

Computation of flows using unstructured staggered grids

Printed by:

Optima Grafische Communicatie
Pearl Buckplaats 37
Postbus 84118, 3009 CC Rotterdam
The Netherlands
Phone: +31 - 10 - 220 1149
Fax: +31 - 10 - 456 6354
E-mail: optima@ogc.nl
WWW: <http://www.ogc.nl>

Computation of flows using unstructured staggered grids

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.dr.ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen
op maandag 18 februari 2002 te 16:00 uur

door

Ivo WENNEKER
natuurkundig ingenieur

geboren te Groenlo.

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

Toegevoegd promotor: Ir. A. Segal

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. P. Wesseling,	Technische Universiteit Delft, promotor
Ir. A. Segal,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr. A.E.P. Veldman,	Rijksuniversiteit Groningen
Prof.dr.ir. G.S. Stelling,	Technische Universiteit Delft en WL Delft Hydraulics
Prof.dr.ir. H.W.M. Hoeijmakers,	Technische Universiteit Twente
Dr. T.J. Barth,	NASA Ames Research Center
Dr.ir. B. Oskam,	Nationaal Lucht- en Ruimtevaartlaboratorium
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft, reservelid

Computation of flows using unstructured staggered grids.
Dissertation at Delft University of Technology.
Copyright © 2002 by I. Wenneker.

The work described in this thesis was financially supported by the Netherlands
Organization for Scientific Research (NWO).

Voor mijn ouders

Summary

Computation of flows using unstructured staggered grids

Thanks to the tremendous increase in computer resources and many advances in the field of numerical techniques, computational fluid dynamics (CFD) is nowadays capable of computing flows in complex domains. One of the consequences is that grid generation has become a major bottleneck for utilization of codes based on the structured multiblock approach. Experience namely has shown that it is impossible to gain a satisfactory reduction of human labor involved in generation of structured grids. Another disadvantage of structured grids consists of inherent difficulties associated with local grid refinement and grid adaptation. These difficulties have led to the emergence of unstructured grids in CFD. Generation of unstructured grids can be automated to a much larger extent than generation of structured grids, and local and adaptive grid refinement are much easier to realize on unstructured grids. Another well-known issue in CFD is the design of Mach-uniform methods, i.e. methods that are capable of computing flows with accuracy and efficiency uniform in the Mach number M , from incompressible flows ($M = 0$) to supersonic flows ($M > 1$). In order to avoid spurious pressure oscillations in incompressible or weakly compressible flows, a staggered grid, in which the normal momentum components are located at the cell faces and the scalar variables are positioned in the cell centroids, can be employed. In this thesis, a novel Mach-uniform method to compute flows using unstructured staggered grids is described.

In Chapter 2, the Navier-Stokes equations, governing the motion of fluid flows, are introduced. In addition, some important approximations of these equations, namely the Euler equations (viscous and diffusive effects are omitted) and the incompressible flow equations (the density of each material particle remains constant) are addressed. The issue of structured versus unstructured grids is discussed. A literature survey of compressible and incompressible flow solvers on unstructured grids is given, and it is concluded that the method introduced in this thesis is novel with respect to: (i) the spatial discretization, and (ii) the Mach-uniform solution procedure.

Three solution procedures are discussed in Chapter 3. The well-known pressure-correction approach (PC) is used to compute incompressible flows. For simplicity, a straightforward sequential update procedure for fully compressible flows (FC) is introduced to test the spatial discretization. The need for Mach-uniform methods is motivated and we address the difficulties associated with the incompressible limit encountered by the standard compressible flow solvers. The Mach-uniform approach (MU) employed in this thesis is a generalization of the PC approach and incorporates all compressibility effects. Hence, the MU scheme is pressure-based, is valid even

for $M > 1$, and updates the variables sequentially. The MU approach reverts back to the standard PC scheme for $M = 0$. Time integration is done by means of the θ -method, and the resulting linear systems (two for the PC approach, and three for the FC and MU approaches if $\theta > 0$) are solved by means of preconditioned Krylov subspace methods.

The spatial discretization of the governing equations on unstructured staggered grids forms the subject of Chapter 4. Due to the staggered grid arrangement, it is impossible to define ‘left’ and ‘right’ states as for standard colocated methods for compressible flows. This is why schemes based on approximate Riemann solvers cannot be applied. On the other hand, discretization by a simple finite difference or finite volume scheme for each primary variable separately (a segregated approach) is natural on a staggered grid, and this is the strategy that we pursue. We have adopted a finite volume approach, thereby ensuring satisfaction of the Rankine-Hugoniot shock conditions, see also Appendix D. The momentum equation is integrated over the two triangles adjacent to the considered face, while integration over two half triangles turns out to yield an inconsistent scheme, see also Appendix B. For computation of the convecting velocity at the control volume faces, a central and an upwind biased approximation are devised. The convected momentum term itself is evaluated by means of a first order upwind or a central scheme. Four different pressure gradient methods are designed: the path integral (PI) method, the contour integral (CI) method, the auxiliary point (AP) method and the four quadrant (FQ) method. Discretization of the viscous term is discussed. The continuity equation, the energy equation (in the FC approach) and the Mach-uniform pressure-correction equation (in the MU approach) are integrated over each triangle, and the convection term is evaluated by means of a first order upwind or a central scheme. Computation of the pressure (in the FC approach) or the enthalpy (in the MU approach) by means of the equation of state for a perfect gas is described. The proposed central discretization of the convection operator of the scalar equations preserves the underlying symmetry of the considered operator, while for the inertia term this is only the case on regular grids, see Appendix C.

Numerical results for viscous incompressible flows (Poiseuille flow, the backward facing step and the lid-driven cavity) are discussed in Chapter 5. On grids that are made non-smooth deliberately, the PI method performs best. The CI method can lead to spurious pressure modes. The PI method results in a nonsymmetric pressure-correction matrix that is not an M-matrix; nevertheless the eigenvalues of this matrix have, as they should, a very small imaginary part as compared to the (positive) real part. Solving the resulting pressure-correction equation is done by means of Bi-CGSTAB together with a so-called lumped ILU preconditioner. Solutions for the flow over the backward facing step and the lid-driven cavity are compared with other, both experimental and numerical, results.

In Chapter 6, numerical results for inviscid compressible flows are given. The Burgers equation is suitable to test the discretization of the inertia term. It is found that the first order upwind scheme introduces a small overshoot, which does not grow in time. The convecting velocity should, in order to avoid excessive artificial diffusion, be approximated in a central manner. The issue of monotonicity is addressed in Appendix A. When applied to Riemann problems, our scheme is found to converge, without giving rise to spurious modes, to the entropy solution. The accuracy is similar to that of the well-accepted Roe and AUSM schemes. The PI method is found to perform slightly better than the other pressure gradient schemes. Central discretization of the inertia term leads to a matrix that almost preserves the symmetry of the underlying operator: the eigenvalues have, compared to the imaginary part, a relatively small real part. A grid refinement study for subsonic flow in a channel with 10% sinusoidal bump demonstrates that our scheme performs almost as well on general unstructured grids as on Courant grids, and that the order of the spatial accuracy of the first order upwind scheme is one. Using a structured staggered scheme with the same number of nodes on the boundary leads to slightly less diffusive results. Supersonic flow in a channel with 4% circular bump shows the capability to deal with supersonic flow: a good agreement with previously published results is obtained. Flows with different freestream Mach numbers ($M_\infty = 0, 10^{-3}, 10^{-2}, 10^{-1}, 0.63, 0.8, 1.2$) around the NACA 0012 airfoil are used to study the accuracy and efficiency of our schemes for incompressible and compressible flows. It is found that the MU formulation is over the whole studied range of Mach numbers much more efficient than the FC approach. This is due to the fact that in the MU approach the pressure is taken implicitly, which enables the use of much larger time steps, which results on its turn in fewer time steps before convergence is reached. In the MU formulation, quick convergence is obtained for Courant numbers typically in the order of a few hundred, whereas the FC becomes unstable for Courant numbers larger than about 2 or 3. In the FC approach, most CPU time is spent in computation of the matrix elements and right-hand side, whereas the MU approach spends most of its CPU time in the linear solver. As is common for density-based methods, the FC becomes less accurate in the low Mach number regime; this is not the case for the pressure-based MU approach. An example in which both incompressible and compressible regions occupy a large part of the flow domain is that of flow in a converging-diverging nozzle. Solutions demonstrate good agreements with structured grid and theoretical results. Mach-uniform efficiency is obtained, since the CPU time per time step only depends on the magnitude of the time step size and not on the contraction ratio of the nozzle.

Samenvatting

Berekening van stromingen met behulp van ongestructureerde, ‘staggered’ roosters

Dankzij de enorme toename in computerkracht en de vooruitgang op het gebied van numerieke technieken, is de numerieke stromingsleer (CFD) tegenwoordig in staat om stromingen in complexe domeinen te berekenen. Een van de gevolgen is dat rooster generatie een belangrijk knelpunt voor het toepassen van codes gebaseerd op de gestructureerde multi-blok aanpak is geworden. Ervaring heeft namelijk uitgewezen dat het onmogelijk is tot een aanvaardbare reductie van het aantal manuren dat benodigd is voor generatie van gestructureerde roosters te komen. Een ander nadeel van gestructureerde roosters is gerelateerd aan de niet of nauwelijks te overkomen moeilijkheden met betrekking tot lokale en adaptieve rooster verfijning. Deze moeilijkheden hebben geleid tot de opkomst van ongestructureerde roosters in CFD. Het genereren van ongestructureerde roosters kan veel verder geautomatiseerd worden dan dat van gestructureerde roosters, en lokale en adaptieve rooster verfijning zijn op ongestructureerde roosters veel eenvoudiger te realiseren. Een ander bekend onderwerp binnen de CFD is de ontwikkeling van Mach-uniforme methoden. Dit zijn methoden die in staat zijn stromingen te berekenen met een nauwkeurigheid en efficiëntie die uniform is in het Mach getal M , van incompressibele stromingen ($M = 0$) tot en met supersonische stromingen ($M > 1$). Om niet-fysische druk oscillaties in incompressibele of zwak compressibele stromingen te voorkomen, kan een ‘staggered’ (lett. verschoven) rooster, waarin de normale impuls componenten op de celzijden en de scalaire grootheden in de celcentra zijn gepositioneerd, gebruikt worden. In dit proefschrift wordt een nieuwe Mach-uniforme methode om stromingen met behulp van ongestructureerde ‘staggered’ roosters beschreven.

In Hoofdstuk 2 worden de Navier-Stokes vergelijkingen, die de beweging van fluida beschrijven, geïntroduceerd. Verder worden enige belangrijke vereenvoudigingen hiervan, namelijk de Euler vergelijkingen (verwaarlozing van visceuze en diffusie effecten) en de incompressibele stromingsvergelijkingen (de dichtheid van ieder materieel deeltje blijft constant), behandeld. De kwestie van gestructureerde versus ongestructureerde roosters wordt aangeroerd. Een literatuuroverzicht van compressibele en incompressibele flow solvers op ongestructureerde roosters wordt gegeven, en men kan concluderen dat de aanpak die in dit proefschrift beschreven wordt met betrekking tot de volgende aspecten nieuw is: (i) de ruimtelijke discretisatie, en (ii) de Mach-uniforme oplosstrategie.

Drie oplosstrategieën worden bediscussieerd in Hoofdstuk 3. De bekende druk-correctie aanpak (PC) wordt gebruikt om incompressibele stromingen te berekenen. Een voor de hand liggende sequentiële update procedure voor volledig compressibele

stromingen (FC) is bedacht om de ruimtelijke discretisatie te testen. We motiveren de behoefte aan Mach-uniforme methoden en we behandelen de moeilijkheden die standaard flow solvers hebben met de incompressibele limiet. De Mach-uniforme oplosstrategie (MU), die in dit proefschrift gebruikt wordt, is een generalisatie van de PC aanpak en bevat alle compressibiliteitseffecten. Dat betekent dat de MU aanpak de druk als primaire variabele heeft, zelfs geldig is voor $M > 1$ en de grootheden op een sequentiële wijze uitrekent. De MU aanpak reduceert tot de standaard PC aanpak voor $M = 0$. Tijdsintegratie wordt gedaan middels de θ -methode, en de resulterende lineaire systemen (twee voor de PC aanpak, en drie voor de FC en MU strategieën, tenminste wanneer $\theta > 0$) worden opgelost middels gepreconditioneerde Krylov methoden.

De ruimtelijke discretisatie van de stromingsvergelijkingen op ongestructureerde roosters vormt het onderwerp van Hoofdstuk 4. Wegens het gebruik van een ‘staggered’ rooster is de definitie van ‘linker’ en ‘rechter’ toestanden zoals in standaard gecolocoerde methoden voor compressibele stromingen onmogelijk. Dit verklaart waarom schema’s gebaseerd op ‘approximate Riemann solvers’ niet gebruikt kunnen worden. Echter, discretisatie middels een eindige differentie of eindige volume aanpak voor iedere primaire variabele apart (een gesegegreerde aanpak) ligt voor de hand bij een ‘staggered’ rooster, en dit is de strategie die wij volgen. We maken gebruik van een eindige volume formulering, zodat we aan de Rankine-Hugoniot schokrelaties voldoen, zie ook Appendix D. De impuls vergelijking wordt geïntegreerd over de twee driehoeken die grenzen aan de beschouwde driehoekszijde. Integratie over twee halve driehoeken blijkt een inconsistent schema op te leveren, zie ook Appendix B. Voor de berekening van de convecterende snelheid op de controle volume zijden hebben we een centrale en upwind benadering ontworpen. De geconvecteerde impuls term zelf wordt bepaald middels een eerste orde upwind schema of een centraal schema. Vier verschillende druk gradiënt methoden zijn getest: de pad integraal (PI) methode, de contour integraal (CI) methode, de hulp punt (AP) methode en de vier quadranten (FQ) methode. De discretisatie van de visceuze term wordt behandeld. De continuïteitsvergelijking, de energie vergelijking (in de FC aanpak) en Mach-uniforme druk-correctie vergelijking (in de MU aanpak) worden over iedere driehoek geïntegreerd, en de convectie term wordt benaderd met behulp van een eerste orde upwind of een centraal schema. We behandelen de berekening van de druk (in de FC aanpak) of de enthalpie (in de MU aanpak) door middel van de toestandsvergelijking. De voorgestelde centrale discretisatie van de convectie operator van de scalaire vergelijkingen behoudt de onderliggende symmetrie van de beschouwde operator, terwijl voor de traagheidsterm dit alleen het geval is op regelmatige roosters, zie Appendix C.

Numerieke resultaten voor visceuze incompressibele stromingen (Poiseuille stromingen, de ‘backward facing step’ en de ‘lid-driven cavity’) worden bediscussieerd in

Hoofdstuk 5. Het blijkt dat de PI methode de beste resultaten levert op roosters die opzettelijk onregelmatig gemaakt zijn. De CI methode kan tot ongewenste oscillaties in de druk leiden. De PI methode resulteert in een niet-symmetrische druk-correctie matrix die geen M-matrix is; desondanks bezitten de eigenwaarden van deze matrix, zoals gewenst, slechts een zeer klein imaginair deel in verhouding tot het (positieve) reële deel. Oplossen van de bijbehorende druk-correctie vergelijking wordt gedaan middels Bi-CGSTAB en een zogenaamde ‘lumped’ ILU-preconditioning. Oplossingen voor de stroming over de ‘backward facing step’ en de ‘lid-driven cavity’ worden vergeleken met zowel experimentele als numerieke resultaten.

In Hoofdstuk 6 worden numerieke resultaten voor niet-visceuze compressibele stromingen behandeld. De Burgers vergelijking is geschikt om de discretisatie van de traagheidsterm te testen. We vinden dat het eerste orde upwind schema tot een iets te grote waarde leidt vlak voor de schok die niet in de tijd groeit. De convecterende snelheid dient, om overmatige numerieke diffusie te vermijden, op een centrale manier berekend te worden. Het punt van monotoniciteit wordt behandeld in Appendix A. Ons schema blijkt, wanneer het wordt gebruikt om Riemann problemen te berekenen, naar de entropie oplossing te convergeren zonder daarbij ongewenste modes te genereren. De nauwkeurigheid is vergelijkbaar met die van de algemeen geaccepteerde Roe en AUSM schema’s. De PI methode blijkt iets nauwkeuriger te zijn dan de andere druk gradiënt schema’s. Centrale discretisatie van de traagheidsterm leidt tot een matrix die vrijwel de symmetrie van de operator behoudt: de eigenwaarden hebben, vergeleken met het imaginaire deel, een klein reëel deel. Een roosterverfijningstudie voor subsone stroming in een kanaal met een 10% sinusvormige vernauwing toont aan dat ons schema vrijwel even goede resultaten levert op willekeurige ongestructureerde roosters als op Courant roosters, en dat de orde van de ruimtelijke discretisatie van het eerste orde upwind schema één is. Het gebruik van een ‘staggered’ gestructureerd schema levert, met hetzelfde aantal knooppunten op de rand, tot iets minder diffusieve resultaten. Supersone stroming in een kanaal met een 4% cirkelvormige vernauwing toont het vermogen om supersone stromingen te berekenen: er wordt een goede overeenkomst met andere gepubliceerde resultaten behaald. Stromingen met verschillende waarden voor het aanstroom Mach getal ($M_\infty = 0, 10^{-3}, 10^{-2}, 10^{-1}, 0.63, 0.8, 1.2$) om een NACA 0012 profiel worden benut om de nauwkeurigheid en efficiëntie van ons schema voor incompressibele en compressibele stromingen te bestuderen. De MU formulering blijkt, over het beschouwde bereik van Mach getallen, veel efficiënter te zijn dan de FC aanpak. Dit komt omdat de druk in de MU aanpak impliciet genomen wordt, wat het gebruik van veel grotere tijdstappen toelaat, hetgeen op zijn beurt weer resulteert in een kleiner aantal tijdstappen om convergentie te bereiken. Binnen de MU formulering wordt een snelle convergentie bereikt voor Courant getallen in de orde van enkele honderden, terwijl de FC aanpak instabiel wordt voor Courant getallen groter dan 2 à 3. Binnen de FC aanpak wordt het grootste deel van de CPU

tijd gebruikt om de matrix elementen te berekenen, terwijl binnen de MU aanpak de meeste CPU tijd in de lineaire solver zit. Zoals gebruikelijk voor methoden die de dichtheid als primaire variabele hebben, wordt de FC aanpak minder nauwkeurig wanneer het Mach getal laag is; dit is niet het geval voor de MU aanpak die de druk als primaire variabele heeft. Een voorbeeld van een stroming waarbinnen zowel incompressibele als compressibele gebieden een groot deel van het domein omvatten is dat van een stroming in een convergerende-divergerende tuit. Berekeningen resulteren in een goede overeenkomst met gestructureerde rooster resultaten en theorie. Een Mach-uniforme efficiëntie wordt behaald, omdat de CPU tijd per tijdstap slechts afhangt van de grootte van de tijdstap en niet van de mate van contractie van de tuit.

Ivo Wenneker

Contents

Summary	vii
Samenvatting	xi
1 Introduction	1
1.1 The processes involved in using CFD	2
1.1.1 General outline	2
1.1.2 Bottleneck: grid generation	3
1.1.3 Mach-uniform methods	4
1.2 Objectives of this thesis	4
2 Computation of flows using unstructured grids	7
2.1 Equations of motion	7
2.1.1 Basic equations of fluid dynamics	8
2.1.2 Euler equations	8
2.1.3 Incompressible flow equations	9
2.2 Types of grids	10
2.2.1 Structured grids	10
2.2.2 Unstructured grids	11
2.2.3 Other grid types	14
2.3 Compressible flow solvers on unstructured grids	15
2.4 Incompressible flow solvers on unstructured grids	17
2.4.1 Staggered and colocated grids	18
2.4.2 Colocated schemes	18
2.4.3 Staggered schemes	21
2.4.4 Closing remarks	31
3 Solution strategy	33
3.1 Time integration	34
3.2 Pressure-correction approach for incompressible flows	35
3.3 Sequential update procedure for fully compressible flows	37
3.4 The need for unified methods	38

3.4.1	Difficulties with the zero Mach number limit	38
3.4.2	Proposed remedies	41
3.5	Mach-uniform solution algorithm	42
3.5.1	Dimensionless formulation	43
3.5.2	Mach-uniform sequential update procedure	44
3.6	Linear solver	48
3.6.1	Krylov subspace methods	48
3.6.2	Preconditioning	50
4	Spatial discretization on unstructured staggered grids	51
4.1	Unstructured staggered grids	52
4.2	Discretization of the momentum equation	53
4.2.1	Choice of control volume	54
4.2.2	Discretization of the time derivative	55
4.2.3	Discretization of the inertia term (two triangles)	55
4.2.4	Discretization of the inertia term (two half triangles)	59
4.2.5	Discretization of the pressure gradient	63
4.2.6	Discretization of the viscous term	71
4.3	Discretization of the continuity equation	75
4.4	Discretization of the energy equation	76
4.5	Discretization of the Mach-uniform pressure-correction equation	78
4.6	Evaluation of the equation of state	80
4.6.1	Mach-uniform approach	80
4.6.2	Fully compressible flow approach	81
4.7	Postprocessing	83
5	Results for viscous incompressible flows	85
5.1	Poiseuille flow	85
5.1.1	Problem definition	85
5.1.2	Non-smooth grids	86
5.1.3	Stretched grids	89
5.1.4	Properties of the viscous and Laplacian matrices	95
5.1.5	Conclusions	95
5.2	Backward facing step	97
5.2.1	Problem definition	97
5.2.2	Solution for $Re = 100$	99
5.2.3	Solution for $Re = 389$	101
5.2.4	Solution for $Re = 500, 600, 700$ and 800	103
5.2.5	Reattachment length	103
5.2.6	Conclusions	104
5.3	Lid-driven cavity flow	105
5.3.1	Problem definition	105

5.3.2	Results	106
5.3.3	Discussion and conclusions	108
6	Results for inviscid flows	111
6.1	Burgers equation	112
6.1.1	Solution with a shock wave	112
6.1.2	Solution with a shock wave and expansion fan	114
6.2	Riemann problems	115
6.2.1	Choices w.r.t. energy variable and interpolation methods	116
6.2.2	Sod's shocktube problem	117
6.2.3	Test case of Lax	119
6.2.4	Mach 3 test case	120
6.2.5	Riemann problems on a less smooth grid	121
6.2.6	Stationary contact discontinuity and slip flows	121
6.3	Flow in a channel with a bump	124
6.3.1	Incompressible flow	124
6.3.2	Supersonic flow	130
6.4	Flows around the NACA 0012 airfoil	132
6.4.1	Low subsonic flow around the NACA 0012 airfoil	133
6.4.2	Subcritical flow around the NACA 0012 airfoil	134
6.4.3	Transonic flow around the NACA 0012 airfoil	135
6.4.4	Supersonic flow around the NACA 0012 airfoil	137
6.4.5	Mach uniform efficiency	138
6.5	Supersonic flow over a circular blunt body	142
6.6	Converging-diverging nozzle	143
6.7	Conclusions	146
A	Monotonicity considerations	149
A.1	Requirements for convergence	149
A.2	Monotone discretization of the Burgers equation on a 1D grid	150
A.3	Monotone discr. of the Burgers eq. on an unstr. stagg. grid	151
A.4	Definition of the Courant number	153
A.4.1	Courant number for the continuity and energy equation	153
A.4.2	Courant number for the momentum equation	155
A.4.3	Courant number for the Euler equations	156
B	Consistent discretization of the inertia term	157
C	Symmetry-preserving discretization	159
C.1	What is a symmetry-preserving discretization?	159
C.2	Symm.-pres. properties of the unstr. stagg. scheme	164

D Conservation properties of the discretization	167
D.1 Conservative discr. of the continuity and energy eq.	168
D.2 Conservative discretization of the momentum equation	169
D.3 Example of a result computed with a nonconservative scheme	171
Bibliography	173
Curriculum vitae	185
Dankwoord	187

Chapter 1

Introduction

Problems involving flows are encountered in many branches of engineering science. Examples are flows in turbomachinery, arteries, seas and rivers, and around aircraft and ships. The mathematical formulation of the laws that govern the motion of a fluid is known already for well over a century. This formulation consists of some thermodynamical relations and a set of coupled partial differential equations, the so-called Navier-Stokes equations, that describe conservation of mass, momentum and energy for a fluid. Appropriately defined boundary conditions and initial conditions complete the problem definition. In general, the resulting mathematical problem is far too difficult to be solved by analytical means. Therefore, in the past one had to rely heavily on experiments or greatly simplified mathematical models for the majority of flow problems. However, experiments are often too expensive, difficult, dangerous or even impossible to perform. On the other hand, the use of methods that solve the governing equations by numerical means is an obvious alternative. Thanks to the tremendous increase in computational power in the last decades, this approach has gained substantial significance. The new scientific discipline that has evolved is called computational fluid dynamics (CFD). Nowadays, CFD has, besides the more traditional experimental and analytical approaches, become an indispensable tool for the fluid dynamicist. The existence of many commercial, freeware and in-house computer codes suited for doing CFD attest to this fact.

In the majority of cases, one basically divides the flow domain into small cells (grid generation), in which the primary unknowns are sought, and one replaces differentials by differences (discretization). The core of a CFD code is formed by the discretization, i.e. the way in which the flow equations are formulated mathematically and are approximated. The numerics involved in this process have to ensure that the computed solution approximates the exact solution of the mathematical model as well as possible. Of course, aspects like robustness, efficiency and ease of implementation also play a prominent role in devising good numerical algorithms.

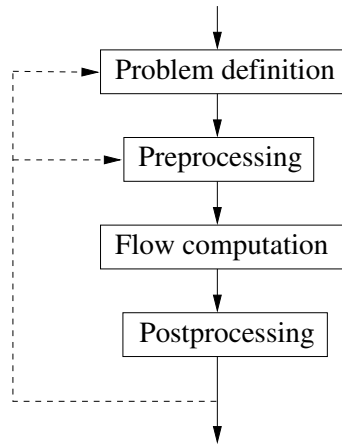


Figure 1.1: Flow diagram describing the processes involved in using CFD.

1.1 The processes involved in using CFD

1.1.1 General outline

The CFD-user needs a clear definition of the problem, including values for several flow parameters, specification of the geometry of the flow domain and formulation of the governing equations to be solved. After the problem has been defined, preprocessing can start, see Figure 1.1. Preprocessing is the process of entering the problem specification and the geometry definition into the computer in such a way that the actual flow computation can start. In general, one needs to generate a grid and write some inputfiles that indicate the numerical schemes to be used.

After preprocessing is finished, flow computation takes place. The aim is to find numerical values for the solution of the flow equations in all grid cells. Of the processes distinguished in Figure 1.1, this is usually the most computing intensive part. The flow computation leads to huge amounts of data, that have to be transformed (post-processed) such that interpretation of the results becomes feasible. One important feature in postprocessing is visualization, without which a good understanding of the computed solution is impossible. Also the evaluation of derived quantities like the Mach number, shear stress, lift, drag, and so on, forms an indispensable part of postprocessing.

After interpretation of the results, the user has to decide whether he or she is satisfied with the results, or perhaps wants to redo the computation. The latter might be the case if, for example, the grid or the employed physical model are considered 'not good enough'. This is indicated by the dashed lines in Figure 1.1.

1.1.2 Bottleneck: grid generation

All processes briefly touched upon above are major disciplines in themselves. We now make a few remarks about grid generation, because this has become a major bottleneck. Grid generation is the division of the domain of interest into small cells, in which the unknowns are sought. Thanks to the advances in computer technology and numerical algorithms, the problems that CFD can handle have become more and more complex. Just to mention one example, computation of transonic flow around a complete 3D aircraft, for the first time done in 1986 [54], has now become daily routine in aerospace industry. Of course, such a leap forward in technology has its repercussions on the processes distinguished in Figure 1.1. Inevitably, automation of these processes is one of them. Experience has shown that it is impossible to gain a satisfactory reduction of human labor involved in generation of structured grids. As a consequence, it is not uncommon that most of the CFD turn-around time is devoted to generation of structured grids.

Another disadvantage of structured grids consist of difficulties associated with local or adaptive grid refinement. Often, the important flow features occur in a relatively small part of the considered domain. This demands locally a fine grid. Without losing accuracy, much computing time can be saved by keeping the grid coarse in the other parts of the domain. However, this is hard to realize on structured grids. Closely related is the issue of local grid adaptation: can we, during the computation and without user-interference, adapt the grid locally in order to increase the accuracy significantly (i.e. capture all relevant flow details) while not increasing the computation time drastically? Again, this is difficult to realize on structured grids.

These difficulties have led in many areas of CFD to the emergence of unstructured grids, which will be defined later. Generation of unstructured grids can be automated to a larger extent than generation of structured grids. In addition, local grid refinement and grid adaptation are much easier to realize on unstructured grids. It must be noted that discretization of the flow equations on unstructured grids is considered to be more difficult than on structured grids. Furthermore, the data structures needed when dealing with unstructured grids are more complicated to program, and as a result of the required indirect addressing computational costs are higher. The computational costs are also increased by the absence of a bandstructure in the matrices (relevant when implicit methods are used to discretize in time), which prevents the use of certain efficient iterative methods.

Of course, the accumulated experience in using structured grids is of great help for designing discretization schemes for unstructured grids. Consequently, many schemes, originally developed for structured grids, have been modified for use on unstructured grids.

1.1.3 Mach-uniform methods

It is well-known that the efficiency and accuracy of methods designed to compute compressible flows (Mach number $M > 0.2$) deteriorate drastically when M decreases below 0.2. On the other hand, when the Mach number remains uniformly small (say, below 0.2), an accurate and useful approximation is to consider the flow as incompressible. This observation has led to the development of computational methods exclusively suited for incompressible flows. However, neither class of methods is capable of computing flows in domains in which incompressible flow subregions as well as compressible flow subregions occur simultaneously. Two strategies to deal with such flows can be distinguished: extension of compressible flow methods to incompressible flows, or extension of incompressible flow schemes to the compressible case.

In this discussion a related issue is of importance. In (almost) all compressible flow methods, the variables are located in the same nodes in the grid cells. This is called a collocated grid. Such a distribution of the variables is also not uncommon in incompressible flow CFD, but in this field several arguments tend to favor another option, namely that the velocity and scalar variables should be located at different grid locations. When this is the case, we say that the grid is staggered.

1.2 Objectives of this thesis

The use of a staggered positioning of the variables on structured grids has led to accurate algorithms to compute incompressible flows. Take for example the MAC scheme on Cartesian grids [44], which is also successfully applied for DNS (direct numerical simulation) of turbulence, see [68, 122, 123], and LES (large eddy simulation) [1]. Computation of incompressible flow is also feasible on staggered curvilinear grids, as demonstrated for instance in [130, 131] and references quoted therein. The inclusion of turbulence models on staggered grids is described in [133]. In addition, staggered schemes have been found capable of computing compressible flows in a fashion that is uniformly accurate and efficient in the Mach number, see [8, 9, 111, 132], and even arbitrary equations of state can be dealt with [110].

In this thesis, a new method to compute flows using unstructured staggered grids is described. The method is capable of computing flows with an efficiency and accuracy that is uniform in the Mach number, from $M = 0$ to $M > 1$. The remainder of this thesis is subdivided as follows:

- In Chapter 2, the flow equations are introduced, and a literature survey of compressible and incompressible flow solvers on unstructured grids is given.
- In Chapter 3, three different time-marching solution procedures are discussed: the pressure-correction approach for incompressible flows, a sequential update

procedure for compressible flows and a Mach-uniform sequential update procedure for both compressible and incompressible flows.

- The spatial discretization of the flow equations on unstructured staggered grids forms the subject of Chapter 4.
- Numerical results for viscous incompressible flows (Poiseuille flow, the backward facing step and the lid-driven cavity) are discussed in Chapter 5.
- In Chapter 6, numerical results for inviscid compressible flows are given. Various test cases (Burgers equation, Riemann problems, flow in a channel with bump, flows around the NACA 0012 airfoil, supersonic flow over a circular blunt body and flow in a converging-diverging nozzle) are considered.
- Several appendices are included. In Appendix A, considerations with respect to monotonicity of the scheme are given. Numerical tests of the consistency of the discretized inertia term are described in Appendix B. The issue whether certain underlying symmetries of the differential operators carry over to the discretization is addressed in Appendix C, and in Appendix D it is shown that the discretization conserves mass, energy and momentum.

Chapter 2

Computation of flows using unstructured grids

In this chapter a survey of the literature on the various unstructured grid methods is given. In Section 2.1 the flow equations that will be used throughout this thesis are summarized. An overview of the various grid types, including some remarks on the generation of unstructured grids, is given in Section 2.2. The methodology used in compressible flow solvers on unstructured grids is briefly discussed in Section 2.3. Some incompressible flow solvers on unstructured grids are treated in Section 2.4, including a more extensive overview of the staggered schemes among them.

2.1 Equations of motion

The equations that describe the motion of fluids (liquids and gases) are, together with some thermodynamical relations, mathematical formulations of the conservation laws for mass, momentum and energy. A derivation of these equations can be found in many textbooks, for instance the one by Batchelor [7]. We will introduce these equations in Section 2.1.1. One important simplified model is that of inviscid compressible flow. Such type of flow is governed by the Euler equations, and these form the subject of Section 2.1.2. Another important simplified model is that of incompressible flow. In such a flow, the density of each material particle in the fluid remains constant during motion. The incompressible flow equations are addressed in Section 2.1.3. In many occasions, a flow can be considered as incompressible when the Mach number in the whole flow domain is smaller than, say, 0.2. When the Mach number is larger than 0.2, the compressible flow equations must be employed. This division has become rather rigid in the field of CFD, which is due to the inherently different nature that the flow equations possess in both cases. We will come back to this in Section 3.4.

2.1.1 Basic equations of fluid dynamics

The basic equations of fluid dynamics are the ones that govern the motion of fluids, and they are given by:

$$\frac{\partial \rho}{\partial t} + (u^\alpha \rho)_{,\alpha} = 0, \quad (2.1)$$

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = \sigma_{,\beta}^{\alpha\beta} + \rho f^\alpha, \quad (2.2)$$

$$\frac{\partial(\rho E)}{\partial t} + (u^\alpha \rho E)_{,\alpha} = (u^\alpha \sigma^{\alpha\beta})_{,\beta} + (k T_{,\alpha})_{,\alpha} + \rho u^\alpha f^\alpha + \rho q. \quad (2.3)$$

Cartesian tensor notation and the summation convention are applied, meaning that we have $p_{,\alpha} = \partial p / \partial x^\alpha$ for each Cartesian coordinate x^α , $v_{,\alpha}^\alpha$ for the divergence of vector v^α (hence, $v_{,\alpha}^\alpha = \nabla \cdot \mathbf{v}$) and $v^\alpha w^\alpha$ for the inner product $\mathbf{v} \cdot \mathbf{w}$. Equation (2.1) is the mathematical formulation for conservation of mass and is often called the continuity equation. The density is indicated by ρ , the velocity vector by u^α and t refers to time. Equation (2.2) is the second law of Newton, describing conservation of momentum $m^\alpha = \rho u^\alpha$ in a fluid. Here the stress tensor is given by the constitutive relation for a Newtonian fluid:

$$\sigma^{\alpha\beta} = -p \delta^{\alpha\beta} + 2\mu(e^{\alpha\beta} - \frac{1}{3}\Delta \delta^{\alpha\beta}), \quad (2.4)$$

where $\delta^{\alpha\beta}$ denotes the Kronecker delta and μ the dynamic viscosity coefficient. The rate of strain tensor is defined by

$$e^{\alpha\beta} = \frac{1}{2}(u_{,\beta}^\alpha + u_{,\alpha}^\beta), \quad (2.5)$$

and

$$\Delta = e^{\alpha\alpha} = u_{,\alpha}^\alpha. \quad (2.6)$$

Body forces, e.g. gravity, are put in the term f^α . We will neglect body forces in this thesis. Conservation of energy is expressed by (2.3), in which E denotes the total energy per unit mass, T the temperature, k the thermal conductivity and q the added heat per unit of mass per unit time. In this thesis, we will restrict ourselves to the case in which $q = 0$.

2.1.2 Euler equations

With respect to compressible flows, we will restrict ourselves in this thesis to the motion of ideal fluids, i.e. flows in which friction and heat conduction are neglected: $\mu = 0$, $k = 0$. The Euler equations that govern this type flows can easily be derived

from (2.1)–(2.3), and are given by:

$$\frac{\partial \rho}{\partial t} + (u^\alpha \rho)_{,\alpha} = 0, \quad (2.7)$$

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = -p_{,\alpha}, \quad (2.8)$$

$$\frac{\partial(\rho E)}{\partial t} + (u^\alpha \rho H)_{,\alpha} = 0. \quad (2.9)$$

The Euler equations are often abbreviated as follows:

$$\frac{\partial U}{\partial t} + \nabla \cdot F = 0, \quad (2.10)$$

where $F = F(U)$ is called the flux vector function and U the vector of conserved variables. The following thermodynamic relations, restricting ourselves to calorically perfect gases, will be used frequently:

$$H = h + \frac{1}{2}u^2, \quad (2.11)$$

$$E = e + \frac{1}{2}u^2, \quad (2.12)$$

$$h = \gamma e, \quad (2.13)$$

$$\rho H = \rho E + p, \quad (2.14)$$

where h is the enthalpy, e the internal energy and $u^2 = u^\alpha u^\alpha$. For the specific heat ratio the value $\gamma = 7/5$ will be used throughout. The system of equations is closed by the equation of state:

$$p = (\gamma - 1)\rho e. \quad (2.15)$$

For future purposes, we define the Mach number as

$$M = u/a, \quad (2.16)$$

where u is the local flow velocity, and

$$a = \sqrt{(\gamma - 1)h} \quad (2.17)$$

is the local speed of sound.

2.1.3 Incompressible flow equations

The incompressible flow equations govern the motion of a fluid under the assumption of negligible density variations in each fluid particle. This approximation holds to a fairly good extent as long as $M < 0.2$, heat addition is limited and acoustics is absent. In contrast with the compressible flow case, we do not neglect the viscous forces in

the incompressible flow equations. The incompressible Navier-Stokes equations are given by, assuming constant viscosity μ :

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = -p_{,\alpha} + \mu u_{,\beta\beta}^\alpha, \quad (2.18)$$

$$u_{,\alpha}^\alpha = 0. \quad (2.19)$$

Equation (2.18) is the momentum equation, and the kinematic constraint (2.19) is the continuity equation.

2.2 Types of grids

Because of the geometric complexity of the domains in which many flows take place, grid generation is a major issue in CFD. By grid generation we mean the subdivision of the domain into small cells. The resulting set of cells is called the (computational) grid. Two primary types of grids, namely structured and unstructured, are discussed subsequently.

2.2.1 Structured grids

A grid is called structured if all interior cell vertices belong to the same number of cells and if the grid can be mapped onto a rectangle (in 2D) or a block (in 3D). As a consequence, each cell can be labelled by a set of integers (i, j) in 2D or (i, j, k) in 3D, with $i = 1, \dots, I$, $j = 1, \dots, J$ and $k = 1, \dots, K$. When the grid cells conform to the domain boundaries, the grid is called boundary-fitted. In Cartesian grids, with cells being rectangles or blocks, the domain boundary usually does not coincide with the cell faces, making accurate implementation of the boundary conditions difficult. Discretization of the flow equations in the interior, on the other hand, is much easier on Cartesian grids than on boundary-fitted grids. Grid topologies in use include single-block and multiblock with matching or overlapping block interfaces, the latter approach being called the chimera or overset approach. A single-block grid can be mapped onto a Cartesian grid. In a multiblock grid, the element connectivity is inherently structured within each block, but between blocks the connectivity is unstructured. The multiblock approach allows structured grids to be used for complex geometries. A structured grid approach offers advantages in solution algorithm efficiency and ease of datastructure-handling. A drawback of a multiblock structured grid approach is the significant human time required to generate a complete grid in complicated domains. Another drawback of structured grids consists of the difficulties associated with local or adaptive grid refinement. Beforehand, generally one does not know exactly where the grid should be fine in order to capture all relevant flow details, nor how fine it should be. It might happen that, after completing a flow computation, one arrives at the conclusion that the grid was too coarse in one or more regions of the flow domain. This leads to the necessity to adapt the

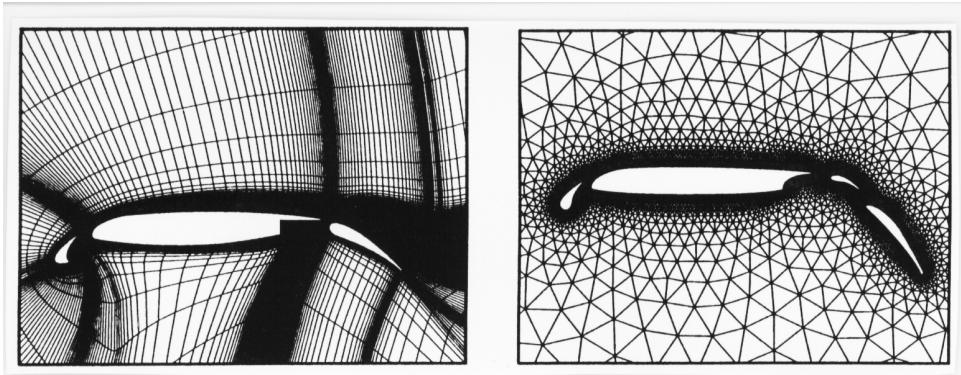


Figure 2.1: Multiblock structured grid (left) and an unstructured grid for a multi-element airfoil configuration. Picture is taken from [114].

grid and to restart the computation, thereby increasing significantly the amount of labor and CPU time. A related disadvantage of the structured grid approach is that local refinements are ‘transported’ to the far field region where refinement is not needed, leading to more grid points than are strictly necessary. This is illustrated in Figure 2.1.

2.2.2 Unstructured grids

Unstructured grids subdivide the domain into simple, usually triangular (in 2D) or tetrahedral (in 3D), elements with no implied connectivity. In comparison to multiblock structured grids, an unstructured grid offers considerably more geometric flexibility and, because its generation can be automated to a much larger extent, a substantial reduction of the human labor time required to generate a grid. In addition, local and adaptive grid refinement (and de-refinement, if necessary) can be realized with much more ease. Though used already for a long time in finite element methods, only fairly recently the CFD community, traditionally more biased towards finite volume methods, has embraced unstructured grid technology. Nice surveys on unstructured mesh generation technology can be found on the internet [85] or in the paper by Mavriplis [75]. In view of the arguments given above, the problem of unstructured mesh generation is largely one of designing algorithms that are automatic, robust, and yield suitable element shapes and distributions for the flow solver. Most unstructured grid generation techniques, both for 2D and 3D applications, fit into one of the three main categories to be addressed below.

Delaunay triangulation methods

Very popular among the triangle and tetrahedral meshing techniques are those utilizing the Delaunay criterion. The Delaunay criterion, which is sometimes called the ‘empty sphere’ property, states that no circumsphere of a tetrahedron in the grid can

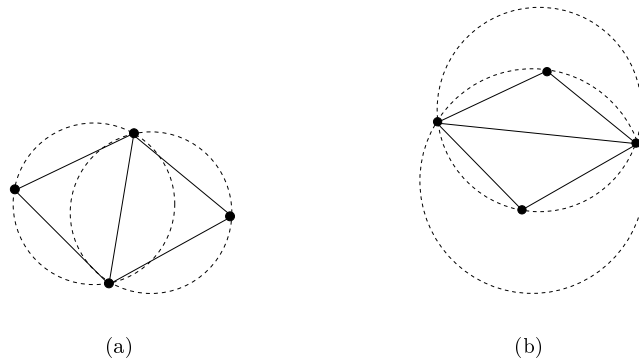


Figure 2.2: Example of the Delaunay criterion: (a) satisfies the criterion while (b) does not.

contain a grid point other than its four constitutive vertices. Figure 2.2 illustrates the criterion in two dimensions. Note that the Delaunay criterion in itself is not an algorithm for generating a mesh; it merely provides a criterion for connecting a set of existing points in space. A frequently used point-insertion algorithm is the one developed by Bowyer and Watson [10, 126]. The main disadvantage of Delaunay triangulation techniques relate to their inability to guarantee boundary integrity. Edge and face-edge swapping, in two and three dimensions, respectively, have been developed for locally modifying Delaunay meshes to conform to boundaries. In boundary layers, stretched obtuse triangles that contain one large angle and two small angles are, for accuracy reasons, to be avoided, while stretched right-angle triangles are to be preferred for efficiency reasons. Delaunay triangulations, which maximize the minimum angles of any triangulation (the so-called maxmin property), tend to produce equiangular triangulations and are thus ill suited for the construction of highly stretched triangular elements. Hence, minmax triangulations, which minimize the maximum angle of the triangulation, are to be preferred, but such triangulations are usually not of Delaunay type. Connecting the circumcenters of the Delaunay triangulation results, as illustrated in Figure 2.3, in a Voronoi polygon. The collection of Voronoi polygons forms a Voronoi tessellation. The edges of either mesh are perpendicular to the faces of the other. This so-called dual property is explicitly used in the covolume technique and the discretization as proposed by Perot, both to be discussed in Section 2.4.3.

Advancing front techniques

Another very popular triangle and tetrahedral mesh generation algorithm is the advancing, or moving front method [71, 72, 73, 74]. In this method, the triangles or tetrahedra are built progressively inward from the domain boundaries. As demon-

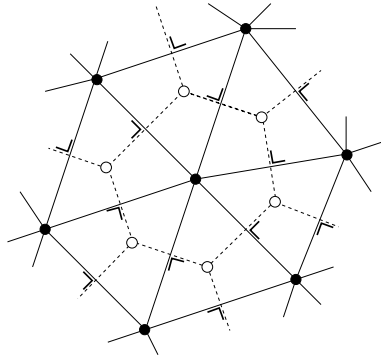


Figure 2.3: Delaunay triangulation (continuous line) and the associated Voronoi tessellation (dashed line). The vertices and the circumcenters are indicated by the symbols \bullet and \circ , respectively.

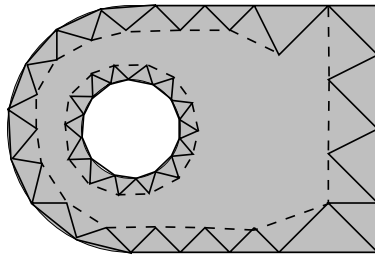


Figure 2.4: Example of the advancing front method, with two layers of triangles already in place. The dashed line indicates the active front.

strated in Figure 2.4, an active front is maintained where new elements are formed. As the algorithm progresses, the fronts will advance, in a judiciously chosen manner, to fill the remainder of the domain with elements. Advancing front methods generally result in smooth, high-quality triangulations in most regions of the flow domain. However, difficulties may be encountered in regions where fronts must be merged.

Octree technique

With the octree technique [97, 136], squares are recursively subdivided, as is illustrated in Figure 2.5, until the desired resolution is reached. Irregular cells are created where cubes intersect the boundary. Triangles are generated from both the irregular cells on the boundary and the internal regular cells.

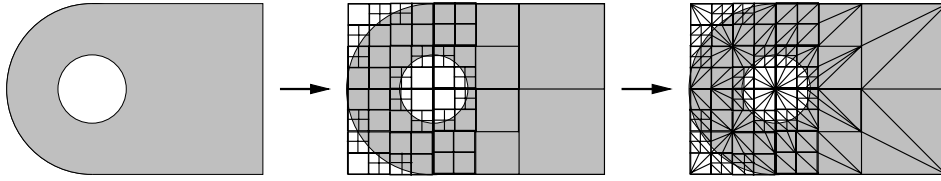


Figure 2.5: Quadtree (middle) and octree (right) decomposition of a simple 2D object.

2.2.3 Other grid types

In order to resolve thin boundary layers and wakes, stretched grids are indispensable in order to maintain efficiency when computing high-Reynolds number flows. It turns out to be difficult to obtain accurate flow solutions on highly stretched tetrahedral cells. This has led to an interest in hybrid grids - with tetrahedral cells away from the body surfaces, while using prismatic cells admitting high aspect ratios near body surfaces. The layers in the region containing the prisms are distributed such as to resolve the boundary layer, while the regular distribution near the surface may result in a local cancellation of truncation errors. To give one particular example, development of hybrid grid technology has been an essential part of the Brite/Euram FASTFLO II project, which aims were stated as follows [109]:

The industrial objective of the FASTFLO II project is to develop a common, automated CFD system that satisfies two basic requirements for industrial application:

1. *CFD-problem-turnaround-time of a day to a week (or less) for very complex geometries (including automatic grid generation).*
2. *High accuracy of aerodynamic entities (forces, pressure, skin friction, etc.) on Navier-Stokes level.*

This demonstrates the level of maturity that hybrid grid technology and flow solvers have reached nowadays.

Yet other approaches that have emerged recently are the so-called particle methods and meshless methods. In particle methods [64, 65, 137], each particle is followed in a Lagrangian manner which facilitates the analysis of moving boundaries and interfaces, since fixed grids are not necessary. Independently, meshless methods, utilizing the idea of a polynomial interpolant which fits a number of points minimizing the distance between the interpolated function and the value of the unknown point, have been developed [27]. A combination of these two methods is described in [26]. These methods will not be considered further in this thesis.

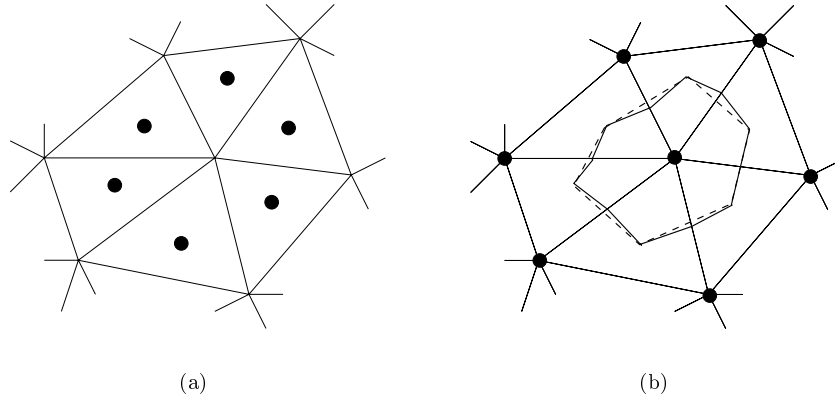


Figure 2.6: Cell-centered (a) and vertex-centered (b) unstructured collocated grid, with the symbol \bullet indicating the location of the variables. The dotted and continuous lines around the central vertex in (b) represent the centroid and median dual, respectively.

2.3 Compressible flow solvers on unstructured grids

In this section a brief survey of the current state of the art in the field of compressible flow solvers on unstructured grids is given. Good review articles are the ones written by Venkatakrisnan [121] and Mavriplis [75]. In the present section, emphasis is put on issues related to finite volume discretization of the inviscid flow equations, omitting subjects like finite element discretization, turbulence modeling, multigrid, parallelization, adaptive grids, and so on.

As far as we know, all compressible flow solvers on unstructured grids use a collocated grid, which means that all variables reside in the same grid nodes. Two choices prevail as to where to locate the variables. In the vertex-centered approach, the variables are stored in the vertices of the grid, whereas in the cell-centered approach they are stored in the centroids of the cells. In Figure 2.6 these approaches are illustrated for a 2D unstructured grid. Most compressible flow solvers are based on the finite volume technique, which means that first the flow equations (2.10) are integrated over a suitably chosen control volume (CV). Application of Gauss's theorem transforms the fluxterm into a summation of fluxes over the CV boundaries:

$$\begin{aligned}
 \int_{CV} \frac{\partial U}{\partial t} d\mathbf{x} + \int_{CV} \nabla \cdot \mathbf{F} d\mathbf{x} &= \frac{d}{dt} \int_{CV} U d\mathbf{x} + \oint_{\partial CV} \mathbf{F} \cdot \mathbf{n} d\Gamma = \\
 &= \frac{d}{dt} \int_{CV} U d\mathbf{x} + \sum_{e \in (CV)} (F_e \cdot \mathbf{n}_e) l_e = \mathbf{0}, \quad (2.20)
 \end{aligned}$$

where \mathbf{n}_e is the outward normal at CV boundary e , which has length (area) l_e . The discrete set of equations (2.20) satisfies the underlying conservation properties exactly, and as a desirable consequence the jump conditions over discontinuities will be satisfied [66]. The grid cell forms the CV for the cell-centered approach, whereas usually for the vertex-centered approach the CV is formed by the centroid dual or median dual. The centroid dual is constructed by joining the centroids neighboring the vertex under consideration, and for construction of the median dual the mid-points of the adjacent faces are also taken into account, see Figure 2.6b. There seems to be no clear-cut choice between cell-centered and vertex-centered schemes [121]. The fluxes have to be evaluated at the CV boundaries, and depend on the state U at both sides. The quest for accurate and efficient flux evaluation techniques has given birth to a large variety of methods, most of which can be considered as a form of flux vector splitting (FVS) or flux difference splitting (FDS). The FVS approach relies on the homogeneity property $F = AU$ of the Euler equations, and selects the upwind directions by considering the eigenvalues of the Jacobian $A = A(U) = \partial F / \partial U$. Well-known examples of FVS schemes are the van Leer scheme [118] and AUSM (advection upstream splitting method) [70]. The FDS methods originate from the Godunov method [35], and aim at approximating the fluxes by basing a local Riemann problem on the flux difference at the CV boundary. The FDS methods most frequently encountered are the Roe [92] and Osher [28, 83, 84] schemes. In order to compute accurate solutions while avoiding excessively fine grids, the spatial discretization needs to be of (at least) second order accuracy. However, second (or higher) order accurate schemes result easily in the appearance of spurious wiggles near steep gradients when dealing with convection-dominated flows. Van Leer [117] devised the MUSCL (monotonic upstream-centered scheme for conservation laws) concept to obtain monotonicity-preserving schemes for conservation laws. This concept relies on a piecewise-polynomial reconstruction procedure, and monotonicity is enforced by using nonlinear functions called limiters. This boils down to maintaining second (or higher) order accuracy in regions where the solution is smooth, while reverting back to a first order monotone scheme in the vicinity of discontinuities. In this manner, unphysical oscillations can be avoided while keeping a high level of accuracy. For a good introduction to flux evaluation and MUSCL techniques, the book by Toro [104] is recommended. It was not until 1989 that the MUSCL ideas were generalized successfully to unstructured grids. In that year Barth and Jespersen [6] came up with a truly multidimensional limiter, ensuring that the reconstructed distribution in the control volume is bounded everywhere by the values of its neighbors and the considered point itself. Since then, various improvements have been proposed, and in view of recent publications [5, 51, 55, 82, 86, 120], a generally accepted approach has not yet emerged.

Another promising approach is the discontinuous Galerkin finite element method (DGFEM) [16, 17, 18, 112], which is a mixture of a finite volume and finite element

method. In the DGFEM, the flow field in each element is locally approximated by a polynomial, which is discontinuous over the element faces. No interpolation is necessary to determine the flow state at the element faces in the flux calculation, since information about the flow state can be directly obtained from this polynomial expansion. An approximate Riemann solver is used to compute the flux at the element faces. The only additional information from neighboring elements that is needed is the mean flow state, to be used in the slope limiter. In this way an almost completely local scheme is obtained, which is advantageous for mesh adaptation and parallelization.

Yet other interesting alternatives are the essentially non-oscillatory (ENO) schemes [46] and the weighted ENO (WENO) schemes [32, 50]. Both ENO and WENO schemes use the idea of adaptive stencils in the reconstruction procedure that are based on the local smoothness of the numerical solution to automatically achieve a high order (at least third order) of accuracy and non-oscillatory results near discontinuities. ENO schemes use one (optimal in some sense) out of many candidate stencils when doing the reconstruction, while WENO schemes take a convex combination of all the candidate stencils, each being assigned a nonlinear weight which depends on the local smoothness of the numerical solution based on that stencil. WENO schemes improve upon ENO schemes in robustness, smoothness of fluxes, steady state convergence and efficiency. For a comparison between WENO schemes and DGFEM methods, we refer to [98].

2.4 Incompressible flow solvers on unstructured grids

In this section, a brief survey of incompressible flow solvers on unstructured grids is given. We will restrict ourselves to finite volume methods. Two possible ways of locating the variables in the grid can be distinguished, leading to the so-called staggered and colocated grids, see Section 2.4.1. Some flow solvers using colocated grids are discussed in Section 2.4.2, and a more detailed description of flow solvers on staggered grids is given in Section 2.4.3. Integration of the flow equations (2.18)–(2.19) over a control volume (CV) results in:

$$\sum_{e(\text{CV})} (\mathbf{u} \cdot \mathbf{n})_e l_e = 0, \quad (2.21)$$

$$\Omega_{\text{CV}} \frac{d\mathbf{m}_{\text{CV}}}{dt} + \sum_{e(\text{CV})} (\mathbf{u} \cdot \mathbf{n})_e \mathbf{m}_e l_e = - \sum_{e(\text{CV})} p_e \mathbf{n}_e l_e + \sum_{e(\text{CV})} \mu (\nabla \mathbf{u} \cdot \mathbf{n})_e l_e. \quad (2.22)$$

The summation over $e(\text{CV})$ is over the CV faces, Ω_{CV} stands for the area of the CV, \mathbf{n}_e is the outward unit normal at face e and l_e is its length. With $\nabla \mathbf{u} \cdot \mathbf{n}$ we mean $u_{,\beta} n^\beta$.

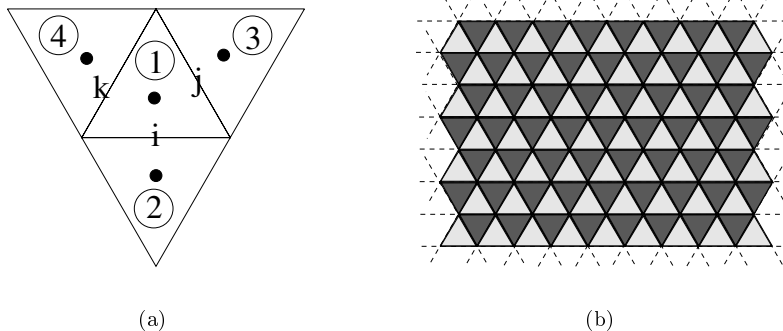


Figure 2.7: Colocated positioning of the variables (a) and resulting odd-even decoupling of the pressure (b).

2.4.1 Staggered and colocated grids

A straightforward discretization of the incompressible flow equations with a colocated positioning of the variables leads to an unphysical odd-even decoupling of the pressure in the numerical solution, as will be discussed in Section 2.4.2. This problem can be alleviated by adding a small artificial diffusion term in the discretized continuity equation. The most common way to do this is the pressure-weighted interpolation method, which was originally developed by Rhie and Chow [90]. However, the appearance of pressure oscillations can also be circumvented without introduction of artificial terms, namely by a suitably chosen staggering of the variables in the grid. In a staggered grid, the pressure and the velocity are stored at different grid nodes. The most commonly used structured staggered grid is the one in which the pressures reside in the cell centers and the normal components of the velocity are positioned at the cell faces. A staggered Cartesian grid approach to compute solutions to the incompressible Navier-Stokes equations was first proposed by Harlow and Welch [44] in 1965, and for a thorough review of possible ways to extend the method to boundary-fitted grids containing quadrilateral cells, we refer to Chapter 13 of [129]. The reason that nowadays colocated schemes are more prominent in use than staggered schemes is generally attributed to difficulties associated with discretization on non-orthogonal curvilinear grids.

2.4.2 Colocated schemes

Before we give a survey of the literature related to incompressible flow solvers using unstructured colocated grids, we demonstrate that also on unstructured grids with triangular cells the problem of odd-even decoupling of the pressure is likely to crop up. Consider a cell-centered colocated grid, see Figure 2.7a. The discretized pressure

gradient in cell 1 follows from, see also equation (2.22):

$$(\nabla p)_1 = \frac{1}{\Omega_1} \sum_{e(1)} p_e l_e \mathbf{n}_e, \quad (2.23)$$

where the summation runs over the three faces of cell 1, l_e is the length of face e , Ω_1 is the cell area and p_e is the pressure at face e , to be approximated yet. In a grid that is equilateral an accurate way of approximating the pressure at a face is by a simple averaging of the two neighboring cells. This leads to:

$$\begin{aligned} (\nabla p)_1 &= \frac{1}{\Omega_1} \sum_{e(1)} p_e l_e \mathbf{n}_e = \frac{1}{\Omega_1} \left[\frac{1}{2}(p_1 + p_2)l_i \mathbf{n}_i + \frac{1}{2}(p_1 + p_3)l_j \mathbf{n}_j + \frac{1}{2}(p_1 + p_4)l_k \mathbf{n}_k \right] = \\ &= \frac{1}{2\Omega_1} [p_1(l_i \mathbf{n}_i + l_j \mathbf{n}_j + l_k \mathbf{n}_k) + p_2 l_i \mathbf{n}_i + p_3 l_j \mathbf{n}_j + p_4 l_k \mathbf{n}_k] = \\ &= \frac{1}{2\Omega_1} [p_2 l_i \mathbf{n}_i + p_3 l_j \mathbf{n}_j + p_4 l_k \mathbf{n}_k], \end{aligned} \quad (2.24)$$

where we made use of the identity

$$l_i \mathbf{n}_i + l_j \mathbf{n}_j + l_k \mathbf{n}_k = \mathbf{0} \quad (2.25)$$

in the last step. We observe that p_1 itself does not contribute to the pressure gradient in cell 1. This leads to the unacceptable situation that an odd-even distribution of the pressure as depicted in Figure 2.7b yields a zero pressure gradient, in the discrete sense, in all cells. A similar reasoning leading to the same conclusion holds when a collocated vertex-centered grid is considered. Now we will discuss several incompressible collocated flow solvers on unstructured grids, including the measures that are taken to get rid of spurious pressure modes.

Chan and Anastasiou (1999)

In the work of Chan and Anastasiou [13] a cell-centered finite volume formulation with Roe's flux function is presented, that uses the concept of pseudocompressibility in order to compute unsteady flows with or without free surfaces. The idea behind pseudocompressibility is the introduction of a time derivative of the pressure in the continuity equation according to:

$$\frac{\partial p}{\partial t} + \beta^2 \nabla \cdot \mathbf{u} = 0, \quad (2.26)$$

where β is the coefficient of pseudocompressibility, representing an artificial speed of sound (density is put to 1). Typical values for β are between 10 and 100. The advantage is that pressure and velocity are coupled such as to produce, in the inviscid case, a hyperbolic system of equations consisting of (2.18) and (2.26), for which flux difference and flux splitting techniques originally developed for compressible flow solvers can be employed. The original continuity equation (2.19) is recovered

when steady state is reached. An implicit pseudotime stepping technique is used to deal with unsteady flows. In the paper by Pan et al. [87], a method similar to the one developed by Chan et al. is described. The main difference between these two papers is that Pan includes, using the Boussinesq approximation, the temperature equation and the source terms representing thermal buoyancy.

