Superlinearly Convergent Unstructured Staggered Schemes for Compressible and Incompressible Flows

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 14 februari 2005 om 15:30 uur

door

Dragan VIDOVIĆ
diplomirani matematičar, Univerzitet u Beogradu

geboren te Banja Luka, Joegoslavië.

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

Samenstelling promotiecomissie:

Rector Magnificus,	voorzitter
Prof.dr.ir. P. Wesseling,	Technische Universiteit Delft, promotor
Ir. A. Segal,	Technische Universiteit Delft, toegevoegd promotor
Prof.dr.ir. A.W. Heemink,	Technische Universiteit Delft
Prof.dr.ir. B. Koren,	Technische Universiteit Delft
Prof.dr. P. W. Hemker,	Universiteit van Amsterdam
Prof.dr.ir. H.W.M. Hoeijmakers,	Universiteit Twente
Prof.dr. B. Perot,	University of Massachusetts

Superlinearly Convergent Unstructured Staggered Schemes for Compressible and Incompressible Flows.
Dissertation at Delft University of Technology.

The work described in this thesis was financially supported by the Netherlands Technology Foundation (STW).

ISBN 90-9018969-6

Summary

The subject of this thesis is the accurate computation of fluid flow at speeds ranging from zero to supersonic. At low Mach number (which is the ratio between the velocity of the flow and the sound speed) the flow is incompressible, but when the Mach number rises (above 0.2, say, depending on the geometry) compressibility effects make themselves felt. Because of different mathematical properties of the governing equations, classical methods for flow computation assume either incompressible or compressible flow. Hence, for flows containing both compressible and incompressible flow regions, such as the flow in internal combustion engines or flow around aircraft in take-off or landing conditions, classical methods cannot be used, and methods with accuracy and efficiency uniform in the Mach number (Mach-uniform methods) are required. In this thesis, we use such a Mach-uniform method. Its principles are described in Chapters 2 and 3. To ensure good accuracy and efficiency at low Mach numbers, the point of departure is the classical incompressible flow method that uses staggered grids. This method is generalized to the compressible flow case.

The major issue addressed in this thesis is the development of a Mach-uniform method that uses a staggered scheme on planar unstructured grids with triangular cells, that has a superlinear rate of convergence. This is the topic of Chapter 4. In our staggered grid, the normal velocity components are associated with cell faces, whereas scalar variables are assigned to cell centers. In unstructured grids, the number of cells that share a vertex is arbitrary and varying; in structured grids this number is constant. Unstructured grid schemes are more complicated than structured grid schemes, but their important advantage is that unstructured grids are much easier to generate in geometrically complicated domains than structured grids.

In order to obtain a superlinearly convergent scheme, it is necessary to interpolate vector fields from staggered components with sufficient accuracy in a robust way. Several methods to achieve this objective are investigated in Chapter 4.

Like all higher order methods, the superlinearly convergent scheme obtained suffers from unphysical oscillations near discontinuities such as shocks. This difficulty is overcome by combining two approaches. We use an upwind-biased superlinearly convergent scheme, so that oscillations are damped in smooth parts of the flow. Furthermore, we follow the usual flux limiting approach to combat oscillations near discontinuities, extended to unstructured staggered schemes. An oscillation detecting limiter function is used to switch between the superlinearly convergent scheme and a first order scheme that generates (almost) no oscillations. For this we adapt the limiter of Barth and Jespersen.

The resulting scheme is validated in Chapter 5. Simple incompressible inviscid flows for which the exact solution is known are computed, and the accuracy is assessed. Next, the incompressible Navier-Stokes equations are solved for the exact Kovasznay problem solution, and the behavior of the error as the grid is refined is studied. The backward facing step and circular cylinder flows are computed and comparison is made with some known properties. Next, the Euler equations are solved for some compressible flow problems with our Mach-uniform scheme. The Ringleb flow is a transonic flow with known exact solution. The accuracy of the superlinearly convergent scheme with and without limiting is assessed. Flow in a channel with a bump is computed at low and high

Mach number. A transonic airfoil flow for which the pressure distribution is a fairly well-known test case is computed. To determine carefully whether the Rankine-Hugoniot conditions are satisfied by the scheme, a shocktube problem is computed.

Concluding remarks are presented in Chapter 6. An outlook to extension to three dimensions is given. The prospects for vector field interpolation from staggered components to non-triangular and non-tetrahedral grids is discussed. We conclude that a Mach-uniform unstructured staggered scheme with superlinear rate of convergence has been obtained.

Superlineair Convergerende Ongestructureerde "Staggered" Schema's voor Samendrukbare en Onsamendrukbare Stromingen

Samenvatting

Het onderwerp van dit proefschrift is de nauwkeurige berekening van vloeistofstromingen met snelheden varierend van nul tot supersoon. Bij een laag getal van Mach (dit is de verhouding tussen de stroomsnelheid en de geluidssnelheid) is de stroming onsamendrukbaar, maar als het Machgetal stijgt (zeg boven 0,2) dan worden samendrukbaarheidseffecten merkbaar. Vanwege de verschillende wiskundige eigenschappen van de bewegingsvergelijkingen wordt bij klassieke methoden voor stromingsberekening of onsamendrukbare stroming of samendrukbare stroming aangenomen. Derhalve kunnen klassieke methoden niet worden gebruikt voor stromingen waarin zowel samendrukbare als onsamendrukbare deelgebieden voorkomen, zoals in de stroming in inwendige verbrandingsmotoren of de omstroming van vliegtuigen in start- of landingsconfiguratie. Voor dit type toepassingen zijn methoden vereist met nauwkeurigheid en eficiëntie uniform in het Machgetal (Mach-uniforme methoden). In dit proefschrift wordt zo'n Mach-uniforme methode gebruikt. De principes van deze methode worden weergegeven in de hoofdstukken 2 en 3. Om een goede nauwkeurigheid en eficiëntie te waarborgen voor lage Machgetallen, wordt als vertrekpunt gekozen voor de klasieke methode voor onsamendrukbare stromingen, die een "staggered" rooster gebruikt. Deze methode wordt gegeneraliseerd naar het samendrukbare geval.

De belangrijkste kwestie waar dit proefschrift zich op richt is de ontwikkeling van een Mach-uniforme methode die een "staggered" schema gebruikt op ongestructureerde roosters met driehoekige cellen, en die superlineair convergeert. Dit is het onderwerp van hoofdstuk 4. In ons "staggered" rooster zijn de normaalcomponenten van de snelheid geassociërd met celwanden, terwijl de scalaire onbekenden geassociërd zijn met de cel centra. In ongestructureerde roosters is het aantal cellen dat bij een hoekpunt hoort willekeurig en variabel; in een gestructureerd rooster is dit aantal constant. Ongestructureerde schema's zijn ingewikkelder dan gestructureerde schema's, maar hebben het belangrijke voordeel dat ongestructureerde rooster veel gemakkelijker gegenereerd kunnen worden in geometrisch gecompliceerde gebieden dan gestructureerde roosters.

Teneinde een superlineair schema te verkrijgen is het nodig vectorvelden te interpoleren vanuit normaalcomponenten op de celwanden. Dit dient met voldoende nauwkeurigheid op robuuste wijze te geschieden. Enkele methoden om dit doel te bereiken worden onderzocht in hoofdstuk 4.

Als alle hogere orde methoden lijdt het superlineair convergerende schema aan onfysische oscillaties nabij discontinuiteiten, zols schokken. Deze moeilijkheid wordt overwonnen via twee wegen. We gebruiken een upwind-bias in het superlineair convergerende schema, zodat oscillaties demping ondergaan in gebieden waar de oplossing glad is. Voorts volgen we de gebruikelijke aanpak om oscillaties nabij discontinuiteiten tegen te gaan, uitgebreid naar ongestructureerde "staggered" schema's. Een oscillatie-detecterende limiter functie wordt

gebruikt om over te schakelen tussen het superlineair convergerende schema en een eerste orde schema dat vrijwel geen oscillaties veroorzaakt. Hiertoe wordt de limiter van Barth en Jespersen aangepast.

Het resulterende schema wordt gevalideerd in hoofdstuk 5. Eenvoudige onsamendrukbare niet-visceuze stromingen waarvoor de exacte oplossing bekend is worden berekend, en de nauwkeurigheid van het schema wordt nagegaan. Vervolgens worden de onsamendrukbare Navier-Stokes vergelijkingen opgelost voor de exacte Kovasznay oplossing, en het gedrag van de fout als het rooster wordt verfijnd wordt bestudeerd. De "backward facing step" stroming en stromingen om een cirkel cylinder worden berekend en er wordt vergeleken met enkele bekende eigenschappen van deze stromingen. Vervolgens worden de Euler vergelijkingen opgelost voor enkele samendrukbare stromingsproblemen met ons Mach-uniforme schema. De Ringleb stroming is een transone stroming met een bekende exacte oplossing. De nauwkeurigheid van ons superlineair convergerende schema met en zonder limiting wordt bestudeerd. De stroming in een kanaal met een hobbel wordt berekend voor een laag en een hoog Machgetal. Een transone profiestroming waarvoor de drukverdeling redelijk goed bekend is wordt berekend. Om nauwkeurig na te gaan of aan de Rankine-Hugoniot relaties wordt voldaan wordt een schokbuis stroming berekend.

Concluderende opmerkeingen worden gepresenteerd in hoofdstuk 6. De mogelijkheden voor uitbreiding naar drie dimensies worden besproken. De vooruitzichten voor vectorveld extrapolatie uitgaande van normaalcomponenten op celwanden op niet-driehoekige en niet-tetrahedrale roosters worden besproken. We concluderen dat een Mach-uniform ongestructureerd "staggered" schema met superlineaire convergentie is verkregen.

Contents

Summary iii

Samenvatting v

1 Introduction 1
- 1.1 The challenge of computational fluid dynamics . . . 1
- 1.2 Purpose of this thesis . . . 2

2 Problem formulation 5
- 2.1 Summary . . . 5
- 2.2 Basic equations of fluid dynamics . . . 5
- 2.3 Incompressible flow . . . 7
- 2.4 Inviscid flow . . . 8
- 2.5 Dimensionless equations . . . 8
- 2.6 Choice of domain, initial conditions and boundary conditions . . . 10
- 2.7 Flow solvers . . . 11
  - 2.7.1 Structured versus unstructured grids . . . 11
  - 2.7.2 Finite volumes versus other methods . . . 12
  - 2.7.3 Staggered versus collocated grids . . . 12

3 Time stepping 15
- 3.1 Summary . . . 15
- 3.2 Introduction . . . 15
- 3.3 Pressure-correction method for incompressible flow . . . 16
- 3.4 Time-stepping method for compressible flow . . . 16
- 3.5 Mach-uniform pressure-correction method . . . 17
  - 3.5.1 Boundary conditions and the equation of state . . . 18

4 Finite volume discretization on staggered unstructured grids 19
- 4.1 Summary . . . 19
- 4.2 Introduction . . . 19
- 4.3 Discretization of the divergence of the velocity . . . 21
- 4.4 Discretization of the continuity equation for compressible flow . . . 22
- 4.5 Linear reconstruction of scalars . . . 23
- 4.6 Linear reconstruction of staggered vector fields . . . 24
  - 4.6.1 Divergence-free linear reconstruction . . . 27
- 4.7 Quadratic reconstruction . . . 28
  - 4.7.1 Divergence-free quadratic reconstruction . . . 30

4.8 Cubic reconstruction . . . 31
4.9 A note on boundary conditions in reconstruction . . . 32
4.10 Hierarchical least squares method . . . 33
4.11 Discretization of momentum equation . . . 35
4.11.1 Viscous term . . . 36
4.11.2 Inertia term . . . 51
4.11.3 Pressure term . . . 53
4.11.4 Treatment of boundaries . . . 54
4.12 Pressure-correction equation in incompressible flow . . . 56
4.13 Mach-uniform pressure-correction equation . . . 56
4.14 Energy equation . . . 57
4.15 Equation of state . . . 57
4.16 Monotonicity considerations . . . 57

5 Numerical validation **61**
5.1 Summary . . . 61
5.2 Incompressible inviscid flow . . . 61
5.2.1 Converging flow . . . 62
5.2.2 Solid body rotation . . . 65
5.2.3 Stagnation point flow . . . 65
5.3 Incompressible viscous flow . . . 70
5.3.1 Kovasznay flow . . . 71
5.3.2 Backward facing step . . . 76
5.3.3 Flow around a circular cylinder . . . 80
5.4 Compressible flow . . . 81
5.4.1 Ringleb flow . . . 81
5.4.2 Channel with a bump . . . 82
5.4.3 NACA 0012 airfoil . . . 88
5.4.4 Sod's shocktube problem . . . 88

6 Conclusion **91**
6.1 Possible extensions . . . 91
6.1.1 Momentum reconstruction on three-dimensional tetrahedral grids . . . 91
6.1.2 Non-triangular and non-tetrahedral grids . . . 92
6.2 Closing remarks . . . 92

Curriculum vitae **97**

List of publications **99**

Chapter 1

Introduction

1.1 The challenge of computational fluid dynamics

The basic laws that govern most of the physical phenomena that we experience in our everyday life have been known for more than a century. Still, in practice we can model and predict these phenomena only to a limited extent. The reason is that the phenomena governed by these laws are too complex. The basic laws are expressed in the form of systems of nonlinear partial differential equations for which mathematics does not have explicit solutions, except in a limited number of simple cases.

