alpha^{(\gamma)} T^{\alpha 1})_{i+1/2,j} = (\sqrt{g} a_{(\alpha)}^\beta T^{\alpha 1})_{i+1/2,j} a_{\beta \ i,j}^{(\gamma)} \quad (2.97)$$

Hence

$$\begin{aligned} \int_{\Omega} W_\alpha^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega &\approx a_{\beta \ i,j}^{(\gamma)} \{ (\sqrt{g} a_{(\alpha)}^\beta T^{\alpha 1})_{i-1/2,j}^{i+1/2,j} \delta\xi^2 \\ &+ (\sqrt{g} a_{(\alpha)}^\beta T^{\alpha 2})_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \} \end{aligned} \quad (2.98)$$

Formula (2.96) contains implicitly an approximation of a Christoffel symbol, though only first derivatives of the mapping are present. This can be shown by Taylor expansion of $W_{\alpha(i+1/2,j)}^{(\gamma)}$:

$$W_{\alpha(i+1/2,j)}^{(\gamma)} \approx W_{\alpha(i,j)}^{(\gamma)} + \frac{1}{2} \frac{\partial W_{\alpha(i,j)}^{(\gamma)}}{\partial \xi^1} \delta\xi^1 + O(\delta\xi^1)^2 + \dots \quad (2.99)$$

$W_{\alpha,1}^{(\gamma)}|_{i,j} = 0$, so (2.40) gives

$$\frac{\partial W_{\alpha(i,j)}^{(\gamma)}}{\partial \xi^1} = \left\{ \begin{matrix} \beta \\ \alpha 1 \end{matrix} \right\} W_{\beta(i,j)}^{(\gamma)} \quad (2.100)$$

and thus

$$W_{\alpha(i+1/2,j)}^{(\gamma)} \approx W_{\alpha(i,j)}^{(\gamma)} + \frac{1}{2} \left\{ \begin{matrix} \beta \\ \alpha 1 \end{matrix} \right\} W_{\beta(i,j)}^{(\gamma)} \delta\xi^1 + O(\delta\xi^1)^2 + \dots \quad (2.101)$$

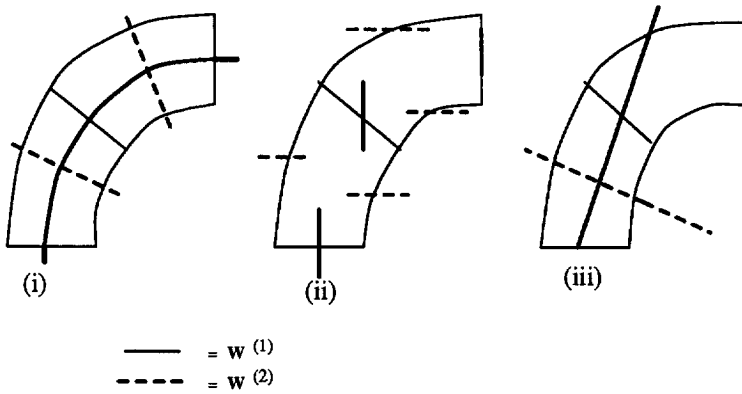


Figure 2.8: The representation of the w vector from the diverse methods in physical space. (i) a curved w orthogonal to the faces. (ii) a constant w , which is the same for all finite volumes. (iii) a constant w , which is different for each finite volume.

2.5 Conclusions.

Several discretization methods for the discretization of tensor equations have been discussed. Some requirements for the geometric quantities are derived in order to obtain accurate discretizations. It has been shown that flux unknowns V^α are to be preferred as primary unknowns, instead of contravariant velocity unknowns U^α .



3 Finite volume discretization of the incompressible Navier-Stokes equations in general coordinates.

3.1 Introduction.

Sections of this chapter are also presented in [113], [65] and [69].

Using the tensor analysis tools presented in the preceding chapter, the finite volume discretization of the incompressible Navier-Stokes equations in general coordinates is easily obtained after a discussion of some discretization aspects.

Velocity components. There is for example a choice in velocity components. The governing equations can be expressed in Cartesian or grid-oriented velocity components (shown in Figure 3.1) together with pressure. The specification of boundary conditions will be easier using grid-oriented velocity components, like the U^α -variables in Chapter 2. Using grid-oriented velocity components usually gives rise to curved coordinate lines along which the momentum equations are solved, like method 2 from Section 2.4, and thus to additional body force terms. When using Cartesian velocity unknowns the momentum equations are solved along the Cartesian (fixed) directions. No curvature terms will appear, but unstable discretizations like in method 3 from Section 2.4 and Appendix A might occur. Boundary conditions are more difficult to specify when the boundary is not aligned with the Cartesian axes, which may affect the accuracy near the boundary. Here grid-oriented unknowns will be used, namely the fluxes V^α , for reasons explained in Chapter 2.

Grid arrangement. In order to avoid non-physical pressure oscillations, which occur when all unknowns are stored in the center of a cell, Harlow and Welch ([38]) introduced the so-called staggered grid arrangement. Pressures are located at the cell centers, while velocity components are placed at cell faces. Staggering of variables is a successful technique for the incompressible Navier-Stokes equations, quite popular in Cartesian coordinates (see for example [107]) to avoid the oscillations mentioned. A disadvantage is that in a finite difference or finite volume scheme different control volumes need to be used for different variables, which does not simplify the programming of computer codes. A staggered grid with Cartesian velocity components leads to unstable discretizations in general coordinates, and therefore all Cartesian velocity unknowns should be stored at the cell faces and not just one per face. From 1983 the non-staggered approach was given new life by an article of Rhie and Chow ([82]). In [82] a special kind of artificial interpolation, which was shown to be more or less identical to the classical staggered approach, is proposed so that pressure oscillations can be avoided with a non-staggered grid arrangement. With the use of colocated grids the Cartesian velocity unknowns can

again be used as unknowns. In [74] a comparison is made between staggered and non-staggered discretizations for several problems in Cartesian coordinates. It was found that there was not much difference between the solutions.

In our research a staggered grid has been chosen. An advantage of this choice is that existing codes written for staggered grids in Cartesian coordinates can fairly easily be extended to arbitrary domains. The Cartesian equations need only to be extended with additional terms.

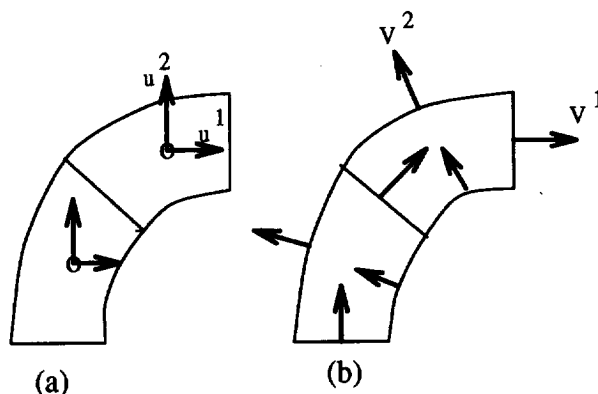


Figure 3.1: Cartesian and grid-oriented velocity components in a curved mesh. (a) Cartesian velocity components in a colocated arrangement, (b) Grid-oriented components in a staggered arrangement.

Literature. Some important publications concerning the numerical solution of the incompressible Navier-Stokes equations in general coordinates will be mentioned here. Three classes may be distinguished, namely publications about staggered grids, about colocated grids and publications using a pseudo-compressibility approach. The third class is actually a subset of the second but deserves to be distinguished from the other approaches because of significant differences. First, investigations on colocated grids with Cartesian velocity components as dependent variables are reported. In [84] this choice is made; here a clear comparison is given between the several choices to be made to obtain accurate discretizations. We mentioned already the pioneering papers by Rhie et al. ([82], [83]). This approach is also used in [62]. Many years of development of discretizations for non-staggered grids with Cartesian velocity unknowns is presented in [73], [22] and in [23], where a set of benchmark solutions is proposed for incompressible Navier-Stokes equations in non-orthogonal domains. A $k - \epsilon$ model is incorporated in a colocated approach in [15], containing references to earlier papers by the same authors. Other papers based on the colocated approach are [25], [26], where results with advanced up-wind schemes on colocated grids and fast solution methods are presented. In [52]

non-staggered grids are adopted to investigate the effect of several upwind schemes and turbulence models on the convergence of a multigrid solution technique for the incompressible Navier-Stokes equations. Some commercial codes have adopted colocated grids for solutions of complex flows in arbitrarily shaped domains, like *Flow3D*, ([46], [13]).

Discretizations on staggered grids are also applied on a large scale. Interesting papers have been published by Rosenfeld et al. ([88], [89], [87]), where on a staggered grid contravariant fluxes V^α and pressures are used as primary unknowns. The approach used is similar to method 4 described in the previous chapter. (The method 4-discretization is written down in more detail in Appendix B.) Two- and three-dimensional results with moving grids and multigrid acceleration are described in these papers. Another group adopting staggered grids for two- and three-dimensional problems is the so-called ISNaS group ([61], [119], [91]). Contravariant volume fluxes and pressure are dependent variables. Domain decomposition techniques and inclusion of turbulence models ([128]) are currently under investigation. In Japan staggered grids are widely investigated: in [44] for two-dimensional and in [45] for three-dimensional problems with dependent variables contravariant fluxes V^α and contravariant vorticities. In [48] physical components ([2]), are used with pressure as unknowns. A similar approach using the same unknowns can be found in [49].

Some other papers will be mentioned briefly. Discretization accuracy of the treatment of a pressure equation on a curvilinear staggered grid is presented in [6]. In [96] the stability problems on staggered grids with Cartesian velocity components are investigated and remedies are given. In [97] the staggered approach is chosen for the discretization in curvilinear coordinates, because of merits in boundary treatment, the compactness of effective mesh spacing, and the insensitivity of solutions with respect to the values of relaxation factors used in the iterative procedure. The group of Fletcher investigates flow problems on both staggered ([127]) and non-staggered ([16]) grids.

Another approach adopts pseudo-compressibility methods. With an artificial time-dependent pressure term in the continuity equation the incompressible Navier-Stokes equations can be handled with discretization and solution methods similar to those for the compressible Navier-Stokes equations. All unknowns are stored in the centers of grid cells; Cartesian velocity components and pressure are dependent variables. This method is adopted for discretizations in curvilinear coordinates by Kwak et al. ([50], [14], [85]) for example. An advanced upwind scheme based on flux difference splitting is incorporated in [85], a turbulence model in [50].

An approach different from all others is described in [19], where on a staggered grid covariant velocity components are used to solve the incompressible Navier-Stokes equations.

It is hard to say which approach is best in general coordinates. With pseudo-compressibility methods it is hard to obtain accurate mass conservation, especially

for time-dependent flows. The other non-staggered methods require artificial stabilization measures, that may influence accuracy negatively, and may require tuning. The staggered approach really needs invariant (e.g. contravariant) unknowns for stability, and leads to complicated discretization formulae in general coordinates, and lack of smoothness of the mapping has to be taken into account carefully. It has been decided to use a staggered grid with contravariant unknowns in the present work. The main motivation is that it is mathematically most elegant and straightforward to discretize a coordinate invariant physical law in a coordinate invariant way. The required techniques and insights can be applied to such laws in general, not just to the Navier-Stokes equations.

3.2 The incompressible Navier-Stokes equations in general coordinates.

A physical law in general coordinates can be found from the invariant formulation in a Cartesian coordinate system (Ricci's lemma). In Cartesian tensor notation the incompressible Navier-Stokes equations are given by:

$$u_{\alpha,\alpha} = 0 \quad (3.1)$$

$$\frac{\partial \rho u_\alpha}{\partial t} + (\rho u_\alpha u_\beta)_{,\beta} + p_{,\alpha} - \tau_{\alpha\beta,\beta} = \rho f_\alpha \quad (3.2)$$

where $\tau_{\alpha\beta}$ represents the deviatoric stress tensor given by:

$$\tau_{\alpha\beta} = \mu(u_{\alpha,\beta} + u_{\beta,\alpha}) \quad (3.3)$$

Here u_α are Cartesian velocity components, ρ is density (assumed constant per fluid particle), p is pressure and μ is the (turbulent or laminar) viscosity coefficient. Equations (3.1) and (3.2) are not invariant formulations. We have summation over two subscript indices, instead of a subscript and a superscript. However, in a Cartesian coordinate system $u_\alpha = u^\alpha = U^\alpha$, so that we can write (using (2.31)):

$$U_{,\alpha}^\alpha = 0 \quad (3.4)$$

$$\frac{\partial \rho U_\alpha}{\partial t} + (\rho U_\alpha U^\beta)_{,\beta} + p_{,\alpha} - \tau_{\alpha,\beta}^\beta = \rho f_\alpha \quad (3.5)$$

with $\tau_\alpha^\beta = \mu(g^{\beta\gamma} U_{\alpha,\gamma} + U_{,\alpha}^\beta)$.

In (3.5) U_α as well as U^β appears. This hampers the numerical treatment. Therefore we write (using again (2.31) and (2.44))

$$U_{,\alpha}^\alpha = 0 \quad (3.6)$$

$$\frac{\partial \rho U^\alpha}{\partial t} + (\rho U^\alpha U^\beta)_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau_{,\beta}^{\alpha\beta} = \rho f^\alpha \quad (3.7)$$

where $\tau^{\alpha\beta}$ is given by:

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma}U_{,\gamma}^{\beta} + g^{\gamma\beta}U_{,\gamma}^{\alpha}) \quad (3.8)$$

For a steady solution of the equations the steady variant of (3.7) is solved, which is given by:

$$T_{,\beta}^{\alpha\beta} \equiv (\rho U^{\alpha}U^{\beta})_{,\beta} + (g^{\alpha\beta}p)_{,\beta} - \tau_{,\beta}^{\alpha\beta} = \rho f^{\alpha} \quad (3.9)$$

3.3 Finite volume discretization.

The staggered grid. Figure 3.2 shows a part of the computational grid in the ξ -plane. The pressure and the density are given in the cell centers (i, j) (see Figure

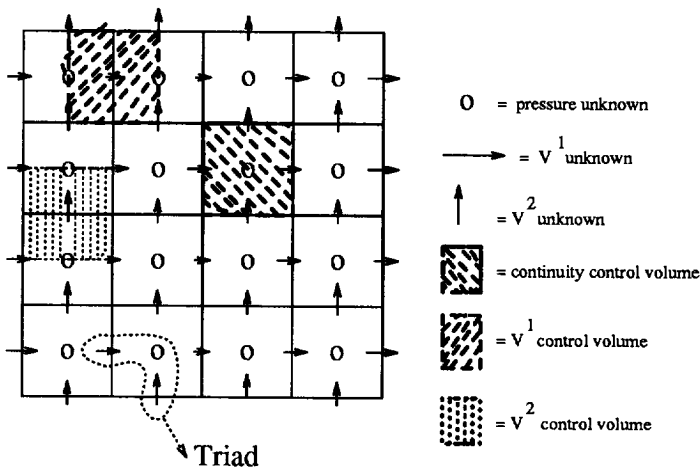


Figure 3.2: The rectangular grid in the computational domain with a staggered grid arrangement, control volumes and a triad.

2.4); V^1 is given in the centers of the cell faces $(i + 1/2, j)$ and V^2 is given in the centers of faces $(i, j + 1/2)$. Latin indices refer to points in the grid; Greek indices are used solely for tensor components. Figure 3.2 shows the arrangement of the unknowns together with their corresponding control volumes in the ξ -plane. The variables are grouped together in groups of three, sharing the same Latin indices. These groups are called triads.

The continuity equation. For the finite volume discretization of the continuity equation (3.6) the continuity control volumes shown in Figure 3.2 are used. In Section 2.3 it is shown that discretization of (3.6) leads to (2.67) or (2.71) depending on the type of unknowns used. However from the discussion in Section 2.3 it is clear that the unknown $V^{\alpha} = \sqrt{g}U^{\alpha}$ is preferable. For that reason we shall restrict ourselves

to the formulation in terms of V^α . For ease of reference we repeat the discretization (2.67):

$$\oint_S \mathbf{u} \cdot d\mathbf{S} \approx (-V_{i,j-1/2}^2 + V_{i,j+1/2}^2)\delta\xi^1 + (V_{i+1/2,j}^1 - V_{i-1/2,j}^1)\delta\xi^2 \quad (3.10)$$

The discretization of the momentum equations. We consider the space discretization of each of the terms of the steady equations (3.9). For the discretization of (3.9) we can use the methods discussed in Section 2.4. Method 2 (i.e. equation (2.89)) is used here. Discretization with method 4 from Section 2.4 is presented in Appendix B, to show the equivalence of the Gibbs' vector formulation in the physical domain in the articles of Rosenfeld et al. ([88], [89]), and our invariant tensor formulation according to method 4 in the computational domain.

For convenience we introduce a local numbering for each control volume, as shown in Figure 3.3. The center of the volume under consideration will have indices (0, 0). Because the discretization of the convection term will depend on the stress term

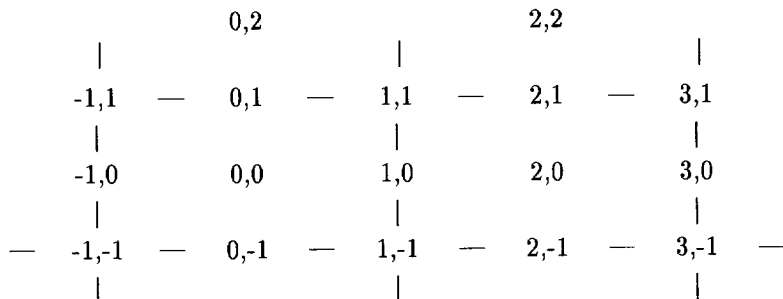


Figure 3.3: A local numbering system in a control volume.

because of the use of a hybrid difference scheme, the stress term is treated first.

The stress term. Due to the presence of contravariant derivatives in the stress tensor (3.8), this term is more difficult to discretize than the other terms. First we reduce the summation in $\tau^{\alpha\beta}$ to a sum over two terms using the incompressibility constraint. From (3.8) one immediately obtains

$$\begin{aligned} \tau^{11} &= 2\mu(g^{11}U_{,1}^1 + g^{12}U_{,2}^1) \\ \tau^{22} &= 2\mu(g^{12}U_{,1}^2 + g^{22}U_{,2}^2) \\ \tau^{21} = \tau^{12} &= \mu(g^{11}U_{,1}^2 + g^{22}U_{,2}^1) + \mu(g^{12}U_{,1}^1 + g^{12}U_{,2}^2) \\ &= \mu(g^{11}U_{,1}^2 + g^{22}U_{,2}^1) \end{aligned} \quad (3.11)$$

where we used $U_{,\alpha}^{\alpha} = 0$ and $g^{12} = g^{21}$. Equation (2.89) applied to the stress tensor for a V^1 control volume gives:

$$\begin{aligned}
 - \int_{\Omega} (\tau^{1\beta})_{,\beta} d\Omega &\approx -\sqrt{g} \tau^{11}|_{(-1,0)}^{(1,0)} \delta\xi^2 - \sqrt{g} \tau^{12}|_{(0,-1)}^{(0,1)} \delta\xi^1 - \left\{ \frac{1}{\gamma\beta} \right\} \tau^{\gamma\beta} \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2 \\
 &= -\sqrt{g} 2\mu(g^{11}U_{,1}^1 + g^{12}U_{,2}^1)|_{(-1,0)}^{(1,0)} \delta\xi^2 - \\
 &\quad \sqrt{g} \mu(g^{11}U_{,1}^2 + g^{22}U_{,2}^1)|_{(0,-1)}^{(0,1)} \delta\xi^1 - \\
 &\quad \left\{ \frac{1}{\gamma\beta} \right\} \tau^{\gamma\beta} \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2
 \end{aligned} \tag{3.12}$$

The same method applied to a V^2 -cell gives

$$\begin{aligned}
 - \int_{\Omega} (\tau^{2\beta})_{,\beta} d\Omega &\approx -\sqrt{g} \tau^{21}|_{(-1,0)}^{(1,0)} \delta\xi^2 - \sqrt{g} \tau^{22}|_{(0,-1)}^{(0,1)} \delta\xi^1 - \left\{ \frac{2}{\gamma\beta} \right\} \tau^{\gamma\beta} \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2 \\
 &= -\sqrt{g} \mu(g^{11}U_{,1}^2 + g^{22}U_{,2}^1)|_{(-1,0)}^{(1,0)} \delta\xi^2 - \\
 &\quad \sqrt{g} 2\mu(g^{12}U_{,1}^2 + g^{22}U_{,2}^2)|_{(0,-1)}^{(0,1)} \delta\xi^2 - \\
 &\quad \left\{ \frac{2}{\gamma\beta} \right\} \tau^{\gamma\beta} \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2
 \end{aligned} \tag{3.13}$$

Using the covariant derivative from (2.38) or (2.39) introduces extra Christoffel symbols. When (2.38) is used and subsequently the variables are changed: $U^{\alpha} = V^{\alpha}/\sqrt{g}$, problems occur when the coordinate mapping is non-smooth, e.g. when the physical cell size changes abruptly or when there is a kink in the grid lines. Using (2.39) will lead to more accurate discretizations under these circumstances ([91]).

The sensitivity of the accuracy to details of the implementation of the geometric quantities in the case of non-differentiable coordinate mappings is illustrated by the following example. Straightforward approximation of the viscous stress leads to terms (3.11). By using the following identity

$$- \frac{\partial g^{11}}{\partial \xi^1} = \left\{ \frac{1}{1\delta} \right\} g^{\delta 1} + \left\{ \frac{1}{1\delta} \right\} g^{1\delta} = 2 \left(\left\{ \frac{1}{11} \right\} g^{11} + \left\{ \frac{1}{12} \right\} g^{21} \right) \tag{3.14}$$

the stresses τ^{11} and τ^{22} can be rewritten as

$$\tau^{11} = 2\mu * \left[(g^{11})^{3/2} \frac{\partial}{\partial \xi^1} \left(\frac{V^1}{\sqrt{g_{22}}} \right) + g^{12} \frac{\partial}{\partial \xi^2} \left(\frac{V^1}{\sqrt{g}} \right) + \frac{V^2}{\sqrt{g}} (g^{11} \left\{ \frac{1}{12} \right\} + g^{12} \left\{ \frac{1}{22} \right\}) \right] \tag{3.15}$$

$$\tau^{22} = 2\mu * \left[(g^{22})^{3/2} \frac{\partial}{\partial \xi^2} \left(\frac{V^2}{\sqrt{g_{11}}} \right) + g^{12} \frac{\partial}{\partial \xi^1} \left(\frac{V^2}{\sqrt{g}} \right) + \frac{V^1}{\sqrt{g}} (g^{12} \left\{ \frac{2}{11} \right\} + g^{22} \left\{ \frac{2}{21} \right\}) \right] \tag{3.16}$$

Both versions (3.11), (3.13), and (3.15), (3.16) are applied to a test problem, a flow through a curved channel, where in the domain a non-smooth grid is generated.

Figure 3.4 shows that the second option is more accurate. This version is used in the sequel.

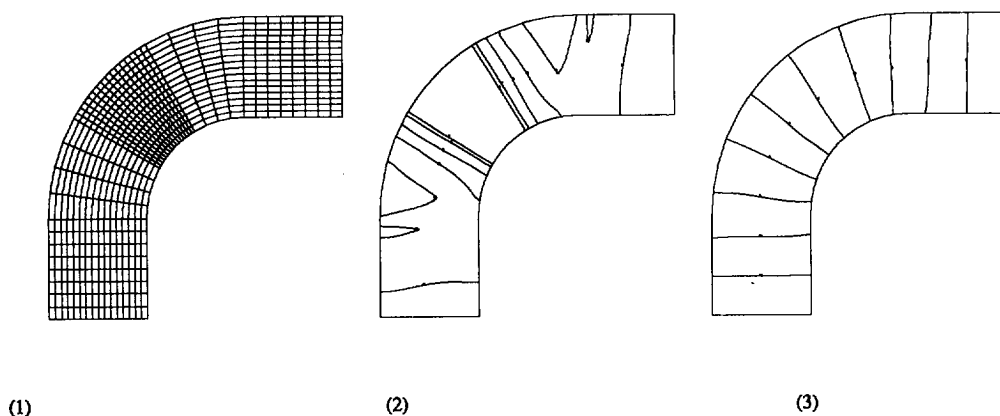


Figure 3.4: A low Reynolds flow through a curved channel. Comparison of two discretizations; (1) the non-smooth grid, (2) inaccurate isobars based on the straightforward approximation of the viscous stress, and (3) more accurate isobars based on the adapted version.

The structure of the resulting 13-point molecule for the stress tensor is sketched in Figure 3.7, with a numbering of unknowns used in the following.

The convective term. In order to obtain equations suitable for solution strategies like multigrid some form of upwind discretization has to be used for the convective terms. A brief explanation of the need for upwind discretization schemes is given. Suppose we have a linear operator D , given by

$$(D\psi)_{k,l} = \sum_{i,j} \alpha_{k,l}^{ij} \psi_{i,j} \quad (3.17)$$

in which k and l are part of the collection of indices I° or ∂I_N belonging to the grid points in the interior (G°) or on a Neumann boundary (∂G_N) or on a Dirichlet boundary (∂G_D) of domain G_h . This operator D is called a K -operator if:

$$(1) \sum_{i,j} \alpha_{k,l}^{ij} = 0, \quad (k,l) \in I^\circ \cup \partial I_N$$

$$(2) \alpha_{k,l}^{ij} \leq 0, \quad (k,l) \in I^\circ \cup \partial I_N, \quad i \neq k, \text{ or } j \neq l.$$

$$(3) \Sigma_{i,j} \alpha_{k,l}^{ij} > 0, \quad (k,l) \in \partial I_D \neq \emptyset$$

(4) The system (3.17) is irreducible (i.e. does not contain independent subsystems).

A matrix representing a K -operator is a K -matrix. The definition of a K -matrix is as follows:

The $n \times n$ matrix $A = (A_{pq})$ is called a K -matrix if:

$$(1) A_{pp} > 0$$

$$(2) A_{pq} \leq 0, \quad p \neq q$$

$$(3) A_{pq} \geq -\Sigma_{q=1; p \neq q}^n A_{pq}, \text{ with a strict inequality for at least one } q.$$

If matrix A is a K -matrix the numerical treatment of $A\psi = \mathbf{b}$ is favourable. It may be shown that a K -matrix is an M -matrix. For the definition of the concept of M -matrix, see [114]. The M -matrix property makes the efficient use of classical iterative methods possible ([114]). We want to use such methods for smoothing in a multigrid method. Therefore, supposing that all dependent variables except V^1 are known, we would like the V^1 -equation to be a K -operator; and similarly for V^2 . From the discretization of the stress tensor in the momentum equations a 13 point stencil results (see Figure 3.7). Because of reasons pointed out above we would like to get negative coefficients for the atoms 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13. This is not possible, because the stress term gives positive coefficients for some of these atoms. However, it always gives negative values to the atoms 3, 5, 6, 8, so we want to keep these coefficients negative. The convective terms give negative as well as positive contributions to these atoms. When a positive convection term is bigger than the absolute value of the stress term contributing to one of these atoms, the corresponding coefficient gets the wrong sign.

Hybrid difference scheme. To prevent this, we use the so-called hybrid discretization scheme for the convective terms ([71], [72]), a very popular scheme in Cartesian coordinates, introduced in [71].

The nonlinear convection tensor is linearized using

$$(\rho U^\alpha U^\beta)_{,\beta} \simeq (\rho U^{\alpha(n+1)} U^{\beta(n)})_{,\beta} \quad (3.18)$$

where the superscript n is an iteration index.

Equation (2.89) applied to $(\rho U^\alpha U^\beta)_{,\beta}$ leads to the following discretization of the convection term for a V^1 -cell (using $U^\alpha = \frac{1}{\sqrt{g}} V^\alpha$ and deleting the superscript $n+1$):

$$\begin{aligned}
\int_{\Omega} (\rho U^1 U^\beta)_{,\beta} d\Omega &\approx (\rho \frac{1}{\sqrt{g}} V^{1(n)} V^1) |_{(-1,0)}^{(1,0)} \delta\xi^2 + (\rho \frac{1}{\sqrt{g}} V^{2(n)} V^1) |_{(0,-1)}^{(0,1)} \delta\xi^1 \\
&+ \rho \frac{1}{\sqrt{g}} |_{(0,0)} (\{ \frac{1}{11} \} V^{1(n)} V^1 + \{ \frac{1}{12} \} V^{1(n)} V^2 \\
&+ \{ \frac{1}{12} \} V^{2(n)} V^1 + \{ \frac{1}{22} \} V^{2(n)} V^2) |_{(0,0)} \delta\xi^1 \delta\xi^2 \quad (3.19)
\end{aligned}$$

in which $V^{\alpha(n)}$ is a known estimate value of V^α .

Because many terms in (3.19) are evaluated in points where the respective quantities are not available, further approximation of (3.19) is required by means of interpolations. Here the hybrid discretization, choosing central discretization if possible and switching to an upwind discretization, when an atom would get the wrong sign, will be used. By choosing the interpolations appropriately we can obtain either central or upwind discretization or a mixture of the two for the flux-terms of the convection tensor, in order to satisfy the requirements formulated above. The volume term will be discretized with a central difference scheme. In the central discretization scheme the V^α unknowns are found by linear interpolation, for example:

$$V_{i,j}^1 = \frac{1}{2}(V_{i-1/2,j}^1 + V_{i+1/2,j}^1) \quad (3.20)$$

Here the numbering of Figure 2.4 is used.

In the first order upwind discretization scheme interpolation takes place as follows:

$$\begin{aligned}
(V^{1(n)} V^1)_{i,j} &= V_{i,j}^{1(n)} V_{i+1/2,j}^1 \quad \text{if } V^{1(n)} < 0 \\
&= V_{i,j}^{1(n)} V_{i-1/2,j}^1 \quad \text{if } V^{1(n)} > 0 \quad (3.21)
\end{aligned}$$

where $V_{i,j}^{1(n)}$ is evaluated with (3.20).

For the first flux term in (3.19) we can write in case of first order upwind discretization:

$$\begin{aligned}
(\rho \frac{1}{\sqrt{g}} V^{1(n)} V^1) |_{(-1,0)}^{(1,0)} \delta\xi^2 &\approx \{-\frac{1}{2} \frac{\rho}{\sqrt{g}} |_{(-1,0)} (|V^{1(n)}|_{(-1,0)} + V_{(-1,0)}^{1(n)}) V_{(-2,0)}^1 + \\
\frac{1}{2} \frac{\rho}{\sqrt{g}} |_{(-1,0)} (|V^{1(n)}|_{(-1,0)} + V_{(-1,0)}^{1(n)}) &+ \frac{1}{2} \frac{\rho}{\sqrt{g}} |_{(1,0)} (|V^{1(n)}|_{(1,0)} - V_{(1,0)}^{1(n)}) V_{(0,0)}^1 - \\
\{\frac{1}{2} \frac{\rho}{\sqrt{g}} |_{(1,0)} (|V^{1(n)}|_{(1,0)} - V_{(1,0)}^{1(n)}) &V_{(2,0)}^1\} \delta\xi^2 \quad (3.22)
\end{aligned}$$

Equation (3.22) is equivalent to (3.21) but does not contain a conditional statement, which is favourable for efficient coding on vector machines.

For the choice between upwind and central discretization the ratio of the absolute

magnitudes of the stress and the convection is important. Therefore the mesh-Reynolds number $Re^{(k,l)}$ is defined to be this ratio, for example

$$Re^{(1,3)} = \left| \frac{0.5\rho V^{2(n)}/\sqrt{g}\delta\xi^2}{stress} \right|_{(0,-1)} \quad (3.23)$$

It is clear that with the central difference scheme atom 3 will be positive when $Re^{(1,3)} > 1$. This will be avoided by switching from central to upwind discretization when $Re^{(k,l)} > 1$. Depending on the mesh-Reynolds number ($Re^{(k,3)}$, $Re^{(k,5)}$, $Re^{(k,6)}$, and $Re^{(k,8)}$, $k = 1, 2$) the convection flux terms will be discretized as follows:

$$\begin{aligned} Re^{(k,l)} \leq 1 &: \text{central discretization,} \\ Re^{(k,l)} > 1 &: \text{upwind discretization.} \end{aligned} \quad (3.24)$$

This is done as follows:

$$\text{CONV. TERM} = (1 - \alpha^{(k,l)})\text{CENTRAL} + \alpha^{(k,l)}\text{UPWIND} \quad (3.25)$$

with

$$\alpha^{(k,l)} = \frac{Re^{(k,l)} - 1 + |Re^{(k,l)} - 1|}{2|Re^{(k,l)} - 1| + \epsilon} \quad (3.26)$$

where ϵ is a small number to prevent the denominator to be zero.

A similar treatment is given to the V^2 convection term:

$$\begin{aligned} \int_{\Omega} (\rho U^2 U^\beta)_{,\beta} d\Omega &\approx (\rho \frac{1}{\sqrt{g}} V^{1(n)} V^2)_{(-1,0)}^{(1,0)} \delta\xi^2 + (\rho \frac{1}{\sqrt{g}} V^2 V^{2(n)})_{(0,-1)}^{(0,1)} \delta\xi^1 \\ &+ \rho \frac{1}{\sqrt{g}} \{ \begin{matrix} 2 \\ 11 \end{matrix} \}_{(0,0)} V^{1(n)} V^1 + \{ \begin{matrix} 2 \\ 12 \end{matrix} \}_{(0,0)} V^{1(n)} V^2 \\ &+ \{ \begin{matrix} 2 \\ 12 \end{matrix} \}_{(0,0)} V^{2(n)} V^1 + \{ \begin{matrix} 2 \\ 22 \end{matrix} \}_{(0,0)} V^{2(n)} V^2 \delta\xi^1 \delta\xi^2 \end{aligned} \quad (3.27)$$

A "sudden" switch from central to upwind discretization (as (3.24) or (3.25), (3.26)) is shown graphically in Figure 3.5. However, a sudden switch may cause convergence difficulties in the neighbourhood of $Re^{(k,l)} = 1$, because stencils are different at the switch. A "smoother" switch will be used in order to avoid these problems:

$$\begin{aligned} Re^{(k,l)} \leq 0.9 &: \quad \alpha^{(k,l)} = 0 \\ 0.9 < Re^{(k,l)} \leq 1.1 &: \quad \alpha^{(k,l)} = 5Re^{(k,l)} - 4.5 \\ Re^{(k,l)} > 1.1 &: \quad \alpha^{(k,l)} = 1 \end{aligned} \quad (3.28)$$

This smoother switch (shown in Figure 3.5) does not lead to convergence problems.

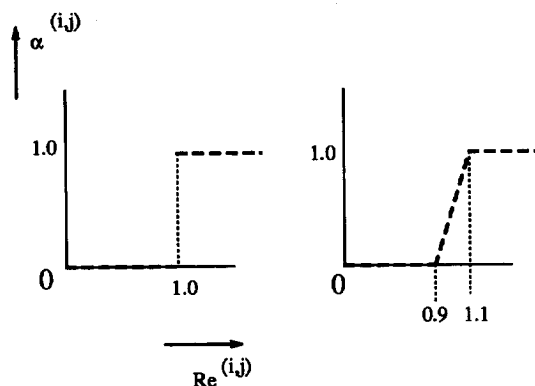


Figure 3.5: A sudden switch and a smooth switch from central to upwind discretization.

The structure of the molecules corresponding to (3.22) is given in Figure 3.7.

Literature. Upwind difference schemes are well-known in the literature. Upwind schemes are only first order accurate and may lead to non-physical boundary layers, due to numerical diffusion. The hybrid difference scheme is a satisfactory remedy (also used in [84] in general coordinates) for these problems. Second order accuracy can be obtained with the so-called defect correction technique ([39], [118]). First order accurate upwind alternatives leading to less numerical diffusion are schemes into which a preferred direction (weighted upstream) is taken into account, like the Skewed Upwind Difference Scheme developed by Raithby et al. ([79], [80], [30] and [20]). In these articles the deficiencies of upwind schemes are also shown. Discretizations using a larger stencil for the convective terms are the second order accurate upwind scheme and the quite popular third order accurate QUICK scheme ([51]). They are for example investigated in [124] and [95]. QUICK formally has no numerical diffusion, but it is not monotone. In flow regions where steep gradients occur it is found to give oscillatory solutions. Advanced upwind methods are derived in the field of compressible Euler equations with the introduction of limiters. An overview of the so-called TVD- (Total Variation Diminishing) schemes can be found in [40], [41]. These sophisticated upwind schemes have been used for the incompressible Navier-Stokes equations in general coordinates on a staggered grid in [44] and [45], where a Chakravarthy-Osher TVD scheme is used for the convective terms and a modified QUICK scheme. On collocated grids these upwind techniques are used in a flux difference formulation in [25]; in a pseudo-compressibility method in [85]. In [52] several upwind schemes for the incompressible Navier-Stokes equations are used on non-staggered grids: a hybrid difference scheme, a QUICK scheme and a TVD-type scheme.

The higher order upwind schemes do not give a K-operator, and will not be considered here.

The pressure term. Applying (2.89) to the pressure term gives,

$$\int_{\Omega} (g^{1\beta} p)_{,\beta} d\Omega \approx g^{11} \sqrt{g} p|_{(-1,0)}^{(1,0)} \delta\xi^2 + \sqrt{g} g^{12} p|_{(0,-1)}^{(0,1)} \delta\xi^1 + \left\{ \frac{1}{\gamma\beta} \right\} g^{\gamma\beta} p \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2 \quad (3.29)$$

and to a V^2 -cell:

$$\int_{\Omega} (g^{2\beta} p)_{,\beta} d\Omega \approx g^{21} p \sqrt{g}|_{(-1,0)}^{(1,0)} \delta\xi^2 + g^{22} p \sqrt{g}|_{(0,-1)}^{(0,1)} \delta\xi^1 + \left\{ \frac{2}{\gamma\beta} \right\} g^{\gamma\beta} p \sqrt{g}|_{(0,0)} \delta\xi^1 \delta\xi^2 \quad (3.30)$$

On a non-smooth grid discretizing all Christoffel symbols in the pressure terms according to (2.35) was found to cause non-physical pressure distributions on heavily distorted grids. This is avoided by eliminating the Christoffel symbols from the pressure terms in (3.29) and (3.30) with the relation:

$$\sqrt{g} \left\{ \frac{\alpha}{\beta\gamma} \right\} g^{\beta\gamma} = - \frac{\partial(\sqrt{g} g^{\alpha\beta})}{\partial\xi^\beta} \quad (3.31)$$

Figure 3.6 shows an example of a Poiseuille flow on an irregular grid. The grid is shown, as are the isobars, when the Christoffel symbols in the pressure term are kept and discretized according to (2.35). These isobars are not correct. Figure 3.6 shows correct isobars obtained by using (3.31).

The structure of the molecules coming from the pressure terms of the V^1 -momentum equation is given in Figure 3.7.