Davidson (1996)

Davidson [20] describes a cell-centered finite volume method in which SIMPLEC [115] is used to solve for the pressure and velocity. Central differencing in conjunction with fourth-order numerical dissipation is used for the inertia terms, while the pressure-weighted interpolation method [90] is employed to avoid decoupling between pressure and velocity. Similar to the method developed by Davidson is the one developed by Jiang and Pzrekwas [56].

Foy and Dawes (2000)

A 3D vertex-centered method based on the pressure-correction approach is proposed by Foy and Dawes in [21]. The paper focuses on a consistent discretization of the Laplacian operator for the Poisson equation for the pressure. The forward Euler integration scheme and a finite volume scheme based on central differences are used to do the time and spatial discretization. A fourth-order artificial dissipation term is added both to eliminate velocity-pressure decoupling and to preserve stability.

Williams (1993)

A vertex-centered scheme for the computation of 2D turbulent incompressible steady flows is developed by Williams [134]. However, instead of the usual Poisson pressure equation, a Helmholtz pressure method is used to enforce continuity. A fourth-order pressure dissipation term similar to that of Rhie and Chow [90] is employed to avoid decoupling between velocities and pressures.

Lien (2000)

Lien [69] formulated an algorithm that employs the SIMPLE pressure-correction scheme [88] and which is suitable for both incompressible and compressible flows. The scheme uses a cell-centered storage arrangement and checkerboard oscillations are eliminated by using a pressure-weighted interpolation method similar to that of Rhie and Chow [90].

Watterson (1994)

A pressure-based flow solver for the 3D Navier-Stokes equations on unstructured and adaptive meshes is introduced by Watterson [127]. The method is capable of computing incompressible and compressible flows, and uses a vertex-centered storage. Artificial dissipation is used to get rid of the velocity-pressure decoupling and to improve shock capturing properties.

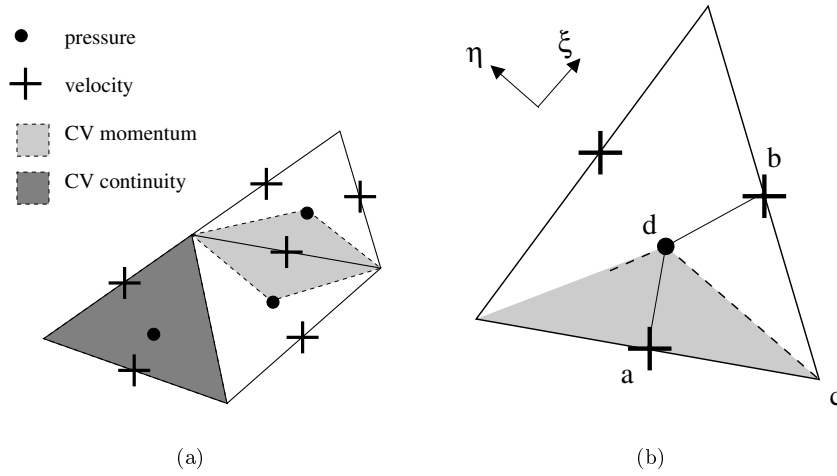


Figure 2.8: Staggered grid and control volumes (a) in the method of Hwang. Part of the CV for the momentum equation (b).

2.4.3 Staggered schemes

Because the scheme forming the main part of this thesis uses a staggered positioning of the variables, a detailed discussion of staggered unstructured schemes that appeared in the literature is useful.

Despotis and Tsangaris (1995, 1997)

In [24, 25], Despotis and Tsangaris introduce an implicit Chorin type fractional step method [14, 15] to compute 2D time-dependent laminar flows. The velocity vector is stored at the vertices of the triangular cells and the pressure in the cell centers. However, they restrict themselves, in the discretization as well in the grids used in their test examples, to grids in which six cells meet at each interior node. Therefore, their method cannot be considered as an unstructured scheme, and we will not discuss it further.

Hwang (1995)

In [52], Hwang proposes a staggered grid in which the pressure is stored at the centroids and the velocity vector at the cell faces, as depicted in Figure 2.8a, together with a scheme devised to find steady solutions. Integration of the continuity equation over a triangular cell leads to (2.21). Because the velocity is located at the triangles' faces, no interpolation is required. For the evaluation of the terms in (2.22), consider the CV-face cd between vertex c and centroid d , see Figure 2.8b.

The convecting velocity at this face is computed from:

$$\mathbf{u}_{cd} = \frac{1}{2}(\mathbf{u}_a + \mathbf{u}_b). \quad (2.27)$$

The convected momentum, with ϕ standing for a Cartesian momentum component, follows from a Peclet number dependent blend between upwind approximation ϕ^u and central approximation ϕ^c :

$$\phi_{cd}^u = \frac{1}{2} \left[(\phi_a + \phi_b) + (\phi_a - \phi_b) \frac{u_{cd}}{|u_{cd}|} \right], \quad \phi_{cd}^c = \frac{1}{2}(\phi_a + \phi_b), \quad (2.28)$$

where $u_{cd} = (\mathbf{u} \cdot \mathbf{n})_{cd}$. The pressure p_e is put equal to the pressure in the cell in which CV-face e is located, hence $p_{cd} = p_d$. To approximate the viscous flux, a local non-orthogonal coordinate system is set up, where the position vector from point a to b is defined to be the ξ -direction and the one from point c to d to be the η -direction. The term $(\mathbf{n} \cdot \nabla \phi)_{cd}$ can then be written, with help of a coordinate transformation, in terms of the transformed (ξ, η) -frame:

$$(\mathbf{n} \cdot \nabla \phi)_{cd} = [(\phi_\xi y_\eta - \phi_\eta y_\xi) n_x + (-\phi_\xi x_\eta + \phi_\eta x_\xi) n_y] / J, \quad (2.29)$$

where $\phi_\xi = \partial \phi / \partial \xi$, $y_\eta = \partial y / \partial \eta$, and so on, and for brevity we wrote $\mathbf{n}_{cd} = (n_x, n_y)$. The Jacobian of the transformation is given by $J = x_\xi y_\eta - x_\eta y_\xi$. The discretized expressions for the geometric quantities are

$$x_\xi = x_b - x_a, \quad y_\xi = y_b - y_a, \quad x_\eta = x_d - x_c, \quad y_\eta = y_d - y_c. \quad (2.30)$$

The momentum derivatives are approximated as follows:

$$\phi_\xi = (\phi_b - \phi_a) / l_{ab}, \quad \phi_\eta = (\phi_d - \phi_c) / l_{cd}, \quad (2.31)$$

where l_{ab} and l_{cd} stand for distance between a and b , and c and d , respectively. Quantities at c and d , which are not velocity nodes, are approximated by interpolation from neighboring nodes. Pressure-velocity coupling is done by means of the standard SIMPLE algorithm [88]. Numerical experiments, including a lid-driven cavity, sudden expansion flow and flow over a backward-facing step, performed with this method are described in a subsequent article [53]. Niceno and Nobile [4] state that in some occasions difficulties crop up with this method when the velocity lies parallel to one of the triangle sides. In this situation, the velocity becomes practically ‘invisible’ to the pressure corrections, and this is assumed to cause erroneous results.

Kobayashi, Nobe and Oka (1999)

A conservative second order finite volume method on hybrid grids consisting of triangles and quadrilaterals to compute steady 2D incompressible viscous flows is presented by Kobayashi et al. in [60]. The momentum vector is stored in the cell

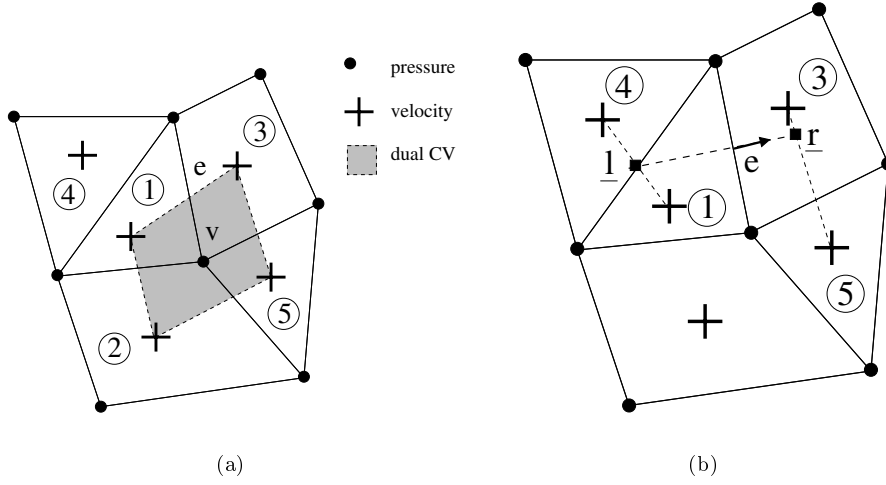


Figure 2.9: Staggered grid (a) in the method of Kobayashi. Nomenclature for viscous discretization (b).

centers and the pressure in the vertices, as depicted in Figure 2.9a. The CV for the discretized momentum equation, see (2.22), is the grid cell. For discretization of the inertia term, the so-called upwind least squares scheme (ULSS) is used. The ULSS is based on a pointwise reconstruction of a Cartesian momentum component ϕ by a piecewise polynomial $R(\mathbf{x})$ which, similarly to the essentially non-oscillatory (ENO) scheme [46], is required to be:

- consistent, in the sense that

$$\frac{1}{\Omega_1} \int_{T_1} R(\mathbf{x}) d\mathbf{x} = \frac{1}{\Omega_1} \int_{T_1} \phi(\mathbf{x}) d\mathbf{x} = \bar{\phi}_1, \quad (2.32)$$

where Ω_1 represents the area of cell 1, and to be of

- high order of accuracy:

$$R(\mathbf{x}) = \phi(\mathbf{x}) + O(h^r), \quad (2.33)$$

where h is a grid parameter and r is the order of the method, which is 2 in the work of Kobayashi.

For triangular cell 1, see Figure 2.9a, we write

$$R_1(\mathbf{x}) = \bar{\phi}_1 + ax + by, \quad (2.34)$$

where x and y are the local Cartesian coordinates with respect to the centroid of cell 1. The coefficients a, b follow from minimization of the functional:

$$\chi(a, b) = \sum_i [R_1(x_i, y_i) - \bar{\phi}_i]^2, \quad (2.35)$$

where the sum runs over the set of adjacent cells, i.e. cells 2, 3 and 4. When the considered CV is a quadrilateral, in (2.34) a term cxy is added. This procedure is done for all cells, and the upwinded momentum at e , the midpoint of the face between cells 1 and 3, follows from:

$$\phi_e = \begin{cases} R_1(\mathbf{x}_e) & \text{if } u_e > 0; \\ R_3(\mathbf{x}_e) & \text{if } u_e < 0, \end{cases} \quad (2.36)$$

where $u_e > 0$ means a flow from cell 1 to 3. The velocity $u_e = (\mathbf{u} \cdot \mathbf{n})_e$ follows from averaging the velocity field at the two centroids that define face e . The normal derivative at face e follows from:

$$(\mathbf{n} \cdot \nabla \phi)_e = (\phi_r - \phi_l) / l_{rl}, \quad (2.37)$$

where ϕ_r and ϕ_l are linearly interpolated values of ϕ at the points \underline{r} and \underline{l} , which are indicated in Figure 2.9b, and l_{rl} represents the distance between these points. The pressure gradient in cell 1 is computed from:

$$(\nabla p)_1 = \frac{1}{\Omega_1} \sum_{e(1)} p_e l_e \mathbf{n}_e, \quad (2.38)$$

with summation over the faces of cell 1, p_e is the mean value of the values of p at the vertices associated with face e , l_e is the length of e and \mathbf{n}_e the outward unit normal. The discrete divergence in vertex v , see Figure 2.9a, is defined as:

$$(\nabla \cdot \mathbf{u})_v = \frac{1}{\Omega_v} \sum_{e(v)} u_e l_e, \quad (2.39)$$

where Ω_v is the area of the dual volume around vertex v and the sum runs over the faces of this volume. The velocity $u_e = (\mathbf{u} \cdot \mathbf{n})_e$ follows from the mean value of the velocity field at the two associated centroids. The pressure-velocity coupling is taken care of by means of the Chorin fractional step method [14, 15].

Perot (2000)

The conservation properties of several unstructured staggered schemes, proposed by Perot, are investigated in [89]. The mesh has to be of Delaunay type, see Section 2.2.2. The normal velocity components and pressures are stored at the face midpoints and cell circumcenters, respectively. The CV for the normal momentum component at face i , indicated by the shaded box in Figure 2.10, has dimensions l_i

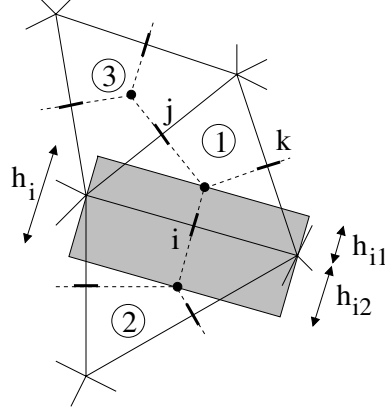


Figure 2.10: Notation for the staggered grid arrangement as used by Perot.

times h_i , where l_i is the length of face i and h_i the distance between the circumcenters of cells 1 and 2. Perot proposes the following discretization for (2.18):

$$l_i h_i \frac{m_i^{n+1} - m_i^n}{\Delta t} + \mathbf{n}_i \cdot (h_{i1} \mathbf{c}_1 + h_{i2} \mathbf{c}_2) l_i = -l_i h_i \frac{p_2 - p_1}{h_i} + \mathbf{n}_i \cdot (h_{i1} \mathbf{d}_1 + h_{i2} \mathbf{d}_2). \quad (2.40)$$

Here $m_i = (\mathbf{m} \cdot \mathbf{n})_i$ represents the normal momentum at face i , Δt stands for the time step, and superscripts n and $n + 1$ refer to the time level. The normal \mathbf{n}_i at face i points from cell 1 to 2, and is part of the line connecting circumcenters 1 and 2 (this is a consequence of the definition of circumcenter). The distance between the midpoint of face i and circumcenters 1 and 2 is indicated by h_{i1} and h_{i2} respectively, hence $h_i = h_{i1} + h_{i2}$. The inertia term in cell 1 is defined as follows, reverting back to Cartesian tensor notation for a moment:

$$c_1^\alpha = \frac{1}{\Omega_1} \int_{T_1} (m^\alpha u^\beta)_{,\beta} d\mathbf{x} = \frac{1}{\Omega_1} \oint_{\partial T_1} m^\alpha u^\beta n^\beta d\Gamma \approx \frac{1}{\Omega_1} \sum_{e(1)} m_e^\alpha u_e l_e, \quad (2.41)$$

where Ω_1 refers to the area of cell 1, the summation runs over the three faces of cell 1, and u_e is the outward normal velocity at face e . With the momentum vector in the two neighboring cells available (how this is achieved will be discussed later), the momentum vector \mathbf{m}_e at face e follows from a simple averaging. For the viscous term a completely similar reasoning as for the inertia term applies, resulting in

$$\mathbf{d}_1 = \frac{1}{\Omega_1} \sum_{e(1)} \nu \left(\frac{\partial \mathbf{u}}{\partial n} \right)_e l_e. \quad (2.42)$$

The normal velocity gradient at face j , see Figure 2.10, is approximated by:

$$\left(\frac{\partial \mathbf{u}}{\partial n} \right)_j = \frac{\mathbf{u}_3 - \mathbf{u}_1}{h_j}, \quad (2.43)$$

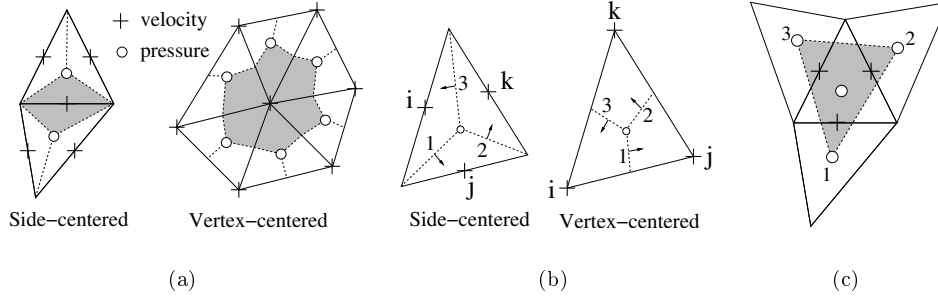


Figure 2.11: CV for the side-centered scheme and the vertex-centered scheme of Rida et al. (a), details of discretization and related nomenclature (b) and macro-element involved in the evaluation of the pressure gradient (c).

with \mathbf{u}_1 and \mathbf{u}_3 the velocity vector in cells 1 and 3. For the computation of the velocity vector in cell 1, the following first-order approximation is used:

$$\mathbf{u}_1 = \frac{1}{\Omega_1} \sum_{e(1)} h_{e1} l_e u_e \mathbf{n}_e, \quad (2.44)$$

with summation over the faces of cell 1 and h_{e1} the distance between the circumcenter of cell 1 and the midpoint of face e . The derivation of this expression will be given in Section 4.6. Perot proves that the discretization as given above conserves kinetic energy (in the inviscid limit of zero viscosity) and momentum locally and globally. In addition, he proposes a discretization of the rotational form (also known as the Lamb-Gromeka form) of the momentum equation for incompressible flow:

$$\frac{\partial \mathbf{u}}{\partial t} + (\boldsymbol{\omega} \times \mathbf{u}) = -\frac{1}{\rho} \nabla p^d - \nabla \times (\nu \boldsymbol{\omega}), \quad (2.45)$$

where $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity and $p^d = p + \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$ the specific dynamic pressure, that conserves circulation and kinetic energy locally and globally. The discretization of (2.45) will not be discussed here, and we refer the interested reader to the original article [89].

Rida, McKenty, Meng and Reggio (1997)

In the paper by Rida et al. [91], two similar staggered schemes are introduced. In both schemes the pressure is stored at the centroids, whereas the momentum vector is stored either at the faces of the elements, resulting in the side-centered scheme, or at the vertices, leading to the vertex-centered scheme. This is shown in Figure 2.11a. Integration of the momentum equation leads to summation over the CV-faces, see

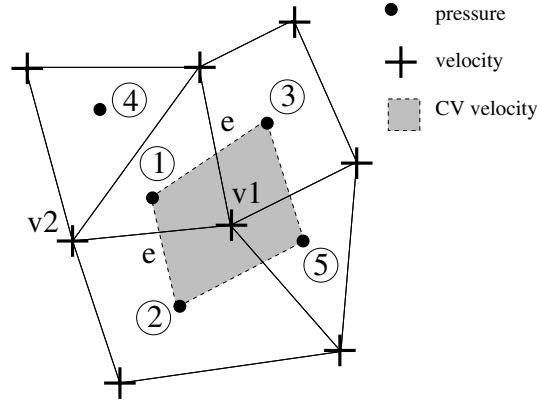


Figure 2.12: Positioning of the variables and CV for the momentum equation in the scheme of Thomadakis and Leschziner.

equation (2.22). The convecting velocity $(\mathbf{u} \cdot \mathbf{n})_e$ and the gradient $(\nabla \phi \cdot \mathbf{n})_e$ at CV-face e , with ϕ a Cartesian momentum component, are computed by assuming linear variation over each triangle. For the evaluation of the convected momentum the skewed mass-weighted upwind function of Schneider and Raw [94] is employed:

$$\phi_1 = \begin{cases} f_1 \phi_3 + (1 - f_1) \phi_i, & f_1 = \min[\max(u_3/u_1, 0), 1], \quad \text{if } u_1 > 0; \\ f_1 \phi_2 + (1 - f_1) \phi_j, & f_1 = \min[\max(u_2/u_1, 0), 1], \quad \text{if } u_1 < 0, \end{cases} \quad (2.46)$$

where Figure 2.11b displays the related nomenclature for the side-centered and vertex-centered scheme. Combination of the expression for ϕ_1 with similar expressions at the remaining two CV-faces inside the considered triangle leads to a 3×3 -system from which (ϕ_1, ϕ_2, ϕ_3) can be solved. Several methods to obtain the pressure gradient have been described in [91], and the best method for the side-centered scheme is the one in which a linear variation over the macro-element 123 is assumed, see Figure 2.11c. The SIMPLER-algorithm is utilized to couple velocity and pressure. Numerical experiments showed that both schemes yield results with comparable accuracy, but it turned out that the side-centered scheme converges more rapidly and that in this scheme Neumann-type boundary conditions on solid walls are easier to implement.

Thomadakis and Leschziner (1996)

Thomadakis and Leschziner [103] come up with a method for the computation of incompressible steady viscous flows. A staggered positioning as indicated in Figure 2.12, with the pressure in the centroids and the velocity vector in the vertices, is employed. The CV for the momentum equation is formed by the centroid dual. The convecting velocity, cf. equation (2.22), at the CV-face e joining centroids 1 and 2

follows from:

$$u_e = \mathbf{u}_e \cdot \mathbf{n}_e = \frac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2) \cdot \mathbf{n}_e, \quad (2.47)$$

where the velocities at the cell centers 1 and 2 are obtained by a distance-weighted interpolation from surrounding cell-vertex values. First order upwind interpolation of the form

$$u_e \phi_e = \max(0, u_e) \phi_{v1} + \max(0, -u_e) \phi_{v2} \quad (2.48)$$

is used to enhance stability, with ϕ a Cartesian component of the momentum in its corresponding vertex. The velocity derivatives on CV-edge e are considered to be constant in the auxiliary volume formed by vertices $v1$ and $v2$ and centroids 1 and 2, and are approximated by

$$\frac{\partial u^i}{\partial x^j} = \frac{(-1)^{j+1}}{2V_e} \sum_{k=1}^4 u_k^i (x_{k+1}^{j+1} - x_{k-1}^{j+1}). \quad (2.49)$$

In this expression, V_e is the area of this auxiliary volume, $i = 1, 2$ (cyclic), $j = 1, 2$ (cyclic), $k = 1, \dots, 4$ (cyclic), $u^1 = u^x$, $u^2 = u^y$, $x^1 = x$ and $x^2 = y$. The pressure at CV-face e in equation (2.22) is approximated by means of the trapezoidal rule

$$p_e = \frac{1}{2}(p_1 + p_2), \quad (2.50)$$

with the pressure located in the associated centroid. The SIMPLE procedure [88] is used to ensure pressure-velocity coupling in the solution procedure. In [103] it is stated that, although the staggered formulation provides adequate damping of pressure-velocity oscillations in most applications, it is occasionally necessary to introduce a small measure of additional dissipation. This is done using the PWI method of Rhie and Chow [90]. The cause of pressure-velocity oscillations was already discussed in Section 2.4.2, and, as the reader may easily verify, the arguments given there hold also for the present staggered formulation. In [135] an extension of the work by Thomadakis and Leschziner is proposed.

Covolume method

The covolume method, developed by Nicolaidis, Hall and others [12, 38, 39, 40, 41, 79, 80, 81], is designed to compute incompressible laminar flows. An essential feature of the covolume method is the use of two sets of control volumes, with the property that the edges of each set are perpendicular to the faces of the other set. This naturally leads to the use of a Delaunay type of grid, which was discussed in Section 2.2.2. The Delaunay triangulation forms the so-called primal grid, whereas the associated Voronoi tessellation is called the dual grid. The velocity components normal to the faces of the primal grid are stored, whereas the pressure and other scalars are located at the grid points of the dual grid, i.e. the circumcenters of the triangles. This is illustrated in Figure 2.13. The continuity equation is integrated over each triangle T , resulting in (2.21), where we note that the required normal

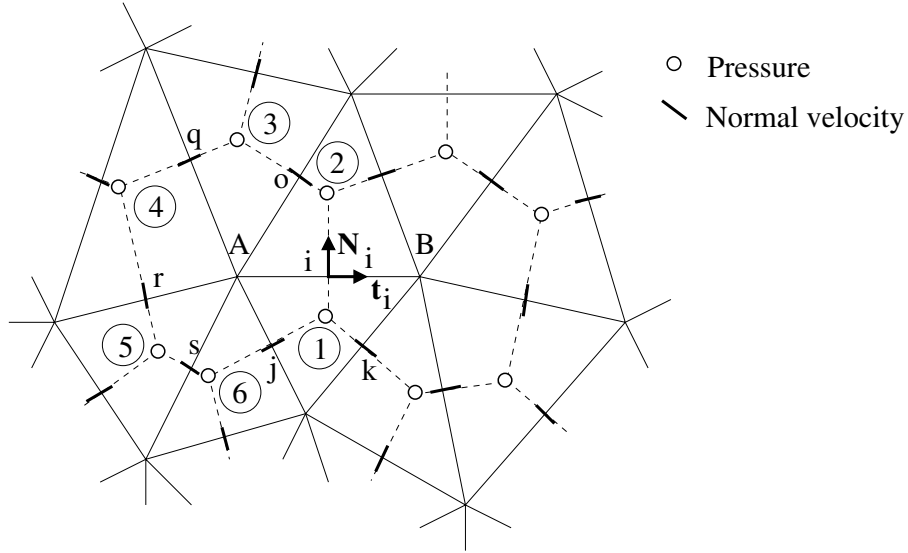


Figure 2.13: The primal and dual grid are indicated by the continuous and dashed line, respectively. Numbers indicate the pressure points, letters refer to the velocity points, and A and B are vertices of the primal grid.

velocity components are located at the triangle's faces. Projection of the momentum equation at the normal vector \mathbf{N}_i at face i yields:

$$\frac{\partial m_i}{\partial t} + \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] = -\frac{\partial p}{\partial N_i} + \mu \frac{\partial \omega}{\partial t_i}, \quad (2.51)$$

where $m_i = \mathbf{m}_i \cdot \mathbf{N}_i$, the scalar vorticity ω is the z -component of $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, and \mathbf{t}_i is the tangential vector at face i , obtained by rotating \mathbf{N}_i over 90° in the clockwise direction. To obtain the expression for the viscous term as given in (2.51), the viscosity μ is assumed to be constant and the identity

$$\nabla^2 \mathbf{u} = \nabla(\nabla \cdot \mathbf{u}) - \nabla \times (\nabla \times \mathbf{u}) \quad (2.52)$$

is used, together with the incompressibility constraint (2.19). Because the nodes of the dual grid are located at the line passing through the midpoint of face i , the pressure gradient can be evaluated consistently by means of

$$\frac{\partial p}{\partial N_i} = \frac{p_2 - p_1}{l_{12}}. \quad (2.53)$$

Here $l_{12} = (\mathbf{x}_2 - \mathbf{x}_1) \cdot \mathbf{N}_i$ stands for the distance between the two pressure points, in the direction of \mathbf{N}_i . Note that the Delaunay triangulation does not prevent the situation in which $l_{12} < 0$ or $l_{12} = 0$, with the former probably resulting in inaccurate

results and the latter in numerical breakdown of the computation. The viscous term at face i follows from

$$\frac{\partial \omega}{\partial t_i} = \frac{\omega_B - \omega_A}{l_i}, \quad (2.54)$$

where l_i refers to the length of face i , which is also the distance between vertices A and B . With help of Stokes's theorem, the vorticity at node A is approximated as follows:

$$\omega_A = \frac{1}{\Omega_A} \oint_{\partial \Omega_A} (\mathbf{u} \cdot \boldsymbol{\tau}) d\Gamma = \frac{1}{\Omega_A} \sum_{e \in (A)} (\mathbf{u} \cdot \boldsymbol{\tau})_e l_e. \quad (2.55)$$

In this expression, Ω_A represents the area of the Voronoi cell surrounding node A , and the summation is over the edges $e \in \{i, o, q, r, s, j\}$, having length l_e , connected to node A . Since the vector $\boldsymbol{\tau}$ is tangent to the boundary of the Voronoi cell and traverses it in counterclockwise direction, we have $\boldsymbol{\tau}_e = \pm \mathbf{N}_e$, with \mathbf{N}_e the normal vector at face e , and consequently $(\mathbf{u} \cdot \boldsymbol{\tau})_e = \pm u_e$.

Apparently it turned out to be a hard task to treat the inertia term accurately, since various formulations have been proposed in the articles devoted to the covolume method. In [38] the following central difference approximation for the inertia term at face i is introduced:

$$\nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)]_i = \frac{\Omega_1}{\Omega_1 + \Omega_2} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)]_{T1} + \frac{\Omega_2}{\Omega_1 + \Omega_2} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)]_{T2}, \quad (2.56)$$

where Ω_1 and Ω_2 refer to the areas of cells $T1$ and $T2$. The inertia term in cell 1 is obtained from

$$\begin{aligned} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)]_{T1} &= \frac{1}{\Omega_1} \int_{T1} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] d\mathbf{x} = \frac{1}{\Omega_1} \oint_{T1} (\mathbf{u} \cdot \mathbf{n})(\mathbf{m} \cdot \mathbf{N}_i) d\Gamma \approx \\ &\approx \frac{1}{\Omega_1} \sum_{e \in (1)} u_e l_e (\mathbf{m}_e \cdot \mathbf{N}_i), \end{aligned} \quad (2.57)$$

where $u_e = (\mathbf{u}\mathbf{n})_e$ and the summation is over the three faces $e \in \{i, j, k\}$ of cell 1. For the evaluation of \mathbf{m}_e , more than five different schemes are proposed in the various references describing the covolume method, and we advise the interested reader to consult these. In [41], a first order scheme and a central difference scheme, with the latter different from the one described in (2.56)–(2.57), are introduced. However, these schemes are fairly complicated, and we will omit a discussion of them here. Application of the chain rule and the incompressible continuity equation result in the expression

$$\nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] = \mathbf{u} \cdot \nabla (\mathbf{m} \cdot \mathbf{N}_i) = |\mathbf{u}| \frac{d(\mathbf{m} \cdot \mathbf{N}_i)}{ds}, \quad (2.58)$$

with $\mathbf{s} = \mathbf{u}/|\mathbf{u}|$ the unit vector in the direction of \mathbf{u} . The expression above, see [40], serves as a starting point for the first order upwind discretization given by

$$|\mathbf{u}_i| \frac{d(\mathbf{m}_i \cdot \mathbf{N}_i)}{ds} \approx |\mathbf{u}_i| \frac{(\mathbf{m}_i \cdot \mathbf{N}_i) - (\mathbf{m}_{i'} \cdot \mathbf{N}_i)}{l_{ii'}}. \quad (2.59)$$

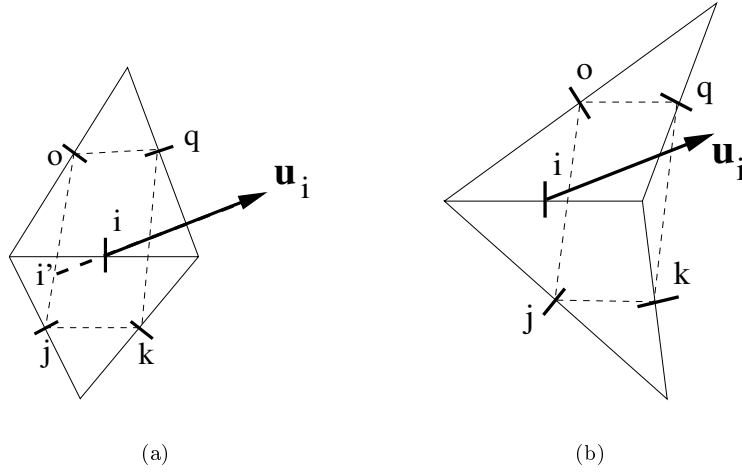


Figure 2.14: Possible first order upwind scheme for the covolume scheme (a). A pair of non-convex triangles (b).

If both triangles adjacent to face i are convex, then $\mathbf{m}_{i'}$, with i' the intersection of the line $\mathbf{x}_i - \lambda \mathbf{u}_i$, $\lambda \in \mathbb{R}^+$ with the parallelogram $jk o q$ as depicted in Figure 2.14a, follows from

$$\mathbf{m}_{i'} = (l_{i'j} \mathbf{m}_o + l_{i'o} \mathbf{m}_j) / l_{jo}. \quad (2.60)$$

In these expressions, $l_{ii'}$ represents the distance between points i and i' , and similarly for $l_{i'j}$, $l_{o'o}$ and l_{jo} . However, the discretization becomes very complicated in the situation in which non-convex triangles, as shown in Figure 2.14b, occur.

The covolume method can be extended to three dimensions [12], but it is recognized that this is complicated [81]. For the coupling of velocity and pressure, the so-called dual variable method is employed. This method originates from network theory, and makes extensive use of the dual grid properties discussed above. This will not be discussed further in this thesis.

2.4.4 Closing remarks

In the previous sections several collocated and staggered schemes on unstructured grids have been described. With respect to the spatial discretization, none of the schemes that have appeared in literature so far use the same staggered positioning as we do. Most of the schemes store the complete velocity vector, either in the cell centers (Kobayashi et al.), in the cell vertices (Rida et al. and Thomadakis et al.), or at the cell faces (Hwang and Rida et al.). The first two choices can give rise to odd-even decoupling of the pressure, while the third choice can lead to

undesired results as well, see [4]. The spatial discretizations as proposed by Perot and in the covolume method use a positioning of the variables in which the normal velocities are stored at the cell faces and the pressures in the cell circumcenters. Such a staggered grid arrangement prevents spurious pressure oscillations, but both approaches require Delaunay grids, and generating them has some disadvantages as pointed out in Section 2.2.2. Also, there is the possibility of numerical breakdown, see the discussion following (2.53). Furthermore, it seems to us that having the pressure in the circumcenter rather than in the centroids might hamper accuracy, especially when some of the circumcenters are located relatively far from the center of gravity, or even located outside the triangle to which they belong. Our scheme stores the normal velocities at the cell faces, but stores the pressure, unlike in the work of Perot or in the covolume method, in the cell centroids. With such a positioning of the variables, odd-even decoupling of the pressure is avoided, while there is no need to restrict ourselves to Delaunay triangulations.

All methods, except for the colocated scheme by Lien, are designed exclusively for either compressible or incompressible flows. Our scheme, to be introduced in the next chapters of this thesis, is cast into a Mach-uniform formulation, which means that both incompressible and compressible flows can be computed accurately and efficiently.

In conclusion, it can be said that our scheme is novel in two respects: (i) the spatial discretization, and (ii) the Mach-uniform solution procedure.

Chapter 3

Solution strategy

Definition of ‘left’ and ‘right’ state vectors at a control volume face forms the starting point for the familiar flux difference and flux splitting schemes for the Euler equations. In this type of solution methods, the elements of the state vector in a grid point are usually updated collectively. However, definition and collective updates of such state vectors are not naturally given on a staggered grid. On the other hand, discretization by a simple finite difference or finite volume scheme for each primary variable separately is natural on a staggered grid. It is also natural to update the primary variables sequentially in a time-stepping or iterative procedure. For the primary variables we take the momentum vector and two thermodynamic variables. All other variables then follow from algebraic relations, e.g. the equation of state. Upwind or central interpolation for the convection term in each of the governing equations results in a very simple scheme. Such a staggered and segregated approach is common in incompressible flow CFD and the field of shallow-water equations, see for example [11, 102], but is not often encountered in compressible flow CFD. The spatial discretization is the subject of the next chapter; in this chapter we focus on the various sequential update procedures that we have investigated. The time-integration method that we have adopted is discussed in Section 3.1. The well-known pressure-correction approach for incompressible flows is repeated in Section 3.2. The sequential update procedure introduced in Section 3.3 is designed for fully compressible flows. This procedure is kept as simple as possible, since its aim is to test our novel spatial discretization scheme in the realm of compressible flows. Some remarks on the difficulties associated with the computation of low subsonic flows using standard compressible flow solvers are given in Section 3.4. The topic of Section 3.5 is the introduction of a Mach-uniform sequential update procedure, which is a way to compute flows that is uniformly accurate and efficient in the Mach number. How the linear systems that result from the spatial discretization are solved is discussed in Section 3.6.

3.1 Time integration

For computation of both stationary flows and time-accurate solutions of nonstationary flows, time stepping is used. For steady flows, time integration can be viewed as an iterative procedure to arrive at the steady state solution. With $w^n = (w_1^n, \dots, w_N^n)$ the solution vector of primary variable w at time-level n , where N equals the number of unknowns, a stationary problem is said to have converged sufficiently to steady state if the termination criterion

$$\|w^{n+1} - w^n\|_2 \leq \epsilon \frac{1 - \lambda}{\lambda} \|w^{n+1}\|_2, \quad (3.1)$$

with a user-specified relative accuracy ϵ , is satisfied for all primary variables. The rate of convergence λ is defined by:

$$\lambda = \frac{\|w^{n+1} - w^n\|_2}{\|w^n - w^{n-1}\|_2}, \quad (3.2)$$

and $\|\cdot\|_2$ is the standard L_2 -norm. This stopping criterion is based on the assumption of linear convergence behavior. Replacing the L_2 -norm by the L_∞ -norm (max-norm) makes little difference, since numerical experiments show that the behavior of both $\|w^n\|$ and $\|w^{n+1} - w^n\|$ as function of n is similar in both norms. Time integration is performed by means of the θ -method, discussed below.

The θ -method

Assume we have the following differential equation for the solution vector w :

$$\frac{dw}{dt} = L_h w + f, \quad (3.3)$$

where f represents a given source term and L_h a linear operator, for example stemming from spatial discretization. Application of the θ -method, a one-step time-integration method, to this differential equation yields:

$$\frac{w^{n+1} - w^n}{\Delta t} = (1 - \theta)(L_h w^n + f^n) + \theta(L_h w^{n+1} + f^{n+1}), \quad 0 \leq \theta \leq 1, \quad (3.4)$$

with superscripts n and $n + 1$ indicating the time level, and the time step is given by $\Delta t = t^{n+1} - t^n$. Parameter θ should satisfy: $0 \leq \theta \leq 1$. Note that $\theta = 0, 1, 1/2$ correspond to the Euler explicit, Euler implicit and Crank-Nicolson method, respectively. For $1/2 \leq \theta \leq 1$, the θ -method is stable for all Δt , and for $\theta = 1/2$ the method is second-order accurate in time while being first-order accurate for all other θ -values. An efficient way to implement (3.4), for $0 < \theta \leq 1$, is the following:

1. Solve equation (3.3) by means of the implicit Euler method for a time step equal to $\theta \Delta t$:

$$\frac{w^{n+\theta} - w^n}{\theta \Delta t} = (L_h w^{n+\theta} + f^{n+\theta}), \quad f^{n+\theta} = f^n + \theta(f^{n+1} - f^n). \quad (3.5)$$

2. Quantity w at time-level $n + 1$ follows from extrapolation:

$$w^{n+1} = \frac{1}{\theta}w^{n+\theta} + \left(1 - \frac{1}{\theta}\right)w^n. \quad (3.6)$$

The advantage of this approach with respect to implementation of (3.4) is that the term $L_h w^n$ needs not be computed. We put θ equal to 1 (implicit Euler) in this thesis. Since our aim is the testing of a novel spatial discretization scheme, we did not strive for efficiency in the solver and we kept the time step constant throughout the time-marching procedure.

3.2 Pressure-correction approach for incompressible flows

The Navier-Stokes equations for incompressible flow (2.18)–(2.19) do not contain the time derivative of the pressure, or, in other words, there seems to be no equation governing the evolution of the pressure field. Furthermore, the velocity field should satisfy the kinematic constraint (2.19). These two difficulties can be resolved at once by considering the pressure as a Lagrangian parameter that allows the velocity to be divergence free. The pressure-correction approach is based on this concept. We apply the so-called discrete pressure-correction method, in which first the temporal and spatial discretization are formulated and afterwards the pressure-correction equation is derived. By doing this one avoids the necessity to define boundary conditions for the pressure equation, which would be required if the pressure-correction equation were derived from the continuous equations. Note that the pressure-correction approach is time accurate; it should therefore not be confused with for example the SIMPLE approach.

Discretization in space and time of (2.18)–(2.19), assuming for simplicity that $\rho = 1$, is written symbolically as:

$$Du^{n+1} = 0, \quad (3.7)$$

$$R_m \frac{u^{n+1} - u^n}{\Delta t} + C(u^{n+1})u^{n+1} = -R_m Gp^{n+1} + Vu^{n+1}, \quad (3.8)$$

where u and p stand for the velocity and pressure solution vectors at discrete time levels indicated by n and $n+1$. The implicit Euler scheme is used for time integration. The matrix D corresponds to the discrete divergence operator, R_m is a diagonal matrix containing the area of the control volumes of the discretized momentum equations, and C , G and V refer to the discrete convection, gradient and viscous operator, respectively. Derivation of expressions for these operators is postponed to the next chapter.

The first step in the pressure-correction algorithm is to compute u^* , the predictor

of the new velocity, from:

$$R_m \frac{u^* - u^n}{\Delta t} + C(u^n)u^* = -R_m G p^n + V u^*. \quad (3.9)$$

Note that the pressure is taken at the previous time level and that the nonlinear inertia term is Picard linearized:

$$C(u^{n+1})u^{n+1} \approx C(u^n)u^*. \quad (3.10)$$

In general, the velocity prediction is not divergence free: $Du^* \neq 0$. Subtracting (3.9) from (3.8) yields:

$$R_m \frac{u^{n+1} - u^*}{\Delta t} + C(u^{n+1})u^{n+1} - C(u^n)u^* = -R_m G(p^{n+1} - p^n) + V(u^{n+1} - u^*). \quad (3.11)$$

According to Van Kan [116], neglecting the term

$$C(u^{n+1})u^{n+1} - C(u^n)u^* - V(u^{n+1} - u^*) \quad (3.12)$$

does not affect the temporal accuracy for first and second order accurate time integration methods. Defining the pressure correction as $\delta p = p^{n+1} - p^n$, we arrive at the following relation between u^{n+1} , u^* and δp :

$$u^{n+1} = u^* - \Delta t G \delta p. \quad (3.13)$$

Application of (3.7) to (3.13) yields the pressure-correction equation:

$$\Delta t D G \delta p = D u^*. \quad (3.14)$$

After having solved this equation for δp , the new velocity u^{n+1} follows easily from (3.13). The use of the implicit Euler scheme in (3.9) strongly weakens the stability requirements, since stability of (3.9) is considered sufficient for stability of the pressure-correction approach.

Summarizing, the pressure-correction algorithm for incompressible flows consists of three steps:

1. Compute the velocity prediction u^* from (3.9).
2. Compute the pressure-correction δp from (3.14).
3. The new velocity u^{n+1} follows from (3.13).

Solution of the linear systems in steps (1) and (2) is done by means of Krylov subspace methods, see Section 3.6. Discretization of the momentum equation yielding the velocity prediction is the topic of Section 4.2, and discretization of the pressure-correction equation and the correction to the velocity is discussed in Section 4.5.

3.3 Sequential update procedure for fully compressible flows

For the novel spatial discretization, to be introduced in the next chapter, a sequential update procedure has to be designed. Because, to the best of our knowledge, nobody else has yet entered the field of compressible CFD using a staggered scheme on unstructured grids, the proposed sequential update procedure is made on purpose very straightforward in order to avoid complications as much as possible. In other words, since our aim is testing of a novel spatial discretization scheme, we did (initially) not strive for efficiency in the solver. The primary variables are the momentum vector, the density and an energy variable, which we will indicate by Ψ . Selection of a suitable energy variable Ψ from the set $\{\rho H, \rho E, H, h\}$ will be based on numerical experiments, see Section 6.2.1. Using notation defined in the previous section, the following sequential update procedure for the Euler equations (2.7)–(2.9) is proposed:

1. Compute the new momentum m^{n+1} from:

$$R_m \frac{m^{n+1} - m^n}{\Delta t} + C(u^n)m^{n+1} = -R_m G p^n, \quad (3.15)$$

where we use Picard linearization (3.10) for the inertia term.

2. Compute the new density ρ^{n+1} from:

$$R_\rho \frac{\rho^{n+1} - \rho^n}{\Delta t} + D(\rho^{n+1}u) = 0 \quad (3.16)$$

where R_ρ is a diagonal matrix containing the areas of the control volumes, and the velocity u follows from m^{n+1} and ρ^n .

3. Compute the new energy variable Ψ^{n+1} from:

$$R_\Psi \frac{(\rho E)^{n+1} - (\rho E)^n}{\Delta t} + D(u\rho H)^{n+1} = 0, \quad (3.17)$$

where R_Ψ is a diagonal matrix containing the areas of the control volumes, and Ψ is related to ρE and ρH by means of standard thermodynamic relations.

4. The new pressure p^{n+1} follows from the equation of state (2.15).

This procedure performs, as we will see, satisfactorily for fully compressible flows, but turns out to suffer from loss of accuracy and efficiency when the Mach number becomes small. This explains why we call this the ‘fully compressible flow approach’ or, more briefly, the FC approach. The linear systems, appearing in the first three stages of the update procedure, are solved by means of Krylov subspace methods, see Section 3.6. Discretization of the momentum, continuity, energy equation and the equation of state is discussed in Sections 4.2, 4.3, 4.4 and 4.6.2, respectively.

3.4 The need for unified methods

The equations of motion for compressible flows as discussed in Section 2.1.1 are uniformly valid for Mach numbers M ranging from zero to supersonic (until real gas effects or other deviations from the given assumptions set in). However, the standard methods that have been developed for the computation of compressible flows suffer from efficiency and accuracy problems when $M \lesssim 0.2$. If the Mach number in the flow remains bounded below 0.2 and heat addition is zero or negligibly small, then the incompressible flow equations given in Section 2.1.3 accurately model the flow, and incompressible flow solvers do a good job. This leaves us with the question of how to deal with flows in which both compressible and incompressible flow regions are present. Typical examples for which a Mach-uniform method seems indispensable are the flow around an aircraft in landing or take-off configuration ($M \approx 0.2$) or the flow in turbomachinery nozzles, in which both compressible and virtually incompressible flow zones are present. Also for the computation of flows in which the overall Mach number is very small but in which substantial density variations are present, for example in flows with cavitation or combustion, a unified method seems handy. Even in hypersonic flight situations, thick boundary layers and separation zones occur in which the Mach number can be smaller than 0.2. From this we conclude that it would be ideal if a method could be found that is efficient and accurate uniformly in the whole Mach number range. A method meeting these requirements will be called a Mach-uniform or a unified method. In Section 3.4.1, the difficulties encountered by standard flow solvers in the zero Mach number limit are addressed, and in Section 3.4.2 some of the proposed remedies are discussed. In Section 3.5 a Mach-uniform method will be introduced.

3.4.1 Difficulties with the zero Mach number limit

Efficiency problem

When using an explicit time-integration scheme, as is often done in standard compressible flow solvers, one needs to satisfy a stability restriction of the following form:

$$\Delta t_c \leq C_c \frac{\Delta x}{u+a}, \quad (3.18)$$

where C_c is a constant of the order unity, Δx the meshwidth and $(u+a)$ is the speed of the acoustic modes, with u the velocity of the fluid particles and a the speed of sound. For incompressible flows, the stability restriction is less severe:

$$\Delta t_i \leq C_i \frac{\Delta x}{u}, \quad (3.19)$$

with C_i a constant of the order unity. Note that Δt_i , if acoustic effects are absent, is in balance with the physical time scale. With Mach number $M = u/a$ we find

$$\frac{\Delta t_c}{\Delta t_i} = C \frac{u}{u+a} = C \frac{M}{1+M}, \quad C = C_c/C_i, \quad (3.20)$$

so that the numerical time step for compressible flow methods needs to be much smaller than the physical time step if $M \ll 1$. If no special measures are taken, this results in numerical inefficiency caused by the need to resolve acoustic modes, because (3.18) remains valid even if there are no acoustic modes. For low Mach number flow we have $u \ll (u + a)$, so that a fluid particle travels only over a small distance in the grid cell during the time step dictated by (3.18). Hence, a huge number of time steps is required before phenomena, the evolution of which is characterized by the physical time scale, can be resolved. If, instead of time marching to steady state, an iterative steady state flow solver is used, a similar convergence problem appears. For such methods the convergence rate is proportional to the condition number of the Jacobian. The eigenvalues of the Jacobian $A = \partial F / \partial U$, see equation (2.10), are given by u , $u \pm a$. The condition number is the ratio of the largest and smallest eigenvalue, and is for $u \ll (u + a)$ given by:

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = \frac{u + a}{u} = 1 + \frac{1}{M}. \quad (3.21)$$

Hence, the condition number tends to infinity as M approaches zero, resulting in slow convergence.

Accuracy problem associated with proper choice of units

Another indication of numerical trouble related to the low Mach number limit reveals itself when the momentum equation is made dimensionless. Nondimensionalization of the flow equations is done by choosing appropriate reference values for four independent quantities. By means of identities and thermodynamic relations, all other reference values are readily obtained. The following four reference values, to be indicated by subscript r , are usually chosen in compressible fluid dynamics: density ρ_r , velocity u_r , temperature T_r and length L_r . Dimensionless quantities, labeled with tildes, are defined by $\tilde{\rho} = \rho / \rho_r$, $\tilde{p} = p / p_r$, $\tilde{t} = t / t_r$, with $t_r = L_r / u_r$, and so on. The dimensionless form of the inviscid momentum equation reads, upon deleting tildes for brevity:

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = -\frac{p_r}{\rho_r u_r^2} p_{,\alpha}. \quad (3.22)$$

The reference pressure follows from the equation of state (2.15):

$$p_r = (\gamma - 1)\rho_r e_r, \quad e_r = c_v T_r, \quad (3.23)$$

with c_v the specific heat at constant volume of the fluid. An estimate for the magnitude of the speed of sound is

$$a_r = \sqrt{\gamma(\gamma - 1)e_r}, \quad (3.24)$$

which follows from (2.13) and (2.17). Hence, with $M_r = u_r / a_r$ being representative for the Mach number in the flow, the dimensionless momentum equation becomes:

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = -\frac{1}{\gamma M_r^2} p_{,\alpha}. \quad (3.25)$$

We see that this equation becomes singular as $M_r \downarrow 0$, which spells numerical troubles, for instance round-off error difficulties, for low subsonic flows with methods developed for compressible flows only.

Accuracy problem related to asymptotic expansion of the continuous and discrete equations

As we saw above, the flow equations become singular as the Mach number tends to zero if these equations are made dimensionless in the way that is customary for compressible flows. It is therefore interesting to study the way in which solutions of the compressible flow equations converge to solutions satisfying the equations for incompressible flows. To this aim, we postulate an asymptotic expansion of the following form

$$p(\mathbf{x}; t, \tau) = p_0(\mathbf{x}; t, \tau) + \epsilon p_1(\mathbf{x}; t, \tau) + \epsilon^2 p_2(\mathbf{x}; t, \tau) + \mathcal{O}(\epsilon^3), \quad \epsilon = \sqrt{\gamma} M_r, \quad (3.26)$$

where t and τ represent the flow and acoustic time scale, respectively, and $\tau = t/\epsilon$. After introduction of similar expansions for the other dependent variables, the low Mach number limit can be studied, as described for example in [58, 59] or in Section 2.4 of [8], by inserting these expansions in the Euler equations and equating terms with equal powers of ϵ . We get the following results for the continuous flow equations:

- If there is no global expansion or compression, $p_0(\mathbf{x}; t, \tau) = p_0$ is constant in space and time. Here p_0 can be considered as the global thermodynamic pressure part, which is basically the constant background pressure level.
- Since $p_1(\mathbf{x}; t, \tau)$ is governed by a wave equation with waves moving at speed a , the term $p_1(\mathbf{x}; t, \tau)$ is identified as the acoustic part of the pressure. If acoustic effects are omitted, then the term p_1 as well as the acoustic time scale τ disappear from (3.26).
- Under the assumption of absent acoustic modes, the expansion leads for terms of the order ϵ^2 to:

$$\frac{\partial m_0^\alpha}{\partial t} + (u_0^\beta m_0^\alpha)_{,\beta} = -(p_2)_{,\alpha}, \quad m_0^\alpha = \rho_0 u_0^\alpha. \quad (3.27)$$

Since $p_2(\mathbf{x}, t)$ ensures compliance with the divergence constraint on the velocity, the term $\epsilon^2 p_2(\mathbf{x}, t)$ in (3.26) is called the incompressible flow part of the pressure.

- In contrast to what is sometimes believed, it is the energy equation (and not the continuity equation) from which the kinematic constraint

$$\nabla \cdot \mathbf{u}_0 = 0 \quad (3.28)$$

in the limit of $M_r \downarrow 0$ is derived.

These results will be used when designing a Mach-uniform formulation, see Section 3.5. In [36] it is shown that solutions obtained by means of flux difference schemes contain pressure fluctuations of the order M_r , while, as we saw above, the continuous pressure scales with M_r^2 (no acoustics involved). This loss of accuracy explains why flux difference methods actually fail to compute weakly compressible flow. As the Mach number decreases, the results obtained by means of these methods become worse and the solutions do not converge to a reasonable approximation of the incompressible flow solution. As pointed out in [36, 124], the computed $M_r = 10^{-1}$ solution of the flow around a NACA 0012 airfoil is closer to the incompressible flow one than is the computed $M_r = 10^{-3}$ solution.

Weak pressure-density coupling

When computing low subsonic flow, the weak pressure-density coupling has consequences for the choice of primary variables. In incompressible flows, the density is constant along particle lines, but the pressure is not. Therefore, for low Mach number flow it is not a good idea to use density as a primary variable and compute the pressure from the equation of state. In that case we would, in the limit $M_r \downarrow 0$, compute the zeroth order pressure in (3.26) which is not the pressure component we need. The other way around, i.e. computation of the pressure and afterwards evaluation of the density through the equation of state or the continuity equation, does not degrade performance for decreasing Mach numbers, and can without problem even be applied for incompressible flows. But standard compressible flow methods are density-based, and consequently suffer from weak pressure-density coupling when the Mach number is small. The fully compressible flow approach introduced in Section 3.3 is an example of a density-based method.

3.4.2 Proposed remedies

Various strategies have been proposed to relieve or get rid of the difficulties touched upon in the previous section, and we will briefly describe the two most prevalent approaches.

Preconditioning

The stiffness problem that occurs when $M_r \downarrow 0$ can be alleviated by modifying the equations (2.10) artificially by multiplication of the time derivative by a matrix P^{-1} :

$$P^{-1} \frac{\partial U}{\partial t} + \nabla \cdot F = 0. \quad (3.29)$$

This procedure is called preconditioning, and the preconditioning matrix $P = P(U)$ should be chosen such that the modified system is less stiff than the original system. In other words, the eigenvalues of PA should lie closer together than the eigenvalues of the Jacobian $A = \partial F / \partial U$. The design of P is difficult and remains subject of much research, see for example [19, 61, 63, 105, 106, 107, 128]. Since multiplication

by P artificially diminishes acoustic speeds, time accuracy is lost. This does not affect stationary solutions. However, restoring time accuracy is awkward. This can be done by employing a dual time-stepping procedure, which involves an inner iteration loop in pseudotime that is wrapped in an outer loop stepping through physical time. Thus, the flow field at each physical time level is treated as a steady state problem in pseudo time. In this way, the physical time-step size is not affected by the stiffness of the system. Nevertheless, because of the presence of an inner iteration loop, dual time stepping is computationally expensive [128]. A major practical advantage of the preconditioning approach is that existing codes for computation of compressible flows can easily be extended to improve their performance in the low subsonic flow region. Another advantage is, as shown in [36], that correct preconditioning of the numerical dissipation tensor recovers a correct scaling of the pressure, i.e. the pressure scales with M_r^2 , for M_r small.

Pressure-based methods

Instead of improving accuracy and efficiency of compressible flow codes by means of preconditioning, one can take the other way around and incorporate compressibility in incompressible flow methods. Because of the lack of pressure-density coupling in incompressible flows, in these methods pressure serves as a primary variable. Consequently, Mach-uniform formulations based on incompressible flow methods are pressure-based. Some examples of unified methods using colocated grids are described in [22, 49, 69, 77], and some unified methods on staggered grids are introduced in [9, 42, 43, 99, 110, 111]. Essential features in all pressure-based methods are the presence of (i) the pressure or pressure-correction equation, which is a Poisson-like equation yielding the new pressure, and (ii) corrections to the velocity (or momentum) in order to ensure compliance with the continuity equation.

3.5 Mach-uniform solution algorithm

We discuss in this section the pressure-based conservative Mach-uniform solution algorithm that recently is introduced in [111]. The advantage of this algorithm over its precursor introduced in [8, 9] is that the latter uses a nonconservative discretization of the energy equation, which leads to discrepancies in satisfaction of the Rankine-Hugoniot conditions, as shown in [111]. With the Mach-uniform scheme one can compute flows with a Mach number ranging from the incompressible flow limit $M \downarrow 0$ up to supersonic flow $M > 1$, with nearly uniform efficiency and accuracy. This uniformity in the Mach number will be demonstrated in Chapter 6. In the incompressible flow limit, the Mach-uniform scheme reduces to the standard pressure-correction scheme discussed in Section 3.2. The dimensionless formulation of the Navier-Stokes equations is derived in Section 3.5.1, and the solution procedure is given in Section 3.5.2.

3.5.1 Dimensionless formulation

Definition of dimensionless pressure

With four independent reference values (for example, temperature T_r , length L_r , velocity u_r and density ρ_r) given, the Navier-Stokes equations (2.1)–(2.3) and the equation of state (2.15) can be made dimensionless. With ϕ_r a reference value, quantity ϕ and its dimensionless equivalent $\tilde{\phi}$ are related through $\tilde{\phi} = \phi/\phi_r$. This holds for all quantities except the pressure, which is nondimensionalized in our Mach-uniform formulation as follows:

$$\tilde{p} = \frac{p - p_r}{\rho_r u_r^2}, \quad (3.30)$$

with p_r the background thermodynamic pressure given in (3.23). This particular definition of the dimensionless pressure is an essential feature of the Mach-uniform formulation. The same pressure scaling is also used in [9, 110, 111]. The physical significance of (3.30) can be found when we return to (3.26), which reduces to

$$p(\mathbf{x}, t) = p_0 + \gamma M_r^2 p_2(\mathbf{x}, t) + \mathcal{O}(M_r^4) \quad (3.31)$$

if acoustic effects are ignored. This shows that with the obvious alternative $\tilde{p}_s = p/p_0$ the variation of \tilde{p}_s is $\mathcal{O}(M_r^2)$, so there is a risk of loss of significant digits for M_r small. However, substitution of (3.31) in (3.30) gives (choosing $p_r = p_0$):

$$\tilde{p} = \frac{p_2}{p_r} + \mathcal{O}(M_r^2), \quad (3.32)$$

where we have used (3.23) and (3.24). This shows that \tilde{p} has the nice property of being $\mathcal{O}(1)$ as $M_r \downarrow 0$. Of course, this is already obvious from (3.30), if one thinks of Bernoulli's theorem. Furthermore, with (3.31) the singular factor $1/\gamma M_r^2$ present in the dimensionless momentum equation (3.25) disappears. The relation between the dimensionless pressure \tilde{p} , equation (3.30), and the obvious alternative $\tilde{p}_s = p/p_0$ can easily be derived. It is trivial that:

$$\tilde{p} = \frac{p - p_r}{\rho_r u_r^2} = \frac{p_r}{\rho_r u_r^2} \frac{p - p_r}{p_r} = \frac{p_r}{\rho_r u_r^2} (\tilde{p}_s - 1), \quad (3.33)$$

and we get:

$$\tilde{p}_s = 1 + \gamma M_r^2 \tilde{p} \quad (3.34)$$

with $p_r/\rho_r u_r^2 = 1/\gamma M_r^2$.

Dimensionless equations

Nondimensionalization as described above leaves the continuity equation invariant (deleting tildes for brevity):

$$\frac{\partial \rho}{\partial t} + (\rho u^\alpha)_{,\alpha} = 0. \quad (3.35)$$

The dimensionless momentum equation is given by, omitting body forces f^α :

$$\frac{\partial m^\alpha}{\partial t} + (u^\beta m^\alpha)_{,\beta} = -p_{,\alpha} + \frac{1}{Re} \left[\mu (u_{,\beta}^\alpha + u_{,\alpha}^\beta) - \frac{2}{3} u_{,\gamma}^\gamma \delta^{\alpha\beta} \right]_{,\beta} \quad (3.36)$$

where the Reynolds number is defined in the usual manner:

$$Re = \frac{\rho_r u_r L_r}{\mu_r}. \quad (3.37)$$

The thermodynamic relations gathered in Section 2.1.2 are used in order to arrive at the dimensionless equation of state:

$$h = \frac{1}{\rho}(1 + \gamma M_r^2 p). \quad (3.38)$$

When written in the form

$$\rho = \frac{1}{h}(1 + \gamma M_r^2 p), \quad (3.39)$$

it is immediately clear that, as required, the density becomes independent of the pressure in the limit $M_r \downarrow 0$. This is another desirable consequence of (3.30)

The dimensionless energy equation (2.3) can be written, omitting the source terms f^α and q , in the following form:

$$\begin{aligned} M_r^2 \left\{ \frac{\partial}{\partial t} \left[p + \frac{1}{2}(\gamma - 1)\rho u^\gamma u^\gamma \right] + \left[u^\alpha \left(\gamma p + \frac{1}{2}(\gamma - 1)\rho u^\gamma u^\gamma \right) \right]_{,\alpha} \right\} + u_{,\beta}^\beta = \\ = \frac{1}{Re} \left[2u^\alpha \mu \left(e^{\alpha\beta} - \frac{1}{3}\Delta\delta^{\alpha\beta} \right) \right]_{,\beta} + \frac{1}{Pr Re} (kT_{,\alpha})_{,\alpha}, \end{aligned} \quad (3.40)$$

where the Prandtl number is defined as

$$Pr = \frac{c_p \mu_r}{k_r}, \quad (3.41)$$

and c_p stands for the specific heat at constant pressure. Apart from rather unlikely circumstances, see Section 3.6 of [7], the right-hand side of (3.40) can be neglected for $M_r \ll 1$. A closer look at (3.40) reveals, as was already stated in Section 3.4.1, that it is the energy equation (and not the continuity equation) from which the kinematic constraint $(\nabla \cdot \mathbf{u}) = 0$ is derived in the limit $M_r \downarrow 0$. Equation (3.40) forms the basis for the Mach-uniform pressure-correction equation, to be discussed in the next section.