It is sometimes said that mathematics is solving problems that it can solve, while numerical mathematics is solving problems that need to be solved. Finding approximate solutions for physical laws that cannot be solved exactly is the major task of numerical analysis. Another discipline that deals with these problems, though from a different point of view, is statistics.

However, solutions that numerical analysis is able to provide may not always be satisfactory. Some physical phenomena are far too complex even for modern numerical analysis. Examples can be found in fluid flows, multibody problems involving gravity, or in biology. The discipline that deals with solving flow problems numerically is known as *computational fluid dynamics* (CFD). The equations that model flow of fluids are known as Navier-Stokes equations. An effect of the interaction between the inertia and the sheer stress, which are modeled by these equations, is the turbulence. This is a chaotic movement of a fluid resulting in a huge number of eddies of very wide scale span. It is surprising that such complexity can arise from seemingly simple conservation laws, which the Navier-Stokes equations actually are. Even storing the information of all these eddies is often far beyond the capabilities of computers that we nowadays have, and the time needed to accurately compute turbulent flows is usually longer than our lifetime.

This does not mean that CFD is not a useful tool when dealing with flow problems, as it was actually the case few decades ago, when fluid dynamists had to rely mainly on experiments. In the past decades we have seen a tremendous increase of computational power. The processing speed and the storage capabilities have increased thousands of times compared to the situation thirty

years ago. The numerical algorithms have also become much more efficient. In order to make computations feasible, today we usually still have to simplify flow problems. Some flows are laminar (non-turbulent) which makes them much less complex and easier to solve, and some can be treated as laminar, even though they are actually not. Turbulence models may be used for all flow scales, or only the smallest eddies may be modeled. Furthermore, some problems can successfully be solved in two dimensions. Since fluid flow is an unstable phenomenon, meaning that the smallest change of conditions can result in drastic changes of the flow, the solution of such simplified problem may or may not be a useful approximation of the real flow. For many applications, numerical methods and computational power that we can use today are far from satisfactory.

The trend of increasing computational capabilities still continues. Computations that were not feasible few years ago are feasible today, or may become feasible in several years. As Stephen Hawking said in [18], *even if we do find a complete set of basic laws, there will still be in the years ahead the intellectually challenging task of developing better approximation methods.*

1.2 Purpose of this thesis

The purpose of this thesis is to improve computing methods that use so-called *unstructured grids*, because they are easier to generate and more flexible than *structured grids*. Furthermore, we aim at a unified computing method for both both incompressible and compressible flow, for reasons that will be given later. To achieve this, *staggered grids* are used. The types of grid just mentioned will be described later.

The outline of this thesis is as follows:

- In Chapter 2 the Navier-Stokes equations are presented. A non-dimensionalization which facilitates unified treatment of the compressible and incompressible flow case is presented. Then follows a discussion on how to choose the domain and initial and boundary conditions properly. Finally, a general discussion on the most popular types of flow solvers is given, together with some basic notions such as a specification of an unstructured staggered grid.

- In Chapter 3 we discuss time integration. We propose three schemes for solving the flow problems. The first method is for incompressible flows, the second one is for compressible flows, and the third one is a unified method for both incompressible and compressible flows. The third method works efficiently for Mach numbers from zero until hypersonic values. Therefore the method is called *Mach-uniform*.

- Chapter 4 is the central part of this thesis. A novel spatial discretization scheme is presented. It is an unstructured staggered finite-volume scheme with order of accuracy better than 1. Several options for polynomial reconstruction of staggered vector and scalar fields are considered. A special method for solving systems of linear equations called *hierarchical least squares method* is developed to facilitate this reconstruction. Flux limiting is used to combat numerical wiggles.

- Chapter 5 contains numerical results. The accuracy of the incompressible and the Mach-uniform scheme is investigated using various test cases, which include Kovasznay flow, backward facing step, flow around a cylinder, Ringleb flow, supersonic flow in a channel with a bump, transonic flow around the NACA 0012 airfoil, and Sod's shocktube problem. The accuracy of the schemes appears to be superlinear for smooth problems, and resolution of discontinuities is also better than obtained with first order schemes.

Chapter 2

Problem formulation

2.1 Summary

In this chapter the partial differential equations that govern the motion of the fluid are given. Various basic notions of fluid dynamics are introduced, including Newtonian fluids, Mach number and compressibility. A non-dimensionalization that is suitable for Mach-uniform flows is presented, followed by a discussion on how to choose the domain and boundary conditions properly. Finally, we give a brief survey of the most common types of flow solvers.

2.2 Basic equations of fluid dynamics

The laws that govern the motion of fluids are the mass conservation law (referred to as continuity, mass, or density equation), the Navier-Stokes equations or momentum conservation law, and the energy equation describing the conservation of energy:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{u}\rho) = 0 \tag{2.1}$$

$$\frac{\partial m_\alpha}{\partial t} + \nabla \cdot (\mathbf{u} m_\alpha) = \sigma_{\alpha\beta,\beta} + \rho f_\alpha \tag{2.2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\mathbf{u}\rho E) = (u_\alpha \sigma_{\alpha\beta})_{,\beta} + (kT_{,\alpha})_{,\alpha} + \rho u_\alpha f_\alpha + \rho q. \tag{2.3}$$

Greek indices indicate coordinate directions. Cartesian tensor notation is applied, which means that $\phi_{,\alpha} = \partial\phi/\partial x_\alpha$, and the summation convention is used: summation takes place over Greek indices that occur twice in a term or a product, for example $v_{\alpha,\alpha} = \nabla \cdot \mathbf{v}$, or $u_\alpha v_\alpha = \mathbf{u} \cdot \mathbf{v}$. The meaning of the symbols is as follows:

- ρ - density, kg/m^3
- $\mathbf{u}$ - velocity, m/s
- t - time, s
- $\mathbf{m}$ - momentum, $\mathbf{m} = \rho\mathbf{u}$

- σ - stress tensor, $\sigma_{\alpha\beta} = -p\delta_{\alpha\beta} + \tau_{\alpha\beta}$, $\mathrm{N/m}^2$
- τ - deviatoric stress tensor, $\tau_{\alpha\beta} = 2\mu\left(e_{\alpha\beta} - \frac{1}{3}\Delta\delta_{\alpha\beta}\right)$, $\mathrm{N/m}^2$
- p - pressure, $\mathrm{N/m}^2$
- $\delta_{\alpha\beta}$ - Kronecker delta symbol
- μ - dynamic viscosity coefficient, kg/ms
- $e_{\alpha\beta}$ - rate of strain tensor, $e_{\alpha\beta} = \frac{1}{2}\left(u_{\alpha,\beta} + u_{\beta,\alpha}\right)$
- Δ - divergence of velocity, $\Delta = u_{\alpha,\alpha}$
- $\mathbf{f}$ - body force, N/kg
- E - total energy per unit mass, J/kg
- T - temperature, K
- k - thermal conductivity, J/msK
- q - added heat per unit of mass per unit time, J/kgs

This implies that we assumed Newtonian fluid. The nonlinear terms $\nabla \cdot (\mathbf{u}\rho)$, $\nabla\cdot(\mathbf{u}m_\alpha)$ and $\nabla\cdot(\mathbf{u}\rho E)$ represent the convection of mass, momentum and energy, respectively. They are called the *convective terms*; in the case of momentum one also speaks of *inertia term*.

Assuming a perfect gas, we have

$$p = \rho RT \tag{2.4}$$

with R a constant characteristic of the gas.

The total energy is defined as

$$E = e + \tfrac{1}{2}\mathbf{u}\cdot\mathbf{u}, \tag{2.5}$$

where e is the internal energy. Assuming a calorically perfect gas, we have

$$e = c_v T \tag{2.6}$$

with c_v the specific heat at constant volume, assumed constant. The system of equations is closed by the thermal equation of state $p = p(\rho, T)$.

The speed of sound a satisfies

$$a^2 = \gamma RT = \gamma(\gamma - 1)e \tag{2.7}$$

where we have used $R = c_p - c_v$, $\gamma = c_p/c_v$, where c_p is the specific heat at constant pressure. For gases consisting of di-atomic molecules, such as N_2 and O_2, and hence to a good approximation air, we have

$$\gamma = \tfrac{7}{5} \tag{2.8}$$

which is the value we will use.

Fluids satisfying the relation between the deviatoric stress tensor $\tau_{\alpha\beta}$ and the rate of strain tensor $e_{\alpha\beta}$ as specified above are called Newtonian fluids.

Many fluids have Newtonian behavior. Examples are gases and water at not too extreme conditions. Examples of non-Newtonian fluids are polymers, blood, some greases, sludges and quicksand. We shall assume a Newtonian fluid.

In this thesis we shall neglect the body force $\mathbf{f}$, and we shall also assume that $q = 0$. Furthermore, we shall consider viscous flow only in the incompressible flow case. Although most of the concepts presented here are applicable in the three-dimensional case, we shall consider only the two-dimensional case.

In order to obtain a unified method for compressible and incompressible flow, use will be made later of a different form of the energy equations. Assuming a calorically perfect gas, by substituting (2.5) and the following form of the equation of state

$$e = \frac{1}{\gamma - 1}\frac{p}{\rho} \tag{2.9}$$

in the energy equation (2.3), after multiplying by $\gamma - 1$ the energy equation becomes (assuming $f_\alpha \equiv 0$ and $q \equiv 0$)

$$\begin{aligned}\frac{\partial}{\partial t}\left[p + \frac{1}{2}(\gamma-1)\rho u^2\right] + \nabla \cdot \left[\mathbf{u}\left(\gamma p + \frac{1}{2}(\gamma-1)\rho u^2\right)\right] = \\ = (\gamma-1)(u_\alpha \tau_{\alpha\beta})_{,\beta} + (\gamma-1)(kT_{,\alpha})_{,\alpha},\end{aligned} \tag{2.10}$$

where $u^2 = \mathbf{u} \cdot \mathbf{u}$.

2.3 Incompressible flow

The flow is incompressible if the fluid is not compressed by pressure, i.e. if variations in density in each fluid particle are negligible. This means that

$$\frac{\partial \rho}{\partial t} + (\mathbf{u} \cdot \nabla)\rho = 0 \tag{2.11}$$

From the mass conservation equation (2.1) it then follows that

$$\nabla \cdot \mathbf{u} = 0. \tag{2.12}$$

This is the distinguishing property of incompressible flow. Vector fields satisfying (2.12) are called *solenoidal*. If all fluid particles emanate from a region where ρ is uniform, (2.11) tells us that ρ is constant. This is frequently the case in incompressible flows, but not always.

Conditions under which a flow is (to a good approximation) incompressible are discussed in [5] Sect. 3.6 and [38] Sect. 1.12. The main conditions are

$$\mathrm{M} = |\mathbf{u}|/a \ll 1 \tag{2.13}$$

where M is the Mach number, and

$$\frac{L}{\tau a} \ll 1 \tag{2.14}$$

where L is a characteristic length scale of the flow, and τ a characteristic time scale. The first condition limits the velocity, whereas the second condition effectively says that τ cannot be too small, precluding acoustics.

When isothermal incompressible flow is considered, there are only four independent variables: the pressure and the three components of momentum. Therefore, the energy equation is not needed.