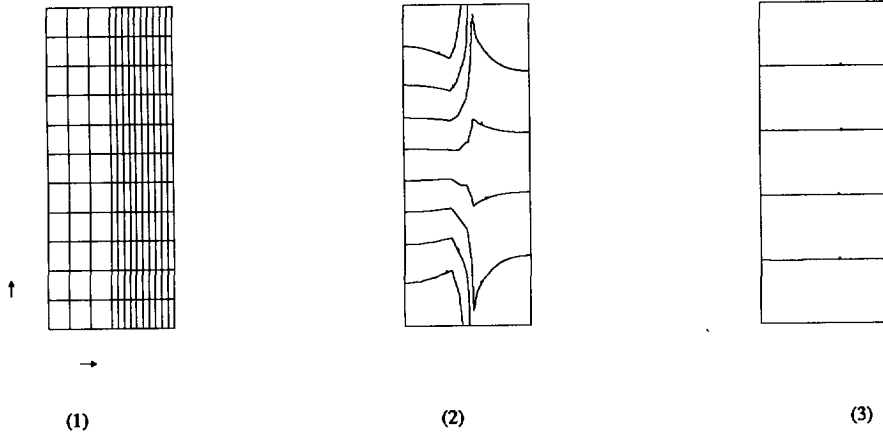


Figure 3.6: A Poiseuille flow on an irregular grid, (1) the grid, (2) incorrect isobars obtained with Christoffel symbols, (3) accurate isobars obtained with Christoffel symbols eliminated.

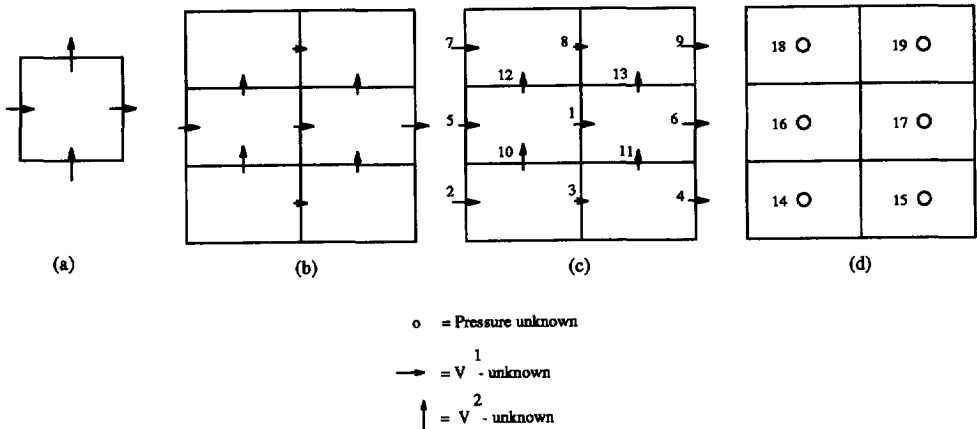


Figure 3.7: The V^1 -molecules for the separate terms in the spatial discretization. (a) the continuity, (b) the convective term, (c) the stress term, (d) the pressure term.

3.4 Benchmark solutions.

Recently two benchmark problems were proposed in [23] for discretizations on non-orthogonal grids in two dimensions in order to compare accuracy. The Navier-Stokes equations were solved on a non-staggered grid with Cartesian velocity unknowns; the discretization is described in [22]. Here these benchmark problems are solved for the discretization on a staggered grid presented in the previous sections. The flow problems are driven skewed cavity problems. The domain, depicted in Figure 3.8, is a parallelogram with boundary length $L = 1$. Angle β is 45° for case 1, and 30° for case 2, so that highly non-orthogonal cells occur in the \mathbf{x} -image of the grid. The top wall is moving with Cartesian velocity components $u^1 = 1, u^2 = 0$. On all other boundaries $\mathbf{u} = 0$ is prescribed. The flow is calculated for Reynolds numbers (Re) equal to 100 and 1000. The discretization is investigated on four

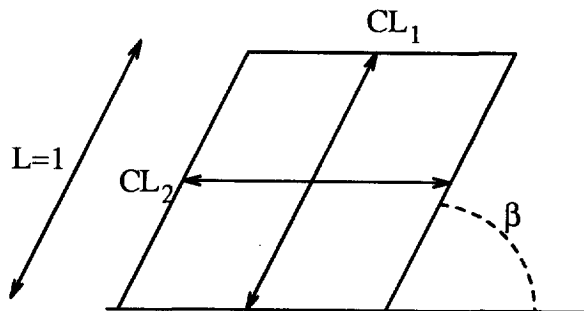


Figure 3.8: Domain for the skewed driven cavity problem.

grids, consisting of 32×32 , 64×64 , 128×128 and 256×256 cells, respectively. The streamline patterns obtained agree closely with those in [23]. They are presented in Figure 3.9 for $\beta = 45^\circ$, and in Figure 3.10 for $\beta = 30^\circ$. The prescribed streamline values are identical to those in [23]. The maximum and minimum values of the stream function Ψ_{max} and Ψ_{min} are determined for all grids together with their coordinates (x_{min}^1, x_{min}^2) and (x_{max}^1, x_{max}^2) and an error measure ϵ , defined as in [23]:

$$\epsilon = \left| \frac{\Psi_{256} - \Psi_{128}}{\Psi_{256}} \right| \cdot 100 \quad (3.32)$$

These values are presented in Table 3.1 for $\beta = 45^\circ$ and for $\beta = 30^\circ$. Results agree very well with [23]. The local error is less than 0.03 % of the maximum of the absolute values of Ψ over the domain for $\beta = 45^\circ$ and less than 0.1 % for $\beta = 30^\circ$. The results on all grids are accurate for $Re = 100$. For $Re = 1000$ the values for the 128×128 - and the 256×256 -grid do not differ much. This indicates that the exact solution is closely approximated on these grids. This is also true for velocity profiles along the centerlines CL_1 and CL_2 (Figure 3.8). Figure 3.11a shows the

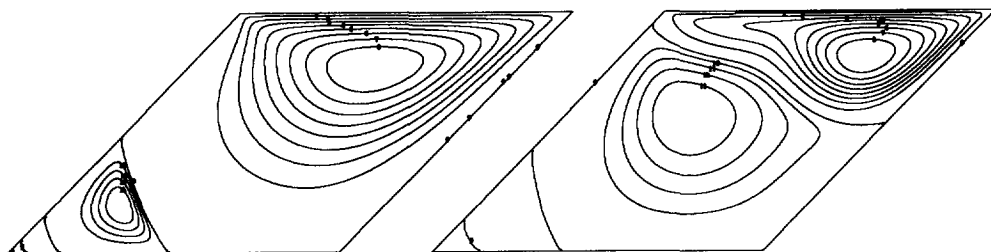


Figure 3.9: Streamlines for $Re = 100$ and $Re = 1000$ for testcase 1, skewness angle $\beta = 45^\circ$.

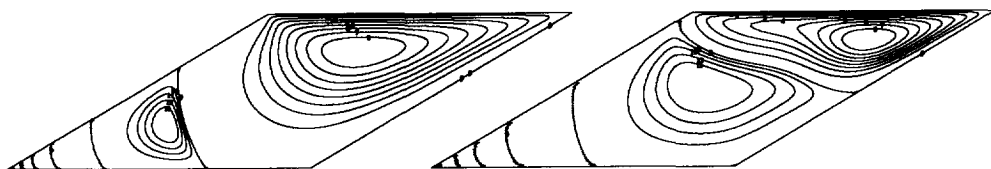


Figure 3.10: Streamlines for $Re = 100$ and $Re = 1000$ for testcase 2, skewness angle $\beta = 30^\circ$.

Cartesian velocity component u^1 along line CL_1 for $Re = 100$, $\beta = 45^\circ$, Figure 3.11b gives u^2 along CL_2 . Figure 3.12a shows u^1 along line CL_1 for $Re = 1000$, $\beta = 45^\circ$, Figure 3.12b presents u^2 along CL_2 . Figure 3.13a shows u^1 along line CL_1 for $Re = 100$, $\beta = 30^\circ$, Figure 3.13b gives u^2 along CL_2 . Figure 3.14a shows u^1 along line CL_1 for $Re = 1000$, $\beta = 30^\circ$, Figure 3.14b presents u^2 along CL_2 . In all figures some selected reference values from [23] are presented as well. They agree very well with the velocity profiles depicted. Again for $Re = 100$ all curves are identical. These velocity profiles are given in Figures 3.11 and 3.13.

Table 3.1: Minimum and maximum stream function values in vortex centers and their position for all grids and for both Reynolds numbers, skewed cavity, $\beta = 45^\circ$ and $\beta = 30^\circ$.

$\beta = 45^\circ$	$Re = 100$		$Re = 1000$	
	min.	max.	min.	max.
32×32				
Ψ	-7.0260×10^{-2}	5.6150×10^{-5}	-4.6914×10^{-2}	7.3515×10^{-3}
x	1.1149	0.3422	1.2933	0.7415
y	0.5524	0.1547	0.5745	0.3978
64×64				
Ψ	-7.0266×10^{-2}	4.1058×10^{-5}	-5.1778×10^{-2}	8.6275×10^{-3}
x	1.1039	0.3468	1.3089	0.7682
y	0.5414	0.1436	0.5745	0.4088
128×128				
Ψ	-7.0253×10^{-2}	3.7725×10^{-5}	-5.3456×10^{-2}	1.0024×10^{-2}
x	1.1094	0.3401	1.3089	0.7783
y	0.5496	0.1436	0.5745	0.4033
256×256				
Ψ	-7.0238×10^{-2}	3.6932×10^{-5}	-5.3523×10^{-2}	1.0039×10^{-2}
x	1.1100	0.3390	1.3128	0.7775
y	0.5469	0.1409	0.5745	0.4005
ϵ	0.0214	2.1472	0.1252	0.1494

$\beta = 30^\circ$	$Re = 100$		$Re = 1000$	
	min.	max.	min.	max.
32×32				
Ψ	-5.3045×10^{-2}	1.5597×10^{-4}	-3.7309×10^{-2}	4.2221×10^{-3}
x	1.1808	0.5519	1.4537	0.8976
y	0.3750	0.1563	0.4063	0.2656
64×64				
Ψ	-5.3126×10^{-2}	7.4524×10^{-5}	-3.8833×10^{-2}	4.4938×10^{-3}
x	1.1651	0.3750	1.4537	0.8997
y	0.5384	0.1484	0.4063	0.2578
128×128				
Ψ	-5.3152×10^{-2}	5.9527×10^{-5}	-3.8698×10^{-2}	4.2446×10^{-3}
x	1.1719	0.5316	1.4526	0.8997
y	0.3789	0.1445	0.4102	0.2578
256×256				
Ψ	-5.3149×10^{-2}	5.6228×10^{-5}	-3.8600×10^{-2}	4.1657×10^{-3}
x	1.1680	0.5291	1.4565	0.9036
y	0.3789	0.1426	0.4102	0.2559
ϵ	0.0060	5.8672	0.2539	1.8940

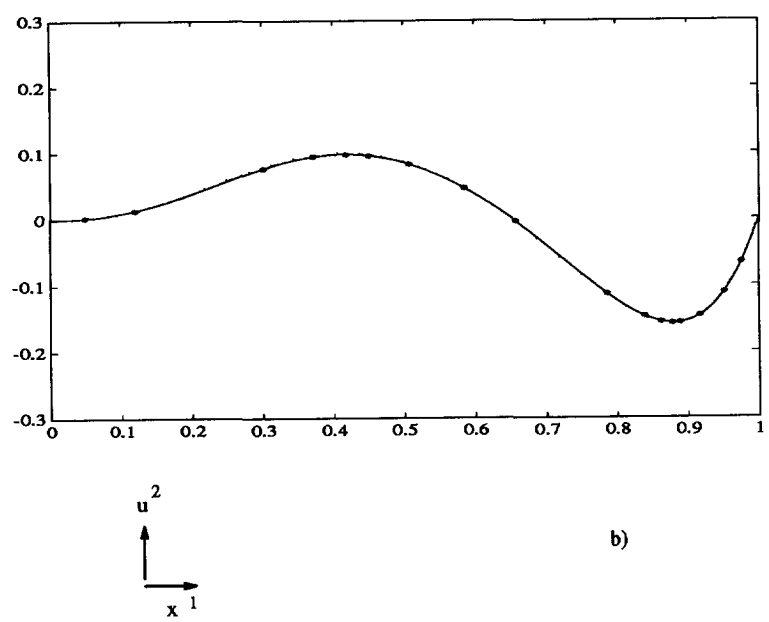
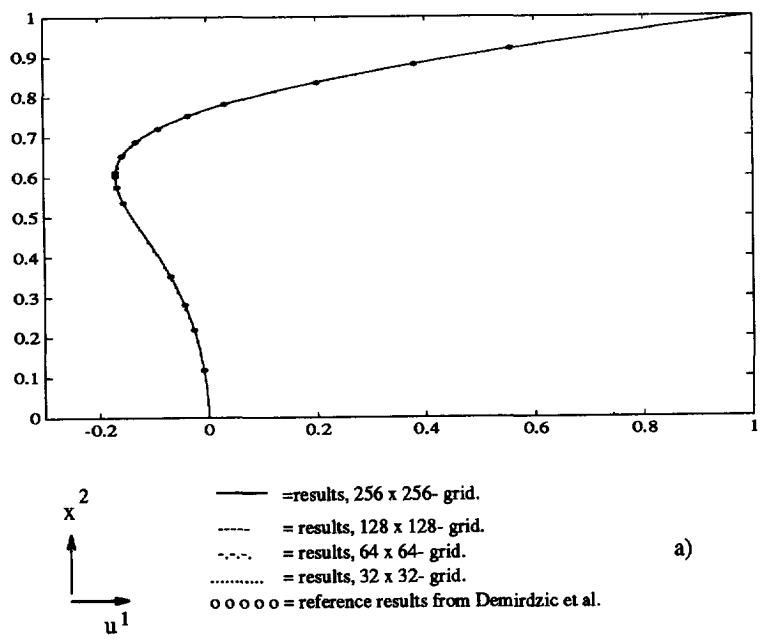


Figure 3.11: Velocity profiles for $\beta = 45^\circ$; $Re = 100$; a) u^1 along CL_1 ; b) u^2 along CL_2 .

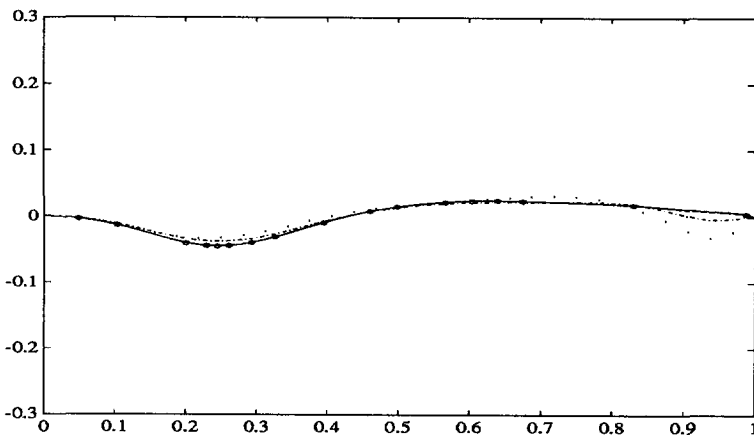
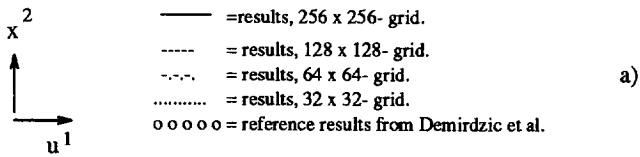
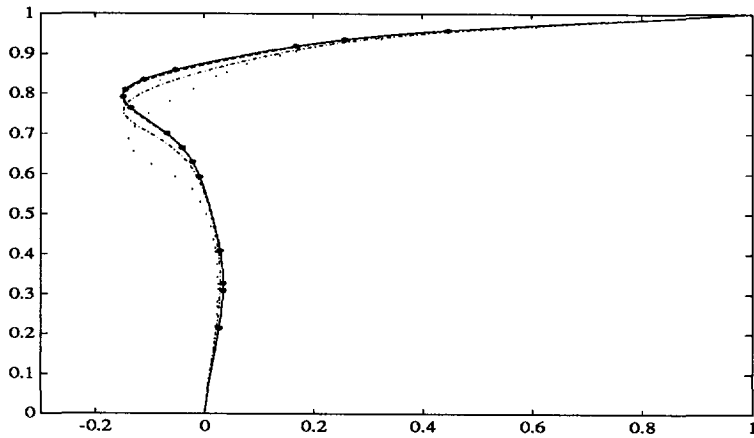
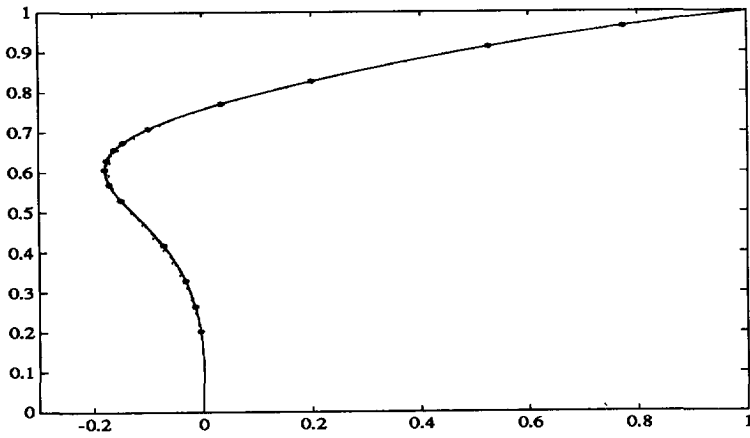
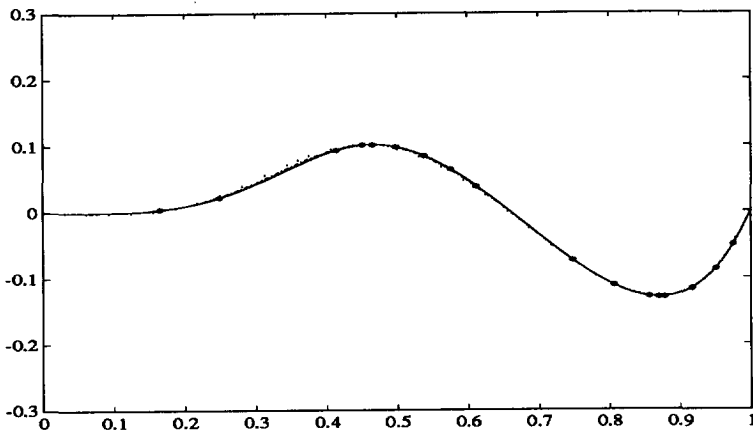


Figure 3.12: Velocity profiles for $\beta = 45^\circ$; $Re = 1000$; a) u^1 along CL_1 ; b) u^2 along CL_2 .



— = results, 256 x 256- grid.
 - - - = results, 128 x 128- grid.
 ····· = results, 64 x 64- grid.
 = results, 32 x 32- grid.
 o o o o o = reference results from Demirdzic et al.

a)



b)

Figure 3.13: Velocity profiles for $\beta = 30^\circ$; $Re = 100$; a) u^1 along CL_1 ; b) u^2 along CL_2 .

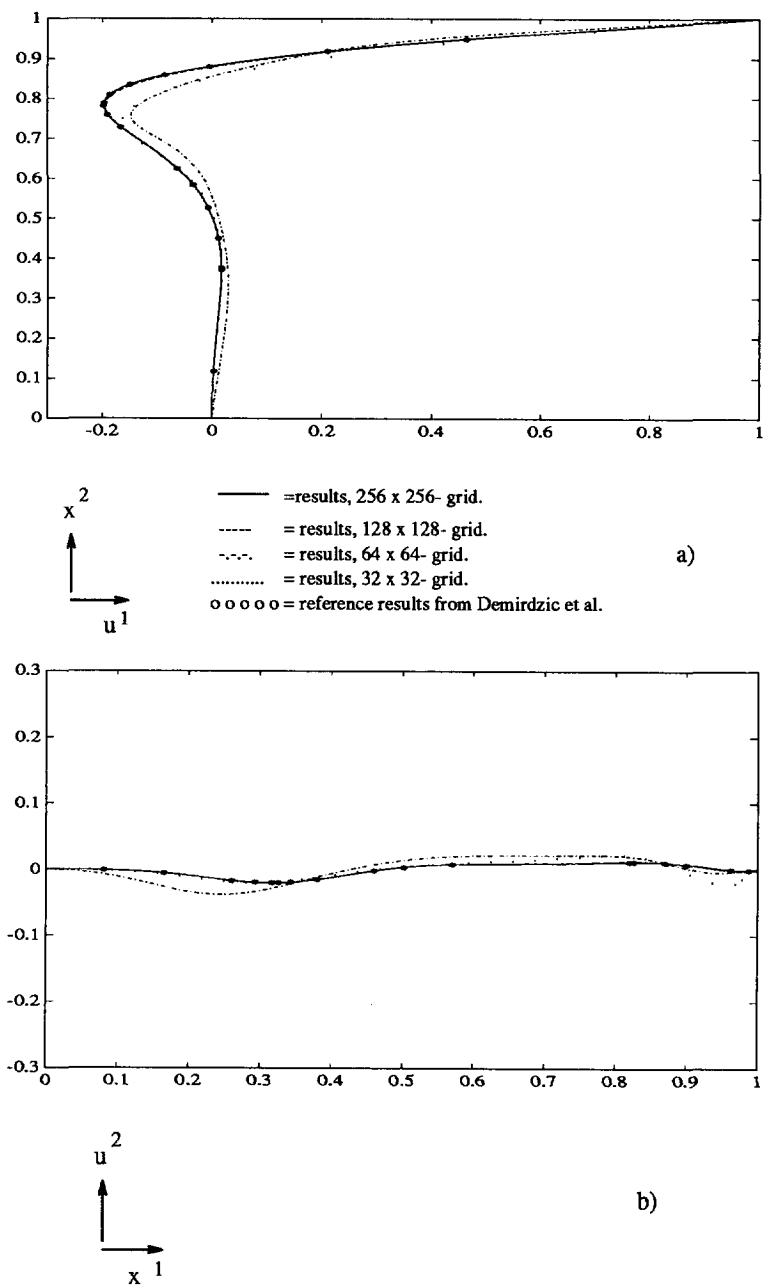


Figure 3.14: Velocity profiles for $\beta = 30^\circ$; $Re = 1000$; a) u^1 along CL_1 ; b) u^2 along CL_2 .

L-shaped cavity. We introduce another benchmark problem, because the grids of testcases 1 and 2 mainly test the effect of non-orthogonality of coordinates. Results for this benchmark problem will also appear in *International Journal for Numerical Methods in Fluids*. In order to also include the effect of curvature of coordinate lines (such testcases are also presented in [23], but only for buoyancy driven flows) we propose the problem of flow in an L-shaped driven cavity. This test problem is not only interesting for testing discretizations, but also for domain decomposition techniques, as used in [75]. The domain is depicted in Figure 3.15. It can be seen

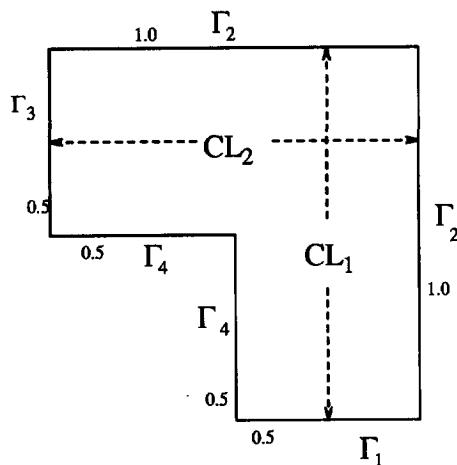


Figure 3.15: The domain for an L-shaped cavity problem.

in Figure 3.15 that boundaries Γ_2 and Γ_4 consist of two parts. On the upper part of Γ_2 $u^1 = 1, u^2 = 0$ is prescribed, on all other boundary parts we have $\mathbf{u} = 0$. The Reynolds numbers investigated are again 100 and 1000. If the grid is non-smooth, like the one shown in Figure 3.16, our discretization is not accurate for high Reynolds numbers ($Re = 1000$). Figure 3.17 shows for this 128×128 -grid streamlines for $Re = 100$ and $Re = 1000$. It can be seen that the non-smoothness of the grid causes artificial non-smoothness in the streamlines near the kink in the grid lines for $Re = 1000$.

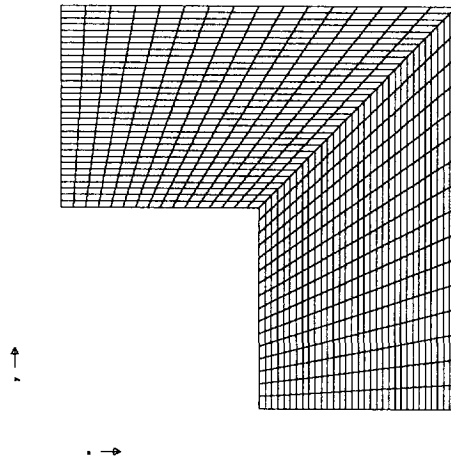


Figure 3.16: A non-smooth grid in an L-shaped cavity.

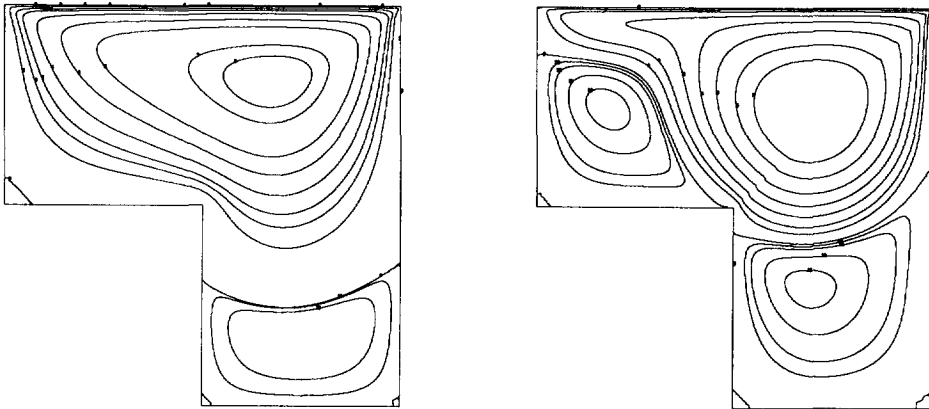


Figure 3.17: Streamlines for $Re = 100$ and $Re = 1000$ for an L-shaped cavity with the grid of Figure 16 (128×128 -cells).

Inaccuracy of the discretization on the grid of Figure 3.16 is to be expected with the method chosen due to the presence of Christoffel symbols, which involve second derivatives of the coordinate mapping. More smooth grids are needed, like the grid in Figure 3.18, constructed with a bi-harmonic grid generator ([102]). A generating system of higher order, like the biharmonic grid generator allows more boundary conditions. Coordinate line angles are specified to be orthogonal at the boundary. (With a Laplace grid generator we also obtained smooth grids.) Figures 3.19 and 3.20 present streamlines and isobars for $Re = 100$ and $Re = 1000$, respectively. Local maximum and minimum values of the stream function Ψ_{max}

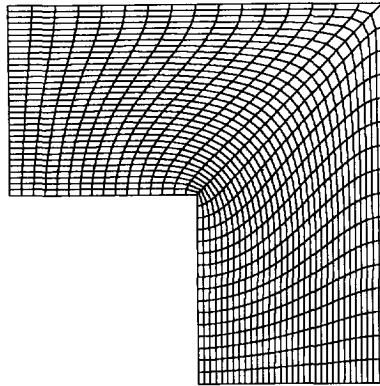


Figure 3.18: A smooth grid, in an L-shaped cavity obtained with a bi-harmonic grid generator.

and Ψ_{min} are presented for all grids together with their coordinates (x_{min}^1, x_{min}^2) and (x_{max}^1, x_{max}^2) in Table 3.2. From Figure 3.20 it can be seen that the flow at $Re = 1000$ has three streamfunction extrema, one for $x^1 < 0.5$ and two for $x^1 > 0.5$. All are given in Table 3.2. In order to judge accuracy as well as possible a 512×512 -grid is needed for $Re = 1000$. One should keep in mind that the moving part of boundary Γ_2 is then discretized with 256 cells, like in the first two testcases. The results obtained on the 256×256 -grid and the 512×512 -grid are identical to a satisfactory extent. For $Re = 100$ the finest grid is a 256×256 -grid. The error

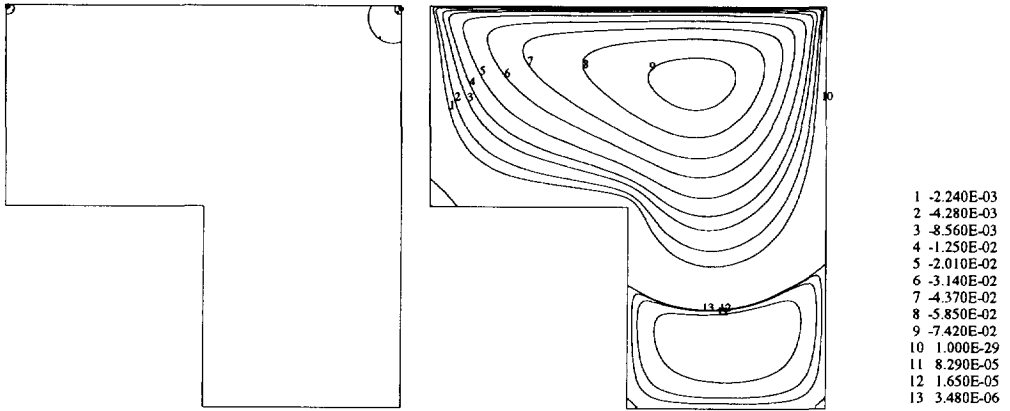


Figure 3.19: Isobars and streamlines for an L-shaped cavity for $Re = 100$ with the grid of Figure 18; 256×256 cells.

measure ϵ is defined for the finest grid as in (3.32). We may conclude that the discretization error of our discretization method shows satisfactory convergence on sufficiently smooth but otherwise general grids.

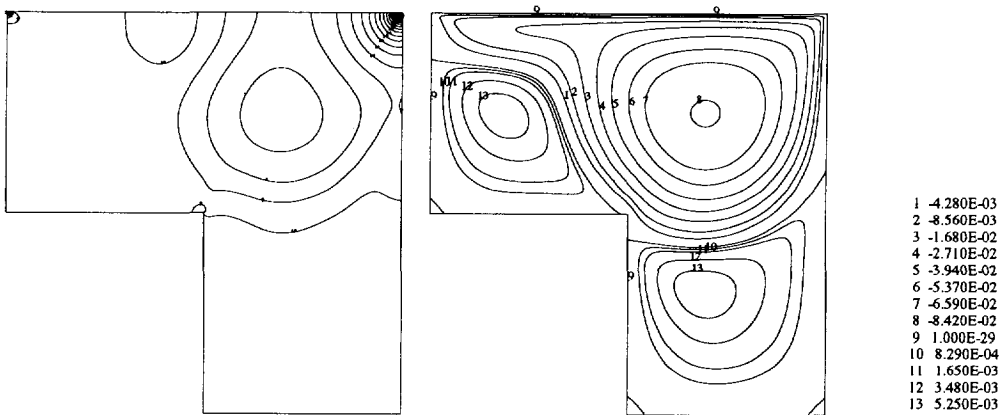


Figure 3.20: Isobars and streamlines for an L-shaped cavity for $Re = 1000$ with the grid of Figure 18; 256×256 cells.

Table 3.2: Minimum and maximum stream function values in vortex centers and their position for all grids, L-shaped cavity, for both Reynolds numbers; grids similar to Figure 18. (For $Re = 100$ the finest grid consists of 256×256 cells.)

	$Re = 100$		$Re = 1000$		
	min.	max.	min.	max. ($x^1 < 0.5$)	max. ($x^1 > 0.5$)
32^2					
Ψ	-8.118×10^{-2}	3.331×10^{-4}	-6.157×10^{-2}	4.382×10^{-3}	3.518×10^{-3}
x	0.6737	0.7518	0.7326	0.1904	0.6589
y	0.8184	0.1720	0.7326	0.6895	0.2521
64^2					
Ψ	-8.107×10^{-2}	2.795×10^{-4}	-7.322×10^{-2}	5.285×10^{-3}	6.362×10^{-3}
x	0.6679	0.7599	0.7026	0.1845	0.6604
y	0.8134	0.1747	0.7503	0.7289	0.2886
128^2					
Ψ	-8.092×10^{-2}	2.631×10^{-4}	-8.342×10^{-2}	6.701×10^{-3}	5.808×10^{-3}
x	0.6763	0.7643	0.6983	0.1825	0.6855
y	0.8092	0.1747	0.7464	0.7525	0.3009
256^2					
Ψ	-8.086×10^{-2}	2.581×10^{-4}	-8.539×10^{-2}	6.202×10^{-3}	6.250×10^{-3}
x	0.6734	0.7658	0.6947	0.1819	0.6877
y	0.8127	0.1713	0.7488	0.7505	0.3069
ϵ	0.0626	1.929	-	-	-
512^2					
Ψ	-	-	-8.543×10^{-2}	6.402×10^{-3}	6.271×10^{-3}
x	-	-	0.6938	0.1822	0.6868
y	-	-	0.7509	0.7515	0.3089
ϵ	-	-	0.0386	3.126	0.333

For use by other researchers some more information about the solution is given. Figure 3.21a shows the u^1 velocity profile along line CL_1 for $Re = 100$; Figure 3.21b gives the u^2 velocity profile along CL_2 . Figure 3.22a shows u^1 along line CL_1 for $Re = 1000$, Figure 3.22b presents u^2 along CL_2 . The velocity profiles predicted on the finest grid are given in tabulated form in Table 3.3 for $Re = 100$ and in Table 3.3 for $Re = 1000$, in which 14 reference points for reproduction of the profiles are given. At $(x^1, x^2) = (0.5, 0.5)$ a singular point occurs. For $Re = 1000$ it is inspected by giving, for all grids the pressure along the line $x^1 = 0.5$ and the pressure along the line $x^2 = 0.5$ in Figure 3.23. A large change in pressure, found to be similar on the three finest grids, can be observed at the singular point.

3.5 Conclusions.

A two-dimensional discretization has been presented for the steady incompressible Navier-Stokes equations in general coordinates. The discretization is coordinate invariant. The equations are discretized on a staggered grid. Christoffel symbols occur in the discretization chosen, which may cause inaccuracy on non-smooth grids; for example when a high Reynolds flow problem is solved on a grid with a kink in the grid lines. The convection tensor is discretized with a hybrid difference technique. The benchmark solutions show that our method gives a reliable discretization for the two-dimensional incompressible Navier-Stokes equations on sufficiently smooth general grids. The flow in an L-shaped cavity is proposed as an interesting benchmark problem, testing the effect of grid line curvature.

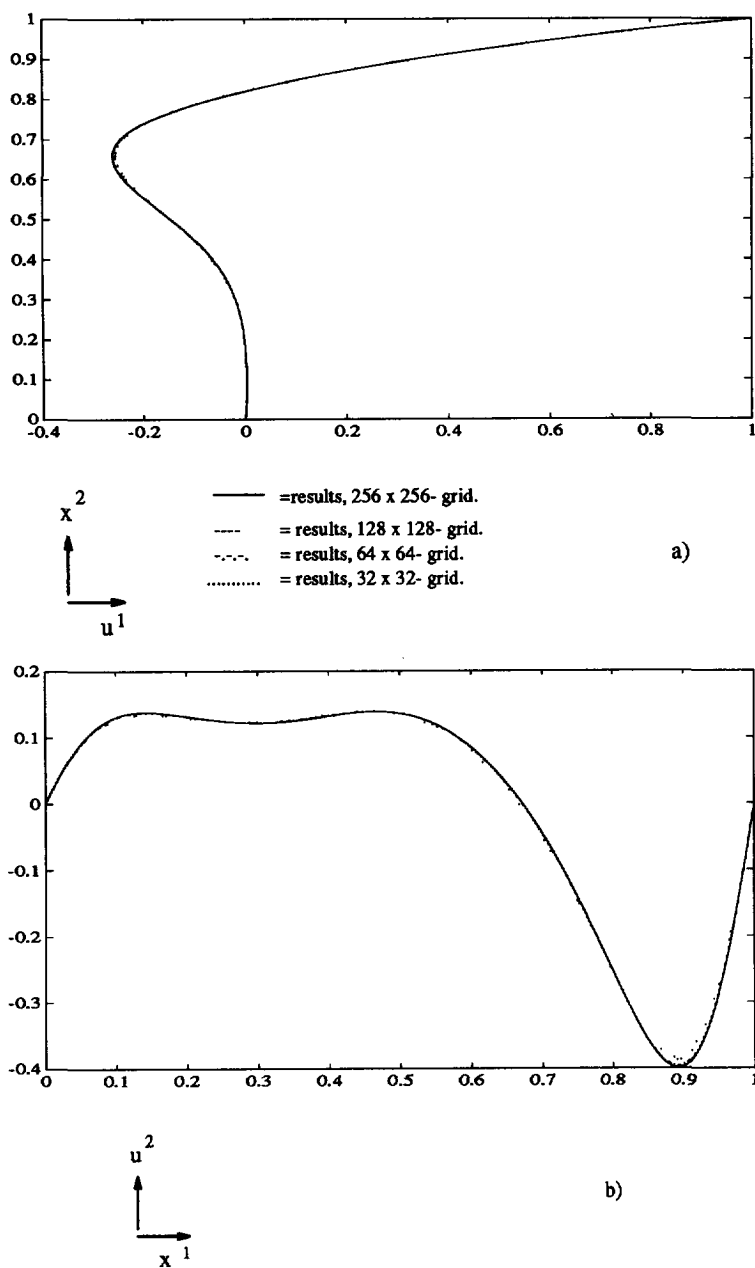


Figure 3.21: Velocity profiles for $Re = 100$ in the L-shaped cavity; a) u^1 along CL_1 ; b) u^2 along CL_2 .

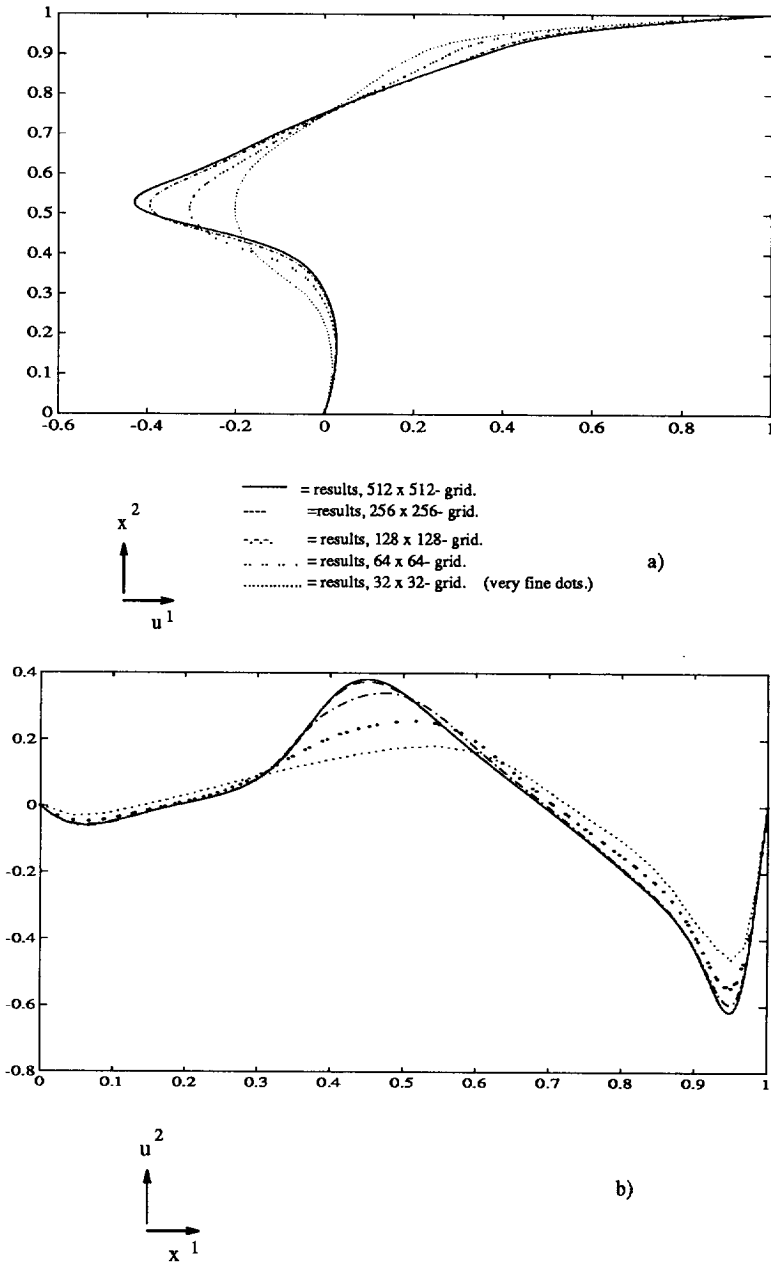


Figure 3.22: Velocity profiles for $Re = 1000$, L-shaped cavity; a) u^1 along CL_1 ; b) u^2 along CL_2 .