3.5.2 Mach-uniform sequential update procedure

With respect to the Mach-uniform approach we restrict ourselves to inviscid flows, although this is not a prerequisite. The following Mach-uniform (MU) solution algorithm is proposed, where we use notation as in Sections 3.2 and 3.3:

1. Compute the new density ρ^{n+1} from:

$$R_\rho \frac{\rho^{n+1} - \rho^n}{\Delta t} + D(u^n \rho^{n+1}) = 0. \quad (3.42)$$

2. Compute the momentum prediction m^* from:

$$R_m \frac{m^* - m^n}{\Delta t} + C(u^n)m^* = -R_m G p^n. \quad (3.43)$$

3. The new pressure follows from the pressure-correction equation.
4. A correction is added to the predictor of the momentum to obtain the new momentum.
5. The new enthalpy follows from (3.38).

The second step is identical to the first step in the update procedure for fully compressible flows, but in the MU approach the computed quantity is interpreted as the predictor of the momentum rather than the momentum at the new time level. Note that steps 1 and 2 can be interchanged without affecting the algorithm. For (nearly) incompressible flows, the density variations should remain (nearly) zero, which is ensured by inserting the (nearly) divergence-free u^n in the continuity equation. We found out that, especially for low subsonic flow, computing times for steady flows can be reduced drastically by starting with divergence-free initial conditions (usually free flow) and to use, in step 1 in the first time step, these initial conditions as boundary conditions at the surface of the obstacles in the flow rather than the given boundary conditions there. The reason is that, if the given boundary conditions are used, the density is changed dramatically after the first time step (because of $Du^0 \neq 0$ caused by impermeable boundaries), and it takes long before this transient behavior is convected out of the domain. Starting with zero velocity is not a good option, since this requires time-dependent inflow conditions, and this again will be felt in the density for a long time. When applied to incompressible flows with constant density, steps 1 and 5 in the Mach-uniform solution algorithm are not of interest, and the pressure-correction approach for incompressible flows, see Section 3.2, is recovered. Derivation of the Mach-uniform pressure-correction equation forms the remainder of this section, and discretization of the governing equations is discussed in Sections 4.3 (step 1 of the algorithm), 4.2 (step 2), 4.5 (steps 3 and 4) and 4.6.1 (step 5). Krylov subspace methods, see Section 3.6, are used to solve the linear systems appearing in steps 1 to 3.

Mach-uniform pressure-correction equation

The following relation between the momentum at time-level $n+1$ and the predictor of the momentum is postulated, cf. (3.13):

$$\mathbf{m}^{n+1} = \mathbf{m}^* - \Delta t \nabla \delta p, \quad (3.44)$$

where the pressure correction is defined as

$$\delta p = p^{n+1} - p^n. \quad (3.45)$$

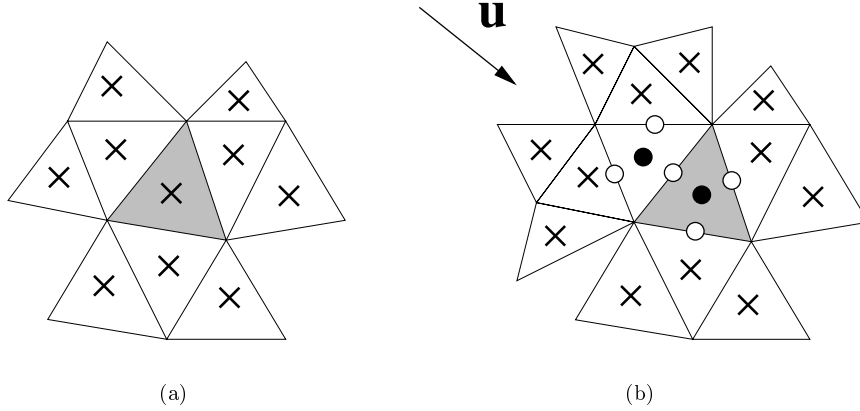


Figure 3.1: Pressure-correction stencil in incompressible flow computations (a). Enlarged pressure-correction stencil (b).

Inserting

$$\mathbf{m}^* = \rho^{n+1} \mathbf{u}^*, \quad (3.46)$$

where \mathbf{u}^* is the predictor of the velocity, and

$$(\rho \mathbf{u}^2)^{n+1} = (\mathbf{m}^2 / \rho)^{n+1} = (\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} \quad (3.47)$$

into the energy equation (3.40) and discretizing in time with Euler implicit yields a nonlinear equation for δp :

$$\begin{aligned} M_r^2 \left\{ \frac{\delta p}{\Delta t} + \frac{1}{2}(\gamma - 1) \frac{(\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} - (\mathbf{m}^n)^2 / \rho^n}{\Delta t} + \right. \\ \left. + \nabla \cdot \left[\left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) \left(\gamma(p^n + \delta p) + \frac{1}{2}(\gamma - 1)(\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} \right) \right] \right\} + \\ + \nabla \cdot \left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) = 0. \end{aligned} \quad (3.48)$$

The right-hand side is put equal to zero since we restrict ourselves, although this is not a prerequisite, to the Euler equations. For brevity, we have used the notation $\mathbf{m}^2 = \mathbf{m} \cdot \mathbf{m}$. The stencil that results from discretization of the Laplacian term $\nabla \cdot \nabla \delta p$ is depicted in Figure 3.1a, where the shaded cell is the cell under consideration. We would like to stick to this stencil in the spatial discretization of (3.48). In the second line of (3.48), a term representing the convected kinetic energy at time-level $n + 1$ is present. The convected kinetic energy has, as we will see, to be evaluated in each cell, after which an upwind or central approximation is applied to arrive at an appropriate value at the cell face. How the kinetic energy in a cell is obtained

will be discussed in Section 4.6.2. It suffices here to state, because the convected term $\nabla\delta p$ is part of the kinetic energy, that the pressure gradient at its cell faces needs to be evaluated. Suppose a first order upwind interpolation is used, and the velocity is directed as indicated in Figure 3.1b. This means that, in order to obtain the pressure-correction equation for the shaded cell, the kinetic energy has to be evaluated in the cells indicated by \bullet . In order to obtain the kinetic energy in these cells, the pressure gradient has to be evaluated at the faces indicated by \circ . As a consequence, the stencil for the pressure-correction equation is enlarged, see Figure 3.1b, and this is what we do not want to happen. Hence, we need to take the kinetic energy in the convection term at the \star level instead of the new time level. This means that we approximate the kinetic energy in the second line of (3.48) by:

$$(\rho\mathbf{u}^2)^{n+1} = (\mathbf{m}^2/\rho)^{n+1} \approx (\mathbf{m}^*)^2/\rho^{n+1}. \quad (3.49)$$

This will not affect the scheme in the limit $M_r \downarrow 0$, and the standard 10 point stencil is maintained, because the convected term containing $\nabla\delta p$ has disappeared. It is clear that the $(\nabla \cdot \mathbf{u})$ -term in the last line of (3.48) has to be discretized implicitly in order to have the scheme reduce to the standard incompressible flow pressure-correction scheme ($M_r = 0$). In the derivation to arrive at (3.40), one finds that this term stems from the convection velocity of the pressure, which means that we also have to evaluate the convection velocity in the second line of (3.48) at the new time level. These remarks all add up to:

$$\begin{aligned} M_r^2 \left\{ \frac{\delta p}{\Delta t} + \frac{1}{2}(\gamma - 1) \frac{(\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} - (\mathbf{m}^n)^2 / \rho^n}{\Delta t} + \right. \\ \left. + \nabla \cdot \left[\left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) \left(\gamma(p^n + \delta p) + \frac{1}{2}(\gamma - 1)(\mathbf{m}^*)^2 / \rho^{n+1} \right) \right] \right\} + \\ + \nabla \cdot \left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) = 0. \end{aligned} \quad (3.50)$$

We could decide to omit the term $-\Delta t \nabla \delta p$ in the time derivative of (3.50). Numerical experiments, see Section 6.4.5, show that linearization rather than omission of this term enhances stability significantly. Linearization, i.e. keeping terms linear in δp and omitting the higher order terms, and some rearranging results in:

$$\begin{aligned} M_r^2 \left\{ \frac{\delta p}{\Delta t} + \frac{1}{2}(\gamma - 1) \frac{[(\mathbf{m}^*)^2 - 2\Delta t \mathbf{m}^* \cdot \nabla \delta p] / \rho^{n+1} - (\mathbf{m}^n)^2 / \rho^n}{\Delta t} \right\} + \\ + \nabla \cdot \left[\mathbf{u}^* \left(1 + \gamma M_r^2 (p^n + \delta p) + \frac{1}{2}(\gamma - 1) M_r^2 (\mathbf{m}^*)^2 / \rho^{n+1} \right) \right] - \\ - \Delta t \nabla \cdot \left\{ \left[\left(1 + \gamma M_r^2 p^n + \frac{1}{2}(\gamma - 1) M_r^2 (\mathbf{m}^*)^2 / \rho^{n+1} \right) / \rho^{n+1} \right] \nabla \delta p \right\} = 0. \end{aligned} \quad (3.51)$$

After some manipulations one can prove that $H = \text{constant}$ is, as it should, a steady state solution of (3.51). For $M_r \downarrow 0$, the pressure-correction equation for incompressible flows is recovered. Discretization of (3.44) and (3.51) forms the topic of Section 4.5.

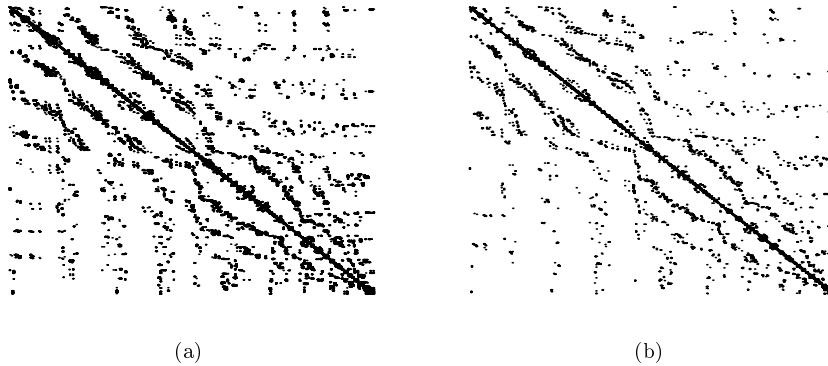


Figure 3.2: Nonzero structure of the momentum (a) and pressure-correction matrix (b) for the flow around an airfoil.

3.6 Linear solver

In the various sequential update procedures discussed in Sections 3.2, 3.3 and 3.5, linear systems have to be solved for the primary variables. These systems will be denoted as

$$Ax = b, \quad (3.52)$$

where A is an $N \times N$ matrix, b is the right-hand side, and x the solution vector. The number of unknowns N is equal to the number of cells for the scalar equations and the number of cell faces for the momentum equation. Information on the linear solver and the preconditioner that we use is given in the following sections.

3.6.1 Krylov subspace methods

Matrix A in (3.52) is in general large and sparse, and the nonzero entries are, because of the grids being unstructured, not in a structured (e.g. block diagonal) form. An example of the nonzero structure of the momentum and pressure-correction matrix is illustrated in Figure 3.2. Therefore it is attractive, because of efficiency and simplicity reasons, to use A only to multiply with. This consideration leads for solution of (3.52) to the use of Krylov subspace methods, which are iterative methods that look for optimal approximations to $x - x^{(0)}$, with $x^{(0)}$ a given start vector, in the Krylov subspace. The Krylov subspace is the m -dimensional ($m \leq N$) space spanned by the residuals $r^{(i)} = b - Ax^{(i)}$ of the iterates $x^{(i)}$, $i = 0, \dots, m - 1$.

For symmetric positive definite matrices A , the conjugate gradient (CG) method is the method of choice, since it satisfies an optimality property (the error, measured in some norm, is minimal) while only information of the previous iterate is needed.

The latter property, implying the use of short recurrences, is desirable, because it prevents increasing amounts of work per iteration and memory as the iteration process progresses. For general matrices, it is impossible to construct a Krylov subspace method that has both the optimality property and the pleasant property of short recurrences. In this situation, two avenues can be taken:

1. *Methods based on short recurrences.*

Examples of these methods are Bi-CG (bi-conjugate gradient), CGS (conjugate gradient squared) and Bi-CGSTAB (bi-conjugate gradient stabilized). Because these methods do not satisfy the optimality property, convergence is not guaranteed and breakdown may occur. Bi-CGSTAB, proposed in [113], is probably the most favored among the methods that are based on short recurrences, since no multiplication by A^T is required (as it is for Bi-CG) and its convergence behavior is smoother than that of CGS and Bi-CG, reducing the undesirable effects of round-off errors ('truncation'). Per iteration, Bi-CGSTAB requires two matrix-vector multiplications and only a few vector updates and inner products.

2. *Methods based on the optimality property.*

This type of methods satisfies the optimality property, and consequently is robust, but uses long recurrences. As a consequence, storage requirements and the amount of work per iteration increase as the number of iterations increases. In order to prevent this from becoming too time and memory consuming, these methods need to be restarted or truncated if the number of iterations has grown too large. Restart after r iterations means restart of the algorithm with $x^{(0)} := x^{(r-1)}$ as new start vector. An alternative is to allow only t search directions, and to overwrite old search directions by new ones once the number of iterations exceeds t ; this is called truncation. Of course, the optimality property gets lost when restart or truncation takes place. Examples of this kind of methods are GMRES (generalized minimum residual), GCR (generalized conjugate residual) and GMRESR (GMRES recursive). GMRES [93], having identical convergence properties as GCR while being cheaper, is the most popular among these methods. Per iteration, GMRES requires one matrix-vector multiplication, while the amount of work in the Gram-Schmidt procedure grows quadratically with the number of iterations.

The equations for the momentum, density and energy are solved with help of GMRES, whereas the pressure-correction equation is solved by Bi-CGSTAB; see also the discussion in Section 5.1.3. The solution vector obtained at the previous time level is used as start vector. Restart of GMRES for the density and energy equation is done after 40 iterations, and after 20 iterations for the momentum equation. If the norm of the residual $r^{(k)} = b - Ax^{(k)}$, with $x^{(k)}$ the approximate solution after k iterations, is less than $\epsilon \|r^{(0)}\|$, the iterative process is assumed to be converged. For

the momentum and pressure-correction equation we use $\epsilon = 10^{-6}$, and for the other equations we use $\epsilon = 10^{-4}$; these turn out to be good stopping criteria, see [125].

3.6.2 Preconditioning

Since the convergence behavior of Krylov subspace methods depends critically on the spectrum of A , it is worthwhile to cluster the eigenvalues. To achieve this, (3.52) is replaced by the preconditioned system:

$$A'x = b'; \quad A' = P^{-1}A, \quad b' = P^{-1}b, \quad (3.53)$$

where P is a preconditioner. This preconditioning should not be confused with the preconditioning that one applies to arrive at unified compressible/incompressible flow methods, see Section 3.4.2. Several choices for P are possible, all striving for: (i) the eigenvalues of P are close to those of A , and (ii) computation of P^{-1} , the inverse of P , is ‘cheap’. In our work, see also [125], ILU is used as preconditioner for all equations apart from the momentum equation, for which ILUD is employed. We will describe these preconditioners briefly.

Incomplete LU factorization (ILU)

The ILU-preconditioner [76] is given by $P = LU$, with L and U a lower and upper triangular matrix, respectively. The nonzero structure of the matrices L and U is comparable to the corresponding parts in A , and L and U are determined by the following requirements: (i) $(LU)_{ij} = a_{ij}$ for all matrix elements $a_{ij} \neq 0$, and (ii) the diagonal elements of L are equal to 1.

ILUD

This preconditioner is given by $P = LD^{-1}U$, where D , L and U are a diagonal, lower and upper triangular matrix, respectively. These matrices follow from: (i) the main diagonals of L , U and the diagonal matrix D are the same, (ii) the off-diagonal parts of L and U are equal to the corresponding parts in A , and (iii) the main diagonal of P equals the main diagonal of A .

Chapter 4

Spatial discretization of the flow equations on unstructured staggered grids

A common feature of most established methods for compressible flows is the use of colocated schemes. For incompressible flows, a straightforward discretization on a colocated grid leads to odd-even oscillations of the pressure, see also Section 2.4.2. To remedy this, artificial stabilizing measures have to be taken. The most popular method that has evolved is the pressure-weighted interpolation of Rhie and Chow [90], by which artificial pressure diffusion is introduced in the mass conservation law. This problem of spurious pressure oscillations does not occur with staggered schemes. The difficulty also does not arise in the compressible flow case. Because on non-orthogonal grids colocated discretization is more straightforward than staggered discretization, colocated schemes are prevalent for fully compressible flows, and have reached a certain degree of maturity. However, staggered schemes can be devised that are accurate on highly non-orthogonal grids, see [129, 130, 131]. Furthermore, the classic incompressible staggered scheme of [44] can be applied to compressible flows, as shown already in [42, 43]. For more recent work in this direction, see [9, 110, 111, 132] and references quoted there.

In this chapter, spatial discretization of the flow equations on unstructured staggered grids is discussed. The staggered positioning of the variables on triangular grids is introduced in Section 4.1. Due to the staggered grid arrangement, it is not possible to define the usual ‘left’ and ‘right’ states containing all primitive variables. This explains why schemes based on Riemann solvers cannot be applied. On the other hand, discretization by a simple finite difference or finite volume scheme for each primary variable separately is natural on a staggered grid. It is also natural to update the primary variables sequentially in a time-stepping or iterative procedure,

as discussed in Chapter 3. Three different segregated approaches were proposed: the pressure-correction approach (PC) for incompressible flows (Section 3.2), a sequential update procedure for fully compressible (FC) flows (Section 3.3) and a Mach-uniform (MU) solution algorithm (Section 3.5). We have adopted a finite volume approach, ensuring conservation of mass, momentum and energy in the numerical scheme; see also Appendix D. Discretization of the momentum equation, which is the same for all three solution procedures, forms the subject of Section 4.2. In Section 4.3 the discretization of the continuity equation is discussed. Discretization of the energy equation as encountered in the FC approach is the topic of Section 4.4. Discretization of the energy equation in the MU approach, which is the Mach-uniform pressure-correction equation, is given in Section 4.5. The incompressible flow pressure-correction equation is a special form of the Mach-uniform pressure-correction equation, and will not be discussed separately. Evaluation of the equation of state is dealt with in Section 4.6, and some remarks concerning postprocessing are gathered in Section 4.7.

4.1 Unstructured staggered grids

We will restrict ourselves, although this is not a prerequisite, to two dimensions, and consider grids consisting solely of triangles. In Figure 4.1 the employed staggered placement of variables in the grid is shown. Throughout this thesis, the words ‘triangle’ and ‘cell’, and ‘edge’ and ‘face’ will refer to the same objects. At the cell centroids the scalar variables, e.g. p , ρ , h and ρH , are located. The normal momentum components m are stored at the midpoints of the faces. This placement of the variables is similar to the classic staggered scheme on structured grids with quadrilateral cells as introduced by Harlow and Welch [44] and which is used by our group on boundary-fitted grids in, for example, [9, 110, 111, 130, 131, 133]. Note that, in contrast with the covolume method or the method of Perot that are discussed in Section 2.4.3, we do not put the scalar variables in the circumcenters and that we do not require grids to be of Delaunay type.

At every face e there are two unit normal vectors \mathbf{n}_e , pointing in opposite directions. By some unambiguous procedure we select at each face one of these to be the so-called unique normal vector, indicated by \mathbf{N}_e . Obviously,

$$\mathbf{N}_e = (\mathbf{n}_e \cdot \mathbf{N}_e) \mathbf{n}_e, \quad \mathbf{n}_e = (\mathbf{n}_e \cdot \mathbf{N}_e) \mathbf{N}_e, \quad (\mathbf{n}_e \cdot \mathbf{N}_e) = \pm 1. \quad (4.1)$$

Among the number of faces E , the number of vertices V and the number of cells C there exists for 2D triangular grids with H holes a simple relation [29]:

$$C + V = E + 1 - H. \quad (4.2)$$

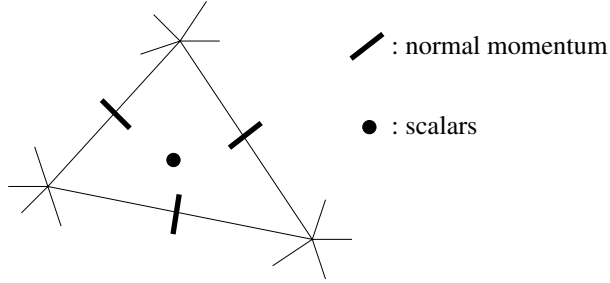


Figure 4.1: Staggered positioning of the variables in an unstructured grid.

Another useful equality, derived in [29] as well, is

$$C = \frac{1}{3}(E + E_i), \quad (4.3)$$

with E_i the number of internal faces. In the (common) situation that the number of boundary faces is negligible compared to the total number of faces ($E \approx E_i$), the approximations:

$$E \approx 3V, \quad C \approx 2V, \quad (4.4)$$

are useful.

4.2 Discretization of the momentum equation

At the midpoint of each face i , the normal component of the momentum vector $m_i = \mathbf{m}_i \cdot \mathbf{N}_i$ is stored. The first step in deriving a discrete equation for m_i is to project the momentum equation (2.2) on the unique normal \mathbf{N}_i at face i :

$$\frac{\partial m_i}{\partial t} + \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] = -\nabla p \cdot \mathbf{N}_i + \boldsymbol{\tau} \cdot \mathbf{N}_i. \quad (4.5)$$

We use a finite volume approach, and consequently a control volume has to be specified. This is the topic of Section 4.2.1. Discretization of the time derivative is the subject of Section 4.2.2. Central and first order upwind schemes for the inertia term are introduced in Sections 4.2.3 and 4.2.4. Various possibilities to approximate the pressure gradient are discussed in Section 4.2.5, and in Section 4.2.6 the discretization of the viscous term is given. In order to maintain completeness while avoiding lengthy digressions from the main line of reasoning, some parts have been moved to appendices. A discussion on monotonicity together with a suitable definition of the Courant number is given in Appendix A. Consistency of various schemes for the inertia term is checked numerically, see Appendix B for more details. Whether the discretization preserves symmetry is discussed in Appendix C, and

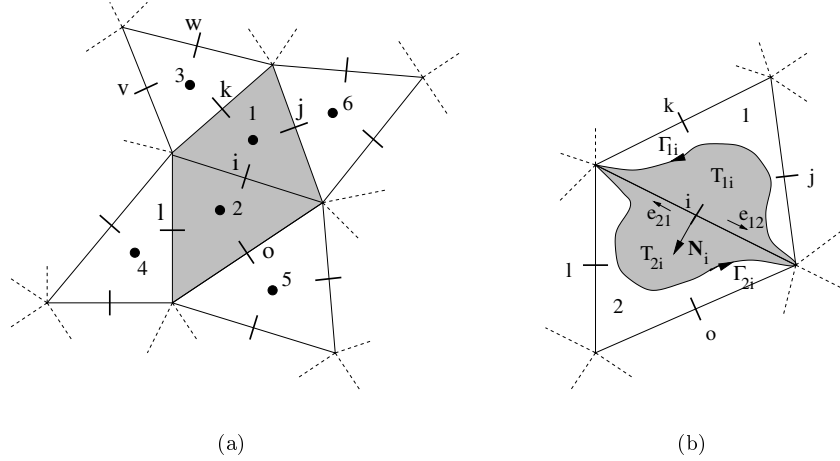


Figure 4.2: The two options that we studied for the CV for the normal momentum equation at face i are shaded. Numbers denote cells, while faces are indicated by letters.

in Appendix D it is demonstrated that the resulting finite volume discretization conserves momentum.

4.2.1 Choice of control volume

We integrate (4.5) over a suitably chosen control volume (CV). There are various possibilities to select a CV for the momentum equation on unstructured staggered grids. We have investigated two options.

Option 1. Integration over two triangles

We choose the union of the two triangles adjacent to face i , see Figure 4.2a, as CV for the normal momentum component m_i located at face i . For a boundary face, the CV is the corresponding boundary cell.

Option 2. Integration over two half triangles

The shaded region in Figure 4.2b that is formed by uniting T_{1i} and T_{2i} forms the CV. We construct T_{1i} , the part of the CV for face i that lies in cell 1, in the following manner: (i) face i is part of the boundary of T_{1i} , (ii) T_{1i} is inside cell 1, and (iii) the area of T_{1i} is half the area of cell 1. Of course, T_{2i} is constructed in the same fashion. The boundary of T_{1i} is given by $\partial T_{1i} = \Gamma_{1i} \cup e_{12}$, and is traversed in counterclockwise direction. Here e_{12} represents face i , with the first index referring to the cell lying left and Γ_{1i} forms, as sketched in Figure 4.2b, the remainder of ∂T_{1i} . Assuming that \mathbf{N}_i points from cell 1 to cell 2, the outward normal vector \mathbf{n}_{12} at e_{12} is pointing in

the same direction. Similarly, we have $\partial T_{2i} = \Gamma_{2i} \cup e_{21}$ and $\mathbf{n}_{21} = -\mathbf{N}_i$. The CV at face i is formed by $T_{1i} \cup T_{2i}$, and its boundary is given by $\Gamma_i = \Gamma_{1i} \cup \Gamma_{2i}$. Note that we do not specify the precise shape of Γ_{1i} and Γ_{2i} .

Discussion

In Section 4.2.4 we give a motivation for proposing option 2 that is based on a projection onto the one-dimensional staggered scheme. Numerical experiments, see Appendix B, reveal that option 1 leads to a consistent scheme, and option 2 does not. Hence, unless stated otherwise, the CV for the momentum equation consists of the two triangles adjacent to the considered face.

4.2.2 Discretization of the time derivative

Integration over the CV of the time derivative in (4.5) is done in the usual manner:

$$\int_{\text{CV}} \frac{dm_i}{dt} d\mathbf{x} \approx \Omega_i \frac{m_i^{n+1} - m_i^n}{\Delta t}, \quad (4.6)$$

where Ω_i is the area of the CV, $\Delta t = t^{n+1} - t^n$ the time step, and the superscript refers to the time level. In the incompressible flow and Mach-uniform pressure-correction approach, m_i^{n+1} should be considered as the predictor m_i^* , since afterwards a correction is carried out.

4.2.3 Discretization of the inertia term (two triangles)

Integration of the inertia term over two triangles results in:

$$\int_{\text{CV}} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] d\mathbf{x} = \oint_{\partial \text{CV}} (\mathbf{u} \cdot \mathbf{n})(\mathbf{m} \cdot \mathbf{N}_i) d\Gamma \approx \sum_{e(i)} (\mathbf{u}_e \cdot \mathbf{N}_e)(\mathbf{m}_e \cdot \mathbf{N}_i) \bar{l}_e, \quad (4.7)$$

where \mathbf{n} is the outward unit normal at the boundary of the CV. In the last step we used relation (4.1), and we define

$$\bar{l}_e = l_e (\mathbf{n}_e \cdot \mathbf{N}_e). \quad (4.8)$$

The summation is over the faces of the CV, i.e. summation over the faces in $e(i) \in \{k, l, o, j\}$, see Figure 4.2a. Before we specify how the convecting velocity $\mathbf{u}_e \cdot \mathbf{N}_e$ and convected projected momentum $\mathbf{m}_e \cdot \mathbf{N}_i$ are obtained, a procedure to reconstruct vector quantities needs to be introduced.

Reconstruction procedure

Because solely the normal component of the momentum is stored at the cell faces, some sort of interpolation needs to be performed in order to arrive at components pointing in other directions. This can be done by what we will call the reconstruction procedure, which is introduced here for two dimensions. Extension to more

dimensions is straightforward. Let the vectors \mathbf{V}_i , \mathbf{N}_j and \mathbf{N}_k be given and let \mathbf{N}_j and \mathbf{N}_k not be parallel to each other. Here, i , j and k indicate faces. Suppose further that $q_j = \mathbf{q}_j \cdot \mathbf{N}_j$ and $q_k = \mathbf{q}_k \cdot \mathbf{N}_k$ are given. Here, $\mathbf{q} = \mathbf{q}(\mathbf{x})$ represents an unknown vector field and $\mathbf{q}_j = \mathbf{q}(\mathbf{x}_j)$, with \mathbf{x}_j the coordinates of the midpoint of face j . Our aim is, using the specified quantities, to find an approximation for $q_i = \mathbf{q}_i \cdot \mathbf{V}_i$. Because \mathbf{N}_j and \mathbf{N}_k are linearly independent, there exists a unique solution to the reconstruction coefficients χ_j^i and χ_k^i , which are defined by means of the expression:

$$\mathbf{V}_i = \chi_j^i \mathbf{N}_j + \chi_k^i \mathbf{N}_k. \quad (4.9)$$

Let \mathbf{t}_j and \mathbf{t}_k be tangential vectors at faces j and k , respectively, i.e. $\mathbf{t}_j \cdot \mathbf{N}_j = \mathbf{t}_k \cdot \mathbf{N}_k = 0$, then

$$\chi_j^i = (\mathbf{V}_i \cdot \mathbf{t}_k) / (\mathbf{N}_j \cdot \mathbf{t}_k), \quad \chi_k^i = (\mathbf{V}_i \cdot \mathbf{t}_j) / (\mathbf{N}_k \cdot \mathbf{t}_j). \quad (4.10)$$

An approximation for q_i follows from:

$$\begin{aligned} q_i &= \mathbf{q}_i \cdot \mathbf{V}_i = \mathbf{q}_i \cdot (\chi_j^i \mathbf{N}_j + \chi_k^i \mathbf{N}_k) = \chi_j^i (\mathbf{q}_i \cdot \mathbf{N}_j) + \chi_k^i (\mathbf{q}_i \cdot \mathbf{N}_k) \approx \\ &\approx \chi_j^i (\mathbf{q}_j \cdot \mathbf{N}_j) + \chi_k^i (\mathbf{q}_k \cdot \mathbf{N}_k) = \chi_j^i q_j + \chi_k^i q_k, \end{aligned} \quad (4.11)$$

with equality for constant vector fields $\mathbf{q}(\mathbf{x}) = \mathbf{q}_0$. In Section 4.3 we will prove that the reconstruction procedure satisfies a nice property for incompressible flows and for stationary compressible flows. Relation (4.11) forms, with the reconstruction coefficients defined in expression (4.9), the basis of the reconstruction procedure. Similar procedures are introduced in the covolume method, see also Section 2.4.3 for suitable references.

Computation of the convecting velocity

We have devised two ways to compute the convecting velocity $\mathbf{u}_e \cdot \mathbf{N}_e$.

1. Central approximation

At face k , the normal velocity $u_k = \mathbf{u}_k \cdot \mathbf{N}_k$ follows from

$$u_k = \frac{m_k}{\rho_{k,av}}, \quad \rho_{k,av} = \frac{\Omega_3}{\Omega_1 + \Omega_3} \rho_1 + \frac{\Omega_1}{\Omega_1 + \Omega_3} \rho_3. \quad (4.12)$$

Here m_k is the given normal momentum component and $\rho_{k,av}$ approximates the density at face k by means of a weighted averaging. Here, Ω_1 and Ω_3 stand for the areas of the adjacent cells, see Figure 4.2a. Note that in case the density is constant, as is often the case for incompressible flows, then (4.12) is exact. This is an advantage of using the CV consisting of two triangles.

2. Upwind approximation

Motivated by arguments related to monotonicity preservation, see Appendix A.2, an upwind approximation for evaluation of the convecting velocity is developed. For

the reconstruction coefficients ζ that are defined by, see Figure 4.2a for notation:

$$\mathbf{N}_k = \zeta_i \mathbf{N}_i + \zeta_j \mathbf{N}_j, \quad \mathbf{N}_k = \zeta_v \mathbf{N}_v + \zeta_w \mathbf{N}_w, \quad (4.13)$$

a unique solution exists. Inspired by (4.11), we arrive at the following upwind approximation for the convecting velocity at face k :

$$\mathbf{u}_k \cdot \mathbf{N}_k = \mathbf{m}_k \cdot \mathbf{N}_k / \rho_k = \begin{cases} (\zeta_i m_i + \zeta_j m_j) / \rho_1 & \text{if } m_k \bar{l}_k > 0; \\ (\zeta_v m_v + \zeta_w m_w) / \rho_3 & \text{if } m_k \bar{l}_k < 0; \\ 0 & \text{if } m_k = 0. \end{cases} \quad (4.14)$$

The flow is directed from cell 1 to 3 if $m_k \bar{l}_k > 0$, and vice versa.

In Section 6.1 numerical results obtained by both methods for the Burgers equation will be compared. It will turn out that the central approximation (4.12) is to be preferred in most cases, since it gives crisper shock resolution while spurious oscillations are almost absent.

First order upwind discretization of the inertia term

In order to compute the convected momentum terms $\mathbf{m}_e \cdot \mathbf{N}_i$ at the four CV faces, we want to restrict ourselves to the momentum stencil consisting of the 13 faces indicated in Figure 4.2a. This limits the computation of $\mathbf{m}_k \cdot \mathbf{N}_i$ to the use of normal momentum components at the faces i , j , k , v and w . Assume for the moment that the flow is directed from cell 3 to 1. We decide to use m_v and m_w to approximate $\mathbf{m}_k \cdot \mathbf{N}_i$. Since \mathbf{N}_v and \mathbf{N}_w cannot be parallel to each other, there always exists a unique solution for the reconstruction coefficients η defined by:

$$\mathbf{N}_i = \eta_v \mathbf{N}_v + \eta_w \mathbf{N}_w. \quad (4.15)$$

Substitution gives, cf. (4.11):

$$\mathbf{m}_k \cdot \mathbf{N}_i = \eta_v m_v + \eta_w m_w, \quad (4.16)$$

which is exact for constant momentum vector fields. If the fluid moves from cell 1 to 3, $\mathbf{m}_k \cdot \mathbf{N}_i$ is approximated in terms of m_i and m_j . In the same way as before we now arrive at

$$\mathbf{m}_k \cdot \mathbf{N}_i \approx m_i. \quad (4.17)$$

Summarizing, the first order upwind approximation for $\mathbf{m}_k \cdot \mathbf{N}_i$ is:

$$\mathbf{m}_k \cdot \mathbf{N}_i = \begin{cases} \eta_v m_v + \eta_w m_w & \text{if } u_k \bar{l}_k < 0; \\ m_i & \text{if } u_k \bar{l}_k > 0. \end{cases} \quad (4.18)$$

Note that when the flow is directed outwards of the CV, the main diagonal is increased. Properties related to monotonicity preservation of the first order upwind scheme are investigated in more detail in Appendix A.

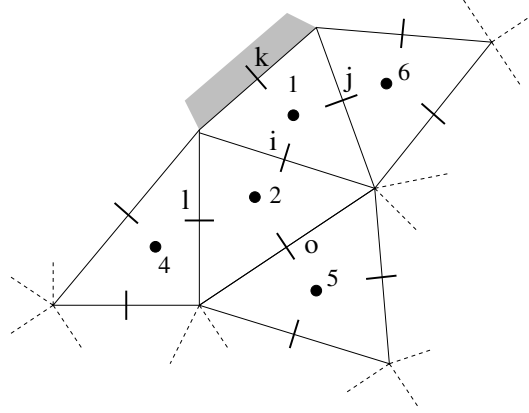


Figure 4.3: Stencil for the momentum equation in the neighborhood of boundaries.

Another upwind scheme that we have investigated is based on the notion that we can decompose the momentum field at face k on a local coordinate system formed by \mathbf{N}_k and \mathbf{t}_k :

$$\mathbf{m}_k = m_k \mathbf{N}_k + \tilde{m}_k \mathbf{t}_k, \quad (4.19)$$

where \mathbf{t}_k is a tangential vector at face k . Here, the normal momentum component m_k needs not be interpolated (it is already located at the appropriate position), while the tangential momentum component $\tilde{m}_k = \mathbf{m}_k \cdot \mathbf{t}_k$ follows from the reconstruction procedure. Let

$$\mathbf{t}_k = \xi_i \mathbf{N}_i + \xi_j \mathbf{N}_j, \quad \mathbf{t}_k = \xi_v \mathbf{N}_v + \xi_w \mathbf{N}_w \quad (4.20)$$

define the reconstruction coefficients ξ , then the following upwind scheme for \tilde{m}_k is proposed:

$$\tilde{m}_k = \begin{cases} \xi_v m_v + \xi_w m_w & \text{if } u_k \bar{l}_k < 0; \\ \xi_i m_i + \xi_j m_j & \text{if } u_k \bar{l}_k > 0. \end{cases} \quad (4.21)$$

In Appendix B we will show that this leads to an inconsistent scheme.

Central discretization of the inertia term

The following central scheme at face k is proposed:

$$\mathbf{m}_k \cdot \mathbf{N}_i = \frac{1}{2}(\eta_v m_v + \eta_w m_w + m_i). \quad (4.22)$$

In Appendix C we show that this scheme preserves, on regularly shaped grids, the symmetry properties of the underlying continuous operator.

Boundary conditions

Let face k be part of the boundary, see Figure 4.3. Consider the momentum equation for face i . The momentum vector at face k is decomposed, cf. (4.19), on a normal

and tangential component. The reason to do so is that we need to be able to cope with situations in which the normal and/or tangential momentum component are given. It is obvious that

$$\mathbf{m}_k \cdot \mathbf{N}_i = m_k(\mathbf{N}_k \cdot \mathbf{N}_i) + \tilde{m}_k(\mathbf{t}_k \cdot \mathbf{N}_i). \quad (4.23)$$

Dealing with the contribution $m_k(\mathbf{N}_k \cdot \mathbf{N}_i)u_k \bar{l}_k$ is trivial. If the tangential momentum component \tilde{m}_k is given, then the term $-\tilde{m}_k(\mathbf{t}_k \cdot \mathbf{N}_i)u_k \bar{l}_k$ is added to the right-hand side. If, on the other hand, \tilde{m}_k is not given, the reconstruction procedure is applied to approximate this momentum component:

$$\mathbf{N}_k = \zeta_i \mathbf{N}_i + \zeta_j \mathbf{N}_j, \quad \tilde{m}_k = \zeta_i m_i + \zeta_j m_j. \quad (4.24)$$

Next, we consider the momentum equation for boundary face k , and we omit from the discussion the trivial case in which m_k is given. Integration of the inertia term over the CV, now consisting of boundary cell 1, leads to:

$$\int_{\text{CV}} \nabla \cdot [(\mathbf{m} \cdot \mathbf{N}_k) \mathbf{u}] d\mathbf{x} \approx \sum_{e(k)} (\mathbf{m}_e \cdot \mathbf{N}_k) u_e \bar{l}_e, \quad (4.25)$$

where the summation runs over the faces in $e(k) \in \{i, j, k\}$. The contributions stemming from faces j and k fit in the framework given above. With

$$(\mathbf{m}_k \cdot \mathbf{N}_k) u_k \bar{l}_k = m_k u_k \bar{l}_k, \quad (4.26)$$

the contribution from face k itself is approximated, resulting in an enlargement of the main diagonal. Note that (4.26) is in correspondence with (4.23), and that consequently conservation is ensured; see also Appendix D.2.

4.2.4 Discretization of the inertia term (two half triangles)

Now we turn to discuss the discretization of the inertia term that results from choosing two half triangles as control volume (i.e. option 2, see Section 4.2.1). Integration of the inertia term over this CV, indicated by the shaded region in Figure 4.2b, results in:

$$\begin{aligned} \int_{\text{CV}} \nabla \cdot [(\mathbf{m} \cdot \mathbf{N}_i) \mathbf{u}] d\mathbf{x} &= \oint_{\Gamma_i} (\mathbf{m} \cdot \mathbf{N}_i) (\mathbf{u} \cdot \mathbf{n}) d\Gamma = \left[\int_{\Gamma_{1i}} + \int_{\Gamma_{2i}} \right] (\mathbf{m} \cdot \mathbf{N}_i) (\mathbf{u} \cdot \mathbf{n}) d\Gamma \approx \\ &\approx (\mathbf{m}_1 \cdot \mathbf{N}_i) \mathbf{u}_1 \cdot \int_{\Gamma_{1i}} \mathbf{n} d\Gamma + (\mathbf{m}_2 \cdot \mathbf{N}_i) \mathbf{u}_2 \cdot \int_{\Gamma_{2i}} \mathbf{n} d\Gamma. \end{aligned} \quad (4.27)$$

Here \mathbf{n} is the outward pointing normal, and \mathbf{m}_1 , \mathbf{m}_2 , \mathbf{u}_1 and \mathbf{u}_2 represent the momentum and velocity vectors, which we approximate by a constant, in the corresponding

cells. Since $\oint \mathbf{n} d\Gamma = \mathbf{0}$ for every closed contour, we can write (assuming that \mathbf{N}_i points from cell 1 to cell 2):

$$\oint_{\partial T_{1i}} \mathbf{n} d\Gamma = \int_{\Gamma_{1i}} \mathbf{n} d\Gamma + \int_{e_{12}} \mathbf{n} d\Gamma = \int_{\Gamma_{1i}} \mathbf{n} d\Gamma + l_i \mathbf{N}_i = \mathbf{0}. \quad (4.28)$$

Doing the same for ∂T_{21} results in the identities:

$$\int_{\Gamma_{1i}} \mathbf{n} d\Gamma = -l_i \mathbf{N}_i, \quad \int_{\Gamma_{2i}} \mathbf{n} d\Gamma = l_i \mathbf{N}_i. \quad (4.29)$$

Insertion of these relations in (4.27) leads to:

$$\int_{CV} \nabla \cdot [(\mathbf{m} \cdot \mathbf{N}_i) \mathbf{u}] dx \approx l_i [(\mathbf{m}_2 \cdot \mathbf{N}_i)(\mathbf{u}_2 \cdot \mathbf{N}_i) - (\mathbf{m}_1 \cdot \mathbf{N}_i)(\mathbf{u}_1 \cdot \mathbf{N}_i)]. \quad (4.30)$$

Approximating \mathbf{u} and \mathbf{m} in the cell centers requires application of the reconstruction procedure. When the flow is directed from cell 1 to 2, the following upwind approximation in cell 1 is made:

$$\begin{aligned} [\mathbf{m}_1 \cdot \mathbf{N}_i][\mathbf{u}_1 \cdot \mathbf{N}_i] &= [\mathbf{m}_1 \cdot (\eta_j \mathbf{N}_j + \eta_k \mathbf{N}_k)][\mathbf{u}_1 \cdot (\eta_j \mathbf{N}_j + \eta_k \mathbf{N}_k)] \approx \\ &\approx (\eta_j m_j + \eta_k m_k)(\eta_j u_j + \eta_k u_k), \end{aligned} \quad (4.31)$$

which is exact for constant momentum. We use

$$[\mathbf{m}_1 \cdot \mathbf{N}_i][\mathbf{u}_1 \cdot \mathbf{N}_i] \approx u_i m_i. \quad (4.32)$$

for a flow in the opposite direction.

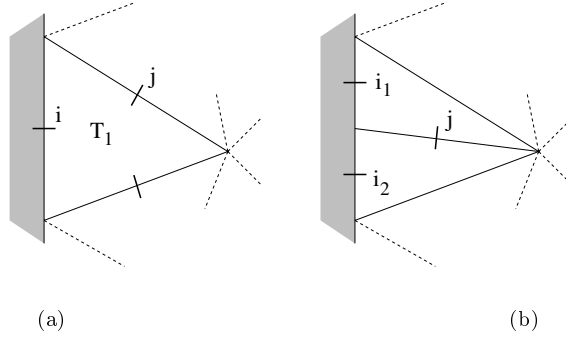
Boundary conditions

The following types of boundary conditions are encountered in the consistency tests for the inertia term described in Appendix B: (i) at inflow boundaries the momentum vector is given, and (ii) at outflow boundaries nothing is given. At faces located at the inflow boundaries, discretization of the momentum equation is trivial. A remaining question is the incorporation of a given tangential momentum component in the discretization. For this we propose the following. Let at boundary faces i (Figure 4.4a), i_1 and i_2 (Figure 4.4b) the momentum vector be given. In Figure 4.4a, face j is part of boundary cell T_1 which has one boundary face. At face j we prescribe the normal momentum component, with \mathbf{m}_i the given momentum vector at face i , using

$$m_j = \mathbf{m}_i \cdot \mathbf{N}_j. \quad (4.33)$$

In Figure 4.4b, the boundary cells connected to internal face j have two boundary faces in total. In this situation, we prescribe the normal momentum component at internal face j using

$$m_j = \frac{1}{2}[(\mathbf{m}_{i_1} \cdot \mathbf{N}_j) + (\mathbf{m}_{i_2} \cdot \mathbf{N}_j)]. \quad (4.34)$$

Figure 4.4: Face j in the vicinity of boundaries.

A similar procedure is followed when an internal face is surrounded by three boundary faces at which the momentum vector is given. Since \mathbf{N}_j is not parallel to \mathbf{N}_i , \mathbf{N}_{i_1} or \mathbf{N}_{i_2} , the prescribed tangential momentum is embedded in the numerical solution. We will show in Appendix B that discretization on half triangles is inconsistent, hence there is no need to improve upon the proposed discretization of the boundary conditions. When at a boundary face i the normal momentum component is not given, we integrate over half of the boundary triangle, resulting in:

$$\int_{T_{1i}} \nabla \cdot [\mathbf{u}(\mathbf{m} \cdot \mathbf{N}_i)] d\mathbf{x} \approx l_i [u_i m_i - (\mathbf{m}_1 \cdot \mathbf{N}_1)(\mathbf{u}_1 \cdot \mathbf{N}_1)] \quad (4.35)$$

where $(\mathbf{m}_1 \cdot \mathbf{N}_1)(\mathbf{u}_1 \cdot \mathbf{N}_1)$ is evaluated as in (4.31).

Motivation for selecting two half triangles as control volume

First, consider the 1D grid shown in Figure 4.5. The cell centers are located half way between the momentum grid points. Integration of the 1D momentum equation, with the pressure and viscous terms left out, over a CV formed by $[x_{i-1}, x_i]$ results in

$$\Delta x_i \frac{dm_i}{dt} + (um)_{\bar{i}} - (um)_{i-1} = 0, \quad (4.36)$$

where

$$\Delta x_i = x_{\bar{i}} - x_{i-1} = \frac{1}{2}(x_{i+1} - x_{i-1}). \quad (4.37)$$

It can be shown, see Appendix A.2, that upwind approximation of both the velocity and momentum, i.e.

$$(um)_{\bar{i}} = u_i m_i, \quad u > 0, \quad (4.38)$$

is a monotone discretization of the inertia term. The attractive property of monotone conservative schemes is that they avoid spurious modes and that they converge to weak solutions satisfying the entropy condition [47].

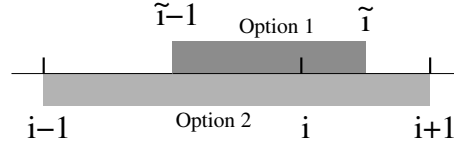


Figure 4.5: A 1D staggered grid. Cell centers are, in contrast with cell faces, indicated by tildes.

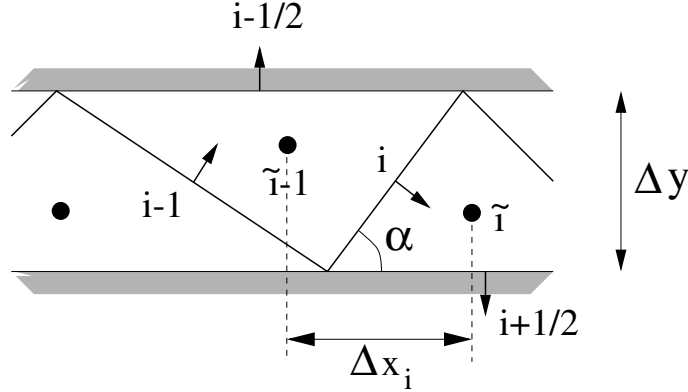


Figure 4.6: A 2D unstructured grid consisting of one row of triangles.

Next, consider the grid illustrated in Figure 4.6. By studying a 1D flow in the x -direction, which is realized by putting the normal momentum at all horizontal faces equal to zero, we can study some properties of a 2D discretization of the momentum equation by comparing it with the 1D discretization given above. We will refer to this as the quasi 1D test case. The unit normal at face i is given by $\mathbf{N}_i = (\sin \alpha, -\cos \alpha)$, and the length of face i equals $l_i = \Delta y / \sin \alpha$, with Δy the height of the domain. The 1D velocity and momentum vector fields are given by $\mathbf{u} = (\hat{u}, 0)$ and $\mathbf{m} = (\hat{m}, 0)$, where the hat is used to indicate the x -component. Our goal is to find a 2D discretization that, when applied to the quasi 1D test case, reduces to (4.36). We choose the CV as small as possible, and such that the ‘left’ and ‘right’ states in the inertia term are located in cells $\tilde{i} - 1$ and \tilde{i} . In order to keep the discretization dimensionally correct, the inertia term has to be multiplied by a vector quantity with the dimension of length, for which we take $l_i \mathbf{N}_i$. These considerations lead to:

$$\Omega_i \frac{d(\mathbf{m}_i \cdot \mathbf{N}_i)}{dt} + l_i [(\mathbf{m}_i \cdot \mathbf{N}_i)(\mathbf{u}_i \cdot \mathbf{N}_i) - (\mathbf{m}_{i-1} \cdot \mathbf{N}_i)(\mathbf{u}_{i-1} \cdot \mathbf{N}_i)] = 0, \quad (4.39)$$

where Ω_i is the area of the CV (yet to be determined). With

$$m_i = \mathbf{m}_i \cdot \mathbf{N}_i = \hat{m}_i \sin \alpha, \quad \mathbf{m}_i \cdot \mathbf{N}_i = \hat{m}_i \sin \alpha, \quad \mathbf{u}_i \cdot \mathbf{N}_i = \hat{u}_i \sin \alpha, \quad (4.40)$$

and similar relations for $(\mathbf{m}_{i-1} \cdot \mathbf{N}_i)$ and $(\mathbf{u}_{i-1} \cdot \mathbf{N}_i)$, expression (4.39) can be written as

$$\Omega_i \frac{\hat{m}_i^{n+1} - \hat{m}_i^n}{\Delta t} + l_i \sin \alpha [(\hat{u}\hat{m})_{\bar{i}} - (\hat{u}\hat{m})_{\bar{i}-1}] = 0. \quad (4.41)$$

If we choose Ω_i equal to half of the area of the two adjacent triangles, then (4.41) reduces to (4.36). This is easy to see, since

$$\Omega_i = (x_{i+1} - x_i)\Delta y \quad \Omega_{i-1} = (x_i - x_{i-1})\Delta y, \quad (4.42)$$

where x_i refers to the x -coordinate of the midpoint of face i . Inserting

$$\Omega_i = \frac{1}{2}(\Omega_{i-1} + \Omega_i) = \frac{1}{2}(x_{i+1} - x_{i-1})l_i \sin \alpha = \Delta x_i l_i \sin \alpha \quad (4.43)$$

into (4.41) leads to the aforementioned result. This result suggests that the CV for each face has to consist of the half of the two adjacent triangles. Application of upwind scheme (4.31)–(4.32) to (4.39) gives, after some manipulations and assuming that $\hat{u} > 0$:

$$\Delta x_i \frac{d\hat{m}_i}{dt} + (\hat{u}\hat{m})_i - (\hat{u}\hat{m})_{i-1} = 0, \quad (4.44)$$

which is identical to 1D upwind method (4.38). Hence, for the quasi 1D test case discussed here, we have demonstrated that integration of the momentum equation over half triangles results in the appropriate 1D discretization. However, in Appendix B we will show that the resulting method is not consistent when applied to 2D flows.

4.2.5 Discretization of the pressure gradient

Integration of the projected pressure gradient term in (4.5) over a CV with area Ω_i can be written as:

$$\int_{\text{CV}} \nabla p \cdot \mathbf{N}_i d\mathbf{x} = \Omega_i (\nabla p \cdot \mathbf{N})_i. \quad (4.45)$$

The projected pressure gradient will be approximated as follows:

$$(\nabla p \cdot \mathbf{N})_i = \sum_{j(i)} \gamma_j p_j, \quad (4.46)$$

where the summation runs over a set of cells surrounding face i , the so-called stencil for the pressure gradient. The gradient coefficients γ_j depend solely on the geometry of the grid and hence can be computed and stored in advance. For constant pressure fields, (4.46) reduces to

$$\sum_{j(i)} \gamma_j = 0, \quad (4.47)$$

which is a necessary requirement for consistency. We will discuss four schemes to compute the pressure gradient. First we formulate the path integral (PI) formulation for unstructured staggered grids. The next method uses a contour integral (CI)

formulation. Another method, based on the introduction of auxiliary points (AP), results in a simple finite difference equation. The three schemes above can yield a value for the pressure gradient that possibly even has the wrong sign. This might occur especially in the vicinity of steep gradients. We will formulate a requirement, the so-called sign-criterion, that the gradient coefficients γ_j must satisfy in order to have the correct sign. The four quadrant (FQ) method is designed with this criterion in mind. Computation of the pressure gradient at boundary faces is discussed.

Path integral formulation

The path integral (PI) formulation, introduced for the approximation of $(\nabla p \cdot \mathbf{N})_i$ on curvilinear structured staggered grids in [108], can be used equally well on unstructured grids. In this method first an approximation of the pressure gradient vector $(\nabla p)_i$ is made, after which the inner product with \mathbf{N}_i is taken. The path integral formulation is based on the identity

$$p_b - p_a = \int_a^b \nabla p \cdot d\mathbf{x}, \quad (4.48)$$

where a and b refer to points with coordinates \mathbf{x}_a and \mathbf{x}_b , and $p_a = p(\mathbf{x}_a)$ and $p_b = p(\mathbf{x}_b)$. This expression is approximated by

$$p_b - p_a \approx (\nabla p)_{ab} \cdot (\mathbf{x}_b - \mathbf{x}_a), \quad (4.49)$$

where \mathbf{x}_{ab} is a point in the vicinity of a and b . Application of (4.49) to a path from cell-center 1 to cell-center 2, see Figure 4.2a for notation, gives

$$p_2 - p_1 \approx (\nabla p)_i \cdot (\mathbf{x}_2 - \mathbf{x}_1), \quad (4.50)$$

where we assume, as is usually the case, that the midpoint of face i is located in the vicinity of the line connecting \mathbf{x}_1 and \mathbf{x}_2 . To obtain $(\nabla p)_i$ one additional relation is required. Application of (4.49) to the path from cell-center 5 to 3 leads to

$$p_3 - p_5 \approx (\nabla p)_i \cdot (\mathbf{x}_3 - \mathbf{x}_5), \quad (4.51)$$

and, similarly, to the path from cell-center 6 to 4:

$$p_4 - p_6 \approx (\nabla p)_i \cdot (\mathbf{x}_4 - \mathbf{x}_6). \quad (4.52)$$

Averaging these expressions results in:

$$p_3 - p_6 + p_4 - p_5 \approx (\nabla p)_i \cdot (\mathbf{x}_3 - \mathbf{x}_6 + \mathbf{x}_4 - \mathbf{x}_5). \quad (4.53)$$

Equations (4.50) and (4.53) together determine $(\nabla p)_i$. As a consequence, the stencil for the pressure gradient consists of the six centroids depicted in Figure 4.2a. Since for linear pressure fields $p(x, y) = p_0 + ax + by$ we have equality in (4.49), the

approximation is exact for linear pressure fields. Solution of the system (4.50)–(4.53) leads, with $\mathbf{x}_j = (x_j, y_j)$, $\mathbf{N}_i = (N_x, N_y)$ and auxiliary coefficients given by

$$a_{11} = x_2 - x_1; \quad a_{12} = y_2 - y_1; \quad (4.54a)$$

$$a_{21} = x_3 - x_6 + x_4 - x_5; \quad a_{22} = y_3 - y_6 + y_4 - y_5, \quad (4.54b)$$

to the following expressions for the gradient coefficients:

$$\gamma_1 = -\gamma_2 = \frac{a_{21}N_y - a_{22}N_x}{a_{11}a_{22} - a_{12}a_{21}}, \quad (4.55a)$$

$$\gamma_3 = \gamma_4 = -\gamma_5 = -\gamma_6 = \frac{a_{11}N_y - a_{12}N_x}{a_{11}a_{22} - a_{12}a_{21}}. \quad (4.55b)$$

A special situation occurs when vector $(\mathbf{x}_2 - \mathbf{x}_1)$ is parallel to \mathbf{N}_i . The PI method then reduces to the simple difference equation:

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_2 - p_1}{|\mathbf{x}_2 - \mathbf{x}_1|}. \quad (4.56)$$

When one or more centroids in the stencil are absent, as occurs in the vicinity of boundaries, the paths in (4.53) are truncated such as to use only existing centroids.

Contour integral formulation

The contour integral (CI) formulation to compute the pressure gradient is based on the identity:

$$\int_{CV} \nabla p \cdot \mathbf{N}_i \, d\mathbf{x} = \mathbf{N}_i \cdot \int_{CV} \nabla p \, d\mathbf{x} = \mathbf{N}_i \cdot \oint_{\partial CV} p \mathbf{n} \, d\Gamma, \quad (4.57)$$

where \mathbf{n} is the outward unit normal on the boundary of the CV. Hence, the pressure gradient at face i follows from (assuming that the CV consists of the two adjacent triangles):

$$(\nabla p \cdot \mathbf{N})_i = \frac{1}{\Omega_i} \mathbf{N}_i \cdot \oint_{\partial CV} p \mathbf{n} \, d\Gamma \approx \frac{1}{\Omega_i} \sum_{e(i)} p_e \bar{l}_e (\mathbf{N}_e \cdot \mathbf{N}_i). \quad (4.58)$$

In this expression, the sum runs over the four CV faces, and $\bar{l}_e = l_e (\mathbf{n}_e \cdot \mathbf{N}_e)$. The pressure at face e follows from a weighted averaging of the two neighboring pressure values. For example, the pressure at face k is computed from:

$$p_k = \frac{\Omega_3}{\Omega_1 + \Omega_3} p_1 + \frac{\Omega_1}{\Omega_1 + \Omega_3} p_3, \quad (4.59)$$

and consequently the stencil consists of the six cells drawn in Figure 4.2a. Note that the CI formulation does not, unlike the PI formulation, reduce to (4.56) in the case that $(\mathbf{x}_2 - \mathbf{x}_1)$ is parallel to \mathbf{N}_i . If all cell areas are the same, then one can show that the CI method yields a pressure gradient that remains unaffected, i.e. $(\nabla p \cdot \mathbf{N}_i) = (\nabla p' \cdot \mathbf{N}_i)$, when pressure fields p and p' are related as follows:

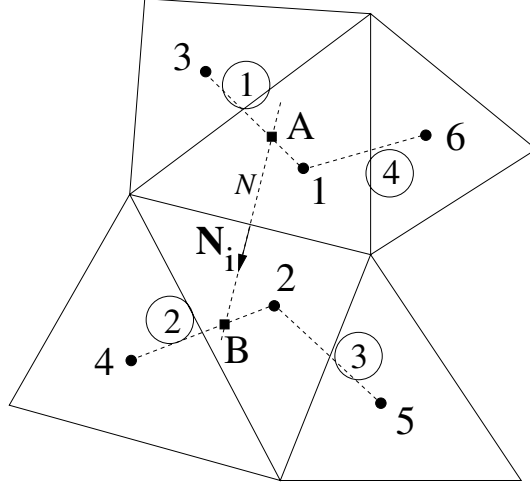


Figure 4.7: Auxiliary point method. Line N represents the line parallel to \mathbf{N}_i and passing through the midpoint of face i . Points A and B are the intersections between line N and lines connecting the centroids of the triangles.

$p'_c = p_c + \delta p$ for $c \in \{1, 4, 5\}$ and $p'_c = p_c - \delta p$ for $c \in \{2, 3, 6\}$. This can lead, as will be observed in Section 5.1.3, in incompressible flows to spurious pressure modes similar to odd-even decoupling. At boundary faces, either the pressure is given or a one-sided approximation is used.

Auxiliary point method

The auxiliary point (AP) method is a generalization of a method proposed in Section 8.6 of [30] for structured collocated grids. It is also discussed in [78]. This method is based on finding a finite difference approximation for the pressure gradient using the line normal to face i that passes through the midpoint of face i . In two auxiliary points located at this line the pressure is obtained by means of linear interpolation, after which simple differencing yields an approximation for the pressure gradient. The line N , see Figure 4.7, is parallel to \mathbf{N}_i and passes through \mathbf{x}_i , the midpoint of face i . This line is parameterized by

$$\mathbf{l}_N = \mathbf{x}_i + \lambda \mathbf{N}_i, \quad \lambda \in \mathbb{R}. \quad (4.60)$$

Lines 1, 2, 3 and 4, indicated in the figure, connect centroids and are parameterized by

$$\mathbf{l}_{p,q} = \mathbf{x}_p + \sigma(\mathbf{x}_q - \mathbf{x}_p), \quad \sigma \in \mathbb{R}, \quad (4.61)$$

where p and q refer to cell centroids. We have $\{p, q\} = \{1, 3\}$ for line 1, $\{p, q\} = \{2, 4\}$ for line 2, $\{p, q\} = \{3, 5\}$ for line 3 and $\{p, q\} = \{1, 6\}$ for line 4. Intersections of line N with the lines 1 to 4 are found by solving $\mathbf{l}_N = \mathbf{l}_{p,q}$, resulting in corresponding

values $\sigma_1, \dots, \sigma_4$ and $\lambda_1, \dots, \lambda_4$. With the situation as depicted in Figure 4.7, we will find that $0 < \sigma_1 < 1$, $0 < \sigma_2 < 1$, $\sigma_3 < 0$ and $\sigma_4 < 0$. The relevant intersections are obviously the ones with lines 1 and 2. The coordinates of the auxiliary points A and B , i.e. the intersections between line N and lines 1 and 2, follow from

$$\mathbf{x}_A = \mathbf{x}_1 + \sigma_1(\mathbf{x}_3 - \mathbf{x}_1), \quad \mathbf{x}_B = \mathbf{x}_2 + \sigma_2(\mathbf{x}_4 - \mathbf{x}_2). \quad (4.62)$$

The pressure in these points is approximated by means of linear interpolation:

$$p_A = p_1 + \sigma_1(p_3 - p_1), \quad p_B = p_2 + \sigma_2(p_4 - p_2). \quad (4.63)$$

The pressure gradient at face i is computed using the simple finite difference equation:

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_B - p_A}{|\mathbf{x}_B - \mathbf{x}_A|}. \quad (4.64)$$

Implementation of this approach, taking only the intersections with $\sigma \geq 0$ into account, is done in a simple and robust manner:

$$\begin{aligned} p_A &= p_1 + \max(\sigma_1, 0)[p_3 - p_1] + \max(\sigma_4, 0)[p_6 - p_1], \\ p_B &= p_2 + \max(\sigma_2, 0)[p_4 - p_2] + \max(\sigma_3, 0)[p_5 - p_2], \\ \mathbf{x}_A &= \mathbf{x}_1 + \max(\sigma_1, 0)[\mathbf{x}_3 - \mathbf{x}_1] + \max(\sigma_4, 0)[\mathbf{x}_6 - \mathbf{x}_1], \\ \mathbf{x}_B &= \mathbf{x}_2 + \max(\sigma_2, 0)[\mathbf{x}_4 - \mathbf{x}_2] + \max(\sigma_3, 0)[\mathbf{x}_5 - \mathbf{x}_2]. \end{aligned}$$

Note that, with extremely skewed triangles, the lines \mathbf{l}_N and $\mathbf{l}_{p,q}$ may become (close to) parallel, leading to inaccurate results or even breakdown of the algorithm. If this is the case, one has to use another gradient scheme or generate a smoother grid. Consider the situation in which cell 3 is not present, hence cell 1 is a boundary cell with boundary face j , see Figure 4.8. The location of \mathbf{x}_3 , positioned now at face j , is chosen such that the line through this point and \mathbf{x}_1 is normal to face j . The pressure p_3 in point \mathbf{x}_3 is either given or put equal to p_1 . The latter is based on $\partial p / \partial \mathbf{n} = 0$, which is usually a good approximation when the wall has a small curvature, see for example equation (19.2.26) in [48]. Other situations in the vicinity of boundaries are treated similarly.