2.4 Inviscid flow

An ideal flow is a flow in which the friction and the heat conduction are negligible, i.e. one can take $\mu = 0$ and $k = 0$. This is a good approximation for flow at high Reynolds number (to be defined shortly) over objects (aerodynamics) outside a thin layer just next to the object, which is called the boundary layer. In this case equations (2.1)–(2.3) reduce to

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\mathbf{u}\rho) = 0, \tag{2.15}$$

$$\frac{\partial m_\alpha}{\partial t} + \nabla \cdot (\mathbf{u} m_\alpha) = -p_{,\alpha}, \tag{2.16}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\mathbf{u}\rho H) = 0. \tag{2.17}$$

These are called the Euler equations. The total enthalpy H is related to the other thermodynamical quantities in the following ways:

$$H = h + \frac{1}{2}u^2, \tag{2.18}$$

$$h = \gamma e, \tag{2.19}$$

where $u = |\mathbf{u}|$. We have

$$\rho H = \rho E + p. \tag{2.20}$$

The system of equations (2.15)–(2.17) can be written in the following form, that will be referred to later:

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0 \tag{2.21}$$

where

$$U = \begin{bmatrix} \rho \\ \mathbf{m} \\ \rho E \end{bmatrix} \quad \text{and} \quad F_\alpha = \begin{bmatrix} {}^{\alpha} \\ u_\alpha m_1 + p\delta_{\alpha 1} \\ u_\alpha m_2 + p\delta_{\alpha 2} \\ u_\alpha m_3 + p\delta_{\alpha 3} \\ m_\alpha H \end{bmatrix} \tag{2.22}$$

Equation (2.21) is called a *system of conservation laws*.

The pressure-based inviscid energy equation is obtained from (2.10) by deleting the right-hand side:

$$\frac{\partial}{\partial t}\left[p + \frac{1}{2}(\gamma - 1)\rho u^2\right] + \nabla \cdot \left[\mathbf{u}\left(\gamma p + \frac{1}{2}(\gamma - 1)\rho u^2\right)\right] = 0, \tag{2.23}$$

2.5 Dimensionless equations

In the remainder of this thesis, dimensional quantities will be indicated by a hat symbol, e.g. $\hat{\phi}$. Quantities without a hat are dimensionless, e.g. ϕ. The following reference units are chosen for velocity, length, density, temperature and pressure: $\hat{u}_r$, $\hat{L}$, $\hat{\rho}_r$, $\hat{T}_r$, $\hat{p}_r$. We require that reference quantities satisfy the equation of state $\hat{p}_r = R\hat{\rho}_r\hat{T}_r$. The reference Mach number is given by

$$\mathrm{M}_r = \hat{u}_r / \sqrt{\gamma R \hat{T}_r}. \tag{2.24}$$

The variables are made dimensionless in the usual way, e.g. $x_\alpha = \hat{x}_\alpha/\hat{L}$, $t = \hat{t}\hat{u}_r/\hat{L}$, $T = \hat{T}/\hat{T}_r$, $\rho = \hat{\rho}/\hat{\rho}_r$, $u_\alpha = \hat{u}_\alpha/\hat{u}_r$, and $p = \hat{p}/\hat{p}_r$. The mass conservation equation (2.1) is invariant under non-dimensionalization. The dimensionless momentum equation (2.2) is found to be, taking $f_\alpha \equiv 0$,

$$\frac{\partial m_\alpha}{\partial t} + \nabla \cdot (\mathbf{u} m_\alpha) = -\frac{1}{\gamma \mathrm{M}_r^2} p_{,\alpha} + \tau_{\alpha\beta,\beta}, \tag{2.25}$$

where $\tau_{\alpha\beta} = \frac{2}{\mathrm{Re}}(e_{\alpha\beta} - \frac{1}{3}\Delta\delta_{\alpha\beta})$ and $\mathrm{Re} = \hat{\rho}_r\hat{u}_r L/\mu$ is the Reynolds number. This equation is singular as $\mathrm{M}_r \downarrow 0$.

In order to obtain Mach-uniform dimensionless equations, the pressure is made dimensionless in the following way, as in [6, 7, 32, 36]:

$$p = \frac{\hat{p} - \hat{p}_r}{\hat{\rho}_r \hat{u}_r^2}. \tag{2.26}$$

Now the dimensionless momentum equation is found to be

$$\frac{\partial m_\alpha}{\partial t} + \nabla \cdot (\mathbf{u} m_\alpha) = -p_{,\alpha} + \tau_{\alpha\beta,\beta}. \tag{2.27}$$

The dimensionless equation of state can be written as

$$T = \frac{1}{\rho}(1 + \gamma \mathrm{M}_r^2 p), \tag{2.28}$$

and the dimensionless form of the pressure-based energy equation (2.10) is found to be

$$\mathrm{M}_r^2 \Bigg\{ \frac{\partial}{\partial t}\left[p + \tfrac{1}{2}(\gamma-1)\rho u^2\right] + \nabla \cdot \left[\mathbf{u}\left(\gamma p + \tfrac{1}{2}(\gamma-1)\rho u^2\right)\right] -$$
$$(\gamma-1)(u_\alpha \tau_{\alpha\beta})_{,\beta} - \gamma\left[\frac{1}{\mathrm{RePr}}\left(\frac{p}{\rho}\right)_{,\alpha}\right]_{,\alpha}\Bigg\} + \nabla \cdot \mathbf{u} = \left[\frac{1}{\mathrm{RePr}}\left(\frac{1}{\rho}\right)_{,\alpha}\right]_{,\alpha} \tag{2.29}$$

where

$$\mathrm{Pr} = \frac{c_p \hat{\mu}}{\hat{k}} \tag{2.30}$$

is the Prandtl number. When $\mathrm{M}_r = 0$, equation (2.29) reduces to

$$\nabla \cdot \mathbf{u} = \left[\frac{1}{\mathrm{RePr}}\left(\frac{1}{\rho}\right)_{,\alpha}\right]_{,\alpha}. \tag{2.31}$$

We see that in order to obtain the familiar solenoidality condition for incompressible flow, it is not sufficient to let $\mathrm{M}_r \downarrow 0$. To see which additional condition is required, we use the $\mathrm{M}_r \downarrow 0$ limit of the dimensionless equation of state (2.28) to rewrite (2.31) as

$$\nabla \cdot \mathbf{u} = \left(\frac{1}{\mathrm{RePr}} T_{,\alpha}\right)_{,\alpha}. \tag{2.32}$$

We see that in order to have $\nabla \cdot \mathbf{u} = 0$ it is required, in addition to $\mathrm{M}_r \downarrow 0$, that temperature variations are small. With $q = 0$ and no large heat transfer trough the boundaries (for example, adiabatic flow), the right-hand-side can be safely neglected.

2.6 Choice of domain, initial conditions and boundary conditions

The momentum, the continuity, and the energy equation together govern the flow of a Newtonian fluid in any circumstances. To determine a specific flow (in a specific domain, under specific temperature, pressure etc), initial and boundary conditions need to be specified.

Computations are always carried out in a specified finite domain. This is the region in space in which are computing the flow. Unless the fluid in the domain is isolated from the outer world, as in the case of a cavity flow, there is an inflow boundary and an outflow boundary. A simple example is the flow inside a pipe. In this case the interior of the pipe is the computational domain, one end of the pipe is the inflow, and another end is the outflow boundary. If the pipe is too long, we may compute the flow only in a portion of the pipe. In this case the domain is reduced.

Mathematical considerations determine how boundary conditions should be prescribed along the boundaries of the domain to ensure that the problem is well-posed. A survey of suitable boundary conditions is given in [38]. If a viscous problem is solved, the normal stress $\sigma_{nn} = \mathbf{n} \cdot \sigma \cdot \mathbf{n}$ and the tangential stress $\sigma_{nt} = \mathbf{n} \cdot \sigma \cdot \mathbf{t}$ are prescribed at the outflow boundary, where $\mathbf{n}$ and $\mathbf{t}$ are the unit normal and the unit tangential vector at the boundary, and the momentum is prescribed at the inflow boundary. Viscous fluids cling to solid surfaces, and this is called the *no-slip* condition. In this case the momentum is set to zero along solid walls. If there is no friction along a solid wall, then we prescribe $\sigma_{nt} = 0$ and $\mathbf{m} \cdot \mathbf{n} = 0$. This is called the *free slip* condition. There is also a *free surface* condition, where we allow that the position of the boundary may change, but we do not consider this case.

For the Euler equations the deviatoric stress is zero, so the only stress component that may need to be prescribed is the normal stress, which reduces to the pressure. If the flow is subsonic at the outflow boundary, the outflow pressure is prescribed, otherwise no boundary condition is needed at the outflow boundary. At the inflow boundary, momentum and density or static enthalpy are prescribed. If the inflow is supersonic, pressure is prescribed as well.

Computations are carried out for some time interval. The flow in this interval depends on the initial state of the fluid. This means that at the initial time all variables must be known. Of course, this is often not the case. In real life problems, one usually starts with a simple and possibly non-realistic solution, for example by setting all variables to constant values, and then during the initial state of computation these quantities hopefully evolve toward more realistic values.

The initial solution should satisfy the boundary conditions. However, this does not guarantee convergence, and if this condition is violated such inconsistency can sometimes be resolved in the initial stage of the computation, when the solution adapts to the boundary conditions. On the other hand, this inconsistency may also cause schemes to break down.

Frequently we want to compute a steady solution, i.e. a solution that does not change in time. The steady solution is the solution of the flow equations from which the time derivatives are deleted. However, it is hard to solve such equations because the diagonal dominance in the related matrix comes from

the time term. Therefore we compute steady solutions by time-stepping, and we stop the computations when the variation in time of the solution becomes small enough. It has been proven (see [38]) that the steady solution is unique under some restrictions. This means that the initial solution has no influence on the steady solution, and we may choose any initial solution that allows the convergence to the steady solution.

The real flow domain may be infinite. An example is the flow around an airplane. In such cases, for computational purposes one has to take a finite portion of space around the airplane as computational domain. This domain must be large enough to ensure that the flow at the outer boundaries is nearly uniform, otherwise specification of the boundary conditions as being equal to the flow state at infinity would perturb the solution too much.

2.7 Flow solvers

The systems of partial differential equations presented in the previous sections are too complex to be solved analytically, except in a very limited number of cases. Therefore they have to be solved approximately by numerical methods. The usual way to achieve this is to represent the continuous scalar and vector fields like momentum, density etc. by a finite number of their values in certain points and in certain instants in time, or by a finite number of their local averages, and then to approximate partial derivatives used in these equations by linear combinations of these values. Such a process is called *discretization*. It results in large systems of algebraic equations, which then have to be solved.

To facilitate the discretization, the computational domain is usually decomposed into small polygons called *cells*. Such a division is called *mesh* or *grid*. The boundaries between the cells are the *faces* of the grid (or *edges* in the two-dimensional case), and points where these edges meet are called *vertices*.

2.7.1 Structured versus unstructured grids

One of the bottlenecks in CFD is grid generation in complex domains. This is often a time-consuming task, especially in three dimensions.

A grid is called *structured* if all interior vertices belong to a constant number of cells. A structured grid is called *Cartesian* if its cells are rectangles or rectangular blocks.

It is much easier to discretize partial differential equations on structured Cartesian grids than on non-Cartesian unstructured grids. An easy way to discretize on a Cartesian grid is to use finite differences. For example,

$$\frac{\partial \phi(x,y)}{\partial x} \approx \frac{\phi(x+h,y)-\phi(x,y)}{h}. \tag{2.33}$$

However, complicated domains can rarely be decomposed into rectangles. One approach is to smoothly map a Cartesian grid into the domain. This process cannot at present be fully automated and sometimes requires much human work. Another problem is that in some applications the mesh needs to be finer in some areas than in the others, i.e. these areas need to be divided into more smaller cells than the others, in order to capture the relevant flow details. This is also hard to achieve with Cartesian grids.

Another approach is to divide the domain into triangles. This process is called triangulation. It is far less simple than it seems, especially when additional requirements related to the quality of the grid are imposed. For example, many numerical schemes do not perform well with long and thin or obtuse triangles. Triangulation forms a separate discipline. However, robust algorithms have been developed that generate high-quality unstructured triangular meshes, and generating an unstructured triangular mesh can be fully automated. Therefore we shall use such meshes. But more complicated data structures and algorithms are required compared to the schemes developed for structured grids. The present trend in CFD is towards unstructured grids.

2.7.2 Finite volumes versus other methods

When unstructured grids are used, finite difference methods are difficult to apply. The most popular options are to use the finite volume, finite element, or discontinuous Galerkin method.