Table 3.3: Selected Cartesian components (u^1, u^2) along lines CL_1 and CL_2 for an L-shaped cavity, $Re = 100$ and $Re = 1000$.

L-shape, $Re = 100$			
x^1	u^1	x^2	u^2
1.391067×10^{-1}	1.427423×10^{-3}	4.999769×10^{-2}	9.035508×10^{-2}
3.249394×10^{-1}	-2.673955×10^{-2}	9.866030×10^{-2}	1.299809×10^{-1}
4.496036×10^{-1}	-9.856450×10^{-2}	1.403140×10^{-1}	1.379881×10^{-1}
5.024952×10^{-1}	-1.467907×10^{-1}	2.924033×10^{-1}	1.215604×10^{-1}
5.689500×10^{-1}	-2.121597×10^{-1}	4.691068×10^{-1}	1.387131×10^{-1}
6.387627×10^{-1}	-2.584130×10^{-1}	5.298618×10^{-1}	1.280727×10^{-1}
6.509587×10^{-1}	-2.609123×10^{-1}	6.011217×10^{-1}	8.317737×10^{-2}
6.582105×10^{-1}	-2.613650×10^{-1}	7.020829×10^{-1}	-4.919975×10^{-2}
6.776128×10^{-1}	-2.580779×10^{-1}	8.846614×10^{-1}	-3.980422×10^{-1}
7.103055×10^{-1}	-2.364301×10^{-1}	8.926683×10^{-1}	-3.995829×10^{-1}
8.010405×10^{-1}	-5.735737×10^{-2}	9.004891×10^{-1}	-3.975273×10^{-1}
8.441612×10^{-1}	8.892393×10^{-2}	9.252045×10^{-1}	-3.641550×10^{-1}
9.043458×10^{-1}	3.615964×10^{-1}	9.501776×10^{-1}	-2.843395×10^{-1}
9.524048×10^{-1}	6.458474×10^{-1}	9.841114×10^{-1}	-1.055517×10^{-1}

L-shape, $Re = 1000$			
x^1	u^1	x^2	u^2
1.008904×10^{-1}	2.214952×10^{-2}	1.012788×10^{-1}	-4.598057×10^{-2}
1.012788×10^{-1}	2.220298×10^{-2}	2.001027×10^{-1}	8.319043×10^{-3}
3.083983×10^{-1}	-3.958527×10^{-4}	3.002279×10^{-1}	8.151612×10^{-2}
3.559005×10^{-1}	-2.947157×10^{-2}	4.024018×10^{-1}	3.277335×10^{-1}
4.004040×10^{-1}	-8.499573×10^{-2}	4.496036×10^{-1}	3.822040×10^{-1}
4.790068×10^{-1}	-3.279580×10^{-1}	5.024953×10^{-1}	3.360547×10^{-1}
5.224505×10^{-1}	-4.280438×10^{-1}	5.505747×10^{-1}	2.495101×10^{-1}
5.295923×10^{-1}	-4.297859×10^{-1}	7.001215×10^{-1}	-1.130681×10^{-2}
6.003576×10^{-1}	-3.051670×10^{-1}	9.380253×10^{-1}	-6.011807×10^{-1}
6.549982×10^{-1}	-1.944255×10^{-1}	9.481664×10^{-1}	-6.196227×10^{-1}
7.116315×10^{-1}	-8.681319×10^{-2}	9.547680×10^{-1}	-6.092847×10^{-1}
8.233341×10^{-1}	1.566785×10^{-1}	9.701669×10^{-1}	-4.919449×10^{-1}
9.185696×10^{-1}	4.041500×10^{-1}	9.851177×10^{-1}	-2.647262×10^{-1}
9.740718×10^{-1}	7.053604×10^{-1}	9.950405×10^{-1}	-8.754424×10^{-2}

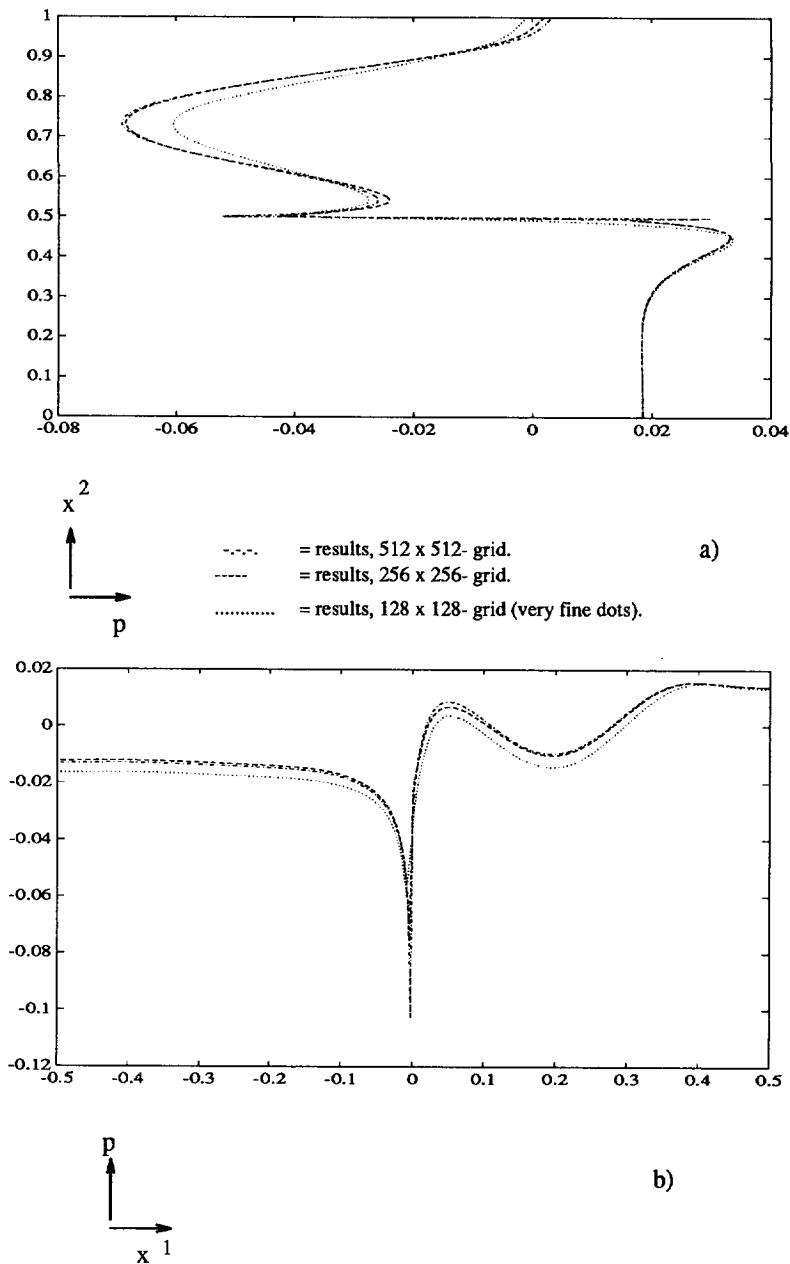
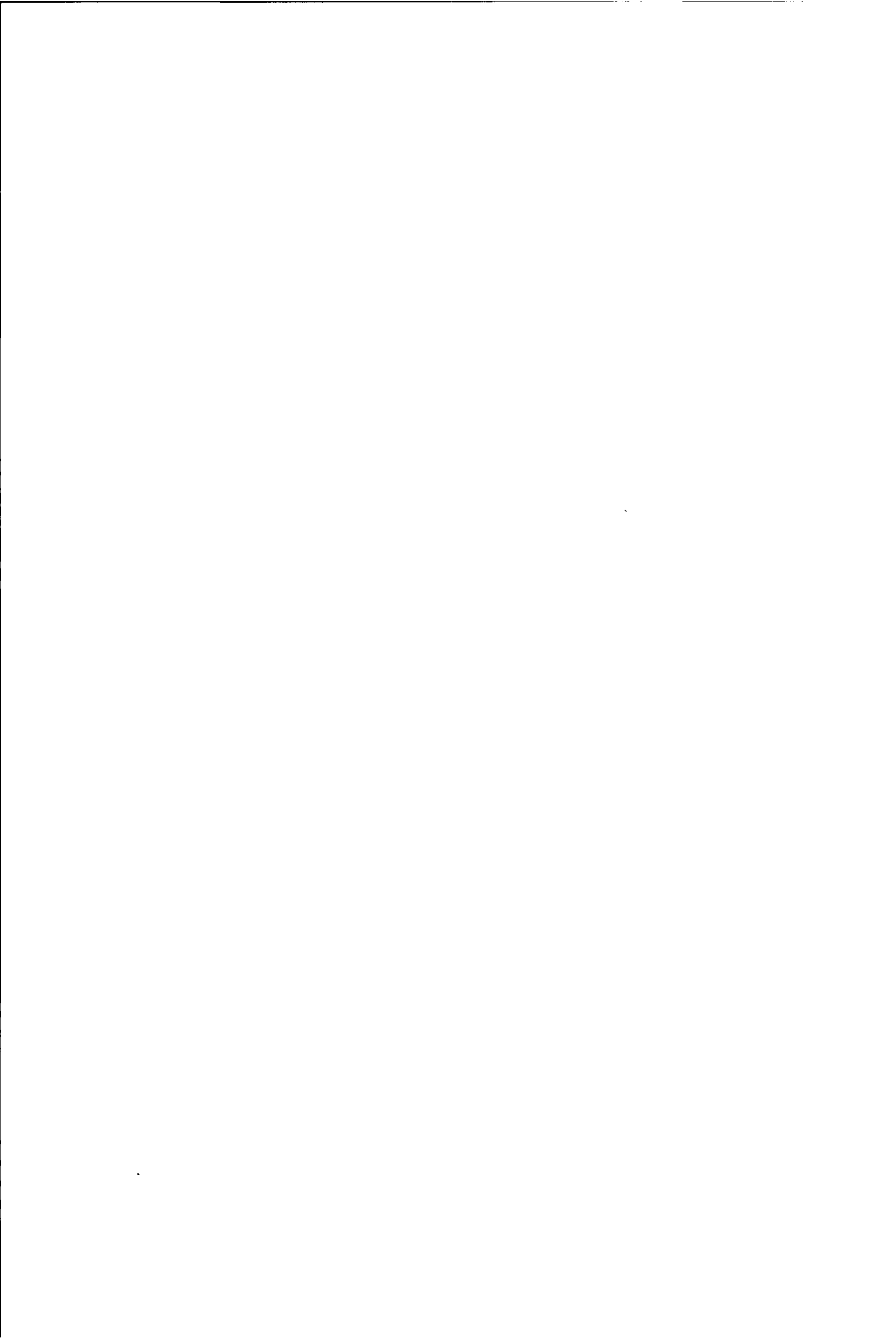


Figure 3.23: Pressure profiles along the line $x^1 = 0.5$ (a), and line $x^2 = 0.5$ (b), both through the singular point $(x^1, x^2) = (0.5, 0.5)$ for $Re = 1000$.



4 Multigrid schemes for steady incompressible Navier-Stokes equations in general coordinates.

4.1 Introduction.

Parts of this Chapter are also presented in [65], [66], [67], [68], and [69].

The finite volume discretization of the incompressible Navier-Stokes equations results in a large set of nonlinear algebraic equations. In order to solve very large two- and three-dimensional problems within reasonable time powerful solution methods are needed. Furthermore, it is interesting to look for vectorizable or parallelizable algorithms to use modern computer architectures as efficiently as possible.

Next to efficiency, robustness is an important aspect for solution algorithms; the solution in general domains for incompressible flows at high and low speed should be handled well by the solution method. In the early seventies the fastest solution methods known were Successive Overrelaxation (SOR) Methods. The relaxation parameters were problem-dependent and the convergence rates for large problems were much lower than for smaller problems. With the development of new computers with large relatively cheap storage facilities, there was a need for more efficient solution algorithms. On two fronts efficient methods were developed.

Krylov subspace methods. The Conjugate Gradient (CG) methods were given new life with the insight in [56] that preconditioning techniques can change the spectrum in a way that is favourable for fast convergence. This insight started a rapid development of the so-called Krylov subspace methods. In the eighties CG-Squared ([100]) was developed for solving large non-symmetrical problems. A relatively new variant called BICGSTAB ([103]) improved the robustness of the Conjugate Gradient methods. An other method in this class is the GMRES method, presented in [90]. This robust method is further developed in, amongst others [104]. Efficient implementations of GMRES on parallel machines are also considered, for example in [24]. An advantage of these methods is that linear systems are solved with a given matrix, which means that one does not need to have much knowledge of underlying discretizations. Furthermore, major parts of these algorithms are matrix-vector multiplications, which can be executed in vector speed on a vector machine. The largest programming effort required is the implementation of efficient preconditioners, like incomplete LU (ILU) techniques. A recent publication on Krylov subspace methods for nonlinear equations is [10]. An overview of existing iterative methods is presented in [32]. Problems with m unknowns can typically be solved in $O(m^\alpha)$ work, with α not much above one; in two-dimensional model situations, $\alpha = \frac{5}{4}$.

The multigrid principle. The second field in which fast solution methods are developed is the field of multigrid methods. The main idea of a multigrid solution technique is the acceleration of the convergence of basic iterative methods by using solutions obtained on coarser grids as additional corrections. It is a known fact

that basic iterative relaxation methods, like Jacobi or Gauss-Seidel methods are usually smoothing methods, which means that they damp out short wavelength components of the error of a solution efficiently. Then, when the error has become smooth convergence slows down; the remaining long wavelength errors are damped much more slowly. The insight that long wavelength errors act as shorter wavelength components on coarse grids and that these components are smoothed more efficiently on coarse grids gave rise to multigrid solution methods. After relaxation sweeps with a smoothing method on the finest grid a residual is transferred to a coarser grid, where relaxation sweeps are performed to smooth the corresponding error components. This procedure is repeated until the coarsest grid is reached, after which the coarse grid corrections are transferred back to finer grids, again with additional smoothing. There are several possibilities for cycling through the fine and coarse grids, called multigrid cycles. When coarse grids are visited more frequently this results in better coarse grid corrections. Typical for well-designed multigrid solution methods is a rapid convergence, with convergence rates bounded by a constant (much smaller than one), that does not depend on the mesh size. Problems with m unknowns can be solved in $O(m)$ operations.

Pioneers in multigrid methods were Brandt ([7]) and Hackbusch ([33]), who were able to produce efficient multigrid codes out of theoretical ideas, that already existed in the sixties ([28], [3]). Many papers on multigrid solution methods have been produced since. A recent overview of literature is given in [118], where the basic multigrid principles are introduced and results in computational fluid dynamics are presented. A mathematical description of the multigrid method can be found in [35]. The reader is referred to one of these books for more details on standard multigrid methods.

4.2 Multigrid methods for steady incompressible Navier-Stokes equations.

Nonlinear multigrid. A multigrid variant, called the nonlinear multigrid method ([35], [8]) has been constructed to solve nonlinear equations very efficiently. In this variant discretizations are adapted to new solutions during an iteration cycle. An advantage is that the number of iterations needed to solve nonlinear equations is much lower than when linearized equations are used ([36]). A disadvantage is that adaptation of discretizations to new solutions inhibits efficient use of vector facilities of computers.

In our research the nonlinear multigrid algorithm is used to solve the incompressible Navier-Stokes equations in general coordinates. The nonlinear two-grid algorithm is presented below:

Nonlinear two-grid algorithm: (to solve $T_h(\mathbf{u}) = \mathbf{f}_h$)	
begin	
• Apply ν_1 pre-smoothing iterations	
• Compute the fine grid residual:	
$\mathbf{r}_h = \mathbf{f}_h - T_h(\mathbf{u}_h)$	(4.1)
• Choose $\tilde{\mathbf{u}}_H$ (for example: $\tilde{R}^H \mathbf{u}_h$).	
• Apply Restriction R^H to the fine grid residual:	
$\mathbf{rhs}_H = R^H \mathbf{r}_h$	(4.2)
• Solve the coarse grid equation for \mathbf{u}_H .	
$T_H(\mathbf{u}_H) = \mathbf{f}_H = s_H \cdot \mathbf{rhs}_H + T_H(\tilde{\mathbf{u}}_H)$	(4.3)
• Apply Prolongation:	
$\mathbf{u}_h = \mathbf{u}_h + \frac{1}{s_H} P^h(\mathbf{u}_H - \tilde{\mathbf{u}}_H)$	(4.4)
• Apply ν_2 post-smoothing iterations	
end nonlinear two-grid algorithm.	

Here subscript h is used for fine grid operators and variables, subscript H for coarse grid operators and variables. The restriction (transferring from fine to coarse) operator for unknowns (\tilde{R}^H) does not need to be the same as the restriction operator for residuals (R^H). The nonlinear multigrid algorithm is drawn up in such a way that the prolongation operator transfers a *correction* ($\mathbf{u}_H - \tilde{\mathbf{u}}_H$) to a finer grid to improve a previously obtained approximation. The parameter s_H is used to keep the right-hand-side of the coarse grid equation in the range of the coarse grid operator ([35]). The coarse grid equation can be solved in a similar way, i.e. by introducing a third coarser grid and by executing a number of two-grid cycles, etcetera. The introduction of a sequence of grids leads to the nonlinear multigrid algorithm with various iteration cycles. Here the V-, F- and W-cycles (shown in Figure 4.1) are implemented.

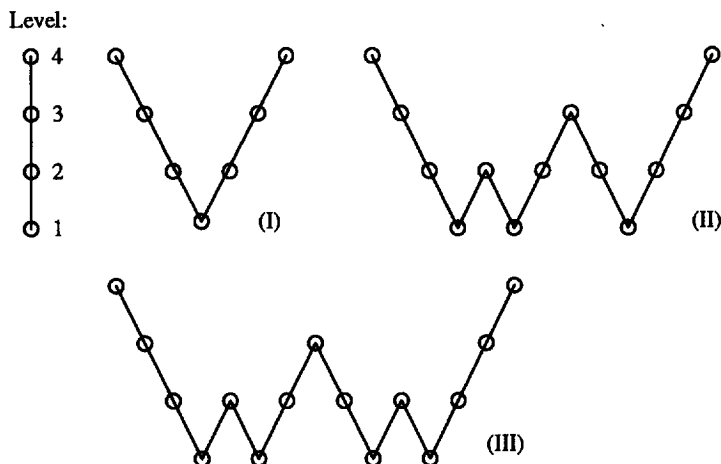
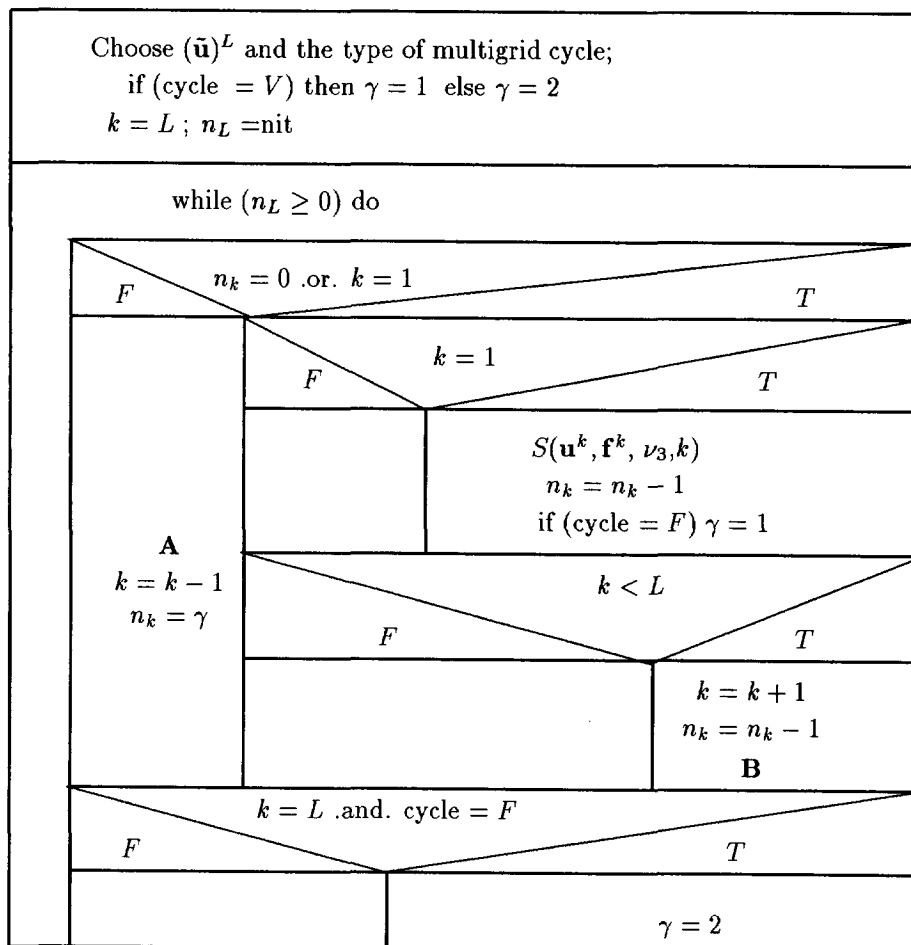


Figure 4.1: Three different iteration cycles visiting four grid levels: (I) the V-cycle, (II) the F-cycle, (III) the W-cycle.

The FAS algorithm ([8]) is a special case of the preceding nonlinear multigrid algorithm; a special choice for parameter s_H has been made ($s_H = 1$) and for $\tilde{\mathbf{u}}$ ($= \tilde{R}^H \mathbf{u}_h$). In [36] it is shown that for hard nonlinear problems one needs to allow more general choices for these quantities, but in our research $s_H = 1$ suffices. In a more extensive study of convergence rates, for example for flows at high Reynolds numbers, variation of this parameter could be included, but this is not done here. A non-recursive well-structured version of the nonlinear multigrid method, which is an improved version ([118]) of the algorithm described in [117] is used. The structure diagram of the algorithm including the V-, F-, and W-cycles is presented in Figure 4.2, for an arbitrary number of grid levels L . The superscript k indicates the grid level: $k = 1$ indicates the coarsest and $k = L$ the finest grid.



A : $S(\mathbf{u}^k, \mathbf{f}^k, \nu_1, k)$ $\mathbf{r}^k = \mathbf{f}^k - T^k(\mathbf{u}^k)$ Choose $\tilde{\mathbf{u}}^{k-1} (= \tilde{R}^{k-1} \mathbf{u}^k), s_{k-1}$ $\mathbf{u}^{k-1} = \tilde{\mathbf{u}}^{k-1}$ $\mathbf{f}^{k-1} = T^{k-1}(\tilde{\mathbf{u}}^{k-1}) + s_{k-1} R^{k-1} \mathbf{r}^k$
--

B : $\mathbf{u}^k = \mathbf{u}^k + \frac{1}{s_{k-1}} P^k(\mathbf{u}^{k-1} - \tilde{\mathbf{u}}^{k-1})$ $S(\mathbf{u}^k, \mathbf{f}^k, \nu_2, k)$

Figure 4.2: The structure diagram of the non-recursive multigrid algorithm including the V-, F- and W-cycle.

The algorithm carries out *nit* V -, F - or W -cycles, S is a smoothing subroutine. The number of pre-, post- and coarse grid smoothing iterations is indicated with ν_1 , ν_2 and ν_3 respectively.

In order to provide a reasonable starting solution on the finest grid the multigrid solution technique can be preceded by "nested iteration", which means that the solution process is started on the coarsest grid. The coarse grid solutions are prolonged to finer grids possibly with different multigrid cycles and prolongations. The more often the coarse grids are visited the better the starting approximation on the finest grid will resemble the final solution.

The different components of the nonlinear multigrid algorithm will be described now for the solution of the incompressible Navier-Stokes equations in general coordinates on a staggered grid. We now have a system of differential equations with $\mathbf{u} = (\mathbf{V}^1, \mathbf{V}^2, \mathbf{p})^T$.

Restriction. A reference paper, in which the incompressible Navier-Stokes equations are solved in Cartesian coordinates with a multigrid method is [107]. Restriction (fine-to-coarse) and prolongation (coarse-to-fine) transfer operators are described in detail in that paper. These operators are dictated by the staggered grid arrangement and the uniform computational domain G (Figure 4.3) that are chosen. More general domains are mapped on G by a boundary-fitted coordinate mapping.

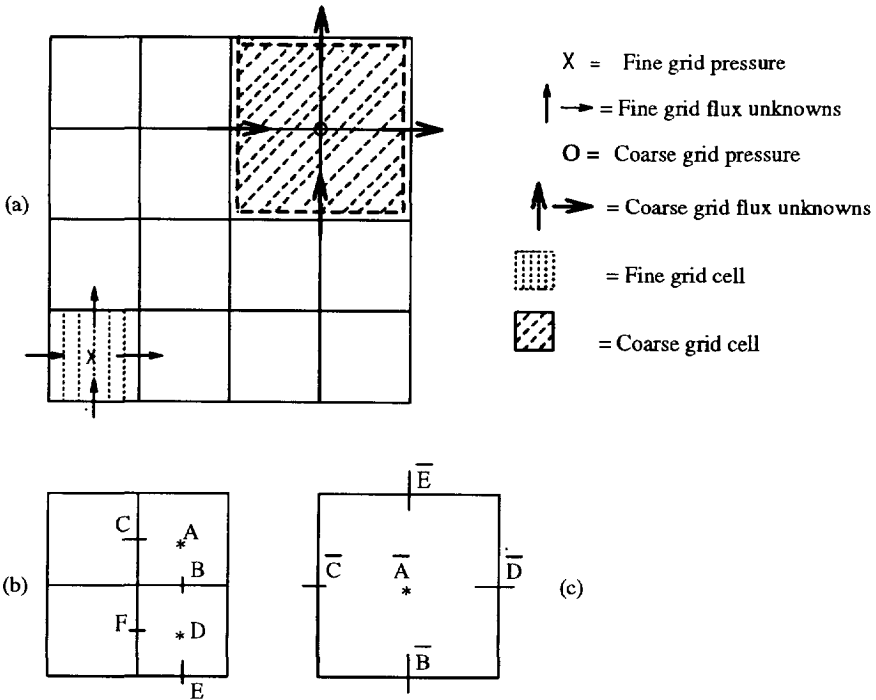


Figure 4.3: (a): A coarse and a fine grid cell in the computational domain G , with coarse and fine grid unknowns, (b): Definition of grid point numbering. The coarse grid cell shown is the union of the four fine grid cells shown. The grids are divided into so-called triads with identical gridpoint indices: $\bar{A} = \bar{B} = \bar{C} = (i, j)$. Furthermore, $\bar{D} = (i + 1, j)$, $\bar{E} = (i, j + 1)$. Two fine grid triads are $A = B = C = (2i, 2j)$, $D = E = F = (2i, 2j - 1)$.

The restricted fine grid approximate solution is only a starting approximation on the coarse grid, and therefore a simple restriction operator \tilde{R}^k is sufficient. The mean of neighbouring fine grid unknowns is taken to define coarse grid unknowns.

$$\begin{aligned} V_{i,j}^{1(k-1)} &= \frac{1}{2}(V_{2i-1,2j}^{1(k)} + V_{2i-1,2j-1}^{1(k)}) \\ V_{i,j}^{2(k-1)} &= \frac{1}{2}(V_{2i,2j-1}^{2(k)} + V_{2i-1,2j-1}^{2(k)}) \\ p_{i,j}^{(k-1)} &= \frac{1}{4}(p_{2i-1,2j}^{(k)} + p_{2i-1,2j-1}^{(k)} + p_{2i,2j}^{(k)} + p_{2i,2j-1}^{(k)}) \end{aligned} \quad (4.5)$$

For the evaluation of the coarse grid residual (4.2) area weighting for the fine grid residuals is used, as follows,

$$\begin{aligned} r_{i,j}^{1(k-1)} &= 1/8(r_{2i-2,2j}^{1(k)} + r_{2i-2,2j-1}^{1(k)} + r_{2i,2j}^{1(k)} + r_{2i,2j-1}^{1(k)}) + \\ &\quad 1/4(r_{2i-1,2j}^{1(k)} + r_{2i-1,2j-1}^{1(k)}) \\ r_{i,j}^{2(k-1)} &= 1/8(r_{2i,2j-2}^{2(k)} + r_{2i-1,2j-2}^{2(k)} + r_{2i,2j}^{2(k)} + r_{2i-1,2j}^{2(k)}) + \\ &\quad 1/4(r_{2i,2j-1}^{2(k)} + r_{2i-1,2j-1}^{2(k)}) \\ r_{i,j}^{3(k-1)} &= 1/4(r_{2i-1,2j}^{3(k)} + r_{2i-1,2j-1}^{3(k)} + r_{2i,2j}^{3(k)} + r_{2i,2j-1}^{3(k)}) \end{aligned} \quad (4.6)$$

Prolongation. Prolongation operators are applied to *corrections* in (4.4) using bilinear interpolation.

For V^1 the prolongation operator looks like:

$$V_{2i-1,2j+1}^{1(k)} = \frac{1}{4}(V_{i,j}^{1(k-1)} + 3V_{i,j+1}^{1(k-1)}) \quad (4.7)$$

$$V_{2i,2j+1}^{1(k)} = \frac{1}{8}(V_{i,j}^{1(k-1)} + 3V_{i,j+1}^{1(k-1)} + V_{i+1,j}^{1(k-1)} + 3V_{i+1,j+1}^{1(k-1)}) \quad (4.8)$$

$$V_{2i-1,2j}^{1(k)} = \frac{1}{4}(3V_{i,j}^{1(k-1)} + V_{i,j+1}^{1(k-1)}) \quad (4.9)$$

$$V_{2i,2j}^{1(k)} = \frac{1}{8}(3V_{i,j}^{1(k-1)} + V_{i,j+1}^{1(k-1)} + 3V_{i+1,j}^{1(k-1)} + V_{i+1,j+1}^{1(k-1)}) \quad (4.10)$$

At the boundaries all prolongation operators are modified to ensure that boundary conditions are not violated.

For V^2 the prolongation operator looks like:

$$V_{2i+1,2j-1}^{2(k)} = \frac{1}{4}(V_{i,j}^{2(k-1)} + 3V_{i+1,j}^{2(k-1)}) \quad (4.11)$$

$$V_{2i+1,2j}^{2(k)} = \frac{1}{8}(V_{i,j}^{2(k-1)} + V_{i,j+1}^{2(k-1)} + 3V_{i+1,j}^{2(k-1)} + 3V_{i+1,j+1}^{2(k-1)}) \quad (4.12)$$

$$V_{2i,2j-1}^{2(k)} = \frac{1}{4}(3V_{i,j}^{1(k-1)} + V_{i+1,j}^{1(k-1)}) \quad (4.13)$$

$$V_{2i,2j}^{2(k)} = \frac{1}{8}(3V_{i,j}^{2(k-1)} + 3V_{i,j+1}^{2(k-1)} + V_{i+1,j}^{2(k-1)} + V_{i+1,j+1}^{2(k-1)}) \quad (4.14)$$

For p the prolongation operator looks like:

$$p_{2i,2j}^{(k)} = \frac{1}{16}(9p_{i,j}^{(k-1)} + 3p_{i+1,j}^{(k-1)} + 3p_{i,j+1}^{(k-1)} + p_{i+1,j+1}^{(k-1)}) \quad (4.15)$$

$$p_{2i+1,2j}^{(k)} = \frac{1}{16}(3p_{i,j}^{(k-1)} + 9p_{i+1,j}^{(k-1)} + p_{i,j+1}^{(k-1)} + 3p_{i+1,j+1}^{(k-1)}) \quad (4.16)$$

$$p_{2i,2j+1}^{(k)} = \frac{1}{16}(3p_{i,j}^{(k-1)} + p_{i+1,j}^{(k-1)} + 9p_{i,j+1}^{(k-1)} + 3p_{i+1,j+1}^{(k-1)}) \quad (4.17)$$

$$p_{2i+1,2j+1}^{(k)} = \frac{1}{16}(p_{i,j}^{(k-1)} + 3p_{i+1,j}^{(k-1)} + 3p_{i,j+1}^{(k-1)} + 9p_{i+1,j+1}^{(k-1)}) \quad (4.18)$$

These relations between coarse and fine pressure corrections are also adapted at the boundaries to impose a zero normal derivative condition at the boundaries.

The coarse grid operator. Solving on coarse grids implies that a coarse grid operator must be evaluated. There are two methods to construct a coarse grid operator; by direct discretization of the partial differential equation on the coarser grid, called Discretization Coarse grid Approximation (DCA), or by Galerkin Coarse grid Approximation (GCA). The advantage of GCA is that after applying a matrix multiplication:

Restriction operator $R \times$ *matrix* $A_h \times$ *Prolongation operator* P (RA_hP ([118])) the coarse grid matrix A_H is computed without knowledge of the underlying discretization. However, this method is only applicable to linear(ized) problems. The method most suitable for nonlinear multigrid algorithms and used in this thesis is DCA. The nonlinear coarse grid operator is constructed with an intermediate coarse grid solution $\tilde{\mathbf{u}}$. On a coarse grid the even grid points from the fine grid are removed and the geometric quantities are calculated as was done on the finest grid. Therefore requirements for the geometric quantities derived in Section 2.3 are also satisfied on a coarse grid, which is not always the case when the coarse grid geometric quantities are derived with restriction of fine grid quantities. We will not go into this further here. The relation $d\xi^{\alpha(k-1)} = 2d\xi^{\alpha(k)}$ does not only hold for $d\xi^{\alpha}$ occurring in (2.71) and (2.89), but also for the derivation of the coarse grid geometric quantities.

The smoothing algorithm. The most important part of a multigrid method is the smoothing algorithm. Robustness depends in many cases primarily on the smoothing method, as does efficiency. A smoothing method is the most time consuming part of a multigrid algorithm. In other recently developed non-standard multigrid methods ([55], [59]) the coarse grid correction is made more robust, but these methods are not considered here.

The resulting operator for the incompressible Navier-Stokes equations in general coordinates is sparse (as can be seen in Figure 4.4), due to the fact that structured meshes are used and unknowns in a momentum equation are only connected to their neighbours. A discrete operator for momentum equations contains at most (when the grid is non-orthogonal) 19 unknowns, namely 13 velocity component terms and 6 pressure terms (see Figure 3.7).

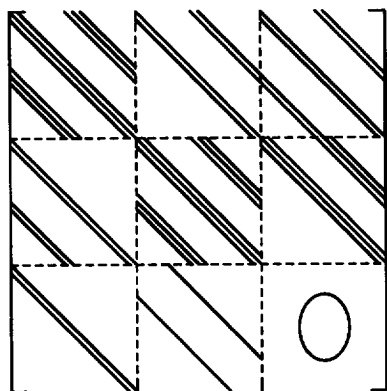


Figure 4.4: The matrix structure for the incompressible Navier-Stokes equations.

A major problem for the numerical solution of the incompressible Navier-Stokes equations is the absence of a pressure term in the continuity equation; the resulting linearized matrix A contains a zero diagonal block (as is also shown in Figure 4.4). There is no explicit equation for the pressure. This means that the matrix A of the linearized discrete system is not an M -matrix, and iterative methods based on a splitting $A = M - N$ according to

$$M\mathbf{u}^{n+1} = N\mathbf{u}^n + \mathbf{f}^{n+1} \quad (4.19)$$

will not converge in general.

Sometimes this problem is avoided by adopting instead of the so-called primitive formulation with velocity components and pressure as primary unknowns the vorticity-streamfunction formulation. This formulation is difficult to extend to three-dimensional problems and not used here. In order to solve steady incompressible Navier-Stokes equations in the so-called primitive formulation with velocity components and pressure as primary unknowns two different approaches have been developed. A first approach is called the uncoupled solution technique. For steady equations uncoupled solution techniques are of so-called distributive iteration type ([121], [123]).

In this approach the matrix A is post-conditioned by a matrix B , such that the

zero block disappears; the situation is too complex to guarantee that AB will be an M -matrix. The resulting system can now be split as follows:

$$AB = M - N \quad (4.20)$$

This gives rise to the distributed iteration methods:

$$MB^{-1}\mathbf{u}^{n+1} = NB^{-1}\mathbf{u}^n + \mathbf{f} \quad (4.21)$$

In the context of incompressible Navier-Stokes three methods have been used as smoothing methods in a multigrid algorithm ([122], [123]). With

$$B = \begin{bmatrix} I & 0 & -(S^{11})^{-1}A^{13} \\ 0 & I & -(S^{22})^{-1}A^{23} \\ 0 & 0 & I \end{bmatrix} \quad (4.22)$$

where $(S^{ii})^{-1}$ is an easy to evaluate approximation to $(A^{ii})^{-1}$, we find the SIMPLE-type algorithms [72]. They are used as smoother for incompressible Navier-Stokes equations in Cartesian coordinates in [93], and recently in arbitrarily shaped domains in [81]. Difficulties with SIMPLE are found when the Reynolds number is high, due to the diagonal term in matrix AB : $-(A^{31}(S^{11})^{-1}A^{13} + A^{32}(S^{22})^{-1}A^{23})$, which approaches zero for $Re \rightarrow \infty$ ([123]). Therefore underrelaxation parameters are introduced depending on the shape of the domain and the Reynolds number, but robustness is impaired.

With

$$B = \begin{bmatrix} I & 0 & A^{13} \\ 0 & I & A^{23} \\ 0 & 0 & A^{31}A^{13} + A^{32}A^{23} \end{bmatrix} \quad (4.23)$$

the DGS (Distributive Gauss-Seidel) method [9] is obtained.

Difficulties arise because of the off-diagonal terms in AB : $A^{11}A^{13} + A^{12}A^{23} + A^{13}(A^{31}A^{13} + A^{32}A^{23})$ and $A^{21}A^{13} + A^{22}A^{23} + A^{23}(A^{31}A^{13} + A^{32}A^{23})$, which can be dominant over the diagonal term, for example for certain values of the Reynolds number. Difficulties with DGS are found for several flow problems. For driven cavity problems the multigrid method did not converge for $Re > 100$ ([29]). Distributive ILU smoothing on matrix AB from (4.23) is performed in [123], with

$$AB = LU - N \quad (4.24)$$

where LU is an incomplete LU factorization of AB . This smoother is robust and fairly efficient, and is investigated for a discretization of the unsteady incompressible Navier-Stokes equations in general coordinates in a pressure correction method in [126].