Four quadrant method

The three schemes for computing the pressure gradient described above may result, as we will show in a moment, in results that sometimes even have an incorrect sign. Let \mathbf{N}_i point from cell 1 to 2, see Figure 4.2a. Suppose p_4 is increased by an amount δp . Then one would like $(\nabla p \cdot \mathbf{N})_i$ not to decrease. According to (4.46), this requires $\gamma_4 \geq 0$. Similarly, we get the criteria $\gamma_2 \geq 0$, $\gamma_5 \geq 0$, and $\gamma_1 \leq 0$, $\gamma_3 \leq 0$ and $\gamma_6 \leq 0$. More generally speaking, the gradient coefficient γ_c of a centroid located at \mathbf{x}_c must be larger than (or equal to) zero when $(\mathbf{x}_c - \mathbf{x}_i) \cdot \mathbf{N}_i > 0$, and smaller than (or equal to) zero when $(\mathbf{x}_c - \mathbf{x}_i) \cdot \mathbf{N}_i < 0$, where \mathbf{x}_i indicates the location of the midpoint of face i . This is what we call the *sign-criterion*. This criterion is not satisfied by the

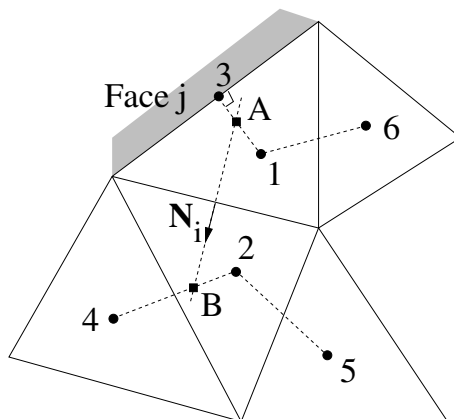


Figure 4.8: Treatment of boundaries in the auxiliary point method.

PI formulation, see relation (4.55), except for the special case in which $(\mathbf{x}_2 - \mathbf{x}_1)$ is parallel to \mathbf{N}_i , because then $\gamma_3 = \gamma_4 = \gamma_5 = \gamma_6 = 0$. The AP method meets the mentioned requirement as long as the values for σ are not larger than one. The CI formulation satisfies this criterion as long as the angles in the triangles are smaller than or equal to 90° .

Now we will introduce a method that uses a stencil consisting of four points that, away from the boundaries, always meets the sign-criterion. At face i a local $(\mathbf{N}_i, \mathbf{t}_i)$ -coordinate system is defined, see Figure 4.9, that divides the domain into four quadrants. With \mathbf{x} the coordinates of a cell centroid relative to \mathbf{x}_i , its quadrant is determined as follows: it is located in the first quadrant when $\mathbf{x} \cdot \mathbf{t}_i > 0$ and $\mathbf{x} \cdot \mathbf{N}_i \leq 0$; in the second quadrant when $\mathbf{x} \cdot \mathbf{t}_i \leq 0$ and $\mathbf{x} \cdot \mathbf{N}_i < 0$; in the third quadrant when $\mathbf{x} \cdot \mathbf{t}_i < 0$ and $\mathbf{x} \cdot \mathbf{N}_i \geq 0$, and it is located in the fourth quadrant when $\mathbf{x} \cdot \mathbf{t}_i \geq 0$ and $\mathbf{x} \cdot \mathbf{N}_i > 0$. In each of the quadrants the centroid being closest to \mathbf{x}_i is selected to be part of the stencil. The next question is how to compute the pressure gradient at face i , given a four-point stencil? We will show that the path integral formulation and the contour integral formulation applied to a four-point stencil lead to identical expressions for the gradient coefficients. In addition, it is shown that the resulting scheme, which we will call the four quadrant (FQ) method, satisfies the sign-criterion. Application of the path integral method to the four-point stencil, see equation (4.49) and Figure 4.9, gives:

$$p_c - p_a = (\nabla p)_i \cdot (\mathbf{x}_c - \mathbf{x}_a), \quad p_d - p_b = (\nabla p)_i \cdot (\mathbf{x}_d - \mathbf{x}_b). \quad (4.65)$$

Solution of this system leads, with help of the identity

$$(\mathbf{x}_c - \mathbf{x}_a) \times (\mathbf{x}_d - \mathbf{x}_b) = 2\Omega, \quad (4.66)$$

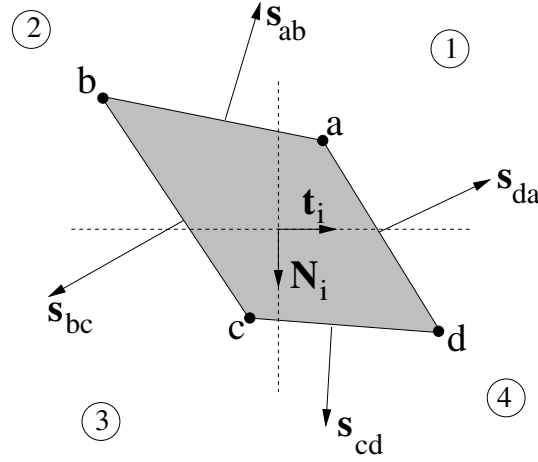


Figure 4.9: Four quadrant method. Points a , b , c and d are in the first, second, third and fourth quadrant, respectively.

where Ω represents the area of the shaded region in Figure 4.9, to

$$(\nabla p)_i = \frac{1}{2\Omega} \begin{bmatrix} (y_d - y_b)(p_c - p_a) - (y_c - y_a)(p_d - p_b) \\ -(x_d - x_b)(p_c - p_a) + (x_c - x_a)(p_d - p_b) \end{bmatrix}. \quad (4.67)$$

Now we consider the contour integral formulation for a four-point stencil. Let \mathbf{s}_{jo} , with $jo \in \{ab, bc, cd, da\}$, be the outward normal at line jo , which is obtained by rotating vector $(\mathbf{x}_o - \mathbf{x}_j)$ over 90° in the clockwise direction. Hence, with $(\mathbf{x}_o - \mathbf{x}_j) = (x_o - x_j, y_o - y_j) = (x_{jo}, y_{jo})$ we get $\mathbf{s}_{jo} = (y_{jo}, -x_{jo})$. Note that \mathbf{s}_{jo} is not normalized. It is trivial to prove that $\mathbf{s}_{jo} + \mathbf{s}_{oq} = \mathbf{s}_{jq}$ and $\mathbf{s}_{jo} = -\mathbf{s}_{oj}$. The pressure gradient follows from integration over the shaded region in Figure 4.9:

$$\begin{aligned} (\nabla p)_i &= \frac{1}{\Omega} \oint p \mathbf{n} d\Gamma = \\ &= \frac{1}{\Omega} \left[\frac{1}{2}(p_a + p_b)\mathbf{s}_{ab} + \frac{1}{2}(p_b + p_c)\mathbf{s}_{bc} + \frac{1}{2}(p_c + p_d)\mathbf{s}_{cd} + \frac{1}{2}(p_d + p_a)\mathbf{s}_{da} \right] = \\ &= \frac{1}{2\Omega} \begin{bmatrix} (y_d - y_b)(p_c - p_a) - (y_c - y_a)(p_d - p_b) \\ -(x_d - x_b)(p_c - p_a) + (x_c - x_a)(p_d - p_b) \end{bmatrix}, \end{aligned} \quad (4.68)$$

which is the same result as obtained with the path integral method. When two cell centers, say b and d , are located on the normal line through \mathbf{x}_i , they are considered to be in quadrant 2 and 4, respectively. In this case (4.70) reduces to

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_d - p_b}{|\mathbf{x}_d - \mathbf{x}_b|}. \quad (4.69)$$

In the situation that one quadrant is empty, which occurs in the vicinity of boundaries, we use a three-point stencil. Also for the three-point stencil, the path integral

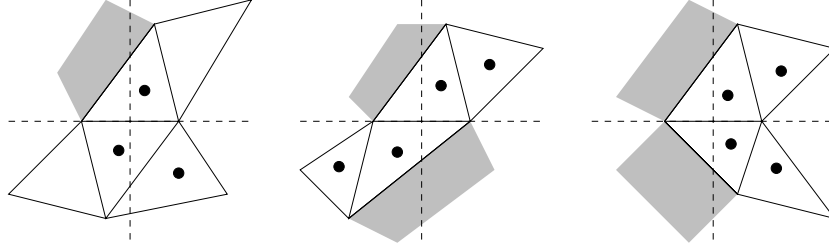


Figure 4.10: The pressure stencil for the four quadrant method in the vicinity of boundaries is indicated by bullets (\bullet).

and contour integral formulation yield identical results, but the sign-criterion is not always satisfied. In the situation that two quadrants are empty, we use in each of the non-empty quadrants one additional point, namely the one second closest to face i . This is summarized in Figure 4.10.

In order to see whether the sign-criterion is satisfied, consider again the contour integral formulation, and use a projection on \mathbf{N}_i :

$$\begin{aligned}
 (\nabla p \cdot \mathbf{N})_i &= \frac{1}{\Omega} \mathbf{N}_i \cdot \oint p \mathbf{n} d\Gamma = \\
 &= \frac{1}{\Omega} \left[\frac{1}{2} (p_a + p_b) \mathbf{s}_{ab} + \frac{1}{2} (p_b + p_c) \mathbf{s}_{bc} + \frac{1}{2} (p_c + p_d) \mathbf{s}_{cd} + \frac{1}{2} (p_d + p_a) \mathbf{s}_{da} \right] \cdot \mathbf{N}_i = \\
 &= \frac{1}{2\Omega} [p_a (\mathbf{s}_{db} \cdot \mathbf{N}_i) + p_b (\mathbf{s}_{ac} \cdot \mathbf{N}_i) + p_c (\mathbf{s}_{bd} \cdot \mathbf{N}_i) + p_d (\mathbf{s}_{ca} \cdot \mathbf{N}_i)]. \quad (4.70)
 \end{aligned}$$

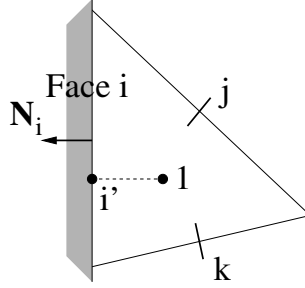
It is easy to see that the gradient coefficients all have the correct sign (in view of the assumed direction of \mathbf{N}_i , consider negative increments of p_a and p_b , and positive increments of p_c and p_d), and that this is due to their location in the different quadrants. Furthermore, the gradient coefficients satisfy $\gamma_c = -\gamma_a$ and $\gamma_d = -\gamma_b$.

Evaluation of the pressure gradient at boundary faces

When the normal momentum component is not given, as is usually the case at outflow boundaries, the pressure gradient must be computed. We have implemented two ways, depending on the gradient scheme used, to approximate the normal pressure gradient there. With $p_{i'}$ the given pressure at $\mathbf{x}_{i'}$, see Figure 4.11, the normal pressure gradient at the outflow boundary face i is computed from:

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_{i'} - p_1}{|\mathbf{x}_{i'} - \mathbf{x}_1|} \quad (4.71)$$

when the PI method, the AP method or the FQ method is employed. At a supersonic outflow boundary, $p_{i'}$ is not given, and we put it equal to p_1 , which is equivalent to putting the normal pressure gradient equal to zero. If the CI method is used,

Figure 4.11: Boundary face i and boundary cell 1.

another strategy is pursued. Integration over cell 1 leads to:

$$(\nabla p \cdot \mathbf{N})_i = \frac{1}{\Omega_1} \sum_{e(1)} p_e \bar{l}_e (\mathbf{N}_e \cdot \mathbf{N}_i), \quad (4.72)$$

where the summation runs over the faces in $e(1) \in \{i, j, k\}$. The pressure p_i is given, and pressures p_e , $e \in \{j, k\}$, follow from a weighted interpolation, cf. equation (4.59), if e is an internal face. If e is a boundary face, p_e is put equal to the prescribed pressure, or, if p_e is not given, we put it equal to p_1 .

4.2.6 Discretization of the viscous term

Integration of the viscous term over the two cells connected to face i yields:

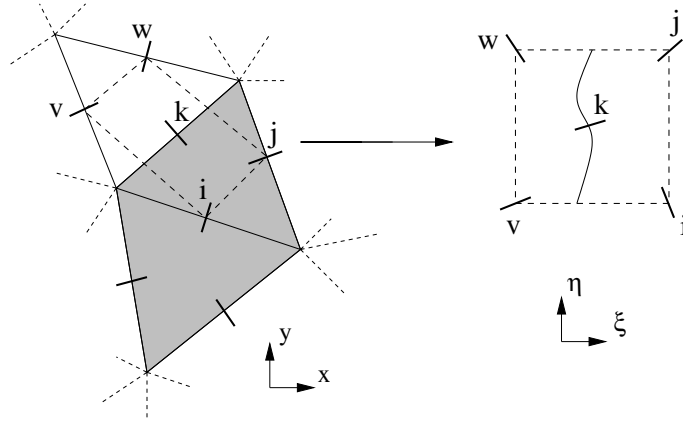
$$\int_{\text{CV}} N_i^\alpha \tau_{,\beta}^{\alpha\beta} d\mathbf{x} = \oint_{\partial\text{CV}} N_i^\alpha \tau^{\alpha\beta} n^\beta d\Gamma \approx N_i^\alpha \sum_{e(i)} \tau_e^{\alpha\beta} N_e^\beta \bar{l}_e, \quad (4.73)$$

where the summation runs over the control volume faces. We repeat the expression for the deviatoric stress tensor:

$$\tau^{\alpha\beta} = \mu \left(u_{,\beta}^\alpha + u_{,\alpha}^\beta - \frac{2}{3} \Delta \delta^{\alpha\beta} \right), \quad \Delta = u_{,\alpha}^\alpha. \quad (4.74)$$

We have the normal velocities, obtained from the normal momentum components using (4.12), as unknowns at the faces. From (4.73) and (4.74) we conclude that the velocity gradient must be computed at the CV-faces. The velocity gradient at face k , see Figure 4.12, is obtained by taking the gradient of the bilinear interpolation of the velocity in the quadrilateral formed by the four surrounding velocity points, i.e. v , i , j and w . The gradient of this bilinear polynomial defines the gradient of the velocity vector. The bilinear polynomial is constructed in the usual finite element sense. To that end, the quadrilateral is mapped onto the unit square in the (ξ, η) -domain by means of an isoparametric transformation. The latter is defined by

$$\mathbf{x} = \sum_s \mathbf{x}_s \phi_s(\xi, \eta), \quad (4.75)$$

Figure 4.12: Mapping of the quadrilateral vjw to a unit square.

where summation runs over the faces in $s \in \{v, i, j, w\}$, and $\phi_v = (1 - \xi)(1 - \eta)$, $\phi_i = \xi(1 - \eta)$, $\phi_j = \xi\eta$ and $\phi_w = (1 - \xi)\eta$. The velocity is approximated in the same way, i.e.

$$\mathbf{u} = \sum_s \mathbf{u}_s \phi_s(\xi, \eta). \quad (4.76)$$

The transformed coordinates (ξ_k, η_k) of face k are found by solving the nonlinear set of equations (4.75) using the standard Newton-Raphson procedure. The gradient of the velocity components follows with help from equations (4.75) and (4.76):

$$u_{,\beta}^\alpha = \sum_s u_s^\alpha (\phi_s)_{,\beta} \quad (4.77)$$

where

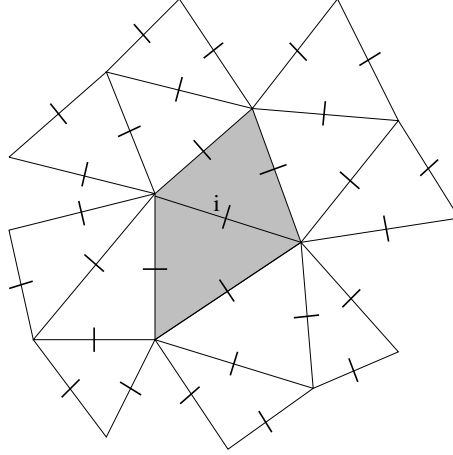
$$(\phi_s)_{,\beta} = \frac{\partial \phi_s}{\partial x^\beta} = \frac{\partial \phi_s}{\partial \xi} \frac{\partial \xi}{\partial x^\beta} + \frac{\partial \phi_s}{\partial \eta} \frac{\partial \eta}{\partial x^\beta}. \quad (4.78)$$

Hence, the gradient of the velocity vector is expressed in terms of the velocity vector in the four surrounding faces. Since only normal components of the velocity at the faces are given, the tangential components must be recovered, and this is done by means of the reconstruction procedure. The velocity vector at face i is decomposed into a normal and tangential component:

$$\mathbf{u}_i = u_i \mathbf{N}_i + \tilde{u}_i \mathbf{t}_i, \quad (4.79)$$

where $u_i = \mathbf{u}_i \cdot \mathbf{N}_i$ is the normal velocity component and $\tilde{u}_i = \mathbf{u}_i \cdot \mathbf{t}_i$ represents the tangential velocity component. Using the decomposition of the tangential vector \mathbf{t}_i at face i on adjacent normal vectors, see Figure 4.2a for notation:

$$\mathbf{t}_i = \xi_j \mathbf{N}_j + \xi_k \mathbf{N}_k, \quad \mathbf{t}_i = \xi_l \mathbf{N}_l + \xi_o \mathbf{N}_o, \quad (4.80)$$

Figure 4.13: Faces that form the viscous stencil for face i .

approximations for \tilde{u}_i in cells 1 and 2 follow from

$$\tilde{u}_{i,1} = \xi_j u_j + \xi_k u_k, \quad \tilde{u}_{i,2} = \xi_l u_l + \xi_o u_o. \quad (4.81)$$

Weighted averaging results in an approximation for the tangential velocity component at face i :

$$\tilde{u}_i = \frac{\Omega_2}{\Omega_1 + \Omega_2} \tilde{u}_{i,1} + \frac{\Omega_1}{\Omega_1 + \Omega_2} \tilde{u}_{i,2}. \quad (4.82)$$

In the end we arrive at a stencil containing at most (some faces may coincide) 29 faces, see Figure 4.13. This is a bit more than the 21 faces that are included in the viscous stencil on structured curvilinear grids.

Computation of the divergence of \mathbf{u} at face k is done as follows:

$$\Delta_k = \frac{1}{\Omega_1 + \Omega_3} \int \nabla \cdot \mathbf{u} \, d\mathbf{x} = \frac{1}{\Omega_1 + \Omega_3} \oint \mathbf{u} \cdot \mathbf{n} \, d\Gamma \approx \frac{1}{\Omega_1 + \Omega_3} \sum_{e \in (k)} u_e \bar{l}_e, \quad (4.83)$$

where integration is over the two cells neighboring face k , and summation runs over the faces in $e(k) \in \{i, j, v, w\}$. Note that we do not take the trace of the velocity gradient to compute Δ_k , since this results easily in $\Delta_k \neq 0$ in incompressible flows. On the other hand, expression (4.83) yields always $\Delta_k = 0$ for incompressible flows, see also equation (4.92), provided that the velocity in this expression is taken at the previous time level.

Boundary conditions

Due to the large stencil, discretization in the vicinity of the boundaries requires care. Suppose first that a next-neighboring face, for example face v in Figure 4.2a,

is located at the boundary. At this face, the velocity vector is required to obtain the velocity gradient at face k . If the tangential velocity \tilde{u}_v is not given, the reconstruction procedure is applied in a similar fashion as in equation (4.24). Next we consider the situation in which a CV-face is located at the boundary, for instance face k in Figure 4.3. The following boundary conditions are frequently encountered for viscous flow: (i) \mathbf{u}_k is given; (ii) u_k and σ_k^{nt} are given; (iii) \tilde{u}_k and σ_k^{nn} are given, or (iv) σ_k^{nn} and σ_k^{nt} are given. Here u_k and \tilde{u}_k are the normal and tangential velocity at face k , respectively, and

$$\sigma_k^{nn} = N_k^\alpha \sigma^{\alpha\beta} N_k^\beta, \quad \sigma_k^{nt} = N_k^\alpha \sigma^{\alpha\beta} t_k^\beta \quad (4.84)$$

represent the normal and tangential stress component at face k , with $\sigma^{\alpha\beta}$ is defined in (2.4). Since

$$\mathbf{N}_i = \cos \phi \mathbf{N}_k + \sin \phi \mathbf{t}_k, \quad (4.85)$$

where ϕ refers to the angle between the vectors \mathbf{N}_i and \mathbf{N}_k , and

$$\cos \phi = \mathbf{N}_i \cdot \mathbf{N}_k, \quad \sin \phi = \mathbf{N}_i \cdot \mathbf{t}_k, \quad (4.86)$$

the following relation between the stress tensor projected on \mathbf{N}_i and its normal and tangential components at face k holds:

$$N_i^\alpha \sigma_k^{\alpha\beta} N_k^\beta = \cos \phi \sigma_k^{nn} + \sin \phi \sigma_k^{nt}. \quad (4.87)$$

The boundary conditions and, if necessary, linear interpolation in the boundary cell based on the velocity vector in its three faces are used in order to arrive at a suitable discretization. At boundary face k , (4.83) is replaced by:

$$\Delta_k \approx \frac{1}{\Omega_1} \sum_{e(1)} u_e \bar{l}_e, \quad (4.88)$$

with summation over the three faces of cell 1.

Now consider the momentum equation for boundary face k . When the normal velocity at face k is given (boundary condition types (i) and (ii)), the discretized momentum equation at this face is trivial. Also the situation in which the normal velocity is not given is trivial, because the only required component of the stress tensor is σ_k^{nn} , which is given.

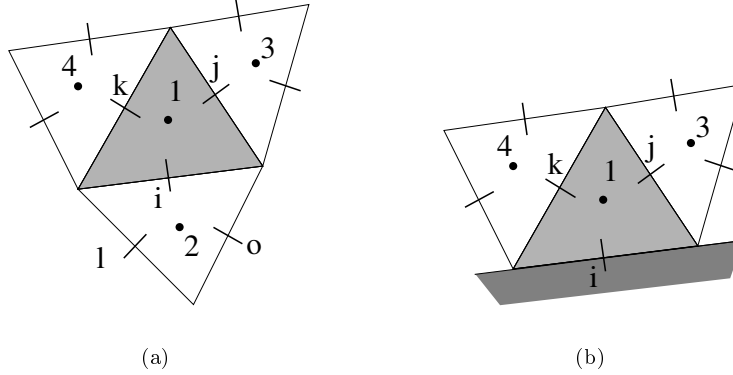


Figure 4.14: The CV for the continuity and energy equation is shaded, with cell 1 an internal cell (a) or a boundary cell (b).

4.3 Discretization of the continuity equation

The triangles serve as CV for the continuity equation. Integration of the continuity equation (2.1) over triangle T_1 , see Figure 4.14, is done as follows:

$$\begin{aligned} \int_{T_1} \frac{\partial \rho}{\partial t} d\mathbf{x} + \int_{T_1} \nabla \cdot (\rho \mathbf{u}) d\mathbf{x} &= \frac{d}{dt} \int_{T_1} \rho d\mathbf{x} + \oint_{\partial T_1} \rho (\mathbf{u} \cdot \mathbf{n}) d\Gamma \approx \\ &\approx \Omega_1 \frac{\rho_1^{n+1} - \rho_1^n}{\Delta t} + \sum_{e(1)} \rho_e u_e \bar{l}_e = 0, \end{aligned} \quad (4.89)$$

with \mathbf{n} the outward unit normal at the boundary ∂T_1 of cell 1. Superscripts n and $n+1$ refer to the time level and Δt is the time step. The area of T_1 is denoted by Ω_1 , summation takes place over the three faces $e(1) \in \{i, j, k\}$ of cell 1, the normal velocity at face e is given by $u_e = (\mathbf{u}_e \cdot \mathbf{N}_e)$ and \bar{l}_e is defined in (4.8). Note that we do not insert $\rho_e u_e = m_e$, with m_e evaluated from the momentum equation. The reason is that we consider the continuity equation as a convection equation for the density and that we want to be able to use an upwind bias in order to introduce a measure of irreversibility, which is beneficial for satisfying the entropy condition, see also [9, 37, 69]. The normal velocity components u_e are obtained by means of (4.12). In the fully compressible flow approach, we use $u_e = m_e^{n+1} / \rho_e^n$, hence with the momentum at the new time level and the density at the previous one. We think this enhances stability with respect to taking $u_e = m_e^n / \rho_e^n$. In the Mach-uniform approach, the velocity is taken at the previous time level: $u_e = m_e^n / \rho_e^n$. The reason to do so is that in (nearly) incompressible flows this results in (nearly) zero density variations, since the discrete velocity at each time level is (nearly) divergence free; this obviously does not hold in general for $u_e = m_e^{n+1} / \rho_e^n$.

Discretization of the convection term

The first order upwind approximation for the density at face i is

$$\rho_i = \begin{cases} \rho_1 & \text{if } u_i \bar{l}_i > 0; \\ \rho_2 & \text{if } u_i \bar{l}_i < 0. \end{cases} \quad (4.90)$$

The central scheme is given by

$$\rho_i = \frac{1}{2}(\rho_1 + \rho_2). \quad (4.91)$$

Remark that we use a factor $1/2$ and not, for example, a weighted averaging. By doing this the symmetry of the underlying operator is preserved. This is discussed in more detail in Appendix C. At boundaries, as shown for instance in Figure 4.14b, one-sided differences are taken, i.e. ρ_i is put equal to ρ_1 when the normal momentum component m_i is not prescribed (e.g. at outflow boundaries). If m_i is given, we insert $u_i \rho_i = m_i$ into equation (4.89).

Continuity equation in the incompressible flow case

Discretization of the continuity equation (2.19) for incompressible flow leads to:

$$\int_{T_i} \nabla \cdot \mathbf{u} \, d\mathbf{x} = \oint_{\partial T_i} (\mathbf{u} \cdot \mathbf{n}) \, d\Gamma = \sum_{e(1)} u_e \bar{l}_e = 0, \quad (4.92)$$

which specifies the operator D in (3.7). Note that, given the normal velocities at two triangle faces, expression (4.92) determines the third one. With u_j and u_k given, (4.92) results in:

$$u_i = -(u_j \bar{l}_j + u_k \bar{l}_k) / \bar{l}_i. \quad (4.93)$$

One may wonder whether the reconstruction procedure discussed in Section 4.2.3 yields the same value for this normal velocity component. We will show that this is indeed the case. Making use of

$$l_i \mathbf{n}_i + l_j \mathbf{n}_j + l_k \mathbf{n}_k = \bar{l}_i \mathbf{N}_i + \bar{l}_j \mathbf{N}_j + \bar{l}_k \mathbf{N}_k = \mathbf{0}, \quad (4.94)$$

which is obtained with help of (4.1) and (4.8), we arrive with $\mathbf{V}_i = \mathbf{N}_i$ in (4.9) at:

$$\chi_j^i = -\bar{l}_j / \bar{l}_i, \quad \chi_k^i = -\bar{l}_k / \bar{l}_i. \quad (4.95)$$

Inserting this in (4.11) gives (4.93). For stationary compressible flows, the continuity equation reduces to $\nabla \cdot \mathbf{m} = 0$, which is of course similar to (4.92). In conclusion, the reconstruction procedure yields, for incompressible flows and stationary compressible flows, a normal velocity (momentum) that satisfies the discrete continuity equation.

4.4 Discretization of the energy equation

In the Mach-uniform approach, the energy equation is replaced by a pressure-correction equation; see Section 4.5 for a discussion on its discretization. In the

fully compressible flow (FC) approach, see Section 3.3, a formulation of the inviscid energy equation (2.9) is required that is suitable for updating the primary variables sequentially in the order (\mathbf{m}, ρ, Ψ) , with Ψ a suitably chosen energy variable. Possible formulations are easily obtained from manipulations with (2.11)–(2.15):

$$\frac{\partial}{\partial t} \left[\frac{1}{\gamma} \Psi + \frac{1}{2} \frac{\gamma-1}{\gamma} \rho(\mathbf{u} \cdot \mathbf{u}) \right] + \nabla \cdot (\Psi \mathbf{u}) = 0, \quad \Psi = \rho H; \quad (4.96a)$$

$$\frac{\partial \Psi}{\partial t} + \nabla \cdot \left[\mathbf{u} \left\{ \gamma \Psi - \frac{1}{2} (\gamma-1) \rho(\mathbf{u} \cdot \mathbf{u}) \right\} \right] = 0, \quad \Psi = \rho E; \quad (4.96b)$$

$$\frac{\partial}{\partial t} \left[\frac{1}{\gamma} \rho \Psi + \frac{1}{2} \frac{\gamma-1}{\gamma} \rho(\mathbf{u} \cdot \mathbf{u}) \right] + \nabla \cdot (\Psi \mathbf{m}) = 0, \quad \Psi = H; \quad (4.96c)$$

$$\frac{\partial}{\partial t} \left[\frac{\rho}{\gamma} \Psi + \frac{1}{2} \rho(\mathbf{u} \cdot \mathbf{u}) \right] + \nabla \cdot \left[\mathbf{m} \left\{ \Psi + \frac{1}{2} (\mathbf{u} \cdot \mathbf{u}) \right\} \right] = 0, \quad \Psi = h. \quad (4.96d)$$

These relations, although mathematically equivalent, lead to different discretizations and as a consequence one might perform better than the other. Because it is hard to see a priori which energy variable is to be preferred, numerical experiments will play a decisive role in this matter. They will lead, see Section 6.2.1, to the conclusion that it is best to select ρH as primary energy variable in the FC approach. The energy equation is integrated over each triangle. There are, from a discretizational point of view, four distinct terms in equations (4.96a)–(4.96d).

- The time derivative of Ψ is discretized as

$$\int_{T_1} \frac{d\Psi}{dt} d\mathbf{x} \approx \Omega_1 \frac{\Psi_1^{n+1} - \Psi_1^n}{\Delta t}, \quad (4.97)$$

where T_1 stands for a triangle which area equals Ω_1 , the superscripts refer to the time level and Δt is the time step.

- The time derivative of the kinetic energy, needed when $\Psi \in \{\rho H, H, h\}$, is evaluated as

$$\frac{d}{dt} \int_{T_1} (\rho \mathbf{u} \cdot \mathbf{u}) d\mathbf{x} \approx \Omega_1 \frac{(\rho \mathbf{u} \cdot \mathbf{u})_1^{n+1} - (\rho \mathbf{u} \cdot \mathbf{u})_1^n}{\Delta t}. \quad (4.98)$$

Approximation of the term $(\rho \mathbf{u} \cdot \mathbf{u})_1$ will be discussed in Section 4.6.

- The convection term is computed as follows:

$$\int_{T_1} \nabla \cdot (\Psi \mathbf{u}) d\mathbf{x} = \oint_{\partial T_1} \Psi (\mathbf{u} \cdot \mathbf{n}) d\Gamma \approx \sum_{e(1)} \Psi_e u_e \bar{l}_e, \quad (4.99)$$

with Ψ_e evaluated using the upwind or central scheme given for ρ_e in (4.90) or (4.91). The normal velocities at $e(1) \in \{i, j, k\}$, i.e. the faces of T_1 , follow from (4.12).

- When ρE or h is selected as primary variable, a term representing convection of kinetic energy is present. The following approximation is used for this term:

$$\int_{T_1} \nabla \cdot [\mathbf{u} \rho (\mathbf{u} \cdot \mathbf{u})] d\mathbf{x} = \oint_{\partial T_1} (\mathbf{u} \cdot \mathbf{n}) (\rho \mathbf{u} \cdot \mathbf{u}) d\Gamma \approx \sum_{e(1)} u_e (\rho \mathbf{u} \cdot \mathbf{u})_e \bar{l}_e. \quad (4.100)$$

Originating from $\nabla \cdot (\mathbf{u} \rho H)$, also for this term the same sort of interpolation (central or upwind) as for the convected energy needs to be applied. The terms $(\rho \mathbf{u} \cdot \mathbf{u})_e$ are evaluated using the upwind or central scheme given in (4.90) or (4.91). Approximation of $(\rho \mathbf{u} \cdot \mathbf{u})$ in the cell centers is discussed in Section 4.6.

4.5 Discretization of the Mach-uniform pressure-correction equation

In this section we discuss the discretization of the Mach-uniform pressure-correction equation (3.51). As stated in Section 3.2, we apply the discrete pressure-correction method, which means that first the spatial and temporal discretization are performed, and afterwards the pressure-correction is derived. By doing this, one avoids the need to define additional boundary conditions for the pressure, which is fortunate since no such conditions are given.

Discretization of the time derivative

Integration of the time derivative, i.e. the first line in equation (3.51), over cell 1 yields:

$$\Omega_1 M_r^2 \left\{ \frac{(\delta p)_1}{\Delta t} + \frac{1}{2}(\gamma - 1) \frac{[(\mathbf{m}_1^*)^2 - 2\Delta t (\mathbf{m}^* \cdot \nabla \delta p)_1] / \rho_1^{n+1} - (\mathbf{m}_1^n)^2 / \rho_1^n}{\Delta t} \right\}, \quad (4.101)$$

where Ω_1 is the area of cell 1. Quantities $(\mathbf{m}_1^*)^2$ and $(\mathbf{m}_1^n)^2$ are obtained using one of the methods described in Section 4.6.2. The vector quantity $(\nabla \delta p)_1$ is expressed in terms of normal pressure gradients $(\nabla \delta p \cdot \mathbf{N})_e$ at the three faces of cell 1. Note that only methods 1 and 4 of the ones discussed in Section 4.6.2 are appropriate for this, since method 2 leads to an undesirable enlargement of the pressure stencil and method 3 would yield $(\nabla \delta p)_1 \cdot (\nabla \delta p)_1$ which is not the quantity we need. By default, method 1 is used. Now let face i of cell 1 be located at the boundary. If the normal momentum component m_i is given, then there is no need to compute the pressure-correction term $(\nabla \delta p \cdot \mathbf{N})_i$ since it is zero (because we have put $m_i^* = m_i^{n+1}$). This is implemented by putting all gradient coefficients γ_j corresponding to face i , cf. equation (4.46), equal to zero. If m_i is not given, the term $(\nabla \delta p \cdot \mathbf{N})_i$ needs to be computed at the boundary face. Note that the ways to evaluate the pressure gradient at boundary faces, see Section 4.2.5, are suited for this, and that a given pressure difference $(\delta p)_i$ at the boundary has to be put in the right-hand side.

Discretization of the convection term

Integration of the convection term (the second line in (3.51)) over cell 1 and application of Gauss's theorem results in

$$\sum_{e(1)} u_e^* \bar{l}_e \left\{ 1 + \gamma M_r^2 [p_e^n + (\delta p)_e] + \frac{1}{2}(\gamma - 1) M_r^2 (\mathbf{m}_e^*)^2 / \rho_e^{n+1} \right\}, \quad (4.102)$$

with summation over the faces of cell 1. The normal component of the predictor of the velocity is computed using $u_i^* = m_i^* / \rho_{i,\text{av}}^{n+1}$, where $\rho_{i,\text{av}}$ follows from a weighted averaging like in (4.12). The kinetic energy $(\mathbf{m}^*)^2 / \rho^{n+1}$ is evaluated in each cell by means of one of the methods discussed in Section 4.6.2. Because the complete term between curly braces originates from the term ρH , each part in it needs to be evaluated in the same fashion, e.g. using upwind or central interpolations. The first order upwind and central schemes used for this purpose are identical to the ones applied for the continuity equation, see equations (4.90) and (4.91). Let face i of cell 1 be located at a boundary. The term

$$\frac{1}{2}(\gamma - 1) M_r^2 u_i^* \bar{l}_i (\mathbf{m}_1^*)^2 / \rho_1^{n+1}, \quad (4.103)$$

with u_i^* evaluated by means of one-sided differences for $\rho_{i,\text{av}}$, is put into the right-hand side. If both the enthalpy and the normal component of the momentum are given at this face, then

$$\begin{aligned} u_i^* \bar{l}_i \{ 1 + \gamma M_r^2 [p_i^n + (\delta p)_i] \} &= u_i^{n+1} \bar{l}_i \{ 1 + \gamma M_r^2 p_i^{n+1} \} = \\ &= u_i^{n+1} \bar{l}_i \rho_i^{n+1} h_i^{n+1} = m_i^{n+1} h_i^{n+1} \bar{l}_i, \end{aligned} \quad (4.104)$$

in which we used (3.38), is moved to the right-hand side. If a central scheme is applied and p_i is given, then we write

$$u_i^* \bar{l}_i \{ 1 + \gamma M_r^2 [p_i^n + (\delta p)_i] \} = u_i^* \bar{l}_i \left\{ 1 + \frac{1}{2} \gamma M_r^2 [p_1^n + (\delta p)_1 + p_i^{n+1}] \right\}. \quad (4.105)$$

One-sided differences of the form

$$u_i^* \bar{l}_i \{ 1 + \gamma M_r^2 [p_i^n + (\delta p)_i] \} = u_i^* \bar{l}_i \{ 1 + \gamma M_r^2 [p_1^n + (\delta p)_1] \} \quad (4.106)$$

are employed in all other situations (e.g. homogeneous Neumann conditions for the enthalpy and pressure at freeslip walls, or first order upwind and the pressure given).

Discretization of the Laplacian term

Integration of the last line in (3.51) over cell 1 leads to:

$$- \sum_{e(1)} \frac{\Delta t}{\rho_e^{n+1}} \bar{l}_e \left\{ 1 + \gamma M_r^2 p_e^n + \frac{1}{2}(\gamma - 1) M_r^2 (\mathbf{m}_e^*)^2 / \rho_e^{n+1} \right\} \sum_{c(e)} \gamma_c (\delta p)_c, \quad (4.107)$$

with $e(1)$ indicating the faces of cell 1, and $c(e)$ referring to the gradient stencil of face e , cf. equation (4.46). The quantities ρ_e , p_e and $(\mathbf{m}_e^*)^2 / \rho_e$ at faces e are

obtained using weighted averaging. If face i is located at the boundary, then the prescribed values or, if the required quantities are not given, one-sided differences are used to arrive at ρ_i , p_i and $(\mathbf{m}_i^*)^2/\rho_i$. At the faces at which the normal momentum component is given, the gradient coefficients γ_c are put to zero. At faces i at which the pressure is given, we have

$$(\nabla \delta p \cdot \mathbf{N})_i = \sum_{c(i)} \gamma_c (\delta p)_c = \gamma_i (\delta p)_i + \sum_{c(i) \neq i} \gamma_c (\delta p)_c, \quad (4.108)$$

with $(\delta p)_i = p_i^{n+1} - p_i^n$ the prescribed pressure difference at boundary face i , and summation over $c(i) \neq i$ implies summation over neighboring cells.

Incompressible flow pressure-correction equation

As we saw in Section 3.5.2, the Mach-uniform pressure-correction equation reduces to the standard incompressible pressure-correction equation for $M_r = 0$ and constant density. Inserting this in the discretization given above leads to:

$$\rho \sum_{e(1)} u_e^* \bar{t}_e - \Delta t \sum_{e(1)} \sum_{c(e)} \gamma_c (\delta p)_c = 0, \quad (4.109)$$

which is the standard discretization of the incompressible pressure-correction equation.

Discretization of the correction equation

Discretization of the correction equation (3.44) for a normal momentum component at face i is trivial:

$$m_i^{n+1} = (\mathbf{m}_i^{n+1} \cdot \mathbf{N}_i) = (\mathbf{m}_i^* \cdot \mathbf{N}_i) - \Delta t (\nabla \delta p \cdot \mathbf{N})_i = m_i^* - \Delta t \sum_{j(i)} \gamma_j (\delta p)_j. \quad (4.110)$$

Note that at boundary faces at which the normal momentum component is given, inserting this in the discretization of the momentum equation and putting the corresponding gradient coefficients to zero suffices.

4.6 Evaluation of the equation of state

In the type of systems studied here there are at most two independent state variables; all other state variables depend on these through equations of state. Discretization of the equation of state for the Mach-uniform approach and the fully compressible flow approach forms the topic of Sections 4.6.1 and 4.6.2, respectively.

4.6.1 Mach-uniform approach

In the Mach-uniform approach, evaluation of the equation of state (3.38) in cell 1 yields, with the density and pressure given, the enthalpy in this cell:

$$h_1 = \frac{1}{\rho_1} (1 + \gamma M_r^2 p_1). \quad (4.111)$$

Evaluation of this expression is trivial, since all present quantities are located in the cell centers.

4.6.2 Fully compressible flow approach

In the fully compressible flow approach, the equation of state is an equation for the pressure. The equation of state, yielding the pressure in cell 1, is given by:

$$p_1 = \frac{\gamma - 1}{\gamma} \left[\Psi_1 - \frac{1}{2}(\rho \mathbf{u} \cdot \mathbf{u})_1 \right]; \quad \Psi = \rho H, \quad (4.112a)$$

$$p_1 = (\gamma - 1) \left[\Psi_1 - \frac{1}{2}(\rho \mathbf{u} \cdot \mathbf{u})_1 \right]; \quad \Psi = \rho E, \quad (4.112b)$$

$$p_1 = \frac{\gamma - 1}{\gamma} \left[\rho_1 \Psi_1 - \frac{1}{2}(\rho \mathbf{u} \cdot \mathbf{u})_1 \right]; \quad \Psi = H, \quad (4.112c)$$

$$p_1 = \frac{\gamma - 1}{\gamma} \rho_1 \Psi_1; \quad \Psi = h, \quad (4.112d)$$

where Ψ refers to the primary energy variable. Computation of the kinetic energy term in the cell center, appearing in (4.112a)–(4.112c), and also in (4.98), (4.100), (4.101), (4.102) and (4.107), is not trivial, since the velocity vector is not located in the cell center. Rewriting the kinetic energy in terms of primary variables gives

$$\frac{1}{2}(\rho \mathbf{u} \cdot \mathbf{u})_1 = \frac{1}{2} \frac{(\mathbf{m} \cdot \mathbf{m})_1}{\rho_1}, \quad (4.113)$$

where the difficulty now lies in finding a suitable approximation for $(\mathbf{m} \cdot \mathbf{m})_1$. Several methods are possible, and we will describe four of them.

1. Computation of the kinetic energy using a least squares approach

Since we have three knowns, namely the normal momentum components at the three cell faces, and two unknowns, the Cartesian components of $\mathbf{m}_1 = (m_{x,1}, m_{y,1})$, the use of a least squares approach appears as an obvious choice. If the flow field is smooth, we have $m_e = \mathbf{m}_e \cdot \mathbf{N}_e \approx (\mathbf{m}_1 \cdot \mathbf{N}_e)$ for all $e(1) \in \{i, j, k\}$. We choose \mathbf{m}_1 such that the least squares functional

$$\mathcal{F}(\mathbf{m}_1) = \sum_{e(1)} [m_e - (\mathbf{m}_1 \cdot \mathbf{N}_e)]^2 \quad (4.114)$$

is minimal. The minimum of this functional is found there where

$$\begin{cases} \frac{\partial \mathcal{F}}{\partial m_{x,1}} = \sum_{e(1)} -2N_{x,e}(m_e - m_{x,1}N_{x,e} - m_{y,1}N_{y,e}) = 0, \\ \frac{\partial \mathcal{F}}{\partial m_{y,1}} = \sum_{e(1)} -2N_{y,e}(m_e - m_{x,1}N_{x,e} - m_{y,1}N_{y,e}) = 0. \end{cases} \quad (4.115)$$

This boils down to solving the system:

$$\begin{bmatrix} \sum N_{x,e}^2 & \sum N_{x,e}N_{y,e} \\ \sum N_{x,e}N_{y,e} & \sum N_{y,e}^2 \end{bmatrix} \begin{bmatrix} m_{x,1} \\ m_{y,1} \end{bmatrix} = \begin{bmatrix} \sum m_e N_{x,e} \\ \sum m_e N_{y,e} \end{bmatrix}, \quad (4.116)$$

with summations again over the three faces of cell 1. This method is discussed in [96] for grids consisting of quadrilaterals.

2. Computation of the kinetic energy using the reconstruction procedure

In a manner identical to (4.79)–(4.82), the momentum vector at the cell faces can be obtained. Linear interpolation is used to evaluate the momentum vector in the cell center, after which computation of $(\mathbf{m} \cdot \mathbf{m})_1$ is trivial.

3. Computation of the kinetic energy using Gauss's theorem

We begin with Gauss's divergence theorem for triangle T_1 and vector quantity \mathbf{f} :

$$\int_{T_1} \nabla \cdot \mathbf{f} \, d\mathbf{x} = \oint_{\partial T_1} \mathbf{f} \cdot \mathbf{n} \, d\Gamma = \sum_{e(1)} \mathbf{n}_e \cdot \int_{\partial T_e} \mathbf{f} \, d\Gamma, \quad (4.117)$$

where the summation runs over the faces of the triangle and \mathbf{n}_e is the outward unit normal at face e . We set $\mathbf{f} = (\mathbf{m} \cdot \mathbf{r})\mathbf{m}$, where $\mathbf{r} = (\mathbf{x} - \mathbf{x}_0)$ is the position vector with arbitrary origin \mathbf{x}_0 . Then Gauss's theorem gives:

$$\int_{T_1} \nabla \cdot (\mathbf{m} \cdot \mathbf{r})\mathbf{m} \, d\mathbf{x} = \sum_{e(1)} \mathbf{n}_e \cdot \int_{\partial T_e} (\mathbf{m} \cdot \mathbf{r})\mathbf{m} \, d\Gamma. \quad (4.118)$$

Writing the left-hand side of this expression in Cartesian tensor notation yields

$$\int_{T_1} (m^\alpha r^\alpha m^\beta)_{,\beta} \, d\mathbf{x} = \int_{T_1} m^\alpha m^\beta r_{,\beta}^\alpha \, d\mathbf{x} + \int_{T_1} r^\alpha (m^\alpha m^\beta)_{,\beta} \, d\mathbf{x}. \quad (4.119)$$

With $r_{,\beta}^\alpha = \delta^{\alpha\beta}$, where $\delta^{\alpha\beta}$ is the Kronecker delta, and assuming that \mathbf{m} is constant over T_1 , we arrive after evaluation of the integrals at:

$$\Omega_1 (\mathbf{m} \cdot \mathbf{m})_1 = \sum_{e(1)} m_e (\mathbf{m} \cdot \mathbf{r})_e \bar{l}_e. \quad (4.120)$$

If we put \mathbf{x}_0 in the circumcenter of cell 1, then we get for the position vector: $\mathbf{r}_e = h_{e1} \mathbf{n}_e$, with h_{e1} the distance between the circumcenter of cell 1 and the midpoint of face e . This leads to

$$(\mathbf{m} \cdot \mathbf{m})_1 = \frac{1}{\Omega_1} \sum_{e(1)} m_e^2 h_{e1} l_e, \quad (4.121)$$

which states that the cell kinetic energy can be approximated by summation over the cell faces of the 'kinetic energy' of the normal momentum components multiplied

by some weight. This method is introduced in Section 5.2 of [89].

4. Computation of the kinetic energy using the momentum vector in the cell

We set $\mathbf{f} = (\mathbf{a} \cdot \mathbf{r})\mathbf{m}$, where $\mathbf{r} = (\mathbf{x} - \mathbf{x}_0)$ and arbitrary \mathbf{x}_0 , while $\mathbf{a} \neq \mathbf{0}$ is an arbitrary constant vector. Gauss's theorem gives

$$\int_{T_1} \nabla \cdot (\mathbf{a} \cdot \mathbf{r})\mathbf{m} \, d\mathbf{x} = \sum_{e(1)} \mathbf{n}_e \cdot \int_{\partial T_e} (\mathbf{a} \cdot \mathbf{r})\mathbf{m} \, d\Gamma. \quad (4.122)$$

In Cartesian tensor notation, the left hand side of this equation can be written as

$$\int_{T_1} (a^\alpha r^\alpha m^\beta)_{,\beta} \, d\mathbf{x} = \int_{T_1} a^\alpha r_{,\beta}^\alpha m^\beta \, d\mathbf{x} + \int_{T_1} a^\alpha r^\alpha m_{,\beta}^\beta \, d\mathbf{x}, \quad (4.123)$$

since $a_{,\beta}^\alpha = 0$. With $r_{,\beta}^\alpha = \delta^{\alpha\beta}$ and since \mathbf{a} is an arbitrary vector,

$$\int_{T_1} \mathbf{m} \, d\mathbf{x} + \int_{T_1} \mathbf{r} (\nabla \cdot \mathbf{m}) \, d\mathbf{x} = \sum_{e(1)} \int_{\partial T_e} (\mathbf{m} \cdot \mathbf{n})\mathbf{r} \, d\Gamma. \quad (4.124)$$

Assuming that the momentum field is constant and putting \mathbf{x}_0 in the circumcenter of triangle 1, we arrive at

$$\mathbf{m}_1 = \frac{1}{\Omega_1} \sum_{e(1)} m_e \mathbf{N}_e h_{e1} \bar{\mathbf{l}}_e, \quad (4.125)$$

where h_{e1} represents the distance between the circumcenter of cell 1 and the midpoint of face e . With this, it is trivial to compute $(\mathbf{m}_1 \cdot \mathbf{m}_1)$. This method is described in Section 5.4 of [89].

Discussion

All four methods to obtain the kinetic energy are first order accurate. The second method uses 9 momentum points, whereas the other three restrict themselves to the three momentum points located at the faces of the considered cell. In Section 6.2.1 we will investigate numerically the accuracy of the first two methods, and it will turn out that method 1 performs best. Method 3 and 4 have not been tested yet, but since they are of the same order of accuracy, we do not expect major differences with the first two methods.

4.7 Postprocessing

For visualization purposes, the primary variables are interpolated to the grid vertices. All other variables are then easily obtained by means of algebraic relations.

Interpolation of scalar quantities to the grid vertices

Let Q_j indicate the value of scalar primary variable Q at the center of cell j . The value of Q at vertex V is computed using

$$Q_V = \frac{\sum_j w_j Q_j}{\sum_j w_j}, \quad (4.126)$$

where summations run over the cells connected to vertex V . For the weights w_j we choose the inverse of the distance between the cell center of j and vertex V , since this is second order accurate in the one-dimensional case. In the covolume method the same interpolation for postprocessing is used, see equation (18) in [39].

Interpolation of the momentum vector to the grid vertices

At each of the faces the tangential component of the momentum vector is obtained by means of reconstruction, see (4.79)–(4.82). For interpolation to the vertices expression (4.126) is employed, where the summations now run over the faces connected to vertex V , and w_j is the inverse of the distance between V and the midpoint of face j .

Chapter 5

Results for viscous incompressible flows

In this chapter numerical results for viscous incompressible flows, computed using the pressure-correction approach described in Section 3.2, will be presented. The Poiseuille flow, discussed in Section 5.1, is a useful test case because the analytical solution is known. This test case is employed to study the accuracy on grids that are chosen irregular deliberately, the behavior of the error under grid refinement, and the convergence behavior of the linear solver on stretched and nonstretched grids. In addition, the symmetry properties of the discrete viscous and Laplacian matrix are studied. Far less trivial test cases are the backward facing step, studied in Section 5.2, and the lid-driven cavity, which is the topic of Section 5.3. For several values of the Reynolds number results computed with our unstructured staggered scheme are compared to experimental and numerical results obtained by other authors.

5.1 Poiseuille flow

The problem definition of a Poiseuille flow is stated in Section 5.1.1. Numerical results obtained with different pressure gradient methods on deliberately made non-smooth grids are discussed in Section 5.1.2. In Section 5.1.3, some interesting issues, including accuracy and efficiency, concerning computations on stretched grids are given. The eigenvalues of the viscous and Laplacian matrices on various grids form the topic of Section 5.1.4, and some conclusions are collected in Section 5.1.5.

5.1.1 Problem definition

A steady laminar incompressible flow between two fixed infinite parallel walls is called a Poiseuille flow. The exact solution for a horizontal Poiseuille flow is given

by:

$$u(x, y) = \frac{G}{2\mu}(a^2 - y^2), \quad v(x, y) = 0, \quad p(x, y) = -Gx + p_0, \quad (5.1)$$

where u and v stand for the velocity in the x - and y -direction, respectively, $G > 0$ is the constant pressure gradient, the pressure p is fixed up to an additive constant p_0 and the horizontal walls are located at $y = \pm a$. The Reynolds number is defined by

$$Re = \frac{\rho u_r L_r}{\mu}, \quad (5.2)$$

where we choose $u_r = \max(u) = Ga^2/2\mu$ as reference velocity and $L_r = 2a$ as reference length. The drop in pressure over a domain with length equals GL_x .

In our computations we choose the following parameters: $a = 1/2$, $\rho = 1$, $u_r = 1$, $Re = 1$, $\mu = 1$ and $G = 8$. At the inlet ($x = 0$), the exact velocity is given and at the horizontal walls the no-slip condition is enforced. At the outflow, located at $x = L_x$, zero stress ($\sigma^{nn} = \sigma^{nt} = 0$) is prescribed. Consequently, $p(x, y) = -8x + 8L_x$ and $u(x, y) = 1 - 4y^2$ are the exact solution for the pressure and velocity respectively, where $0 \leq x \leq L_x$ and $-1/2 \leq y \leq 1/2$. As initial condition we take $u = v = 0$ and $p = \text{constant}$. We assume to have arrived at the steady flow solution when a relative accuracy of 10^{-4} is reached for all unknowns; choosing an even smaller value was found to make negligible difference. It made hardly any difference whether central or upwind interpolation was applied for the inertia term; this was to be expected, since convection is absent in the analytic solution.

5.1.2 Non-smooth grids

In this section, we compare the accuracy of various pressure gradient schemes for the Poiseuille flow on grids that are made non-smooth deliberately. In [78], a similar investigation is described for a structured collocated scheme. We recall the gradient schemes, described in full detail in Section 4.2.5: the path integral (PI) method, the contour integral (CI) method, the auxiliary point (AP) method and the four quadrant (FQ) method. We compare our results with finite element method (FEM) results on the same grid. For this purpose we have used a Taylor-Hood element in which both the pressure and the velocity basisfunctions, both located in the grid vertices, are linear and continuous. This is the classical mini-element, in which the velocity is extended with a bubble function which is zero at the boundaries of the element and 1 in the centroid. This bubble function is necessary in order to satisfy the Brezzi-Babuška (BB) condition, i.e. to avoid spurious pressure modes. Both the velocity and pressure are solved simultaneously.

Distorted grid 1

The distorted grid is shown in Figure 5.1a. Figures 5.1b–e illustrate the isobars obtained by the various gradient schemes. As we see, the results obtained with

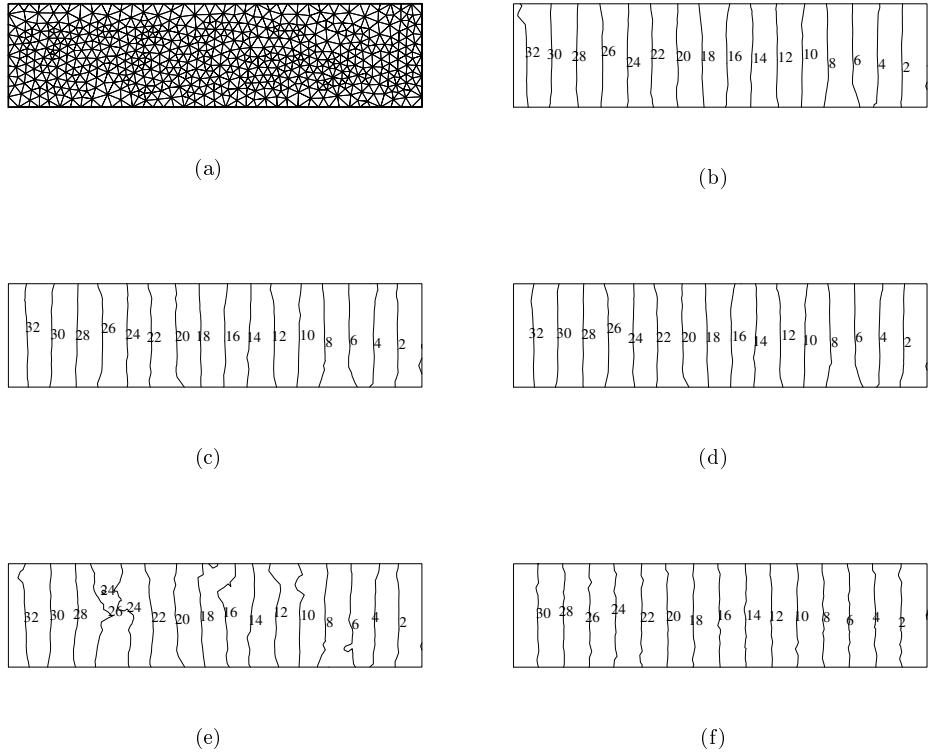
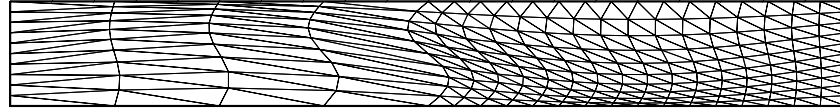


Figure 5.1: Distorted grid 1 (a), and isobars computed with the gradient schemes PI (b), CI (c), AP (d) and FQ (e). Results computed with the finite element method (f).

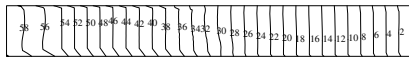
the FQ method are worse than the other ones, which show isobars that are quite straight and uniformly spaced. The CPU time for each of the computations is roughly the same. The finite element result, see Figure 5.1f, appears to be slightly more accurate than the finite volume results, since the isobars are more straight and a more accurate value for the pressure drop is obtained (should be: 32). Note that this may (partly) be caused by the interpolations needed for postprocessing in the finite volume method, see Section 4.7; in Taylor-Hood elements the pressure is already located at the vertex .

Distorted grid 2

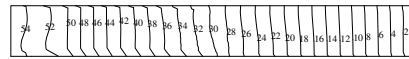
The distorted grid, containing an abrupt change in meshsize near $x = L_x/2$, and the computed isobars are depicted in Figure 5.2. The grid is too skewed for the current implementation of the FQ method, i.e. for at least one grid face the standard six



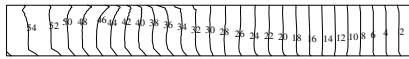
(a)



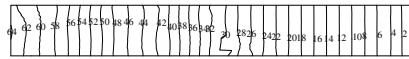
(b)



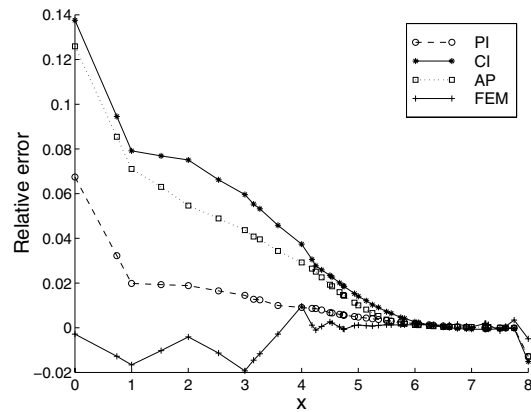
(c)



(d)



(e)



(f)

Figure 5.2: Distorted grid 2 (a), and isobars computed with the gradient schemes PI (b), CI (c) and AP (d). Results computed with the finite element method (e). The relative error in the pressure along the horizontal line through the middle of the domain (f).

point stencil did not cover all four surrounding quadrants. Since we do not expect, in view of the results on distorted grid 1, the FQ method to perform exceptionally well on grid 2, we did not bother to improve upon this scheme. The quality of the isobars stemming from the AP method is somewhat less than that of the other two schemes, of which the PI method seems to be the most accurate one. The pressure gradient as computed by means of the CI and AP method is, in the left part of the domain, somewhat too small. The interpolation required for the postprocessing causes the artefacts in the isobars near the inflow and solid walls in the left part of the domain. Also for this test case, there were no large differences between computation times. Again, the result obtained by means of the finite element method is slightly more accurate than the finite volume results.

5.1.3 Stretched grids

In order to resolve accurately and efficiently the thin boundary layers and wakes that appear in high-Reynolds number flows, stretched grids are a prerequisite in these regions. Proper boundary-layer resolution requires mesh spacings that are much smaller in the direction normal to the boundaries than in the streamwise direction, resulting in large cell aspect-ratios in these regions. We will study the behavior of the error and the linear solver for large aspect-ratio cells by means of computing solutions to the Poiseuille flow.

Grid definition

Examples of what is called a Courant grid, i.e. a grid that is formed by subdividing the rectangles of a structured Cartesian grid, are shown in Figure 5.3. The aspect-ratio of the triangles in these Courant grids is defined as $r = \Delta x / \Delta y$, with Δx and Δy the cell size in the x - and y -direction, respectively. We will restrict ourselves in this section to three types of grids: (i) type *R1* is characterized by $r = 1$ and $L_x = 4$; (ii) type *R8* is characterized by $r = 8$ and $L_x = 4$, and (iii) type *L8* is characterized by $r = 8$ and $L_x = 32$.