The finite element method is the most popular method in structural mechanics, as well as in some other fields. The reasons are that it works well with meshes of low quality, and that it has a solid mathematical basis. But in fluid dynamics the method has always suffered from a lack of overall stability and accuracy. These problems are successfully solved by introducing so-called penalty functions or other stabilization methods, but this makes the method rather complicated.

Finite volume methods remain the most popular choice for computational fluid dynamics. Most of the commercial codes used for engineering applications use finite volume methods. The disadvantages of these methods are that they are usually only first order accurate, that they usually require meshes of high quality to ensure good performance, and that their mathematical foundation is less complete than for finite element method. Spectral element methods are higher order finite volume methods.

The discontinuous Galerkin method received much attention in recent years. It is said to be "in between" the finite elements and the finite volumes. Like finite elements, it is based on the standard Galerkin method.

2.7.3 Staggered versus collocated grids

A grid is called *collocated* if the pressure and the velocity vectors are stored in the same points. They may be stored in mesh vertices or in cell centers. Collocated finite volume methods for the incompressible Navier-Stokes equations suffer from pressure-velocity decoupling. This gives rise to spurious pressure modes, namely checkerboard type oscillations.

In *staggered* grids the pressure and the momentum are stored in different points. The usual choice is to store the pressure in the cells, as well as the other scalars, and to associate normal momentum components with the faces. This is illustrated in Fig. 2.1.

For numerical approximation of the incompressible Navier-Stokes equations on Cartesian grids, the classical staggered Marker and Cell scheme of Harlow and Welch [17] is the original staggered mesh method. The reasons are absence of spurious modes, local mass conservation (see Sect. 4.3) and the fact that artificial boundary conditions are not needed.

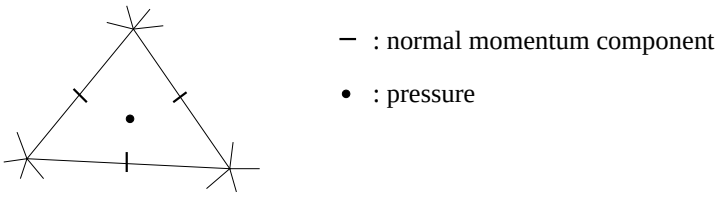

Figure 2.1: Staggered positioning of the variables in an unstructured grid.

Several generalizations of this scheme have been proposed for unstructured triangular meshes in [13, 14, 25, 26, 27, 28, 36, 37]. All these methods are first-order accurate in space on irregular meshes. Except [36, 37], the schemes described are restricted to incompressible flows.

Since the momentum vector is not known in any point, the preprocessing and the postprocessing stages are more complicated than in the collocated case. Additional difficulties that arise on unstructured grids may be the reason why the staggered unstructured schemes are not widely accepted. But this is the approach to be investigated here.

Chapter 3

Time stepping

3.1 Summary

In this chapter we introduce three time stepping methods. These are the classical incompressible pressure-correction method, a simple compressible method, and the Mach-uniform pressure-correction method, which is an extension of the incompressible pressure-correction method towards the compressible flow case.

3.2 Introduction

The equations are solved using time-stepping. This means that, starting with the initial conditions at time 0 and choosing some time step Δt, one sequentially computes the unknown physical quantities at times Δt, $2\Delta t$, etc.

For ease of exposition we restrict our considerations here to the inviscid flow case. Extension to the viscous flow case is straightforward. This means that we will consider the system of conservation laws (2.21):

$$\frac{\partial U}{\partial t} + \nabla \cdot F(U) = 0 \tag{3.1}$$

All equations are discretized in time with the implicit Euler scheme. The reason is that it allows larger time steps than explicit schemes. More sophisticated time-stepping schemes will not be considered, because our main aim is to develop a novel spatial discretization scheme. Euler implicit time-discretization results in

$$\frac{U^{n+1} - U^n}{\Delta t} + [\nabla \cdot F(U)]^{n+1} = 0. \tag{3.2}$$

As the Mach-number $\mathrm{M} \downarrow 0$ the system (3.2) becomes extremely stiff and therefore hard to solve efficiently. The difficulties associated with the limit $\mathrm{M} \downarrow 0$ are discussed in [38] Chapter 14, [7] and references quoted there. These difficulties are the reason why historically methods for incompressible and compressible flows have developed along quite different lines. We will use a Mach-uniform method, obtained by generalizing a method for incompressible flows to the compressible flow case.

3.3 Pressure-correction method for incompressible flow

The easiest way to see what becomes of (3.2) in the incompressible limit $\mathrm{M} \downarrow 0$ is to replace the energy equation (2.17) by the pressure-based energy equation (2.29). We then see that the equation for incompressible flow becomes, adding the viscous terms at the new time level, and supposing that the temperature variations are small (see Sect. 2.5):

$$\frac{\rho^{n+1}-\rho^n}{\Delta t} + \nabla\cdot(\mathbf{u}^{n+1}\rho^{n+1}) = 0, \tag{3.3}$$

$$\frac{m_\alpha^{n+1}-m_\alpha^n}{\Delta t} + \nabla\cdot(\mathbf{u}^{n+1}m_\alpha^{n+1}) = -p_{,\alpha}^{n+1} + \tau_{\alpha\beta,\beta}^{n+1}, \tag{3.4}$$

$$\nabla\cdot\mathbf{u}^{n+1} = 0. \tag{3.5}$$

This system lacks a time-derivative for p^{n+1}. Again we see why special methods have evolved for the incompressible flow case.

An efficient way to solve (3.3)–(3.5) is the pressure-correction method. Assume that the density is constant (this is not necessary). A prediction m^* of m^{n+1} is computed from

$$\frac{m_\alpha^{*}-m_\alpha^n}{\Delta t} + \nabla\cdot(\mathbf{u}^{n}m_\alpha^{*}) = -p_{,\alpha}^{n} + \tau_{\alpha\beta,\beta}^{*}, \tag{3.6}$$

where we have used Picard linearization, i.e. $\mathbf{u}^{n+1}$ is replaced by $\mathbf{u}^n$. The following pressure-correction is postulated:

$$\mathbf{m}^{n+1} = \mathbf{m}^* - \Delta t\nabla\delta p, \qquad \delta p = p^{n+1}-p^n. \tag{3.7}$$

The pressure correction δp is computed from

$$\Delta t\nabla\cdot(\nabla\delta p/\rho^{n+1}) = \nabla\cdot\mathbf{u}^*, \tag{3.8}$$

where $\mathbf{u}^* = \mathbf{m}^*/\rho^{n+1}$. Finally, the new velocity and the new pressure follow from (3.7). For the derivation of these formulas see [38].

3.4 Time-stepping method for compressible flow

A simple way to compute compressible inviscid flows is the following:

1. Compute the new momentum from the discretized momentum equation:

$$\frac{m_\alpha^{n+1}-m_\alpha^n}{\Delta t} + \nabla\cdot(\mathbf{u}^{n}m_\alpha^{n+1}) = -p_{,\alpha}^{n} \tag{3.9}$$

2. Compute the new density from the discretized density equation:

$$\frac{\rho^{n+1}-\rho^n}{\Delta t} + \nabla\cdot(\mathbf{u}^{n}\rho^{n+1}) = 0, \tag{3.10}$$

3. Compute the new energy variable ρH from the discretized energy equation

$$\frac{(\rho E)^{n+1}-(\rho E)^n}{\Delta t} + \nabla\cdot(\mathbf{u}\rho H)^{n+1} = 0, \tag{3.11}$$

where $\rho E = \frac{1}{\gamma}\rho H + \frac{1}{2}\frac{\gamma-1}{\gamma}\rho u^2$ follows from (2.5), (2.18) and (2.19).

4. Compute the new pressure from the equation of state:

$$p = \frac{\gamma - 1}{\gamma} \left[\rho H - \frac{1}{2}(\rho u^2) \right]. \tag{3.12}$$

We use ρH as the primary energy variable because the numerical experiments presented in [36] show that this choice gives better results than the alternatives.

This scheme was implemented as an intermediate step towards the Mach-uniform scheme because of its simplicity. It has no other advantages. It suffers from loss of accuracy when Mach number approaches zero. Due to linearization and uncoupling of the equations, very small time steps are required. In general gas dynamics flow there is no reason to use staggered schemes for fully compressible flows.

3.5 Mach-uniform pressure-correction method

In certain applications, as for example flow in internal combustion engines or flow around aircraft in take-off or landing conditions, compressible and incompressible flow regions occur simultaneously. Standard compressible flow schemes suffer from efficiency and accuracy loss when the Mach number becomes small (below 0.2). Such problems require methods that can handle flows at all speeds. Staggering of the grid is attractive for unified methods because of superior properties in the incompressible flow case. Mach-uniform methods that use structured staggered grids were presented in [15, 16], and more recently in [32, 6]. The only Mach-uniform method that uses staggered unstructured grids that we know of is presented in [36]. Below we describe the sequential update procedure presented in the latter paper. It is an extension of the pressure-correction method to the compressible flow case. We assume inviscid flow; however, viscous terms can be added easily.

First the new density is computed from the discretized mass conservation equation

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} + \nabla \cdot (\mathbf{u}^n \rho^{n+1}) = 0. \tag{3.13}$$

Next, as in the incompressible flow case, a prediction of the momentum field m^*_α is computed from the momentum equation:

$$\frac{m^*_\alpha - m^n_\alpha}{\Delta t} + \nabla \cdot (\mathbf{u}^n m^*_\alpha) = -p^n_{,\alpha} \tag{3.14}$$

Discretizing the inviscid flow version of (2.29) in time with the implicit Euler scheme, inserting (3.7) and substituting

$$\mathbf{m}^* = \rho^{m+1} \mathbf{u}^* \tag{3.15}$$

in the dimensionless pressure-based energy equation (2.29) leads to

$$\begin{aligned} \mathrm{M}_r^2 \left\{ \frac{\delta p}{\Delta t} + \tfrac{1}{2}(\gamma - 1) \frac{(\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} - (\mathbf{m}^n)^2 / \rho^n}{\Delta t} + \right. \\ \left. + \nabla \cdot \left[\left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) \left(\gamma (p^n + \delta p) + \tfrac{1}{2}(\gamma - 1)(\mathbf{m}^* - \Delta t \nabla \delta p)^2 / \rho^{n+1} \right) \right] \right\} + \\ + \nabla \cdot \left(\mathbf{u}^* - \frac{\Delta t}{\rho^{n+1}} \nabla \delta p \right) = 0. \end{aligned} \tag{3.16}$$

Linearization and some additional simplification gives

$$\begin{aligned} \mathrm{M}_r^2 \left\{ \frac{\delta p}{\Delta t} + \tfrac{1}{2}(\gamma-1) \frac{\left[(\mathbf{m}^*)^2 - 2\Delta t \mathbf{m}^* \cdot \nabla \delta p\right]/\rho^{n+1} - (\mathbf{m}^n)^2/\rho^n}{\Delta t} \right\} + \\ + \nabla \cdot \left[\mathbf{u}^* \left(1 + \gamma \mathrm{M}_r^2 (p^n + \delta p) + \frac{1}{2}(\gamma-1)\mathrm{M}_r^2 (\mathbf{m}^*)^2/\rho^{n+1} \right) \right] - \\ - \Delta t \nabla \cdot \left\{ \left[\left(1 + \mathrm{M}_r^2 p^n + \frac{1}{2}(\gamma-1)\mathrm{M}_r^2 (\mathbf{m}^*)^2 \rho^{n+1} \right) \rho^{n+1} \right] \nabla \delta p \right\} = 0. \end{aligned} \tag{3.17}$$

For detailed derivation of this equation see [36, 32]. It is used to compute δp. Finally, the new pressure and momentum are computed from (3.7).

3.5.1 Boundary conditions and the equation of state

It should be noted that the equation of state is not used in the time stepping in Mach-uniform pressure-correction method. The energy variable has been eliminated in the derivation of the pressure-based energy equation using the equation of state. The above time-stepping method employs only momentum, pressure, and density.

However, due to backward compatibility with the earlier code of [36], in our code sometimes we only specify the momentum, the pressure, and the static enthalpy at boundaries, so we do not specify the density, which is needed at the inflow boundary for the continuity equation. Since we do not know the new pressure at the inflow in the subsonic inflow case, we cannot compute the new density from the equation of state.

One should not use the density from the previous time level in such a situation. If we use the Mach-uniform pressure correction method, such an approach completely ignores the boundary condition for the enthalpy, which is not used anywhere else in the scheme. Therefore, even if the boundary conditions do not change over time, if the computations are carried long enough, the solution will certainly deteriorate at the inflow due to the round-off errors.