4.2.1 Results with SCGS.

Another class of smoothing methods, not of distributive type, is called the coupled solution method; the discretized momentum and continuity equations are solved simultaneously. A coupled cell-by-cell solution technique is introduced in [108] and [109], as smoothing method in a nonlinear multigrid method for Cartesian incompressible Navier-Stokes equations. In this smoother, called the Symmetric Coupled Gauss-Seidel method (SCGS) a system of coupled equations per cell ($2 * V^1$ -momentum, $2 * V^2$ -momentum and continuity) is updated cell-by-cell, as follows.

Suppose the residual \mathbf{r} before a SCGS-update is given by

$$\begin{pmatrix} \mathbf{r}^1 \\ \mathbf{r}^2 \\ \mathbf{r}^3 \end{pmatrix} = \begin{pmatrix} \mathbf{f}^1 \\ \mathbf{f}^2 \\ 0 \end{pmatrix} - \begin{pmatrix} A^{11} & A^{12} & A^{13} \\ A^{21} & A^{22} & A^{23} \\ A^{31} & A^{32} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}^{1(n)} \\ \mathbf{V}^{2(n)} \\ \mathbf{p}^{(n)} \end{pmatrix} \quad (4.25)$$

where $(\mathbf{V}^{1(n)}, \mathbf{V}^{2(n)}, \mathbf{p}^{(n)})^T$ represents the current solution, and where the operators $A^{\alpha\beta}$ may depend on the solution.

Corrections $[\delta\mathbf{V}^1, \delta\mathbf{V}^2, \delta\mathbf{p}]^T$ are calculated and added to the current solution. So:

$$\begin{pmatrix} A^{11} & A^{12} & A^{13} \\ A^{21} & A^{22} & A^{23} \\ A^{31} & A^{32} & 0 \end{pmatrix} \begin{pmatrix} \delta\mathbf{V}^1 \\ \delta\mathbf{V}^2 \\ \delta\mathbf{p} \end{pmatrix} = \begin{pmatrix} \mathbf{r}^1 \\ \mathbf{r}^2 \\ \mathbf{r}^3 \end{pmatrix} \quad (4.26)$$

System (4.26) is solved with a collective Gauss-Seidel type smoother. For cell (i, j) the following system is solved:

$$\begin{pmatrix} (c_1^1)_{i-1/2,j} & & & & (c_{17}^1)_{i-1/2,j} \\ & (c_1^1)_{i+1/2,j} & & & (c_{16}^1)_{i+1/2,j} \\ & & (c_1^2)_{i,j-1/2} & & (c_{17}^2)_{i,j-1/2} \\ & & & (c_1^2)_{i,j+1/2} & (c_{16}^2)_{i,j+1/2} \\ -\frac{1}{\delta\xi^1} & \frac{1}{\delta\xi^1} & -\frac{1}{\delta\xi^2} & \frac{1}{\delta\xi^2} & 0 \end{pmatrix} \begin{pmatrix} \delta V_{i-1/2,j}^1 \\ \delta V_{i+1/2,j}^1 \\ \delta V_{i,j-1/2}^2 \\ \delta V_{i,j+1/2}^2 \\ \delta p_{ij} \end{pmatrix} = \begin{pmatrix} r_{i-1/2,j}^1 \\ r_{i+1/2,j}^1 \\ r_{i,j-1/2}^2 \\ r_{i,j+1/2}^2 \\ r_{ij}^3 \end{pmatrix} \quad (4.27)$$

where the c -coefficients are extracted from $A^{\alpha\beta}$ in (4.26) in a local linearization procedure, and where, for example:

$$\begin{aligned}
 r_{i+1/2,j}^1 = & c_2^1 V_{i-1/2,j-1}^1 + c_3^1 V_{i+1/2,j-1}^1 + c_4^1 V_{i+3/2,j-1}^1 + \\
 & c_5^1 V_{i-1/2,j}^1 + c_1^1 V_{i+1/2,j}^1 + c_6^1 V_{i+3/2,j}^1 + \\
 & c_7^1 V_{i-1/2,j+1}^1 + c_8^1 V_{i+1/2,j+1}^1 + c_9^1 V_{i+3/2,j+1}^1 + \\
 & c_{10}^1 V_{i,j-1/2}^2 + c_{11}^1 V_{i+1,j-1/2}^2 + \\
 & c_{12}^1 V_{i,j+1/2}^2 + c_{13}^1 V_{i+1,j+1/2}^2 + \\
 & c_{14}^1 p_{i,j-1} + c_{15}^1 p_{i+1,j-1} + c_{16}^1 p_{i,j} + \\
 & c_{17}^1 p_{i+1,j} + c_{18}^1 p_{i,j+1} + c_{19}^1 p_{i+1,j+1}
 \end{aligned} \tag{4.28}$$

The coefficient superscripts 1 and 2 refer to the two momentum equations. The subscripts are based on the numbering in the stencil (see Figure 3.7):

Equations (4.27) and (4.28) generalize in straightforward way from the Cartesian case to general coordinates.

Underrelaxation which has the effect of a false time-dependent term and possesses stabilizing characteristics is implemented as in [109]:

$$\begin{aligned}
 (c_1^1)_{i\pm 1/2,j} & := (c_1^1)_{i\pm 1/2,j} / \alpha_1 \\
 (c_1^2)_{ij\pm 1/2} & := (c_1^2)_{ij\pm 1/2} / \alpha_2
 \end{aligned} \tag{4.29}$$

As in [86] and [109] different underrelaxation parameters α_k are needed in SCGS for different Reynolds numbers. Here good values are found to be:

$$\begin{aligned}
 Re < 400 : & \quad \alpha_k = 0.7 \\
 400 \leq Re < 1000 : & \quad \alpha_k = 0.4 \\
 Re = 1000 : & \quad \alpha_k = 0.3 \quad (k = 1, 2).
 \end{aligned} \tag{4.30}$$

Equations (4.27) are solved with an explicit formula, after which the corrections are added to the current solution, the residual and the nonlinear discretization operator are updated, and the computation continues with the next cell. All velocity unknowns are updated twice during a relaxation sweep, while the pressures are updated once. The singularity in the system of equations, due to the incompressibility condition, had to be overcome in SCGS by fixing a pressure value in a corner of the coarsest grid (like in [109]).

Cell orderings. There are several possibilities for visiting the grid cells, based on the ordering of cells. Some well-known orderings are shown in Figure 4.5. An overview of orderings together with a survey on robustness of basic iterative methods for the anisotropic diffusion equation and the rotated convection-diffusion equation is presented in [118]. A theoretical survey on robustness for the incompressible

Navier-Stokes equations is difficult. Some results of Fourier analysis for SCGS-based multigrid methods and Cartesian equations are presented in [86], for uncoupled solution techniques theoretical results are presented in [121].

For low Reynolds numbers (< 400) satisfactory convergence results are obtained with SCGS using the forward lexicographical ordering (Figure 4.5a), for high Reynolds numbers it is necessary to move in alternating directions: the forward lexicographical sweep is followed by a forward sweep along vertical lines (Figure 4.5b), as in [86]. A combination of forward and backward sweeps would lead to a more robust (but also more expensive) smoothing method; this is not done here.

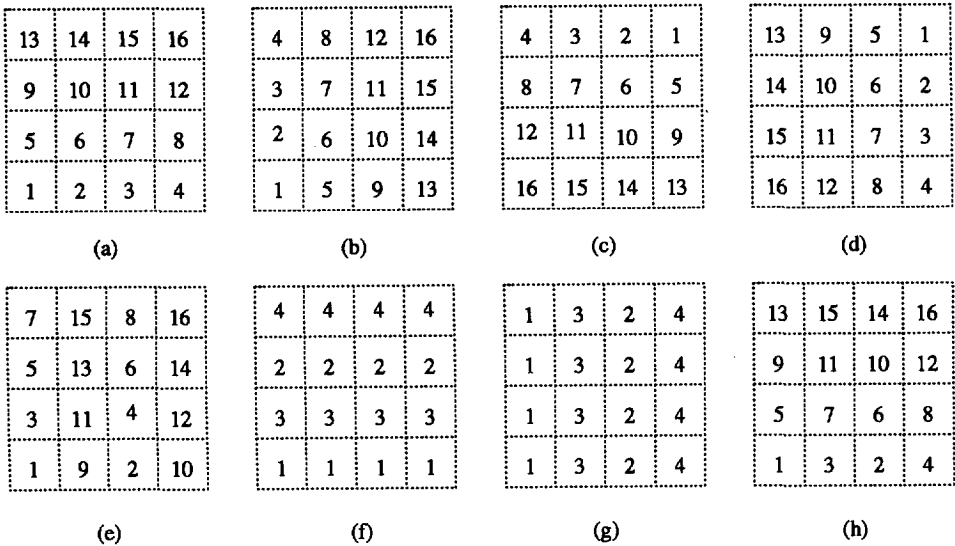


Figure 4.5: Grid cell orderings for SCGS-type methods, (a) forward lexicographical ordering, (b) vertical forward ordering, (c) backward ordering, (d) vertical backward ordering, (e) white-black ordering, (f) horizontal line ordering, (g) vertical line ordering, (h) white-black horizontal line ordering.

SCGS performs very well for several problems in square domains with equidistant grids [107], [108], [109], [110], [111], [124]. In [111] a forward white-black line SCGS smoother (Figure 4.5h) is investigated to improve the vectorizability of the SCGS method.

In a multigrid code based on SCGS it is not necessary to store a whole matrix. The operator elements for a cell can be constructed, when the cell is visited. This saves a lot of computer storage. Furthermore, it is possible to update operator elements during a relaxation sweep. This is the procedure followed here.

SCGS results. In order to compare the convergence history of the multigrid meth-

ods for several test problems average reduction factors r_{nit} are compared, defined as

$$r_{nit} = \left(\frac{\|res_{nit}\|}{\|res_0\|} \right)^{\frac{1}{nit}} \quad (4.31)$$

i.e. the l_2 -norm of the residual after nit iterations divided by the l_2 -norm of the starting residual.

The first test problem investigated is the driven cavity flow problem. We compared r_{nit} for a skewed cavity (i.e. a parallelogram, skewing angle 66°) and a unit rectangular cavity for several Reynolds numbers (Re). The starting vector is the zero-solution on the finest grid. Here $nit = 20$; the number of pre-smoothing iterations ν_1 is 1, the number of post-smoothing iterations ν_2 is 1, the number of coarse grid relaxation iterations ν_3 is 10. The results given are reduction factors for the W-cycle. Other iteration cycles will be compared and used at a later stage with another smoother. Table 4.1 presents the reduction factors. In this Table an unusual improvement of r_{20} is observed, when the grid is refined in the square cavity at $Re = 1000$. This improvement is probably due to the fact that the central difference scheme will be applied in large parts of the domain when the grid is very fine. This can improve the convergence rate. It looks as if r_{20} is bounded well away

Table 4.1: Average reduction factors r_{20} for driven cavity problems calculated with the W-cycle, with the SCGS smoother, $\nu_1 = \nu_2 = 1$.

grid	# levels	$\alpha_k = 0.7$		$\alpha_k = 0.4$		$\alpha_k = 0.3$	
		$Re = 100$		$Re = 400$		$Re = 1000$	
		r_{skewed}	r_{square}	r_{skewed}	r_{square}	r_{skewed}	r_{square}
16×16	4	.390	.328	.502	.454	.508	.516
32×32	5	.353	.315	.445	.446	.619	.565
64×64	6	.345	.310	.463	.398	.646	.564
128×128	7	.338	.313	.310	.310	.710	.517

from 1 independent of the number of levels, except for the flow in the skewed cavity at $Re = 1000$. This effect is absent when the cavity is closer to a square; with a skew angle of 79° $r_{20} = 0.526$ on a 128×128 grid. The slower convergence for the skewed grids is due to the central-upwind switch in the hybrid difference scheme for the convection terms, as will be seen later when the multigrid results for the benchmark problems presented in Section 3.4 are shown.

The following test problems further test the multigrid method for low Reynolds numbers. A low Reynolds flow ($Re = 1$) through a straight pipe (equidistant grid, square cells), an L-shaped pipe and a nozzle flow are considered. Parabolic velocity

profiles are prescribed at in- and outflow. This is unphysical of course, but of no concern here. The average reduction factors r_{20} are compared for these geometries for several grids. Table 4.2 shows r_{20} for the W-cycle. The reduction factors for the other cycles did not differ much. The parameters for the MG-algorithm are: $\nu_1 = 1$, $\nu_2 = 1$, $\nu_3 = 10$, lexicographical forward ordering. Figures 4.6 and 4.7 show

Table 4.2: Channel flows at $Re = 1$, lexicographical SCGS, $\nu_1 = \nu_2 = 1$, W-cycle.

grid	levels	r_{20} (pipe)	r_{20} (L-shape)	r_{20} (nozzle)
4×10	2	.254	.376	.329
8×20	3	.281	.385	.376
16×40	4	.301	.437	.468

grids, streamlines and isobars for the L-shape and nozzle geometry. It seems that for the nozzle reduction factors are not level-independent, but Figure 4.7 shows that the 16×40 grid is much more non-uniform than the 8×20 grid, which may be of greater consequence for r_{20} than the number of levels.

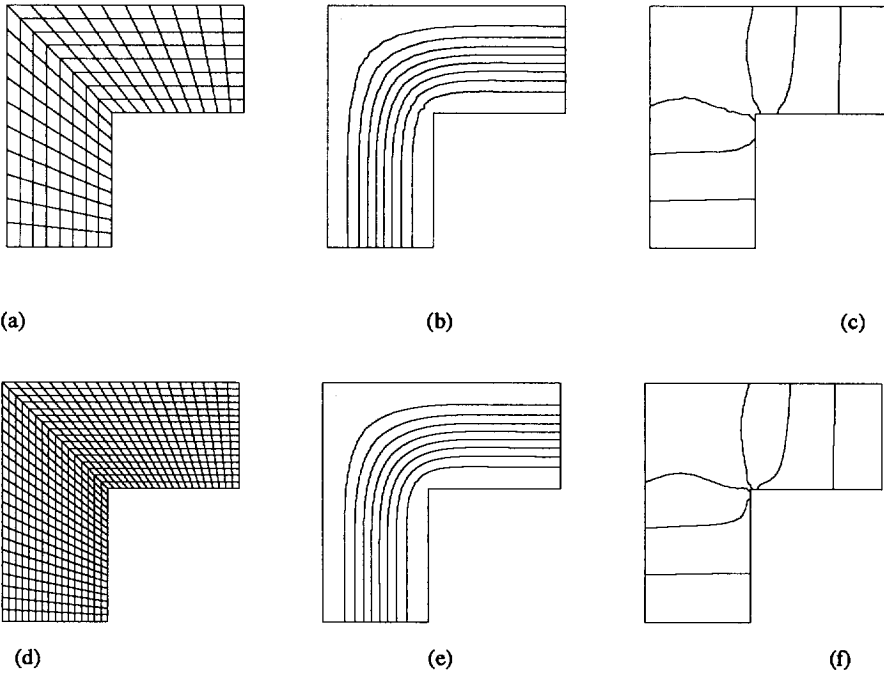
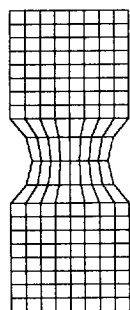
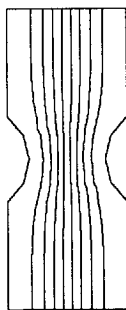


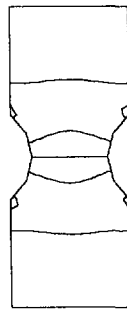
Figure 4.6: A low Reynolds flow through an L-shaped channel, (a) the mesh (8×20), (b) streamlines, (c) isobars, (d) the mesh (16×40), (e) streamlines, (f) isobars .



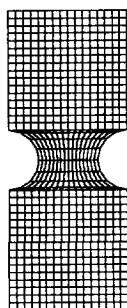
(a)



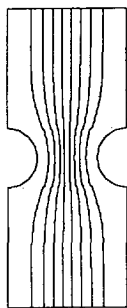
(b)



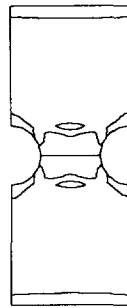
(c)



(d)



(e)



(f)

Figure 4.7: A low Reynolds flow through a nozzle, (a) the mesh (8×20), (b) streamlines, (c) isobars, (d) the mesh (16×40), (e) streamlines, (f) isobars

The average reduction factors for different geometries show that the multigrid approach used generalizes in satisfactory manner from Cartesian coordinates to curvilinear coordinates. The convergence of the nonlinear multigrid method for rectangular and some more complex geometries does not differ much. Similar results are presented in [47]. However, it is a well-known fact that point Gauss-Seidel-type relaxation methods are not good smoothing methods in multigrid algorithms when the equations involve strong coupling of unknowns in some direction, which is the case when stretched cells occur. When the cell aspect ratio is high SCGS smoothing factors tend to one. A more robust smoother is needed for efficient solution of flow problems in general domains, where due to numerical grid generation procedures and also for accuracy reasons highly stretched cells are not unusual. Furthermore, it was found that the cell-by-cell calculation of intricate operator elements for general coordinates is time consuming for fine grids. The use of intermediate solutions in the convection terms also proved to be less robust than the use of a stored matrix during a relaxation sweep. The code is then less sensitive to variations in relaxation parameters for higher Reynolds numbers, while similar convergence results were obtained. The code is especially robust for low (< 400) Reynolds numbers; variations in underrelaxation parameters are allowed.

More SCGS results are presented after the discussion of two other smoothers. Then, the three smoothers will be compared.

SCGS/LS. Another SCGS-type smoother that is investigated for robustness is called SCGS/LS (Line Solver), and is presented in [86] and [92]. In this line version of SCGS only the pressures along a line are coupled. Instead of (4.27) the set is rewritten for a coupled solution of all pressures along a ξ^1 -line in terms of corrections, as follows:

$$\begin{aligned}
 (c_1^1)_{i-1/2,j} \delta V_{i-1/2,j}^1 - (c_{16}^1)_{i-1/2,j} \delta p_{i-1,j} - (c_{17}^1)_{i-1/2,j} \delta p_{i,j} &= r_{i-1/2,j}^1 \\
 (c_1^1)_{i+1/2,j} \delta V_{i+1/2,j}^1 - (c_{16}^1)_{i+1/2,j} \delta p_{i,j} - (c_{17}^1)_{i+1/2,j} \delta p_{i+1,j} &= r_{i+1/2,j}^1 \\
 (c_1^2)_{i,j-1/2} \delta V_{i,j-1/2}^2 - (c_{17}^2)_{i,j-1/2} \delta p_{i,j} &= r_{i,j-1/2}^2 \\
 (c_1^2)_{i,j+1/2} \delta V_{i,j+1/2}^2 - (c_{16}^2)_{i,j+1/2} \delta p_{i,j} &= r_{i,j+1/2}^2 \\
 (\delta V_{i+1/2,j}^1 - \delta V_{i-1/2,j}^1) / \delta \xi^1 + (\delta V_{i,j+1/2}^2 - \delta V_{i,j-1/2}^2) / \delta \xi^2 &= r_{i,j}^3 \quad (4.32)
 \end{aligned}$$

(The equations for solving along a ξ^2 -line are found in a similar way).

Solving the set of equations (4.32) leaves us a tri-diagonal system for pressures along the line:

$$(P_w)_{i-1,j} \delta p_{i-1,j} + (P_c)_{i,j} \delta p_{i,j} + (P_e)_{i+1,j} \delta p_{i+1,j} = r_{i,j}^p \quad (4.33)$$

where,

$$(P_w)_{i-1,j} = \left(-\frac{c_{16}^1}{c_1^1} \right)_{i-1/2,j} \cdot \frac{1}{\delta \xi^1}$$

$$\begin{aligned}
(P_e)_{i+1,j} &= \left(\frac{c_{17}^1}{c_1^1}\right)_{i+1/2,j} \cdot \frac{1}{\delta\xi^1} \\
(P_c)_{i,j} &= \left(-\frac{c_{17}^1}{c_1^1}\right)_{i-1/2,j} \cdot \frac{1}{\delta\xi^1} + \left(\frac{c_{16}^1}{c_1^1}\right)_{i+1/2,j} \cdot \frac{1}{\delta\xi^1} - \left(\frac{c_{17}^2}{c_1^2}\right)_{i,j-1/2} \cdot \frac{1}{\delta\xi^2} + \\
&\quad \left(\frac{c_{16}^2}{c_1^2}\right)_{i,j+1/2} \cdot \frac{1}{\delta\xi^2} \\
r_{i,j}^p &= r_{i,j}^3 + \left(\frac{r^1}{c_1^1}\right)_{i-1/2,j} \cdot \frac{1}{\delta\xi^1} - \left(\frac{r^1}{c_1^1}\right)_{i+1/2,j} \cdot \frac{1}{\delta\xi^1} + \left(\frac{r^2}{c_1^2}\right)_{i,j-1/2} \cdot \frac{1}{\delta\xi^2} - \\
&\quad \left(\frac{r^2}{c_1^2}\right)_{i,j+1/2} \cdot \frac{1}{\delta\xi^2} \tag{4.34}
\end{aligned}$$

After solving the tridiagonal system the corrections for p are added to the current solution.

The corrections for V^1 and V^2 are found from (4.32):

$$\begin{aligned}
\delta V_{i-1/2,j}^1 &= [r_{i-1/2,j}^1 + (c_{16}^1)_{i-1/2,j} \delta p_{i-1,j} + (c_{17}^1)_{i-1/2,j} \delta p_{i,j}] / (c_1^1)_{i-1/2,j} \\
\delta V_{i+1/2,j}^1 &= [r_{i+1/2,j}^1 + (c_{16}^1)_{i+1/2,j} \delta p_{ij} + (c_{17}^1)_{i+1/2,j} \delta p_{i+1,j}] / (c_1^1)_{i+1/2,j} \\
\delta V_{i,j-1/2}^2 &= [r_{i,j-1/2}^2 + (c_{17}^2)_{i,j-1/2} \delta p_{i,j}] / (c_1^2)_{i,j-1/2} \\
\delta V_{i,j+1/2}^2 &= [r_{i,j+1/2}^2 + (c_{16}^2)_{i,j+1/2} \delta p_{ij}] / (c_1^2)_{i,j+1/2} \tag{4.35}
\end{aligned}$$

The velocities are updated twice, pressures are updated once. Underrelaxation is implemented as in (4.29). For SCGS/LS the optimal α_k 's are the same as for SCGS. Several test problems in general domains have been calculated with SCGS/LS. Some results will be presented after the explanation of the next smoother, when three coupled smoothing methods are compared.

However, in arbitrarily shaped domains SCGS/LS, as SCGS, is found not to be a robust smoother; when the cell aspect ratio is much larger than one convergence factors tend to one.

All unknowns in a line of cells must be updated simultaneously. An attempt by the author of this thesis to construct a two-stage smoother was not successful. In stage one the two momentum equations were updated with an alternating damped line Jacobi method (ADLJ). This is a robust method with damping parameter 0.7 for the rotated convection diffusion equation and for the anisotropic diffusion equation ([118]). In stage two the continuity was taken into account and the pressures were updated with SCGS or SCGS/LS. However, the underrelaxation parameters were problem dependent and large aspect ratios could still not be handled.

4.2.2 Results with the SCAL smoother.

A fully coupled line version of SCGS is presented in [101] for the Cartesian incompressible Navier-Stokes equations, and is called Symmetric Coupled Alternating Lines (SCAL). All unknowns on a line of cells are updated simultaneously with alternating zebra-sweeping. The ordering is odd horizontal lines, even horizontal lines, odd vertical lines, even vertical lines, like Figures 4.5f and 4.5g. If for a sweep along horizontal lines the unknowns per line are ordered in the following way: $V_{i,j+1}^2, p_{ij}, V_{i,j}^2, V_{i+1,j}^1, V_{i+1,j+1}^2$ etc., the resulting matrix is a band matrix. The band contains 13 elements for the equations in curvilinear coordinates (in Cartesian coordinates it is 9). This band matrix is solved with a direct band solver from the *Linpack*-library, and appeared to be the most expensive part of the multigrid algorithm. The same solver from the *Lapack*-library was not faster, neither was a GMRES-based iterative solver. The cost of using the line smoother is about 50% larger than SCGS ([101]). In [101] it is also reported that SCAL is much more robust than SCGS and less sensitive to variations of the underrelaxation parameters. Therefore we implemented SCAL as smoother in our multigrid algorithm. It is interesting to investigate the behaviour of this smoother for problems in general domains.

Underrelaxation is implemented in the "classical" way. Instead of calculating corrections $\delta \mathbf{V}$ new values \mathbf{V}^* are being calculated and $\mathbf{V}^{(n+1)}$ is found as follows:

$$\begin{aligned} \mathbf{V}^{1(n+1)} &= \mathbf{V}^{1(n)} + \alpha_1(\mathbf{V}^{1*} - \mathbf{V}^{1(n)}) \\ \mathbf{V}^{2(n+1)} &= \mathbf{V}^{2(n)} + \alpha_2(\mathbf{V}^{2*} - \mathbf{V}^{2(n)}) \\ \mathbf{p}^{(n+1)} &= \mathbf{p}^{(n)} + \alpha_3(\mathbf{p}^* - \mathbf{p}^{(n)}) \end{aligned} \quad (4.36)$$

The underrelaxation parameters α_k depend on the Reynolds number.

For $Re \leq 150$, $\alpha_k = 0.7$ for $k = 1, 2, 3$ seems to be good and gives level-independent reduction factors.

For $Re = 1000$, $\alpha_k = 0.4$ for $k = 1, 2$ and $\alpha_3 = 1.0$.

With parallelization statements or special ordering strategies for vectorization purposes the zebra smoother may be faster on parallel or vector computers. These strategies are not implemented in the code.

It appeared not to be necessary to fix a pressure on the coarsest grid for SCAL; the singularity in the equations disappears in a natural way.

There are some advantages for the calculation and storage of a complete momentum matrix. It is already stated that the calculation of operator elements depending on intermediate solutions is costly and sensitive. Computer code for the discretization of the momentum equations in general coordinates is extensive and elaborate, especially in three dimensions. The code for other boundary conditions or changes in the discretization are much easier implemented when a matrix is stored. Furthermore, a flexible Navier-Stokes code generator has been developed ([77]) that produces a discretization matrix with for example different scaling of unknowns

or with other interpolation rules fast. It is necessary and interesting that future changes in the discretization can be implemented within some days. We will not go into details here.

Three smoothers compared. In order to compare the behaviour of SCGS, SCGS/LS and SCAL in the nonlinear multigrid method some driven cavity problems with more or less regular cell distributions are solved. The first problem is the classical problem: an equidistant grid in a square cavity. Then: a non-equidistant grid in a square cavity, in which the boundary cells have aspect ratio 4, and a non-equidistant grid in an L-shaped cavity. Figure 4.8 shows the last two grids. It should be stated that for SCGS and SCGS/LS these are more or less limiting problems. Grids with more stretching in the cells, or sharper L-shapes are not handled well by these smoothers.

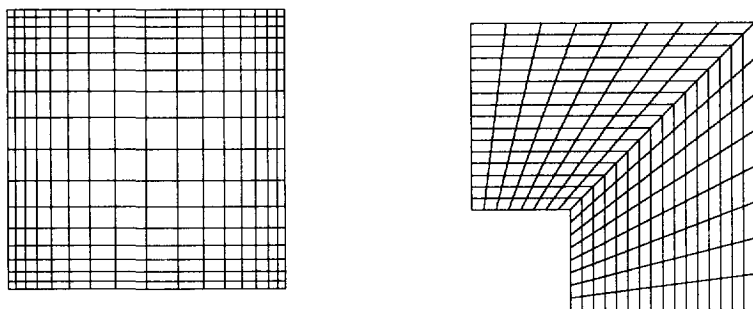


Figure 4.8: The different cavities, 16×16 grid

Boundary conditions for the cavities are: $u = 0$ on left, right and lower boundary, $u^1 = -1, u^2 = 0$ on the upper boundary. The kink in the grid lines of the L-shaped cavity does not lead to discretization accuracy difficulties for the problem considered, because the high velocity region is in a smooth part of the grid, and, as in Section 3.4, low velocity regions do not lead to accuracy problems in a non-smooth part of a grid.

All problems are solved for $Re = 100$ and $Re = 1000$. For both Reynolds numbers the smoothing iterations are performed in alternating directions, although this is not necessary for SCGS for $Re = 100$. The algorithm starts with nested iteration; the starting vector is the zero solution on the coarsest ($= 2 \times 2$) grid.

The results given are average reduction factors r_{10} for $Re = 100$ and r_{20} for $Re = 1000$ for the W-cycle ($\nu_1 = \nu_2 = 1$).

Table 4.3 presents r_{10} for the first two problems for $Re = 100$. Table 4.4 shows r_{20} for $Re = 1000$.

Table 4.3: Average reduction factors r_{10} for the square cavity problems, $Re = 100$.

grid	# levels	sq. cav. eq. grid			sq. cav. neq. grid		
		SCGS	SCGS/LS	SCAL	SCGS	SCGS/LS	SCAL
16 × 16	4	.208	.220	.124	.155	.209	.140
32 × 32	5	.153	.200	.104	.227	.270	.209
64 × 64	6	.147	.153	.069	.242	.336	.200
128 × 128	7	.142	.169	.055	.241	.520	.158

For the square cavity with the non-equidistant 128 × 128-grid r_{10} , obtained with SCGS seems better than r_{10} for SCGS/LS. This is due to very good reduction in the first iterations. The convergence factor $\rho (= r_{\infty})$ however, is the same for both smoothing methods: $\rho \approx 0.55$, while r_{10} obtained with SCAL is representative for ρ . For low Reynolds numbers it can be observed that SCAL performs better than the other two smoothers. For the equidistant cell distribution reduction factors even improve when the grid is larger than 64 × 64; for the non-equidistant distribution level-independent convergence factors ρ are found only with SCAL.

Table 4.4: Reduction factors r_{20} for the square cavity problems, $Re = 1000$.

grid	# levels	sq. cav. eq. grid			sq. cav. neq. grid		
		SCGS	SCGS/LS	SCAL	SCGS	SCGS/LS	SCAL
16 × 16	4	.516	.466	.192	.405	.459	.361
32 × 32	5	.565	.495	.339	.445	.459	.366
64 × 64	6	.564	.569	.443	.530	.499	.400
128 × 128	7	.517	.558	.493	.456	.519	.449

For $Re = 1000$ there is not much difference between the three smoothers. All reduction factors are level-independent. For the non-equidistant grid r_{20} is not worse than for the equidistant distribution. It is surprising that reduction factors for SCAL are not much better than those for the other smoothers when almost square cells are generated. This is also mentioned in [101].

For the problems investigated the behaviour of the two smoothing methods SCGS and SCGS/LS is similar. It seems that SCGS is more robust in the variation of underrelaxation factors, especially for low Reynolds flows and in the choice of smoothing directions.

In Figure 4.9 the streamlines for the L-shaped cavity are presented for the 128×128 grid. The same problem has been solved with the finite element package Sepran, and it appeared that, apart from some upwind phenomena in high velocity regions a similar streamline pattern was found. Table 4.5 gives r_{nit} for both Reynold num-

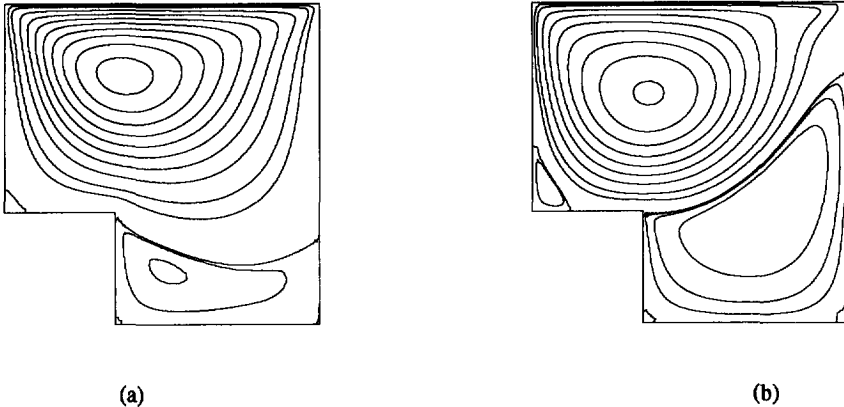


Figure 4.9: Streamlines for the L-shaped cavity, a) $Re = 100$, b) $Re = 1000$, 64×64 grid.

bers in the L-shaped cavity. It can be observed that indeed the SCAL smoother is

Table 4.5: Average reduction factors for the L-shaped cavity problems, $Re = 100$ and $Re = 1000$.

grid	# levels	$Re = 100$			$Re = 1000$		
		SCGS	SCGS/LS	SCAL	SCGS	SCGS/LS	SCAL
16×16	4	.334	.311	.208	.608	.637	.390
32×32	5	.402	.370	.196	.686	.687	.562
64×64	6	.411	.410	.186	.767	.688	.507
128×128	7	.391	.473	.250	.813	.822	.586

more robust, because r_{nit} is clearly better for SCAL and comparable to r_{nit} found for the square cavity.

Reduction factors for domains in which stretched cells occur are not satisfactory for SCGS and SCGS/LS, as can be seen in the following simple example: the flow through a straight channel. Figure 4.10 shows a sketch of the channel. The length

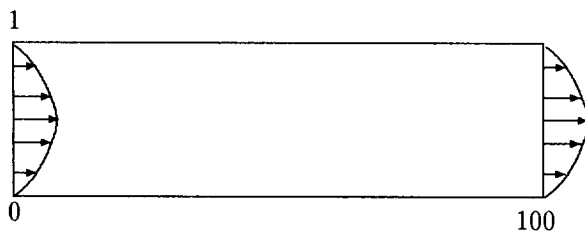


Figure 4.10: Sketch of the domain for the straight channel.

of the channel $L = 100$; the width $d = 1$. An equidistant grid with cells of aspect ratio 100 is put over the domain. Parabolic in- and outflow velocity profiles are prescribed. The exact Poiseuille flow solution is recovered accurately. The pressure distribution is linear, as expected. Table 4.6 presents r_{10} for Stokes flow and r_{20} for a flow at $Re = 1000$. These reduction factors are satisfactory and level-independent,

Table 4.6: Average reduction factors for a flow through a straight channel with a mesh with aspect ratio 100.

grid	r_{10} , Stokes	r_{20} , $Re = 1000$
32 x 32	.122	.463
64 x 64	.126	.487
128 x 128	.129	.500

whereas SCGS and SCGS/LS reduction factors are close to 1 ($\rho = 0.992$ for SCGS).

SCAL results. We will continue with the investigation of SCAL for more difficult grid distributions. The discretization in curvilinear coordinates is tested on domains in which some grids with severe jumps in mesh size and mesh direction are put. Two types of flows have been investigated: channel flows and driven cavity flows. For the channel flow problems the average reduction factors r_{nit} , defined in (4.31) are calculated. The multigrid schedule used is the W-cycle, unless stated otherwise. For all problems the number of pre-smoothing iterations is 1, as is the number of post-smoothing iterations. On the coarsest grid the problem is solved "exactly" by performing many smoothing iterations (10 or more).

Channel flow problems.

Because only Dirichlet boundary values are implemented in the code at present we are somewhat restricted. However, we are still able to investigate some interesting channel flow problems. For all channel flow problems nested iteration is used, so the algorithm starts on the coarsest grid.

Flow over a cylindrical protuberance. The first flow considered here is the linear shear flow over a cylindrical protuberance in a channel (Figure 4.11). This flow is

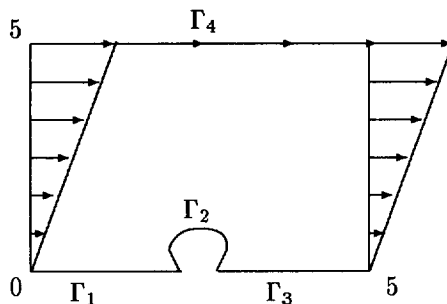


Figure 4.11: A sketch of the problem of a flow over a cylindrical protuberance.

a two-dimensional version of the flow over a spherical protuberance in the viscous sublayer in a turbulent boundary layer, which is of interest to turbulent flow researchers. The problem is artificial with only Dirichlet boundary conditions. We are restricted to a Stokes flow. However, it is interesting to show the possibilities of a curvilinear discretization for this case. A multidomain approach is a natural approach but we forego this approach on purpose here, to study multigrid convergence on a contorted grid. Boundary Γ_4 consists of three parts, where on the in-

and outflow part a linear velocity profile is prescribed:

$$u_n = y ; u_t = 0.$$

On the upper boundary u_t is prescribed: $u_t = y$; $u_n = 0$, while on boundary Γ_1, Γ_2 and Γ_3 $u_n = u_t = 0$. Figure 4.12 shows the 16×40 -mesh, streamlines and isobars. The multigrid algorithm performs well, and gives $r_{10} = 0.105$.

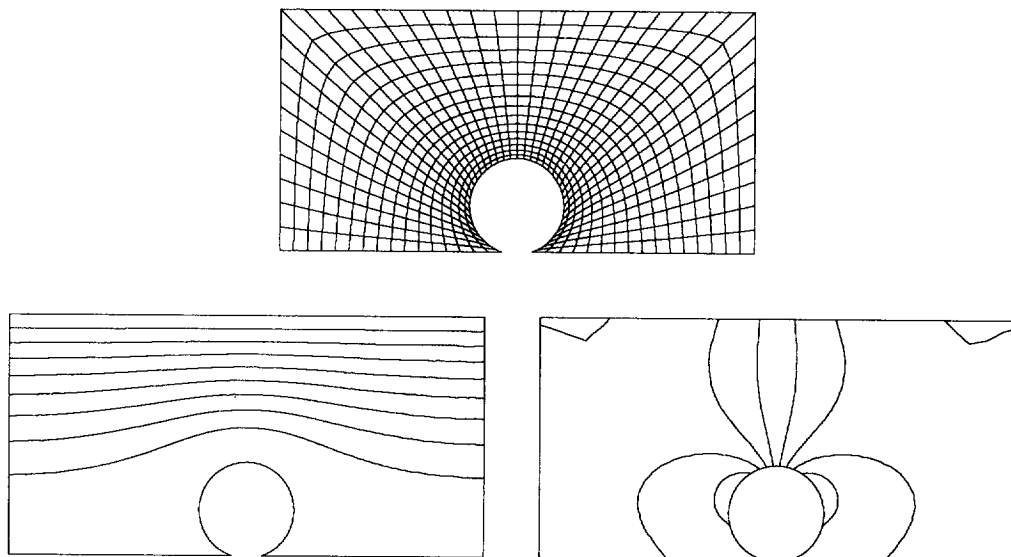


Figure 4.12: The 16×40 -mesh, streamlines and isobars for a Stokes flow over a cylindrical protuberance.