Information with respect to the pressure-correction matrix

The elements of the pressure-correction matrix on a Courant grid depend solely on the value of r and the gradient scheme employed. Numerical values for the matrix elements in the interior of the domain for grids with $r = 1$ and $r = 8$ are given in Figure 5.4. The nonsymmetric matrix resulting from the PI method is not an M-matrix, hence the resulting linear system might be awkward to solve. The CI method leads to a symmetric Laplacian matrix in which there is no coupling between triangles with different orientations in the domain interior; this spells accuracy troubles as we will see later on. The AP method and FQ method yield in the domain interior (but not near the boundaries) the same values for the matrix elements, resulting in a nonsymmetric M-matrix.

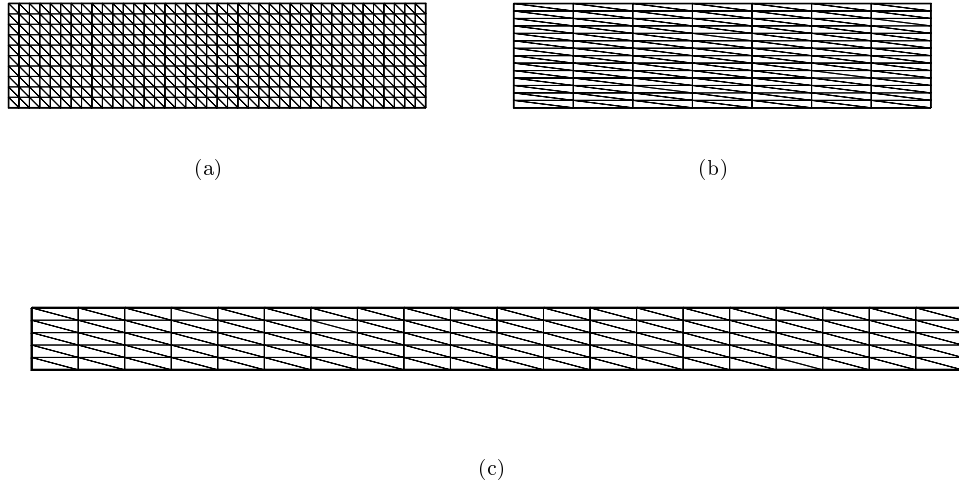


Figure 5.3: Grids with $r = 1$ and $r = 8$. These grids are representatives of types $R1$, $R8$ and $L8$, respectively.

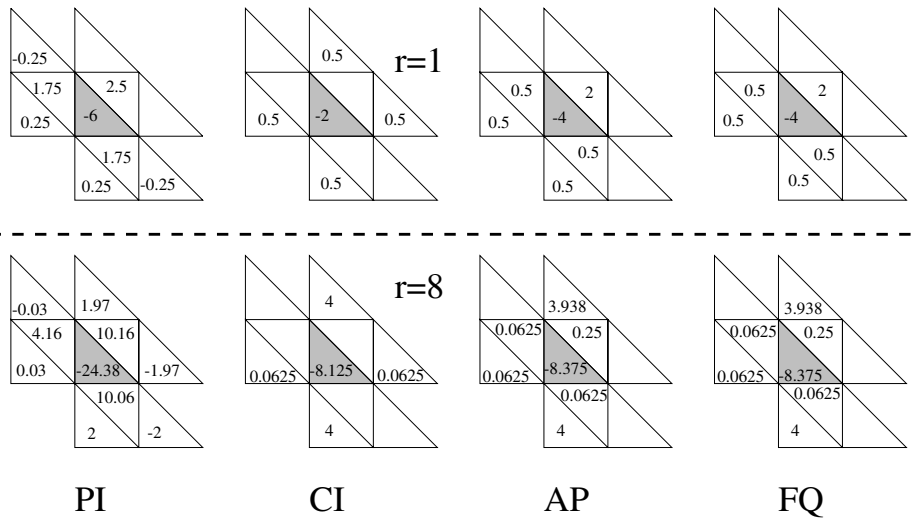


Figure 5.4: Pressure-correction matrix elements resulting from the various gradient schemes for Courant grids with $r = 1$ and $r = 8$. The diagonal element is shaded.

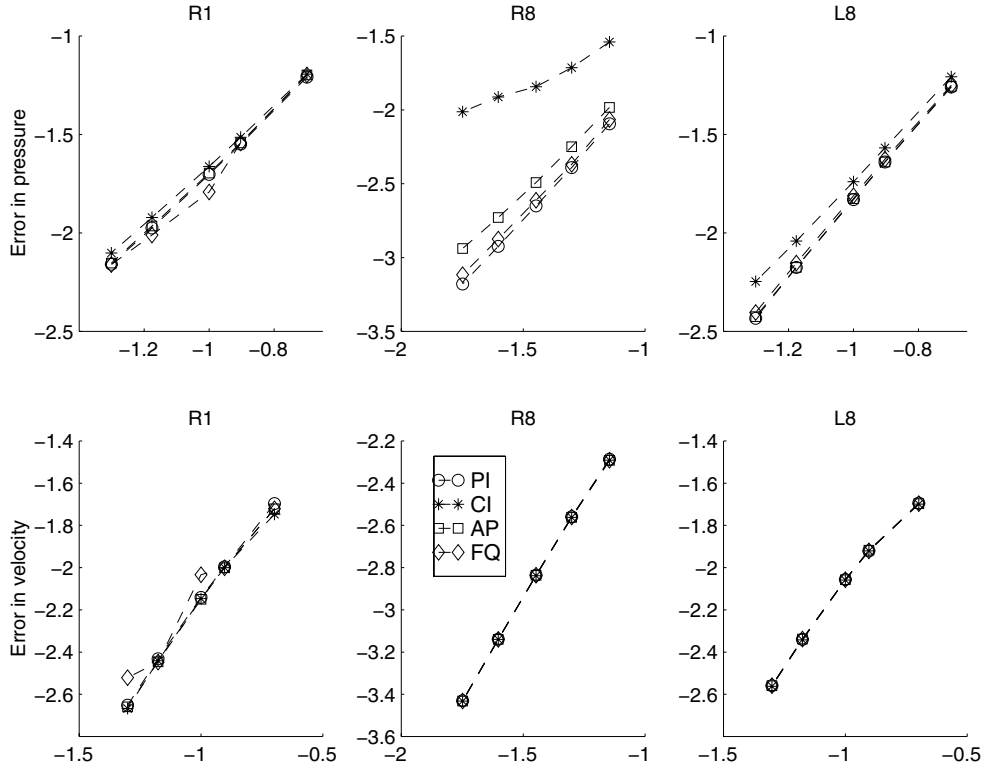


Figure 5.5: Logarithm of the maximum norm of the error in pressure (upper row) and velocity (lower row) at the several types of grid versus the logarithm of Δy .

Behavior of the error during grid refinement

A grid refinement study has been performed on the three types of grids described above. For the error we take the maximum of the difference between the numerical and (pointwise) exact value of the corresponding variable in each designated grid point (i.e. centroid for the pressure and face-midpoint for the normal velocity). The errors in the pressure and the velocity have been scaled with respect to the total pressure drop and u_r , respectively. From Figure 5.5 we conclude that the errors in the pressure and velocity behave as $\mathcal{O}(\Delta y^{1.7})$, independently of the gradient scheme (apart from the CI method, see below) or type of grid used. Dependence on Δy was to be expected, since the pressure behaves linearly with x while the profile of the velocity is parabolic in y . The reason for the power in Δy being 1.7 is probably due to a combination of the discretization, being $\mathcal{O}(\Delta y)$ on non-smooth grids, and the smoothness of the grids which usually tends to increase the order of accuracy. The CI method demonstrates an anomalous behavior of the pressure on grids of type $L8$ and $R8$, which is explained in the next item.

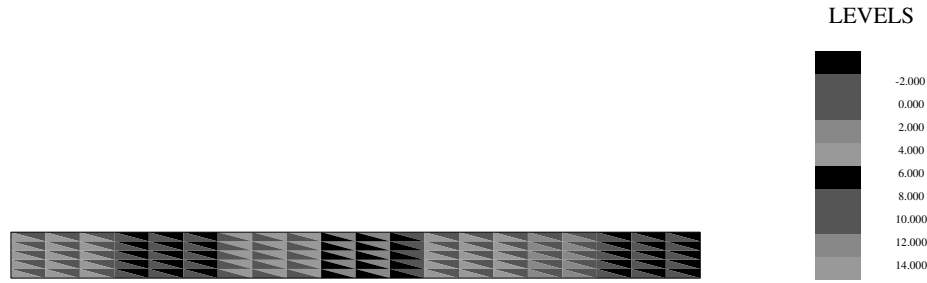


Figure 5.6: Odd-even distribution at a grid of type $L8$. Shown is the error, i.e. the exact minus computed value, of cell values of the pressure.

Odd-even decoupling for the CI method

The anomalous behavior of the error of the pressure as resulting from the CI method at grids of type $L8$ and $R8$ apparently does not reduce the accuracy of the velocity and seems not present at grids of type $R1$. It is not difficult to show that, in incompressible flows, the CI method does not suppress pressure oscillations on Courant grids between triangles with different orientations. This is an immediate consequence of the fact that all cells have the same area, see the description of the CI method in Section 4.2.5. In Figure 5.6, the spurious pressure distribution between triangles with different orientations is clearly visible, and this undesirable result has caused the anomalous behavior. Note that the error is rather large (up to 14 pressure units) compared to the total pressure drop ($8 \times 32 = 256$ pressure units). The only coupling between triangles with different orientations is present near solid walls, due to the truncated gradient stencil, and at the outflow boundaries, where the pressure is prescribed. It can be shown after a rather lengthy and tedious analysis that the strength of this coupling diminishes for increasing values of r , explaining why the spurious pressure modes are most prominent at grids of type $R8$.

Solution of the pressure-correction equation

Solution of the pressure-correction equation is computationally expensive, and this renders a study of the efficiency of different linear solvers worthwhile. Because the pressure-correction matrix is generally not symmetric, we cannot use the CG method and therefore we decided to study the performance of Bi-CGSTAB and GMRES, see also Section 3.6.1. Information concerning the start vector, stopping criterion and restarts can be found in the section just mentioned. From Figure 5.4 we deduce that some off-diagonal elements stemming from the PI method have the ‘wrong’ sign, preventing the matrix from being an M-matrix. In order to avoid a bad ILU preconditioning in such a situation, a so-called lumped ILU preconditioner can be used (C. Vuik, private communication). This type of preconditioning can be briefly described as follows: (i) define an auxiliary matrix \hat{A} in which the off-diagonal ele-

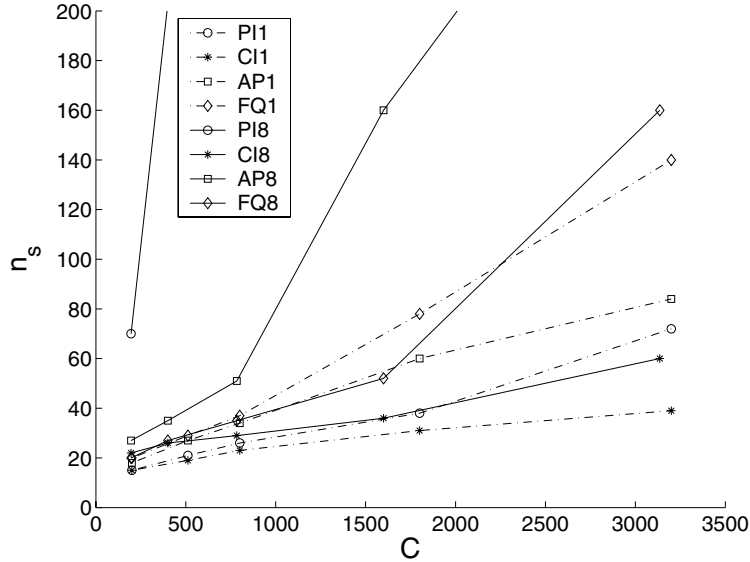


Figure 5.7: Average number of solver iterations for ILU preconditioned GMRES as function of the number of unknowns. The labels 1 and 8 refer to grids of type $R1$ and $R8$.

ments with the wrong sign are lumped in a carefully designed way to the diagonal, and (ii) ILU preconditioning is taken with respect to \hat{A} . This amounts up to, in total, four distinct solver strategies: ILU preconditioned GMRES or Bi-CGSTAB, and lumped ILU preconditioned GMRES or Bi-CGSTAB. The average number of solver iterations needed to solve one pressure-correction equation is $n_s = N_s/N_t$. Here, N_s stands for the total number of solver iterations in the time-stepping process, which consists of N_t time steps from initial to steady state. Although the pressure-correction matrix remains unaltered in the time-stepping process (it only depends on grid topology), the actual number of solver iterations in a time step may show large fluctuations around n_s , especially when n_s is in the order of several hundreds. This is caused by variations in the right-hand side.

The number of GMRES-iterations required to solve the pressure-correction equation depends strongly on the aspect-ratio r and the number of unknowns C , see Figure 5.7. We see that grid stretching leads to an enormous deterioration of the performance of ILU preconditioned GMRES applied to matrices stemming from the AP method and, especially, the PI method. This is disadvantageous, since in Section 5.1.2 we found that the PI method is very accurate. In Figure 5.8 we observe that using a lumped ILU preconditioning results in a major improvement with respect to using a standard (not lumped) ILU preconditioning for matrices stemming

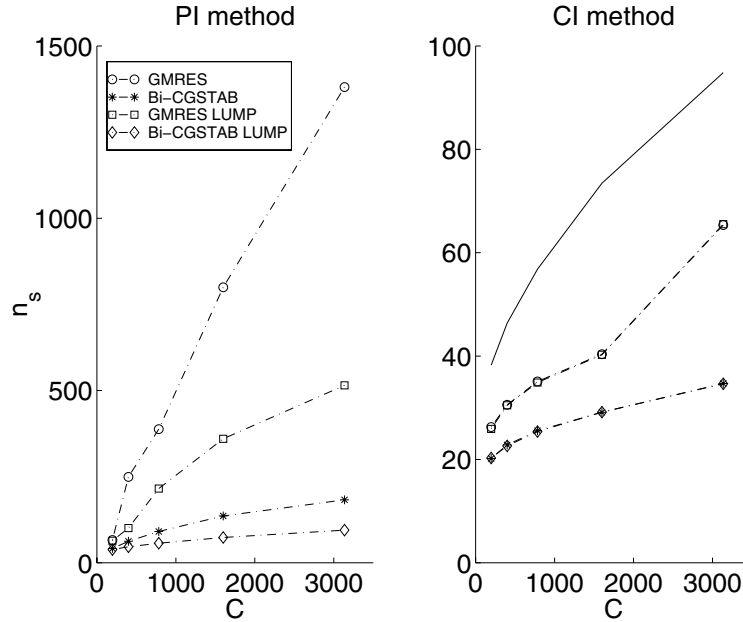


Figure 5.8: Number of solver iterations n_s for GMRES and Bi-CGSTAB, with ILU preconditioner and lumped ILU preconditioner, at grids of type $R8$. The pressure gradient is computed with the PI method (left figure) and the CI method (right figure). The continuous line in the right figure corresponds to the Bi-CGSTAB LUMP line in the left figure.

from the PI method. On the other hand, as was to be expected, lumping does not have any effect on the results for the CI method. Furthermore, Bi-CGSTAB requires substantially fewer iterations than GMRES. Note that the CI results correspond more or less to the efficiency that one would get for a standard Laplacian on a stretched grid.

Since it is not easy to compare the computational effort required for one GMRES iteration with that for one Bi-CGSTAB iteration, we measured for both methods the computation time t_s to solve one pressure-correction equation. We observe, see Figure 5.9, that Bi-CGSTAB needs less computing time than GMRES, and that lumping of the preconditioner, if appropriate, should be employed. The pressure-correction matrix resulting from the CI method is cheaper (roughly a factor 2 or 3 in CPU time) to solve than the matrix arising from the PI method, but, as stated above, the CI method has the danger of odd-even decoupling of the pressure.

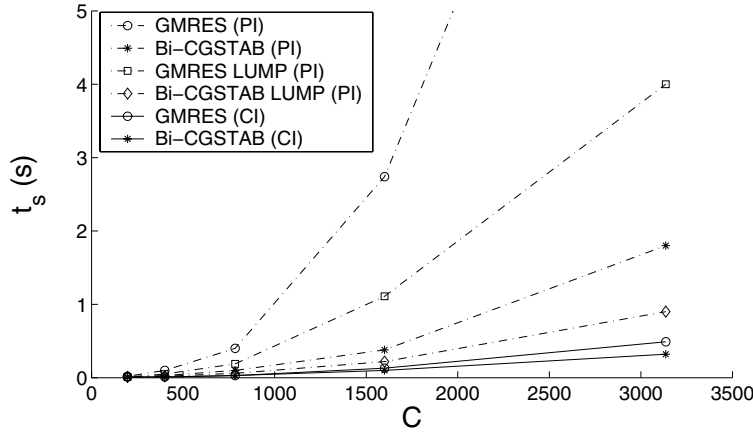


Figure 5.9: Computation time required to solve one pressure-correction equation as function of the number of unknowns at grids of type $R8$.

5.1.4 Properties of the viscous and Laplacian matrices

Both the viscous and Laplacian operators are symmetric positive definite, which makes it desirable that this holds for the discretization as well. The corresponding matrices turn out to be nonsymmetric, hence the present discretization is not symmetry-preserving. Closely related, see Appendix C.1, is the property of positive definiteness. Because the eigenvalues of positive definite matrices are real and positive, deviations of the eigenvalues in the complex plane from the positive real axis may be interpreted as a measure for how ‘close’ the studied matrices are to being positive definite. In Figure 5.10c the eigenvalues of the viscous and Laplacian matrices, the latter with various gradient schemes, as obtained on the grids of Figures 5.10a-b are shown. All eigenvalues have, as required, a positive real part. The imaginary part of the eigenvalues of the viscous matrix is much affected when the grid is made less regular. This conclusion holds, to a somewhat lesser extent, also for the Laplacian matrices computed with the AP, CI and FQ gradient schemes. The PI method yields, of all methods, eigenvalues that are closest to the positive real axis. Note that this is in correspondence with the conclusions drawn in Section 5.1.2, e.g. the PI method is, of the four studied gradient methods here, the most accurate one.

5.1.5 Conclusions

We studied for Poiseuille flow the accuracy of several schemes to compute the pressure gradient. On grids that are made irregular deliberately, the PI is slightly more accurate than the CI method. The AP method is less accurate than the former two, and the FQ method yields unacceptable results. On stretched Courant grids, frequently encountered in the computation of high-Reynolds number flow, spurious

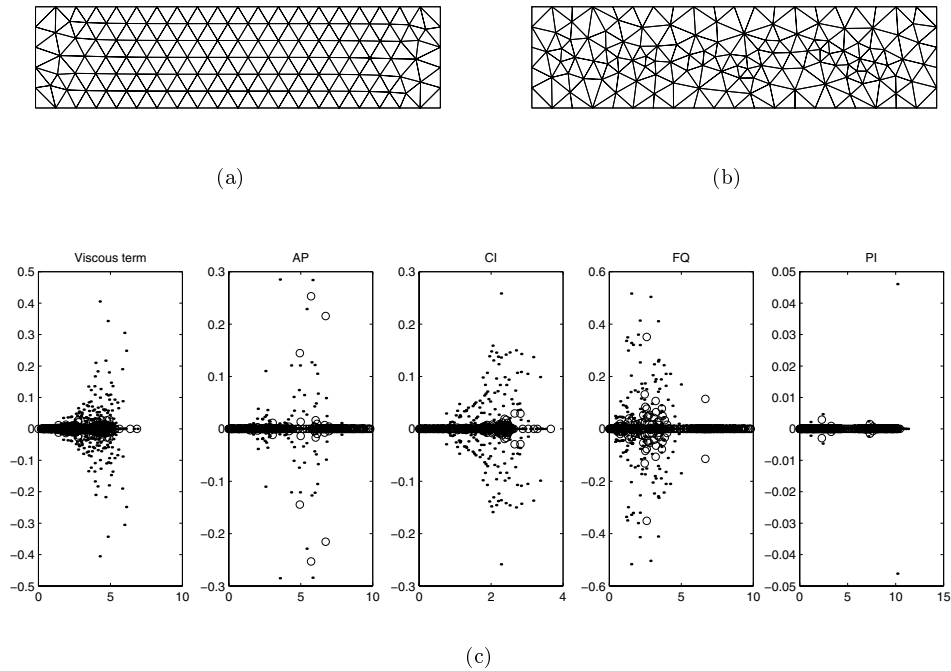


Figure 5.10: Regular (a) and distorted (b) grid. In (c) the eigenvalues of the viscous (left figure) and Laplacian matrices (four remaining figures) as obtained on the regular and distorted grid are indicated by \circ and \cdot , respectively.

pressure modes degrade the accuracy of the CI method, making the use of this method for the computation of incompressible flows questionable. On stretched Courant grids, all schemes (apart from the CI method) are close to second order accuracy. A disadvantage of the PI method is that it can result in pressure-correction matrices that are not M-matrices, and as a consequence the resulting pressure-correction system is expensive to solve. Solving this system should be done by means of Bi-CGSTAB together with a lumped ILU preconditioner, since GMRES and standard (not lumped) ILU preconditioning are much more expensive. A numerical study of the symmetry properties of the viscous and pressure-correction matrices is performed, from which we conclude that the discretized viscous term is affected when the grid is made less regular, while the pressure-correction matrix following from the PI method remains close to positive definite.

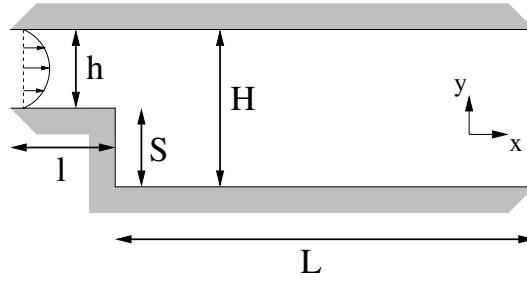


Figure 5.11: Geometry of the backward facing step.

5.2 Backward facing step

A suitable test case for validation of incompressible flow methods is the backward facing step. The problem definition, including information concerning the numerical experiments we performed, is given in Section 5.2.1. Numerical results for various Reynolds numbers between 100 and 800 are discussed in Sections 5.2.2 to 5.2.4. The reattachment length forms the topic of Section 5.2.5, and conclusions are given in Section 5.2.6.

5.2.1 Problem definition

The geometry of the 2D backward facing step is defined in Figure 5.11. The experimental configuration as used in the experiments of Armaly [2] had the following dimensions: $h = 5.2$ mm, $H = 10.1$ mm, $S = H - h = 4.9$ mm, $l = 200$ mm and $L = 500$ mm. The Reynolds number is, following Armaly, defined as

$$Re = \frac{u_r L_r}{\nu}, \quad (5.3)$$

where $L_r = 2h$ is the height of the inlet channel, u_r is two-thirds of the measured maximum inlet velocity, and $\nu = \mu/\rho$ is the kinematic viscosity, which is $\nu = 1.5 \times 10^{-5}$ m²/s for air at 20°C [31]. For laminar flows, to which we will restrict ourselves, the velocity at the inlet is given by a parabolic profile in the x -direction:

$$u = u(y) = u_{\max} \left[1 - \left(\frac{y - h/2}{h/2} \right)^2 \right], \quad 0 \leq y \leq h. \quad (5.4)$$

It is easy to deduce that in the laminar case $u_r = 2u_{\max}/3$ corresponds to the average inlet velocity. Note that in many computations, to name only a few references [20, 24, 56], for apparent reasons of simplicity the stepheight S is put equal to $2h$. In other works, for example [33, 57], the geometry is even more simplified by omitting the entire inlet channel, resulting in a rectangular domain. As far as we know no one has published a systematic study of how these geometrical differences

influence the solution.

In order to investigate grid sensitivity of the numerical solution, we define five different types of grids. The letters labeling these grids refer to the corresponding SEPRAN grid generator [95]: G (general), R (rectangular), Q (quadrilateral), T (triangle) and I (ISNaS), see also Figure 5.12. Grids of type G are smooth unstructured grids. Grid types Q and R are structured triangular grids, where grids of type Q are used to investigate sensitivity with respect to arbitrary orientations of the oblique faces. Grids of type T are less smooth than the other ones, and grids of type I are structured multiblock grids. Computations on the first four grid types are carried out with the unstructured staggered scheme described in this thesis. The classical staggered Marker-And-Cell (MAC) scheme [44], which is known to be very accurate, is used on grid type I . A parameter, to be denoted grid level, indicates the fineness of the grid. Grid-levels 1, 2 and 3 correspond to 3, 6 and 12 vertices per unit of length S , respectively, distributed in an equidistant manner at the domain boundaries. Unless stated otherwise, we locate the inlet at $x = -l = -3$ (again in units of S). The length L of the computational domain behind the step is chosen such that the presence of the step is not felt at the outflow boundary, i.e. at the outflow the flow has reverted back to a parabolic profile. This consideration has led to, in units of S : $L = 10$ for $Re = 100, 200$ and 300 ; $L = 15$ for $Re = 389$ and 500 , and $L = 25$ for $Re = 600, 700$ and 800 . We choose, unless stated otherwise, the same ratio for S/h as in the experimental configuration of Armaly, i.e. $4.9/5.2$. For $L = 10$, the number of vertices in the various grids is given between parentheses: $R2, Q2, I2$ (919); $G2$ (1078); $T2$ (1234); $R3, Q3, I3$ (3493); $G3$ (4168); $T3$ (5144). Obviously, for larger values of L these numbers increase accordingly.

For the inertia term, central differencing is applied, since the numerical diffusion introduced by an upwind scheme would artificially lower the effective Reynolds number. Unless stated otherwise, the path integral method is used for computation of the pressure gradient. At the inlet, the parabolic velocity profile (5.4) is prescribed and at solid walls the no-slip condition is enforced. At the outflow boundary we assume free flow ($\sigma^{nt} = \sigma^{nn} = 0$); whether this is correct was verified afterwards. We impose zero velocity and zero pressure as initial condition. All computations were driven to steady state by time stepping; a relative accuracy of $\epsilon = 10^{-2}$ between two successive time steps turned out to be sufficient.

The advantage of using the implicit instead of the explicit Euler scheme for time stepping can be demonstrated by comparing the stability restriction of the latter with the actual time steps used in our computations. Since the Navier-Stokes equations are extremely complicated, some heuristic arguments have to be employed to arrive at an appropriate estimate for the time-step restriction for the explicit Euler method. Assume that the Neumann stability restrictions for a one-dimensional nonstationary convection-diffusion equation apply equally well to the velocity pre-

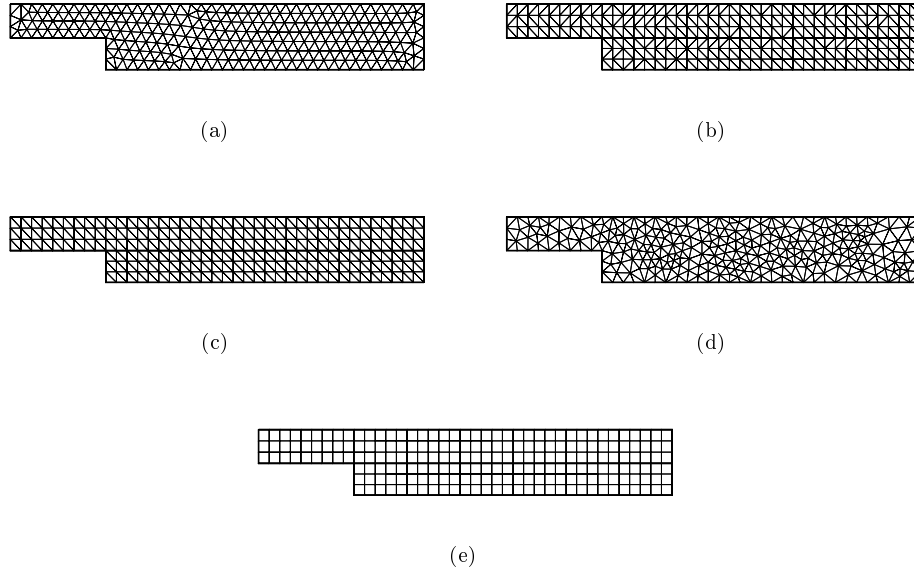


Figure 5.12: Grids $G1$ (a), $Q1$ (b), $R1$ (c), $T1$ (d) and $I1$ (e), with $L = 10$ and $l = 3$.

diction step; the pressure-correction step is thought not to hamper stability. With a diffusion coefficient equal to $1/Re$, the following restriction for Δt for the explicit Euler scheme is derived (see also Section 5.8 of [129]):

$$\Delta t \leq \min\{\Delta t_i, \Delta t_v\}, \quad \Delta t_i = 2/u^2 Re, \quad \Delta t_v = \Delta x^2 Re/2, \quad (5.5)$$

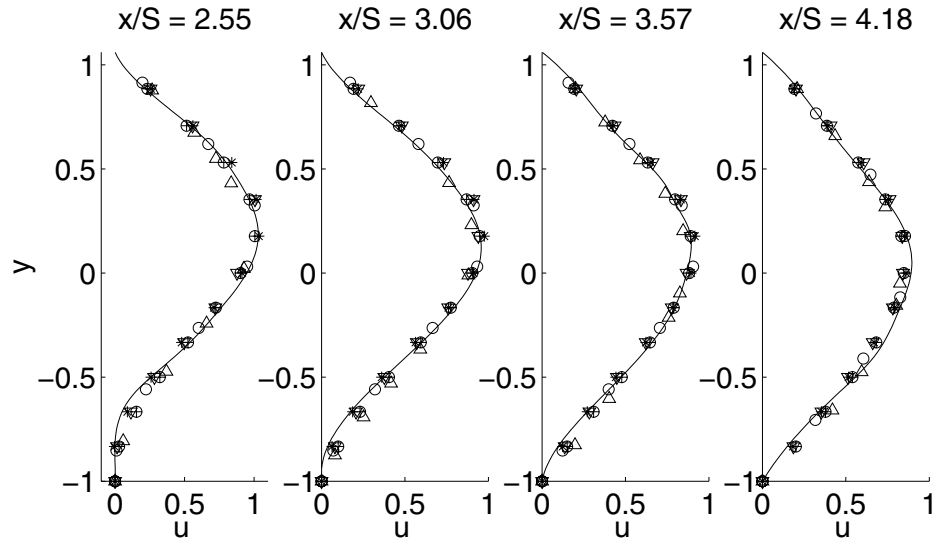
where Δt_i and Δt_v represent the time-step restriction stemming from the inertia and the viscous term, respectively. From these relations we deduce that for $Re > 2/u\Delta x$ the inertia term limits stability; for $Re < 2/u\Delta x$ it is the viscous term that restricts the allowable time step. In the computations for the backward facing step, the values for u , Re and Δx are always such that the inertia term would limit stability if the explicit Euler scheme were used. With $u \approx 1$, the time step for the explicit Euler scheme should be chosen such that $\Delta t < 2/Re$, which ranges from 2.0×10^{-2} to 2.5×10^{-3} for Reynolds numbers varying from 100 to 800. The time steps we used with the implicit Euler scheme are of the order 1, i.e. two to three orders of magnitude larger. For $Re = 100$ and $Re = 389$, the number of time steps required to arrive at a converged solution is about 20 and 50, respectively, for $\Delta t \approx 2$.

5.2.2 Solution for $Re = 100$

The streamlines of the converged solution for $Re = 100$ are shown in Figure 5.13a. The computed values for the reattachment length, i.e. the horizontal distance be-



(a)



(b)

Figure 5.13: Solution for $Re = 100$. Streamlines computed on grid $G3$ (a), and the horizontal velocity, in units of u_r , at several vertical intersections of the domain (b). The line corresponds to the experimental results of Armaly, and the symbols refer to the grid used: \circ : $G2$; \oplus : $Q2$; $*$: $R2$; \triangle : $T2$; ∇ : $I2$.

Table 5.1: Computed values for the reattachment length, in units of S .

Re	Grid-level	G	Q	R	T	I
100	2	2.7	2.6	2.8	2.4	2.5
	3	2.8	2.7	2.8	2.6	2.7
389	2	7.6	6.9	8.0	6.2	7.3
	3	7.9	—	8.0	—	7.5

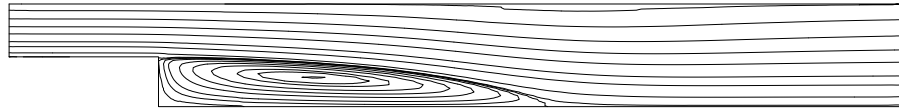
tween the step and the point where the recirculation zone ends, are collected in Table 5.1. Especially those obtained on grids of type G and R are close to $3.0S$, the experimental value. This was to be expected since grid types G and R are smoother than the other ones; apparently grids of type I contain too few cells to match the accuracy attained at grids of type G and R .

The measured and computed velocity in the x -direction at several vertical intersections of the domain are depicted in Figure 5.13b. The results at grids $G2$ and $R2$ seem to be the most accurate, although differences between the various results are small and a good correspondence between numerics and experiment is already present on grid-level 2. The velocity profiles computed on grids with grid-level 3 are not shown, but it suffices to state that they are even closer to the experimental results.

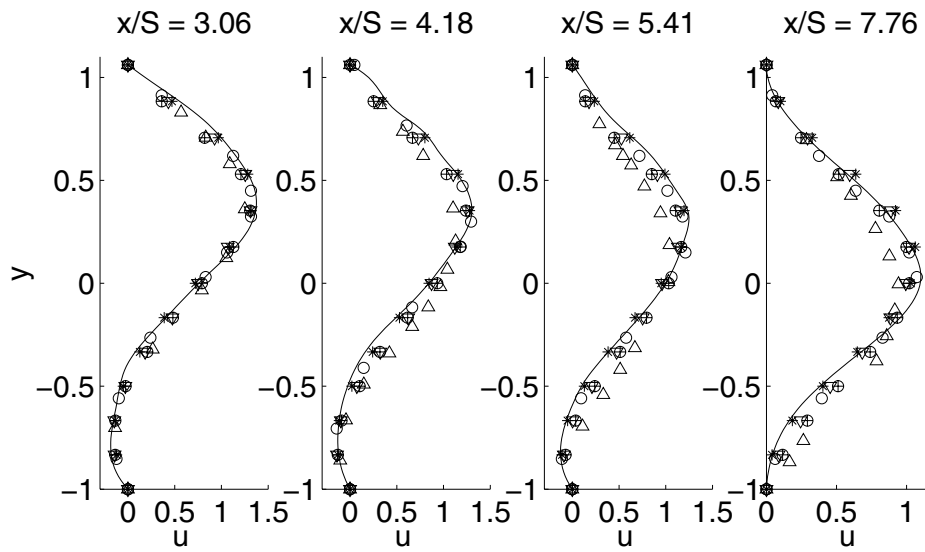
In addition, we studied the use of the various gradient schemes. Application of the AP, FQ, CI and PI method for computation of the pressure gradient leads hardly to any differences between the resulting steady state solution on grids $G2$ and $R2$. On grid $Q2$, the isobars computed with the AP, CI and FQ method are virtually the same, but slightly less accurate than the isobars obtained with the PI method, as we derived from comparison with solutions obtained at finer and smoother grids. On grid $T2$, the CI method does not converge and the FQ method yields less accurate results while needing more computation time than the AP and PI method. That it is best to use the PI method is in agreement with conclusions drawn from the study of the Poiseuille flow, see Section 5.1.

5.2.3 Solution for $Re = 389$

The streamlines of the converged solution on grid $G3$ are shown in Figure 5.14a. Computed values for the reattachment length are given in Table 5.1, and the experimental value is $8.3S$. On grids $Q3$ and $T3$ no computations were done, because, in view of the results on grids $Q2$ and $T2$, the accuracy is expected to be less than on grids G , I and R . It can be concluded from this that computation of the inertia term is by far more sensitive to grid irregularities than that of the pressure gradient and viscous operator, recalling that these irregularities made themselves



(a)



(b)

Figure 5.14: Solution for $Re = 389$. Streamlines computed on grid $G3$ (a), and the horizontal velocity, in units of u_r , at several vertical intersections of the domain (b). The line corresponds to the experimental results of Armaly, and the symbols refer to the grid used: \circ : $G2$; \oplus : $Q2$; $*$: $R2$; \triangle : $T2$; ∇ : $I2$.

less felt in flows with a smaller Reynolds number. Rida et al. [91] obtain for the same Reynolds number a reattachment length equal to $7.65S$ on a grid consisting of 2240 cells that is fine near the inlet and stretched near the outlet. Thomadakis and Leschziner [103] use a structured triangular grid consisting of 21,804 cells. They claim that the experimental and the computed reattachment point lie at $3.7S$ and $3.65S$ respectively; these must be typing mistakes. Profiles of the velocity at some vertical intersections of the domain are shown, for grid-level 2, in Figure 5.14b. We observe a good agreement between theory and experiment at grids of type G , R and I ; the results at grids Q and T are less accurate.

5.2.4 Solution for $Re = 500, 600, 700$ and 800

As we found out the steady state solutions obtained grid-level 2 are not accurate for $Re > 400$ because of the coarseness of the grid. On the other hand, for higher Reynolds numbers it becomes increasingly more difficult to reach steady state, especially on fine grids. This is usually attributed to the fact that the flow becomes three-dimensional [2] for $Re > 400$, questioning the existence of a two-dimensional steady state for these Reynolds numbers. It turns out that for $Re \geq 600$ we could not drive the flow to steady state on grids of grid-level 3 ($G3$, $R3$ and $I3$), not even after restarting with the solution of $Re = 500$ as initial condition. In order to get rid of the singularity at the step, we decided to use the domain as defined for example in the computations of Kim and Moin [57], Gartling [33] and Sohn [101]. The two relevant differences between this domain, which we label with (C), and the experimental one are: (i) in domain (C), the inlet is located on top of the step, resulting in a rectangular domain, and (ii) in domain (C), the stepheight S is taken equal to h , the height of the inlet. A triangulation similar to that of type $G3$ is taken, resulting in a grid containing 6999 vertices. On the same grid, no steady state solution could be obtained for $Re = 700$, not even after restarting from converged solution at $Re = 600$.

5.2.5 Reattachment length

As we saw above, the reattachment length varies strongly with the Reynolds number, and its computation is very sensitive to grid irregularities. Computed values for the reattachment length on grids of type $I3$, $R3$ and $G3$ (both at the experimental geometry and geometry (C)) are shown, together with experimental data from [2] and numerical results from Kim and Moin [57], in Figure 5.15. Note that it is hard to get accurate quantities from [2, 57] since no tabular results were given; graphical results had to be scanned optically. The results of Kim and Moin are obtained, on geometry (C), by means of the MAC scheme on a grid consisting of 101×101 grid points, with the outstream boundary located at $x = 30S$. Up to $Re = 300$, all results do not differ significantly. For larger values of Re , deviations start to

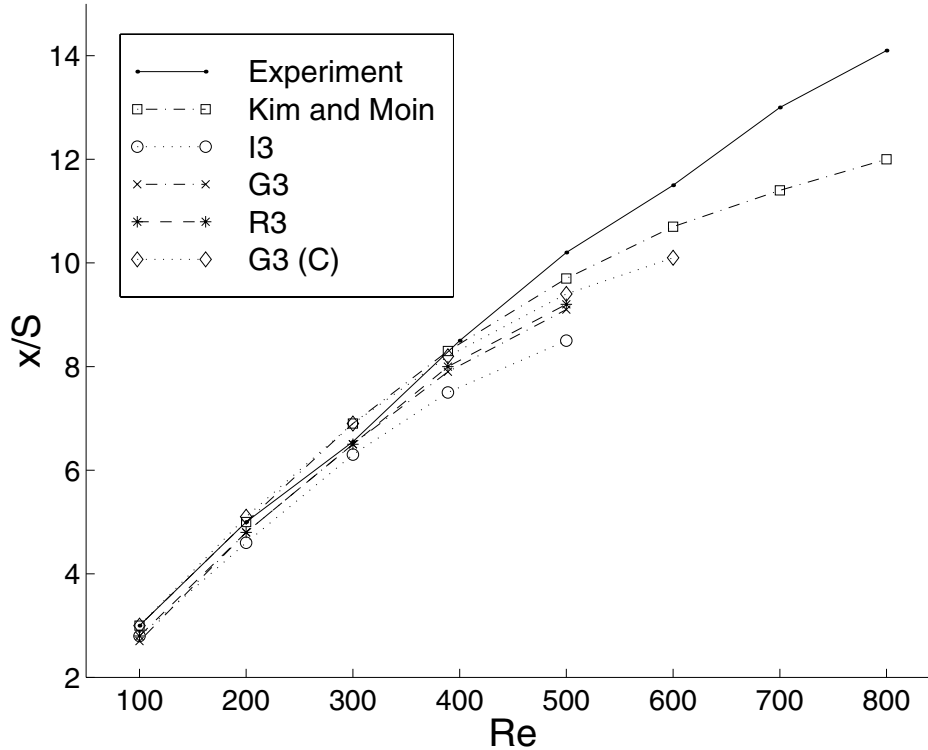


Figure 5.15: Reattachment length versus Reynolds number.

appear. The differences between the experimental results and the numerical results are usually attributed to the fact that the flow becomes three-dimensional for values of Re larger than 400, see [2]. The small differences between the experimental and the computational (C) geometry can be felt from $Re \geq 400$. Only for $Re = 600$, the changes between the Kim and Moin result and the G3 (C) result become significant. Virtually no differences between the G3 and R3 solution can be distinguished. This is desirable, since grids of type R3 are structured triangular grids (hence accurate), while type G3 is a ‘true’ unstructured grid. The less accurate MAC scheme results at grid I3 must be caused by the smaller amount of grid cells and faces for a given node distribution at the boundaries. Indeed, on a finer grid better results are obtained (results not shown).

5.2.6 Conclusions

Computations that we did for the backward facing step using the unstructured staggered scheme are compared with other, both experimental and numerical, results. On relatively coarse grids (grid-levels 1 and 2), steady state solutions were obtained

over the whole studied range in Reynolds numbers ($100 \leq Re \leq 800$). The Euler implicit scheme allows for time steps which are in the order of 10^2 or 10^3 larger than the time-step restriction for the explicit Euler scheme. For values of Re up to 300, differences between numerical and experimental results are small, demonstrating that the viscous and pressure gradient operator perform satisfactorily on relatively coarse and non-smooth grids. For higher Reynolds numbers, the grids need to be fine and smooth in order to come close to other published results. Hence, the inertia term is by far more sensitive to grid irregularities than the pressure and viscous term, while, in addition, it becomes increasingly difficult to arrive at a stationary solution. For fine grids, the flow could not be driven to steady state for $Re \geq 700$, not even if the steady state solution of $Re = 600$ was used as initial condition.

5.3 Lid-driven cavity flow

The lid-driven cavity flow has served over and over again as a model problem for testing and evaluating numerical techniques. The problem definition is given in Section 5.3.1, and numerical results are discussed in Section 5.3.2. A comparison with results obtained using other unstructured grid methods as presented in the literature are, together with some conclusions, gathered in Section 5.3.3.

5.3.1 Problem definition

The lid-driven cavity flow is a laminar incompressible flow in a square cavity with a top wall that moves with a uniform velocity in its own plane. At all walls, the no-slip condition is enforced. The Reynolds number Re is based on the velocity of the top wall and the size of the cavity. The selected Reynolds numbers are 100, 400, 1000, 3200, 5000, 7500 and 10,000. To minimize CPU time and to improve initial conditions for the high-Reynolds number cases, a restart procedure with incremental increase of Re is used. In the case of $Re = 100$, the solution of Stokes flow ($Re = 0$) is chosen as initial condition. The relative stationary accuracy, see equation (3.1), is put to 10^{-2} . Central differencing for the inertia term and the path integral method for the pressure gradient are employed. Three grids, with $n = 30$, 60 and 120 nodes at each wall, are used. The size of the elements is roughly the same over the domain, see Figure 5.16. The coarse grid ($n = 30$) has 2058 cells, 3147 faces and 1090 vertices, the grid with $n = 60$ has 8374 cells, 12,681 faces and 4308 vertices, and the fine grid ($n = 120$) has 33,492 cells, 50,478 faces and 16,987 vertices. The solutions of Ghia et al. [34], who used a multigrid finite difference method for the streamline-vorticity formulation with a fine (257×257 grid points) and uniform grid, are generally considered to be of benchmark quality. Another thorough study of the lid-driven cavity is presented in Sohn [101], who used a consistent integration penalty finite element method with $Q2/P1$ (nine-node biquadratic velocity and linear

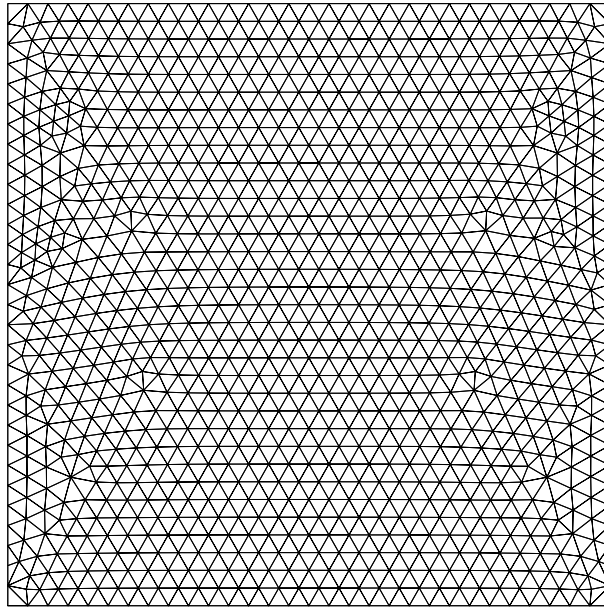


Figure 5.16: Coarse grid used for the lid-driven cavity flow, with 30 nodes at each wall.

discontinuous pressure) finite elements on a nonuniform grid consisting of 40×40 rectangular elements (81×81 grid points).

5.3.2 Results

Figure 5.17 shows the velocity profiles for the horizontal velocity along the vertical line through the geometric center. The thinning of the wall boundary layers with increasing Re is evident for these profiles, although the rate of thinning is very slow for $Re \geq 5000$. The near-linearity of the velocity profile in the central core of the cavity is indicative for the uniform vorticity region that develops here for large Re . We observe that for up to $Re = 1000$, a good correspondence between the velocity profiles at the coarse grid and the benchmark result is obtained. At the grid with $n = 60$, a good agreement holds up to $Re = 5000$. The fine grid solution remains accurate up to $Re = 10,000$.

The streamlines, i.e. contourlines of the streamfunction ψ , for $Re = 1000$ at the coarse grid and for $Re = 10,000$ at the fine grid are depicted in Figure 5.18. As is well-known, the solutions exhibit additional counter-rotating vortices in or near the cavity corners as Re increases. The letters P, T, B, L and R denote primary,

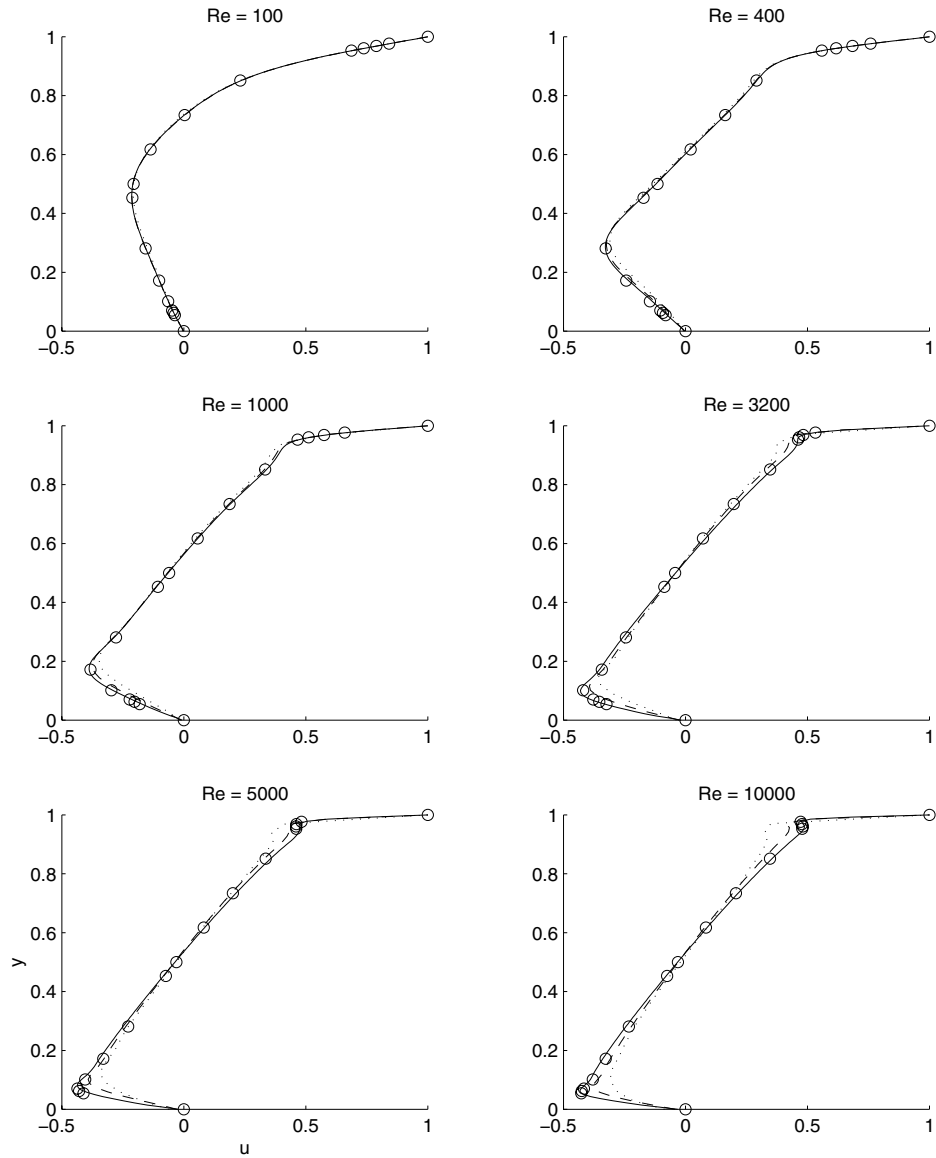


Figure 5.17: Comparison of horizontal velocities along the vertical centerline for several Reynolds numbers. Notation: $n = 30$ (\cdots); $n = 60$ ($- -$); $n = 120$ ($—$); Ghia et al. (\circ).

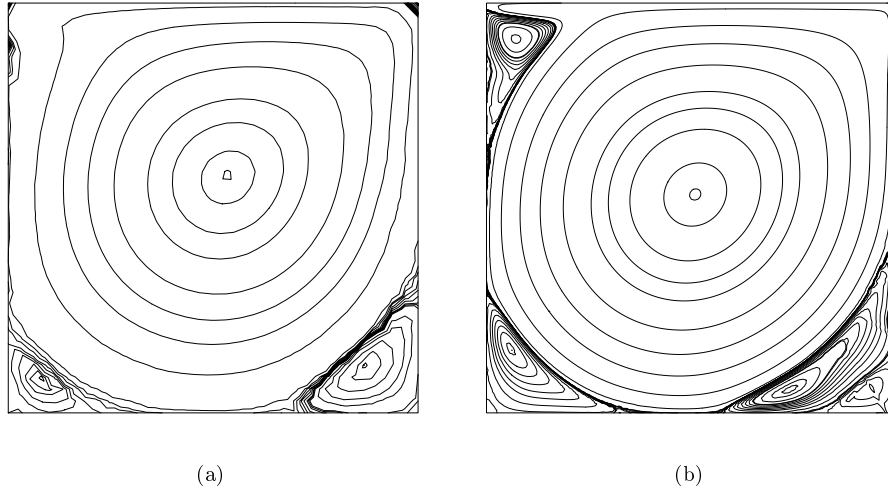


Figure 5.18: Streamlines for $Re = 1000$ (a), obtained at the coarse grid, and $Re = 10,000$ (b), obtained at the fine grid.

top, bottom, left and right respectively, and the numerical subscript denotes the hierarchy of secondary vortices. For example, in Figure 5.18a the vortices P, BL_1 , BR_1 and BR_2 are visible, while in Figure 5.18b the vortices P, BL_1 , BL_2 , BR_1 , BR_2 and TL can be distinguished. For $Re \leq 1000$, the location of the centers of the vortices as computed on the coarse grid is less than 1 percent removed from the coordinates given in [34]. At the grid with $n = 60$, this close correspondence holds for $Re \leq 3200$, and at the fine grid this is the case for the complete range of studied Reynolds numbers. In [34] it is stated that the values of ψ at the vortex centers are more sensitive to the meshwidth than the velocity profiles. In Table 5.2, the extreme values of the streamfunction at various vortices inside the cavity are given, including the values listed in [34, 101]. As usual for this testcase, we put $\psi = 0$ at the boundaries. The fine grid solution shows a good agreement with the benchmark solution of Ghia. The accuracy of the solution of Sohn (81×81 grid points) is somewhere between that of the $n = 60$ and $n = 120$ solutions.

5.3.3 Discussion and conclusions

Many of the incompressible flow solvers on unstructured grids that are described in Section 2.4 have been used to solve the lid-driven cavity problem, and we will discuss their results. In [56], the streamlines of the solution for $Re = 1000$, obtained on a grid consisting of 2450 triangles with a second order upwind scheme, are shown. In order to resolve boundary layers near the wall, the grid is clustered to the walls, while

Table 5.2: Extreme values of the streamfunction at various vortices inside a lid-driven cavity.

Re	Vortex (unit)	$n = 30$	$n = 60$	$n = 120$	Ghia [34]	Sohn [101]
100	P (10^{-1})	-1.01	-1.03	-1.03	-1.03	-1.02
	BR ₁ (10^{-5})	2.54	1.44	1.34	1.26	1.16
	BL ₁ (10^{-6})	2.10	2.04	1.75	1.75	1.69
400	P (10^{-1})	-1.09	-1.13	-1.14	-1.14	-1.12
	BR ₁ (10^{-4})	5.63	5.52	6.55	6.42	6.13
	BL ₁ (10^{-5})	1.55	1.11	1.23	1.42	1.32
1000	P (10^{-1})	-1.09	-1.16	-1.19	-1.18	-1.15
	BR ₁ (10^{-3})	1.26	1.42	1.75	1.75	1.63
	BL ₁ (10^{-4})	2.01	2.06	2.26	2.31	2.17
3200	P (10^{-1})	-1.06	-1.15	-1.24	-1.20	-1.16
	BR ₁ (10^{-3})	1.85	1.92	2.86	3.14	2.60
	BL ₁ (10^{-4})	10.1	9.58	10.3	9.78	10.4
	TL (10^{-4})	15.0	4.33	7.53	7.28	6.06
5000	P (10^{-1})	-1.05	-1.14	-1.24	-1.19	-1.15
	BR ₁ (10^{-3})	1.75	2.09	2.99	3.08	2.80
	BL ₁ (10^{-3})	1.79	1.10	1.26	1.36	1.25
	TL (10^{-3})	1.85	0.97	1.48	1.46	1.29
7500	P (10^{-1})	-1.04	-1.13	-1.24	-1.20	-1.13
	BR ₁ (10^{-3})	0.64	1.94	2.99	3.28	2.86
	BL ₁ (10^{-3})	2.66	1.28	1.38	1.47	1.55
	TL (10^{-3})	0.95	1.86	2.14	2.05	1.83
10,000	P (10^{-1})	-0.99	-1.10	-1.23	-1.20	-1.12
	BR ₁ (10^{-3})	0.77	1.63	2.80	3.42	2.80
	BL ₁ (10^{-3})	3.17	1.34	1.56	1.52	1.37
	TL (10^{-3})	0.33	2.60	2.56	2.42	2.18

the triangular cells are formed by subdividing rectangular cells. A good agreement with the solution of Ghia is reported. In [24], a uniform triangular grid used for $Re = 100$ and 400 consists of 1654 nodes, and of 14,996 nodes for $Re = 3200$. For $Re = 100$ and 400 , the velocity profiles obtained from the central scheme correspond closely to the results of Ghia, but the BL_1 and BR_1 vortices seem not present in the streamline plots. A less good agreement with the results of Ghia, both for the velocity profile as the streamlines, is reported for $Re = 3200$. The results presented in this thesis are closer to the benchmark solution. A grid refinement study on uniform triangular grids for $Re = 400$ is given in [53]. The velocity profile agrees well with the benchmark solution, but the value for the streamfunction in the center of primary vortex is far off the value $\psi = -0.114$ as obtained by Ghia, e.g. $\psi = -0.092$ for $n = 30$ and $\psi = -0.105$ for $n = 60$. The present unstructured staggered grid method yields results that are closer to the benchmark solution, see Table 5.2. A good agreement with the benchmark solution for $Re = 1000$ is reported in [60] with the use of 10,240 control volumes. In [103] a grid refinement study using quadrilateral and triangular grids is described. For $Re = 400$, a 55×55 quadrilateral grid (2916 cells) clustered towards the walls does not yield accurate solutions. Subdivision of each rectangle in this grid into four triangles (11,664 cells) gives a grid on which results are obtained that, in our opinion, still need improvement near the walls. A 120×120 quadrilateral grid (14,161 cells) clustered towards the walls leads to unacceptable results for $Re = 3200$. Repetition of the subdivision strategy just mentioned results in a grid with 56,644 triangles, and on this very fine grid close agreement with the benchmark data is obtained. Our method is, for a given grid size, more accurate. The covolume method is applied on a grid consisting of 1016 triangles to solve the driven cavity problem for $Re = 400$, resulting in a reasonable agreement, see [41]. The upwind scheme is used for $Re = 4000$, since the central scheme is unstable for this Reynolds number. The method fails to resolve the TL vortex, which is present already at $Re = 3200$.

It can be concluded that the present unstructured staggered scheme yields very accurate solutions for the lid-driven cavity flow.

Chapter 6

Results for inviscid flows

In this chapter, numerical results for inviscid flows obtained with the unstructured staggered scheme described before are given. Attention will be paid to the accuracy of the spatial discretization and to the efficiency of the sequential update procedure for fully compressible flows (FC) and the Mach-uniform approach (MU). Unless stated otherwise, the following options are taken as default: the control volume for the momentum equation consists of two triangles; the first order upwind scheme is employed for all equations; the convecting velocity in the inertia term is computed in a central fashion, see (4.12); the pressure gradient is evaluated by means of the path integral method; time stepping is carried out using the implicit Euler method with a fixed time step; defaults for the linear solver as discussed in Section 3.6 are used; the primary energy variable in the FC approach is ρH , and method 1, see Section 4.6.2, is employed to approximate vector quantities in cell centers. Accuracy of the approximation of the inertia term is studied in Section 6.1 by computing solutions to the Burgers equation. In Section 6.2 the accuracy of our scheme for Riemann problems is compared with analytical solutions and solutions obtained by standard first order methods for compressible flows. Furthermore, Riemann problems offer a nice opportunity to study various discretization schemes, since many important flow phenomena that occur in inviscid flows are present while there is no ambiguity concerning the exact solution. A grid refinement study for incompressible, subsonic and supersonic flow in a channel with bump is performed and some properties of the inertia matrix are investigated in Section 6.3. Four distinct flows (low subsonic, subcritical, transonic and supersonic) around a NACA 0012 airfoil are computed in order to compare the Mach dependence of the performance of the FC and MU approach, and this forms the topic of Section 6.4. Supersonic flow over a circular blunt body with a freestream Mach number equal to 4.0 is discussed in Section 6.5. In nozzles, large subregions with incompressible flow as well as large subregions with compressible flow can coexist. This poses a problem for standard compressible flow solvers. Results given in Section 6.6 for various contraction ratios

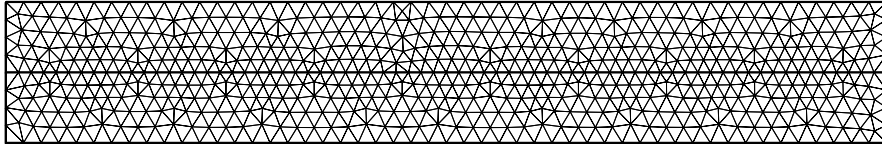


Figure 6.1: Grid on which the Burgers equation and Riemann problems are solved.

demonstrate again the Mach-uniform accuracy and efficiency of the MU approach. Conclusions are gathered in Section 6.7.

6.1 Burgers equation

To test the accuracy of the first order upwind scheme (4.18) for the inertia term, we apply it to a simple flow case in which the pressure and density are constant and the flow is horizontal: $\mathbf{u} = (u, 0)$, with $u = u(x, t)$, so that effectively we are solving the inviscid Burgers equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} = 0. \quad (6.1)$$

Unless specified otherwise, all figures in this section and Section 6.2 are taken along the centerline, having 70 faces of uniform size, of the grid shown in Figure 6.1. A fine grid similar to this one, with 550 faces along the centerline, is used to study grid convergence. This grid, to be referred to as the fine grid, is not shown. We will present numerical results to (6.1) for two different initial conditions. In Section 6.1.1 the solution consists of a uniformly moving shock wave. The solution to the problem given in Section 6.1.2 contains a shock wave and an expansion fan.