Therefore we compute the density from the equation of state, using the prescribed enthalpy and the pressure from the previous time level in the subsonic flow case, or the prescribed pressure in the supersonic flow case. This is enough to keep the solution at the desired value at the inflow.

However, if instead of the enthalpy we we specify the density at the inflow, the scheme converges faster to the steady state.

Chapter 4

Finite volume discretization on staggered unstructured grids

4.1 Summary

In this chapter we present our staggered finite volume method. Various possibilities for polynomial reconstruction of scalar and staggered vector fields are investigated here. To allow some of these polynomial reconstructions, an extension of the least squares method with constraints called *hierarchical least squares method* was devised, and is presented here. The techniques of the reconstruction of the staggered vector fields are applied in the viscous term, and the resulting schemes are tested. The vertex-based divergence-free linear reconstruction that gave the best results for the viscous term because it is accurate and not too memory-consuming is later used to discretize the inertia term and the kinetic energy terms that appear in the pressure-based energy equation. Also the pressure term is discretized here. Finally, several possibilities for flux limiting are considered.

The essentials of the material in this and the following chapter have been published in [34] and [35].

4.2 Introduction

Consider the following scalar conservation law:

$$\begin{aligned} \frac{\partial \phi}{\partial t} + \frac{\partial f(\phi)}{\partial x} = 0 \quad &\text{in} \quad \mathbb{R} \times \mathbb{R}^+, \\ \phi(x,0) = \phi_0(x) \quad &\text{for} \quad x \in \mathbb{R}, \end{aligned} \tag{4.1}$$

where for simplicity we restrict our considerations to the case of an unbounded one-dimensional domain. It is known that solutions of hyperbolic conservation laws may exhibit discontinuities. Since formally solutions of (4.1) are differentiable, another form of the equation is required, that allows discontinuous

solutions. This so-called *weak form* is obtained by multiplication by a test function $\psi \in \mathbb{C}_0^\infty(\mathbb{R} \times \mathbb{R}^+)$ and integration over the domain. Here $\psi \in \mathbb{C}_0^\infty$ is the space of infinitely differentiable functions that vanish at infinity. After partial integration one obtains

$$\int_{\mathbb{R}} \int_{\mathbb{R}^+} \left[\phi \frac{\partial \psi}{\partial t} + f(\phi) \frac{\partial \psi}{\partial x} \right] dt dx + \int_{\mathbb{R}} \phi_0 \psi(.,0) dx = 0, \quad \forall \psi \in \mathbb{C}_0^\infty(\mathbb{R} \times \mathbb{R}^+). \tag{4.2}$$

This is the weak form of (4.1), and $\phi \in \mathbb{L}^\infty(\mathbb{R} \times \mathbb{R}^+)$ is called *weak solution* if (4.2) is satisfied. Weak solutions of (4.1) are not unique. Uniqueness is obtained by adding a condition that ensures that the solution has a physical meaning; such a condition is called an *entropy condition*. Obviously, if $\phi \in \mathbb{C}^1(\mathbb{R} \times \mathbb{R}^+)$ satisfies (4.2), then it satisfies (4.1) in the classical sense, and the other way around. A consequence of (4.2) is that if ϕ is piecewise smooth, then at a discontinuity it satisfies

$$(\phi_L - \phi_R) s = f_L - f_R, \tag{4.3}$$

where L and R refer to opposite sides of the discontinuity and s is the propagation speed of the discontinuity. Equation (4.3) is called the *jump condition* or, especially in gasdynamics, the Rankine-Hugoniot condition.

Unlike the case of the finite element method and other methods based on the standard Galerkin method, for approximation with the finite volume method one does not start from the weak form but from the strong form. Going to two dimensions, we write

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0 \quad \text{in} \quad \Omega \subseteq \mathbb{R}^2 \tag{4.4}$$

where u may depend on ϕ.

The domain Ω is subdivided into a set of so-called *control volumes* or cells Ω_i such that

$$\Omega = \bigcup_i \Omega_i. \tag{4.5}$$

For the time being we also suppose that

$$\Omega_i \cap \Omega_j = \emptyset, \tag{4.6}$$

although this does not hold for the control volumes for the momentum equation, as will be seen in Section 4.11.

We integrate (4.4) over each control volume and apply the divergence theorem to obtain

$$|\Omega_i| \frac{\partial \phi_i}{\partial t} + \oint_{\partial \Omega_i} \phi (\mathbf{u} \cdot \mathbf{n}) d\Gamma = 0 \tag{4.7}$$

where

$$\phi_i = \frac{1}{|\Omega_i|} \int_{\Omega_i} \phi d\Omega. \tag{4.8}$$

A finite volume scheme is obtained after approximation of the cell integral in (4.7). Examples will follow.

Summation of (4.7) over i gives

$$\frac{d}{dt} \int_\Omega \phi d\Omega + \oint_{\partial \Omega} \phi (\mathbf{u} \cdot \mathbf{n}) d\Gamma = 0 \tag{4.9}$$

which shows that $\int_\Omega \phi d\Omega$ can change only by transport trough the boundary $\partial\Omega$. Schemes with this property are called *conservative*. The importance of this property lies in the fact that according to the celebrated Lax-Wendroff theorem [22], if a conservative scheme converges as $\max|\Omega_i| \downarrow 0$, then its solution satisfies the jump condition (4.3). Experience shows that this is usually not the case for nonconservative schemes.

In the present work, the cells Ω_i will be triangles, because this gives maximal flexibility in grid generation, and because triangles are a greater challenge than for instance hexahedra, for which earlier methods developed for structured grids are more directly applicable. In structured grids, every interior vertex belongs to a fixed number of cells. In unstructured grids, the number of cells to which an interior cell belongs is arbitrary. Our method will be developed for unstructured grids, for flexibility in grid generation.

4.3 Discretization of the divergence of the velocity

We integrate the continuity equation for incompressible flow (2.12) over a control volume consisting of one triangle, and apply the divergence theorem:

$$\int_{\Omega_i} (\nabla \cdot \mathbf{u}) d\Omega = \oint_{\partial\Omega_i} (\mathbf{u} \cdot \mathbf{n}) d\Gamma = 0, \tag{4.10}$$

where $\mathbf{n}$ is the outward normal.

Local mass conservation means that this equation should be satisfied exactly. Therefore we choose the averaged normal velocity components

$$u_e = \frac{1}{l_e} \int_{l_e} (\mathbf{u} \cdot \mathbf{N}_e) dl. \tag{4.11}$$

as primary velocity unknowns, where $\mathbf{N}_e$ is one of the two possible unit normal vectors in edge e. Equation (4.10) becomes

$$\sum_{e=1}^{3} u_e \bar{l}_e = 0, \tag{4.12}$$

where $\bar{l}_e = (\mathbf{N}_e \cdot \mathbf{n}_e) l_e$, $\mathbf{n}_e$ is the outward normal in face e, and l_e is the length of face e.

This implies that the average divergence of a vector $\mathbf{v}$ (in our case the velocity or the momentum) in cell Ω_i can be computed exactly as

$$(\nabla \cdot \mathbf{v})_{\Omega_i} = \frac{1}{|\Omega_i|} \sum_{e=1}^{3} v_e \bar{l}_e, \tag{4.13}$$

where

$$v_e = \frac{1}{l_e} \int_{l_e} (\mathbf{v} \cdot \mathbf{N}_e) dl. \tag{4.14}$$

More generally, the average divergence of $\mathbf{v}$ in a union of cells U can be computed as

$$(\nabla \cdot \mathbf{v})_U = \frac{1}{|U|} \sum_{e=1}^{k} v_e \bar{l}_e, \tag{4.15}$$

where summation runs over the boundary faces of this union.

4.4 Discretization of the continuity equation for compressible flow

The semi-discretized density equation for compressible flow (3.13) is integrated over each triangle of the mesh, and the divergence theorem is applied to the convective term. Taking the triangle Ω_1 depicted in Fig. 4.1 for example, one obtains

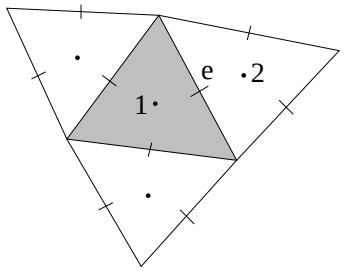

Figure 4.1: Control volume used for the scalar equations.

$$\int_{\Omega_1} \frac{\rho^{n+1}-\rho^n}{\Delta t} d\Omega + \oint_{\partial\Omega_1} \rho^{n+1}(\mathbf{u}^n \cdot \mathbf{n}) d\Gamma = 0. \tag{4.16}$$

We take the triangle averages of ρ as primary density variables. After approximating the line integral in the convective term, equation (4.16) reduces to

$$|\Omega_1| \frac{\rho_1^{n+1}-\rho_1^n}{\partial t} + \sum_e \rho_e^{n+1} u_e^n \overline{l}_e = 0, \tag{4.17}$$

where summation takes place over the three faces.

In the first order upwind scheme [36] the density in face e is taken to be

$$\rho_e^{n+1} = \begin{cases} \rho_1^{n+1} & \text{if} \quad u_e^n \overline{l}_e \geq 0; \\ \rho_2^{n+1} & \text{if} \quad u_e^n \overline{l}_e < 0; \end{cases} \tag{4.18}$$

if e is an internal face, and $\rho_e^{n+1} = \rho_1^{n+1}$ if e is a boundary face where ρ is not prescribed. Note that if $u_e^n \overline{l}_e > 0$ then the flow direction is out of cell 1, otherwise into.

In order to obtain a more accurate scheme, we use upwind-biased linear interpolation to approximate ρ_e^{n+1}. This will be discussed in Sect. 4.5.

The convecting velocity u_e^n is computed by dividing the convected momentum m_e^n by the density ρ_e^n, which needs to be computed. In the first order scheme it is computed as an area-weighted average of the densities in the two neighboring cells:

$$\rho_e^n = \frac{\Omega_2}{\Omega_1+\Omega_2}\rho_1^n + \frac{\Omega_1}{\Omega_1+\Omega_2}\rho_2^n. \tag{4.19}$$

If e is a boundary face and cell 2 is missing then we take $\rho_e^n = \rho_1^n$.

In the new scheme we use face-centered linear reconstruction (see Sect. 4.5) to compute ρ_e^n.

4.5 Linear reconstruction of scalars

A convected scalar quantity ψ is reconstructed in face e by using the formula

$$\psi_e = \begin{cases} \psi_1 + \nabla\psi_{1,e} \cdot (\mathbf{r}_e - \mathbf{r}_1) & \text{if} \quad u_e \overline{l}_e \geq 0; \\ \psi_2 + \nabla\psi_{2,e} \cdot (\mathbf{r}_e - \mathbf{r}_2) & \text{if} \quad u_e \overline{l}_e < 0, \end{cases} \tag{4.20}$$

where $\mathbf{r} = [x\ y]^T$ is the position vector. The gradient $\nabla\psi$ is determined by least squares approximation either from the cells surrounding the upwind cell (1 or 2) or from the cells surrounding the upwind vertex (A or B), as illustrated in Fig. 4.2. These two methods will be called *cell-based* and *vertex-based reconstruction*,

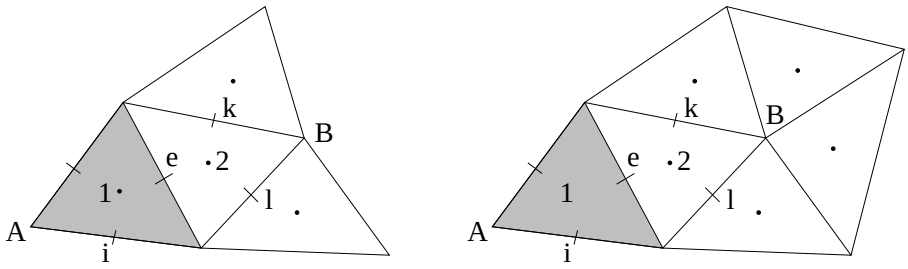

Figure 4.2: Reconstruction stencil in the case when fluid flows into the control volume based in the upwind cell (left) and the upwind vertex (right). Cells that are used for reconstruction are marked by dots.

respectively. Cell-based reconstruction typically involves fewer cells (cf. Fig. 4.2), and gives a unique gradient per cell (i.e. $\nabla\psi_{2,e}$ does not depend on e), which does not hold for the vertex-based reconstruction. The vertex-based reconstruction uses quantities from the upwind side of the edge only; this may enhance monotonicity of numerical solutions. Both approaches were tested and results are presented in Sect. 5.