The hole-pressure problem. A rather difficult channel flow problem is the so-called hole-pressure problem ([53]). The flow domain is depicted in Figure 4.13. It consists of a uniform channel of height d and length L , into which a cavity of width b and depth h has been cut. Dirichlet boundary conditions are prescribed on the whole boundary. In [53] this hole-pressure problem is proposed as new test case for numerical codes. The problem is also of practical importance: It is the basis of a scientific instrument to help characterizing non-Newtonian fluids. The mathematical and experimental problem is to determine the difference between the normal stress acting on the bottom of the slot and that acting on the upper wall above the opening to the slot. This quantity is the so-called 'hole pressure' Δ , given by

$$\Delta = \frac{1}{s_1} \int_{s_1} -p(x, 1) dx - \frac{1}{s_2} \int_{s_2} -p(x, -d) dx \quad (4.37)$$

The geometry of the reference domain is defined by: $L = 5$; $d = 1$; $b = 1$; $h = 1$; $s_1 = s_2 = 1$. The boundary conditions are given by:

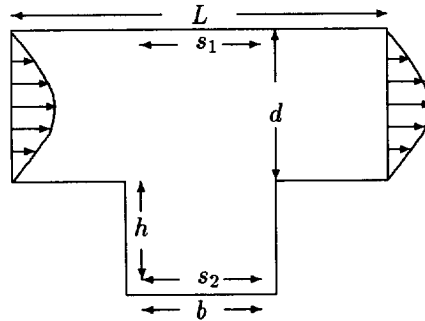


Figure 4.13: The domain for the hole pressure problem.

$$\begin{aligned} \text{On } \Gamma_2 \text{ and } \Gamma_4 & : u_n = 0 ; u_t = 0 \\ \text{On } \Gamma_1 \text{ and } \Gamma_3 & : u_n = y(1 - y) ; u_t = 0 \end{aligned} \quad (4.38)$$

So $u^{max} = 1/4$; $u^{average} = 1/6$ and the mass flow $Q = 1/6$. For the Reynolds number, defined as $Re = Q/\nu$, it follows that $Re = 1/6\nu$.

The mesh is generated as follows. Boundary part Γ_4 is divided into 5 parts (see Figure 4.14). Each part is connected to a part of Γ_2 in the manner shown in the figure. This results in a coarse mesh, which is refined to a 32×112 -mesh, which is then smoothed a little. Figure 4.15 shows a part of this mesh.

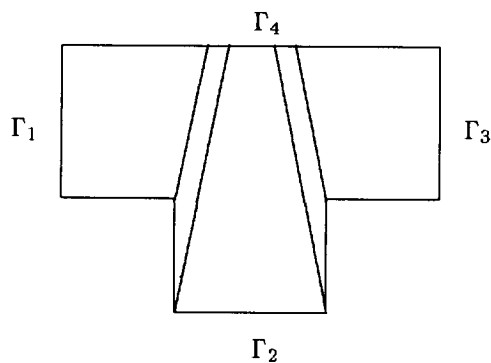


Figure 4.14: A sketch of the parts of the domain, which are connected when the mesh is generated.

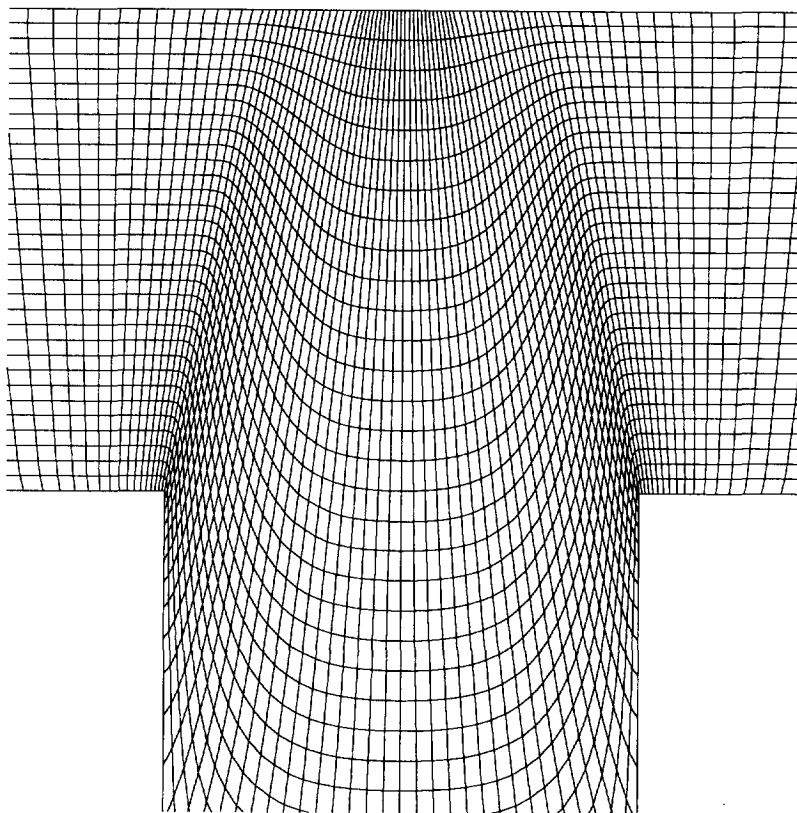


Figure 4.15: A part of the 32 x 112 -mesh for the hole-pressure problem.

In [53] hole pressures are presented up to $Re = 10$ calculated with a finite element code on a very fine mesh. However, the hole pressures from [53] are not calculated with the boundary conditions in [53], but with the boundary conditions presented here ([78]). In Table 4.7 the calculated hole pressures from the 32×112 -mesh are compared to them. Figure 4.16 presents streamlines and the isobars for $Re = 10$.

Table 4.7: Hole pressures Δ compared with [53] for several Reynolds numbers.

Re	Δ calc.	Δ [53]
1	0.064	0.049
5	0.236	0.226
10	0.426	0.383

They resemble the distributions in [53] very well. The multigrid performance is satisfactory as well: $r_{15} = 0.247$ for $Re = 10$ and $r_{25} = .505$ for $Re = 100$.

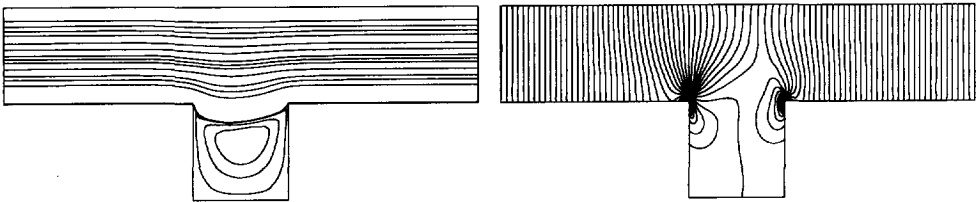


Figure 4.16: Streamlines and isobars for the hole-pressure problem.

Backward facing step. Another flow, heavily studied over the years, is laminar flow over a backward facing step. A GAMM workshop was held in 1983 in *Bièvres* on this flow problem ([58]). Four cases were studied there, two of them are examined here in curvilinear coordinates, namely case 2 and 4, as defined in [58]. The domain for these two cases is sketched in Figure 4.17.

The geometrical parameters are: $L_1 = 22$, $L_2 = 3$, $H = 1$ and $h = 0.5$. The computations presented are for $Re = 50$ and $Re = 150$, with the Reynolds number defined as

$$Re = u^{max} \cdot (H - h) / \nu \quad (4.39)$$

with u^{max} the maximum value of the velocity at the entrance, (the normal and tangential velocity is prescribed: $u_n = 16y(0.5 - y)$ at the entrance and

$u_n = 2y(1 - y)$ at the outlet; $u_t = 0$) and $\nu (= \mu/\rho)$ the viscosity.

Figure 4.18 shows a part of the 32×64 -mesh. Deliberately, no attempt was made to make this mesh more smooth. In Figure 4.19 the isobars and streamlines for the flow at $Re = 50$ are given, together with a more detailed picture of the streamlines in the recirculation zone. In Figure 4.20 these results are shown for the flow at $Re = 150$. In Table 4.8 the MG performance is given for a 16×32 grid and a 32×64 grid. The coarsest grid is taken to be the 4×8 grid. For the 32×64 grid the length of the recirculation zone, scaled by $(H - h)$ is calculated and presented in Table 4.8. This length resembles the length found by many of the GAMM contributors very well ([58]). The reduction factors look satisfactory and level-independent.

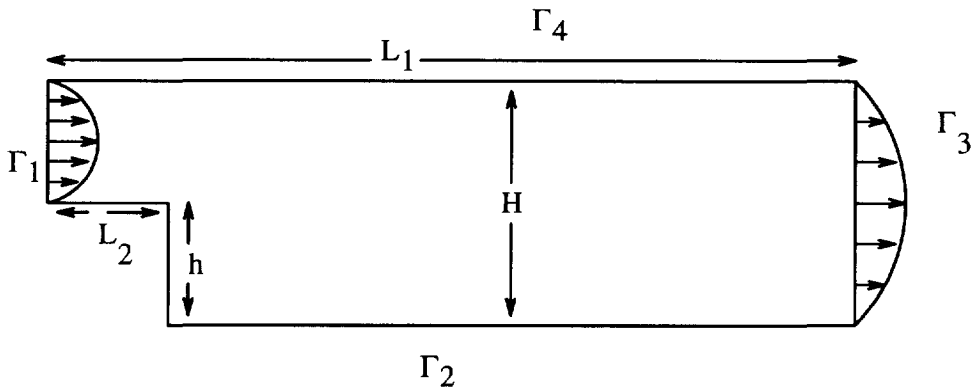


Figure 4.17: The domain (not to scale) for the flow over a backward facing step.

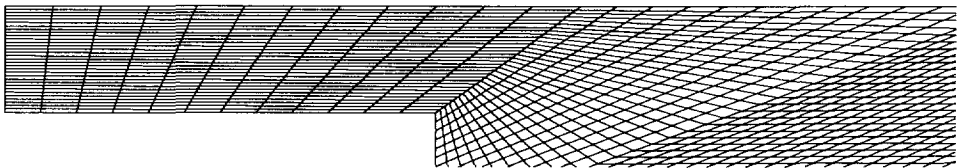


Figure 4.18: A part of the 32×64 mesh.

Table 4.8: Average reduction factors and reattachment point for a flow over a backward facing step.

	$r_{20}(16 \times 32)$	$r_{20}(32 \times 64)$	reattachment point
$Re = 50$.449	.482	2.03
$Re = 150$.654	.656	5.00

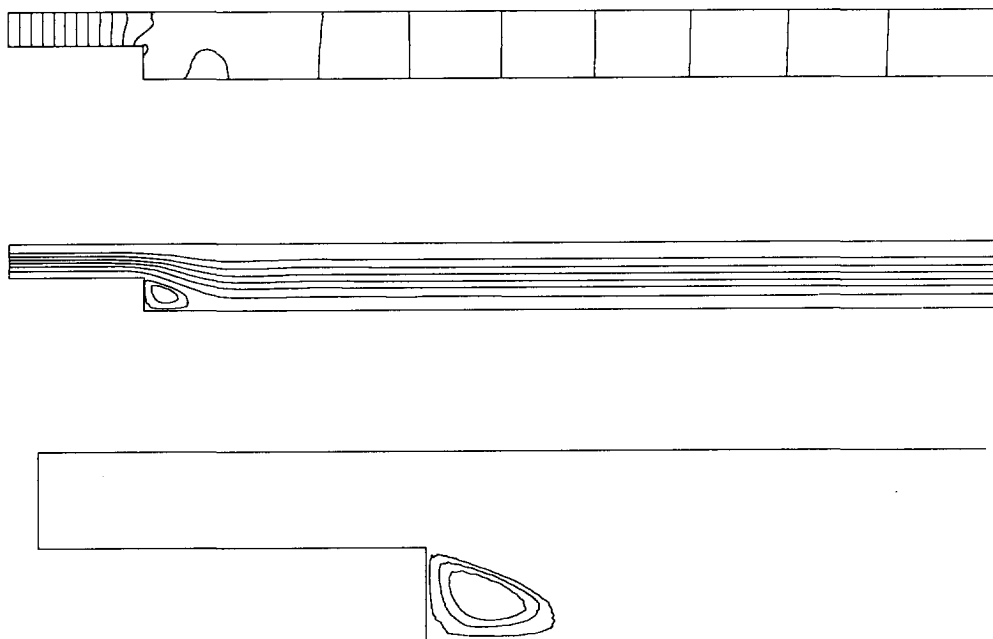


Figure 4.19: Isobars and streamlines for $Re = 50$.

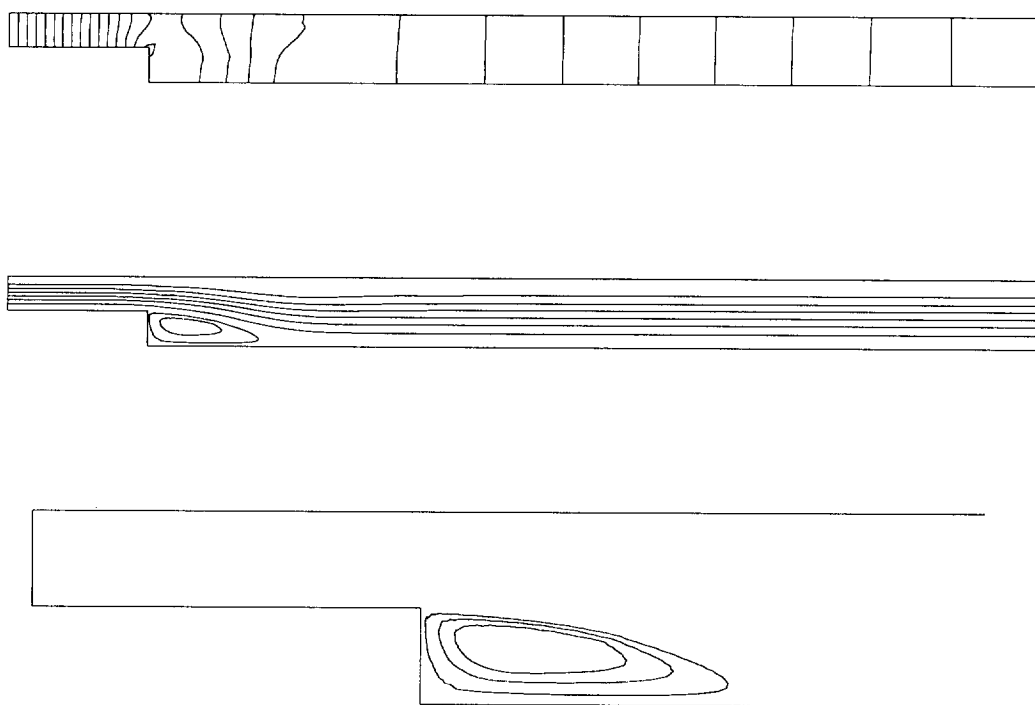


Figure 4.20: Isobars and streamlines for $Re = 150$.

Driven cavity problems

The multigrid method will now be tested even more severely on robustness with driven cavity flows in a square cavity, where the grids are strongly refined near the boundaries. The definition of the average reduction factors will change a little, because in the first iteration the reduction is so good, that this influences r_{nit} in a spurious way. Therefore new average reduction factors μ_{nit} are defined as

$$\mu_{nit} = \left(\frac{\|res_{nit}\|}{\|res_1\|} \right)^{\frac{1}{nit-1}} \quad (4.40)$$

with $\|\cdot\|$ the l_2 -norm, and res_1 the residual after the first iteration. The reduction in the first iteration is not taken into account, which explains the possible differences between the results obtained here, and those in earlier calculations and in [66].

We also give

$$\nu_n \equiv \left\| \frac{res_n}{res_{n-2}} \right\|^{1/2} \text{ for } n = nit. \quad (4.41)$$

In many cases ν_n is found to be approximately constant for n close to nit , in which case we have found the asymptotic convergence factor $\mu_\infty \cong \nu_{nit}$.

The number of MG-iterations that were performed depended on the reduction factor μ_{nit} , as follows:

$$\begin{aligned} \mu_{nit} &\leq 0.15 && ; && nit = 10 \\ 0.15 < \mu_{nit} &\leq 0.4 && ; && nit \approx 20 \\ \mu_{nit} &> 0.4 && ; && nit \geq 25 \end{aligned} \quad (4.42)$$

Three grids were tested:

- i) the equidistant grid.
- ii) a grid in which the boundary cells have aspect ratio 10, see Figure 4.21.
- iii) the boundary cells have aspect ratio 100, see Figure 4.21.

This is a severe test on robustness; stretched cells occur in all directions. The three iteration cycles (V , F and W) are compared for these test problems. Table 4.9 shows μ_{nit} and ν_{nit} for grids with 5 to 7 levels for flows at $Re = 100$ and $Re = 1000$. Figure 4.22 presents the l_2 -norm of the residual graphically on logarithmic scale versus the number of iterations for the three iteration cycli on the 128×128 -grids with aspect ratio 1 (Figure 4.22a and 4.22b), and with aspect ratio 100 (Figure 4.22c and 4.22d) for flows at $Re = 100$ and $Re = 1000$.

The average reduction factors are satisfactory and for many aspect ratios and many Reynolds numbers below 0.4. The performance of SCGS (not shown) is poor in the cases with large aspect ratios. In many cases μ_{nit} for the F-cycle is comparable to μ_{nit} for the W-cycle. For low Reynolds numbers the cheaper V-cycle is even more efficient. A remarkable improvement of reduction factors for the F- and W-cycle is observed when square cells are used. Then, for $Re = 100$ μ_{10} is extremely

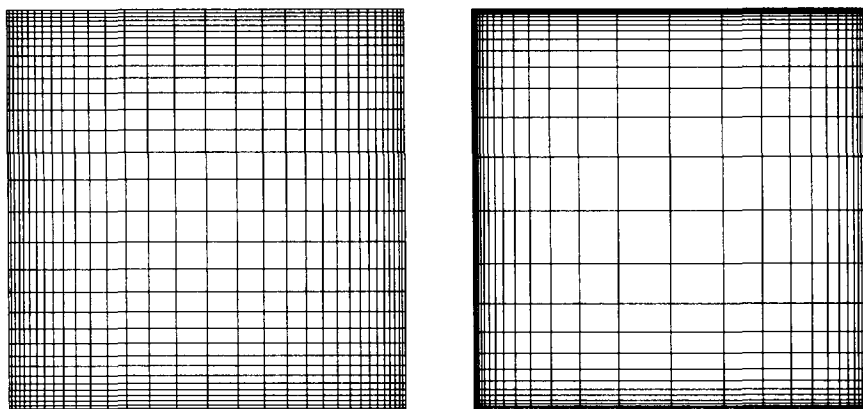


Figure 4.21: The 32×32 mesh with aspect ratio 10 and 100 at the boundary.

satisfactory, while the only significant difference between the W - and F -cycle is for $Re = 1000$ on the 128×128 -grid.

Table 4.9: Average and asymptotic reduction factors for driven cavity problems.

grid:	(i) eq. grid											
	$Re = 100$						$Re = 1000$					
	V		F		W		V		F		W	
	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}
32^2	.26	.26	.12	.15	.12	.15	.55	.61	.37	.38	.34	.36
64^2	.27	.32	.07	.10	.07	.09	.63	.80	.51	.61	.47	.55
128^2	.36	.43	.06	.07	.06	.07	.74	.88	.62	.71	.52	.58

grid:	(ii) ratio 1 : 10											
	$Re = 100$						$Re = 1000$					
	V		F		W		V		F		W	
	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}
32^2	.27	.31	.27	.32	.32	.37	.52	.59	.36	.40	.35	.41
64^2	.31	.35	.28	.33	.28	.33	.58	.65	.40	.46	.38	.42
128^2	.32	.34	.27	.30	.27	.30	.70	.79	.58	.71	.58	.72

grid:	(iii) ratio 1 : 100											
	$Re = 100$						$Re = 1000$					
	V		F		W		V		F		W	
	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}
32^2	.25	.33	.29	.42	.28	.40	.40	.45	.32	.38	.33	.37
64^2	.27	.33	.28	.34	.28	.33	.50	.58	.32	.37	.32	.37
128^2	.36	.39	.32	.35	.31	.35	.60	.69	.49	.63	.50	.64

In Table 4.10 the CPU times on a Convex 3840 computer are presented for one smoothing iteration and one V-, F- and W-cycle. As already stated parallelization is not employed here, no special ordering strategies for the data structure are employed to profit from vectorization. From Table 4.10 it can be seen that the work for a W-cycle is about 4.8 times the work for one smoothing iteration, which is not surprising, since the theoretical ratio counting only smoothing work would be 4 in this case. The theoretical ratio for the computational work of the V-cycle is $8/3$; in Table 4.10 it is 3. For the F-cycle the theory predicts $32/9$, Table 4.10 gives 3.9.

Table 4.10: CPU times on a Convex 3840.

cycle / grid	32×32	64×64	128×128
1 sm.	0.33s	1.24s	4.40s
V	0.94s	3.42s	13.2s
F	1.34s	4.67s	17.4s
W	1.56s	5.47s	21.4s

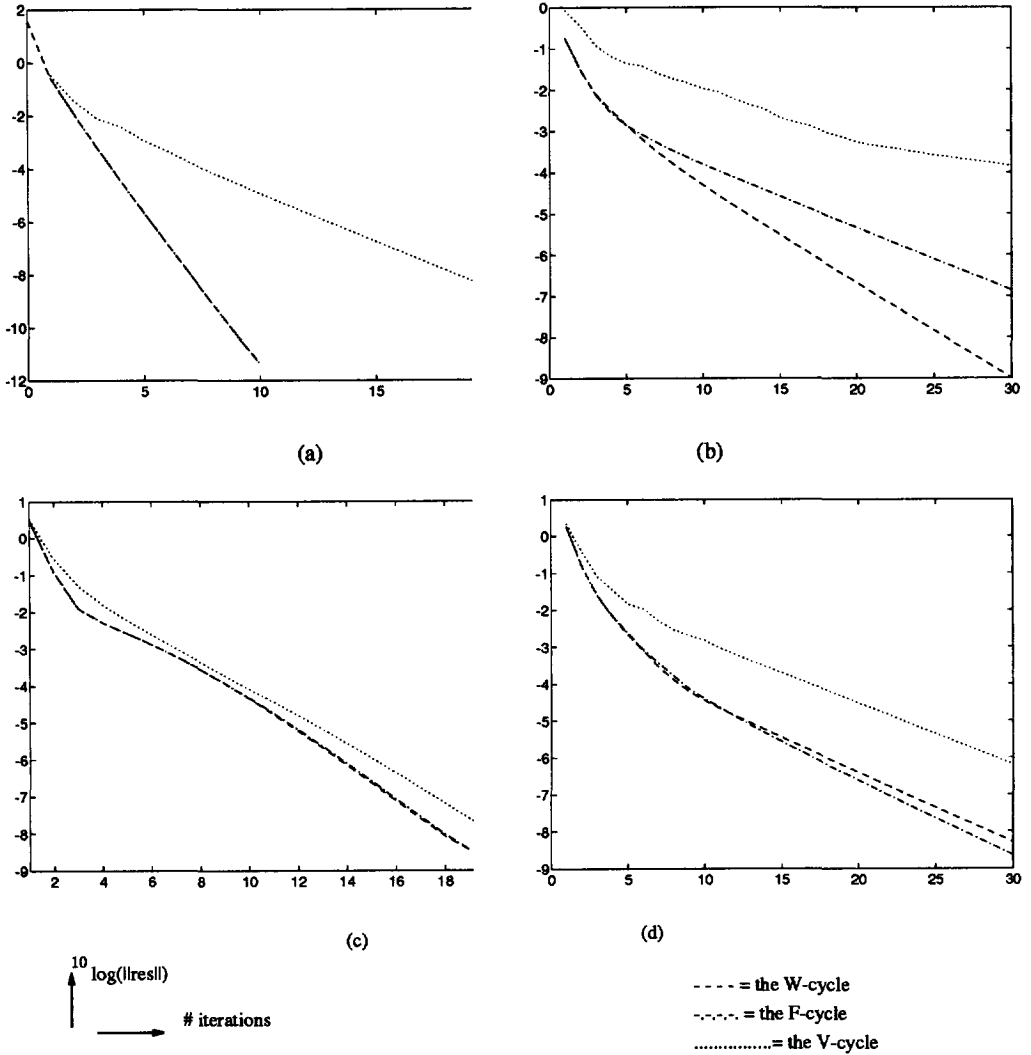


Figure 4.22: A comparison between the V -, F - and W -cycle: the l_2 -norm of the residual (logarithmic scale) versus the number of iterations for the 128×128 meshes; (a) $Re = 100$ (cell aspect ratio 1), (b) $Re = 1000$ (cell aspect ratio 1), (c) $Re = 100$ (cell aspect ratio 100 at the boundary), (d) $Re = 1000$ (cell aspect ratio 100 at the boundary).

Benchmark problems. Finally, in order to investigate the effect of non-orthogonality and grid line curvature the multigrid results of the benchmark problems presented in Section 3.4 are shown. The F -cycle is used, because it is cheaper than the W -cycle, while for many cases similar reduction factors are found (Table 4.9). The multigrid results for the skewed cavities with skewing angle 45° (testcase 1) and 30° (testcase 2) with as finest grid a 256×256 -grid and 8 multigrid levels are presented in Table 4.11.

In the domain with skewing angle 45° the convergence was slowest ($\nu_{30} \approx 0.79$ for the F -cycle on a 128×128 -grid). The W -cycle shows a similar performance, $\nu_{30} \approx 0.75$. The reduction factors for the finest (256×256) grid improve. The slower convergence is due to the upwind-central switch for the convection terms, because in the domain with skewing angle 30° (with even heavier non-orthogonal cells) the reduction factors are satisfactory. Furthermore, the same problem has been investigated with only the first order upwind scheme for the convection terms in (3.8) (prescribing $\alpha^{(k,l)} = 1$ in (3.25)). Much better convergence results were then obtained: $\nu_{30} = 0.53$ for the W -cycle, which confirms our presumption.

The L-shaped benchmark problem is solved on very fine grids (512×512) with 9 multigrid levels. These results are presented in Table 4.12. Figure 4.23 presents the l_2 -norm of the residual (logarithmic scale) versus the number of iterations for the L-shaped cavity for $Re = 100$ and $Re = 1000$. As expected the slopes of all lines in this figure are almost identical. This indicates the level-independent convergence rates of the multigrid method used. On the two finest grids the CPU times on the Convex 3840 for one F -cycle were:

256×256 – grid : 75.2 seconds

512×512 – grid : 350 seconds

The average reduction factors for $Re = 100$ are very good, only 15 iterations are

Table 4.11: Average and asymptotic reduction factors for skewed driven cavity problems, F -cycle, $\nu_1 = \nu_2 = 1$.

angle	$\beta = 45^\circ$				$\beta = 30^\circ$			
	100		1000		100		1000	
Reynolds number	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}	μ_{nit}	ν_{nit}
32×32 -grid	0.191	0.221	0.617	0.695	0.304	0.373	0.629	0.674
64×64 -grid	0.184	0.248	0.712	0.718	0.336	0.412	0.533	0.589
128×128 -grid	0.198	0.253	0.718	0.791	0.345	0.428	0.530	0.607
256×256 -grid	0.194	0.250	0.604	0.719	0.352	0.431	0.445	0.555

Table 4.12: Average and asymptotic reduction factors for an L-shaped cavity problem, F-cycle, $\nu_1 = \nu_2 = 1$. (for $Re = 100$ a 512×512 -grid was not investigated)

Reynolds number	100		1000	
	μ_{nit}	ν_{nit}	μ_{nit}	n_{unit}
32×32 -grid	0.184	0.198	0.479	0.494
64×64 -grid	0.195	0.252	0.566	0.623
128×128 -grid	0.198	0.265	0.595	0.669
256×256 -grid	0.204	0.252	0.588	0.644
512×512 -grid	—	—	0.529	0.592

needed to reduce the residual by 8 orders of magnitude. Also for $Re = 1000$ the average reduction factors are satisfactory; they are similar to the factors for other geometries, and level-independent, which means that almost the same number of iterations is needed to reduce a residual by several orders of magnitude for small and very large grids. The smoother can deal with cells of varying size coming from a mesh generator. Furthermore, it is found that the code is insensitive to large variations of underrelaxation parameters for low Reynolds numbers, and that for the higher Reynolds number small variations are allowed.

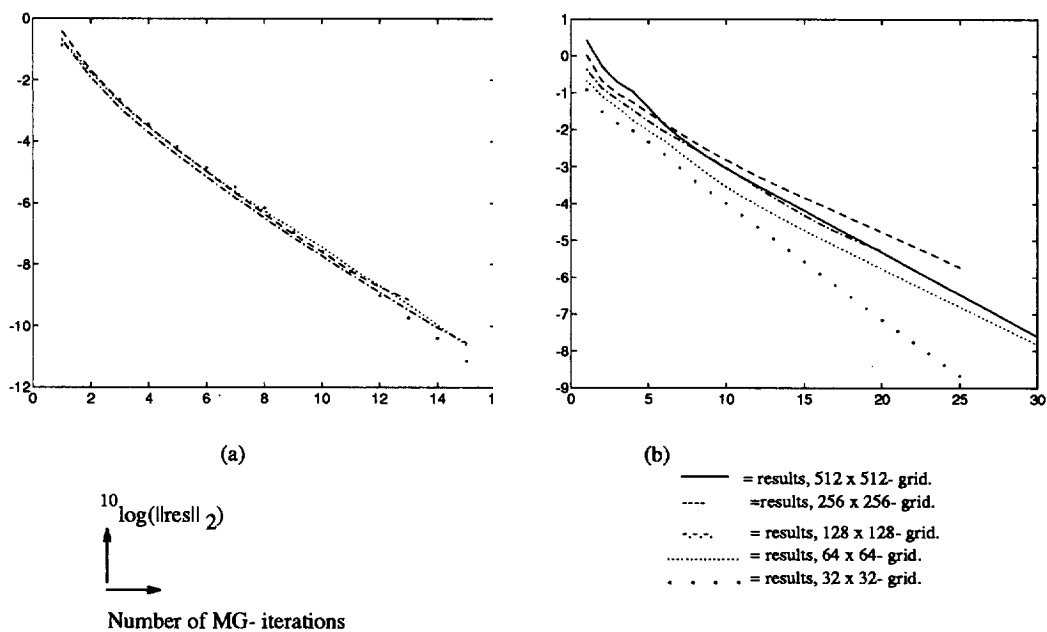


Figure 4.23: The l_2 -norm of the residual versus the number of iterations for the F(1,1)-cycle for the L-shaped cavity (1 pre-, 1 post smoothing), (a) for $Re = 100$ and (b) $Re = 1000$.

4.3 Steady incompressible flow around objects.

A further extension of the code was made to enable a study of flow problems around objects in two (and in the future three) dimensions. In order to solve these flow problems with a single block discretization, periodic boundary conditions along an artificial interior boundary are necessary.

In certain cases the discrete system of equations becomes singular when periodic boundary conditions are used. The solution algorithm must be able to overcome this problem.

The discretization, the adaptation of the existing steady-state multigrid solution algorithm to the singular systems which appear in the smoothing algorithm used, and some steady flow results, like the steady flow in two dimensions around a circle and an ellipse for a (relatively) low Reynolds number are presented here. A logical extension will be the unsteady flow problem for a higher Reynolds number, but these results are not presented in this thesis.

Adaptation of the discretization. In order to solve a flow around an object with a single block discretization an interior (artificial) cut is introduced in the Cartesian domain (Figure 4.24). On this cut (with parts Γ_2 and Γ_4) periodic boundary conditions (PBCs) are prescribed:

$$\mathbf{u}|_{\Gamma_2} = \mathbf{u}|_{\Gamma_4} \quad (4.43)$$

$$p|_{\Gamma_2} = p|_{\Gamma_4} \quad (4.44)$$

In order to satisfy these boundary conditions an extra line of unknowns is introduced on Γ_4 . On the far field outer boundary Γ_1 a uniform flow is prescribed, while

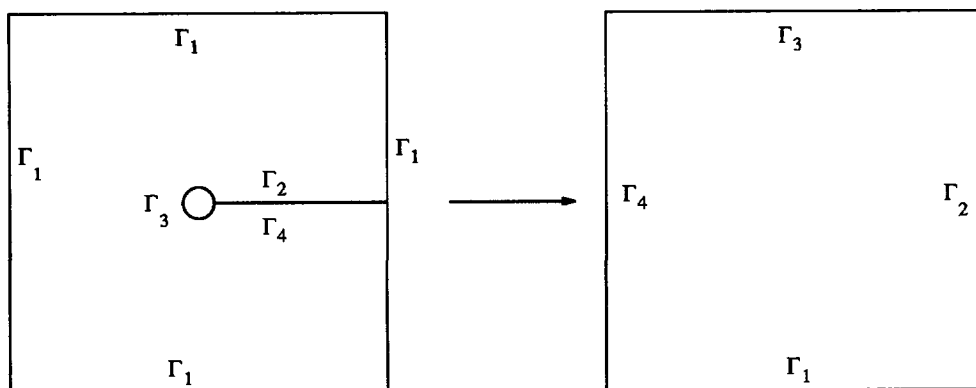


Figure 4.24: The physical domain Ω with an interior cut for flow around an object (Γ_3), and the computational domain G .

on the object, Γ_3 , the no-slip condition $u_t = 0$, $u_n = 0$ is prescribed.

Some adaptations need to be made in the discretization in curvilinear coordinates when PBCs are introduced. For example, on the periodic boundaries of the computational domain one should not discretize the geometrical quantities using extrapolation, as is done when other boundary conditions are involved. The geometrical quantities should be adapted to the periodicity condition; this is explained below. The covariant base vectors $\mathbf{a}_{(\alpha)}$ are given by (2.16):

$$a_{(1)}^\beta = \frac{\delta x^\beta}{\delta \xi^1}, \quad a_{(2)}^\beta = \frac{\delta x^\beta}{\delta \xi^2} \quad (4.45)$$

Choosing $\delta \xi^\alpha = 1$ on the finest grid G_h , it was found that for example in the V^2 -points $\mathbf{a}_{(1)}$ can be calculated straightforward by (2.53):

$$a_{(1)}^\beta|_{i,j-\frac{1}{2}} = x^\beta|_{i-\frac{1}{2},j-\frac{1}{2}} - x^\beta|_{i+\frac{1}{2},j-\frac{1}{2}} \quad (4.46)$$

with the numbering of Figure 2.4.

In all other points in G_h the covariant base vectors are found by means of bilinear interpolation. So in a vertex point $(i - \frac{1}{2}, j - \frac{1}{2})$ it follows that

$$a_{(1)}^\beta|_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{1}{2}(a_{(1)}^\beta|_{i,j-\frac{1}{2}} + a_{(1)}^\beta|_{i-1,j-\frac{1}{2}}) \quad (4.47)$$

In a vertex on boundary Γ_4 $a_{(1)}^\beta|_{i-1,j-\frac{1}{2}}$ is not a part of G_h . When the domain is not periodic the value of this base vector is found with extrapolation. However, in the case of periodic boundaries $a_{(1)}^\beta|_{i-1,j-\frac{1}{2}}$ must have the value of $a_{(1)}^\beta$ in the last interior point near the right vertical boundary Γ_2 . So:

$$a_{(1)}^\beta|_{(i=0,j-\frac{1}{2})} = a_{(1)}^\beta|_{(i=ni,j-\frac{1}{2})} \quad \text{for } j = 1 \text{ to } nj \quad (4.48)$$

Adaptation of the multigrid solution algorithm. Due to the PBCs some changes have to be made in the solution algorithm. The restriction and prolongation operators for example, that have been adapted to boundary conditions in case of Dirichlet conditions ((4.5) to (4.18)), can now employ periodicity. So for the restriction of residuals we have:

$$\begin{aligned} r_{i,j}^{1(k-1)} &= 1/8(r_{2i-2,2j}^{1(k)} + r_{2i-2,2j-1}^{1(k)} + r_{2i,2j}^{1(k)} + r_{2i,2j-1}^{1(k)}) + \\ &\quad 1/4(r_{2i-1,2j}^{1(k)} + r_{2i-1,2j-1}^{1(k)}) \\ r_{i,j}^{2(k-1)} &= 1/8(r_{2i,2j-2}^{2(k)} + r_{2i-1,2j-2}^{2(k)} + r_{2i,2j}^{2(k)} + r_{2i-1,2j}^{2(k)}) + \\ &\quad 1/4(r_{2i,2j-1}^{2(k)} + r_{2i-1,2j-1}^{2(k)}) \\ r_{i,j}^{3(k-1)} &= 1/4(r_{2i-1,2j}^{3(k)} + r_{2i-1,2j-1}^{3(k)} + r_{2i,2j}^{3(k)} + r_{2i,2j-1}^{3(k)}) \end{aligned} \quad (4.49)$$

with: $i = 1$ to n_i , and $r_{0,j}^{i(k)} = r_{n_i,j}^{i(k)}$ etc.

The SCAL smoother used is able to deal with stretched cells. In the present type of flow problem this is a necessary requirement, due to the fact that near the far field outer boundary and in the heavily refined region near the object highly stretched cells in both directions occur. In Figure 4.25 a grid is depicted. The outer boundary is taken to be a square, because this facilitates the prescription of the uniform flow ($u^1 = 1, u^2 = 0$).

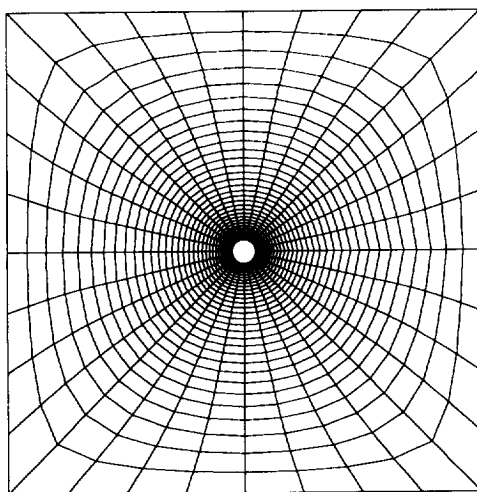


Figure 4.25: The 32×32 grid for a flow around a circular cylinder.

SCAL must also be adapted to the PBCs. For a relaxation sweep along vertical lines the smoothing step hardly changes. The first line of cells ($i = 1; j = 1$ to n_j) becomes an interior line, due to the flux unknowns at boundary Γ_4 , where the PBCs are prescribed. When the unknowns on the last vertical line of cells are smoothed the flux unknowns at boundary Γ_4 are updated for the second time.