6.1.1 Solution with a shock wave

For an initial condition given by

$$u(x, 0) = \begin{cases} u_l & \text{if } x \leq x_0; \\ u_r & \text{if } x > x_0, \end{cases} \quad (6.2)$$

where u_l and u_r are constant states and $u_l > u_r$, the genuine weak solution of (6.1) is

$$u(x, t) = u(x - x_0 - st, 0), \quad s = u_l + u_r, \quad (6.3)$$

which states that the initial discontinuity propagates undisturbed with speed s . We take $u_l = 1$, $u_r = 0.5$ and $x_0 = 0.5$. The time step follows from specification of the Courant number, defined in Appendix A.4.2. In the grid of Figure 6.1, the

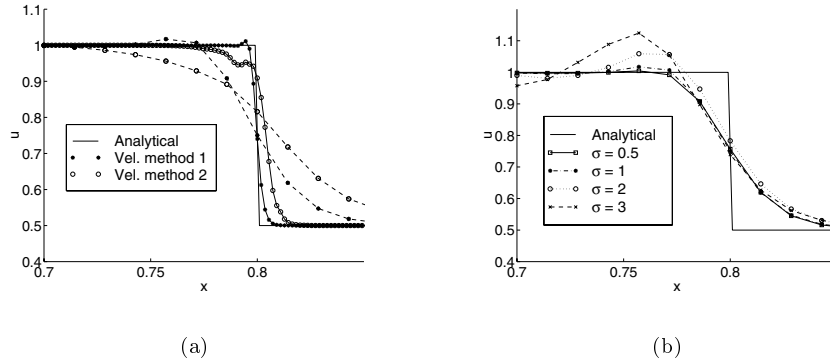


Figure 6.2: Numerical solutions to the Burgers equation with a shock wave. Results obtained with different velocity interpolation methods with 70 (dashed line) and 500 (continuous line) faces along the centerline (a). Results obtained with central approximation of the velocity and different Courant numbers on the coarse grid (b).

smallest value for Δx is 4.3×10^{-3} , and in the fine grid it is 4.4×10^{-4} . The velocity interpolation methods 1 (central interpolation) and 2 (upwind interpolation) for the convecting velocity in the inertia term, see page 56, lead at $t = 0.2$ and a Courant number equal to 1.0 to the results shown in Figure 6.2a. Recall that, for the convecting velocity, the first order upwind scheme (4.18) is applied in all cases. We conclude that both methods lead to the correct shock speed and that upwind approximation of the convecting velocity (method 2) as done in expression (4.14) introduces more numerical diffusion than method 1, the central velocity interpolation method, see equation (4.12). The overshoot, which presence could be expected since the discretization of the inertia term is not monotone, see Appendix A.3, is small and remains small even on fine grids. This overshoot grows, as illustrated in Figure 6.2b, with increasing Courant number. This is not peculiar because also standard compressible flow methods loose their monotonicity preserving properties for Courant numbers larger than a certain threshold. Since the unstructured staggered grid method does not consider the momentum equations for each Cartesian coordinate direction separately, our method considers each flow as a truly 2D flow, even though the exact solution for this problem is 1D. As a consequence, there are small fluctuations in v , the velocity in the y -direction. The largest value for v is 8×10^{-3} for velocity interpolation method 2, and 6×10^{-3} for interpolation method 1. These variations, smaller than a percent of the value of u_l , can be considered as very small. Note that these fluctuations influence on their turn the velocity in the x -direction, and this explains the presence of the strange feature in the shock as computed by velocity interpolation 2 on the fine grid. All computations in the remainder of this thesis are carried out with central interpolation (method 1) of the

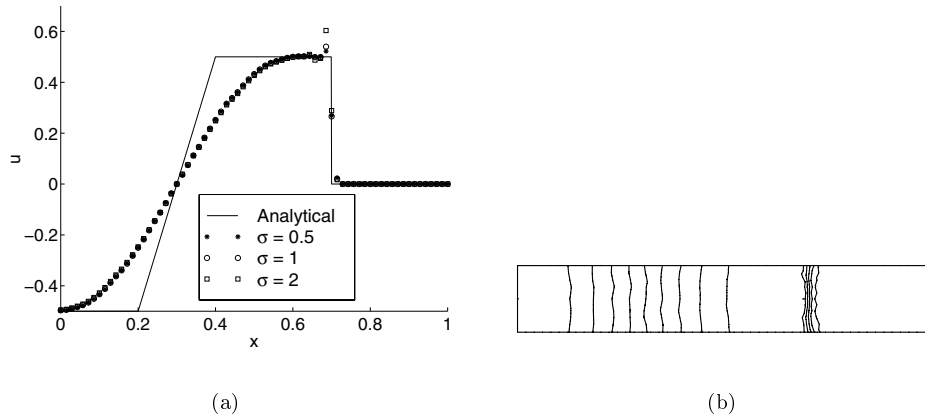


Figure 6.3: Numerical solutions to the Burgers equation with a shock wave and expansion fan for various Courant numbers (a), and a contourplot of u (b).

convecting velocity in the inertia term, since upwind approximation (method 2) of this term is considered to be too dissipative.

6.1.2 Solution with a shock wave and expansion fan

For the following initial condition:

$$u(x, 0) = \begin{cases} u_l & \text{if } x \leq x_l; \\ u_m & \text{if } x_l < x < x_r; \\ u_r & \text{if } x \geq x_r, \end{cases} \quad (6.4)$$

with $u_l < u_m$ and $u_m > u_r \geq 0$, the vanishing viscosity solution of the Burgers equation is given by:

$$u(x, t) = \begin{cases} u_l & \text{if } \frac{(x-x_l)}{t} \leq u_l; \\ \frac{(x-x_l)}{t} & \text{if } u_l < \frac{(x-x_l)}{t} < u_m; \\ u_m & \text{if } u_m \leq \frac{(x-x_l)}{t} \text{ and } x < (st + x_r); \\ u_r & \text{if } x \geq (st + x_r), \end{cases} \quad (6.5)$$

where $s = u_m + u_r$. Between the constant states u_l and u_m , for $t > 0$ an expansion fan is present. We take the numerical values $u_l = -0.5$, $u_m = 0.5$, $u_r = 0$, $x_l = 0.3$ and $x_r = 0.6$. The analytical and numerical solution, computed on the grid of Figure 6.1, at $t = 0.2$ are depicted in Figure 6.3. We observe that the expansion fan is smeared whereas the shock is captured in a few cells at the correct location. The overshoot right before the shock increases, as was also observed in Section 6.1.1,

for increasing Courant numbers. The change in sign of u in the rarefaction wave does not introduce a glitch as for example the Godunov scheme does, see Figure 5.15 of [104]. The contourlines of u are, as they should, vertically aligned to a satisfactory degree.

Even though it is not desirable that monotonicity is lost in the computation of solutions to a nonlinear scalar conservation law, this does *not* imply that wiggles will occur in results obtained for the system case, e.g. for the Euler equations. To investigate this, we have applied our scheme to various so-called Riemann problems, and this will be discussed in the next section.

6.2 Riemann problems

Numerical schemes applied to the Euler equations should converge to the genuine weak solution, often referred to as the entropy solution. This solution represents the vanishing viscosity solution of the physical problem. As far as we know, only for the Osher scheme [28, 83, 84] convergence to the correct weak solution has been proven, provided that the numerical solution converges. Also for our scheme no convergence proof is available, hence we validate it by means of numerical experiments. For an important kind of test case for compressible flow methods, namely the Riemann problem, we will compare the numerical solution resulting from the unstructured staggered method with the exact solution and with solutions computed with some examples of standard compressible flow schemes. The Riemann problem consists of solving the 1D Euler equations for initial conditions of the following form:

$$U(x) = \begin{cases} U_l & \text{if } x < x_0; \\ U_r & \text{if } x > x_0. \end{cases} \quad (6.6)$$

In other words, the initial state vector $U(x) = [u(x), p(x), \rho(x)]$ consists of two constant states $U_l = [u_l, p_l, \rho_l]$ and $U_r = [u_r, p_r, \rho_r]$, separated by an initial discontinuity located at x_0 . For all test cases we used $x_0 = 0.5$ the grid shown in Figure 6.1, which is bounded by $x = 0$ and $x = 1$. Exact solutions for Riemann problems can be determined using solution procedures that are discussed in several textbooks, see for example Chapter 4 of [104] for a thorough discussion. Note that the staggered scheme, because of the arbitrary directions of the normal vectors, considers these 1D Riemann problems as truly 2D flows. The Courant number, defined in (A.30) and with the maximum signal speed taken from the analytical solution, is, unless stated otherwise, put to 1.0.

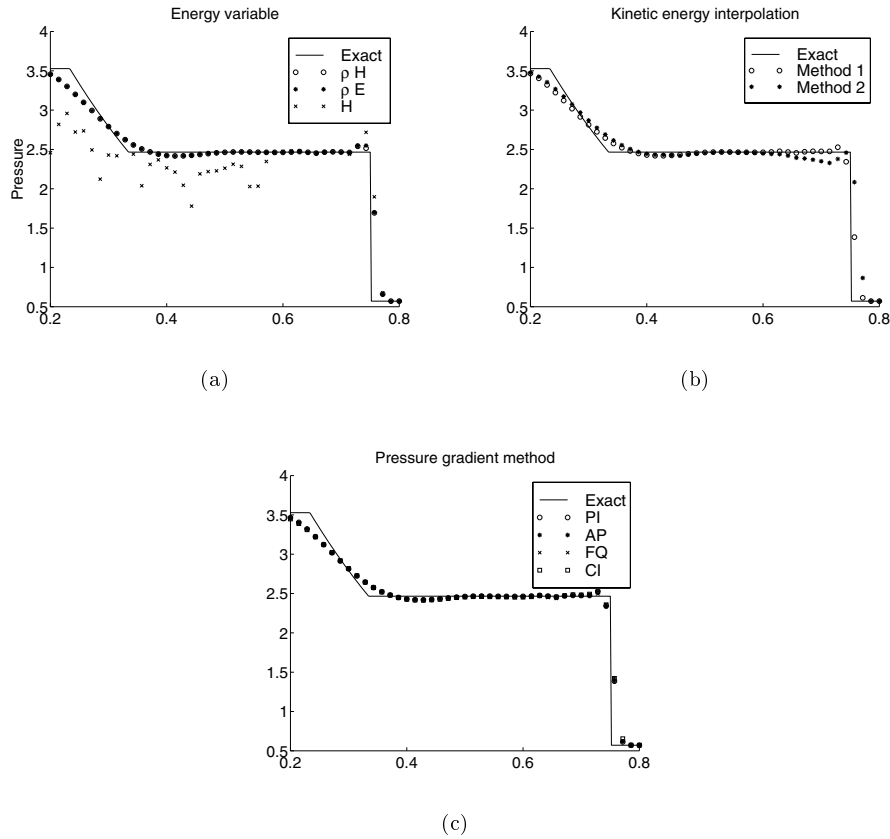


Figure 6.4: Numerical solutions to the test case of Lax at $t = 0.1$, computed with various primary energy variables (a), methods to compute the kinetic energy (b) and pressure gradient schemes (c).

6.2.1 Choices with respect to energy variable and interpolation methods

Three choices concerning the discretization are studied by means of comparing numerical solutions to the test case of Lax, to be specified in Section 6.2.3.

Choice of primary energy variable in the FC approach

In the FC (fully compressible flow) approach, four distinct primary energy variables, namely ρH , ρE , H and h , can be selected. The steady state solution does not depend on which variable is used, but unsteady problems or the route to steady state are affected by this choice. It turns out that results obtained with the several variables for a Courant number equal to 1 are hardly distinguishable. Results for

$\sigma = 2.5$ are shown in Figure 6.4a. The results using H as primary variable are unacceptable, and using h leads even to unphysical results in the form of negative densities. The solutions obtained with ρH and ρE are hardly distinguishable. Since the energy equation with ρH as primary variable is somewhat more straightforward to discretize and easier to implement, we prefer ρH as primary energy variable in the FC approach. In the MU (Mach-uniform) approach such a freedom of choice of primary variables does not exist.

Choice of method to compute kinetic energy

In Section 4.6.2 four different ways to evaluate the kinetic energy in cell centers are introduced. Only the first two methods are actually implemented, and their accuracy is studied here. In order to keep the stencil for the Mach-uniform pressure-correction equation relatively small, only method 1 (the least squares approach) is suitable for the MU approach. For the FC approach, both methods can be applied equally well. Results depicted in Figure 6.4b, with $\sigma = 2.2$, show that method 1 is slightly more accurate than method 2 (linear interpolation of the reconstructed momentum vector). Hence, from now on method 1 is used.

Choice of pressure gradient scheme

In Sections 5.1 and 5.2, among others the accuracy of four schemes to compute the pressure gradient in incompressible flows is investigated. It turned out that it is best to use the path integral (PI) method. For the test problem discussed in this section, these four schemes yield, as can be derived from Figure 6.4c, almost identical results. Also on grids that are made deliberately even more rough, we found out that this remains the case. Note that even though the PI method is not conservative, see the discussion in Appendix D.2, this apparently does not lead to wrong shock speeds. All results shown from now on are computed using the PI method, since in incompressible flows it is the method of choice and in compressible flows it is as accurate as the other proposed schemes.

6.2.2 Sod's shocktube problem

The Riemann problem is called a shocktube problem when $u_l = u_r = 0$. Such a problem can be realized experimentally by removing an impermeable membrane in an infinitely long 1D tube where gases at both sides are initially at rest with different pressure and density. The dimensionless initial conditions for Sod's shocktube problem [100] are:

$$U_l = [u_l, p_l, \rho_l] = [0, 1, 1], \quad U_r = [u_r, p_r, \rho_r] = [0, 0.1, 0.125]. \quad (6.7)$$

With $\max(u + a) = 2.19$ for the exact solution and putting $\sigma = 1$ this leads to $\Delta t = 2.0 \times 10^{-3}$. Numerical solutions of the density resulting from the FC and MU approach are together with the analytical solution shown in Figure 6.5. We have also included results of the Roe [92] and AUSM [70] scheme, obtained on a 1D equidistant

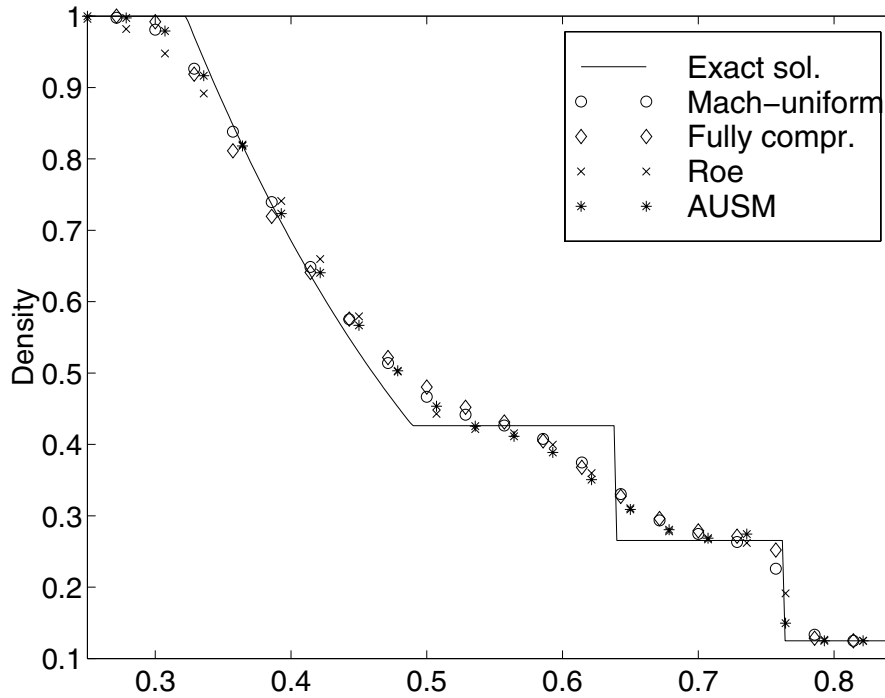


Figure 6.5: Sod's shocktube problem. Comparison of numerical results with the exact solution for the density at $t = 0.15$. For clarity, only half of the solution points are shown.

grid with the same number (70) of grid points as we have on the centerline of the 2D unstructured grid. We see that our schemes converge to the genuine weak solution and that spurious wiggles are absent. The contact discontinuity is, as is common for first order upwind methods, smeared. This is due to the linear character of the contact discontinuity: characteristics on either side of the discontinuity run parallel to it. Since characteristics converge into shocks, this produces a steepening effect resulting in a crisp shock resolution. We observe that both approaches on 2D unstructured staggered grids have similar accuracy as the well-established collocated Roe and AUSM scheme on 1D equidistant grids. Due to the two-dimensional nature of our scheme, the maximum value of the velocity in the y -direction is not equal to zero, but remains everywhere (in absolute value) smaller than 0.02. We consider this value acceptable, since it is only two percent of the maximum value of the velocity in the x -direction. The contourlines of the several variables are, as they should, vertically aligned; see also Figure D.3a and the accompanying discussion in Appendix D.3.

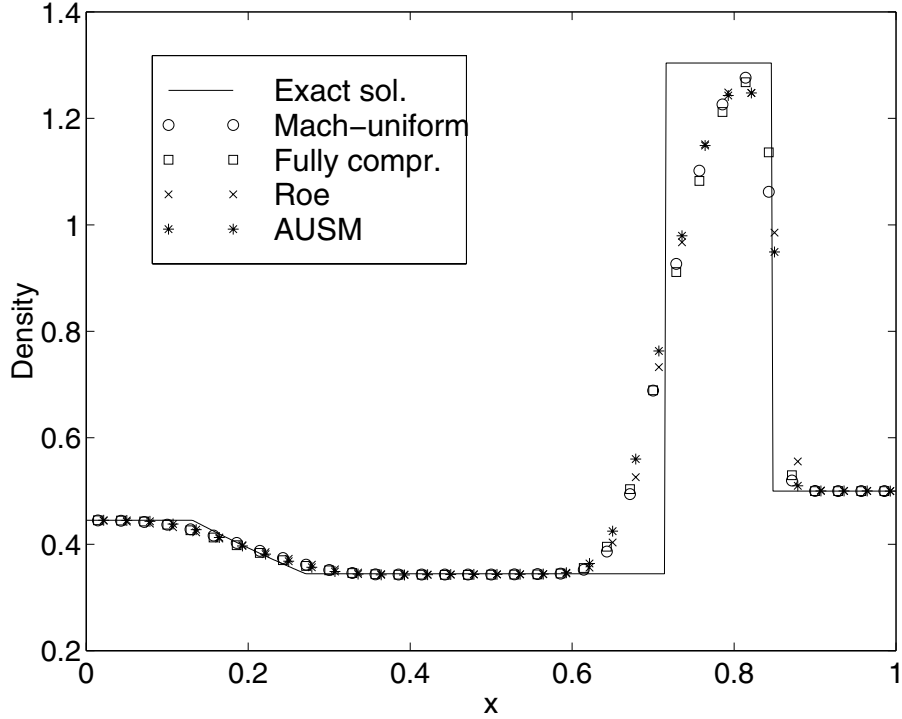


Figure 6.6: Test case of Lax. Comparison of numerical results with the exact solution for the density at $t = 0.14$. For clarity, only half of the solution points are shown.

6.2.3 Test case of Lax

Frequently used is the test case of Lax [67], for which the dimensionless initial state is specified by

$$U_l = [u_l, p_l, \rho_l] = [0.698, 3.528, 0.445], \quad U_r = [u_r, p_r, \rho_r] = [0, 0.571, 0.5]. \quad (6.8)$$

Using the exact solution, the maximum wave speed turns out to be $\max(u+a) = 4.69$, leading to $\Delta t = 1.0 \times 10^{-3}$ for $\sigma = 1$. The main difficulties in computing solutions to this test case are the strong contact discontinuity and shock, which are also relatively close to each other. Results from the FC and MU approach, the Roe and AUSM scheme are, together with the analytical solution, shown in Figure 6.6. We see that our schemes converge to the correct weak solution. As already observed in the results for Sod's shocktube problem, the contact discontinuity is smeared while the shock resolution is crisp. Due to the use of relatively few grid points and the small distance between the contact discontinuity and the shock, the density does not reach its maximum value. Note that the staggered grid approach has similar accuracy as the well-established Roe and AUSM scheme.

6.2.4 Mach 3 test case

In the preceding two test cases, the flow remained subsonic. Supersonic flow may

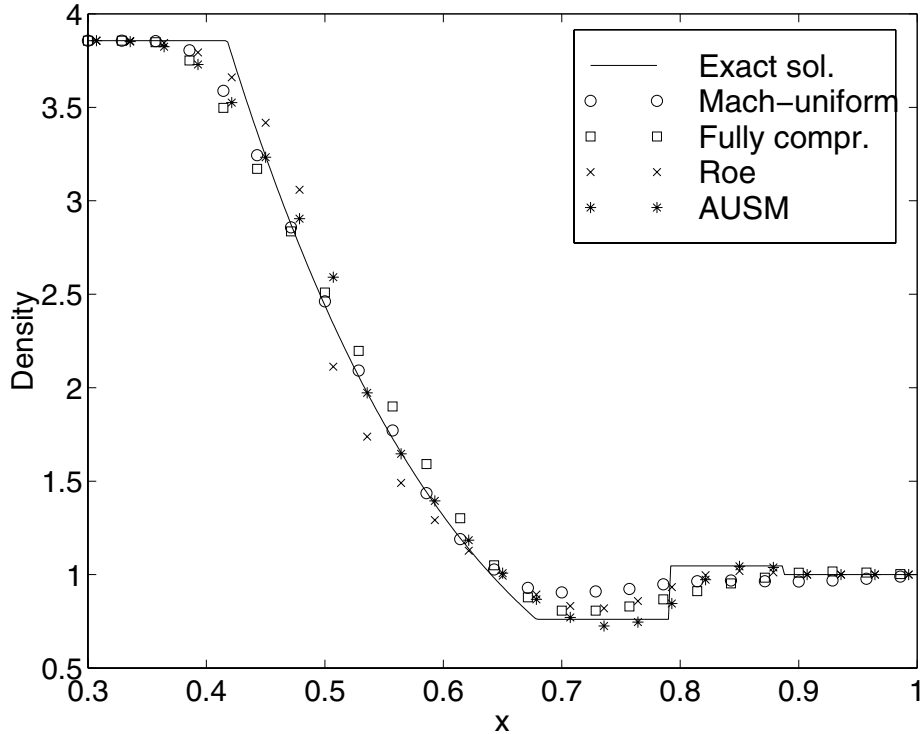


Figure 6.7: Mach 3 test case. Comparison of numerical results with the exact solution for the density at $t = 0.081$. For clarity, only half of the solution points are shown.

bring additional numerical difficulties. In the supersonic flow problem posed by Arora and Roe [3], which we will call the Mach 3 test case, the Mach number rises to a value of approximately 3.0. For this problem the Roe scheme violates the entropy condition by replacing the expansion fan with an expansion shock in the sonic point. A sonic entropy fix, as introduced for example by Harten [45], is necessary to cure this problem, but a sonic glitch remains. Also results from the Godunov, Van Leer, AUSM and Osher schemes give rise to sonic glitches at sonic points, see for example Figures 6.8, 8.4, 8.5 and 12.9 in [104]. The dimensionless initial left and right states for the Mach 3 test case are defined by:

$$U_l = [u_l, p_l, \rho_l] = [0.92, 10.333, 3.857], \quad U_r = [u_r, p_r, \rho_r] = [3.55, 1, 1]. \quad (6.9)$$

The maximum wave speed in the flow, using the exact solution, is $\max(u+a) = 5.00$, resulting in $\Delta t = 9.0 \times 10^{-4}$ for $\sigma = 1$. The exact and numerical solution at $t = 0.081$

are shown in Figure 6.7. For Roe's scheme, Harten's sonic entropy fix with $\epsilon = 0.5$, was employed; this results in a clearly visible sonic glitch at $x \approx 0.5$. Also AUSM give rise to a, somewhat smaller, sonic glitch at this location. Note that, again, the staggered schemes converge to the entropy solution, but without a sonic glitch. The flow is dominated by a huge expansion fan followed by a contact discontinuity and a small shock which, as one observes, are hard to capture accurately using a first order scheme. Of the four considered schemes, AUSM seems to capture the contact discontinuity best. The Roe scheme and the staggered FC approach have similar accuracy concerning resolution of the contact discontinuity, while the MU approach appears to be slightly more diffusive.

6.2.5 Riemann problems on a less smooth grid

The Riemann problems discussed in the three previous sections have been recomputed on the grid depicted in Figure 6.8a. The aim is to study the accuracy on a grid that is made irregular deliberately. The MU formulation is used for this purpose. The number of grid points in the x -direction is approximately 100, and the Courant number, based on the minimal meshwidth, is put to 1.0 for all examples. For comparison, results computed on a Courant grid with 100 grid points in the x -direction and triangular cells with angles equal to 90° and 45° are included. In Figure 6.8b the numerical and analytical solutions for the density are shown. We observe that the quality of the numerical solutions does not degrade when the grid is distorted. This is, of course, a strong feature of the present method.

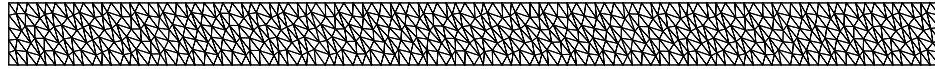
6.2.6 Stationary contact discontinuity and slip flows

As pointed out by Van Leer [118], a significant defect of the van Leer scheme is the fact that stationary or slowly moving contact discontinuities are not resolved satisfactorily: they are approximated by a smooth profile that diffuses as time progresses. We wonder whether the staggered scheme has this defect. The initial condition for a stationary contact discontinuity is given by: $u_l = u_r = 0$, $\rho_l \neq \rho_r$ and $p_l = p_r$. It is trivial to see that, since the discretization of the convection terms in the staggered scheme depends on the velocity, the stationary discontinuity remains unaffected in the computation and does not smear. This is also observed in numerical experiments.

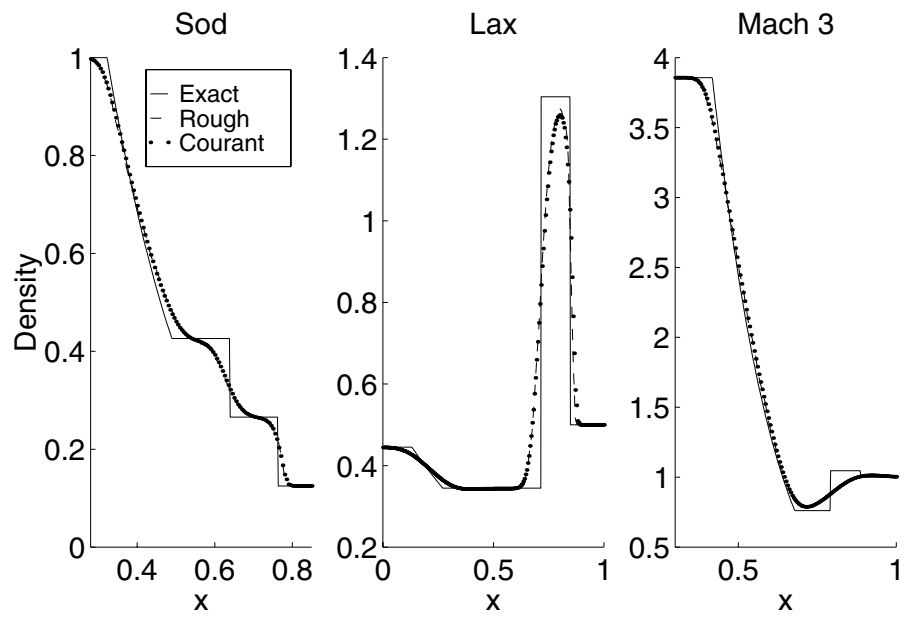
Now we turn to study the crosswind diffusion for a slip flow. States 1 and 2, indicated in Figure 6.9a, are separated at $y = y_0$, and are given by:

$$[u_1, p_1, \rho_1] = [2\sqrt{2}, 1, 1/2], \quad [u_2, p_2, \rho_2] = [0.4, 1, 1], \quad (6.10)$$

where u is the horizontal velocity component, and the horizontal velocity component is zero. The Mach numbers in states 1 and 2 equal $M_1 = 0.8$ and $M_2 = 0.4$. On the grid that is shown in Figure 6.9b, the solution is computed using the exact solution



(a)



(b)

Figure 6.8: Grid (a) and the solution for the density (b).

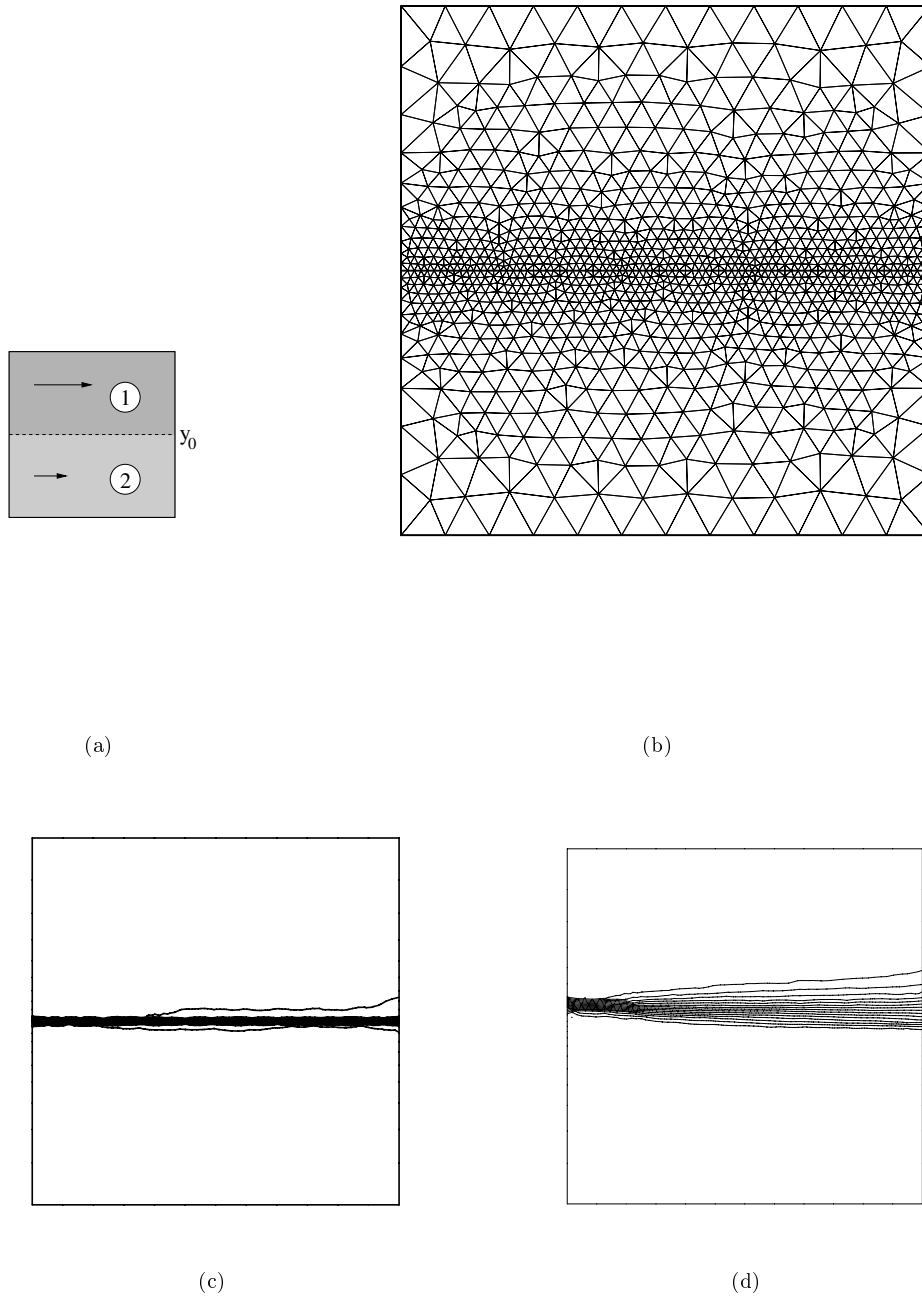


Figure 6.9: Specification of the domain (a) and the grid used (b) for the study of slip flow. Mach number isolines for $y_0 = 0.5$ (c) and $y_0 = 0.55$ (d).

as initial condition. Note that the horizontal bold line in the grid is located at $y = 0.5$. The resulting Mach number isolines of the computed stationary solution for $y_0 = 0.5$ and $y_0 = 0.55$ are given in Figures 6.9c and 6.9d. We observe that the solution is more smeared when the initial discontinuity is not aligned with a grid line, and that the resolution is comparable to that of other schemes, see [62].

6.3 Flow in a channel with a bump

Flow in a channel with a bump was chosen to evaluate the code for computation of inviscid steady state solutions. In Section 6.3.1, special attention is given to the inertia term in incompressible flow. Examples of subsonic (including a grid refinement study) and supersonic flow are addressed in Sections 6.3.1 and 6.3.2.

6.3.1 Incompressible flow

We consider the incompressible, inviscid flow in a channel with a 10% circular arc bump. The aim is to demonstrate clearly the differences between the first order upwind and central difference scheme for the momentum equation. The coarse grids (on fine grids the differences would, of course, be much smaller) used for this purpose are denoted by ‘rectangular’ and ‘general’, see Figures 6.10a and 6.10b. At the upper and lower walls the freeslip condition is applied, at the inflow boundary (on the left) the velocity is prescribed and at the outflow boundary the pressure is given. Freestream flow is taken as initial condition, and a relative accuracy equal to 10^{-2} is used to compute the stationary solution. From the isobar patterns in Figures 6.10c–f one deduces that the central scheme, as opposed to the first order upwind scheme, hardly introduces artificial diffusion. As discussed in Appendix C, it is desirable that the inertia term $C = C(\mathbf{u})$ is discretized in a symmetry-preserving fashion, i.e. $C = -C^T$. Hence, it is desirable that the eigenvalues of C are imaginary, or, if that is not the case, have at least a positive real part in order to avoid instability. In Figure 6.10g the eigenvalues of $C(\hat{\mathbf{u}})$, for central and first order differences, on the rectangular and general grid, are depicted. Here, $\hat{\mathbf{u}}$ represents the stationary solution. The eigenvalues have a non-negative real part, hence the numerical solution is stable; this is confirmed by letting the computation run for a long time.

The eigenvalues corresponding to the first order upwind discretization have a large real part, expressing its diffusive character. The eigenvalues on the rectangular grid form three clusters: the zero eigenvalues correspond to boundary faces with Dirichlet condition for the velocity, and the eigenvalues with a real part roughly equal to 0.125 and 0.25 originate from the vertical and oblique respectively the horizontal faces. This can be explained by noting that on a Courant grid there exists in the upwind discretization only a nonzero coupling between faces with the same orientation. A suitable rearrangement of the numbering of the variables then results in an inertia

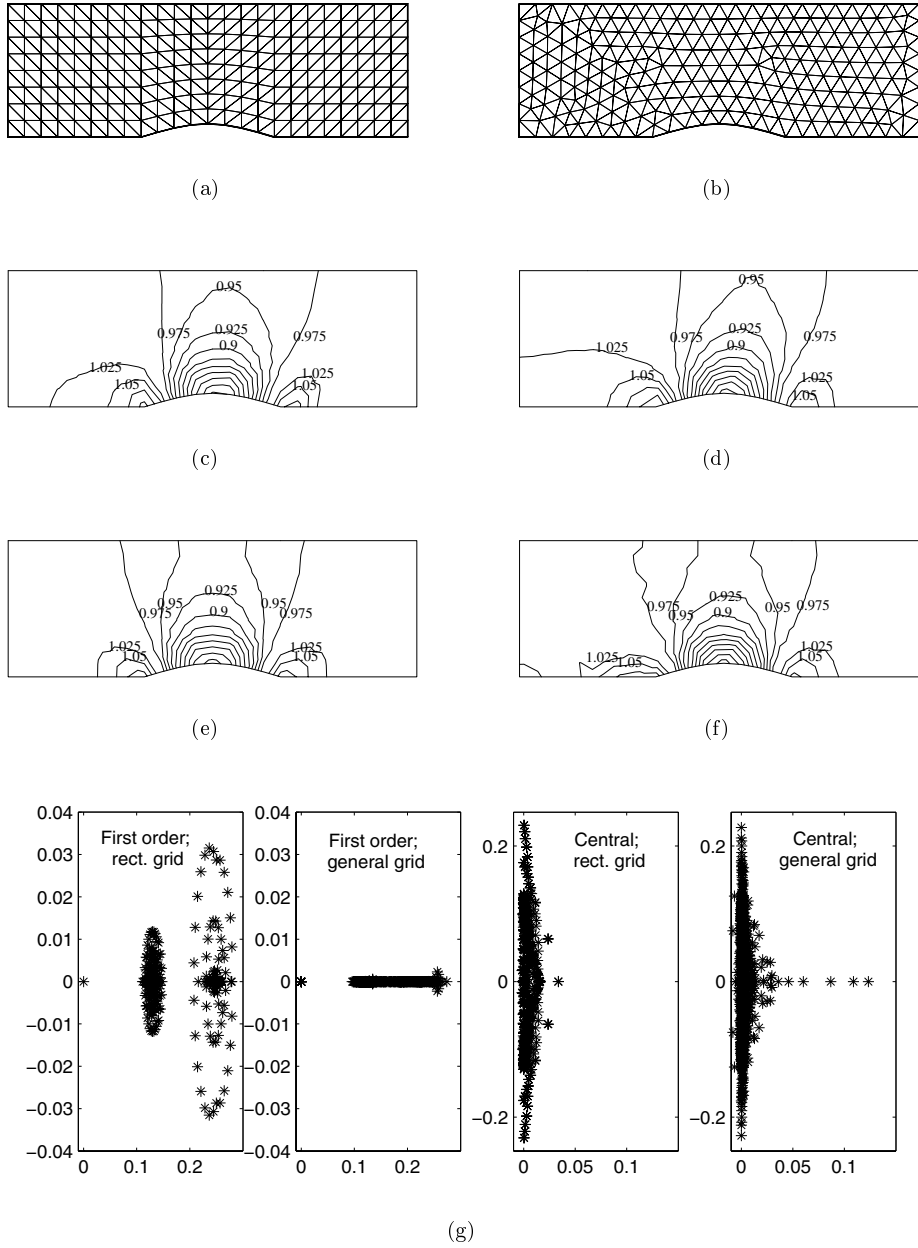


Figure 6.10: Rectangular (a) and general (b) grid in a channel with 10% bump, used for the incompressible computations. Isobars obtained using first order upwind discretization on the rectangular (c) and general (d) grid, and isobars obtained using central discretization on the rectangular (e) and general (f) grid. Eigenvalues of the inertia operator in the complex plane (g).

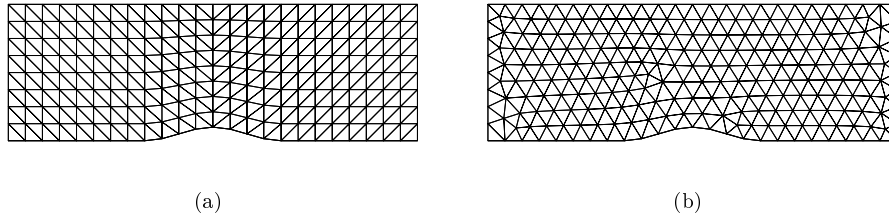


Figure 6.11: Example of grids of type r (a) and g (b), with $n = 8$, for subsonic flow over a sinusoidal bump.

matrix consisting of three lower triangular subblocks, corresponding to the vertical, oblique and horizontal faces. The eigenvalues of these subblocks are easily found to be ul , ul and $2ul$, respectively, with u the velocity and l the size of the horizontal and vertical faces. The flow computed on the rectangular grid in Figure 6.10a is relatively close to uniform flow computed on a Courant grid with $u = 1$ and $l = 0.125$, and this closes the explanation.

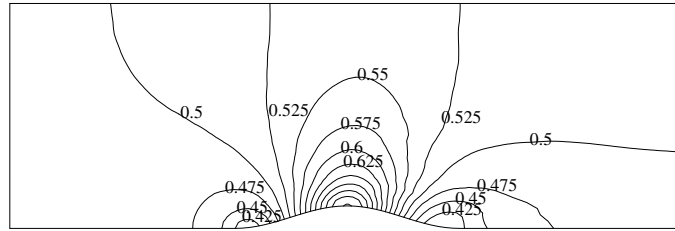
As was to be expected, for the central scheme the imaginary part of the eigenvalues is much larger than the real part. The eigenvalues with real part larger than 0.05 are due to the one-sided differences needed to deal with the outflow boundary. Since there is not much difference between the eigenvalues obtained on the general and rectangular grid, it can be concluded that the central scheme has satisfactory properties on both types of grids.

Subsonic flow: grid refinement

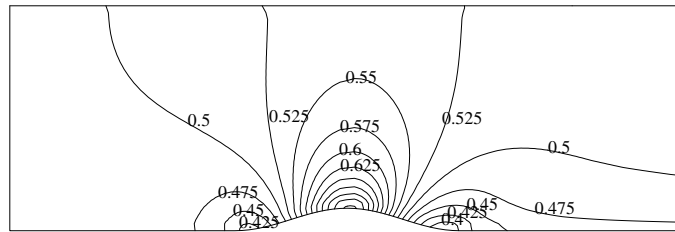
Our purpose is to show that the spatial accuracy of our unstructured upwind scheme is first order, and that the accuracy is relatively insensitive to the shape of the triangles. The curve describing the lower wall of the domain is differentiable; the singularities present in the geometry of a circular arc bump (which is used as a test case in, for example, [9, 22, 77]) may lower the order of the scheme significantly. With the corners of the domain located at $(\pm 1.5, 0)$ and $(\pm 1.5, 1)$, the lower wall is defined by:

$$y = \begin{cases} 0 & \text{if } x \in [-1.5, -0.5) \cup \langle 0.5, 1.5]; \\ 0.05(1 + \cos(2\pi x)) & \text{if } x \in \langle -0.5, 0.5 \rangle. \end{cases} \quad (6.11)$$

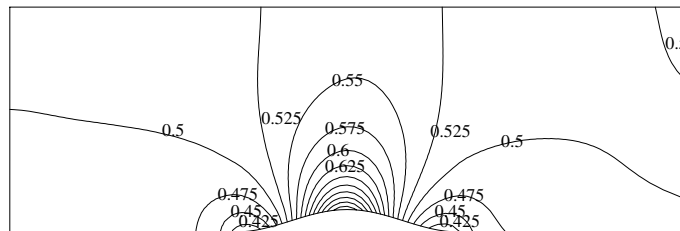
A grid refinement study on structured (quadrilateral cells) and unstructured (rectangular and general cells) grids, labeled s , r and g respectively, has been performed. The structured grid results are computed by means of the Mach-uniform pressure-correction scheme of Bijl and Wesseling [9]. The boundaries of the channel, having



(a)



(b)



(c)

Figure 6.12: The Mach isolines of subsonic flow over a bump, computed with $n = 32$ on grids of type g (a), r (b) and s (c).

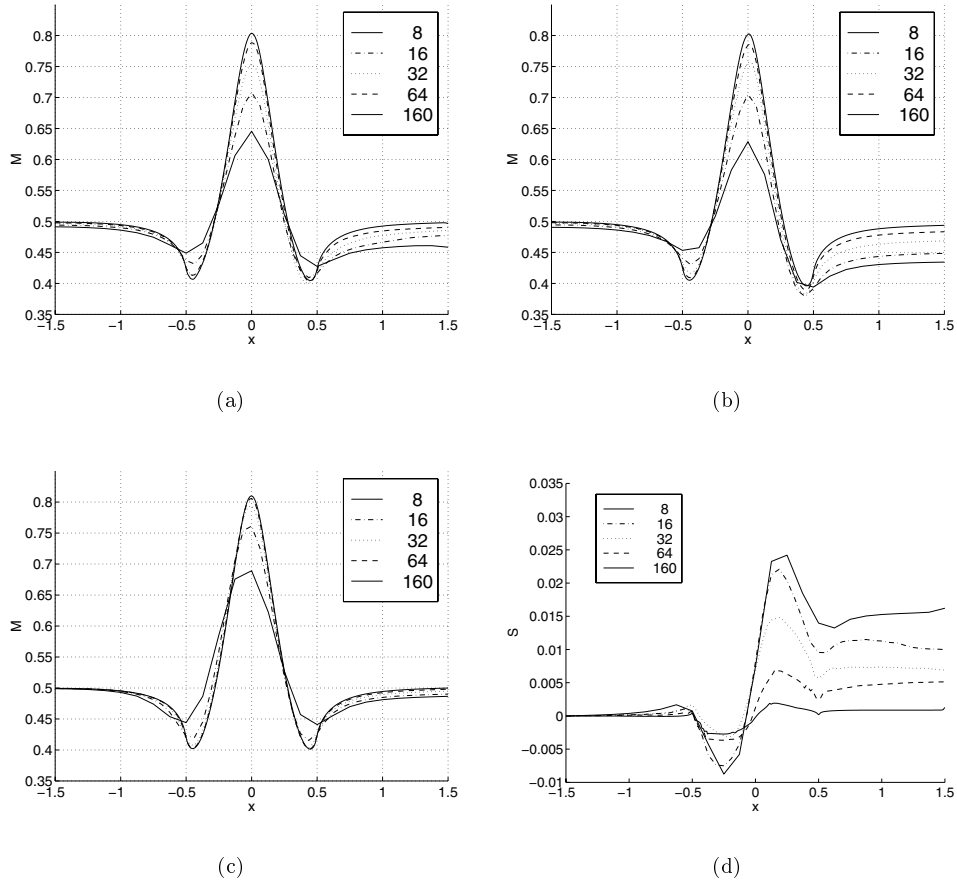


Figure 6.13: The Mach number at the lower wall at grids of type g (a), r (b) and s (c), and the entropy at the lower wall at grids of type g (d), for various values of n .

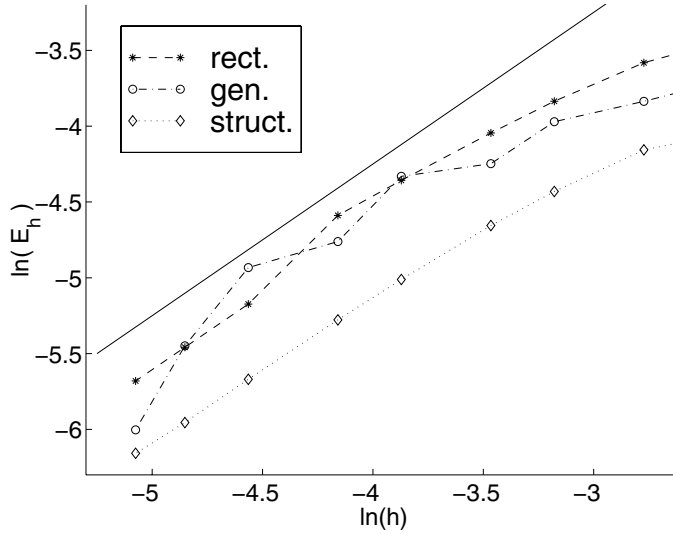


Figure 6.14: The natural logarithm of the error E_h versus the meshwidth h , for the three types of grids considered. The continuous line is an auxiliary line with slope 1.

length 3 and height 1, are divided into $3n \times n$ nodal points, with n an integer. In Figure 6.11 the unstructured grids with $n = 8$ are shown; the structured grid is the same as grid (a), but with the oblique faces omitted. At the inflow boundary (on the left) the Mach number is put to 0.5. The stationary solution is computed, using a relative accuracy equal to 10^{-2} , at grids with $n = 8, 16, 24, 32, 48, 64, 96, 128, 160$.

Differences between the steady state solutions obtained with the FC and the MU approach are, as they should be, negligible. In all computations, the converged value of the total enthalpy is close to the exact value $H = 1.05$, with variations smaller than 2×10^{-4} . The Mach number isolines of the converged solutions obtained with the first order upwind scheme on grids with $n = 32$ are depicted in Figure 6.12. Note that the solutions on both unstructured grids are virtually the same, while the solution on the structured grid shows some differences near the top of the bump and with respect to the shape of the contourline $M = 0.5$. For $n \rightarrow \infty$, the solutions become indistinguishable. The Mach number and entropy at the lower boundary are displayed in Figure 6.13. For inviscid subsonic flow the solution should be symmetric with respect to the symmetry axis of the problem. The asymmetry, visible in Figures 6.12 and 6.13, is due to the numerical diffusion introduced by the first order upwind scheme. This asymmetry becomes, as one observes, less pronounced with finer grids. From Figure 6.13c we deduce that the the spatial discretization used on the structured grid introduces less numerical diffusion than the spatial discretization

on the unstructured grid. This can be attributed to the fact that the gridlines and the stencil used in the structured grid are aligned with or perpendicular to the flow, while this is not the case for the unstructured grid.

In inviscid subsonic flows, the entropy $S = \ln(p/\rho^\gamma)$ is a constant, so that variations in the computed entropy can be regarded as a measure for the numerical error. This naturally leads to the following way to quantify the error of the numerical scheme:

$$E_h = \int_{\Gamma} |S_h| d\Gamma, \quad (6.12)$$

where S_h is the computed entropy along the lower boundary Γ at a grid with mesh-width h ; we choose $h = 1/n$. When h is small, the error is expected to behave as:

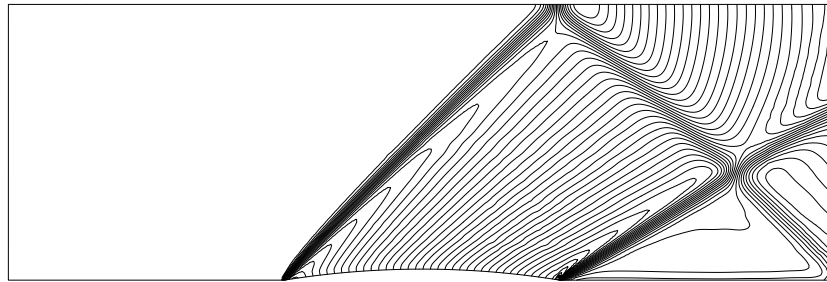
$$E_h = Kh^p, \quad (6.13)$$

where p is the order of the numerical method, and K is a constant. Plotting $\ln(E_h)$ versus $\ln(h)$ gives, see Figure 6.14, for h small a slope roughly equal to $p = 1$. Hence, the first order upwind scheme is, as was to be expected, first order accurate in space. In addition, we note again that on both unstructured grids roughly the same accuracy is obtained, while the error for the structured grid scheme is smaller for a given h .

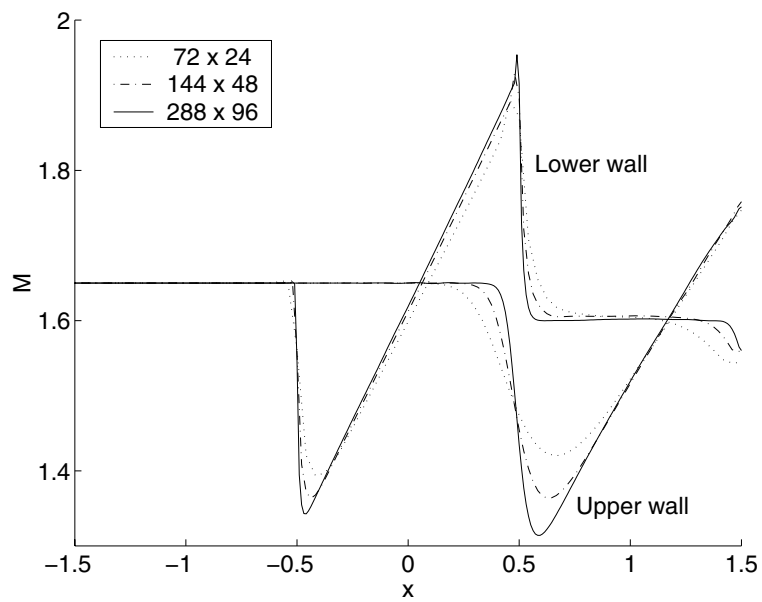
6.3.2 Supersonic flow

Another frequently encountered test case is that of supersonic flow in a channel with 4% circular arc bump and inlet Mach number equal to $M_\infty = 1.65$. At the supersonic inlet, in addition to the momentum and enthalpy, the pressure is prescribed, while at the supersonic outlet no boundary condition is given. Only the MU approach can take the supersonic inflow boundary conditions into account in an appropriate fashion: the way in which a given boundary pressure enters the FC approach is through computation of the pressure gradient, but this quantity needs not be computed when the momentum is already given at the considered boundary, i.e. the inflow boundary. The relative accuracy of all primary variables, see equation (3.1), is put to 10^{-2} .

Three grids similar to the one depicted in Figure 6.10b have been generated, with 72×24 , 144×48 and 288×96 nodes at the domain boundaries. The results illustrated in Figure 6.15 are in excellent agreement with previously published results in, for example, [9, 22, 77].



(a)



(b)

Figure 6.15: Mach number contours for the unstructured grid with 288×96 nodes at the domain boundaries (a), and the Mach number along the upper and lower wall (b).

6.4 Flows around the NACA 0012 airfoil

Flows around airfoils provide excellent test cases for CFD codes, since these flows contain much of the physics involved in aerodynamics while the problem definition, including the geometry, is relatively easy. Four different kinds of inviscid flows,

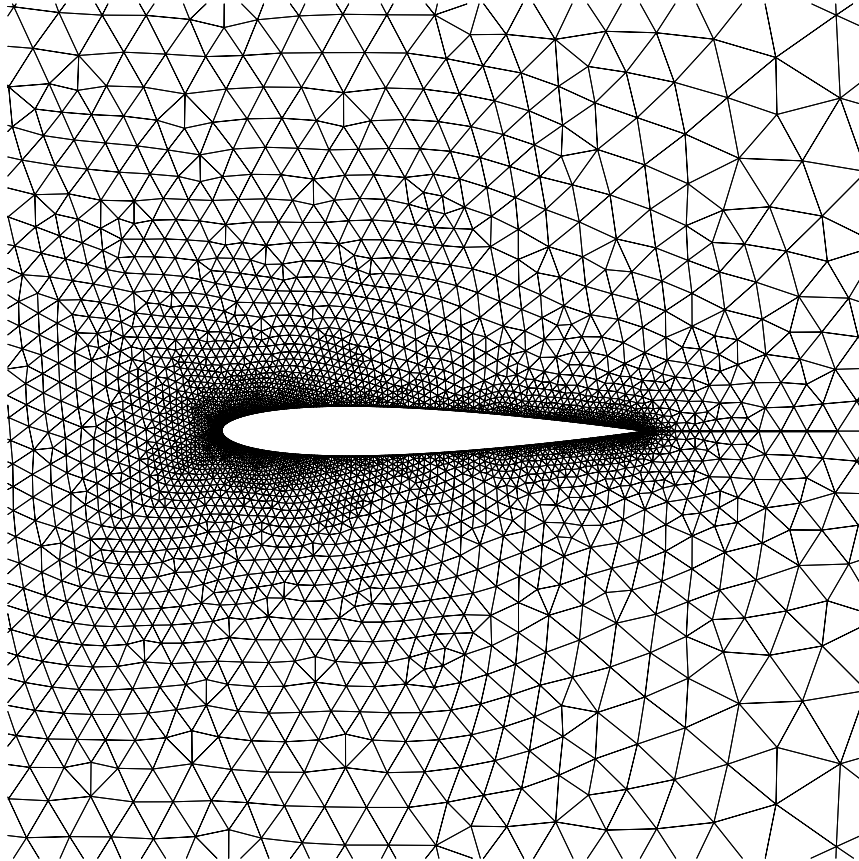


Figure 6.16: Part of the grid that is used to compute flows around the NACA 0012 airfoil.

parameterized by the freestream Mach number M_∞ and angle of incidence α , around the NACA 0012 airfoil have been considered: (i) low subsonic flow with $M_\infty = 0, 10^{-3}, 10^{-2}, 10^{-1}$ and $\alpha = 0^\circ$; (ii) subcritical flow with $M_\infty = 0.63$ and $\alpha = 2^\circ$; (iii) transonic flow with $M_\infty = 0.8$ and $\alpha = 1.25^\circ$, and (iv) supersonic flow with $M_\infty = 1.2$ and $\alpha = 0^\circ$. The accuracy of the results is discussed in Sections 6.4.1, 6.4.2, 6.4.3 and 6.4.4, respectively. The efficiency of the fully compressible flow

(FC) and Mach-uniform (MU) approach forms the subject of Section 6.4.5. Unless stated otherwise, the grid partly depicted in Figure 6.16 is used. With the leading and trailing edge of the airfoil located at coordinates (0,0) and (1,0), the left lower and right upper coordinates of the rectangle defining the considered flow domain are given by (-3,-5) and (5,5). This grid contains 9610 cells, 14612 faces and 5002 vertices, of which 320 are positioned at the airfoil, and the minimal meshwidth equals 1.0×10^{-3} .

6.4.1 Low subsonic flow around the NACA 0012 airfoil

As discussed in Section 3.4.1, the accuracy of standard compressible flow solvers deteriorates with decreasing Mach number. For standard solvers, sometimes the $M_\infty = 0.1$ solution of the flow around an airfoil is even closer to the incompressible one than is the $M_\infty = 10^{-3}$ solution, see [36]. For $\alpha = 0^\circ$ and $M_\infty = 0, 10^{-3}, 10^{-2}, 10^{-1}$, we computed the solution with the MU approach, and for comparison we included the solution obtained with the FC approach for $M_\infty = 10^{-1}$.

Results obtained with the Mach-uniform formulation

For the low subsonic flow computations, we chose $\epsilon = 10^{-2}$ for the relative stationary accuracy and a time step equal to 0.05. The MU algorithm reduces to the incompressible pressure-correction algorithm when $M_r = M_\infty = 0$ is inserted, and this is also confirmed experimentally. The isobars of the incompressible solution are shown in Figure 6.17a. The pressure fluctuations, nondimensionalized as in (3.30), should remain constant for M_r small, see also (3.32). We observe in Figure 6.17 that this is indeed the case: the computed isobars for $M_\infty = 0, 10^{-3}$ and 10^{-2} are virtually identical, while in the $M_\infty = 0.1$ result compressibility effects start to play a nonnegligible role. Hence, the MU approach clearly does not suffer from loss of accuracy in the low Mach number regime.

Results obtained with the fully compressible flow approach

Since the FC approach is density-based, we expect it to perform poorly for low Mach number flow because of the weak pressure-density coupling. From (3.34) and the MU results we find that the dimensionless pressure as used in the FC approach is bounded as follows: $0.9973 < \tilde{p}_s < 1.0066$. This spells accuracy problems, and in order to obtain acceptable results with the FC approach, the relative stationary accuracy ϵ had to be decreased to 10^{-3} . We found that the default stopping criteria of the linear solver, given in Section 3.6.1, were sufficient, because increasing their accuracy did not have any impact on the steady state solution. The computed isobars for $M_\infty = 10^{-1}$, in the same units as the MU results, are shown in Figure 6.17e. Note that for the same parameter the FC result is different from the MU result, compare Figures 6.17d and 6.17e. Apparently, at the airfoil these differences make

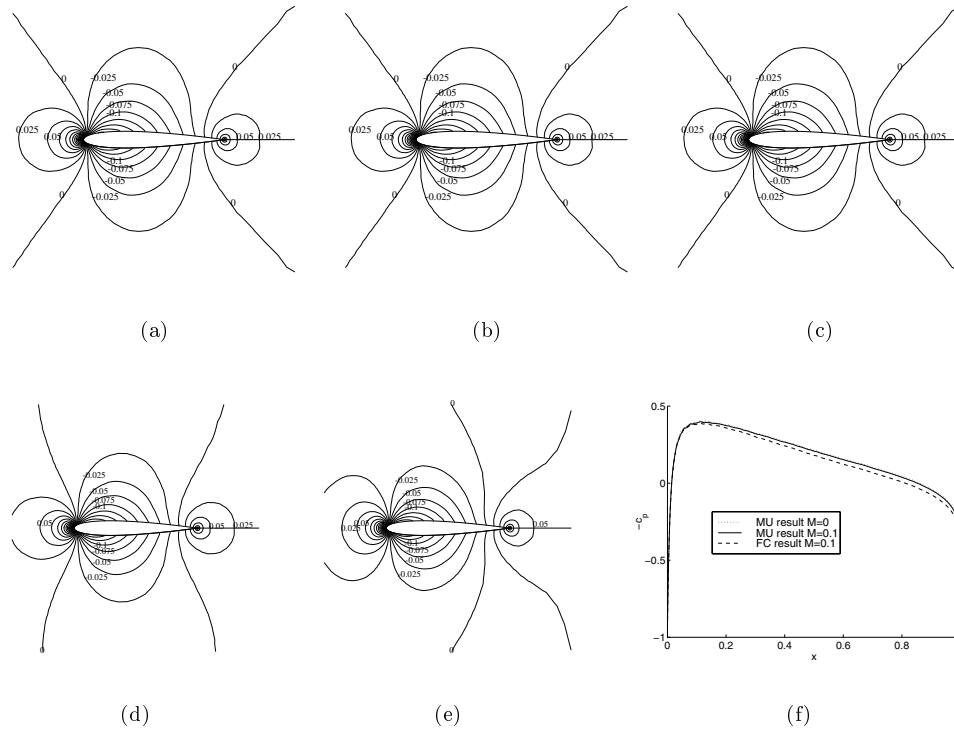


Figure 6.17: Isobars around NACA 0012 profile, computed using the MU approach, for $M_\infty = 0$ (a), $M_\infty = 10^{-3}$ (b), $M_\infty = 10^{-2}$ (c) and $M_\infty = 10^{-1}$ (d). Isobars for $M_\infty = 10^{-1}$ obtained with the FC approach (e). Pressure coefficient for various cases (f).

themselves hardly felt, since the pressure coefficient at the airfoil:

$$c_p = \frac{p - p_\infty}{\frac{1}{2} \rho_\infty u_\infty^2}, \quad (6.14)$$

is almost the same for the FC and MU approach, see Figure 6.17f. Note that the pressure distribution at the airfoil of the incompressible flow solution is still very close to that of the $M_\infty = 10^{-1}$ solution. Hence, the FC approach suffers from some loss of accuracy in the low Mach number regime; the issue of loss of efficiency will be addressed in Section 6.4.5.

6.4.2 Subcritical flow around the NACA 0012 airfoil

The computed isobars and Mach number at the airfoil of subcritical flow around the NACA 0012 airfoil with a freestream Mach number $M_\infty = 0.63$ and an angle of

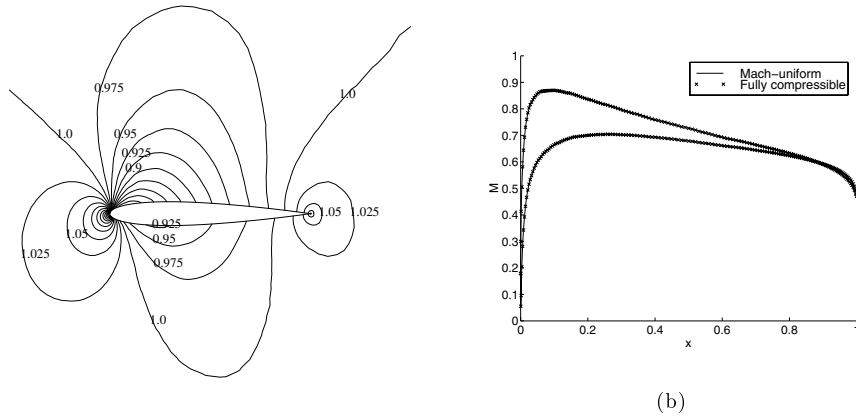


Figure 6.18: Subcritical flow around the NACA 0012 airfoil. Isobars (a) and Mach number at the airfoil (b).

attack $\alpha = 2^\circ$ are illustrated in Figure 6.18. Note that there is, as it should, hardly any difference between the FC and MU solution (differences in the Mach number at the airfoil are smaller than 10^{-5}). The computed values for the lift coefficient, drag coefficient and maximum Mach number in the domain are 0.27, 1.1×10^{-2} and 0.875 respectively, whereas the benchmark solution [23] yields 0.33, 0 and 0.99 for these quantities. The differences are attributed to the fact that we use a simple first order upwind scheme and a relatively coarse grid.