We considered two ways to determine the gradient using the vertex-based reconstruction. One possibility, assuming $u_e \overline{l}_e < 0$, is to find a and $\nabla\psi_B = [b, c]$ such that a linear polynomial

$$P_B(\mathbf{r}) = a + \nabla\psi_B \cdot (\mathbf{r} - \mathbf{r}_B) \tag{4.21}$$

matches the values of the scalar in the cells surrounding vertex B as accurately as possible in the least-squares sense, and to set $\nabla\psi_{2,e} \equiv \nabla\psi_B$ in (4.20). The other possibility is to postulate, similarly to (4.20),

$$\psi(\mathbf{r}) = \psi_2 + \nabla\psi_{2,e} \cdot (\mathbf{r} - \mathbf{r}_2) \tag{4.22}$$

and to determine $\nabla\psi_{2,e}$ by least squares approximation. In the first case one linear polynomial is associated with each vertex. In the second case two linear polynomials are associated with each face, one for each possible flow direction. Since there are roughly three times more faces then vertices, the second method requires six times more memory, while it is found to be only a bit more accurate.

We also tried not to ignore the constant part a of the linear polynomial (4.21) and to use it instead of ψ_2. However, the resulting scheme did not converge in time.

4.6 Linear reconstruction of staggered vector fields

In preparation for a staggered scheme for the momentum equation, to be presented in Sect. 4.11, the following sections will be devoted to reconstruction of vector fields from staggered components.

The scheme developed in [36, 37] uses the following reconstruction to compute a vector component in a cell:

$$\mathbf{N} \cdot \mathbf{m} \approx \xi_i m_i + \xi_j m_j, \tag{4.23}$$

where i and j are two faces of this cell, and ξ_i and ξ_j are chosen such that

$$\mathbf{N} = \xi_i \mathbf{N}_i + \xi_j \mathbf{N}_j. \tag{4.24}$$

This reconstruction is exact for piecewise constant vector fields only.

In order to obtain higher order schemes, the velocity vector field needs to be reconstructed from staggered data with sufficient accuracy. Second order accurate upwind-biased schemes have been developed on collocated grids using polynomial reconstruction (see for example [4]). Reconstruction from staggered data has not yet been much investigated. Polynomial reconstruction in the staggered case is found in [30], which discusses reconstruction on quadrilaterals. We will discuss the triangular staggered case.

We cannot obtain a second-order accurate reconstruction formula for the momentum by adding the gradient part to the first order reconstruction formula (4.23), like we did in the scalar case, because (4.23) is nowhere more than first order accurate. In the scalar case the first order reconstruction is second order accurate in the cell center. Second order reconstruction must be second order accurate when the position vector is zero.

To approximate $\mathbf{m}$ in the vicinity of point with coordinate vector $\mathbf{r}_0$, we postulate a piecewise linear approximation of the following form:

$$\mathbf{m}(\mathbf{r}) \approx \mathbf{P}(\mathbf{r}) = \mathbf{a} + \mathbf{b}x + \mathbf{c}y, \tag{4.25}$$

where $\mathbf{r} = [x\ y]^T$ is the position vector of a point where $\mathbf{m}$ is to be reconstructed, relative to point $\mathbf{r}_0$. Point $\mathbf{r}_0$ may be a face center, a cell centroid, or a vertex. We want to determine $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ such that the face average of $\mathbf{N}_e \cdot \mathbf{P}(\mathbf{r})$ matches the normal components m_e for face e belonging to a certain set (that will be called the *reconstruction stencil*) as closely as possible in the least squares sense. Fig. 4.3 shows possible reconstruction stencils. This leads to the following linear system:

$$\mathbf{N}_e \cdot \mathbf{P}(\mathbf{r}_e) = m_e, \qquad \text{for each face } e \text{ in the reconstruction stencil.} \tag{4.26}$$

This system can be written as

$$Mc = m, \tag{4.27}$$

where

$$M = \begin{bmatrix} N_{x,1} & N_{y,1} & N_{x,1}x_1 & N_{x,1}y_1 & N_{y,1}x_1 & N_{y,1}y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{x,k} & N_{y,k} & N_{x,k}x_k & N_{x,k}y_k & N_{y,k}x_k & N_{y,k}y_k \end{bmatrix}, \tag{4.28}$$

$$c = \begin{bmatrix} a_1 & a_2 & b_1 & b_2 & c_1 & c_2 \end{bmatrix}^T, \quad m = \begin{bmatrix} m_1 & \dots & m_k \end{bmatrix}^T, \tag{4.29}$$

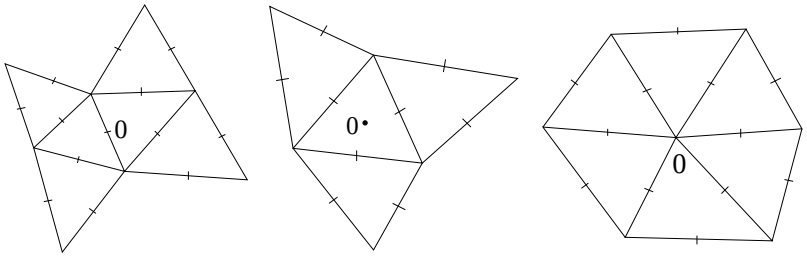

Figure 4.3: Examples of reconstruction stencils for face-based (left), cell-based (middle) and vertex-based (right) linear reconstruction.

and k is the number of faces in the reconstruction stencil.

If the tangential momentum component m_e^t is prescribed as a boundary condition in some of the boundary faces belonging to the reconstruction stencil, equation

$$\mathbf{T}_e \cdot \mathbf{P}(\mathbf{r}_e) = m_e^t \tag{4.30}$$

is added to system (4.26) where $\mathbf{T}_e = (-N_{y,e}, N_{x,e})$ and $m_e^t = \mathbf{T}_e \cdot \mathbf{m}_e$. This can be done for all faces where the tangential momentum is prescribed, or only for the faces closer to the center of the reconstruction stencil.

In [30] it is noted that the system of equations for the reconstruction coefficients may become singular. Here we analyze this situation. In order to determine the least squares solution for c it is necessary that $\text{rank}(M) = 6$. If $\text{rank}(M) < 6$, the stencil needs to be enlarged, or the degree of P must be lowered. We will pursue the first option only. On general unstructured grids it is difficult to specify a priori reconstruction stencils such that $\text{rank}(M) = 6$. However, we can say the following: one triangle Ω in the reconstruction stencil suffices to determine $\nabla \cdot \mathbf{m} = b_1 + c_2$ because

$$b_1 + c_2 = \frac{1}{|\Omega|} \oint_{\partial\Omega} \mathbf{n} \cdot \mathbf{P}(\mathbf{r}) d\Gamma = \frac{1}{|\Omega|} \sum_{e=1}^{3} \mathbf{N}_e \cdot \mathbf{P}(\mathbf{r}_e) \bar{l}_e \tag{4.31}$$

(see 4.26). Therefore

$$\text{rank}(M) \leq k - \tilde{k} + 1, \tag{4.32}$$

where $\tilde{k}$ is the number of triangles in the reconstruction stencil. Hence, for the stencil in Fig. 4.4 we have $\text{rank}(M) \leq 5$, so this stencil is too small, and additional faces must be added.

An additional rank reduction may occur in special situations. Fig 4.5 shows an enlarged vertex-based reconstruction stencil for a boundary vertex 0. Face centers 1, 2 and 3 are collinear, so there exist coefficients α and β such that $\mathbf{r}_3 = \alpha\mathbf{r}_1 + \beta\mathbf{r}_2$ and $\alpha + \beta = 1$. In the case presented in Fig. 4.5, $\alpha = \frac{1}{2}$ and $\beta = \frac{1}{2}$. These faces are also parallel, and we shall suppose that $\mathbf{N}_1 = \mathbf{N}_2 = \mathbf{N}_3$. By substituting this in the left hand side of the equation (4.26) for face 3 we obtain

$$\mathbf{N}_3 \cdot \mathbf{P}(\mathbf{r}_3) = \mathbf{N}_3 \cdot \mathbf{P}(\alpha\mathbf{r}_1 + \beta\mathbf{r}_2) = \alpha\mathbf{N}_1\mathbf{P}(\mathbf{r}_1) + \beta\mathbf{N}_2\mathbf{P}(\mathbf{r}_2). \tag{4.33}$$

The equation for face 3 is linearly dependent on the equations for faces 1 and 2, and $\text{rank}(M) \leq k - \tilde{k} + 1 - 1 < 6$.

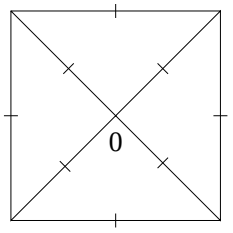

Figure 4.4: Insufficient reconstruction stencil for vertex-based linear reconstruction.

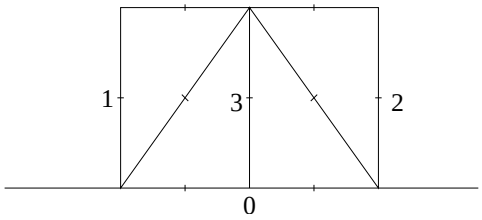

Figure 4.5: Reconstruction stencil at a boundary.

If the grid is unstructured, stencils close to the stencil of Fig. 4.5 frequently appear at boundaries, resulting in ill-conditioned linear systems. In order to detect such situations, singular value decomposition (SVD) is used to find the pseudo-inverse of M. The stencil is enlarged if some singular value is less than some threshold. Because the matrices involved are small, SVD takes only a small part of total computing time.

In order to match the normal velocities in the faces closer to the center more closely than those in the outer part of the reconstruction stencil, we use weight 1 for the equations related to the central faces, and 10^{-2} for all the others. Central faces are the faces meeting in the central vertex in the case of the vertex-based reconstruction, faces of the central triangle in the case of the cell-based reconstruction, or the faces of the two central triangles in the case of the face-based reconstruction. This gives practically absolute priority to the nearest neighbors, but still keeps the singular values of the reconstruction matrix relatively large. Numerical experiments have shown that in this matrix two groups of singular values can be distinguished: those proportional to the larger weight, and those proportional to the smaller weight. For this reason using too small weights can have negative effect on the accuracy. On the other hand, numerical experiments also show that the momentum matrix is better conditioned if the nearest neighbors have much higher weights than the distant faces. Using ratio 1/1000 gives almost the same results. Results are not very sensitive to this parameter.

Still, matrix $M^T M$ emerging from (4.26) when the least squares or the SVD method is used becomes very ill-conditioned as the mesh is refined. This problem is also present in the scalar case. Requirement that a one-dimensional polynomial of degree $n-1$ has specified values in points $x_1, \ldots, x_n$ results in Vandermonde matrix. Determinant of this matrix is $\prod_{i>j}(x_i - x_j)$. As points

x_i get closer to each other, the matrix approaches a singular one. Our situation is similar. In order to avoid growing ill-conditioning, we scale $\mathbf{r}_i$ by some typical length h, for example the length of the central face, the square root of the area of the central cell, or the average length of the faces meeting in the central vertex. Hence, in (4.25)–(4.27) instead of $\mathbf{r}$ and $\mathbf{P}$ we use

$$\tilde{\mathbf{r}} = \begin{bmatrix} \tilde{x} & \tilde{y} \end{bmatrix}^T = \frac{\mathbf{r}}{h}, \quad \tilde{\mathbf{P}}(\tilde{\mathbf{r}}) = \mathbf{P}(\mathbf{r}) = \mathbf{a} + \tilde{\mathbf{b}}\tilde{x} + \tilde{\mathbf{c}}\tilde{y}, \quad \tilde{\mathbf{b}} = h\mathbf{b}, \quad \tilde{\mathbf{c}} = h\mathbf{c}. \tag{4.34}$$

We solve for the scaled coefficients $\mathbf{a}$, $\tilde{\mathbf{b}}$ and $\tilde{\mathbf{c}}$, which are used to calculate $\mathbf{b}$ and $\mathbf{c}$.

The coefficients of the linear polynomial are obtained in the form

$$[a_1, \dots c_2]^T = M^+ \cdot [m_1, \dots m_k]^T, \tag{4.35}$$

where M^+ is the pseudo-inverse of matrix M. Matrix M^+ depends only on the grid and on the weights, and does not change in time. It can be calculated in advance and used later in each time step. This implies that a $6 \times k$ matrix must be stored for each vertex, cell, or face.