For a sweep along an horizontal line more adaptation is needed. Implementation of the PBCs will lead here to (partial) loss of the band structure of the matrix, because the stencil of the first cell is connected to the last cell. The resulting system of equations would have the following structure:

$$\begin{pmatrix} * & \cdot & \cdot & * & & & & * & * & * & * \\ \cdot & & & & & & & & * & * & * \\ \cdot & & & & & & & & & & \\ * & & & & & & & & * & * & * \\ & \cdot & & & & & & & & & \\ & & \cdot & & & & & & & & \\ & & & \cdot & & & & & & & \\ & & & & \cdot & & & & & & \\ & & & & & \cdot & & & & & \\ * & & & & & & & & & & * \\ * & * & * & * & & & & & & & \cdot \\ * & & & & & & & & & & \cdot \\ * & * & * & * & & & & * & \cdot & \cdot & * \end{pmatrix} \begin{pmatrix} V^1 \\ V^2 \\ p \\ V^2 \\ V^1 \\ V^2 \\ p \\ V^2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} = \mathbf{r}^i \quad (4.50)$$

with a bandwidth of 13 elements, where for example:

$$\begin{aligned} r_{i+1/2,j}^1 = & c_2^1 V_{i-1/2,j-1}^1 + c_3^1 V_{i+1/2,j-1}^1 + c_4^1 V_{i+3/2,j-1}^1 + \\ & c_7^1 V_{i-1/2,j+1}^1 + c_8^1 V_{i+1/2,j+1}^1 + c_9^1 V_{i+3/2,j+1}^1 + \\ & c_{14}^1 p_{i,j-1} + c_{15}^1 p_{i+1,j-1} + c_{18}^1 p_{i,j+1} + c_{19}^1 p_{i+1,j+1} \end{aligned} \quad (4.51)$$

However, this resulting matrix becomes ill-conditioned for high Reynolds numbers. A possible cure to this problem, not followed here, is to derive other equations and replace certain almost dependent equations. Here, per momentum equation one essential velocity connection across the boundary is deleted in the matrix. These velocities are taken from the preceding iteration and added to the right hand side of the equation. The resulting system has the following structure:

$$\begin{pmatrix} * & \cdot & \cdot & * & & & & * & * & * \\ \cdot & & & & & & & & * & * \\ \cdot & & & & & & & & & \\ * & & & & & & & & * & * \\ & \cdot & & & & & & & & \\ & & \cdot & & & & & & & \\ & & & \cdot & & & & & & \\ & & & & \cdot & & & & & \\ & & & & & \cdot & & & & \\ * & & & & & & & & & * \\ * & & * & * & & & & & & \cdot \\ * & & & & & & & & & \cdot \\ * & * & * & & & & & * & \cdot & \cdot & * \end{pmatrix} \begin{pmatrix} V^1 \\ V^2 \\ P \\ V^2 \\ V^1 \\ V^2 \\ p \\ V^2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{pmatrix} = \mathbf{r}^i \quad (4.52)$$

The condition of this matrix is much better. So for $r_{1/2,j}^1$ is found:

$$\begin{aligned}
 r_{1/2,j}^1 = & c_2^1 V_{ni-1/2,j-1}^1 + c_3^1 V_{1/2,j-1}^1 + c_4^1 V_{3/2,j-1}^1 + \\
 & c_5^1 V_{ni-1/2,j}^1 + \\
 & c_7^1 V_{ni-1/2,j+1}^1 + c_8^1 V_{1/2,j+1}^1 + c_9^1 V_{3/2,j+1}^1 + \\
 & c_{14}^1 p_{ni,j-1} + c_{15}^1 p_{1,j-1} + c_{18}^1 p_{ni,j+1} + c_{19}^1 p_{1,j+1} \quad (4.53)
 \end{aligned}$$

Periodicity is obtained when in each smoothing two sweeps are applied. In order to solve system (4.52) the LU algorithm for a band solver is rewritten and adapted to a band matrix with a small number of non-band elements.

In a wide range of satisfactory underrelaxation parameters the most suitable parameters are found to be:

$$\alpha_k = 0.5, \quad k = 1, 2, 3 \quad (4.54)$$

Results. The incompressible steady flow around a circle has been computed for several Reynolds numbers. The number of mesh points is 32×32 , 64×64 and 96×96 . Figure 4.26 shows a part of a grid. The number of multigrid levels is 4, 5 and 5. For the multigrid solution algorithm the F-cycle is used. The F-cycles

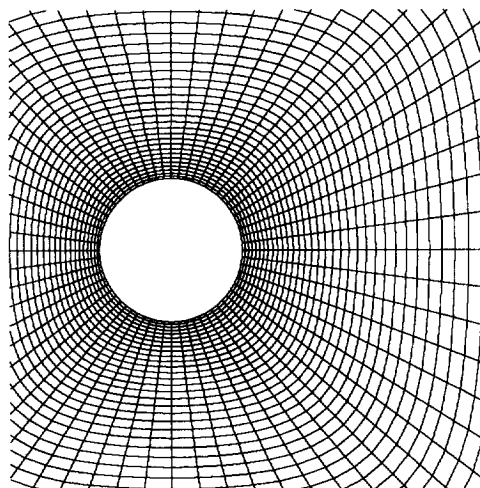


Figure 4.26: A part of the 64×64 -grid.

are preceded by nested iteration (work starts on the 4×4 -, 4×4 - and 6×6 -grid,

respectively). The number of multigrid iterations is 20 ($\nu_1 = \nu_2 = 2$). The outer boundary is constructed 20 units (cylinder diameters) from the cylinder, as in [88]. The cylinder diameter d is 0.2. The Reynolds numbers investigated are 10, 20, 30 and 40 (with Re based on the diameter). The average reduction factors r_{20} are presented in Table 4.13. Again satisfactory rates are found, which are level-independent and well below 0.35, with two pre- and post-smoothing iterations. For higher Reynolds numbers an unsteady flow, a future challenge for the unsteady code, is established. We compare recirculation length and the pressure coefficient here. The ratio between the length of the recirculation zone s and the diameter d has been calculated and compared to earlier results ([18]) in Table 4.13, and graphically in Figure 4.27. In [18] an overview is given of experimental results and of computational results based on matched Stokes and Oseen asymptotic expansions, on finite Fourier series ([63]) and on finite differences in cylindrical coordinates. The pressure coefficient c_p on the surface of the cylinder is calculated:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho u_\infty^2} \quad (4.55)$$

where p_∞ represents the pressure in the far field, and u_∞ the far field constant velocity.

Results are compared with [5] and [63]. The pressure coefficient on the surface at $\phi = \pi$, which is very difficult to calculate accurately, is found to be too large for the higher Reynolds numbers. Apart from this, good agreement is found between the reference results and our calculations. Figure 4.28 show streamlines and isobars for a flow from right to left at $Re = 20$ and $Re = 40$. Figure 4.29 presents the calculated c_p on the surface of the cylinder.

Another flow problem solved is the flow around an ellipse, with major axis $d_1 = 0.2$, and minor axis $d_2 = 0.1$ on a 96×96 -grid. The Reynolds numbers based on the minor axis are 10, 40 and 80. Figure 4.30 presents the c_p -coefficients; figure 4.31 presents streamlines and isobars for the flows at $Re = 40$ and $Re = 80$. The multigrid reduction factors are satisfactory and comparable to those found in Table 4.13:

$$\begin{aligned} Re = 10 : & \quad r_{20} = .371 \\ Re = 40 : & \quad r_{20} = .293 \\ Re = 80 : & \quad r_{20} = .296 \end{aligned} \quad (4.56)$$

It is observed that the present discretization of the incompressible Navier-Stokes equations on a staggered grid in non-orthogonal curvilinear coordinates produces accurate results for flows around objects, although highly non-uniform grids are involved. The single block discretization can handle steady external flow problems well. Level-independent satisfactory convergence rates for the multigrid solution algorithm have been found for the two test problems.

Table 4.13: Average reduction factors r_{20} plus the ratio between the length of the recirculation zone s and the diameter d for a flow around a circular cylinder.

	Grid	(32 × 32)	(64 × 64)	(96 × 96)	(ref[18])
Reynolds	—				
10	r_{20}	.220	.206	.305	.30
	s/d	.221	.257	.250	
20	r_{20}	.190	.207	.307	.90
	s/d	.753	.832	.905	
30	r_{20}	.247	.344	.293	1.5
	s/d	1.03	1.37	1.42	
40	r_{20}	.221	.256	.286	2.1
	s/d	1.03	1.87	2.05	

4.4 Conclusions.

Three smoothers in a standard nonlinear multigrid solution algorithm have been investigated on robustness for the discretized steady incompressible Navier-Stokes equations in general coordinates. The coupled line smoother SCAL performs best and is robust. Satisfactory convergence is observed for several domains, in which grids with stretching, non-orthogonality of cells and curvature are put. Internal driven cavity flows as well as external flows around objects are investigated with SCAL. For all flow problems good average reduction factors and convergence factors are observed. The singularity in the system in SCAL coming from the periodic boundary conditions for flows around objects could be dealt with relatively easily, and also for external flows with highly stretched cells near the object good convergence rates were obtained.

SCGS shows a convergence behaviour similar to SCGS/LS; these methods are satisfactory when almost square cells are used, but for highly stretched cells convergence factors tend to one. Therefore they are not robust enough for discretizations in curvilinear coordinates.

For many problems with very fine grids (even 512×512 -grids) the nonlinear multigrid method with the SCAL smoother produces level-independent convergence rates. The coupled smoothers are especially robust for low (< 400) Reynolds numbers. Then, large variations in the underrelaxation parameters are allowed, while reduction factors are almost the same. For higher Reynolds numbers small variations in the underrelaxation parameter (≈ 0.2) is allowed, but large variations may lead to convergence problems.

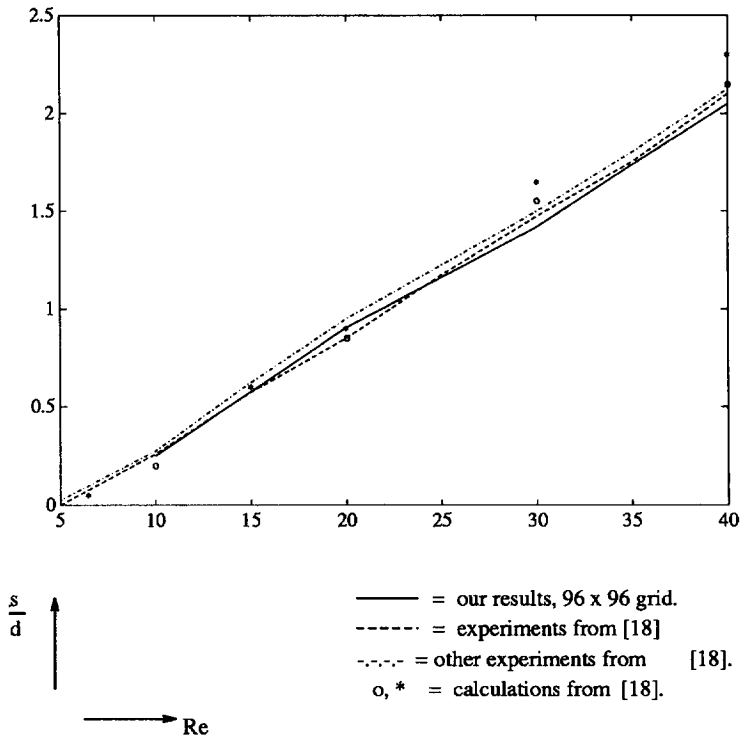


Figure 4.27: Ratio s/d plotted for several Reynolds numbers and compared to [18].

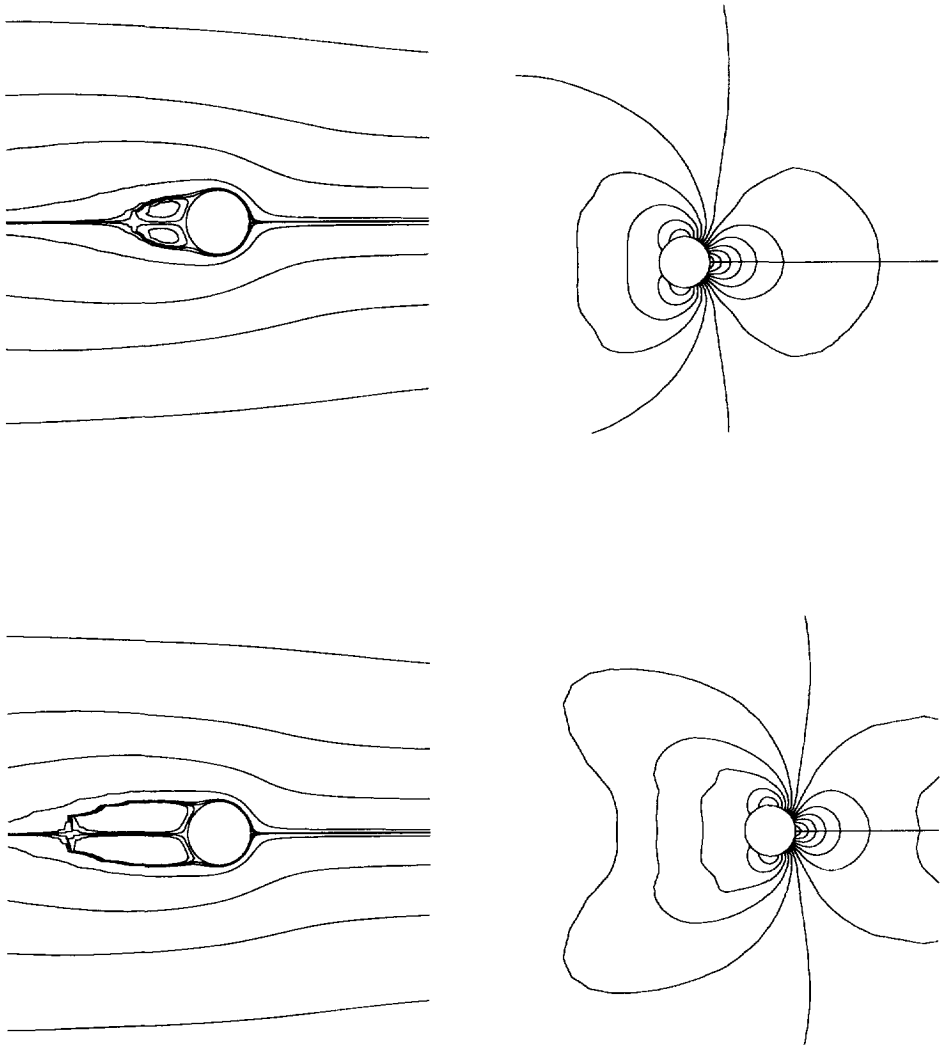


Figure 4.28: Streamlines and isobars for flow around a cylinder, $Re = 20$ and $Re = 40$.

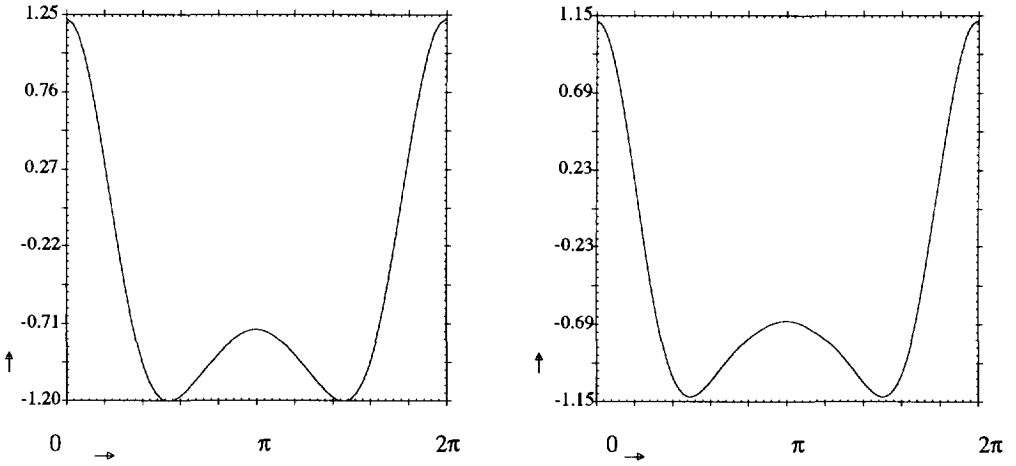


Figure 4.29: Pressure coefficient c_p on the surface of the cylinder, $Re = 20$ and $Re = 40$.

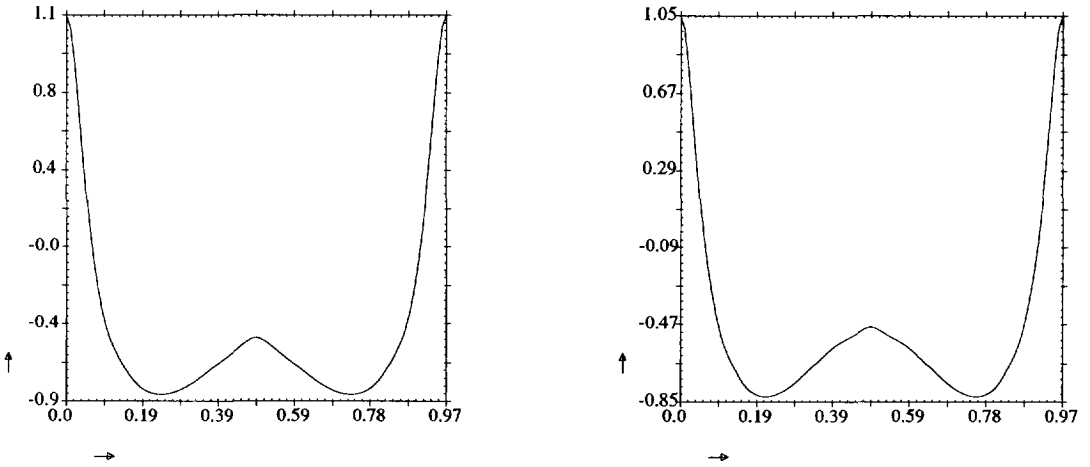


Figure 4.30: Pressure coefficients c_p on the surface of an ellipse, $Re = 40$ and $Re = 80$.

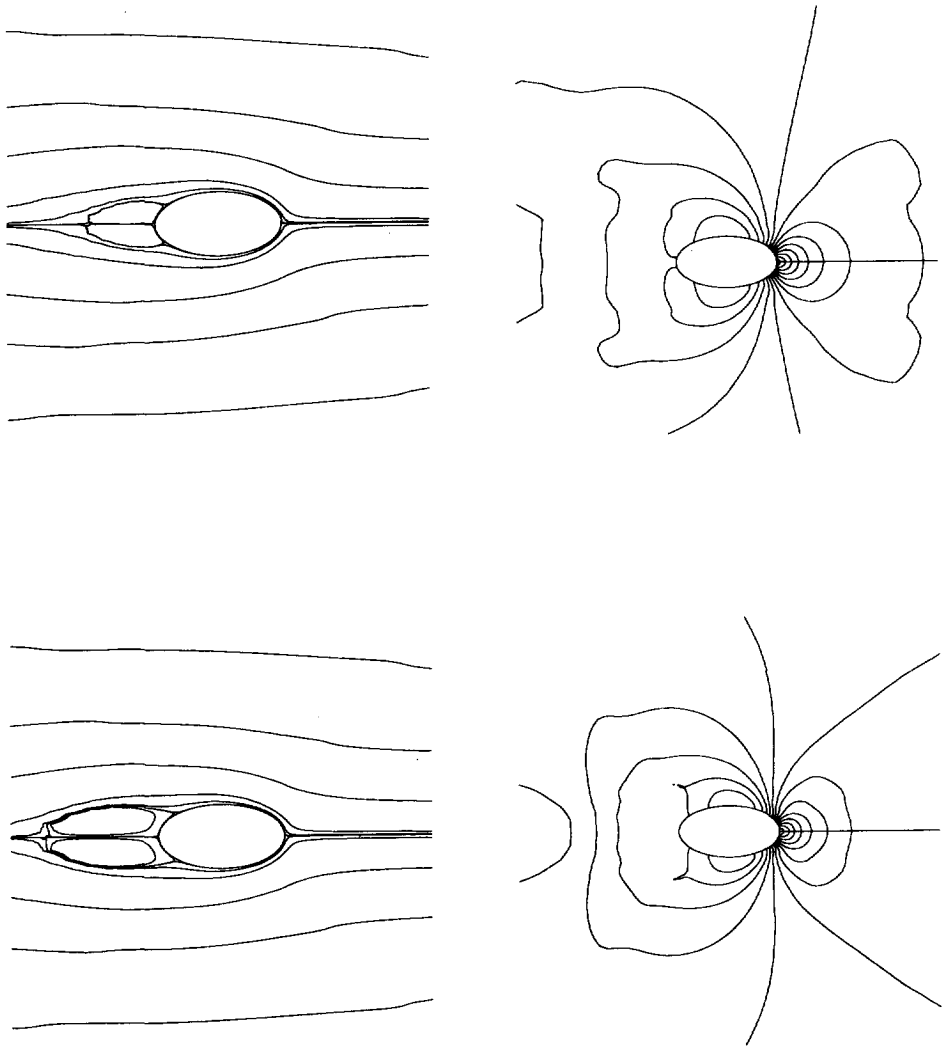


Figure 4.31: Streamlines and isobars for flow around an ellipse, $Re = 40$ and $Re = 80$.

5 Multigrid schemes for the time-dependent incompressible Navier-Stokes equations in general coordinates.

5.1 Introduction.

This chapter is also presented in [70].

There are a number of methods to solve time-dependent equations. Different approaches have been suggested to treat the "time-marching" in this type of equations. A standard approach is to solve the equations one time-step after another; this is called a time-marching scheme. The next time-step is tackled when a converged solution on the former time-level is obtained. Each time-step the equations are solved iteratively in space by, for example, a multigrid method. Examples of this approach are found in [35], [118] and [125]. However, the sequential nature of time-marching schemes does not lend itself well for implementation on parallel machines. Therefore new methods have been developed to solve time-dependent equations more efficiently on parallel machines. In 1984 Hackbusch ([34]) introduced two multigrid approaches called parabolic multigrid methods. Here the time-direction can be seen as one of the axes in a space-time grid. The multigrid procedure updates all unknowns in this space-time grid. In these approaches the equations are not solved, but smoothed time-step per time-step. Some components of these multigrid methods, like prolongation and restriction, can be done in parallel. In one of the methods the smoothing algorithm is sequential. The smoothing procedure on a new time-level uses updated values from the previous time-levels. In the second multigrid method proposed several time-steps can be smoothed simultaneously; "old" values from previous time-levels are then used. This smoothing method can be efficiently implemented on parallel machines. The computation on each time-level can be independently performed by a processor of a parallel computer. Results with these methods are described in [4] and [11] for the one-dimensional unsteady heat equation, in [12] and [42] for the unsteady incompressible Navier-Stokes equations in primitive variables, and in [60] for the unsteady incompressible Navier-Stokes equations in velocity-vorticity formulation. Parabolic multigrid methods or time-parallel multigrid methods for the incompressible Navier-Stokes equations are described in more detail in Section 5.3.

A different approach, based on the so-called waveform relaxation methods, is proposed by (amongst others) Vandewalle and described in detail in [105], [106]. In waveform relaxation methods, an approximation of an unknown in space is calculated along a time interval of interest consisting of a number of time-steps. Instead of updating scalars time-step by time-step functions in time are updated. Waveform relaxation schemes can be accelerated by multigrid. This solution method also lends itself well for parallel implementation, as is shown in [105] for several initial value and time-periodic problems. In Section 5.4 a multigrid waveform relaxation

algorithm is presented for the incompressible Navier-Stokes equations.

The multigrid methods discussed are investigated in this chapter for the incompressible Navier-Stokes equations in general coordinates. All methods will be based on the same smoother SCAL, presented in Section 4.2.2. SCAL has shown to be robust, showed good results solving the steady incompressible Navier-Stokes equations, and possesses good possibilities for efficient implementation on parallel machines. The algorithms are not implemented as a code for parallel machines. We investigate the performance on a Convex 3840 computer on one processor, but efficiency on parallel machines is considered.

5.2 The temporal discretization of the incompressible Navier-Stokes equations.

The incompressible Navier-Stokes equations have been presented in general coordinates in Chapter 3 (equations (3.6), (3.7), (3.8)). Due to the discretization for boundary-fitted grids, flows in domains of general shape can be calculated. For ease of reference the unsteady equations are repeated here:

$$U_{,\alpha}^{\alpha} = 0 \quad (5.1)$$

$$\frac{\partial \rho U^{\alpha}}{\partial t} + (\rho U^{\alpha} U^{\beta})_{,\beta} + (g^{\alpha\beta} p)_{,\beta} - \tau_{,\beta}^{\alpha\beta} = \rho f^{\alpha} \quad (5.2)$$

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

$$\tau^{\alpha\beta} = \mu(g^{\alpha\gamma} U_{,\gamma}^{\beta} + g^{\gamma\beta} U_{,\gamma}^{\alpha}) \quad (5.3)$$

where U^{α} are contravariant velocity components, ρ is density, p is pressure and μ is the viscosity coefficient. f^{α} represents a body force. Terms of the type $T_{,\beta}^{\alpha\beta}$ in (5.2) are given by (2.43). The spatial discretization of the equations on a staggered grid is explained in Chapter 3. The numerical solution of the unsteady incompressible Navier-Stokes equations involves the same problem as the solution for the steady equations: the absence of a pressure term in the continuity equation. For the solution of unsteady incompressible flow problems in the primitive formulation three different approaches have been developed. The uncoupled solution technique, in which the velocity components are calculated from the momentum equations separately from a "pressure-correction equation", a discretized form of the continuity equation combined with the momentum equations, is common use in practice. A second order accurate pressure-correction scheme is presented in [112] and is used for example in the ISNaS code, described in [61], [91]. Pressure-correction methods are very efficient for time-dependent problems, because one only needs to solve convection-diffusion type equations (usually with a large coefficient on the main diagonal coming from the time derivative) and a Poisson type equation for the pressure correction. In the steady case the advantage of uncoupled solution techniques is less pronounced than in the time-dependent case, because they require

assumptions about variables yet to be calculated and consequently need more iterations ([1], [84]).

In coupled solution methods, the discretized momentum and continuity equations are solved simultaneously. Coupled solution techniques have been investigated in the previous chapter for steady equations. For unsteady equations results can be found in [125]. The method adopted in this chapter is a continuation of the work from the previous chapter; a coupled solution method is used.

A third approach is adopted from compressible flow solvers, where for each unknown an explicit equation exists involving its time-derivative, for velocity components the momentum equations, for density the continuity equation, and for pressure and temperature the equation of state and the enthalpy equation. Hence, time-stepping is straightforward. This approach has been extended to incompressible flows by means of the so-called pseudo-compressibility method ([17]). An artificial time-dependent pressure term is added to the continuity equation. A disadvantage of this method is the appearance of a parameter, which is placed in front of the additional term and determines the convergence rate. For the time-dependent incompressible Navier-Stokes equations this approach has been used and investigated in [98], [54] and in many other papers. Accurate solution of time-dependent equations is difficult with pseudo-compressibility methods.

Time discretization with the so-called θ -method is done as follows:

$$U_{,\alpha}^{\alpha(n+1)} = 0 \quad (5.4)$$

$$\frac{\rho U^{\alpha(n+1)} - \rho U^{\alpha(n)}}{\Delta t} + \theta \{ (\rho U^{\alpha} U^{\beta})_{,\beta} - \tau_{,\beta}^{\alpha\beta} + (g^{\alpha\beta} p)_{,\beta} \}^{(n+1)} + (1 - \theta) \{ (\rho U^{\alpha} U^{\beta})_{,\beta} - \tau_{,\beta}^{\alpha\beta} + (g^{\alpha\beta} p)_{,\beta} \}^{(n)} = \theta \rho f^{\alpha(n+1)} + (1 - \theta) \rho f^{\alpha(n)} \quad (5.5)$$

which we abbreviate as:

$$U_{,\alpha}^{\alpha(n+1)} = 0 \quad (5.6)$$

$$\frac{\rho U^{\alpha(n+1)} - \rho U^{\alpha(n)}}{\Delta t} + \theta T_{,\beta}^{\alpha\beta(n+1)} + (1 - \theta) T_{,\beta}^{\alpha\beta(n)} = \theta \rho f^{\alpha(n+1)} + (1 - \theta) \rho f^{\alpha(n)} \quad (5.7)$$

For $\theta = 1$ we obtain the implicit Euler scheme, for $\theta = \frac{1}{2}$ the second order accurate Crank-Nicolson scheme. The time discretization with the second order accurate BDF(2)-scheme ([31]), also called B3-scheme consists of a second order approximation of the time derivative and a fully implicit treatment of all spatial derivatives, as follows,

$$U_{,\alpha}^{\alpha(n+1)} = 0 \quad (5.8)$$

$$\frac{3\rho U^{\alpha(n+1)} - 4\rho U^{\alpha(n)} + \rho U^{\alpha(n-1)}}{2\Delta t} + T_{,\beta}^{\alpha\beta(n+1)} = \rho f^{\alpha(n+1)} \quad (5.9)$$

The time-marching scheme. Each time-step the discretized equations are solved with the standard nonlinear multigrid method ([8], [35]). Details have been presented in Chapter 4. The algorithm can do multigrid V-, F- and W- iteration cycles. Prolongation and restriction operators are described in (4.5) to (4.18). The smoothing method is the Symmetric Coupled Alternating Lines (SCAL), described in Section 4.2.2.

In the steady case new values ($V^{1(n+1)}, V^{2(n+1)}, p^{(n+1)}$) are found with underrelaxation (4.36). For the unsteady case the additional diagonal term $\frac{1}{\Delta t}$ acts as an underrelaxation term, therefore no additional relaxation is needed. SCAL is a zebra-type smoother: first all odd (white) rows are visited, then all even (black) rows are visited. As already stated in Chapter 4, with special ordering strategies acceleration can be obtained on parallel computers.

Each time-step several multigrid iterations are performed until a termination criterion is met. Then, the solution is considered accurate enough and a next time-step is tackled. The termination criterion used is:

$$\|res_h^{(i)}\|_2 < 1 \times 10^{-3} \|rhs_0^{(i)}\|_2 \quad (5.10)$$

That is, a next time-step ($i + 1$) is started when the l_2 -norm of the residual after n iterations is less than 10^{-3} times the norm of the initial right-hand side. This appeared to be a good termination criterion ([115]), while it is scaling invariant and independent of the initial estimate.

An important aspect of time-marching schemes is that every time-step starts with a good initial approximation of the solution:

$$\mathbf{u}_h^{(i+1)} = \mathbf{u}_h^{(i)} \quad (5.11)$$

or:

$$\mathbf{u}_h^{(i+1)} = 2\mathbf{u}_h^{(i)} - \mathbf{u}_h^{(i-1)} \quad (5.12)$$

Comparing this coupled time-marching approach using multigrid to an uncoupled time-marching approach for the unsteady incompressible Navier-Stokes equations ([91]) where a pressure-correction method ([112]) is used and the momentum equations and pressure equations are solved with GMRES-type methods ([115]) it appeared that the coupled approach was slower than the uncoupled approach on a Convex 3840 using one processor.

The pressure-correction technique can be incorporated in a parabolic multigrid

method ([43]) for the unsteady case. However, to the author it is not clear at the moment how an uncoupled solution technique could be incorporated in a waveform algorithm.

Finally it is to be noted that this coupled solution technique is robust; arbitrarily large time-steps can be taken in arbitrary domains. Furthermore, all white rows as well as all black rows can be done in parallel, each on a single processor. Probably for many problems a white-black cell-by-cell smoother will be more efficient (well vectorizable), but certainly less robust !

5.3 The parabolic / time-parallel multigrid method.

The sequential process of solving equations time-step by time-step with a time-marching scheme makes algorithms less efficient on parallel machines. In the following multigrid schemes based on a paper by Hackbusch ([34]) the time-axis in a space-time grid is an axis along which solutions will be updated simultaneously for a number of time-steps. A convergence criterion must be satisfied for all unknowns in this grid, so when the criterion is met all solutions on all time-levels considered will be accurate enough. The implicit Euler scheme can be summarized as

$$\frac{\mathbf{u}_h^{(n+1)} - \mathbf{u}_h^{(n)}}{\Delta t} + T_h(\mathbf{u}_h^{(n+1)}) + \mathbf{f}_h^{(n+1)} \quad (5.13)$$

The time-steps $n_1, n_1 + 1, \dots, n_2$ will be updated simultaneously. Two different (sequential or parallel) smoothing algorithms are now described. They are of the following type:

Sequential smoothing algorithm:

begin

for iteration number $\nu = 1$ **step** 1 **until** ν_{max} **do** :

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** :

for space indices $i \in I^o \cup \partial I_N$ **do**:

$$\text{Solve } \left(\left(\frac{I}{\Delta t} + M_h \right) \mathbf{u}^{(n+1), \nu} \right)_i = (N_h \mathbf{u}^{(n+1), \nu-1})_i + \left(\frac{I}{\Delta t} \mathbf{u}^{(n), \nu} \right)_i + \mathbf{f}_i^{(n+1)}$$

enddo

end sequential smoothing.

Parallel smoothing algorithm:

```

begin
  for  $\nu = 1$  step 1 until  $\nu_{max}$  do :
    for  $n = n_1$  step 1 until  $n_2$  do :
      for  $i \in I^o \cup \partial I_N$  do:

        Solve  $((\frac{I}{\Delta t} + M_h)\mathbf{u}^{(n+1),\nu})_i = (N_h\mathbf{u}^{(n+1),\nu-1})_i + (\frac{I}{\Delta t}\mathbf{u}^{(n),\nu-1})_i + \mathbf{f}_i^{(n+1)}$ 

      enddo
    end parallel smoothing.

```

Note that in the latter case all time-steps can be done in parallel.

Indicating the coarse grid by a subscript H , the nonlinear time-parallel two-grid method is given by:

Nonlinear time-parallel two-grid algorithm:

begin algorithm

for number of iterations $\nu = 1$ **step** 1 **until** ν_{max} **do** :

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** :

for spatial indices $i \in I^\circ \cup \partial I_N$ **do**:

- Apply a pre-smoothing iteration (*sequential or parallel*)

enddo

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** *parallel*:

- Compute residual:

$$\mathbf{r}^{(n+1)} = (\mathbf{f}_h^{(n+1)} + \frac{I}{\Delta t} \mathbf{u}_h^{(n)}) - (\frac{I}{\Delta t} + T_h) \mathbf{u}_h^{(n+1)} \quad (5.14)$$

enddo

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** *parallel*:

- Choose $\tilde{\mathbf{u}}_H^{(n+1)}$.

- Apply Restriction as (4.5) $\tilde{R}^H : \mathbf{u}_H^{(n)} = \tilde{R}^H \mathbf{u}_h^{(n)}$, $\tilde{R}^H : \tilde{\mathbf{u}}_H^{(n)} = \tilde{R}^H \mathbf{u}_h^{(n)}$.

- Apply Restriction R^H as (4.6) to $\mathbf{r}^{(n+1)}$.

enddo

for $n = n_1$ **step** 1 **until** n_2 **do** :

- Solve the coarse grid equation for $\mathbf{u}_H^{(n+1)}$.

(5.15)

$$\frac{\mathbf{u}_H^{(n+1)} - \mathbf{u}_H^{(n)}}{\Delta t} + T_H \mathbf{u}_H^{(n+1)} = \mathbf{f}_H^{(n+1)}$$

$$= s_H R^H(\mathbf{r}^{(n+1)}) + (\frac{I}{\Delta t} + T_H) \tilde{\mathbf{u}}_H^{(n+1)} - \frac{I}{\Delta t} \tilde{\mathbf{u}}_H^{(n)} \quad (5.16)$$

enddo

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** *parallel*:

- Prolongation:

$$\mathbf{u}_h^{(n+1)} = \mathbf{u}_h^{(n+1)} + \frac{1}{s_H} P^h(\mathbf{u}_H^{(n+1)} - \tilde{\mathbf{u}}_H^{(n+1)}) \quad (5.17)$$

enddo

for time-levels $n = n_1$ **step** 1 **until** n_2 **do** :

for number of iterations $\nu = 1$ **step** 1 **until** ν_{max} **do** :

for spatial indices $i \in I^\circ \cup \partial I_N$ **do**:

- Apply a post-smoothing iteration (*sequential or parallel*)

enddo

end nonlinear time-parallel two-grid algorithm.

Again the coarse grid equation can be solved in a similar way, i.e. by introducing a third coarser grid and by executing a number of two-grid cycles, etcetera. The introduction of a sequence of grids leads to the time-parallel multigrid algorithm, also called parabolic multigrid algorithm with different iteration cycles. Here again V-, F- and W-cycles are implemented. Note that different stages in the multigrid algorithm can be performed in parallel in time direction. The restriction and prolongation operators are only spatial operators. In [34] it is stated that coarsening in time does not lead to an efficient algorithm. Errors which are smooth in spatial direction, but nonsmooth in time are then not improved by smoothing, because the defects are smooth in space. Furthermore, the coarse grid correction does not provide a good correction because of the coarser time-step. Therefore we do not apply grid coarsening in time.

In [4] results are obtained efficiently with a parabolic multigrid method on a transputer system. A sequential smoother of Gauss-Seidel type is compared to a parallel smoother of Jacobi type for a parabolic differential equation. It was found that a good speed-up was obtained for the sequential smoother for many processors in a model problem, while the good speed-up with the parallel smoothing method was limited to a small number of processors.

In [12] and [42] a time-parallel version of the SIMPLE algorithm ([71]) is used to solve the incompressible Navier-Stokes equations on a transputer system. Good efficiency is obtained for an unsteady driven cavity problem. In [60] a vorticity-velocity formulation is applied to the incompressible Navier-Stokes equations. The coarse grid equation is set up a little bit differently from [42]. The smoother is a so-called Group Explicit Iterative method (GEI), and good convergence and CPU time results for a small number of processors (varying from 1 to 4) are presented for the steady driven cavity problem solved with unsteady equations.

In this chapter the incompressible Navier-Stokes equations in *general coordinates* will be smoothed in the time-parallel multigrid method with the *sequential* SCAL smoother as smoothing method on all time-levels. The implementation of a robust and efficient *parallel* smoothing variant in time is not trivial and under investigation.

In general coordinate systems there is a wide choice in time-dependent test problems. The computer code is restricted at present to time-dependent Dirichlet problems. In these time-parallel methods (5.11) or (5.12) can not be used to obtain a starting solution. A good initial approximation on each time-level will be produced with "nested iteration". Implemented are nested iteration V- and F-cycles.

5.4 The multigrid waveform relaxation method.

A smoothing algorithm is the most time consuming part of a multigrid method, and it will be interesting to execute this part efficiently on a parallel machine. With a waveform relaxation method communication costs are probably lower than for the other relaxation schemes. Waveform relaxation methods update an unknown in a

grid-point along a time interval consisting of a number of time-steps. If an unknown in space is assigned to a processor, then during the smoothing of that unknown in time there is no need for a lot of communication with other processors, which can be costly. Originally, waveform methods were developed as relaxation schemes in simulation techniques of electrical network problems ([120]). Waveform relaxation methods are found to have qualitatively the same convergence behaviour as basic iterative methods. High frequency errors are smoothed quickly, while low frequency errors are damped slowly. Therefore waveform relaxation methods are also suited for a multigrid acceleration. The nonlinear multigrid waveform algorithm differs from the nonlinear time-parallel method, presented in the previous section, only in the smoothing algorithm. Again the restriction and prolongation operators, which are spatial operators (no coarsening is applied to time-steps), can be done in parallel. In [105] the multigrid waveform algorithm is found to perform very well on several nonlinear initial boundary value and time-periodic parabolic partial differential equations on a parallel machine. The smoother used is a white/black Gauss-Seidel waveform smoother. A comparison between the standard time-marching scheme and the multigrid waveform method even on a single processor shows a competitive performance for the multigrid waveform method in many cases considered ([106]). A multigrid waveform relaxation method to solve the incompressible Navier-Stokes equations has not been found in literature yet. Here it is presented with the SCAL waveform relaxation method as smoother:

First all horizontal "white" rows are updated (which can be done in parallel on a parallel machine) on all time-levels t_{n_1}, \dots, t_{n_2} . Then, all horizontal "black" rows are updated on all time-levels. After a waveform sweep along horizontal rows a sweep along vertical rows will be applied.

Since many time-levels are involved the horizontal and vertical rows (in space) are in fact vertical time-blocks.

For an implicit Euler scheme the waveform smoother looks like:

Waveform smoothing algorithm:

```

begin
  for number of iterations  $\nu = 1$  step 1 until  $\nu_{max}$  do :
    for spatial indices  $i \in I^o \cup \partial I_N$  do:
      for time-indices  $n = n_1$  step 1 until  $n_2$  do :

Solve  $\left(\frac{I}{\Delta t} + M_h\right) \mathbf{u}^{(n+1), \nu}_i = (N_h \mathbf{u}^{(n+1), \nu-1})_i + \left(\frac{I}{\Delta t} \mathbf{u}^{(n), \nu}\right)_i + \mathbf{f}_i^{(n+1)}$ 

      enddo
    end waveform smoothing.