6.4.3 Transonic flow around the NACA 0012 airfoil

For computation of inviscid flow around a NACA 0012 airfoil with a freestream Mach number of $M_\infty = 0.8$ and an angle of attack $\alpha = 1.25^\circ$, the grid shown in Figure 6.16 is too coarse to get an acceptable shock resolution, and consequently these results are not shown here. At a similar but finer grid, with 80,256 cells, 12,1104 faces and 40,848 vertices, of which 1320 are located at the airfoil, we get the solution shown in Figure 6.19. We observe that entropy is generated at the leading edge. This increase in entropy causes boundary-layer-like viscous losses near the airfoil visible in the Mach isolines plot, which are also present for instance in the first order results in Figure 4.1b of [6]. This ‘numerical boundary layer’ turns out to correspond to the region in which the entropy has increased considerably. With respect to the AGARD benchmark solution [138], the shock is shifted towards

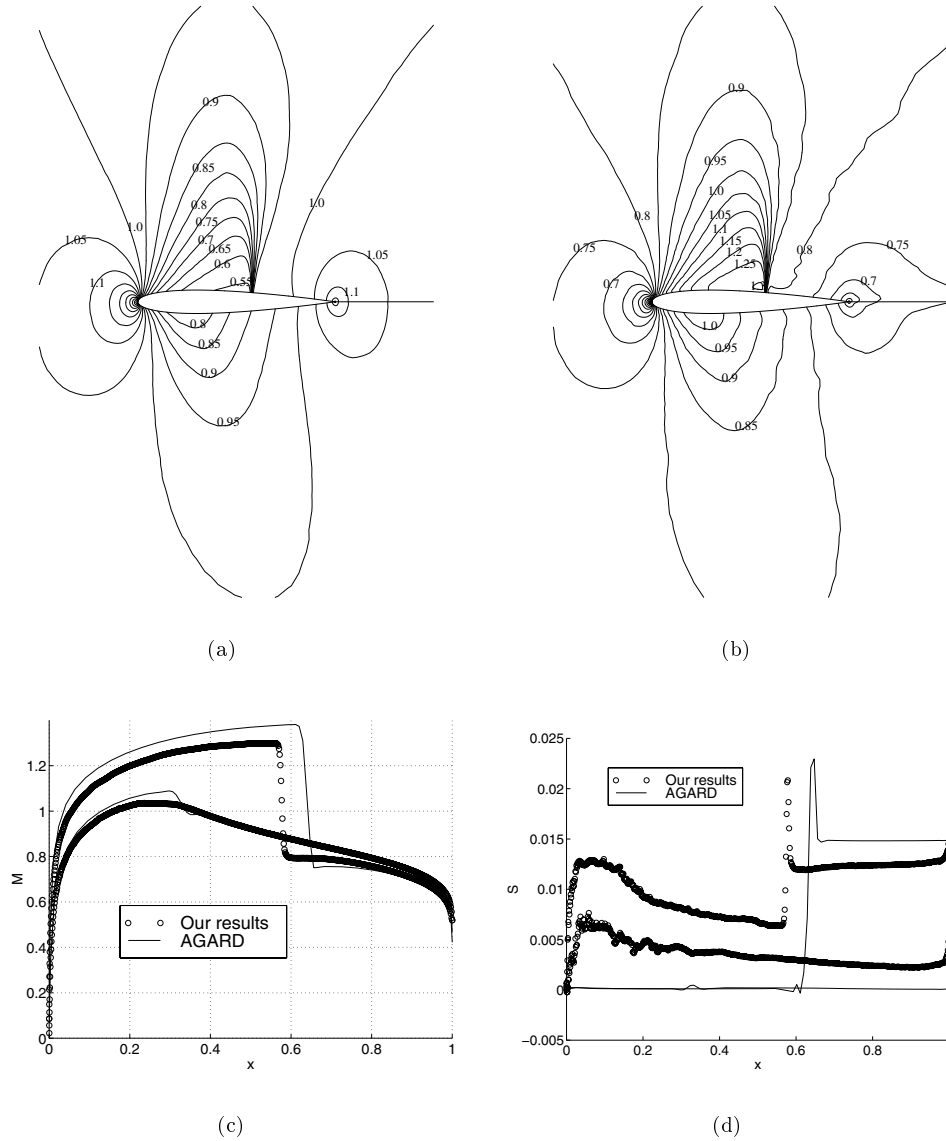


Figure 6.19: Transonic flow around the NACA 0012 airfoil computed on a fine grid. Isobars (a), Mach isolines (b), Mach number (c) and entropy (d) at the airfoil.

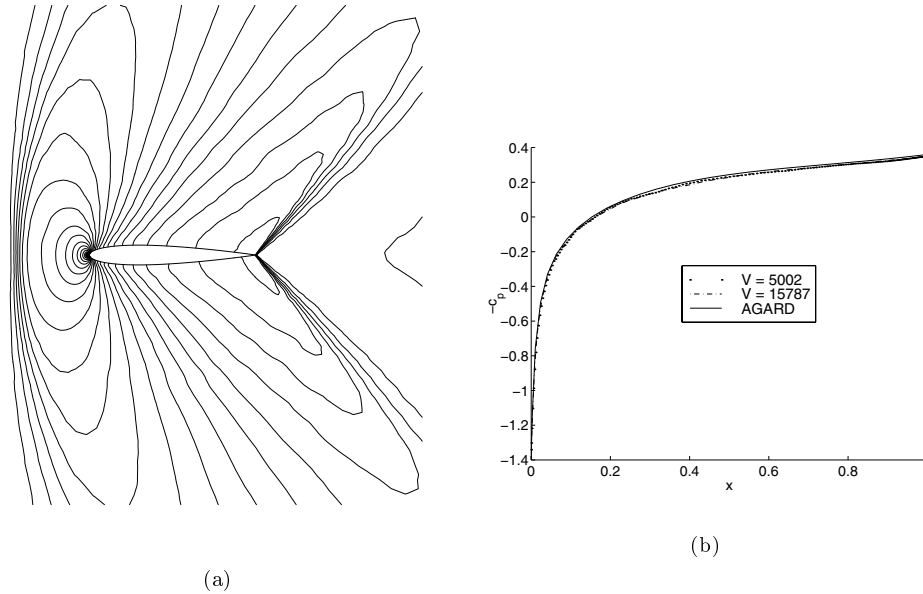


Figure 6.20: Supersonic flow around the NACA 0012 airfoil. Mach isolines (a) and pressure coefficient (b) at the airfoil.

the leading edge. The weak shock at the lower wall is not present in our results. These aspects are attributed to the losses just mentioned, which result effectively in a lowering of the freestream Mach number. Repeating the computation on a larger flow domain does not improve the results. In order to get a closer approximation of the AGARD benchmark solution, higher order methods are indispensable. Nevertheless, we have demonstrated that a staggered discretization on unstructured grids can resolve transonic flows without causing spurious oscillations or other undesired phenomena.

6.4.4 Supersonic flow around the NACA 0012 airfoil

Also for the supersonic flow case, with $M_\infty = 1.2$ and $\alpha = 0^\circ$, around the NACA 0012 airfoil, the grid displayed in Figure 6.16 turns out to be too coarse to get a good resolution of the shocks. We did only perform computations using the MU approach, since the FC approach cannot deal with supersonic flows, see the remarks made in Section 6.3.2. On a grid that is refined in the vicinity of the shocks, with 31,144 cells, 46,931 faces and 15,787 vertices, of which 320 are positioned at the airfoil, we get the results depicted in Figure 6.20. The Mach isolines are virtually identical to

the AGARD benchmark solution [138]. In addition, the pressure distribution at the airfoil can hardly be distinguished from the AGARD result. Furthermore, even the result obtained at the grid of Figure 6.16, with 5002 vertices, turns out to yield a very accurate pressure distribution at the airfoil. So, apparently the lack of shock resolution does not prevent one from having a good agreement on the flow properties at the airfoil. The location of the sonic point upstream of the x -axis (-0.43 in our results and -0.42 in the AGARD solution) and the value for the drag (0.0967 in our results and 0.0960 in the AGARD solution) are also in close correspondence with each other.

6.4.5 Mach uniform efficiency

In the computation of flows around profiles it was verified that the FC and MU approach yield identical steady state solutions, apart from accuracy problems encountered by the FC approach when the Mach number tends to zero or the impossibility of the FC approach to deal with supersonic flows. In addition to the accuracy problem there is, at least for standard flow solvers, the problem of efficiency, i.e. the problem that computation time increases severely for low Mach number flows. In this section we will show that the efficiency of the MU approach is uniform in the Mach number; this in contrast with the FC approach, which has some similarities with the standard flow solvers concerning efficiency. In fact, we will see that the MU approach is more efficient than the FC approach over the whole range of Mach numbers, from low subsonic to supersonic.

Steady state is reached after N_t time steps at $T_{end} = N_t \Delta t$, where one time unit corresponds to the time needed to travel the distance of one chord length at freestream velocity. The relation between the Courant (CFL) number and the time step is, up to a reasonable approximation, given by (in dimensionless units):

$$\sigma = \frac{(u_\infty + \sqrt{h_\infty}/M_r)\Delta t}{\Delta x}, \quad (6.15)$$

where the flow velocity and acoustic speed are based on the freestream values; see also the discussion following equation (A.30). The total number of solver iterations in the process of time stepping from initial to steady state is indicated by N_s , hence the average number of solver iterations per time step equals $n_s = N_s/N_t$. In Table 6.1 results for these quantities are gathered as obtained on the grid displayed in Figure 6.16 during the computations described in Sections 6.4.1 to 6.4.4. The time steps in this table are ‘optimal’, i.e. chosen such that the total computation time T_{CPU} is minimal; this is determined by trial and error. It turns out that then also N_s and N_t are minimal. If the time step is chosen slightly larger than the optimal time step, the transient behavior becomes too strong, and numerical breakdown occurs. Hence, the optimal time step corresponds to the largest possible time step for which convergence is attained. The only exception to this rule is the $M_\infty = 0$

Table 6.1: Results related to efficiency of the airfoil computations. The part left of the double vertical line corresponds to the MU approach, right of it to the FC approach.

M_∞	0	0.1	0.63	0.8	1.2	0.1	0.63	0.8
Δt (10^{-2})	8.0	8.0	9.0	8.0	8.2	0.014	0.1	0.0144
σ	—	800	233	180	150	1.5	2.59	3.25
T_{end}	4.72	2.96	17.2	49.5	52.5	2.02	16.9	28.5
N_t	59	37	191	619	642	14,387	16,910	19,813
T_{CPU} (s)	279	165	668	2021	1645	12,324	13,671	17,112
t_{CPU} (s)	4.73	4.46	3.50	3.26	2.56	0.857	0.808	0.864
N_s	6431	3853	14,554	40,828	30,834	43,305	68,332	80,746
n_s	109	104	76.2	70.0	48.0	3.01	4.05	4.08
n_ρ	3	10	12	10	4	1	2	2
n_m	18	19	22	22	15	1	1	1
$n_{\rho H}$	—	—	—	—	—	1	1	1
n_p	83	75	48	33	22	—	—	—

problem: this one turns out to require less computing time with smaller time steps, e.g. $T_{CPU} = 28$ and $N_t = 6$ for $\Delta t = 0.01$. But we included values belonging to the $\Delta t = 0.08$ solution, so that a better comparison with the $M_\infty = 0.1$ results is achieved. As can be derived from the table, the fact that in the MU approach the pressure is taken implicitly allows for much larger Courant numbers. It is remarkable that, if the Courant number were defined as follows:

$$\tilde{\sigma} = \frac{u_\infty \Delta t}{\Delta x}, \quad (6.16)$$

then the largest allowable value for $\tilde{\sigma}$ remains almost constant over the whole range of Mach numbers $M > 0$ for the MU approach. Inserting $u_\infty = 1$ and $\Delta x = 1.0 \times 10^{-3}$ yields values for $\tilde{\sigma}$ ranging between 80 and 90. The omission of the term $\sqrt{h_\infty}/M_r$ can be made plausible by noting that the pressure, and hence the acoustic waves, are taken into account in an implicit manner in the MU approach, thereby not limiting stability.

It turns out that in the MU approach much fewer time steps are necessary to arrive at steady state, and, even though n_s is larger, the MU approach is much more efficient than the FC approach in terms of computation time. With n_ρ , n_m , $n_{\rho H}$ and n_p we indicate the number of iterations in the linear solver to solve the continuity, momentum, energy and pressure-correction equation, respectively. Recall that one iteration in Bi-CGSTAB, which is used to solve the pressure-correction equation, is more expensive than one GMRES iteration, which is utilized to solve the density,

momentum and energy equation. The reason why $n_s = N_s/N_t$ does not exactly satisfy $n_s = n_\rho + n_m + n_p$ (for the MU approach) or $n_s = n_m + n_\rho + n_{\rho H}$ (for the FC approach), is that the values for n_ρ , n_m , $n_{\rho H}$ and n_p are taken in the final steps of the time-marching procedure, where these values are constant and do not show the large variations that are present in the initial stage. The values for n_p , n_ρ , n_m , $n_{\rho H}$ and n_p are seen to increase for larger values of the time step, which is attributed to the $1/\Delta t$ -behavior of the main diagonal. This is observed in additional experiments, the results of which are not included in Table 6.1. In addition, for a given time step, n_p is seen to decrease with increasing M_r , and this is caused by the M_r^2 term in the main diagonal. Clearly, Table 6.1 shows that T_{CPU} does not blow up as $M_r \downarrow 0$. This is the distinguishing feature of the MU method as compared to methods based on extension of fully compressible flow methods to the weakly compressible case. The time step can be chosen independently of M_r . In the fully compressible flow case the flow takes longer to settle to steady state; this effect would be irrelevant for instationary flows. The computing time per time step depends only weakly on M_r . We may say that the efficiency is uniform in the Mach number.

The total CPU time for a computation consists, apart from the time $T_{i/o}$ needed for initialization and writing output data, of the time T_m needed to construct the linear systems, the time T_s needed to solve these and the time T_{other} for overhead during the computation and, e.g., evaluation of the equation of state. With $T_{i/o}$ negligible when $N_t \gg 1$, and T_{other} much smaller than T_m or T_s , we arrive to a good approximation at:

$$T_{\text{CPU}} = T_m + T_s. \quad (6.17)$$

The time to solve the linear systems encountered in the time stepping procedure is equal to

$$T_s = N_s t_s, \quad (6.18)$$

where t_s is the average CPU time for one iteration in the linear solver. Note that, since both GMRES and Bi-CGSTAB are used, t_s does not correspond to either a GMRES or a Bi-CGSTAB iteration. In a similar fashion, the average time t_m to compute the matrix elements and right-hand sides of the three linear systems in one time step follows from:

$$T_m = N_t t_m. \quad (6.19)$$

Hence,

$$t_{\text{CPU}} = T_{\text{CPU}}/N_t = (N_t t_m + N_s t_s)/N_t = t_m + n_s t_s. \quad (6.20)$$

In Figure 6.21 we have plotted t_{CPU} versus n_s for various Mach numbers and different time steps. From this figure we deduce that expression (6.20) holds to a fairly good extent, with the solid line corresponding to $t_m = 0.86$ and $t_s = 0.035$. Hence, the time needed to build the three matrices (density, momentum and pressure) corresponds to the time needed to do, roughly, $0.86/0.035 \approx 25$ iterations in the linear solver. From Table 6.1 and Figure 6.21 we deduce that, using the optimal time step,

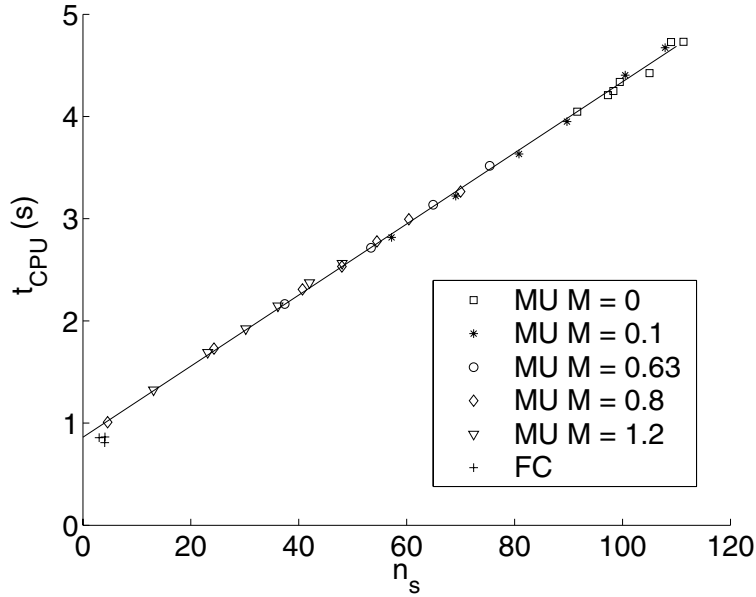


Figure 6.21: CPU time per time step versus n_s for the MU and FC method.

the MU approach spends most of the CPU time in the linear solver, whereas the FC approach spends by far most of its computing time in construction of the linear systems. What does this all imply with respect to Mach-uniform accuracy? For an explicit time integration method, as one usually encounters in compressible codes, obviously no time is spent in the linear solver ($n_s = 0$). This means that the CPU time needed to update the variables can be put in the term T_m . Although for an explicit time integration method the (sparse) matrix needs not be stored explicitly in memory as for our implicit method, each element needs to be computed anyway. Hence, the CPU time required in an explicit flow solver to go from one time level to the next one is approximately the same as the term t_m in (6.20). But an explicit method needs to satisfy a stability criterion of the form $\sigma < C$, with C a constant of the order unity. Especially for small M_r , we saw that (and this is also obvious from expression (6.15)) this limits Δt severely. However, our method does not need to satisfy such a strict stability criterion. Combining the facts that, for the studied case, the CPU time spent in the linear solver is up to four times the CPU time needed to construct the matrices (the ratio $n_s t_s / t_m$), while the factor in the time step that one gains is of the order of several hundreds, it seems that our method is much more efficient than a comparable explicit scheme. In the discussion above we have not addressed the issue of computational efficiency of the flux evaluation. We expect that the present staggered scheme requires significantly less computing time than the schemes based on approximate Riemann solvers, because the numerical

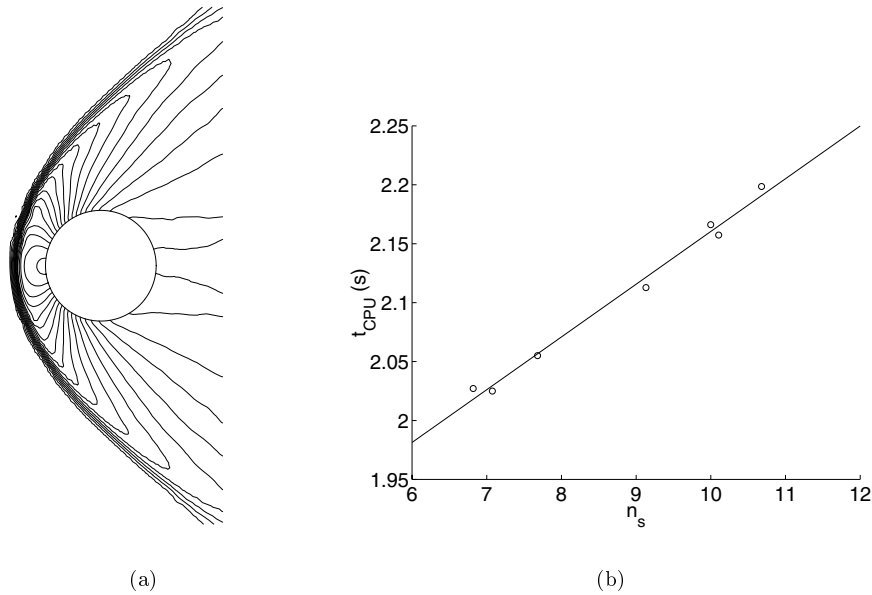


Figure 6.22: Supersonic flow over a circular blunt body. Computed isobars (a) and CPU time and number of solver iterations per time step (b).

fluxes are much simpler.

In the derivation of the Mach-uniform pressure-correction approach, we wonder whether approximation (3.49), necessary to linearize the convection term, should be used in the time derivative of the Mach-uniform pressure-correction equation as well. This, although not necessary for linearization, simplifies implementation of the first line of equation (3.51) since the term $-(\gamma - 1)M_r^2 \mathbf{m}^* \cdot \nabla \delta p / \rho^{n+1}$ needs not be taken into account. We tried so, but found that the maximal Courant number for which the transonic flow test case is stable reduced from 180 (with the mentioned term left in) to 100. With $\sigma = 100$, keeping the mentioned term or omitting it does not affect the CPU time, hence for reasons of stability the term $-(\gamma - 1)M_r^2 \mathbf{m}^* \cdot \nabla \delta p / \rho^{n+1}$ needs is kept in.

6.5 Supersonic flow over a circular blunt body

The 2D supersonic flow over a circular blunt body with $M_\infty = 4.0$ was chosen to demonstrate the ability of the unstructured staggered scheme and the MU approach to correctly resolve the strong bow shock and the acceleration of stagnant flow through a sonic point to supersonic speed. The isobars, computed on a grid

Table 6.2: M_{\max} for the converging-diverging nozzle.

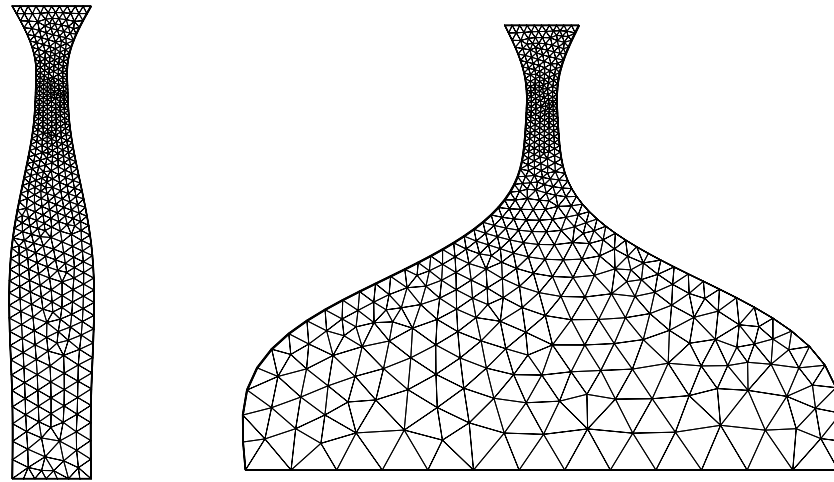
Source	$c = 2.5$	$c = 5$	$c = 10$	$c = 15$	$c = 20$
Bijl	—	0.25	0.53	—	2.67
Theory Laval	0.113	0.232	0.529	2.443	2.443
Our results	0.118	0.237	0.513	2.707	2.707

consisting of 6488 vertices of which 96 are positioned at the body, are illustrated in Figure 6.22a. From Figure 6.22b we conclude that also for this test case expression (6.20) is valid, with $t_m = 1.71$ and $t_s = 0.0448$. Hence, the time needed to build the three matrices corresponds to the time needed to do, roughly, $1.71/0.0448 \approx 38$ iterations in the linear solver. The maximal time step for which the computation remains stable corresponds to a Courant number, defined as in (6.15), equal to 6.6.

6.6 Converging-diverging nozzle

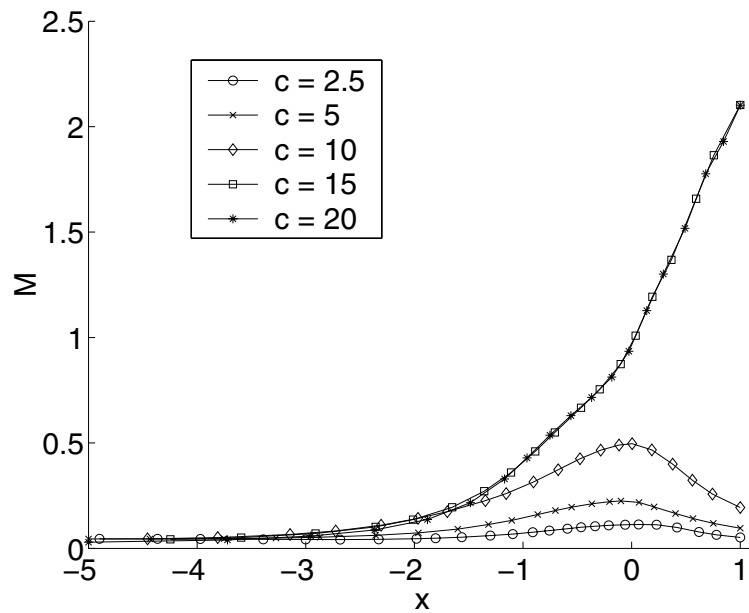
The converging-diverging nozzle, also studied by Bijl using a structured grid of 490 cells in Section 8.2 of her PhD thesis [8] and in [9], is an excellent test case to demonstrate the Mach-uniform accuracy and efficiency of the Mach-uniform pressure-correction method. This is the case because large regions with compressible as well as with (nearly) incompressible flow are present. The following contraction ratios, denoted by c , are considered: 2.5, 5, 10, 15 and 20. The inlet Mach number in all cases is put to equal to 0.045. The dimensionless initial conditions are: $\mathbf{m} = (0, 1)$, $p = 0$ and $h = 1$. The relative stationary accuracy of all variables, see expression (3.1), is put to 10^{-3} . Only in the case of subsonic outflow, the pressure is prescribed at the outlet. The height of the outlet is kept at 2.5 times the height of the throat. For all contraction ratios, the number cells, faces and vertices in the grid are kept roughly the same (850, 1350 and 490, respectively), and the minimal meshwidth is kept approximately 2×10^{-2} . For the time-step Δt , several values between 0.01 and 0.16 are chosen. With the Courant number as defined in (6.15) and $M_r = 0.045$, σ consequently varies between 12 and 186. In Figure 6.23a–b the grids for $c = 2.5$ and $c = 20$ are depicted, and the flow is directed from the bottom of the figure to the top.

The flow remains subsonic for contraction ratios equal to 2.5, 5 and 10, while it becomes supersonic behind the throat for $c = 15, 20$, see Figure 6.23c. The maximum of the Mach number M_{\max} occurs for subsonic flow in the throat, while for supersonic flow M_{\max} is found in the corners near the outlet. In Table 6.2 the values for M_{\max} , as obtained by Bijl [8] and our Mach-uniform unstructured scheme, are given. In addition, values derived from the theory of the Laval nozzle are listed. Under the assumption of isentropic quasi 1D flow, the following relation is derived



(a)

(b)



(c)

Figure 6.23: Grids used for the nozzle flow with a contraction ratio equal to 2.5 (a) and 20 (b). Mach number at the centerline of the domain for various contraction ratios (c). The throat is located at $x = 0$ and the outlet at $x = 1$.

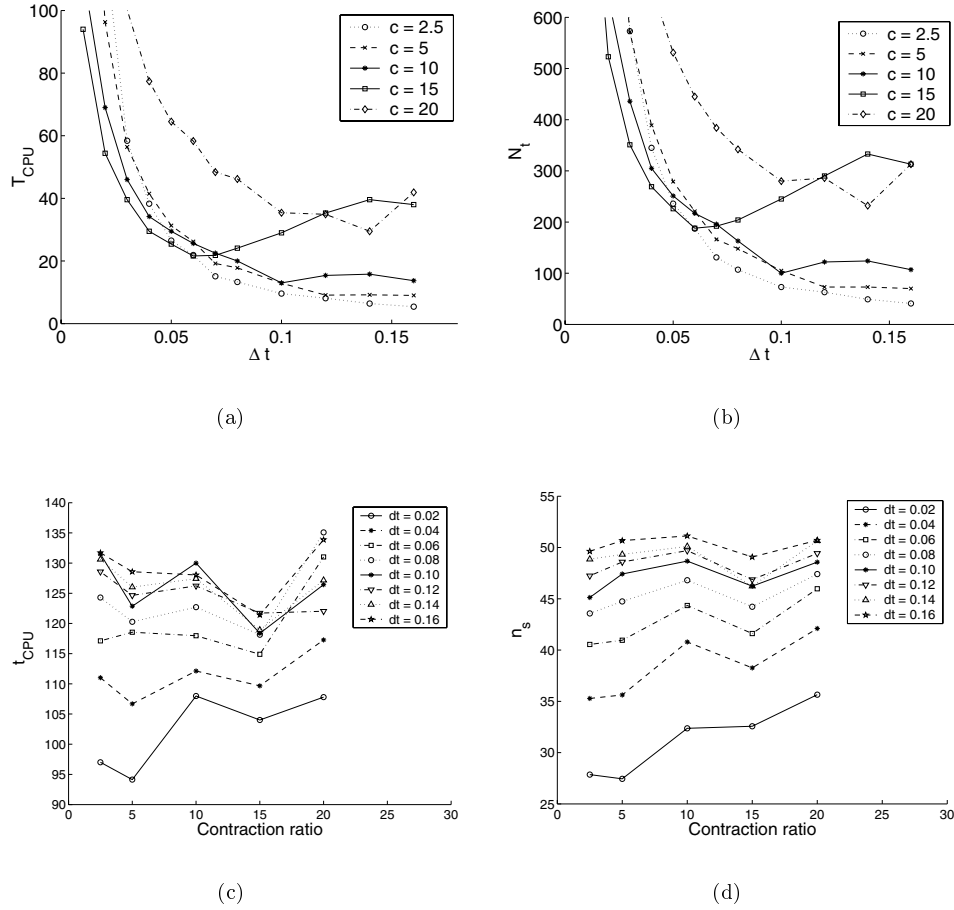


Figure 6.24: For several values of Δt and c the following quantities are shown: total CPU time needed to reach steady state (a), number of time steps needed to reach steady state (b), CPU time per time step (c), and number of solver iterations per time step (d).

in the theory of the Laval nozzle (see standard works on compressible flows):

$$\frac{A}{A^*} = \frac{1}{M} \left[\frac{2}{\gamma + 1} \left(1 + \frac{\gamma - 1}{2} M^2 \right) \right]^{(\gamma + 1)/2(\gamma - 1)}, \quad (6.21)$$

where A is the width of the nozzle at the location where the Mach number equals M , and A^* denotes the width of a reference throat with sonic flow conditions. For $c = 2.5$ and 5 , a good correspondence between theory and the numerical results exists, whereas for $c = 10$ this correspondence is less good for our results. Grid

refinement by a factor of 2 improves the results considerably, resulting in $M_{\max} = 0.528$. For contraction ratios equal to 15 and 20, the flow is choked, leading to supersonic flow after the throat. The assumption of quasi 1D flow clearly does not hold in the diverging part of the nozzle, explaining the large difference between the computed and theoretical values for M_{\max} . In a choked nozzle there is no communication between the (subsonic) converging and (supersonic) diverging part. This explains why the solution in the diverging part is exactly the same for $c = 15$ and 20, see Figure 6.23c.

For contraction ratios equal to 2.5 and 5, we saw that the flow remains virtually incompressible. For the other studied values of c , large regions with incompressible as well as with compressible flow are present. The accuracy is already established for this test case; now we turn to study the computational efficiency. Figures 6.24a and 6.24b illustrate the required computational time T_{CPU} and the elapsed number of time steps N_t to converge to steady state, for several values of c and Δt . For subsonic flow, quickest convergence is obtained for large values of Δt , e.g. $\Delta t > 0.1$. Attaining convergence for supersonic nozzle flow turns out to be slightly more time consuming, at least for larger values of Δt . From Figure 6.24c we conclude that $t_{\text{CPU}} = T_{\text{CPU}}/N_t$ hardly depends on the contraction ratio. This means that both incompressible flows and flows in which both large incompressible and large compressible regions are present can be computed without deteriorating t_{CPU} or T_{CPU} . Hence, Mach-uniform efficiency has been demonstrated. The increase in CPU time per time step for increasing values of Δt is attributed to the $\mathcal{O}(1/\Delta t)$ -behavior of the main diagonals: more solver iterations are needed because for larger values of Δt the matrices become less diagonally dominant. This is demonstrated in Figure 6.24d, where the number of solver iterations $n_s = N_s/N_t$, with N_s the total number of solver iterations in the time-stepping procedure, is displayed for various Δt and c .

6.7 Conclusions

In this chapter, computations for various inviscid compressible flows carried out with the unstructured staggered scheme are described. The Burgers equation is appropriate to test the discretization of the inertia term in the momentum equation. We find that the convecting velocity should, in order to avoid excessive numerical diffusion, be approximated in a central manner according to (4.12), and for the convected momentum term the first order upwind scheme (4.18) should be selected. This scheme, although a small overshoot appears right before discontinuities, is found to perform satisfactorily. Riemann problems are suitable test cases for studying the capability of numerical schemes for the resolution of shocks, contact discontinuities and rarefaction waves. Our scheme is found to converge, without giving rise to spurious modes, to the entropy solution. This confirms numerically that the discretization conserves mass, momentum and energy. The accuracy is similar to that of the well-

accepted Roe and AUSM scheme. For supersonic flow, no entropy fix is required and no glitch appears when the velocity changes sign or when the flow evolves from subsonic to supersonic. The path integral method is found to perform slightly better than the other schemes to compute the pressure gradient. For the fully compressible flow (FC) method, it is best to select ρH as primary energy variable.

Central discretization of the inertia term leads to a matrix that has eigenvalues that are close to the imaginary axis. This is investigated by computing the eigenvalues of this matrix for an incompressible flow in a channel with 10% circular arc bump. A grid refinement study for subsonic flow demonstrates that the upwind scheme is first order accurate in space, and that the accuracy is relatively insensitive to whether Courant or ‘general’ unstructured grids are used. Supersonic flow in a channel with 4% circular bump shows the capability to deal with supersonic flow: good agreement with previously published results is obtained.

Flows around the NACA 0012 airfoil are used to study the accuracy and performance of our schemes for incompressible and compressible flows. It is found that the Mach-uniform (MU) formulation is for the whole studied range of Mach numbers much more efficient than the fully compressible flow (FC) approach. This is due to the fact that in the MU formulation the pressure is taken implicitly, which enables the use of much larger time steps. In the MU formulation, quick convergence is obtained for Courant numbers typically in the order of a few hundred, whereas the FC becomes unstable for Courant numbers larger than 2 or 3. In the FC approach, most CPU time is spent in computation of the matrix elements and right-hand side, whereas the MU approach spends most of its CPU time in the linear solver. As is common for density-based methods, the FC approach becomes less accurate in the low Mach number regime; this is not the case for the pressure-based MU approach.

An example of a flow in which both incompressible and compressible flow regions occupy a large part of the flow domain is that in a converging-diverging nozzle. Solutions demonstrate good agreements with structured grid and theoretical results. Mach-uniform accuracy is obtained, since the computation time per time step only depends on the time step size and not on the contraction ratio of the nozzle.

In conclusion, it can be said that the present discretization on unstructured staggered grids, in conjunction with the Mach-uniform formulation, is capable of computing both incompressible and compressible flows in a way that is accurate and efficient uniform in the Mach number.

Appendix A

Monotonicity considerations

Designing numerical schemes to compute accurate solutions to (scalar) conservation laws is especially difficult when discontinuities or steep gradients are to be expected. Theoretical convergence requirements are summarized briefly in Appendix A.1. Obtaining a monotone discretization for the Burgers equation is described in Appendix A.2 for a 1D grid and in Appendix A.3 for an unstructured staggered grid. A suitable definition for the Courant number on unstructured staggered grids is derived in Appendix A.4.

A.1 Requirements for convergence

The celebrated Lax-Wendroff theorem [66] asserts that if the solution of a conservative and consistent scheme converges as the time step and meshwidth tend to zero, then it converges to a weak solution of the conservation law. Ensuring convergence to the physically relevant weak solution, often called the genuine weak solution, entropy solution or vanishing viscosity solution, requires that the numerical scheme must in addition satisfy some form of the entropy condition. It turns out, see [47], that monotone schemes fulfill this requirement for scalar conservation laws. Let a numerical scheme designed for solving the scalar conservation law

$$\frac{\partial \varphi}{\partial t} + \frac{\partial f}{\partial x} = 0, \quad (\text{A.1})$$

with $f = f(\varphi)$ a given flux function and $\varphi = \varphi(x, t)$ a scalar, be given by

$$\varphi_i^{n+1} = \hat{H}_i(\varphi_{i-m}^n, \dots, \varphi_{i+k}^n). \quad (\text{A.2})$$

Here φ_i^n is an approximation to $\varphi(x_i, t^n)$, \hat{H}_i is the operator representing the discretization, and $n, m, k \geq 0$ are integers. The numerical scheme is monotone when

\hat{H} is a nondecreasing function of each of its arguments:

$$\frac{\partial \hat{H}_i}{\partial \varphi_j^n} \geq 0, \quad \forall j. \quad (\text{A.3})$$

This definition of monotone is a discrete version of the following property of exact solutions of conservation law (A.1): if two initial data functions $\phi(x, 0)$ and $\psi(x, 0)$ satisfy $\phi(x, 0) \leq \psi(x, 0)$ for all x , then their corresponding solutions $\phi(x, t)$ and $\psi(x, t)$ satisfy $\phi(x, t) \leq \psi(x, t)$ for all x and t .

A.2 Monotone discretization of the Burgers equation on a 1D grid

Consider the 1D grid shown in Figure A.1, where the momentum and velocity are located in grid points x_{i+k} , $k \in \mathbb{Z}$. For finite volume integration of the 1D momentum equation in point i we propose the following options for the control volumes (CV):

Option 1. The CV is bounded by adjacent momentum points: $[x_{i-1}, x_{i+1}]$.

Option 2. With x_i positioned half way x_i and x_{i+1} , the CV is formed by $[x_{i-1}, x_i]$.

The length of the CV in option 1 is $2\Delta x_i$, with Δx_i equal to the length of the CV in option 2, i.e. $\Delta x_i = x_i - x_{i-1} = \frac{1}{2}(x_{i+1} - x_{i-1})$. Integration of the 1D momentum equation over the CV, omitting the pressure and viscous terms, results in:

$$\begin{cases} 2\lambda_i m_i + (um)_{i+1} - (um)_{i-1} = 0, & \text{option 1;} \\ \lambda_i m_i + (um)_i - (um)_{i-1} = 0, & \text{option 2,} \end{cases} \quad (\text{A.4})$$

where $\lambda_i = \Delta x_i / \Delta t$. By putting $m = u$, a scalar conservation law, often referred to as the Burgers equation, is obtained; see equation (6.1). Assuming that the velocity is in the positive direction, the following upwind flux evaluations and time step restrictions lead to monotone schemes:

$$\begin{cases} (um)_{i+1} = u_i m_i, & \sigma_i \leq 1, & \text{option 1;} \\ (um)_i = u_i m_i, & \sigma_i \leq 1/2, & \text{option 2.} \end{cases} \quad (\text{A.5})$$

Here σ_i stands for the local Courant number:

$$\sigma_i = u_i / \lambda_i = u_i \Delta t / \Delta x_i. \quad (\text{A.6})$$

Note that, though it may seem to contradict intuition, flux evaluations of the form $(um)_{i+1} = u_{i+1} m_{i+1}$, $(um)_{i+1} = u_{i+1} m_i$ or $(um)_i = \frac{1}{2}(u_i + u_{i+1}) m_i$ are not monotone. The notion that also the convecting velocity must be upwind biased inspired us to propose the upwind approximation (4.14) for the convecting velocity in the 2D case.

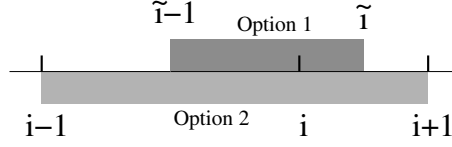


Figure A.1: A 1D grid and possible control volumes.

A.3 Monotone discretization of the Burgers equation on an unstructured staggered grid

Standard multi-dimensional methods often constitute of a more or less straightforward application of one-dimensional methods to the direction normal to the CV face. Properties of multi-dimensional methods are usually considered by studying the one-dimensional version of the underlying scheme and by using then some more or less heuristic arguments. Experience confirms that this approach is useful. Since in our staggered scheme only normal components of the momentum are stored, such an approach is not possible. This is so because the normal vectors at the faces in the grid do not point in the same direction, and, obviously, there exists nothing like monotonicity preservation for arbitrarily directed momentum components. We will study whether our scheme for the momentum equation is monotone by assuming flow in a fixed direction, after which (A.3) is applied to the discrete momentum components *in the given flow direction*. Namely, by imposing a 1D flow, which is obviously a special case of a 2D flow, we can arrive afterwards at the discretization in terms of the horizontal momentum components. And to this underlying discretization, the definition of monotone schemes can be applied.

Consider the grid illustrated in Figure A.2. The normal vectors are indicated and we put $\mathbf{N}_i = (1, 0)$. Assume a 1D flow with a Cartesian velocity vector (\hat{u}, \hat{v}) that follows from:

$$\begin{cases} \frac{\partial \hat{m}}{\partial t} + \frac{\partial \hat{u} \hat{m}}{\partial x} = 0, & \hat{m} = \hat{u}, \\ \hat{v} = 0, \end{cases} \quad (\text{A.7})$$

hence the flow is parallel to \mathbf{N}_i . For reasons of clarity, we keep a difference in notation between \hat{u} and \hat{m} . The relation between the normal velocity (momentum) component and \hat{u} (\hat{m}) at some face e is given by:

$$\mathbf{u}_e = \mathbf{u}_e \cdot \mathbf{N}_e = \hat{u}_e N_{e,x}; \quad \mathbf{m}_e = \mathbf{m}_e \cdot \mathbf{N}_e = \hat{m}_e N_{e,x}, \quad (\text{A.8})$$

with $N_{e,x}$ the x -component of \mathbf{N}_e . Integration over the two triangles adjacent to face i , cf. (4.7), and explicit Euler time integration yields:

$$m_i^{n+1} = m_i - \frac{\Delta t}{\Omega_i} \sum_{e(i)} (\mathbf{u}_e \cdot \mathbf{N}_e) (\mathbf{m}_e \cdot \mathbf{N}_i) l_e. \quad (\text{A.9})$$

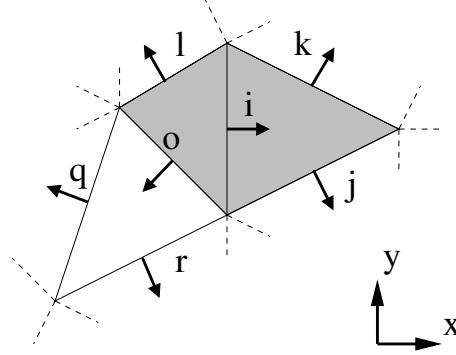


Figure A.2: Part of a 2D grid.

All terms in the right-hand side are taken at time-level n , and summation runs over the faces j, k, l and o . Assume that $\hat{u} > 0$, i.e. the flow is from left to right, and that (part of) it enters the CV through face o . The reconstruction procedure, with

$$\mathbf{N}_i = \eta_q^i \mathbf{N}_q + \eta_r^i \mathbf{N}_r, \quad (\text{A.10})$$

yields the following first order upwind approximation at face o , cf. (4.18):

$$\mathbf{m}_o \cdot \mathbf{N}_i = \eta_q^i m_q + \eta_r^i m_r = \eta_q^i N_{q,x} \hat{m}_q + \eta_r^i N_{r,x} \hat{m}_r. \quad (\text{A.11})$$

For the computation of the convecting velocity, two methods have been designed, see page 56, and we will study both.

1. Central approximation

Application of (4.12) yields for the convecting velocity at face o :

$$\mathbf{u}_o \cdot \mathbf{N}_o = \hat{u}_o N_{o,x}. \quad (\text{A.12})$$

With $\hat{m} = \hat{u}$ we arrive at, cf. (A.3):

$$\frac{\partial \hat{H}_i}{\partial \hat{u}_r} = -\frac{\Delta t}{\Omega_i} \hat{u}_o N_{o,x} \eta_r^i N_{r,x}, \quad (\text{A.13})$$

resulting in a scheme which, recalling $N_{o,x} < 0$, is not monotone if $\eta_r^i N_{r,x} < 0$.

2. Upwind approximation

Upwinding of the convecting velocity according to (4.14) results in:

$$\mathbf{u}_o \cdot \mathbf{N}_o = \mathbf{u}_o \cdot (\zeta_q^o \mathbf{N}_q + \zeta_r^o \mathbf{N}_r) \approx \zeta_q^o u_q + \zeta_r^o u_r = \zeta_q^o N_{q,x} \hat{u}_q + \zeta_r^o N_{r,x} \hat{u}_r. \quad (\text{A.14})$$

Applying the definition of monotone schemes gives:

$$\frac{\partial \hat{H}_i}{\partial \hat{u}_r} = \Delta t [-2\hat{u}_r \zeta_r^o \eta_r^i N_{r,x} - \hat{u}_q N_{q,x} N_{r,x} (\zeta_q^o \eta_r^i + \zeta_r^o \eta_q^i)] / \Omega_i. \quad (\text{A.15})$$

Discussion

With, for example, $\mathbf{N}_r = (1, -2)/\sqrt{5}$, $\mathbf{N}_q = (-5, 2)/\sqrt{29}$ and $\mathbf{N}_o = (-1, -1)/\sqrt{2}$, none of the velocity interpolation methods in combination with (A.11) results in a monotone scheme, since in both cases we have $\partial\hat{H}_i/\partial\hat{u}_r < 0$, and as a result spurious wiggles might appear. On the basis of numerical experiments discussed in Section 6.1, we observe that indeed a spurious mode is generated, but it remains relatively small, especially if the convecting velocity is approximated in an upwind manner. No spurious oscillations appeared in the numerical experiments for the Euler equations, see the rest of Chapter 6, not even with central approximation of the convecting velocity. Hence, the pessimistic conclusions that one could draw from this section are not justified.

A.4 Definition of the Courant number

Usually a condition of the form $\sigma < C$, where σ is the Courant number and C some small positive number, must be satisfied for stability of schemes based on explicit time integration for hyperbolic systems. In this appendix we will derive, using (A.3), such conditions for discretizations on unstructured staggered grids. In Appendix A.4.1 this is done for the continuity and energy equation. A suitable condition for the momentum equation is derived in Appendix A.4.2. A useful condition for the Euler equations is given in Appendix A.4.3.

A.4.1 Courant number for the continuity and energy equation

Consider the continuity equation (2.1) as a scalar conservation law for the density. Discretization on an unstructured staggered grid using the explicit Euler time integration scheme gives, cf. (4.89):

$$\rho_1^{n+1} = \rho_1^n - \frac{\Delta t}{\Omega_1} \sum_{e(1)} \rho_e^n u_e \bar{l}_e, \quad (\text{A.16})$$

where summation runs over the three faces of cell 1. Inserting central scheme (4.91) into this expression and using the notation of Figure 4.14 yields:

$$\rho_1^{n+1} = \rho_1^n - \frac{1}{2} \frac{\Delta t}{\Omega_1} \left[\rho_1 \sum_{e(1)} u_e \bar{l}_e + \rho_2 u_i \bar{l}_i + \rho_3 u_j \bar{l}_j + \rho_4 u_k \bar{l}_k \right]. \quad (\text{A.17})$$

With (A.3) it is easy to see that monotonicity is only preserved if $u_i \bar{l}_i \leq 0$, $u_j \bar{l}_j \leq 0$ and $u_k \bar{l}_k \leq 0$ are satisfied simultaneously, i.e. the flow must enter cell 1 through

all its three faces. Since this is impossible hold for each cell in the grid, the central scheme is, as was of course to be expected, not monotone.

The first order upwind scheme for face i , expression (4.90), can be written in the following form:

$$\rho_i u_i \bar{l}_i = \rho_1 \max(u_i \bar{l}_i, 0) + \rho_2 \min(u_i \bar{l}_i, 0), \quad (\text{A.18})$$

where

$$\max(z, 0) = \frac{1}{2}(z + |z|), \quad \min(z, 0) = \frac{1}{2}(z - |z|). \quad (\text{A.19})$$

It is obvious that $\max(z, 0) \geq 0$ and $\min(z, 0) \leq 0$ for all $z \in \mathbb{R}$. Putting the first order upwind scheme into (A.16) and application of (A.3) results in:

$$\begin{aligned} \frac{\partial \hat{H}}{\partial \rho_1} &= 1 - \frac{\Delta t}{\Omega_1} \sum_{e(1)} \max(u_e \bar{l}_e, 0); & \frac{\partial \hat{H}}{\partial \rho_2} &= -\frac{\Delta t}{\Omega_1} \min(u_i \bar{l}_i, 0); \\ \frac{\partial \hat{H}}{\partial \rho_3} &= -\frac{\Delta t}{\Omega_1} \min(u_j \bar{l}_j, 0); & \frac{\partial \hat{H}}{\partial \rho_4} &= -\frac{\Delta t}{\Omega_1} \min(u_k \bar{l}_k, 0). \end{aligned}$$

Hence, the first order upwind scheme is monotone if

$$\sigma_1 = \frac{\Delta t}{\Omega_1} \sum_{e(1)} \max(u_e \bar{l}_e, 0) \leq 1, \quad (\text{A.20})$$

where σ_1 can be interpreted as the local Courant number.

A global time-stepping scheme, with a fixed time-step Δt given in advance, is used in our numerical experiments. We will show how condition (A.20) can be applied to determine a value for Δt such that the scheme remains monotone. Assuming that the velocity field \mathbf{u} does not vary much over a cell leads to

$$0 = \mathbf{u} \cdot \sum_{e(1)} \mathbf{N}_e \bar{l}_e = \sum_{e(1)} (\mathbf{u} \cdot \mathbf{N}_e) \bar{l}_e \approx \sum_{e(1)} (\mathbf{u}_e \cdot \mathbf{N}_e) \bar{l}_e = \sum_{e(1)} u_e \bar{l}_e. \quad (\text{A.21})$$

This result, replacing the ‘ \approx ’-sign by a ‘=’-sign, is used to arrive at

$$\sum_{e(1)} \max(u_e \bar{l}_e, 0) = \max(|u_i \bar{l}_i|, |u_j \bar{l}_j|, |u_k \bar{l}_k|) \equiv \max_{e(1)} |u_e \bar{l}_e|, \quad (\text{A.22})$$

which follows after some easy manipulations. With $u_e \leq |\mathbf{u}|_1$, where $|\mathbf{u}|_1$ is the maximal magnitude of the velocity in cell 1, it is trivial to see that

$$\max_{e(1)} |u_e \bar{l}_e| \leq |\mathbf{u}|_1 \max_{e(1)} l_e. \quad (\text{A.23})$$

This all leads for the local Courant number in cell 1 as defined in (A.20) to:

$$\sigma_1 \leq \frac{|\mathbf{u}|_1 \Delta t}{\Delta x_1}, \quad \Delta x_1 = \Omega_1 / \max_{e(1)} l_e. \quad (\text{A.24})$$

In order to obtain a safe estimate for the time step, the global Courant number σ is defined as follows, with \max_c indicating the maximum over all cells c :

$$\max_c \sigma_c \leq \max_c \left(\frac{|\mathbf{u}|_c \Delta t}{\Delta x_c} \right) \leq \frac{\max_c |\mathbf{u}|_c \Delta t}{\min_c \Delta x_c} = \frac{\max |\mathbf{u}| \Delta t}{\Delta x} \equiv \sigma, \quad (\text{A.25})$$

and $\max |\mathbf{u}|$ the maximum velocity in the domain. The minimal meshwidth is defined as

$$\Delta x = \min_c \Delta x_c = \min_c \left(\Omega_c / \max_{e(c)} l_e \right), \quad (\text{A.26})$$

where $e(c)$ refers to the three faces of cell c . Monotonicity, and hence stability, for the explicit Euler scheme is ensured for $\sigma \leq 1$. Application of this approach to the energy equation goes in a similar fashion, leading to identical results.

A.4.2 Courant number for the momentum equation

Usually, application of condition (A.3) to the diagonal element (i.e. $\partial \hat{H}_i / \partial \varphi_i > 0$) results in the time-step restriction $\sigma < C$; this also holds for the derivation in Appendix A.4.1. We will use the condition $\partial \hat{H}_i / \partial \varphi_i > 0$ to arrive at a suitable definition for the Courant number for the discretized momentum equation (although, as discussed in Appendix A.3, our discretization of the momentum equation is not monotone since we have $\partial \hat{H}_i / \partial \varphi_j < 0$ for some $j \neq i$).

Application of the first order upwind approximation (4.18) to the discretized momentum equation (A.9) yields:

$$m_i^{n+1} = m_i - \frac{\Delta t}{\Omega_i} \left(m_i \sum_{e(i)} \max(u_e \bar{l}_e, 0) + \sum_{e(i)} (\mathbf{m}_e \cdot \mathbf{N}_i) \min(u_e \bar{l}_e, 0) \right), \quad (\text{A.27})$$

where summations run over the adjacent faces of face i , and in the interpolations to obtain $(\mathbf{m}_e \cdot \mathbf{N}_i)$ only off-diagonal elements are present. Application of (A.3) to the diagonal element results in the following criterion:

$$\sigma_i = \sum_{e(i)} \max(u_e \bar{l}_e, 0) \Delta t / \Omega_i \leq 1. \quad (\text{A.28})$$

Using (A.21), while recalling that the summation now runs over four faces, we get:

$$\sum_{e(i)} \max(u_e \bar{l}_e, 0) \leq 2 \max_{e(i)} |u_e \bar{l}_e|. \quad (\text{A.29})$$

Following the same route as pursued in Appendix A.4.1, we arrive at the same definition for the Courant number and meshwidth, namely equations (A.25) and (A.26), and the same consequence of (A.3): $\sigma \leq 1$.

A.4.3 Courant number for the Euler equations

In the previous two sections we have derived an appropriate definition for the Courant number for the discretized continuity, energy and momentum equations, under the restriction that these equations could be considered as independent scalar conservation laws. However, the Euler equations form a coupled system in which information travels along the characteristics with velocities equal to $(u - a)$, u , $(u + a)$, where a is the speed of sound and $u = |\mathbf{u}|$ the velocity of the fluid. Inspired by this, we replace the maximum velocity in (A.25) by the maximum signal speed $(u + a)$. This leads to our definition of the Courant number for the Euler equations on unstructured staggered grids:

$$\sigma = \frac{\max(u + a)\Delta t}{\Delta x}, \quad \Delta x = \min_c \Delta x_c, \quad \Delta x_c = \Omega_c / \max_{\epsilon(c)} l_{\epsilon}. \quad (\text{A.30})$$

On a given grid it is easy to compute Δx . The problem is finding a suitable value for the maximum signal speed, since this value may vary in time while we keep Δt fixed. For the Riemann problems discussed in Section 6.2 we use the exact solution to determine the maximal signal speed. For all other test cases, we base the signal speed on the freestream conditions, for which we take the prescribed values at the inlet of the flow domain. Note that, since we employ an implicit time-stepping scheme, we are not restricted to time steps satisfying $\sigma \leq 1$. In fact, Courant numbers encountered in our computations are typically in the order of 10^2 .

It is clear that the reasoning given in this and the previous sections contains many heuristic arguments. A practical approach is to consider (A.30) as an easily applicable definition for the Courant number, and choose Δt such that an appropriate value for σ results.

Appendix B

Consistent discretization of the inertia term

In Section 4.2.1, two possible options for a control volume (CV) for the momentum equation are introduced. Option 1 embodies a CV consisting of the two triangles adjacent to the face under consideration, while in option 2 integration over the half of these triangles is performed. These options result in distinct discretizations for the inertia term, see Sections 4.2.3 and 4.2.4, and in this appendix we study whether the resulting schemes are consistent.

Consistency of schemes for the inertia term is studied by postulating an exact solution $\mathbf{m}^{(e)} = \mathbf{m}^{(e)}(\mathbf{x})$. Substitution gives the following equation:

$$(u^\beta m^\alpha)_{,\beta} = f^\alpha, \quad f^\alpha \equiv ((u^{(e)})^\beta (m^{(e)})^\alpha)_{,\beta}. \quad (\text{B.1})$$

Finite volume integration gives, with CV indicating the control volume for face i :

$$\int_{\text{CV}} (u^\beta m^\alpha)_{,\beta} d\mathbf{x} = \int_{\text{CV}} f^\alpha d\mathbf{x}. \quad (\text{B.2})$$

For simplicity, the right-hand side is approximated as follows:

$$\int_{\text{CV}} f^\alpha d\mathbf{x} \approx f_i^\alpha \Omega_i, \quad (\text{B.3})$$

where Ω_i is the area of the CV and $f_i^\alpha = f^\alpha(\mathbf{x}_i)$, with \mathbf{x}_i the location of face midpoint i . This approximation may affect the rate of convergence, but does not cause a convergent scheme to become nonconvergent. The left-hand side of (B.2) is approximated according to the scheme selected, and linearization is achieved by inserting the exact velocity $(u^{(e)})^\beta$ for u^β . A scheme is consistent if the error, i.e. the difference between the exact and numerical solution, tends to zero as the mesh

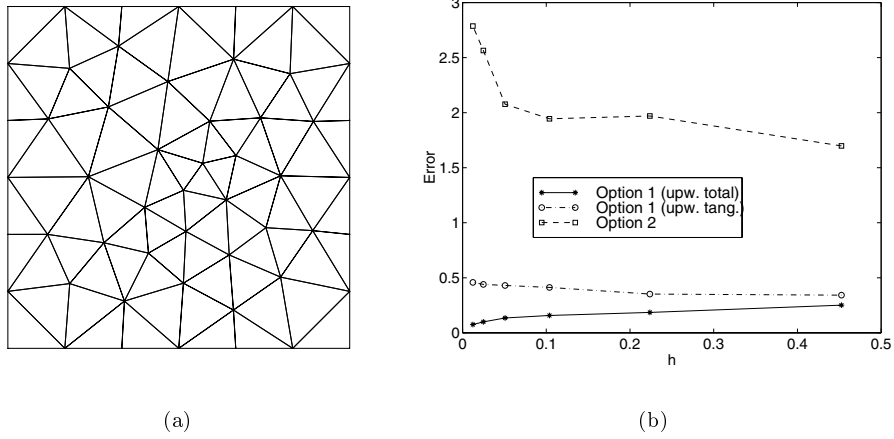


Figure B.1: Example of a grid used for testing consistency (a). Behavior of the maximum norm of the error for the various options (b).

is refined. We choose the following exact solution for the momentum and velocity field, putting $\rho = 1$: $\mathbf{m}^{(e)} = \mathbf{u}^{(e)} = (\sin(x) + 1, \sin(y) + 1)$. The grid on domain $[0, 1\frac{1}{2}] \times [0, 1\frac{1}{2}]$, depicted in Figure B.1a, is refined various times. Figure B.1b shows that a CV formed by two triangles together with upwind method (4.18) (indicated by: Option 1, upw. total) leads to a consistent scheme. On the other hand, using the upwind method given by (4.21) (indicated by: Option 1, upw. tang.), or using integration over half triangles (indicated by: Option 2) leads to inconsistency. It must be noted however, that we computed some realistic 2D flow problems with upwind scheme (4.21), and that the results were about as accurate as the favored scheme.

Appendix C

Symmetry-preserving discretization

In the works of Veldman et al. [119, 122, 123], convincing arguments are given to use a so-called symmetry-preserving or spectro-consistent discretization for the various differential operators (e.g. the convection operator) that govern incompressible flows. In Appendix C.1 we give the definition of a symmetry-preserving discretization, and generalize this idea to compressible flows. As discussed in Appendix C.2, the central scheme for the convection term in scalar equations on unstructured staggered grids is always symmetry-preserving for scalar equations, while some grid regularity is required in order to arrive at a symmetry-preserving discretization for the inertia term in the momentum equation. Discussion of discretization in the vicinity of boundaries is included.

C.1 What is a symmetry-preserving discretization?

Symmetry-preserving properties of the convection term: continuous case

Let $\mathcal{C} : V \rightarrow V$ be a linear operator and let $\mathcal{C}^* : V \rightarrow V$ be its adjoint, defined by

$$\langle \mathcal{C}\phi, \psi \rangle = \langle \phi, \mathcal{C}^*\psi \rangle \quad (\text{C.1})$$

for all ϕ and ψ sufficiently smooth. Let V be the Hilbert space of square-integrable functions on \mathbb{R}^n with the inner product

$$\langle f, g \rangle = \int_S fg \, d\mathbf{x}. \quad (\text{C.2})$$

Integration takes place over a domain S , and $f = f(\mathbf{x})$ and $g = g(\mathbf{x})$. The convection operator is defined by

$$\mathcal{C}\phi = \nabla \cdot (\mathbf{u}\phi), \quad (\text{C.3})$$

where we assume both ϕ and \mathbf{u} to be differentiable. The diffusion and viscous operator will be discussed later on in this section. With help of the identity

$$\psi \nabla \cdot (\mathbf{u}\phi) + \phi \nabla \cdot (\mathbf{u}\psi) = \nabla \cdot (\mathbf{u}\phi\psi) + \phi\psi(\nabla \cdot \mathbf{u}) \quad (\text{C.4})$$

we arrive at:

$$\langle \mathcal{C}\phi, \psi \rangle = - \int_S \phi \nabla \cdot (\mathbf{u}\psi) d\mathbf{x} + \int_S \phi\psi(\nabla \cdot \mathbf{u}) d\mathbf{x} + \oint_{\partial S} (\mathbf{u} \cdot \mathbf{n})\phi\psi d\Gamma, \quad (\text{C.5})$$

where \mathbf{n} is the outward normal at ∂S , the boundary of S . We write $\mathcal{C} = \mathcal{C}_i + \mathcal{C}_b + \mathcal{C}_r$, where operators \mathcal{C}_i , \mathcal{C}_b and \mathcal{C}_r are defined as follows:

$$\langle \mathcal{C}_i\phi, \psi \rangle = - \int_S \phi \nabla \cdot (\mathbf{u}\psi) d\mathbf{x}, \quad (\text{C.6a})$$

$$\langle \mathcal{C}_d\phi, \psi \rangle = \int_S \phi\psi(\nabla \cdot \mathbf{u}) d\mathbf{x}, \quad (\text{C.6b})$$

$$\langle \mathcal{C}_r\phi, \psi \rangle = \oint_{\partial S} (\mathbf{u} \cdot \mathbf{n})\phi\psi d\Gamma. \quad (\text{C.6c})$$

We see that operator \mathcal{C}_r vanishes for flows in closed systems, since in this case $(\mathbf{u} \cdot \mathbf{n}) = 0$ at ∂S . In addition, operator $\mathcal{C}_d = (\nabla \cdot \mathbf{u})$ vanishes for incompressible flows. Note furthermore that operator \mathcal{C}_i is skew-symmetric:

$$\mathcal{C}_i = -\mathcal{C}_i^*. \quad (\text{C.7})$$

Consider a scalar conservation law of the following form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0, \quad (\text{C.8})$$

where $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is given and $\phi = \phi(\mathbf{x}, t)$ is the unknown. We again assume \mathbf{u} and ϕ to be differentiable. Multiplication by ϕ yields, with help of (C.4) and inserting $\psi = \frac{1}{2}\phi$:

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{u}k) = -k(\nabla \cdot \mathbf{u}), \quad k = \frac{1}{2}\phi^2. \quad (\text{C.9})$$

Integration over a domain S with boundary ∂S gives:

$$\frac{dK}{dt} + \int_S k(\nabla \cdot \mathbf{u}) d\mathbf{x} = - \oint_{\partial S} (\mathbf{u} \cdot \mathbf{n})k d\Gamma, \quad K = \int_S k d\mathbf{x}. \quad (\text{C.10})$$

This expresses that in incompressible flows convection does not create nor destroy, but merely redistributes quantity k , and that consequently K is conserved if S is closed. Note that (C.10) is identical to

$$\left\langle \frac{\partial \phi}{\partial t} + \mathcal{C}\phi, \frac{1}{2}\phi \right\rangle = 0. \quad (\text{C.11})$$

In the integral term in the left-hand side of (C.10) we recognize $\langle \mathcal{C}_d \phi, \phi \rangle / 2$, and the right-hand side contains the term $-\langle \mathcal{C}_r \phi, \phi \rangle / 2$. Because of operator \mathcal{C}_i being skew-symmetric, this operator is not present in (C.10).