4.6.1 Divergence-free linear reconstruction

Since $b_1 + c_2 = \nabla \cdot \mathbf{P}$ in (4.25), we can require that

$$b_1 + c_2 = d = \nabla \cdot \mathbf{m} \tag{4.36}$$

exactly, where the divergence is computed as in (4.15):

$$d = \frac{1}{|\Omega|} \int \nabla \cdot \mathbf{m} d\Omega = \frac{1}{|\Omega|} \sum_{\alpha=1}^{k_1} \overline{l}_\alpha m_\alpha. \tag{4.37}$$

Let $b_1 = d/2 + b$, $c_2 = d/2 - b$, and

$$\overline{\mathbf{P}}(\mathbf{r}) = \mathbf{a} + \begin{bmatrix} b & c_1 \\ b_2 & -b \end{bmatrix} \mathbf{r}. \tag{4.38}$$

Note that $\nabla \cdot \overline{\mathbf{P}} = 0$; therefore $\overline{\mathbf{P}}$ will be called the *divergence-free part* of $\mathbf{P}$. The reconstruction polynomial with specified divergence d can be represented in the following way:

$$\mathbf{P}(\mathbf{r}) = \overline{\mathbf{P}}(\mathbf{r}) + \frac{d}{2}\mathbf{r}. \tag{4.39}$$

The number of free parameters has dropped from six to five.

Each face e in the reconstruction stencil contributes the following equation:

$$\mathbf{N}_e \cdot \overline{\mathbf{P}}(\mathbf{r}_e) = m_e - \frac{1}{2}(\mathbf{N}_e \cdot \mathbf{r})d. \tag{4.40}$$

The least squares solution is given by

$$\begin{bmatrix} a_1 \\ a_2 \\ b \\ c_1 \\ b_2 \end{bmatrix} = M^+ \cdot \left(\begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} - \frac{1}{2} \begin{bmatrix} N_{x,1}x_1 + N_{y,1}y_1 \\ \vdots \\ N_{x,k}x_k + N_{y,k}y_k \end{bmatrix} d \right), \tag{4.41}$$

where M^+ is the pseudo-inverse of the matrix M of system (4.40). We substitute (4.37) in (4.41) in order to obtain

$$\begin{bmatrix} a_1 \\ a_2 \\ b \\ c_1 \\ b_2 \end{bmatrix} = M^+ \cdot \left(\begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} - \frac{1}{2\,|\Omega|} \begin{bmatrix} N_{x,1}x_1 + N_{y,1}y_1 \\ \vdots \\ N_{x,k}x_k + N_{y,k}y_k \end{bmatrix} \begin{bmatrix} \overline{l}_1 & \dots & \overline{l}_{k_1} \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ \vdots \\ m_{k_1} \end{bmatrix} \right). \tag{4.42}$$

In this way faces used to specify the divergence d enter the reconstruction stencil.

If at least five faces meet in a vertex, they impose enough conditions to determine the solenoidal part of a vertex-based linear polynomial (see Fig. 4.6 on the left and in the middle). However, if fewer than five faces meet in some vertex, the stencil is enlarged as in Fig. 4.6 on the right (or the order of interpolation must be lowered; this option we do not pursue because we want to preserve the accuracy).

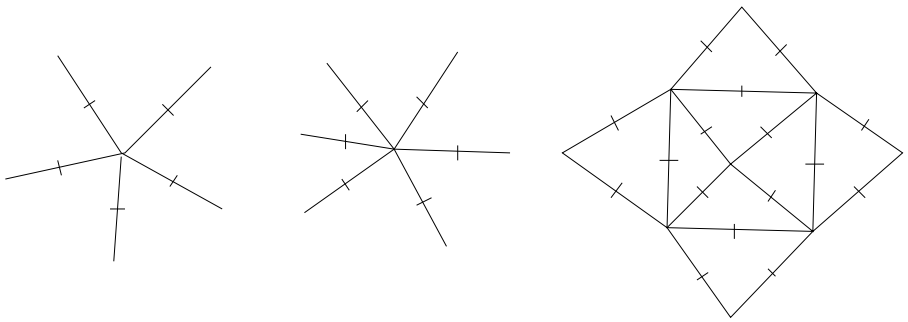

Figure 4.6: Reconstruction stencils for the divergence-free vertex-based linear reconstruction.

Vertex-based divergence-free linear reconstruction has several advantages. The most important one is that it leads to a superlinearly convergent scheme for the Navier-Stokes equations, as will be shown in the following sections, while the non-divergence-free linear reconstruction does not. In addition, it leads to a smaller stencil, and it requires roughly half the memory of the non-divergence-free reconstruction. It does not need weights, thresholds or SVD whenever at least five faces meet in a vertex or if the tangential momentum is prescribed in a sufficient number of faces meeting in the central vertex. It is however not required that at least five faces meet in every internal vertex. Meshes that we used do not have this property.

4.7 Quadratic reconstruction

We postulate

$$\mathbf{m}(\mathbf{r}) \approx \mathbf{P}(\mathbf{r}) = \mathbf{a} + \mathbf{b}x + \mathbf{c}y + \mathbf{d}x^2 + \mathbf{e}xy + \mathbf{f}y^2. \tag{4.43}$$

We have 12 free parameters. The stencils shown in Fig. 4.3 do not contain enough information to determine these parameters, so we must enlarge them. An example of an appropriate cell-based stencil is shown in Fig. 4.7.

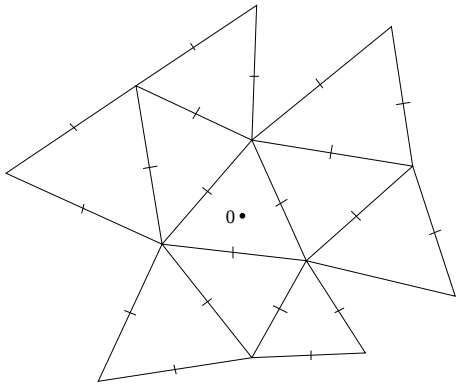

Figure 4.7: Reconstruction stencil for cell-based quadratic reconstruction.

Our primary momentum variables are not the normal momenta in the face centers but their averages along these faces. For this reason, instead of $\mathbf{N}_e \cdot \mathbf{P} = m_e$ we need to satisfy

$$\frac{1}{l_e}\int_{l_e} \mathbf{N}_e \cdot \mathbf{P} dl = m_e \tag{4.44}$$

The average of a linear polynomial along a face is equal to the value of the polynomial in the center of that face, and this is why condition (4.26) was good enough in the linear case. In order to evaluate (4.44) we need the averages of x^2, xy and y^2 over the faces. Face e is parameterized as $x = x_e + tl_{x,e}, y = y_e + tl_{y,e}$, $t \in [-1/2, 1/2]$, where $(l_{x,e}, l_{y,e})$ is the vector of face e and x_e and y_e are the coordinates of the face center. Then

$$\begin{aligned} \frac{1}{l_e}\int_{l_e} x^2 dl = \frac{1}{l_e}\int_{-\frac{1}{2}}^{\frac{1}{2}} (x_e + tl_{x,e})^2 l_e dt = x_e^2 + \frac{l_{x,e}^2}{12}, \\ \frac{1}{l_e}\int_{l_e} xy dl = x_e y_e + \frac{l_{x,e} l_{y,e}}{12}, \\ \frac{1}{l_e}\int_{l_e} y^2 dl = y_e^2 + \frac{l_{y,e}^2}{12}. \end{aligned} \tag{4.45}$$

Each face e in the reconstruction stencil contributes with the following equation:

$$\begin{aligned} N_{x,e}a_1 + N_{y,e}a_2 + N_{x,e}x_e b_1 + N_{y,e}x_e b_2 + N_{x,e}y_e c_1 + N_{y,e}y_e c_2 + \\ + N_{x,e}\left(x_e^2 + \frac{l_{x,e}^2}{12}\right) d_1 + N_{y,e}\left(x_e^2 + \frac{l_{x,e}^2}{12}\right) d_2 + \\ + N_{x,e}\left(x_e y_e + \frac{l_{x,e} l_{y,e}}{12}\right) e_1 + N_{y,e}\left(x_e y_e + \frac{l_{x,e} l_{y,e}}{12}\right) e_2 + \\ + N_{x,e}\left(y_e^2 + \frac{l_{y,e}^2}{12}\right) f_1 + N_{y,e}\left(y_e^2 + \frac{l_{y,e}^2}{12}\right) f_2 = m_e. \end{aligned} \tag{4.46}$$

If the tangential vector component is given in this face, the appropriate equation is obtained by replacing $\mathbf{N}_e$ by $\mathbf{T}_e$ and m_e by m_e^t .

If this reconstruction is needed for approximation of the viscous term, then it might be advantageous to include the stress prescribed at the outflow boundary in the linear system from which the coefficients of the reconstruction polynomial are calculated. For the tangential stress the following equation needs to be satisfied:

$$\frac{1}{l_e}\int_{l_e}\mu\mathbf{N}_e\begin{bmatrix}2\dfrac{\partial P_x}{\partial x}-\dfrac{2}{3}\Delta & \dfrac{\partial P_x}{\partial y}+\dfrac{\partial P_y}{\partial x}\\ \dfrac{\partial P_x}{\partial y}+\dfrac{\partial P_y}{\partial x} & 2\dfrac{\partial P_y}{\partial y}-\dfrac{2}{3}\Delta\end{bmatrix}\mathbf{T}_e dl=\sigma_{nt,e}, \tag{4.47}$$

where $\sigma_{nt,e}$ is the average of the given tangential stress over face e. This leads to the following equation:

$$\begin{aligned}&\mu\left\{(\tfrac{4}{3}N_{x,e}T_{x,e}-\tfrac{2}{3}N_{y,e}T_{y,e})b_1+(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})b_2+\right.\\&+(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})c_1+(\tfrac{4}{3}N_{y,e}T_{y,e}-\tfrac{2}{3}N_{x,e}T_{x,e})c_2+\\&+(\tfrac{8}{3}N_{x,e}T_{x,e}-\tfrac{4}{3}N_{y,e}T_{y,e})x_e d_1+2(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})x_e d_2+\\&+\left((\tfrac{4}{3}N_{x,e}T_{x,e}-\tfrac{2}{3}N_{y,e}T_{y,e})y_e+(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})x_e\right)e_1+\\&+\left((\tfrac{4}{3}N_{y,e}T_{y,e}-\tfrac{2}{3}N_{x,e}T_{x,e})x_e+(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})y_e\right)e_2+\\&\left.+2(N_{x,e}T_{y,e}+N_{y,e}T_{x,e})x_e f_1+(\tfrac{8}{3}N_{y,e}T_{y,e}-\tfrac{4}{3}N_{x,e}T_{x,e})y_e f_2\right\}=\sigma_{nt}.\end{aligned} \tag{4.48}$$

If the normal stress is given in this face, the appropriate equation is obtained by replacing $\mathbf{T}_e$ by $\mathbf{N}_e$ and σ_{nt} by σ_{nn}.

As in the linear case, we combat rounding errors by scaling by h, and actually use $\tilde{\mathbf{r}}$ instead of $\mathbf{r}$ in (4.43). Later we transform the coefficients of this scaled polynomial into the coefficients of the original one by dividing the coefficients with linear terms by h and the coefficients with quadratic terms by h^2.

4.7.1 Divergence-free quadratic reconstruction

The divergence of a quadratic polynomial is a linear function

$$\nabla\cdot\mathbf{P}(\mathbf{r})=b_1+c_2+(2d_1+e_2)x+(e_1+2f_2)y=\alpha+\beta x+\gamma y. \tag{4.49}$$

It is required that $\nabla\cdot\mathbf{P}(\mathbf{r})$ matches the divergence of the reconstructed vector in at least three triangles as closely as possible in the least squares sense. For example, if a cell-based quadratic polynomial is being calculated, then the divergence is calculated from the central and the three surrounding triangles in Fig. 4.7. For each of these triangles the following equation needs to be satisfied as closely as possible:

$$\int_\Omega(\alpha+\beta x+\gamma y)d\Omega=\sum_e m_e\bar{l}_e, \tag{4.50}$$

where the summation runs over the faces of triangle Ω. From these conditions parameters α, β and γ are calculated. By substituting (4.49) in (4.43) one obtains

$$\begin{aligned}P_x(\mathbf{r})=&\quad a_1+(\alpha/2+b)x+c_1y+d_1x^2+(\gamma-2f_2)xy+f_1y^2\\P_y(\mathbf{r})=&\quad a_2+b_2x+(\alpha/2-b)y+d_2x^2+(\beta-2d_1)xy+f_2y^2\end{aligned} \tag{4.51}$$

When the system (4.44) is formed, terms containing α, β and γ are moved to the right hand side, and the rest is calculated in analogy with the linear case.