```

A disadvantage of the waveform method chosen is that extra storage of operators and unknowns is required. A remedy to avoid too much storage is to use small values of $n_2 - n_1$. Waveform relaxation methods are dynamic iteration methods; multigrid can be applied before the time discretization. This means that in the

resulting set of equations corresponding to a place in the spatial grid each equation can be discretized in time independently. Local time-step sizes can be chosen depending on the smoothness of the solution in the spatial grid node. So it is possible to implement a local adaptive time stepping strategy in multigrid waveform relaxation methods. However, data structure problems and implementation problems for operators that handle the data structures can be foreseen. In parabolic multigrid methods it is possible to apply global adaptive time stepping. For every space unknown the same (adaptive) time step can be applied. A drawback compared to sequential adaptive time stepping is that when a time step is rejected a whole interval $n_1 - n_2$ of time levels has to be restarted. A lot of work is then lost. Adaptive time stepping is not done here.

5.5 Results.

An analytic test problem. In order to investigate the accuracy of time integration schemes it is useful to construct a test problem for which the analytical solution $\mathbf{u}(t) = (u^1, u^2, p)$ is known. The error $\Delta \mathbf{u}$ is defined by:

$$\Delta \mathbf{u} = \mathbf{u}(t) - \mathbf{u}_h^{(n)} \quad (5.18)$$

The contravariant vector V^α found after n time integration steps on a domain with spatial grid-size h is transformed to the Cartesian vector $\mathbf{u}_h^{(n)}$. Assuming density ρ and viscosity μ constant an exact solution of the incompressible Navier-Stokes equations is defined by:

$$\begin{aligned} u^1 &= \sin(t)\sin(x)\sin(y) \\ u^2 &= \sin(t)\cos(x)\cos(y) \\ p &= \sin(t)(\sin(x) + \cos(y)) \end{aligned} \quad (5.19)$$

Substitution of (5.19) in (5.2) defines the right-hand side f^α of (5.2). Furthermore, u^1 and u^2 from (5.19) are defined as Dirichlet boundary conditions. We choose $\rho = 1$, $\mu = 0.02$. Streamlines and isobars of the solution are presented in Figure 5.1. A square domain $(0, \pi) \times (0, \pi)$ is discretized with 80×80 cells. The corresponding Reynolds number based on the maximum velocity is 50π . The l_2 -norm of the u^1 -error component is defined by:

$$\|\Delta u^1\| = \sqrt{\frac{1}{m} \sum_{i=1}^m (u_i^1 - u_{i,h}^{1(n)})^2} \quad (5.20)$$

where m represents the number of unknowns. A similar expression is found for $\|\Delta u^2\|$.

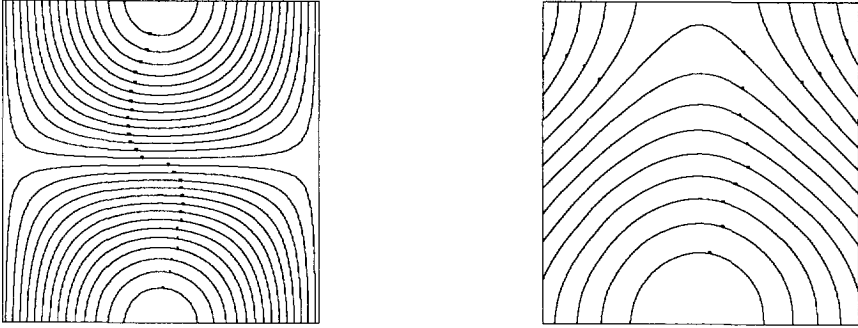


Figure 5.1: Streamlines and isobars of the solution of the analytical test example.

$\|\Delta p\|$ is defined by:

$$\|\Delta p\| = \sqrt{\frac{1}{m} \sum_{i=1}^m \left\{ \left(p_i - \sum_{i=1}^m \frac{p_i}{m} \right) - \left(p_{i,h}^{(n)} - \sum_{i=1}^m \frac{p_{i,h}^{(n)}}{m} \right) \right\}^2} \quad (5.21)$$

The determination of the error in the pressure (taking the difference between the solution and the mean value of the pressure) avoids extra errors coming from the fact that the pressure is determined up to a constant.

In Table 5.1 the error in the time discretization found with the time-marching scheme (very similar results are found with the other multigrid schemes, of course) is presented for three schemes, the first order accurate implicit Euler scheme, the second order accurate Crank-Nicolson scheme and the second order accurate BDF(2) scheme. BDF(2) is started with an implicit Euler step. Time discretization is performed from $t_0 = 0$ to $t_{end} = 1.5$ with respectively 5, 10, 20 and 40 time-steps ($dt = 0.3, \dots, 0.0375$). From Table 5.1 we can observe that, before the spatial discretization error is reached (here for $n = 40$) the different schemes are indeed first and second order accurate in time (for example, the error is a factor 4 smaller for $n = 10$ compared to $n = 5$ for BDF(2) and $\theta = \frac{1}{2}$). Because BDF(2) shows results similar to the Crank-Nicolson scheme, and because BDF(2) is an L-stable scheme, while the Crank-Nicolson scheme is A-stable but not L-stable ([37]) we will continue with the first order Euler scheme and the second order BDF(2)-scheme. (BDF(2) requires extra storage of one solution from a previous time-level, but that is not a problem here, because the parabolic and waveform relaxation multigrid iteration schemes also need these solutions to be stored.) The average number of iterations

per time-step to reach the desired accuracy specified by (5.10) on the 80×80 -grid for the BDF(2) scheme is presented in Table 5.2. The sequential computational complexity of the marching, parabolic and waveform schemes is about the same. For the time-marching scheme we compared the V(0,1)-cycle (meaning V-cycle with 0 pre- and 1 post-smoothing iteration) with V(0,2), F(0,1) and F(0,2). V(0,1) appeared to be the fastest cycle for these unsteady problems, followed by V(0,2). The F-cycles are much more expensive and therefore not taken into account here for the other multigrid schemes. For the parabolic and waveform multigrid algorithms we compared V(0,1) and V(0,2) with FV(0,1) and FV(0,2) (meaning a V-cycle scheme, with an F-cycle nested iteration). Again for this test problem the CPU time for the V(0,1)-cycle was smallest, followed by FV(0,1) and V(0,2). From Table 5.2 we also observe a reduction, due to the good starting solution on each time-level in the number of iterations for the time-marching scheme when the number of time-levels equals 20.

Clearly the smoothing algorithm, though robust, is fairly expensive on a vector computer. An iteration of the time-marching scheme was indeed as expensive as an iteration of the other schemes. It is satisfying to observe that a more or less equal number of iterations is needed to satisfy the termination criterion for the parabolic multigrid scheme and the multigrid waveform relaxation scheme. Also approximations of the convergence factor are investigated:

$$\nu_n^{(i)} \equiv \left\| \frac{res_n^{(i)}}{res_{n-2}^{(i)}} \right\|^{1/2} \quad \text{for large } n \quad (5.22)$$

For i always the last time-level is chosen. There the approximate convergence factor was generally not smaller than for earlier time-steps. In many cases $\nu_n^{(i)}$ is found to be approximately constant (for $n \cong 20$), in which case we have found the asymptotic convergence factor. The size of the grid is chosen to be 16×16 , 32×32 and 64×64 , the number of time-levels is 40. Table 5.3 presents the approximate convergence factors for the V(0,1)-cycle. Again the multigrid waveform relaxation method shows a convergence factor for a large number of time-steps (40 is this case) almost as good as the other two methods.

Table 5.1: The l_2 -norm of the errors for the test problem on a 80×80 -grid, with (n) the number of time-steps between $t_0 = 0$ and $t_{end} = 1.5$.

(n)	Δu	Schemes		
		$\theta = 1$	$\theta = \frac{1}{2}$	BDF(2)
5	Δu	$.1337 \times 10^{-2}$	$.1099 \times 10^{-3}$	$.2940 \times 10^{-3}$
	Δv	$.7069 \times 10^{-3}$	$.6332 \times 10^{-4}$	$.1604 \times 10^{-3}$
	Δp	$.8332 \times 10^{-1}$	$.1169 \times 10^{-1}$	$.7574 \times 10^{-2}$
10	Δu	$.6777 \times 10^{-3}$	$.2268 \times 10^{-4}$	$.7048 \times 10^{-4}$
	Δv	$.3562 \times 10^{-3}$	$.1365 \times 10^{-4}$	$.3919 \times 10^{-4}$
	Δp	$.4213 \times 10^{-1}$	$.2398 \times 10^{-2}$	$.1666 \times 10^{-2}$
20	Δu	$.3354 \times 10^{-3}$	$.6075 \times 10^{-5}$	$.1772 \times 10^{-4}$
	Δv	$.1815 \times 10^{-3}$	$.4457 \times 10^{-5}$	$.1065 \times 10^{-4}$
	Δp	$.2238 \times 10^{-1}$	$.6297 \times 10^{-3}$	$.4989 \times 10^{-3}$
40	Δu	$.1719 \times 10^{-3}$	$.3035 \times 10^{-5}$	$.5520 \times 10^{-5}$
	Δv	$.9018 \times 10^{-4}$	$.2859 \times 10^{-5}$	$.4106 \times 10^{-5}$
	Δp	$.1067 \times 10^{-1}$	$.3752 \times 10^{-3}$	$.2907 \times 10^{-3}$

Table 5.2: The average number of iterations per time-step to satisfy the termination criterion for different multigrid schemes (- = not considered).

(n)	cycle	BDF(2)		
		Marching	Parabolic	Waveform
5	V(0,1)	5	5	5.5
	V(0,2)	4	4	4
	F(0,1)	4.5	-	-
	F(0,2)	4	-	-
	FV(0,1)	-	4	4
	FV(0,2)	-	2	2.5
10	V(0,1)	5	5	5.5
	V(0,2)	4	4	4
	F(0,1)	4	-	-
	F(0,2)	3.5	-	-
	FV(0,1)	-	4	4
	FV(0,2)	-	2	2.5
20	V(0,1)	4	5	5.5
	V(0,2)	3	3	3.5
	F(0,1)	3	-	-
	F(0,2)	2.5	-	-
	FV(0,1)	-	4	4.5
	FV(0,2)	-	2	2.5

Table 5.3: Approximations of the convergence factor $\nu_{20}^{(40)}$ for V(0,1) for the analytical test problem on three grids.

Grid	Marching	Parabolic	Waveform
16 × 16	0.165	0.169	0.198
32 × 32	0.177	0.193	0.215
64 × 64	0.177	0.195	0.235

An unsteady skewed driven cavity. Another test example used here is the unsteady flow in a driven cavity, which has the shape of a parallelogram, with skewness angle 45° . Investigations and reference results for the cavity studied here are described for the steady case in [23] and in Section 3.4. First we show some possibilities of our approach. An unsteady cavity flow at Reynolds number 1000 is presented on a 64×64 -grid. The top wall moves with a speed depending on the time as follows:

$$\begin{aligned} t \leq 100 : & \quad u^1 = 1 \\ 100 < t < 200 : & \quad u^1 = -\cos\left(\frac{\pi t}{100}\right) \\ t \geq 200 : & \quad u^1 = -1 \end{aligned} \tag{5.23}$$

Figure 5.2 shows streamlines of flow patterns at certain times from one steady state to an other obtained with the BDF(2)-scheme. It is clear that most variation takes place from $t = 150$ to $t = 160$, when the flow direction is reversed. We therefore chose variable time-steps: $dt^{(1)} = 100$, $dt^{(i)} = 10$ for $i = 2, \dots, 6$ (to $t = 150$); $dt^{(i)} = 1$ for $i = 7, \dots, 16$ (to $t = 160$) and $dt^{(i)} = 10$ for $i > 16$.

The test problem investigated numerically is described in [43]. There, for a square cavity the topwall moves with velocity $u^1 = \sin(t)$. Here, the topwall of the skewed cavity moves with velocity $u = \sin(t)$ from $t_0 = 0$ to $t_{end} = 1.5$ ($\cong \pi/2$). The number of time-steps (i) is 10, 20 and 40. For this example the behaviour of an F-cycle for a higher Reynolds number is investigated: $Re = 1000$. We obtained the average number of iterations to satisfy termination criterion (5.10) and the convergence factors from (5.22) with $n = 20$ for an F(0,1)-cycle with starting solution (5.11) for the time-marching scheme, and for the other schemes for an FF(0,1)-cycle. Again for the parabolic and waveform relaxation multigrid schemes the *sequential* SCAL smoother is used. These results are presented in Table 5.4. Similar results are

Table 5.4: Average number of iterations per time-step to satisfy the termination criterion and approximations of the convergence factor $\nu_{20}^{(i)}$ for the skewed driven cavity problem on a 64×64 - grid.

(i)	Marching		Parabolic		Waveform	
	# it.	$\nu_{20}^{(i)}$	# it.	$\nu_{20}^{(i)}$	# it.	$\nu_{20}^{(i)}$
10	3.0	0.317	3.5	0.345	3.7	0.363
20	2.3	0.295	3.0	0.324	3.8	0.357
40	2.0	0.268	3.1	0.300	3.5	0.356

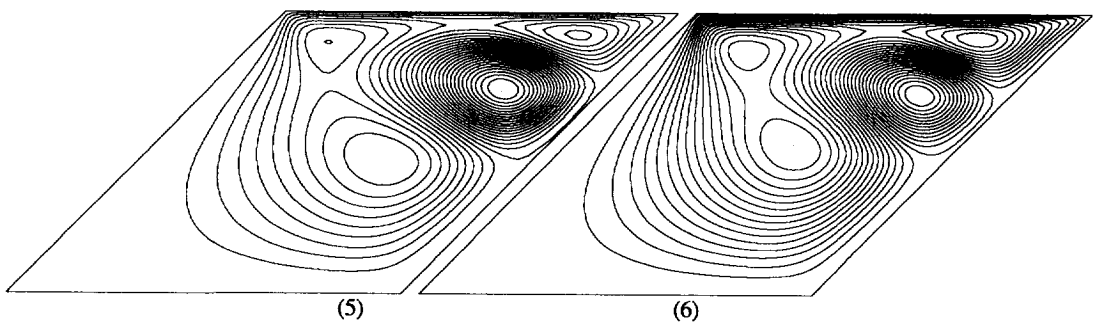
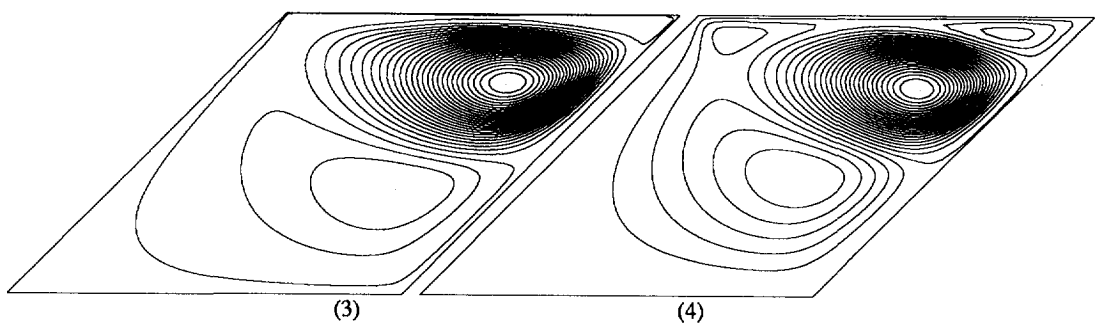
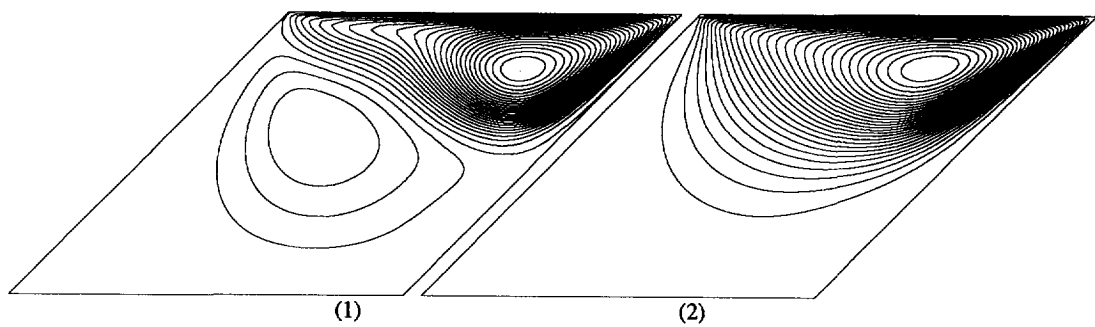
found for the implicit Euler scheme. Again the results from Table 5.4 are satisfactory. The reduction factors for the parabolic and waveform relaxation schemes

do not depend on the number of time-steps, neither does the number of iterations to satisfy (5.10). The number of iterations needed to satisfy (5.10) in Table 5.4 is lower than in Table 5.2, while the convergence factors are higher in Table 5.3. This is due to the fact that with the FF(0,1)-cycle very good first reduction factors are obtained.

5.6 Conclusions.

Three multigrid methods for the time-dependent incompressible Navier-Stokes equations in general coordinates have been compared for several test problems, namely a time-marching scheme, a parabolic multigrid method and a multigrid waveform relaxation method. For all methods an essential part of the algorithm, the smoother, was based on the same method. Contrary to solution methods for the steady incompressible Navier-Stokes equations, no additional underrelaxation is required to solve the unsteady equations. The smoother is a robust method, though fairly expensive (on a vector computer). Three time discretization schemes have been compared, the implicit Euler scheme, the Crank-Nicolson scheme and the BDF(2) scheme. The latter scheme showed second order accuracy, while it satisfies better stability conditions than the Crank-Nicolson scheme.

Furthermore, the storage requirement of the time-marching scheme is lowest, for the multigrid waveform relaxation method it is highest. Approximate convergence factors and the average number of iterations per time-step to satisfy a termination criterion are compared. Satisfactory results were obtained for all three methods. Their sequential computational complexity was about the same, but they differ markedly in their parallelization potential. It will be interesting to study the behaviour of the methods and of a time-parallel variant of the SCAL smoother on a parallel machine.



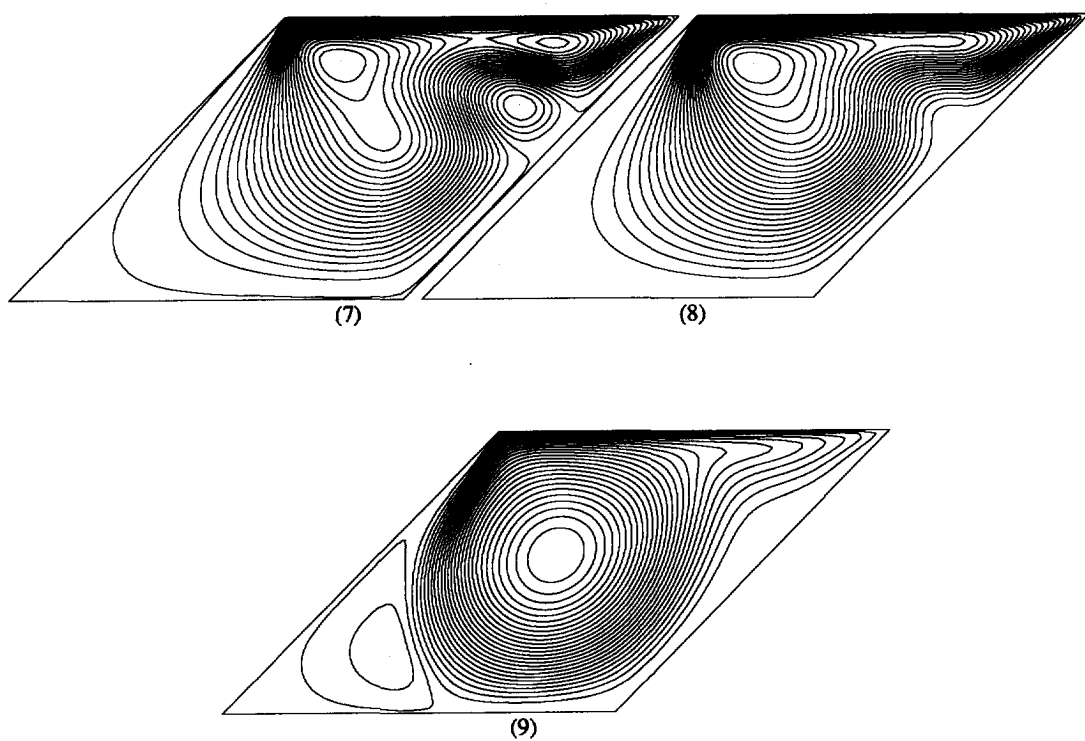


Figure 5.2: Flow patterns of the unsteady skewed driven cavity test problem at time-levels: 1) $t = 100$, 2) $t = 130$, 3) $t = 152$, 4) $t = 153$, 5) $t = 154$, 6) $t = 155$, 7) $t = 156$, 8) $t = 157$ and 9) $t = 200$.

6 Concluding remarks, suggestions for future research.

Several aspects of this thesis will be reconsidered here and suggestions for future research will be given.

The discretization. A two-dimensional spatial discretization in general coordinates of the incompressible Navier-Stokes equations has been presented. The discretization is coordinate invariant. The equations are discretized on a staggered grid. The discretization is restricted at the moment to more or less smooth grids. Grids with kinks in the grid lines may cause troubles, due to the occurrence of Christoffel symbols.

As already mentioned, research is being done ([116]) to avoid interpolations of geometric quantities over cell faces by using appropriate one-sided approximations at places where a geometric quantity is discontinuous. The results could be very interesting.

The convection tensor is discretized with a hybrid difference technique. Benchmark solutions show that our method gives a reliable discretization and produces accurate solutions for the two-dimensional incompressible Navier-Stokes equations on sufficiently fine grids. The implementation of a defect correction technique resulting in a second order spatial discretization could also be very interesting. The aspects of the cost of this technique and the accuracy on coarser (for example 64×64) grids, instead of the very fine grids used in this thesis, is worth further research.

Test problems. A number of test problems with stretching, non-orthogonality and curvature of grid cells have been proposed and computed. Some problems are hard to solve in a single block approach. The results have been written down, so that methods based on multiblock approaches can use and compare the flow results. For the hole-pressure problem, for example, it is difficult to obtain finer satisfactory grids; the generation of structured grids in such a complex domain is very difficult. With a multidomain approach Cartesian grids can be used when the domain is split up in two domains. Then, with finer grids the results from [53] could be approximated even better. The problems are also interesting as three-dimensional problems.

The multigrid method. Three smoothers in a standard nonlinear multigrid solution algorithm have been investigated for robustness and efficiency for the two-dimensional discretized steady incompressible Navier-Stokes equations in general coordinates. The coupled line smoother SCAL performs best and is robust. Smoothing methods based on line solvers lead to the solution of band matrices. They need to be solved with high accuracy, because the continuity equation is involved and mass cannot be lost. Direct solution methods are fairly expensive. CPU times could be improved when vectorization strategies or parallel statements would be included

in the code. This means that the solution algorithm for the band matrix must be rewritten in order to solve each "black" or "white" line in the zebra smoother simultaneously.

It would be interesting to investigate the need for the thirteen element band in SCAL for discretizations in curvilinear coordinates. In Cartesian coordinates the band contains nine elements and the solution process is then cheaper. Implicit treatment of a nine element band is probably sufficient for discretizations in curvilinear coordinates, (handling the remaining co-diagonals in explicit manner) and could also result in satisfactory reduction factors.

Benefit can also be obtained when the structure of the band is taken into account. For example, a matrix row coming from the continuity equation contains at most four non-zero entries, instead of the maximum of thirteen elements for one of the momentum equations and nine for the other.

For the unsteady incompressible Navier-Stokes equations we investigated the multigrid waveform relaxation method on a vector machine using one processor. This method is of benefit on parallel machines. In order to judge the multigrid waveform relaxation method research on complicated systems of partial differential equations, like the incompressible Navier-Stokes equations, is needed on parallel machines.

Three-dimensional problems. A continuation to three-dimensional problems would lead to plane smoothers instead of line smoothers. Plane smoothers are expensive. The resulting matrix system for a plane would be a sparse band. A direct solver for this system is very expensive; iterative solvers are needed. Again efficiency can be enhanced with parallel and vector processing, because several planes can be updated simultaneously.

Other multigrid methods. A cheap and robust solver for steady incompressible Navier-Stokes equations based on a standard multigrid method is difficult to obtain. Therefore research on non-standard multigrid methods based on semi-coarsening, in which the coarse grid correction is made more robust while the methods still have $O(m)$ complexity for problems with m unknowns (for example [59]) is really important and should get attention. With robust coarse grid corrections the smoother can probably be less robust and therefore more efficient and parallelizable. If instead of line solvers point solvers can be used, while robustness is not impaired, then this would be a very interesting development for multigrid methods.

We restricted ourselves to the investigation of coupled smoothers. In this class the investigation on robustness for a white/black cell-by-cell (well vectorizable) smoother in a semi-coarsened multigrid method is interesting. Of course, uncoupled smoothers are also interesting to investigate in these semi-coarsened methods.

A An example of an unstable discretization.

We give an example where method 3 of Chapter 2 leads to an unstable discretization. Choose $w_1^{(1)} = 1$, $w_2^{(1)} = 0$. Choose as transformation

$$\begin{aligned}\xi^1 &= x^2 \\ \xi^2 &= -x^1\end{aligned}\tag{A.1}$$

With (2.11) for $W_\alpha^{(1)}$ is found:

$$W_1^{(1)} = 0, \quad W_2^{(1)} = -1\tag{A.2}$$

Hence from (2.90) follows for a V^1 -control volume,

$$\int_V W_\alpha^{(1)} T_{,\beta}^{\alpha\beta} dV = \oint_S W_2^{(1)} T^{2\beta} dS_\beta\tag{A.3}$$

The right-hand side of (A.3) leads to:

$$\oint_S W_2^{(1)} T^{2\beta} dS_\beta = \{-T^{21}|_{(1,0)} + T^{21}|_{(-1,0)}\} \delta\xi^2 + \{-T^{22}|_{(0,1)} + T^{22}|_{(0,-1)}\} \delta\xi^1\tag{A.4}$$

with in the case of the Stokes equations (introduced in Chapter 3) and (A.1):

$$T^{21} = \mu(U_{,1}^2 + U_{,2}^1); \quad T^{22} = -p + 2\mu U_{,2}^2$$

With linear interpolations we find (in triad notation, see Figure 3.2):

$$\begin{aligned}T_{(1,0)}^{21} \approx & \frac{\mu}{2} \left\{ \frac{1}{2}(U_{i+1,j}^2 + U_{i+1,j+1}^2) - \frac{1}{2}(U_{i-1,j}^2 + U_{i+1,j+1}^2) + \right. \\ & \left. \frac{1}{2}(U_{i,j+1}^1 + U_{i+1,j+1}^1) - \frac{1}{2}(U_{i,j-1}^1 + U_{i+1,j-1}^1) \right\}\end{aligned}\tag{A.5}$$

and

$$\begin{aligned}T_{(0,1)}^{22} \approx & -\frac{1}{4}(p_{i,j} + p_{i,j+1} + p_{i-1,j+1} + p_{i-1,j}) + \\ & \mu \left\{ \frac{1}{2}(U_{i,j+2}^2 + U_{i-1,j+2}^2) - \frac{1}{2}(U_{i,j}^2 + U_{i-1,j+2}^2) \right\}\end{aligned}\tag{A.6}$$

$T_{(-1,0)}^{21}$ and $T_{(0,-1)}^{22}$ are treated in the same way. The molecule for this momentum equation is shown in Figure A.1. Row j does not contain U^1 -variables. Therefore a non-constant U^1 -field: $U_{i-1/2,j}^1 = (-1)^j$ produces a zero residual for this equation, which is to be interpreted as the U^1 -momentum equation. Zebra-type oscillations are to be expected in the numerical solution. Therefore this discretization may be called unstable (in a loose sense).

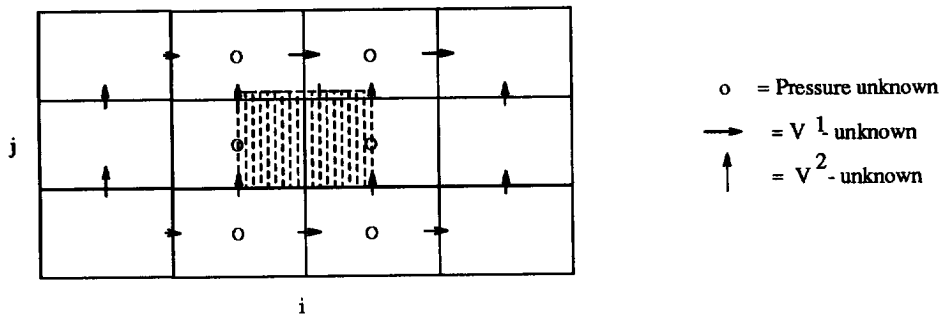


Figure A.1: The molecule for an unstable discretization.

B A discretization based on method 4 of Chapter 2.

This appendix is also presented in [113].

Here the formulae for all spatial terms are derived separately following method 4 of Chapter 2 (i.e. equation (2.98)) and we will arrive at a discretization directly in the physical domain, derived in [88] from a different formulation, namely applying the conservation laws in physical space using Gibbs' vector notation. First we introduce the following notation:

$$[\gamma_\alpha]^{i,j} = a_{\beta(i,j)}^{(\gamma)} a_{(\alpha)}^\beta \quad (\text{B.1})$$

where one of the two base vectors is determined at a fixed grid point (i, j) , and the location where the second base vector is evaluated remains a function of position. Substitution of (B.1) into (2.98) leads to

$$\int_\Omega W_\alpha^{(\gamma)} T_{,\beta}^{\alpha\beta} d\Omega \cong (\sqrt{g} [\gamma_\alpha]^{i,j} T^{\alpha 1})_{i-1/2,j}^{i+1/2,j} \delta\xi^2 + (\sqrt{g} [\gamma_\alpha]^{i,j} T^{\alpha 2})_{i,j-1/2}^{i,j+1/2} \delta\xi^1 \quad (\text{B.2})$$

The convection term. Hence, the convection term for a V^1 -cell is approximated by

$$\begin{aligned} \int_\Omega W_\alpha^{(1)} (\rho U^\alpha U^\beta)_{,\beta} d\Omega &\approx \frac{\rho}{\sqrt{g}} \left(\left[\begin{matrix} 1 \\ 1 \end{matrix} \right]^{0,0} V^1 V^{1(n)} + \left[\begin{matrix} 1 \\ 2 \end{matrix} \right]^{0,0} V^1 V^{2(n)} \right) \Big|_{(-1,0)}^{(1,0)} \delta\xi^2 + \\ &\frac{\rho}{\sqrt{g}} \left(\left[\begin{matrix} 1 \\ 1 \end{matrix} \right]^{0,0} V^{1(n)} V^2 + \left[\begin{matrix} 1 \\ 2 \end{matrix} \right]^{0,0} V^2 V^{2(n)} \right) \Big|_{(0,-1)}^{(0,1)} \delta\xi^1 \quad (\text{B.3}) \end{aligned}$$

where we have linearized, using superscript (n) to indicate a known approximation of unknown V^α . The shape of the molecule is exactly the same as the one found in Chapter 3 (Figure 3.7b). The convection term for a V^2 -cell is approximated in similar fashion by

$$\begin{aligned} \int_\Omega W_\alpha^{(2)} (\rho U^\alpha U^\beta)_{,\beta} d\Omega &\approx \frac{\rho}{\sqrt{g}} \left(\left[\begin{matrix} 2 \\ 1 \end{matrix} \right]^{0,0} V^1 V^{1(n)} + \left[\begin{matrix} 2 \\ 2 \end{matrix} \right]^{0,0} V^1 V^{2(n)} \right) \Big|_{(-1,0)}^{(1,0)} \delta\xi^2 + \\ &\frac{\rho}{\sqrt{g}} \left(\left[\begin{matrix} 2 \\ 1 \end{matrix} \right]^{0,0} V^{1(n)} V^2 + \left[\begin{matrix} 2 \\ 2 \end{matrix} \right]^{0,0} V^2 V^{2(n)} \right) \Big|_{(0,-1)}^{(0,1)} \delta\xi^1 \quad (\text{B.4}) \end{aligned}$$

This discretization is equivalent to the discretization of the convection term presented in [88] in Gibbs' vector notation.

The pressure term. Using (2.98) for a V^1 -cell, the pressure term is approximated by

$$\int_{\Omega} W_{\alpha}^{(1)}(g^{\alpha\beta} p)_{,\beta} d\Omega \approx \sqrt{g} p ([\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}]^{0,0} g^{11} + [\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}]^{0,0} g^{12})|_{(-1,0)}^{(1,0)} \delta\xi^2 + \sqrt{g} p ([\begin{smallmatrix} 1 \\ 1 \end{smallmatrix}]^{0,0} g^{12} + [\begin{smallmatrix} 1 \\ 2 \end{smallmatrix}]^{0,0} g^{22})|_{(0,-1)}^{(0,1)} \delta\xi^1 \quad (\text{B.5})$$

and for a V^2 -cell:

$$\int_{\Omega} W_{\alpha}^{(2)}(g^{\alpha\beta} p)_{,\beta} d\Omega \approx ([\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}]^{0,0} g^{11} + [\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}]^{0,0} g^{12}) \sqrt{g} p|_{(-1,0)}^{(1,0)} \delta\xi^2 + ([\begin{smallmatrix} 2 \\ 1 \end{smallmatrix}]^{0,0} g^{12} + [\begin{smallmatrix} 2 \\ 2 \end{smallmatrix}]^{0,0} g^{22}) \sqrt{g} p|_{(0,-1)}^{(0,1)} \delta\xi^1 \quad (\text{B.6})$$

Since

$$[\begin{smallmatrix} \theta \\ \alpha \end{smallmatrix}]^{i,j} g^{\alpha\gamma} = a_{\beta(i,j)}^{(\theta)} \sum_{\sigma} a_{(\alpha)}^{\beta} a_{\sigma}^{(\alpha)} a_{\sigma}^{(\gamma)} = a_{\beta(i,j)}^{(\theta)} \sum_{\sigma} \delta_{\sigma}^{\beta} a_{\sigma}^{(\gamma)} = \sum_{\beta} a_{\beta(i,j)}^{(\theta)} a_{\beta}^{(\gamma)} \quad (\text{B.7})$$

it is natural to introduce the symbol $[\theta \ \gamma]^{i,j}$ defined by:

$$[\theta \ \gamma]^{i,j} = [\begin{smallmatrix} \theta \\ \alpha \end{smallmatrix}]^{i,j} g^{\alpha\gamma} = \sum_{\beta} a_{\beta(i,j)}^{(\theta)} a_{\beta}^{(\gamma)} \quad (\text{B.8})$$

Using this symbol (B.5) and (B.6) can be written as:

$$\int_{\Omega} W_{\alpha}^{(1)}(g^{\alpha\beta} p)_{,\beta} d\Omega \approx ([1 \ 1]^{0,0} \sqrt{g} p)|_{(-1,0)}^{(1,0)} \delta\xi^2 + ([1 \ 2]^{0,0} \sqrt{g} p)|_{(0,-1)}^{(0,1)} \delta\xi^1 \quad (\text{B.9})$$

and

$$\int_{\Omega} W_{\alpha}^{(2)}(g^{\alpha\beta} p)_{,\beta} d\Omega \approx ([2 \ 1]^{0,0} \sqrt{g} p)|_{(-1,0)}^{(1,0)} \delta\xi^2 + ([2 \ 2]^{0,0} \sqrt{g} p)|_{(0,-1)}^{(0,1)} \delta\xi^2 \quad (\text{B.10})$$

Again the shape of the molecules does not differ from the discretization based on method 2 of Chapter 2 and the formulae are similar to the ones in [88].

The viscous stress term. The most difficult part is the discretization of the viscous stress term. For the derivation of this variant an approach slightly different from the derivation of the stress discretization used throughout this thesis is followed.

From (2.94) and (3.8) is found:

$$\int_{\Omega} W_{\alpha}^{(\theta)} \tau_{\beta}^{\alpha\beta} d\Omega = \oint_S W_{\alpha}^{(\theta)} \tau^{\alpha\beta} \delta S_{\beta}^{(\eta)} \approx [\begin{smallmatrix} \theta \\ \alpha \end{smallmatrix}]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^{\beta} + g^{\gamma\beta} U_{,\gamma}^{\alpha}) \delta S_{\beta}^{(1)} + [\begin{smallmatrix} \theta \\ \alpha \end{smallmatrix}]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^{\beta} + g^{\gamma\beta} U_{,\gamma}^{\alpha}) \delta S_{\beta}^{(2)} \quad (\text{B.11})$$

with $\delta S_\beta^{(\eta)}$ defined as (2.69).

In order to get rid of the contravariant derivatives with their Christoffel symbols we approximate a combination of derivatives in a particular point by the mean value of this combination over a volume surrounding this point. This approach is adopted in [88], and is described in [76]. The new control volume may be regarded as a shifted control volume compared to the original control volume. In Figure B.1 such a shifted control volume for a point (1,0) with respect to a V^1 -cell is indicated. The

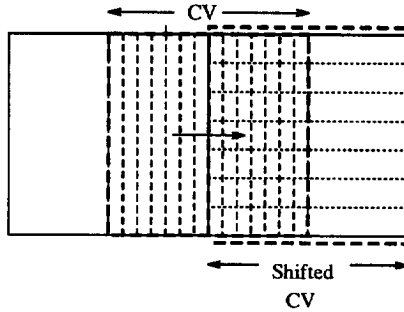


Figure B.1: Control volume for a V^1 -cell and shifted control volume for point (1,0).

local numbering system from Figure 3.3 is used here. Let us confine ourselves to the evaluation of the term in (B.11) corresponding to the point (1,0). Taking the mean value over a shifted control volume and using (B.1) gives

$$\begin{aligned}
 & \left[\begin{array}{c} \theta \\ \alpha \end{array} \right]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} |_{(1,0)} \approx \\
 & a_{\xi 0,0}^{(\theta)} \mu_{1,0} \frac{1}{|\Omega^*|} \int_{\Omega^*} a_{(\alpha)}^\xi (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} d\Omega^* = \\
 & a_{\xi 0,0}^{(\theta)} \mu_{1,0} \frac{1}{|\Omega^*|} \int_{\Omega^*} (a_{(\alpha)}^\xi g^{\alpha\gamma} U^\beta + a_{(\alpha)}^\xi g^{\gamma\beta} U^\alpha)_{,\gamma} \sqrt{g}_{1,0} \delta_\beta^{(1)} d\Omega^* \quad (\text{B.12})
 \end{aligned}$$

using in the last step: $a_{(\alpha),\gamma} = 0$, (2.44) and (2.69).