Symmetry-preserving properties of the convection term: discrete case

Finite volume discretization of (C.8) gives

$$H \frac{d\phi_h}{dt} + C \phi_h = \mathbf{r}, \quad (\text{C.12})$$

where ϕ_h stands for the solution vector containing all unknowns, H is a positive definite diagonal matrix containing the areas of the control volumes, matrix $C = C(\mathbf{u}_h)$ represents the discretized convection operator and right-hand side vector \mathbf{r} corresponds to the boundary conditions. We split C in two parts:

$$C = C_i + \frac{1}{2} H C_d, \quad (\text{C.13})$$

where C_i and C_d can, as we will see in Section C.2, be chosen such as to correspond to the continuous operators \mathcal{C}_i and \mathcal{C}_d respectively, while vector \mathbf{r} will turn out to correspond to $\mathcal{C}_r \phi$. E.g. for incompressible flow in closed systems, C_d and \mathbf{r} vanish, and time evolution of the discrete equivalent of K , which is $K_h = \frac{1}{2} \phi_h^T H \phi_h$, follows from (omitting the factor 1/2):

$$\begin{aligned} \frac{d(\phi_h^T H \phi_h)}{dt} &= \frac{d\phi_h^T}{dt} H \phi_h + \phi_h^T \frac{d(H \phi_h)}{dt} = - (H^{-1} C_i \phi_h)^T H \phi_h - \phi_h^T C_i \phi_h = \\ &= - \left(\phi_h^T C_i^T H^{-T} H \phi_h - \phi_h^T C_i \phi_h \right) = - \phi_h^T (C_i^T + C_i) \phi_h, \end{aligned} \quad (\text{C.14})$$

where we applied (C.12). Hence, if the discrete convection operator C_i is skew-symmetric:

$$C_i = -C_i^T, \quad (\text{C.15})$$

then K_h is, as is K in the continuous case, conserved. If this is so, then the discretization is called spectro-consistent or symmetry-preserving. Note that C_i and $\frac{1}{2} H C_d$ correspond to the skew-symmetric and symmetric part of C , respectively. If C_i cannot be made, for whatever reason, skew-symmetric, then C should for stability reasons be a positive real matrix (i.e. $C + C^T$ is positive definite), and consequently C is N -stable (i.e. all eigenvalues have a positive real part). Preferably, the real part of the eigenvalues is small compared to the imaginary part.

Consider now the general situation, i.e. compressible flow in an open system. In this case, \mathbf{r} and C_d are not equal to zero, and a derivation similar to the one given above results in (assuming (C.15) to hold):

$$\frac{dK_h}{dt} + \frac{1}{2} \phi_h^T H C_d \phi_h = \mathbf{r}^T \phi_h, \quad (\text{C.16})$$

which is clearly a discrete equivalent of (C.10). To arrive at this expression, we used $C_d^T = C_d$.

The reasoning given above, omitting the pressure and viscous terms, can equally well be applied to the momentum equation. One can show that the momentum equation not only governs conservation of momentum, but also conservation of kinetic energy $k = \frac{1}{2} \mathbf{u} \cdot \mathbf{m}$. The kinetic energy of the discrete solution, defined by $K_h = \frac{1}{2} \mathbf{u}_h^T H \mathbf{m}_h$, is conserved under the same conditions as given above for conservation of $K_h = \frac{1}{2} \phi_h^T H \phi_h$.

As we saw, in a symmetry-preserving discretization the convection operator merely redistributes ‘energy’ over the domain without dissipating (or creating) it. Although a symmetry-preserving discretization does not necessarily minimize the local truncation error, it certainly tends to keep the global truncation error small, which is, of course, a very desirable property [119, 122, 123].

Symmetry-preserving properties of the diffusion and viscous operator

The discussion above is limited to convection operator; here we will discuss the symmetry properties of the diffusion and viscous operator. The Laplacian operator $\mathcal{D}\mathcal{G}$, encountered in diffusion terms and in the pressure-correction equation, is defined by

$$\mathcal{D}\mathcal{G}\phi = -\nabla \cdot \nabla \phi. \quad (\text{C.17})$$

Assuming that ψ and ϕ are such that

$$\oint_{\partial S} (\psi \mathbf{n} \cdot \nabla \phi - \phi \mathbf{n} \cdot \nabla \psi) d\Gamma = 0, \quad (\text{C.18})$$

with \mathbf{n} the outward normal at boundary ∂S of domain S , we arrive at

$$\int_S \psi \nabla \cdot \nabla \phi d\mathbf{x} = \int_S \phi \nabla \cdot \nabla \psi d\mathbf{x}. \quad (\text{C.19})$$

In other words, see also definitions (C.1) and (C.2), the diffusion operator is symmetric: $\mathcal{D}\mathcal{G} = (\mathcal{D}\mathcal{G})^*$. For the viscous operator in incompressible flows

$$\mathcal{V}u^\alpha = -u_{,\beta\beta}^\alpha, \quad (\text{C.20})$$

while assuming that u^α and w^α are such that

$$\oint_{\partial S} (u_{,\beta}^\alpha w^\alpha n^\beta - u^\alpha w_{,\beta}^\alpha n^\beta) d\Gamma = 0, \quad (\text{C.21})$$

we get: $\mathcal{V} = \mathcal{V}^*$. In the situation that

$$\oint_{\partial S} \phi \mathbf{n} \cdot \nabla \phi d\Gamma = 0, \quad \oint_{\partial S} u_{,\beta}^\alpha u^\alpha n^\beta d\Gamma = 0, \quad (\text{C.22})$$

we get

$$\langle \mathcal{D}\mathcal{G}\phi, \phi \rangle = \int_S \nabla\phi \cdot \nabla\phi \, d\mathbf{x}, \quad \langle \mathcal{V}u^\alpha, u^\alpha \rangle = \int_S u_{,\beta}^\alpha u_{,\beta}^\alpha \, d\mathbf{x}. \quad (\text{C.23})$$

With ϕ and u^α not identical to zero, we find that: $\langle \mathcal{D}\mathcal{G}\phi, \phi \rangle > 0$ and $\langle \mathcal{V}u^\alpha, u^\alpha \rangle > 0$. In other words, operators $\mathcal{D}\mathcal{G}$ and \mathcal{V} are, in addition to symmetric, positive definite. In a symmetry-preserving discretization, the operators DG and V , i.e. the discrete equivalents of $\mathcal{D}\mathcal{G}$ and \mathcal{V} , inherit these properties, and are consequently symmetric positive definite.

Remark

Assume we have a symmetric operator \mathcal{S} , whose adjoint \mathcal{S}^* , defined by

$$\langle \mathcal{S}\phi, \psi \rangle = \langle \phi, \mathcal{S}^*\psi \rangle, \quad (\text{C.24})$$

satisfies $\mathcal{S} = \mathcal{S}^*$. The discrete equivalent of this expression is:

$$\langle S\phi_h, \psi_h \rangle = \langle \phi_h, S^T\psi_h \rangle, \quad (\text{C.25})$$

where $\langle \mathbf{u}, \mathbf{w} \rangle = \mathbf{u} \cdot \bar{\mathbf{w}}$ is the standard inner product for complex vectors, and $\bar{\mathbf{w}}$ is the complex conjugate of \mathbf{w} . In a symmetry-preserving discretization, the matrix S corresponding to \mathcal{S} is equal to its transpose: $S = S^T$.

Suppose that the proposed discretization is not symmetry-preserving, i.e. $S \neq S^T$, then we can show that (C.25) is still satisfied provided that the eigenvalues of S are real. Let \mathbf{v}_i be the eigenvector of S corresponding to eigenvalue λ_i , and expand ϕ_h and ψ_h in terms of these eigenvectors:

$$S\mathbf{v}_i = \lambda_i\mathbf{v}_i, \quad \phi_h = \sum_j \alpha_j \mathbf{v}_j, \quad \psi_h = \sum_j \beta_j \mathbf{v}_j. \quad (\text{C.26})$$

Assuming that the eigenvectors are real and orthonormal, and assuming that all α and β are real (which implies that ϕ_h and ψ_h are real), we arrive at:

$$\langle S\phi_h, \psi_h \rangle = \sum_j \alpha_j \lambda_j \beta_j, \quad \langle \phi_h, S^T\psi_h \rangle = \sum_j \alpha_j \bar{\lambda}_j \beta_j. \quad (\text{C.27})$$

We can equate these two expressions only if $\lambda_j = \bar{\lambda}_j$ for all j , hence when all eigenvalues are real. In the situation that $\lambda_j > 0$ for all j , S is positive definite. For skew-symmetric operators a similar derivation can be pursued, resulting in the condition that all eigenvalues have to be imaginary. When we studied numerically the symmetry-preserving properties of our discretization, see Sections 5.1.4 and 6.3.1, we considered the location of the eigenvalues in the complex plane.

C.2 Symmetry-preserving properties of the unstructured staggered scheme

In the unstructured staggered scheme, two different types of discretized equations are present: the equations for density and energy on the one hand, and the equation for the momentum on the other. These discretizations are discussed in full detail in Chapter 4, and here we restrict ourselves to study their symmetry-preserving properties. In this section, the symmetry-preserving properties of the discretized convection and inertia terms are studied by analytical means. It turns out to be very elaborate to investigate analytically the symmetry-preserving properties of the diffusion and viscous operators, hence these aspects are investigated only by numerical means, see Section 5.1.4.

Scalar equations

Equation (C.8) is a generic form for a conservation law, similar to the continuity and energy equation. Integration of this expression over cell 1 yields, cf. (4.89) and see Figure 4.14 for notation:

$$\Omega_1 \frac{d\phi_1}{dt} + \sum_{e(1)} u_e \phi_e \bar{l}_e = 0. \quad (\text{C.28})$$

The central scheme (4.91) gives for internal cell 1 the following nonzero elements in the convection matrix (with c_{ij} the element of C in the i th row and j th column):

$$c_{11} = \frac{1}{2}(u_i \bar{l}_i + u_j \bar{l}_j + u_k \bar{l}_k), \quad c_{12} = \frac{1}{2}u_i \bar{l}_i, \quad c_{13} = \frac{1}{2}u_j \bar{l}_j, \quad c_{14} = \frac{1}{2}u_k \bar{l}_k. \quad (\text{C.29})$$

Note that the diagonal term corresponds, as was to be expected, to the velocity divergence term, and has consequently to be incorporated in matrix C_d , see equation (C.13). To be more precise, c_{11} equals the (1, 1)th element of $\frac{1}{2}HC_d$:

$$c_{11} = \frac{1}{2} \sum_{e(1)} u_e \bar{l}_e = \frac{1}{2} \Omega_1 \left(\sum_{e(1)} u_e \bar{l}_e / \Omega_1 \right) = \frac{1}{2} h_{11} c_{d,11}. \quad (\text{C.30})$$

Hence, in agreement with (C.6b), matrix C_d is a diagonal matrix containing the discrete equivalents of $(\nabla \cdot \mathbf{u})$ at its main diagonal. The off-diagonal elements of C form C_i . It is easy to show that, after writing out the discretization for cells 2, 3 and 4, these off-diagonal elements satisfy:

$$c_{21} = -c_{12}, \quad c_{31} = -c_{13}, \quad c_{41} = -c_{14}, \quad (\text{C.31})$$

showing that, in the interior of the domain, property (C.15) holds.

Consider now the situation as indicated in Figure 4.14b in which face i is a boundary face. The off-diagonal element c_{12} does not exist anymore, while the elements

c_{13} and c_{14} remain unaffected with respect to (C.29), hence matrix C_i remains skew-symmetric when the presence of boundaries is included. Now we focus on the diagonal element. Assume first that at face i the quantity $u_i\phi_i$ is given, hence face i is located at an inflow, solid or freeslip boundary. The corresponding elements in the main diagonal and right-hand side vector are:

$$c_{11} = \frac{1}{2}(u_j\bar{l}_j + u_k\bar{l}_k), \quad r_1 = -u_i\phi_i\bar{l}_i. \quad (\text{C.32})$$

Considering only the contribution of face i , we arrive with help of (C.16) at:

$$\frac{dK_h}{dt} + \frac{1}{2}\phi_1^2(u_j\bar{l}_j + u_k\bar{l}_k) = -u_i\phi_i\bar{l}_i\phi_1. \quad (\text{C.33})$$

With $\phi_i \approx \phi_1$ we get:

$$\frac{dK_h}{dt} + \frac{1}{2}\phi_1^2(u_i\bar{l}_i + u_j\bar{l}_j + u_k\bar{l}_k) \approx -\frac{1}{2}u_i\phi_i^2\bar{l}_i, \quad (\text{C.34})$$

which is clearly the discrete analogon of (C.10) when the presence of other cells is omitted. In the situation that $u_i\phi_i$ is not given, i.e. face i is positioned at an outflow boundary, the corresponding elements in the main diagonal and right-hand side vector are:

$$c_{11} = \frac{1}{2}(u_j\bar{l}_j + u_k\bar{l}_k) + u_i\bar{l}_i, \quad r_1 = 0. \quad (\text{C.35})$$

It is easy to verify that (C.34) also holds for this situation. In conclusion, the central scheme for scalar conservation laws on unstructured staggered grids satisfies (C.15) and mimics definitions (C.6), hence it preserves symmetry.

Momentum equation

Suppose for a moment that the full momentum vector is stored at the cell faces. Discretization of the momentum equation, omitting the pressure and viscous terms, gives

$$\Omega_i \frac{d\mathbf{m}_i}{dt} + \sum_{e(i)} u_e \mathbf{m}_e \bar{l}_e = 0, \quad (\text{C.36})$$

see also equation (4.7). We propose the following central scheme for the momentum vector at face k , with notation as in Figure 4.2:

$$\mathbf{m}_k = \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_3), \quad (\text{C.37})$$

where \mathbf{m}_1 and \mathbf{m}_3 are momentum vectors in cells 1 and 3. If we use

$$\mathbf{m}_3 = \frac{1}{2}(\mathbf{m}_v + \mathbf{m}_w), \quad (\text{C.38})$$

and similar relations in other cells, then among others the following elements in the convection matrix for the x - and y -momentum component appear:

$$c_{iw} = \frac{1}{4}u_k\bar{l}_k \quad c_{wi} = \frac{1}{4}u_k\bar{l}_k \quad c_{ii} = \frac{1}{4}(u_j\bar{l}_j + u_k\bar{l}_k + u_l\bar{l}_l + u_o\bar{l}_o) = \frac{1}{4} \sum_{e(i)} u_e \bar{l}_e. \quad (\text{C.39})$$

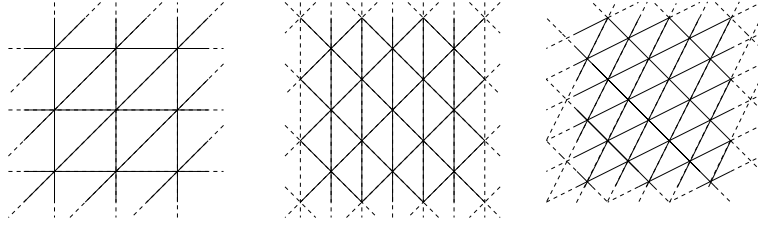


Figure C.1: Some examples of regular grids at which the convection operator for the momentum equation is skew-symmetric.

Note that $c_{iw} = -c_{wi}$ because the sign of \bar{l}_k is in c_{iw} opposite to that in c_{wi} . Furthermore, as was to be expected, the diagonal term corresponds to the divergence of the velocity. Hence, if the momentum vector were stored at the faces and scheme (C.37)–(C.38) were used, the skew-symmetry property (C.15) would be satisfied.

However, we do not store the momentum vector at the faces, but only its normal components. In this situation, the central scheme is given by equation (4.22). Among others, the following elements in the convection matrix for the momentum at face i are present:

$$c_{ii} = \frac{1}{2} \sum_{e(i)} u_e \bar{l}_e, \quad c_{ik} = 0, \quad c_{iv} = \frac{1}{2} u_k \bar{l}_k \eta_v^i, \quad c_{iw} = \frac{1}{2} u_k \bar{l}_k \eta_w^i, \quad (\text{C.40})$$

where the reconstruction coefficients η are defined as follows:

$$\mathbf{N}_i = \eta_v^i \mathbf{N}_v + \eta_w^i \mathbf{N}_w, \quad \mathbf{N}_w = \eta_i^w \mathbf{N}_i + \eta_j^w \mathbf{N}_j. \quad (\text{C.41})$$

One of the elements in the convection matrix for the momentum equation for face w is: $c_{wi} = \frac{1}{2} u_k \bar{l}_k \eta_i^w$. Hence, in order to have the desired skew-symmetric property, the relation

$$\eta_w^i = \eta_i^w \quad (\text{C.42})$$

needs to be satisfied. Unfortunately, this is not true in general. In the special situation in which \mathbf{N}_i and \mathbf{N}_w are parallel and \mathbf{N}_j and \mathbf{N}_v are parallel, the skew-symmetric property holds for the contribution coming from CV-face k . Hence, the central scheme for the momentum equation preserves symmetry on grids that consist of a tessellation of congruent triangles, i.e. when the grid consists of three sets of parallel lines. This is illustrated in Figure C.1. These grids may be viewed as structured grids consisting of identical triangles. In Section 6.3.1 a numerical study of the symmetry-preserving properties of the inertia term is given.

Appendix D

Conservation properties of the discretization

Weak solutions of hyperbolic conservation laws are called genuine if they satisfy the jump conditions; these are called the Rankine-Hugoniot conditions in the case of the Euler equations of gasdynamics. The well-known Lax-Wendroff theorem [66] states that convergent numerical schemes in conservation form converge to a genuine weak solution of the conservation law. Colocated finite volume schemes for conservation laws, i.e. schemes in which all primary unknowns reside in the same nodes, are in conservation form. The proof of the Lax-Wendroff theorem assumes colocated schemes. Because the theorem suggests that conservation is important for satisfying the jump conditions, our staggered scheme is in conservation form for mass and energy, and if applied on Cartesian grids also for momentum. However, on non-Cartesian or unstructured grids, with arbitrarily directed normal momentum components at the cell faces, there is no conservation form, because momentum and pressure balance only in a fixed direction. Therefore we use a generalized concept of conservation for momentum and rely on numerical experiments to show that the jump conditions indeed are satisfied.

Conservation of kinetic energy, circulation and momentum is demonstrated for two different unstructured staggered schemes for incompressible flows by Perot [89]. Perot's scheme, discussed in Section 2.4.3, needs a dual mesh, formed by the associated Voronoi tessellation of the triangular mesh, in his conservation proofs. Hence, a Delaunay triangulation has to be employed. The unstructured staggered scheme described in this thesis does not need a dual mesh, and is therefore not restricted to Delaunay grids. This is advantageous, since in some cases Delaunay triangulations are unacceptable [75] due to, for example, violation of boundary integrity or the presence of obtuse angles. Since our scheme is applied to compressible flows, it is highly desirable that it satisfies the jump conditions. It is the aim of this appendix

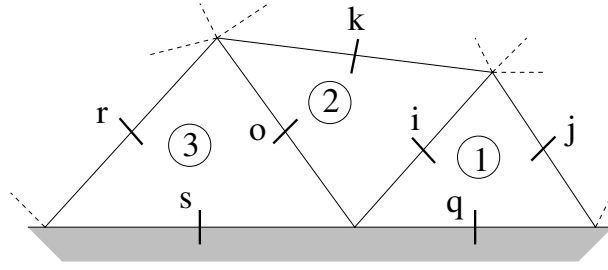


Figure D.1: Part of a staggered grid. Faces are indicated by letters and numbers denote the cells. Faces s and q are boundary faces.

to show that our scheme conserves mass, energy and momentum, which suggests satisfaction of the jump conditions. Numerical experiments that are discussed in Chapter 6 confirm this.

D.1 Conservative discretization of the continuity and energy equation

Consider the conservation law

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0, \quad (\text{D.1})$$

where we have the continuity equation (2.7), the energy equation (2.9) and the Mach-uniform pressure-correction equation (3.51) in mind. Integration over a domain S with boundary ∂S yields:

$$\frac{d\Psi}{dt} + \oint_{\partial S} (\mathbf{u} \cdot \mathbf{n}) \phi \, d\Gamma = 0, \quad \Psi = \int_S \psi \, d\mathbf{x}, \quad (\text{D.2})$$

where \mathbf{n} is the outward normal at ∂S . The only way in which Ψ can change is by having a nonzero net flux over ∂S , which is why (D.1) is called a conservation law. Let S consist of non-overlapping triangles S_c with area Ω_c , and let ∂S be approximated by non-overlapping straight faces ∂S_b with length l_b . Then (D.2) is equivalent to:

$$\sum_c \Omega_c \frac{d\psi_c}{dt} + \sum_b (\mathbf{u}_b \cdot \mathbf{n}_b) \phi_b l_b = 0, \quad (\text{D.3})$$

where

$$\psi_c = \frac{1}{\Omega_c} \int_{S_c} \psi \, d\mathbf{x}, \quad (\mathbf{u}_b \cdot \mathbf{n}_b) \phi_b = \frac{1}{l_b} \int_{\partial S_b} (\mathbf{u} \cdot \mathbf{n}) \phi \, d\Gamma. \quad (\text{D.4})$$

The summations over c and b run over all cells and all boundary faces, respectively. By definition, a numerical scheme that is conservative reduces to (D.3) after summation over all cells. We will prove that the finite volume approximation to (D.1) as proposed in Sections 4.3, 4.4 and 4.5 is conservative. Integration of (D.1) over cell 1, see Figure D.1, leads to

$$\Omega_1 \frac{d\psi_1}{dt} + \sum_{e(1)} (\mathbf{u}_e \cdot \mathbf{n}_e) \phi_e l_e = 0, \quad (\text{D.5})$$

where the summation is over the faces of cell 1, i.e. $e(1) = \{i, j, q\}$, and \mathbf{n}_e is the outward unit normal at face e . We observe that the flux term of each internal face appears in the discretized equations for both adjacent cells, but with opposite sign. For example, $(\mathbf{u}_i \cdot \mathbf{n}_i) \phi_i l_i$ appears in the equations for ψ_1 and ψ_2 . Summation of the discretized equations over all cells then results in cancellation of the flux terms at the internal faces, leaving only a contribution from the boundary faces. This cancellation is essential and leads to expression (D.3), showing that (D.5) is conservative. In this way, a conservation form arises naturally for colocated schemes.

D.2 Conservative discretization of the momentum equation

Integration of the inviscid momentum equation over a domain S with boundary ∂S yields the momentum conservation form:

$$\frac{d}{dt} \int_S \mathbf{m} \, d\mathbf{x} + \oint_{\partial S} [(\mathbf{u} \cdot \mathbf{n}) \mathbf{m} + p \mathbf{n}] \, d\Gamma = \mathbf{0}, \quad (\text{D.6})$$

where \mathbf{n} is the outward unit normal at ∂S . The control volume (CV) for each internal face consists of the union of the two adjacent triangles, whereas the CV for a boundary face consists of the neighboring boundary cell. Assume for the moment that at each triangle face the momentum vector is located; we will return to this later. Integration over the CV for the momentum in face f leads to a discretized momentum vector equation of the following form:

$$\Omega_f \frac{d\mathbf{m}_f}{dt} + \sum_{e(f)} [(\mathbf{u}_e \cdot \mathbf{n}_e) \mathbf{m}_e + p_e \mathbf{n}_e] l_e = \mathbf{0}, \quad (\text{D.7})$$

where $e(f)$ indicates the set of faces of the CV and Ω_f is the area of the CV. For example, in Figure D.1 we have $e(i) = \{j, k, o, q\}$ and $\Omega_i = \Omega_1 + \Omega_2$, and $e(q) = \{i, j, q\}$ and $\Omega_q = \Omega_1$. We will show below that summation of (D.7) over all faces leads to a discrete equivalent of (D.6), hence (D.7) is in conservation form. After that we return to the relation between (D.7) and the discretized equation for normal momentum components: it is the latter that is actually solved.

Time derivative

Summation of the time derivative in (D.7) over all faces in the grid yields:

$$\frac{d}{dt} \left(\sum_f \Omega_f \mathbf{m}_f \right) = \frac{d}{dt} \left(\sum_c \Omega_c \sum_{f(c)} \mathbf{m}_f \right) = 3 \frac{d}{dt} \left(\sum_c \Omega_c \mathbf{m}_c \right), \quad (\text{D.8})$$

where we approximate the momentum vector in the cell centroid by means of:

$$\mathbf{m}_c = \frac{1}{3} \sum_{f(c)} \mathbf{m}_f, \quad (\text{D.9})$$

with $f(c)$ the set of faces of cell c .

Inertia term

The inertia flux term $(\mathbf{u}_i \cdot \mathbf{n}_i) \mathbf{m}_i l_i$ at face i , see Figure D.1, appears in the discretized momentum vector equations for the faces j , k , o and q . The outward unit vector \mathbf{n}_i points in the direction of cell 2 in the equations for faces j and q , and in the opposite direction in the remaining two equations. Hence, summation over all momentum vector equations leads to a cancellation of the inertia flux term at face i . All other flux contributions at internal faces disappear similarly. The inertia term $(\mathbf{u}_q \cdot \mathbf{n}_q) \mathbf{m}_q l_q$ at boundary face q shows up in the momentum vector equations for faces i , j and q . Since this term is not cancelled, we arrive at

$$\sum_f (\mathbf{u}_f \cdot \mathbf{n}_f) \mathbf{m}_f l_f = 3 \sum_b (\mathbf{u}_b \cdot \mathbf{n}_b) \mathbf{m}_b l_b, \quad (\text{D.10})$$

where b refers to the boundary faces.

Pressure term

A completely similar reasoning as used for the inertia term leads for the pressure term to

$$\sum_f p_f \mathbf{n}_f l_f = 3 \sum_b p_b \mathbf{n}_b l_b. \quad (\text{D.11})$$

Of the gradient schemes discussed in Section 4.2.5, only the contour integral formulation is conservative. Nevertheless, so far all numerical experiments done with other gradient schemes resulted in correct shock speeds as well.

Discussion

Summation of (D.7) over all faces and application of (D.8), (D.10) and (D.11) results in, after division by 3:

$$\sum_c \Omega_c \frac{d\mathbf{m}_c}{dt} + \sum_b [(\mathbf{u}_b \cdot \mathbf{n}_b) \mathbf{m}_b + p_b \mathbf{n}_b] l_b = \mathbf{0}, \quad (\text{D.12})$$

which is in conservation form, cf. (D.6). Hence the discretization of the momentum *vector* equation is conservative. The staggered scheme uses only the normal

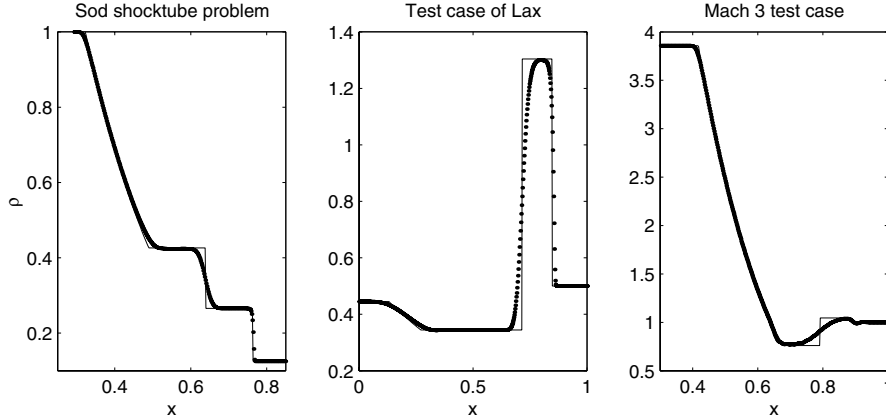


Figure D.2: Fine grid solution for some Riemann problems.

momentum components, hence it uses the projection of (D.7) on the corresponding normal component. The reason to use only the normal components is that we want to avoid undesirable pressure oscillations in the incompressible case. Since the normal vectors can have any direction, summation over all projected momentum equations does in general not lead to cancellation of the internal flux contributions. But the set of the momentum equations solved consists of linear combinations of components of (D.12), and is therefore a subset of the conservative system (D.12). When this is the case, we say that the system is in generalized conservation (GC) form. The Lax-Wendroff theorem [66] does not apply to schemes in GC form. However, it suggests that conservation properties are important, which is why we have introduced the concept of generalized conservation form. In addition, satisfaction of the entropy condition has, for the case of the Euler equations, only been proven for the Osher scheme [28, 84, 83]. Also for our scheme no such proof is available, and in Chapter 6 it is verified for various test cases that our scheme computes the entropy solution. The consequences of having a scheme that is not in GC form are discussed in Appendix D.3.

D.3 Example of a result computed with a nonconservative scheme

The scheme of Bijl [9] is not in conservation form, and this explains the presence of discrepancies in the shock speed as observed on fine grids in [111]. In the previous sections it is demonstrated that the unstructured staggered scheme conserves mass, momentum and energy, and this strongly suggests that computed results will satisfy the Rankine-Hugoniot conditions. In Figure D.2 results of the 2D unstructured

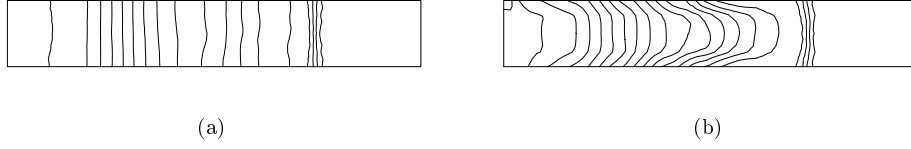


Figure D.3: Density contours resulting from the GC (a) and the not-GC (b) discretization.

staggered schemes, the 1D Roe and AUSM scheme on a fine grid with 550 cells at the centerline are depicted. It is clear that all solutions converge to the entropy solution, and that the shock speeds are correct. For further information, see Section 6.2.

In order to see what loss of conservation might bring, we consider the following, somewhat artificial, change in the discretization. Suppose that the boundary in Figure D.1 is a freeslip wall, which means that the flow is parallel to the boundary. Let us call a face with one vertex at a freeslip wall a quasi-internal face, and a face without a vertex at a wall is called a real-internal face. Assume we do, considering quasi-internal face o , the following. The inertia term $u_o(\mathbf{m}_o \cdot \mathbf{N}_z)\bar{l}_o$ appears in the momentum equations for $z \in \{i, k, r\}$. We will evaluate this term by means of upwind scheme (4.18) when z is a real-internal face, while the boundary condition is used explicitly when z is a quasi-internal face. Explicit use of the boundary condition and using an upwind approximation gives, with the x -axis parallel to the boundary and flow from left to right: $\mathbf{m}_o = (m_{r,x}, 0)$. Here $m_{r,x}$ follows from: $m_r = (\mathbf{m}_r \cdot \mathbf{N}_r) = m_{r,x}N_{r,x}$, hence we explicitly insert the approximation that the flow is parallel to the wall. Consequently, the considered inertia term becomes equal to $u_o m_r N_{z,x} l_o / N_{r,x}$ in the momentum equations for quasi-internal faces $z \in \{i, r\}$. On the other hand, in the momentum equation for real-internal face k the considered inertia term is computed with the standard first order upwind scheme:

$$u_o(\eta_r^k m_r + \eta_s^k m_s)\bar{l}_o, \quad \mathbf{N}_k = \eta_r^k \mathbf{N}_r + \eta_s^k \mathbf{N}_s. \quad (\text{D.13})$$

When at face r the flow is not perfectly aligned with the wall, the cancellation leading to (D.10) does not hold anymore, resulting in loss of conservation. This scheme may look unnatural, but recall that our aim is merely to give a slight modification to the GC (generalized conservative) scheme such that it is not in GC form anymore. The density contours of the solution for the Sod problem are, computed with both the GC and the not-GC scheme on the grid of Figure 6.1, shown in Figure D.3. The undesired influence of the loss of conservation near the freeslip wall is clearly visible, e.g. a wrong shock speed.

Bibliography

- [1] K. Akselvoll and P. Moin. Large eddy simulation of turbulent confined coannular jets. *J. Fluid Mech.*, 315:387–411, 1996.
- [2] B.F. Armaly, F. Durst, J.C.F. Pereira, and B. Schönung. Experimental and theoretical investigation of backward-facing step flow. *J. Fluid Mech.*, 127:473–496, 1983.
- [3] M. Arora and P.L. Roe. A well-behaved TVD limiter for high-resolution calculations of unsteady flow. *J. Comp. Phys.*, 132:3–11, 1997.
- [4] B. Ničeno and E. Nobile. Numerical analysis of fluid flow and heat transfer in periodic wavy channels. *International Journal of Heat and Fluid Flow*, 22:156–167, 2001. Accepted for publication.
- [5] T.J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA Paper 93-0668, 1993.
- [6] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, 1989.
- [7] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 1967.
- [8] H. Bijl. *Computation of flow at all speeds with a staggered scheme*. PhD thesis, Delft University of Technology, The Netherlands, February 1999.
- [9] H. Bijl and P. Wesseling. A unified method for computing incompressible and compressible flows in boundary-fitted coordinates. *J. Comp. Phys.*, 141:153–173, 1998.
- [10] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24:162–166, 1981.
- [11] V. Casulli. Semi-implicit finite difference methods for the two-dimensional shallow water equations. *J. Comp. Phys.*, 86:56–74, 1990.

-
- [12] J.C. Cavendish, C.A. Hall, and T.A. Porsching. A complementary volume approach for modelling three-dimensional Navier-Stokes equations using dual Delaunay/Voronoi tessellations. *Int. J. Num. Meth. Heat Fluid Flow*, 4:329–345, 1994.
- [13] C.T. Chan and K. Anastasiou. Solution of incompressible flows with or without a free surface using the finite volume method on unstructured triangular meshes. *Int. J. Num. Meth. in Fluids*, 29:35–57, 1999.
- [14] A.J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comp.*, 22:745–762, 1968.
- [15] A.J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comp.*, 23:342–353, 1969.
- [16] B. Cockburn, S. Hou, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Math. Comp.*, 54:545–581, 1990.
- [17] B. Cockburn, S.-Y. Lin, and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. *J. Comp. Phys.*, 84:90–113, 1989.
- [18] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comp.*, 52:411–435, 1989.
- [19] D.L. Darmofal. Towards a robust multigrid algorithm with Mach number and grid independent convergence. In K.D. Papailiou, D. Tsahalis, J. Périaux, and D. Knörzer, editors, *Computational Fluid Dynamics '98*, volume 2, pages 90–95, Chichester, 1998. Wiley.
- [20] L. Davidson. A pressure correction method for unstructured meshes with arbitrary control volumes. *Int. J. Num. Meth. in Fluids*, 22:265–281, 1996.
- [21] B. de Foy and W. Dawes. Unstructured pressure-correction solver based on a consistent discretization of the Poisson equation. *Int. J. Num. Meth. in Fluids*, 34:463–478, 2000.
- [22] I. Demirdžić, Z. Lilek, and M. Perić. A collocated finite volume method for predicting flows at all speeds. *Int. J. Num. Meth. in Fluids*, 16:1029–1050, 1993.
- [23] A. Dervieux, editor. *Numerical Simulation of Compressible Euler Flows*, 1989. Notes on Numerical Fluid Mechanics 26.

- [24] G.K. Despotis and S. Tsangaris. Fractional step method for solution of incompressible Navier-Stokes equations on unstructured triangular meshes. *Int. J. Num. Meth. in Fluids*, 20:1273–1288, 1995.
- [25] G.K. Despotis and S. Tsangaris. A fractional step method for unsteady incompressible flows on unstructured meshes. *Int. J. Comp. Fluid Dyn.*, 8:11–29, 1997.
- [26] E. Oñate, S.R. Idelsohn, and M.A. Storti. Meshless approximations and lagrangian formulations to solve free-surface flows. In E. Oñate, G. Bugeda, and B. Suárez, editors, *Proceedings European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000, Barcelona, September 11-14, 2000*, Barcelona, 2000. CIMNE.
- [27] E. Oñate, S.R. Idelsohn, O.C. Zienkiewicz, and R.L. Taylor. A finite point method in computational mechanics. applications to convective transport and fluid flow. *Int. J. Num. Meth. in Eng.*, 39:3839–3886, 1996.
- [28] B. Engquist and S. Osher. One-sided difference approximations for nonlinear conservation laws. *Math. Comp.*, 36:321–351, 1981.
- [29] D.J. Ewing, A.J. Fawkes, and J.R. Griffiths. Rules governing the numbers of nodes and elements in a finite element mesh. *Int. J. Num. Meth. in Eng.*, 2:597, 1970.
- [30] J.H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, Berlin, 1996.
- [31] R.W. Fox and A.T. McDonald. *Introduction to Fluid Mechanics*. Wiley, London, 1985.
- [32] O. Friedrichs. Weighted essentially non-oscillatory schemes for the interpolation of means values on unstructured grids. *J. Comp. Phys.*, 144:194–212, 1998.
- [33] D.K. Gartling. A test problem for outflow boundary conditions - flow over a backward facing step. *Int. J. Num. Meth. in Fluids*, 11:953–967, 1990.
- [34] U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comp. Phys.*, 48:387–411, 1982.
- [35] S.K. Godunov. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik*, 47:271–306, 1959. (in Russian).

- [36] H. Guillard and C. Viozat. On the behavior of upwind schemes in the low Mach number limit. *Computers and Fluids*, 28:63–86, 1999.
- [37] M. Hafez, J. South, and E. Murmann. Artificial compressibility methods for numerical solution of transonic full potential equation. *AIAA J.*, 17:838–844, 1979.
- [38] C.A. Hall, J.C. Cavendish, and W.H. Frey. The dual variable method for solving fluid flow difference equations on Delaunay triangulations. *Computers and Fluids*, 20:145–164, 1991.
- [39] C.A. Hall and T.A. Porsching. A characteristic-like method for thermally expandable flow on unstructured triangular grids. *Int. J. Num. Meth. in Fluids*, 22:731–754, 1996.
- [40] C.A. Hall, T.A. Porsching, and P. Hu. Covolume-dual variable method for thermally expandable flow on unstructured triangular grids. *Int. J. Comp. Fluid Dyn.*, 2:111–139, 1994.
- [41] C.A. Hall, T.A. Porsching, and G.L. Mesina. On a network method for unsteady incompressible fluid flow on triangular grids. *Int. J. Num. Meth. in Fluids*, 15:1383–1406, 1992.
- [42] F.H. Harlow and A.A. Amsden. Numerical calculation of almost incompressible flows. *J. Comp. Phys.*, 3:80–93, 1968.
- [43] F.H. Harlow and A.A. Amsden. A numerical fluid dynamics calculation method for all flow speeds. *J. Comp. Phys.*, 8:197–213, 1971.
- [44] F.H. Harlow and J.E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *The Physics of Fluids*, 8:2182–2189, 1965.
- [45] A. Harten. High resolution total-variation-stable finite-difference schemes. *SIAM J. Num. Anal.*, 21:1–23, 1984.
- [46] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarty. Uniformly high order accurate essentially non-oscillatory schemes, III. *J. Comp. Phys.*, 71:231–303, 1987.
- [47] A. Harten, J.M. Hyman, and P.D. Lax. On finite difference approximations and entropy conditions for shocks. *Comm. Pure and Appl. Math.*, 29:297–322, 1976. (with appendix by B. Keyfitz).
- [48] C. Hirsch. *Numerical Computation of Internal and External Flows. Vol.2: Computational Methods for Inviscid and Viscous Flows*. Wiley, Chichester, 1990.

-
- [49] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comp. Phys.*, 14:227–253, 1974.
- [50] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comp. Phys.*, 150:97–127, 1999.
- [51] M.E. Hubbard. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *J. Comp. Phys.*, 155:54–74, 1999.
- [52] Y.-H. Hwang. Calculations of incompressible flow on a staggered triangular grid, part I: mathematical formulation. *Num. Heat Transfer B*, 27:323–336, 1995.
- [53] Y.-H. Hwang. Calculations of incompressible flow on a staggered triangular grid, part II: mathematical formulation. *Num. Heat Transfer B*, 27:337–353, 1995.
- [54] A. Jameson, T.J. Baker, and N.P. Weatherill. Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103, 1986.
- [55] P. Jawahar and H. Kamath. A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids. *J. Comp. Phys.*, 164:165–203, 2000.
- [56] Y. Jiang and A.J. Pzrekwas. Implicit, pressure-based incompressible Navier-Stokes equations solver for unstructured meshes. AIAA Paper 94-0305, 1994.
- [57] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, 59:308–323, 1985.
- [58] S. Klainerman and A. Majda. Compressible and incompressible flows. *Comm. Pure and Appl. Math.*, 35:629–651, 1982.
- [59] R. Klein. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics I: one-dimensional flow. *J. Comp. Phys.*, 121:213–237, 1995.
- [60] M.H. Kobayashi, J.M.C. Pereira, and J.C.F. Pereira. A conservative finite-volume second-order accurate projection method on hybrid unstructured grids. *J. Comp. Phys.*, 150:40–75, 1999.
- [61] B. Koren. Improving Euler computations at low Mach numbers. *Int. J. Comp. Fluid Dyn.*, 6:51–70, 1996.
- [62] B. Koren and P.W. Hemker. Multi-D upwinding and multigriding for steady Euler flow computations. In J.B. Vos, A. Rizzi, and I.L. Rhyming, editors, *Proceedings of the Ninth GAMM-Conference on Numerical Methods in Fluid*

- Mechanics*, pages 89–98, Braunschweig, 1992. Vieweg. Notes on Numerical Fluid Mechanics 35.
- [63] B. Koren and B. van Leer. Analysis of preconditioning and multigrid for Euler flows with low-subsonic regions. *Advances in Comp. Meth.*, 4:127–144, 1995.
- [64] S. Koshizuka, A. Nobe, and Y. Oka. Numerical analysis of breaking waves using the moving particle semi-implicit method. *Int. J. Num. Meth. in Fluids*, 26:751–769, 1998.
- [65] S. Koshizuka and Y. Oka. Moving particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear engineering science*, 123:421–434, 1996.
- [66] P. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure and Appl. Math.*, 13:217–237, 1960.
- [67] P.D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical approximation. *Comm. Pure and Appl. Math.*, 7:159–193, 1954.
- [68] H. Le, P. Moin, and J. Kim. Direct numerical simulation of turbulent flow over a backward-facing step. *J. Fluid Mech.*, 330:349–374, 1997.
- [69] F.S. Lien. A pressure-based unstructured grid method for all-speed flows. *Int. J. Num. Meth. in Fluids*, 33:355–374, 2000.
- [70] M.-S. Liou and C.J. Steffen. A new flux splitting scheme. *J. Comp. Phys.*, 107:23–39, 1993.
- [71] S.H. Lo. Volume discretization into tetrahedra - part I. Verification and orientation of boundary surfaces. *Computers and Structures*, 39:493–500, 1991.
- [72] S.H. Lo. Volume discretization into tetrahedra - part II. 3D triangulation by advancing front approach. *Computers and Structures*, 39:501–511, 1991.
- [73] R. Lohner. Progress in grid generation via the advancing front technique. *Engineering with Computers*, 12:186–210, 1996.
- [74] R. Lohner, P. Parikh, and C. Gumbert. Interactive generation of unstructured grid for three dimensional problems. In *Numerical Grid Generation in Computational Fluid Mechanics*, pages 687–697. Pineridge Press, 1988.
- [75] D.J. Mavriplis. Unstructured grid techniques. *Annu. Review Fluid Mechanics*, 29:473–514, 1997.
- [76] J.A. Meijerink and H.A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Math. Comp.*, 31:148–162, 1977.

- [77] F. Moukalled and M. Darwish. A high-resolution pressure-based algorithm for fluid flow at all speeds. *J. Comp. Phys.*, 168:101–133, 2001.
- [78] C. Moulinec and P. Wesseling. Colocated schemes for the incompressible Navier-Stokes equations on non-smooth grids for two-dimensional problems. *Int. J. Num. Meth. in Fluids*, 32:349–364, 2000.
- [79] R.A. Nicolaides. Flow discretization by complementary volume schemes. AIAA paper 89-1978, Buffalo, New York, June 1989.
- [80] R.A. Nicolaides. The covolume approach to computing incompressible flows. In M.D. Gunzburger and R.A. Nicolaides, editors, *Incompressible Computational Fluid Dynamics*, pages 295–333, Cambridge, UK, 1993. Cambridge University Press.
- [81] R.A. Nicolaides, T.A. Porsching, and C.A. Hall. Covolume methods in computational fluid dynamics. In M. Hafez and K. Oshima, editors, *Computational Fluid Dynamics Review 1995*, pages 279–299, Chichester, 1995. Wiley.
- [82] P.J. O’Rourke and M.S. Sahota. A variable explicit/implicit numerical method for calculating advection on unstructured meshes. *J. Comp. Phys.*, 143:312–345, 1998.
- [83] S. Osher and S. Chakravarthy. Upwind schemes and boundary conditions with applications to Euler equations in general geometries. *J. Comp. Phys.*, 50:447–481, 1983.
- [84] S. Osher and F. Solomon. Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Comp.*, 38:339–374, 1982.
- [85] S. Owen. A survey of unstructured mesh generation technology. Available at <http://www.andrew.cmu.edu/user/sowen/survey/index.html>.
- [86] D. Pan and J.-C. Cheng. Upwind finite-volume navier-stokes computations on unstructured triangular meshes. *AIAA Journal*, 9:1618–1625, 1993.
- [87] D. Pan, C.-H. Lu, and J.-C. Cheng. Incompressible flow solution on unstructured triangular meshes. *Num. Heat Transfer B*, 26:207–224, 1994.
- [88] S.V. Patankar and D.B. Spalding. A calculation procedure for heat and mass transfer in three-dimensional parabolic flows. *Int. J. Heat and Mass Transfer*, 15:1787–1806, 1972.
- [89] J.B. Perot. Conservation properties of unstructured staggered mesh schemes. *J. Comp. Phys.*, 159:58–89, 2000.
- [90] C.M. Rhie and W.L. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA J.*, 21:1525–1532, 1983.

-
- [91] S. Rida, F. McKenty, F.L. Meng, and M. Reggio. A staggered control volume scheme for unstructured triangular grids. *Int. J. Num. Meth. in Fluids*, 25:697–717, 1997.
- [92] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comp. Phys.*, 43:357–372, 1981.
- [93] 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.
- [94] G.E. Schneider and M.J. Raw. A skewed positive influence coefficient procedure for control-volume-based finite-element convection-diffusion computation. *Num. Heat Transfer*, 9:1–28, 1986.
- [95] Guus Segal. *SEPRAN manuals*. Leidschendam, 1984.
- [96] M. Shashkov, B. Swartz, and B. Wendroff. Local reconstruction of a vector field from its normal components on the faces of grid cells. *J. Comp. Phys.*, 139:406–409, 1998.
- [97] M.S. Shephard and M.K. Georges. Three-dimensional mesh generation by finite octree technique. *Int. J. Num. Meth. in Eng.*, 32:702–749, 1991.
- [98] C.-W. Shu. High order finite difference and finite volume WENO schemes and discontinuous Galerkin methods for CFD. ICASE 01-11, NASA Langley Research Center, Hampton, Virginia, 2001.
- [99] W. Shyy, M.-H. Chen, and C.-S. Sun. Pressure-based multigrid algorithm for flow at all speeds. *AIAA J.*, 30:2660–2669, 1992.
- [100] G.A. Sod. A survey of several finite difference methods for systems of nonlinear conservation laws. *J. Comp. Phys.*, 27:1–31, 1978.
- [101] J.L. Sohn. Evaluation of FIDAP on some classical laminar and turbulent benchmarks. *Int. J. Num. Meth. in Fluids*, 8:1469–1490, 1988.
- [102] G.S. Stelling. *On the construction of computational methods for shallow water flow problems*. PhD thesis, Delft University of Technology, 1983. Also appeared as Rijkswaterstaat Communications 35, 1984. Rijkswaterstaat, The Hague.
- [103] M. Thomadakis and M. Leschziner. A pressure-correction method for the solution of incompressible viscous flows on unstructured grids. *Int. J. Num. Meth. in Fluids*, 22:581–601, 1996.
- [104] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, Berlin, 2nd edition, 1999.

-
- [105] E. Turkel. Preconditioning techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 31:385–416, 1999.
- [106] E. Turkel, A. Fiterman, and B. van Leer. Preconditioning and the limit to the incompressible flow equations. Report 93-42, ICASE, NASA Langley, 1993.
- [107] E. Turkel, R. Radespiel, and H. Kroll. Assessment of preconditioning methods for multidimensional aerodynamics. *Computers and Fluids*, 26:613–634, 1997.
- [108] P. van Beek, R.R.P. van Nooyen, and P. Wesseling. Accurate discretization on non-uniform curvilinear staggered grids. *J. Comp. Phys.*, 117:364–367, 1995.
- [109] J.W. van der Burg, B. Oskam, T. Gerhold, M. Galle, T. Berglind, B. Arlinger, G. Kretzschmar, W. Haase, and P. Wesseling. Fastflo II project programme. Technical report, NLR - TR - 98052, 1998.
- [110] D.R. van der Heul, C. Vuik, and P. Wesseling. A staggered scheme for hyperbolic conservation laws applied to unsteady sheet cavitation. *Computing and Visualization in Science*, 2:63–68, 1999.
- [111] D.R. van der Heul, C. Vuik, and P. Wesseling. A conservative pressure correction method for compressible flow at all speeds. *Int. J. Num. Meth. in Fluids*, 2001. submitted.
- [112] J.J.W. van der Vegt and H. van der Ven. Discontinuous galerking finite element method with anisotropic local grid refinement for inviscid compressible flows. *J. Comp. Phys.*, 141:46–77, 1998.
- [113] H.A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for solution of non-symmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 13:631–644, 1992.
- [114] E. van der Weide. *Compressible flow simulation on unstructured grids using multi-dimensional upwind schemes*. PhD thesis, Delft University of Technology, The Netherlands, 1998.
- [115] J.P. Van Doormaal and G.D. Raithby. Enhancements of the SIMPLE method for predicting incompressible fluid flows. *Num. Heat Transfer*, 7:147–163, 1984.
- [116] 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.
- [117] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *J. Comp. Phys.*, 32:101–136, 1979.
- [118] B. van Leer. Flux-vector splitting for the Euler equations. In E. Krause, editor, *Eighth International Conference on Numerical Methods in Fluid Dynamics*, pages 507–512, Berlin, 1982. Springer. Lecture Notes in Physics 170.

-
- [119] A.E.P. Veldman and K. Rinzema. Playing with nonuniform grids. *J. Eng. Math.*, 26:119–130, 1992.
- [120] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J. Comp. Phys.*, 118:120–130, 1995.
- [121] V. Venkatakrishnan. Perspective on unstructured grid flow solvers. *AIAA Journal*, 34:533–547, 1996.
- [122] R.C.W. Verstappen and A.E.P. Veldman. Spectro-consistent discretization of Navier-Stokes: a challenge to RANS and DNS. *J. of Engineering Math.*, 34:162–179, 1998.
- [123] R.W.C.P. Verstappen and A.E.P. Veldman. Direct numerical simulation of turbulence at lower costs. *J. Eng. Math.*, 32:143–159, 1997.
- [124] G. Volpe. Performance of compressible flow codes at low Mach numbers. *AIAA J.*, 31:49–56, 1993.
- [125] C. Vuik. Fast iterative solvers for the discretized incompressible Navier-Stokes equations. *Int. J. Num. Meth. in Fluids*, 22:195–210, 1996.
- [126] D.F. Watson. Computing the n -dimensional Delaunay tessellation with application to Voronoi polytopes. *The computer journal*, 24:167–172, 1981.
- [127] J.K. Watterson. A pressure-based flow solver for the three-dimensional Navier-Stokes equations on unstructured and adaptive meshes. AIAA Paper 94-2358, 1994.
- [128] J.M. Weiss and W.A. Smith. Preconditioning applied to variable and constant density flows. *AIAA J.*, 33:2050–2057, 1995.
- [129] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, Heidelberg, 2001.
- [130] P. Wesseling, A. Segal, and C.G.M. Kassels. Computing flows on general three-dimensional nonsmooth staggered grids. *J. Comp. Phys.*, 149:333–362, 1999.
- [131] P. Wesseling, A. Segal, C.G.M. Kassels, and H. Bijl. Computing flows on general two-dimensional nonsmooth staggered grids. *J. Eng. Math.*, 34:21–44, 1998.
- [132] P. Wesseling and D.R. van der Heul. Uniformly effective numerical methods for hyperbolic systems. *Computing*, 66:249–267, 2001.
- [133] P. Wesseling, M. Zijlema, A. Segal, and C.G.M. Kassels. Computation of turbulent flow in general domains. *Math. Comp. Sim.*, 44:369–385, 1997.

-
- [134] M. Williams. The solution of the two-dimensional incompressible flow equations on unstructured triangular meshes. *Num. Heat Transfer B*, 23:309–325, 1993.
- [135] J.A. Wright and R.W. Smith. An edge-based method for the incompressible Navier-Stokes equations on polygonal meshes. *J. Comp. Phys.*, 169:24–43, 2001.
- [136] M.A. Yerry and M.S. Shephard. Three-dimensional mesh generation by modified octree technique. *Int. J. Num. Meth. in Eng.*, 20:1965–1990, 1984.
- [137] H.Y. Yoon, S. Koshizuka, and Y. Oka. A particle-gridless hybrid method for incompressible flows. *Int. J. Num. Meth. in Fluids*, 30:407–424, 1999.
- [138] H. Yoshihara and P. Sacher. Test cases for inviscid flow field methods. AGAR-Dograph No. 211, AGARD, Neuilly-sur-Seine, France, 1985.

Curriculum vitae

Ivo Wenneker werd op 17 mei 1973 geboren te Groenlo. Van 1985 tot 1991 doorliep hij het VWO aan de scholengemeenschap Marianum in dezelfde plaats.

In augustus 1991 begon hij met de studie Technische Natuurkunde aan de Technische Universiteit Twente. Binnen de vakgroep Vaste Stoffysica van de Universiteit van Århus (Denemarken) verrichtte hij van januari tot mei 1996 numeriek onderzoek naar de elektrische veld gradiënt en de geldigheid van de 'local density approximation' in het materiaal ϵ -FeSi. Binnen de afdeling Fysische Chemie van de Universiteit van Pavia (Italië) deed hij experimenteel onderzoek naar het elektrische geleidingsvermogen van de supergeleider $\text{SmBa}_2\text{Cu}_3\text{O}_{7-\delta}$ als functie van de chemische samenstelling en de temperatuur. In september 1997 studeerde hij met lof af bij prof.dr.ir. Bene Poelsema van de vakgroep Vaste Stoffysica op de ontwikkeling van een softwarepakket dat de optische reflectie en transmissie aan oppervlakken van vaste stoffen kan doorrekenen.

Vanaf november 1997 was de auteur als assistent in opleiding (later: promovendus) verbonden aan de Technische Universiteit Delft bij de afdeling Toegepaste Analyse binnen de faculteit Informatietechnologie en Systemen. Aldaar heeft hij, onder supervisie van prof.dr.ir. Piet Wesseling en ir. Guus Segal, werk verricht dat tot de totstandkoming van dit proefschrift heeft geleid. Tijdens de eerste paar jaar van het onderzoek werd deel uitgemaakt van het Fastflo II project, tezamen met het NLR (Nationaal Lucht- en Ruimtevaartlaboratorium NLR), DLR (Deutsche Forschungsanstalt für Luft- und Raumfahrt e.V.), FFA (Flygtekniska Försöksanstalten), Saab Scania A.B., Daimler Benz Aerospace AG en Ingenieur Büro Kretschmar. Gedurende de onderzoeksperiode werd werk gepresenteerd op conferenties in Spanje (ECCOMAS 2000) en Italië (AMIF 2000 en ENUMATH 2001) en op diverse symposia in Nederland. Naast zijn promotiewerkzaamheden heeft hij zich als commissaris Interne Zaken ingezet voor de Delftse promovendi in het bestuur van PromooD (Promovendi Overleg Delft) en is hij een jaar lang voorzitter geweest van het J.M. Burgerscentrum AIO-overleg.

Per 1 januari 2002 werkt de auteur als onderzoeker/adviseur bij WL|Delft Hydraulics in zowel de sector Mariene, Kust- en Industriële infrastructuur als de afdeling Speurwerk & Onderzoek.

Publikaties:

- N.E. Christensen, I. Wenneker, A. Svane and M. Fanciulli. Electronic Structures of Semiconductors under Pressure. *Physica Stat. Sol. (b)*, 198, 23 (1996).
- M. Fanciulli, A. Zankevich, I. Wenneker, A. Svane, N.E. Christensen and G. Weyer. The Electric Field Gradient at The Fe Nucleus in ϵ -FeSi. *Phys. Rev. B* 54, 15985 (1996).
- Ivo Wenneker. Berekening van compressibele stromingen op een staggered, ongestructureerd grid. *Symposium Wiskunde Toegepast (Maastricht 27-04-2000)*.
- I. Wenneker, A. Segal and P. Wesseling, Computation of compressible flows on unstructured staggered grids. *Proceedings ECCOMAS 2000 (Barcelona, 11-14 Sept. 2000)*.

- I. Wenneker, A. Segal and P. Wesseling, An unstructured staggered scheme for the Navier-Stokes equations. *Proceedings ENUMATH 2001 (Ischia, 23-28 July 2001)*.
- I. Wenneker, A. Segal and P. Wesseling. Conservation properties of a new unstructured staggered scheme. Accepted for publication in *Computers & Fluids*.
- I. Wenneker, A. Segal and P. Wesseling. A Mach-uniform unstructured staggered grid method. Submitted to *International Journal for Numerical Methods in Fluids*.

Dankwoord

Ondanks het gegeven dat het werken aan een proefschrift uiteindelijk een solitaire bezigheid is, wil dat niet zeggen dat mijn omgeving geen belangrijke, zeg maar essentiële bijdrage heeft geleverd aan de totstandkoming ervan. Ik wil op deze plaats enige mensen noemen zonder wie dit project niet geslaagd zou zijn.

In de eerste plaats wil ik mijn promotor Piet bedanken. Met jouw jeugdige enthousiasme en volwassen scherpzinnigheid heb je een onmiskenbaar stempel op mijn proefschrift en mijn periode aan de TUD gedrukt. Guus, jouw vermogen om in minder dan geen tijd de vinger op de pijnlijke plek te leggen, daarbij een flinke dosis scepsis niet schuwend, heeft me steeds gedwongen de zaken goed te overdenken. Ook je onovertroffen hulp bij alles wat met het schrijven en onderhouden van onze software te maken had, zal ik straks zeker missen. Kortom, zonder jullie hulp zou dit boekje niet alleen dunner zijn, maar tevens meer losse flodders bevatten. De lineaire solvers van Kees heb ik, ondanks de vele miljarden operaties die ze voor mij hebben uitgevoerd, nog niet op een fout kunnen betrappen. Ook onze samenwerking op onderwijsgebied is mij goed bevallen. Ruud, zoals je ziet is het project, dat nog in het kinderstadium was toen jij naar Civiele Techniek ging, flink gevorderd. Nog bedankt voor de vele gezellige uren. Kees, jouw fenomenale vermogens om met computers te werken tarten iedere beschrijving, terwijl jouw uitgesproken visie over goed (Linux) versus kwaad (M\$ Windows) mij altijd in hoge mate geamuseerd heeft. Dragan, succes met het ontwikkelen van een nauwkeuriger versie van de huidige methode. Fred, ik hoop je nog vaak tegen te komen in het fitness-honk, het zwembad of tijdens een loop. Verder hebben mijn (oud) mede-promovendi bij wiskunde altijd voor de vrolijke, serieuze en sportieve noot gezorgd: veel plezier gewenst in de rest van jullie loopbaan. Ook denk ik met veel plezier terug aan de PhDays-weekenden, de BC-AIO vergaderingen en de PromooD bijeenkomsten en borrels.

Gelukkig is er ook nog een wereld die weinig of niets begrijpt van het werk dat mij de afgelopen jaren heeft beziggehouden. En dat heeft mij met beide benen op de grond gehouden. Ik denk hierbij aan mijn studievrienden uit Twente, die nu uitgewaaierd zijn over heel Nederland, of nog verder. Verder wil ik mijn kameraden uit Ruurlo en omstreken, die ik ondertussen al heel wat jaartjes ken, niet vergeten. Samen hebben we heel wat concerten, biertjes en onzinnige gesprekken afgewerkt, en ik ga er van uit dat deze goede dingen des levens blijven. Verder ben ik heel blij dat Bjorn, Ilse en mijn nichtje Lisa gelukkig zijn. Tenslotte wil ik mijn ouders, die het de laatste tijd niet gemakkelijk gehad hebben, bedanken voor alle steun en geluk die ze mij hebben gegeven.

Ivo