With Ω^* the shifted control volume and thus $|\Omega^*| = \sqrt{g}_{1,0} \delta \xi^1 \xi^2$ we find

$$\begin{aligned}
 & \left[\begin{array}{c} \theta \\ \alpha \end{array} \right]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} |_{(1,0)} \approx \\
 & a_{\xi 0,0}^{(\theta)} \frac{\mu_{1,0}}{\delta \xi^1 \delta \xi^2} \int_{\Omega^*} (a_{(\alpha)}^\xi g^{\alpha\gamma} U^\beta + a_{(\alpha)}^\xi g^{\gamma\beta} U^\alpha)_{,\gamma} \delta_\beta^{(1)} d\Omega^* \quad (\text{B.13})
 \end{aligned}$$

The choice for $\delta S_\beta^{(\eta)}$ as (2.69) leads to a treatment of (B.13) similar to (2.93), (2.94), and means that the normal vector $\delta S_\beta^{(\eta)}$ is chosen constant over the shifted control

volume Ω^* . Using (B.11) we arrive at

$$\begin{aligned} & \left[\frac{\theta}{\alpha} \right]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} |_{(1,0)} \approx \\ & a_{\xi 0,0}^{(\theta)} \frac{\mu_{1,0}}{\delta \xi^1 \delta \xi^2} \oint_{S^*} (a_{(\alpha)}^\xi g^{\alpha\gamma} U^\beta + a_{(\alpha)}^\xi g^{\gamma\beta} U^\alpha) \left[\frac{1}{\beta} \right]^{1,0} \delta S_\gamma^{(\kappa)*} \end{aligned} \quad (\text{B.14})$$

Equation (B.14) written out completely, using (B.1), (2.27) and (2.61) gives

$$\begin{aligned} & \left[\frac{\theta}{\alpha} \right]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} |_{(1,0)} \approx \frac{\mu_{1,0}}{\delta \xi^1 \delta \xi^2} a_{\xi(0,0)}^{(\theta)} a_{\delta(1,0)}^{(1)*} \\ & \sum_{\eta} \sum_{\zeta} [a_{(\alpha)}^\xi (a_\eta^{(\alpha)} a_\eta^{(\gamma)} V^\beta + a_\zeta^{(\gamma)} a_\zeta^{(\beta)} V^\alpha) a_{(\beta)}^\delta \delta_\gamma^{(1)} |_{0,0}^{2,0} \delta \xi^2 + \\ & a_{(\alpha)}^\xi (a_\eta^{(\alpha)} a_\eta^{(\gamma)} V^\beta + a_\zeta^{(\gamma)} a_\zeta^{(\beta)} V^\alpha) a_{(\beta)}^\delta \delta_\gamma^{(2)} |_{1,-1}^{1,1} \delta \xi^1] \end{aligned} \quad (\text{B.15})$$

This expression can be slightly simplified using (2.22) and the abbreviations introduced in (B.1) and (B.8).

$$\begin{aligned} & \left[\frac{\theta}{\alpha} \right]^{0,0} \mu (g^{\alpha\gamma} U_{,\gamma}^\beta + g^{\gamma\beta} U_{,\gamma}^\alpha) \delta S_\beta^{(1)} |_{(1,0)} \approx \frac{\mu_{1,0}}{\delta \xi^1 \delta \xi^2} a_{\xi(0,0)}^{(\theta)} a_{\delta(1,0)}^{(1)*} \\ & [(a_\xi^{(1)} V^\beta a_{(\beta)}^\delta + a_\delta^{(1)} V^\alpha a_{(\alpha)}^\xi) |_{0,0}^{2,0} \delta \xi^2 + (a_\xi^{(2)} a_{(\beta)}^\delta V^\beta + a_\delta^{(2)} a_{(\alpha)}^\xi V^\alpha) |_{1,-1}^{1,1} \delta \xi^1] \\ & = \frac{\mu_{1,0}}{\delta \xi^1 \delta \xi^2} \{ ([\theta 1]^{0,0} \left[\frac{1}{\beta} \right]^{1,0} V^\beta + [1 1]^{1,0} \left[\frac{\theta}{\alpha} \right]^{0,0} V^\alpha) |_{0,0}^{2,0} \delta \xi^2 \\ & + ([\theta 2]^{0,0} \left[\frac{1}{\beta} \right]^{1,0} V^\beta + [1 2]^{1,0} \left[\frac{\theta}{\alpha} \right]^{0,0} V^\alpha) |_{1,-1}^{1,1} \delta \xi^1 \} \end{aligned} \quad (\text{B.16})$$

The stencil for the V^1 -molecule is sketched in Figure B.2.

In this stencil the points $(-1,3)$, $(-1,-3)$, $(1,3)$, $(1,-3)$ (with the numbering of Figure 3.3) have been eliminated using the continuity equation. However, the size of the molecule is 17, where the size of the molecule of the discretization used throughout this thesis is only 13. This extension of the size is the result of the introduction of the shifted cell Ω^* . So the elimination of the contravariant derivatives and in consequence their corresponding Christoffel symbols, results in a larger stencil for the velocity components in two dimensions. In three dimensions the size of the molecules are the same, because then the molecule coming from method 2 of Chapter 2 cannot be reduced with the continuity equation. Furthermore, the molecules of the variant presented in this Appendix are more complicated, but the geometric quantities to be computed are more simple, because there are no second derivatives involved. A major difficulty of this discretization is the fact that the storage of all new geometric quantities needs a lot of memory, while the computational work to calculate them each time is very time consuming.

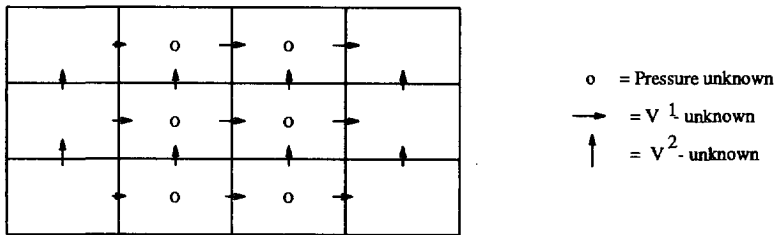
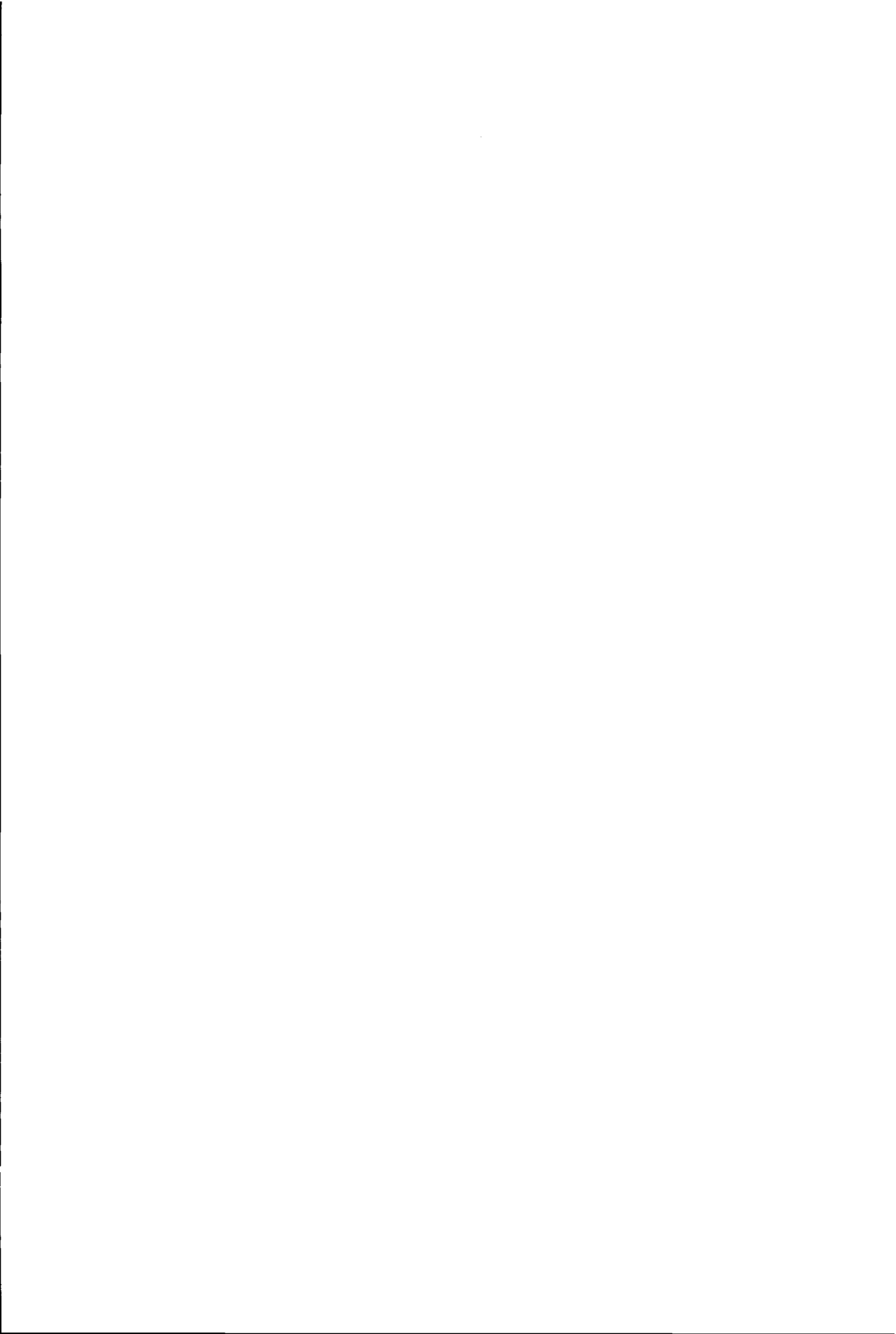


Figure B.2: Molecule corresponding to stress tensor for a V^1 -cell (method 4).



References

- [1] Ch. Arakawa, A. Demuren, W. Rodi, B. Schönung, Application of multi-grid methods for the coupled and decoupled solution of the incompressible Navier-Stokes equations. In: M. Deville (ed.), *Proc. 7th GAMM conf. on Num. Methods in Fluid. Mech.*, Notes on Num. Fluid Mech. **20**, pp. 1-8, Vieweg, Braunschweig, (1988).
- [2] R. Aris, *Vectors, tensors and the basic equations of fluid mechanics*. Prentice-Hall, Englewood Cliffs, N.J. (1962).
- [3] N.S. Bachvalov, On the convergence of a relaxation method with natural constraints on the elliptic operator. *USSR Comp. Math. and Math. Phys.* **6**, 101-135 (1966).
- [4] P. Bastian, J. Burmeister and G. Horton, Implementation of a parallel multi-grid method for parabolic differential equations. In: W. Hackbusch (ed.), *Parallel algorithms for partial differential equations*, Proc. 6th GAMM seminar Kiel, pp. 18-27, Vieweg, Braunschweig, (1990).
- [5] G.K. Batchelor: *An introduction to fluid dynamics*. Cambridge University Press, Cambridge, 1967.
- [6] R.S. Bernard and H. Kapitza, How to discretize the pressure gradient for curvilinear MAC grids. *J. Comp. Phys.* **99**, 288-298 (1992).
- [7] A. Brandt, Multi-level adaptive solutions to boundary value problems. *Math. Comp.* **31**, 333-390 (1977).
- [8] A. Brandt, Guide to multigrid development. In: W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods*, Lecture notes in Mathematics **960**, pp. 220-312, Springer Verlag, Berlin (1982).
- [9] A. Brandt and N. Dinar, Multigrid solutions to flow problems. In: S. Parter (ed.), *Numerical methods for partial differential equations*, pp. 53-147, Academic Press, New York (1979).
- [10] P.N. Brown and Y. Saad, Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comp.*, **11**, 450-481, (1990).
- [11] J. Burmeister, *Paralleles Lösen diskreter parabolischer Probleme mit Mehrgittertechniken* Master's Thesis, Kiel, (1985).
- [12] J. Burmeister and G. Horton, Time-parallel multigrid solution of the Navier-Stokes equations. In: W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods III*, International Series of Numerical Mathematics **98**, pp. 155-166, Birkhäuser, Basel, (1991).

- [13] A.D. Burns and N.S. Wilkes, *A finite difference method for the computation of fluid flows in complex three-dimensional geometries*. Report AERE-R 12342, Harwell Lab. Oxford, (1987).
- [14] J.L.C. Chang, D. Kwak, S.E. Rogers and R.J. Yang, Numerical simulation methods of incompressible flows and an application to the space shuttle main engine. *Int. J. Num. Methods in Fluids* **8**, 1241–1268 (1988).
- [15] H.C. Chen, V.C. Patel and S. Ju, Solutions of Reynolds-averaged Navier-Stokes equations for three-dimensional incompressible flows. *J. Comp. Phys.* **88**, 305–336 (1990).
- [16] N-Y Cho and C.A.J. Fletcher, Computation of turbulent conical diffuser flows using a non-orthogonal grid system. *Comp. and Fluids* **19**, 347–361, (1991).
- [17] A.J. Chorin, A numerical method for solving incompressible viscous flow problems. *J. Comp. Phys.* **2**, 12–26, (1967).
- [18] M. Coutanceau and R. Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow. *Journ. Fluid Mech.* **79**, 231–256, (1977).
- [19] L. Davidson and P. Hedberg, Mathematical derivation of a finite volume formulation for laminar flow in complex geometries. *Int. J. Num. Methods in Fluids* **9**, 531–540 (1989).
- [20] V. De Henau, G.D. Raithby, B.E. Thompson, A total pressure correction for upstream weighted schemes. *Int. J. Num. Meth. Fluids* **9**, 855–864 (1989).
- [21] I. Demirdzic, A.D. Gosman, R. Issa and M. Peric, A calculation procedure for turbulent flow in complex geometries. *Comp. and Fluids* **3**, 251–273, (1987).
- [22] I. Demirdzic and M. Peric, Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *Int. J. Num. Methods in Fluids* **10**, 771–790 (1990).
- [23] I. Demirdzic, Z. Lilek and M. Peric, Fluid flow and heat transfer test problems for non-orthogonal grids: Bench-mark solutions. *Int. J. Num. Methods in Fluids* **15**, 329–354, (1992).
- [24] E. de Sturler, A parallel variant of GMRES(m). In: R. Vichnevetsky and J.J.H. Miller (eds.), *Proc. of 13th IMACS Congr. on Comp. and Appl. Math.*, pp. 682–683, Criterion Press, Dublin (1991).
- [25] E. Dick and J. Linden, A multigrid flux-difference splitting method for steady incompressible Navier-Stokes equations. In: P. Wesseling (ed.), *Proc. of the 8th GAMM Conf. on Num. Meth. in Fluid Mech.*, Notes on Num. Fluid Mech. **29**, pp. 99–108, Vieweg, Braunschweig (1990).

- [26] E. Dick and J. Linden, A multigrid method for steady incompressible Navier-Stokes equations based on flux difference splitting. *Int. J. Num. Methods in Fluids* **14**, 1311-1323 (1992).
- [27] P.R. Eiseman, Grid generation for fluid mechanics computations. *Ann. Rev. Fluid Mech.* **17**, 487-522 (1985).
- [28] R.P. Fedorenko, The speed of convergence of one iterative process. *USSR Comp. Math. and Math. Phys.* **4**(3), 227-235 (1964).
- [29] L. Fuchs, Multi-grid schemes for incompressible flows. In: W. Hackbusch (ed.), *Efficient solutions of elliptic systems*, Notes on Numerical Fluid Mechanics **10**, pp. 38-51, Vieweg, Braunschweig (1984).
- [30] P.F. Galpin, G.D. Raithby, J.P. Van Doormaal, Discussion of upstream-weighted advection approximations for curved grids. *Num. Heat Transfer* **9**, 241-246 (1986).
- [31] C.W. Gear, *Numerical initial value problems for ordinary differential equations*. Prentice Hall, Englewood Cliffs, (1971).
- [32] G.H. Golub and C.F. Van Loan, *Matrix Computations*. Johns Hopkins Univ. Press, Baltimore (1989)
- [33] W. Hackbusch, *Ein iteratives verfahren zur schnellen auflösung elliptischer randwertprobleme*. Report 76/12, Universität Köln, Math. Inst., Köln, (1976).
- [34] W. Hackbusch, Parabolic multigrid methods. In: R. Glowinski, J.R. Lions (eds.), *Computing methods in Applied Sciences and Engineering VI*, Proc. 6th Int. Symp. on Comp. Meth. in Appl. Sciences and Eng. pp. 20-45, North Holland, Amsterdam, (1984).
- [35] W. Hackbusch, *Multi-grid methods and applications*. Springer Verlag, Berlin (1985).
- [36] W. Hackbusch, Comparison of different multigrid variants for non-linear equations. *Z. angew. Math. Mech.* **72**, 148-151 (1992).
- [37] E. Hairer, G. Wanner, *Solving ordinary differential equations II*, Springer Verlag, Berlin (1991).
- [38] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow. *Phys. Fluids* **8**, 2182-2189 (1965).
- [39] P.W. Hemker, Mixed defect correction iteration for the accurate solution of the convection diffusion equation. In: W. Hackbusch and U. Trottenberg (eds.), *Multigrid Methods*, Lecture notes in Mathematics **960**, pp. 485-501, Springer Verlag, Berlin (1982).

- [40] Ch. Hirsch, *Numerical computation of internal and external flows, Vols. 1, 2*, John Wiley, Chichester (1990).
- [41] Ch. Hirsch, Discretization methods for the incompressible Navier-Stokes equations. In: *Computational fluid dynamics for industrial flows*, Lecture Series 90-004, von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium (1990).
- [42] G. Horton, Time-parallel multigrid solution of the Navier-Stokes equations. In: C. Brebbia (ed.), *Applications of supercomputers in Engineering*. Elsevier, Amsterdam, (1991).
- [43] G. Horton, The time-parallel multigrid method. *Comm. Appl. Num. Methods* **8**, 585-596 (1992)
- [44] T. Ikohagi and B.R. Shin, Finite difference schemes for steady incompressible Navier-Stokes equations in general curvilinear coordinates. *Comp. and Fluids* **19**, 479-488, (1991).
- [45] 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. *Comp. and Fluids* **21**, 163-175, (1992).
- [46] I.P. Jones, Experience with the vectorisation of some fluid flow prediction codes on the Harwell Cray I. In: *Vectorisation of computer programs with application to computational fluid dynamics*, Notes on Numerical Fluid Mechanics **8**, pp. 244-253, Vieweg, Braunschweig (1984).
- [47] D.S. Joshi and S.P. Vanka, Multigrid calculation procedure for internal flows in complex geometries. *Num. Heat Transfer* **20**, 61-80, (1991).
- [48] K. Katsuragi and O. Ukai, Navier-Stokes equations on a curvilinear coordinate system. In: P. Wesseling (ed.), *Proc. of the 8th GAMM Conf. on Num. Meth. in Fluid Mech.*, Notes on Num. Fluid Mech. **29**, pp. 233-241, Vieweg, Braunschweig (1990).
- [49] S. Koshizuka, Y. Oka and S. Kondo, A staggered differencing technique on boundary-fitted curvilinear grids for incompressible flows along curvilinear or slant walls. *Comp. Mechanics* **7**, 123-136 (1990).
- [50] D. Kwak, J.L.C. Chang, S.P. Shanks and S.R. Chakravarthy, A three-dimensional incompressible Navier-Stokes flow solver using primitive variables. *AIAA Journ.* **24**, 390-396 (1986).
- [51] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comp. Meth. in Appl. Mech. and Eng.* **19**, 59-98 (1979).

- [52] F-S Lien and M.A. Leschziner, Multigrid convergence acceleration for complex flow including turbulence. In: W. Hackbusch, U. Trottenberg (eds.), *Multigrid Methods III*, Ser. of Numerical Mathematics, **98**, pp. 277-288, Birkhäuser, Basel, (1991).
- [53] A.S. Lodge, W.G. Pritchard, and L.R. Scott, The hole-pressure problem. *IMA Journal of Applied Mathematics* **46**, 39-66 (1991).
- [54] Y.P. Marx, Evaluation of the artificial compressibility method for the solution of the incompressible Navier-Stokes equations. In: J.B. Vos, A. Rizzi, I.L. Ryhming (eds.), *Proc. of the 9th GAMM Conf. on Num. Meth. in Fluid Mech.*, Notes on Num. Fluid Mech. **35**, pp 201 - 210, Vieweg, Braunschweig (1992).
- [55] O.A. McBryan, P.O. Frederickson, J. Linden, A. Schüller, K. Solchenbach, K. Stüben, C-A. Thole and U. Trottenberg, Multigrid methods on parallel computers - A survey of recent developments. *Impact Comp. Science and Eng.* **3**, 1-75 (1991).
- [56] J.A. Meijerink and H.A. Van der Vorst, An iterative solution method for linear systems of which the coefficient is a symmetric M-matrix. *Math. Comp.* **31**, 148-162 (1977).
- [57] H. Miyata and Y. Yamada, A finite difference method for 3D flows about bodies of complex geometry in rectangular co-ordinate systems. *Int. J. Num. Methods in Fluids* **14**, 1261-1287 (1992).
- [58] K.J. Morgan, J. Periaux, and F. Thomasset, editors. *Analysis of Laminar Flow over a Backward Facing Step*, GAMM Workshop held at Bièvres (Fr.), Vieweg, Braunschweig (1984).
- [59] W.A. Mulder, A new multigrid approach to convection problems. *J. Comp. Phys.* **83**, 303-323 (1989).
- [60] S. Murata, N. Satofuka and T. Kushiya, Parabolic multi-grid method for incompressible viscous flows using a group explicit relaxation scheme. *Comp. and Fluids*, **19**, 33-41, (1991).
- [61] A.E. Mynett, P. Wesseling, A. Segal, and C.G.M. Kassels, The ISNaS incompressible Navier-Stokes solver: invariant discretization. *Applied Scientific Research* **48**, 175-191 (1991).
- [62] A. Nakayama, A numerical method for solving momentum equations in generalized coordinates (Its application to three-dimensional separated flows). *J. Fluids Eng.* **107**, 49-54 (1985).
- [63] F. Nieuwstadt and H. B. Keller, Viscous flow past circular cylinders. *Computers and Fluids*, **1**, 59-71, (1973).

- [64] C.W. Oosterlee and P. Wesseling, A multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates. In: J.B. Vos, A. Rizzi, I.L. Ryming (eds.), *Proc. of the 9th GAMM Conf. on Num. Meth. in Fluid Mech.*, Notes on Num. Fluid Mech. **35**, pp 99 – 106, Vieweg, Braunschweig (1992).
- [65] C.W. Oosterlee and P. Wesseling, A multigrid method for an invariant formulation of the incompressible Navier-Stokes equations in general coordinates. *Comm. Appl. Num. Methods* **8**, 721–734 (1992)
- [66] C.W. Oosterlee and P. Wesseling, *A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates* Report 92/14, TU Delft, Fac. Technical Math. Inf., Delft (1992). To appear in *Impact of Comp. in Science and Eng.*
- [67] C.W. Oosterlee and P. Wesseling, A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates. In: Ch. Hirsch, J. Periaux, W. Kordulla (Eds.), *Proc. of the First Europ. Fluid Dyn. Conf.*, pp 101 –108, Elsevier, Amsterdam (1992).
- [68] C.W. Oosterlee and P. Wesseling, *Steady incompressible flow around objects in general coordinates with a multigrid solution method.*, Report 92-48, Fac. Technical Math. and Inf., Delft University of Technology, Delft (1992). Submitted to *Num. Methods for Part. Diff. Equations*.
- [69] C.W. Oosterlee, P. Wesseling, A. Segal and E. Brakkee, *Benchmark solutions for the incompressible Navier-Stokes equations in general coordinates on staggered grids.* Report 92-67, Fac. Technical Math. and Inf., Delft University of Technology, Delft (1992). To appear in *Int. J. Num. Meth. in Fluids*.
- [70] C.W. Oosterlee and P. Wesseling, *Multigrid schemes for time-dependent incompressible Navier-Stokes equations.* Report 92-102, Fac. Technical Math. and Inf., Delft University of Technology, Delft (1992). To appear in *Impact of Comp. in Science and Eng.*
- [71] S. V. Patankar and D. B. Spalding, A calculation procedure for heat and mass transfer in three-dimensional parabolic flows. *Int. J. Heat Mass Transfer* **15**, 1787–1806 (1972).
- [72] S.V. Patankar, *Numerical heat transfer and fluid flow.* McGraw-Hill, New York (1980).
- [73] M. Peric, *A finite volume method for the prediction of three-dimensional fluid flow in complex ducts.* Ph.D. Thesis, University of London, London (1985).
- [74] M. Peric, R. Kessler and G. Scheuerer, Comparison of finite volume numerical methods with staggered and collocated grids. *Comp. and Fluids* **16**, 389–403 (1990)

- [75] C.Y. Pong and R.L. Street, A coupled multigrid-domain-splitting technique for simulating incompressible flows in geometrically complex domains. *Int. J. Num. Methods in Fluids* **13**, 269–286 (1991).
- [76] R. Peyret, T.D. Taylor, *Computational methods for fluid flow*. Springer Verlag, Berlin (1983).
- [77] A.S. Posthuma, *A code generator for the discretization and Fortran coding of 3-D incompressible Navier-Stokes equations*. Master's Thesis, University of Twente, Enschede (1992).
- [78] W.G. Pritchard, Private e-mail communication (1992).
- [79] G.D. Raithby, K.E. Torrance, Upstream-weighted differencing schemes and their application to elliptic problems involving fluid flow. *Comp. and Fluids* **2**, 191–206 (1974).
- [80] G.D. Raithby, Skew upstream differencing schemes for problems involving fluid flow. *Comp. Meth. in Appl. Mech. and Eng.* **9**, 153–164 (1976).
- [81] D. Rayner, Multigrid flow solutions in complex two-dimensional geometries. *Int. J. Num. Methods in Fluids* **13**, 507–518 (1991).
- [82] C.M. Rhie and W.L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing-edge separation. *AIAA Journ.* **21**, 1525–1532 (1983).
- [83] C.M. Rhie *A pressure based Navier-Stokes solver using the multigrid method*. AIAA Paper 86-0207 (1986).
- [84] W. Rodi, S. Majumdar and B. Schöning, Finite volume methods for two-dimensional incompressible flows with complex boundaries. *Comp. Meth. in Appl. Mech. and Eng.* **75**, 369–392 (1989).
- [85] S.E. Rogers and D. Kwak, Upwind differencing scheme for the time-accurate incompressible Navier-Stokes equations. *AIAA Journ.* **28**, 253–262 (1990).
- [86] J.I. Rollet, D.F. Mayers and T.M. Shah, *Analysis and application of an efficient line solver based upon the Symmetric Coupled Gauss-Seidel scheme*. Report 88/10, Oxford Univ. Comp. Lab. (1988).
- [87] M. Rosenfeld, D. Kwak, *Numerical solution of unsteady incompressible viscous flows in generalized moving coordinate systems*. AIAA Paper 89-0466 (1989).
- [88] M. Rosenfeld, D. Kwak and M. Vinokur, A fractional step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems. *J. Comp. Phys.* **94**, 102–137 (1991).
- [89] M. Rosenfeld, D. Kwak, *Multigrid acceleration of a fractional step solver in generalized coordinate systems*. AIAA Paper 92-0185 (1992).

- [90] Y. Saad and M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, **7**, 856–869, (1986).
- [91] A. Segal, P. Wesseling, J. van Kan, C.W. Oosterlee and C.G.M. Kassels, Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted co-ordinates. *Int. J. Num. Methods in Fluids* **15**, 411–426 (1992).
- [92] T.M. Shah, *Analysis of a multigrid method*. Ph.D. Thesis, Oxford Univ. Comp. Lab. (1987).
- [93] G.J. Shaw and S. Sivaloganathan, On the smoothing of the SIMPLE pressure correction algorithm. *Int. J. Num. Methods in Fluids* **8**, 441–462 (1988).
- [94] C.F. Shieh, Three-dimensional grid generation using elliptic equations with direct grid distribution control. *AIAA Journ.* **22**, 361–364 (1984).
- [95] W. Shyy, S.S. Tong, S.M. Correa, Numerical recirculating flow calculation using a body-fitted coordinate system. *Num. Heat Transfer* **8**, 99–113 (1985).
- [96] W. Shyy and T.C. Vu, On the adoption of velocity variable and grid system for fluid flow computation in curvilinear coordinates. *J. Comp. Phys.* **92**, 82–105, (1991).
- [97] W. Shyy and C-S. Sun, Development of a pressure-correction/staggered-grid based multigrid solver for incompressible recirculating flows. *Comp. and Fluids* **22**, 51–76 (1992).
- [98] W.Y. Soh and J.W. Goodrich, Unsteady solution of incompressible Navier-Stokes equations. *J. Comp. Phys.* **79**, 113–134, (1988).
- [99] I.S. Sokolnikoff, *Tensor analysis*. John Wiley, New York (1964).
- [100] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, **10**, 36–52, (1989).
- [101] M.C. Thompson and J.H. Ferziger, An adaptive multigrid technique for the incompressible Navier-Stokes equations. *J. Comp. Phys.* **82**, 94–121 (1989).
- [102] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, *Numerical grid generation*. North Holland, Amsterdam (1985)
- [103] H.A. Van der Vorst, BICGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, **13**, 631–644, (1992).
- [104] H.A. Van der Vorst and C. Vuik, *GMRESR: a family of nested GMRES methods*. Report 91/80, TU Delft, Fac. Technical Math. & Inf., Delft (1991). To appear in *J. Num. Lin. Alg. Appl.*

- [105] S. Vandewalle, *The parallel solution of parabolic partial differential equations by multigrid waveform relaxation methods*. Ph.D. Thesis, Leuven University, Leuven (1992).
- [106] S. Vandewalle, R. Piessens, Efficient parallel algorithms for solving initial-boundary value and time-periodic parabolic partial differential equations. *SIAM J. Sci. Stat. Comp.*, **13**, 1330–1346, (1992).
- [107] S.P. Vanka, Block-implicit calculation of steady turbulent recirculating flows. *Int. J. Heat Mass Transfer* **28**, 2093–2103 (1985).
- [108] S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables. *J. Comp. Phys.* **65**, 138–158 (1986).
- [109] S.P. Vanka, Block implicit multigrid calculation of two-dimensional recirculating flows. *Comp. Meth. in Appl. Mech. and Eng.* **59**, 29–48 (1986).
- [110] S.P. Vanka, A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. *Comp. Meth. in Appl. Mech. and Eng.* **59**, 321–338 (1986).
- [111] S.P. Vanka and K. Misegades, *Vectorized Multigrid Fluid Flow Calculations on a CRAY X-MP48*. AIAA Paper 86-0059 (1986).
- [112] J.J.I.M. van Kan, A second-order accurate pressure correction method for viscous incompressible flow. *SIAM J. Sci. Stat. Comp.* **7**, 870–891, (1986).
- [113] J. van Kan, C.W. Oosterlee, A. Segal, P. Wesseling. *Discretization of the incompressible Navier-Stokes equations in general coordinates using contravariant velocity components*. Report 91/09, TU Delft, Fac. Technical Math. & Inf., Delft (1991).
- [114] R.S. Varga, *Matrix iterative analysis*. Prentice-Hall, Englewood Cliffs, N.J. (1962).
- [115] C. Vuik, Solution of the discretized incompressible Navier-Stokes equations with the GMRES method. *Int. J. Num. Methods in Fluids* **16**, 507–523 (1993).
- [116] P. Wesseling, *Finite volume discretization of the convection diffusion equation in general coordinates*. Report 91/72, TU Delft, Fac. Technical Math. & Inf., Delft (1991).
- [117] P. Wesseling, Two remarks on multigrid methods. In: W. Hackbusch (ed.), *Robust Multigrid methods*, Proc. 4th GAMM seminar Kiel, Notes on Num. Fluid Mech. **23**, pp. 209–216, Vieweg, Braunschweig, (1988).
- [118] P. Wesseling, *An introduction to multigrid methods*. John Wiley, Chichester (1992).

- [119] P. Wesseling, A. Segal, J. 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. *Comp. Fluids Dyn. Journal* **1**, 27–33, (1992).
- [120] J. White and A. Sangiovanni-Vincentelli, *Relaxation techniques for the simulation of VLSI Circuits*. Kluwer Academic Publishers, Boston, (1987).
- [121] G. Wittum, *Distributive Iterationen für indefinite Systeme als Glätter in Mehrgitterverfahren am Beispiel der Stokes- und Navier-Stokes- Gleichungen mit Schwerpunkt auf unvollständigen Zerlegungen*. Ph.D. Thesis, University Kiel, Germany (1986).
- [122] G. Wittum, Mehrgitterverfahren für die Stokesche Gleichung. *Z. angew. Math. Mech.* **67**, 499–501 (1987).
- [123] G. Wittum, Multi-grid methods for Stokes and Navier-Stokes equations with transforming smoothers: algorithms and numerical results. *Numer. Math.* **54**, 543–563 (1989).
- [124] N. Wright. *Multigrid solutions of elliptic fluid flow problems*. Ph.D Thesis, Leeds University (1988).
- [125] Zeng Shi and P. Wesseling, *Numerical solution of a bifurcation problem for the Boussinesq equations at low Prandtl number by a multigrid method*. Report 89/67, TU Delft, Fac. Technical Math. & Inf., Delft (1989).
- [126] Zeng Shi and P. Wesseling, *An ILU smoother for the incompressible Navier-Stokes equations in general coordinates*. Report 92/91, TU Delft, Fac. Technical Math. & Inf., Delft (1992).
- [127] Z. Zhu and C.A. Fletcher, A study of sequential solutions for the reduced/complete Navier-Stokes equations with multigrid acceleration. *Comp. and Fluids* **19**, 43–60 (1991).
- [128] M. Zijlema, *Een studie naar randvoorwaarden en discretisaties bij een vaste wand voor turbulentiemodellen in incompressibele stromingsmodellen*. Master's Thesis (in Dutch), TU Delft, Delft (1991).

Samenvatting.

Robuuste multirooster methoden voor de stationaire en instationaire incompressibele Navier-Stokes vergelijkingen in algemene coördinaten.

Met de huidige generatie supercomputers kunnen zeer grote stelsels gediscrètiseerde differentiaal vergelijkingen numeriek opgelost worden. Het is zelfs mogelijk om drie-dimensionale vergelijkingen door te rekenen.

Voor het oplossen van stromingsleer problemen is de numerieke stromingsleer naast het uitvoeren van schaal experimenten een belangrijke manier om resultaten te verkrijgen. De meest algemene partiële differentiaal vergelijkingen die stromingsfenomenen beschrijven zijn de Navier-Stokes vergelijkingen, opgesteld in de negentiende eeuw. Deze vergelijkingen beschrijven een stroming met behulp van basiswetten als behoud van massa, behoud van impuls en behoud van energie. In dit proefschrift worden de Navier-Stokes vergelijkingen voor niet-samendrukbare Newtonse stromingen beschouwd: de incompressibele Navier-Stokes vergelijkingen. Voor dit type stromingen zijn de vergelijkingen iets te vereenvoudigen, want de divergentie van het snelheidsveld is nul. De energie vergelijkingen kunnen dan ontkoppeld van de massa behoud en impuls vergelijkingen worden opgelost. Teneinde problemen in willekeurige domeinen op te lossen zijn de stationaire en instationaire incompressibele Navier-Stokes vergelijkingen in algemene coördinaten beschouwd. Er wordt gewerkt met een coördinaat-invariante tensorformulering van de vergelijkingen. De gebruikte tensorformules worden gepresenteerd in hoofdstuk 2.

Een vereiste voor het numeriek oplossen van fysische problemen is naast nauwkeurige fysische modellering numerieke nauwkeurigheid. De incompressibele Navier-Stokes vergelijkingen in algemene coördinaten worden gediscrètiseerd met een eindige volume methode. Het verkrijgen van nauwkeurige stromings resultaten uit deze gediscrètiseerde vergelijkingen is verre van triviaal. In dit proefschrift worden verschillende eisen, bijvoorbeeld aan de keuze van onbekenden, opgesteld ten einde nauwkeurige resultaten te verkrijgen. Hoewel stromingen die in de natuur en in de industrie voorkomen vaak zeer complex zijn, we treffen daar vooral turbulente, niet-Newtonse, meer-fase stromingen, of stromingen met chemische reacties aan, zijn de hier behandelde één-fase, laminaire vergelijkingen toch bijzonder interessant. Deze vergelijkingen representeren een basis, waarop door middel van extra aannames en extra vergelijkingen de meer complexe stromingsfenomenen gesuperponeerd kunnen worden. Het is interessant om het gedrag, de mogelijkheden en de beperkingen van onderliggende basis vergelijkingen opgesteld in algemene coördinaten te bestuderen. Deze incompressibele Navier-Stokes vergelijkingen worden geïntroduceerd, gediscrètiseerd en onderzocht in hoofdstuk 3.

In het tweede gedeelte van het proefschrift wordt onderzoek gedaan naar snelle robuuste methoden om de resulterende grote stelsels gediscrètiseerde vergelijkingen op te lossen. Gekozen is voor de multirooster methode, omdat met deze methode, mits de diverse onderdelen goed worden gekozen, goede convergentie factoren on-

afhankelijk van het aantal roosterpunten verkregen kunnen worden. Verder is het werk nodig om problemen met m onbekenden op te lossen van de orde m . Multirooster methoden zijn halverwege de jaren zeventig ([33], [7]) ontdekt als snelle oplosmethoden voor grote stelsels vergelijkingen. Sindsdien is er veel onderzoek gedaan naar de methode in allerlei disciplines waar numerieke modellen gebruikt worden. Numerieke stromingsleer is één van die gebieden. Recente resultaten behaald in dit gebied zijn gepresenteerd in [118]. Voor de stationaire (niet-lineaire) incompressibele Navier-Stokes vergelijkingen wordt gebruik gemaakt van de standaard niet-lineaire multirooster methode ([35], [8]). In hoofdstuk 4 van dit proefschrift wordt deze methode gepresenteerd en worden drie gladstrijk methoden onderzocht op robuustheid door zeer vele test problemen in diverse gebieden door te rekenen.

In hoofdstuk 5 worden drie multirooster methoden om instationaire vergelijkingen op te lossen onderzocht. Deze drie methoden hebben ongeveer dezelfde sequentiële complexiteit, terwijl de vermoedelijke efficiëntie op parallele computers wezenlijk verschillend zal zijn. In dit proefschrift wordt het gedrag op een sequentiële machine vergeleken.

Twee interessante aspecten in dit proefschrift zijn: Allereerst het onderzoek aan de coördinaat-invariante formulering en de eindige volume discretizatie van de incompressibele Navier-Stokes vergelijkingen. Ten tweede, het onderzoek naar een snelle robuuste multirooster oplosmethode om de resulterende stelsels stationaire en instationaire vergelijkingen op te lossen.

Curriculum vitae.

De schrijver van dit proefschrift werd op 12 september 1967 geboren te Zoeterwoude. De middelbare school werd gevolgd aan het Bonaventura college te Leiden, alwaar in juli 1985 het VWO diploma werd behaald.

Vanaf September 1985 heeft hij de studie Technische Wiskunde aan de Technische Universiteit te Delft gevolgd. Op 9 september 1989 werd de graad van wiskundig ingenieur verkregen in de vakgroep Toegepaste Analyse. Het afstudeerwerk onder supervisie van Prof. dr. ir. P. Wesseling had de titel "Een parallele multirooster methode".

In dezelfde vakgroep is hij als AIO in September 1989 het promotiewerk begonnen, dat geresulteerd heeft in dit proefschrift.

Gedurende de promotie periode zijn voordrachten gehouden op conferenties in Copper Mountain (Verenigde Staten), in Lausanne (Zwitserland), in Brussel (Belgie) en in Amsterdam. Verder zijn op diverse symposia en werkgemeenschapsbijeenkomsten in Nederland voordrachten gehouden. Uit het werk zijn verscheidene tijdschriftartikelen voortgekomen.

Gedurende de AIO periode is hij genomineerd geweest in de "student paper competition" in Copper Mountain en heeft hij "eervolle vermelding" in de SIAM student paper competitie verkregen.