4.8 Cubic reconstruction

We consider cubic polynomials:

$$\mathbf{P}(\mathbf{r}) = \mathbf{a} + \mathbf{b}x + \mathbf{c}y + \mathbf{d}x^2 + \mathbf{e}xy + \mathbf{f}y^2 + \mathbf{g}x^3 + \mathbf{h}x^2y + \mathbf{i}xy^2 + \mathbf{j}x^3. \quad (4.52)$$

Here we have 20 free parameters. The stencil is obtained by choosing faces closest to the stencil center until the rank is 20 and all singular values are sufficiently large. Typical stencil contains around 30 faces.

In the similar manner as before, we shall need the following integrals:

$$\begin{aligned} I_{x^3} &= \frac{1}{l_e}\int_{l_e} x^3 dl = x_e^3 + x_e\frac{l_{x,e}^2}{4}, \\ I_{x^2y} &= \frac{1}{l_e}\int_{l_e} x^2 y dl = x_e^2 y_e + \frac{l_{x,e}}{12}(l_{x,e}y_e + 2l_{y,e}x_e), \\ I_{xy^2} &= \frac{1}{l_e}\int_{l_e} xy^2 dl = x_e y_e^2 + \frac{l_{y,e}}{12}(2l_{x,e}y_e + l_{y,e}x_e), \\ I_{y^3} &= \frac{1}{l_e}\int_{l_e} y^3 dl = y_e^3 + y_e\frac{l_{y,e}^2}{4}. \end{aligned} \quad (4.53)$$

For each face e in the reconstruction stencil the following equation needs to be satisfied as closely as possible in the least squares sense:

$$\begin{aligned} &N_{x,e}a_1 + N_{y,e}a_2 + N_{x,e}x_e b_1 + N_{y,e}x_e b_2 + N_{x,e}y_e c_1 + N_{y,e}y_e c_2 + \\ &+N_{x,e}I_{xx}d_1 + N_{y,e}I_{xx}d_2 + N_{x,e}I_{xy}e_1 + N_{y,e}I_{xy}e_2 + N_{x,e}I_{yy}f_1 + N_{y,e}I_{yy}f_2 + \\ &+N_{x,e}I_{x^3}g_1 + N_{y,e}I_{x^3}g_2 + N_{x,e}I_{x^2y}h_1 + N_{y,e}I_{x^2y}h_2 + \\ &+N_{x,e}I_{xy^2}i_1 + N_{y,e}I_{xy^2}i_2 + N_{x,e}I_{y^3}j_1 + N_{y,e}I_{y^3}j_2 = m_e, \end{aligned} \quad (4.54)$$

where

$$I_{xx} = x_e^2 + \frac{l_{x,e}^2}{12}, \qquad I_{xy} = x_e y_e + \frac{l_{x,e}l_{y,e}}{12}, \qquad I_{yy} = y_e^2 + \frac{l_{y,e}^2}{12}. \quad (4.55)$$

Equation for the tangential stress follows:

$$\begin{aligned} &\mu\left\{(\tfrac{4}{3}N_{x,e}T_{x,e} - \tfrac{2}{3}N_{y,e}T_{y,e})b_1 + (N_{x,e}T_{y,e} + N_{y,e}T_{x,e})b_2 + \right. \\ &+(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})c_1 + (\tfrac{4}{3}N_{y,e}T_{y,e} - \tfrac{2}{3}N_{x,e}T_{x,e})c_2 + \\ &+(\tfrac{8}{3}N_{x,e}T_{x,e} - \tfrac{4}{3}N_{y,e}T_{y,e})x_e d_1 + 2(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})x_e d_2 + \\ &+\left[(\tfrac{4}{3}N_{x,e}T_{x,e} - \tfrac{2}{3}N_{y,e}T_{y,e})y_e + (N_{x,e}T_{y,e} + N_{y,e}T_{x,e})x_e\right] e_1 + \\ &+\left[(\tfrac{4}{3}N_{y,e}T_{y,e} - \tfrac{2}{3}N_{x,e}T_{x,e})x_e + (N_{x,e}T_{y,e} + N_{y,e}T_{x,e})y_e\right] e_2 + \\ &+2(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})x_e f_1 + (\tfrac{8}{3}N_{y,e}T_{y,e} - \tfrac{4}{3}N_{x,e}T_{x,e})y_e f_2 + \\ &+(\tfrac{4}{3}N_{x,e}T_{x,e} - \tfrac{2}{3}N_{y,e}T_{y,e})I_{xx}g_1 + 3(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{xx}g_2 + \\ &+\left[(\tfrac{8}{3}N_{x,e}T_{x,e} - \tfrac{4}{3}N_{y,e}T_{y,e})I_{xy} + (N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{xx}\right] h_1 + \\ &+\left[(\tfrac{4}{3}N_{y,e}T_{y,e} - \tfrac{2}{3}N_{x,e}T_{x,e})I_{xx} + 2(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{xy}\right] h_2 + \\ &+\left[(\tfrac{4}{3}N_{y,e}T_{y,e} - \tfrac{2}{3}N_{x,e}T_{x,e})I_{yy} + 2(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{xy}\right] i_1 + \\ &+\left[(\tfrac{8}{3}N_{y,e}T_{y,e} - \tfrac{4}{3}N_{x,e}T_{x,e})I_{xy} + (N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{yy}\right] i_2 + \\ &\left.+ 3(N_{x,e}T_{y,e} + N_{y,e}T_{x,e})I_{yy}j_1 + (\tfrac{4}{3}N_{y,e}T_{y,e} - \tfrac{2}{3}N_{x,e}T_{x,e})I_{yy}j_2\right\} = \sigma_{nt}. \end{aligned} \quad (4.56)$$

For normal stress $\mathbf{T}_e$ is replaced by $\mathbf{N}_e$ and σ_{nt} is replaced by σ_{nn}.

It is possible to make divergence-free cubic reconstruction in an analogous way to the linear and the quadratic case, although we did not do it because using cubic reconstruction does not pay off as long as we use the trapezoid formula to integrate over the faces.

4.9 A note on boundary conditions in reconstruction

As we noticed before, sometimes it might be advantageous to include boundary conditions in the linear systems that determine the reconstruction polynomials. However, using face averages of boundary conditions for this purpose may not be the best solution. Consider a corner of a computational domain depicted in Fig. 4.8. Suppose that the flow is directed from left to right, so that the momentum

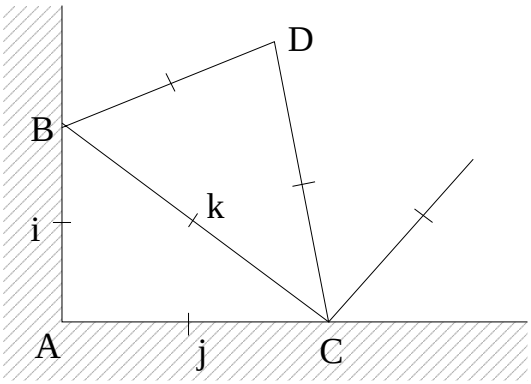

Figure 4.8: A corner.

vector is prescribed at the leftmost boundary, and only the normal momentum component is prescribed at the lower (free-slip) boundary. We want to compute a linear polynomial based in vertex A. We cannot determine such polynomial from faces i, j and k, because we need to determine six degrees of freedom, and we only have four momentum components in these faces, including the boundary conditions. The reconstruction stencil needs to be enlarged with the faces of triangle BCD. If this reconstruction polynomial is needed to evaluate the momentum vector in face k for the convection term of momentum equation (this will be discussed in Sect. 4.11.2), extending the stencil with downwind faces may make the scheme unstable.

In such situation one must remember that by knowing boundary conditions, we know much more than just face averages. We also know boundary conditions in the vertices or in any other point on the boundary curve, and we know the derivatives of these boundary conditions. This information should be used in reconstruction rather than enlarging the reconstruction stencil.

In particular, we may use the momentum vector in vertices A and B, the normal momentum in vertex C, and the normal momentum in face k. This is enough to determine a linear polynomial.

4.10 Hierarchical least squares method

The quality of solutions obtained with a polynomial reconstruction depends sometimes heavily on the choice of the weights used to obtain the reconstruction polynomials (see Sect. 4.6). The weights related to the faces near the center of the stencil should be much larger than those related to the faces more distant, and this can make the matrix of the least squares system ill-conditioned. For large reconstruction stencils that are used with the quadratic and the cubic reconstruction sometimes it is advantageous to match a few central faces as closely as possible in the least squares sense, then to draw some additional information from the closer surrounding faces, and at the end to treat the distant faces. For this we need three very different groups of weights, for example 1, 10^{-4} and 10^{-8}. These weights will be reflected in the singular values. Some singular values will be of order one, some will be of order 10^{-4}, and some of order 10^{-8}. But then if some faces in the second group are almost parallel, one or more of the related singular values will be smaller. This situation is very hard to detect because the number of singular values related to each group depends on the particular reconstruction stencil, so this smaller singular value may be confused with one from the third group. Therefore we do not know if the reconstruction stencil needs to be enlarged. This makes things very complicated and unmanageable.

There is a considerable amount of literature on using the least squares method for reconstruction. This is called the *moving least squares method* (MLS), and it appeared for the first time in [21]. It is particularly popular in meshless methods. For example, in [24] a least squares method with constraints is used to make a reconstruction polynomial which is exact in one point. But this approach cannot be extended to our staggered case in a straightforward way.

For this reason we developed a variant of MLS that does not require choosing weights, allows us to define priority groups of equations, and offers a possibility to easily control the singular values related to each particular group. We shall call this method the *hierarchical least squares* (HLS) method.

The linear system $Ay = b$ is decomposed into a set of linear subsystems:

$$A_i y = b_i, \qquad i = 1, \ldots, s, \tag{4.57}$$

where $y \in \mathbf{R}^n$, $b_i \in \mathbf{R}^{m_i}$ and $\mathrm{rank}(A) \geq n$. We want to find a family of sets $\mathbf{Y}_1 \supseteq \mathbf{Y}_2 \supseteq \cdots \supseteq \mathbf{Y}_s$ such that $\mathbf{Y}_i = \{y \in \mathbf{Y}_{i-1} | \|A_i y - b_i\| \text{ is minimal } \}$ and $\mathbf{Y}_0 = \mathbf{R}^n$.

We calculate the SVD of matrix A_1 and get $A_1 = U_1 S_1 V_1^T$, where U_1 and V_1 are unitary matrices and S_1 is a diagonal matrix containing singular values of matrix A_1. We replace elements of S_1^T by their inverses if they are larger than some tolerance, set others to zero, and denote the obtained matrix by S_1^{-1}. Solution $y^1_{\mathrm{pr}} \in \mathbf{Y}_1$ such that $\|y^1_{\mathrm{pr}}\|$ is minimal is given by

$$y^1_{\mathrm{pr}} = V_1 S_1^{-1} U_1^T b_1. \tag{4.58}$$

Columns of V_1 related to the zero (or below the tolerance) singular values form the basis of the null space $null(A_1) = \{y | A_1 y = 0\}$. We shall denote the matrix consisting only of these columns by V_1^{null}. Now

$$\mathbf{Y}^1 = \{y^1_{\mathrm{pr}} + V_1^{\mathrm{null}} y^1 | y^1 \in \mathbf{R}^{n_1}\}, \tag{4.59}$$

where $n_1 = n - \text{rank}(A_1)$. This is because $\{V_1^{\text{null}} y^1 | y^1 \in \mathbf{R}^{n_1}\} = null(A_1)$ and so $\{y^1_{\text{pr}} + V_1^{\text{null}} y^1 | y^1 \in \mathbf{R}^{n_1}\} = \{y \in \mathbf{R}^n | A_1 y = A_1 y^1_{\text{pr}}\}$.

We proceed by induction. Suppose that we found some $y_0^i \in \mathbf{Y}^i$

$$y_0^i = E_i \begin{bmatrix} b_1 \\ \vdots \\ b_i \end{bmatrix}, \tag{4.60}$$

and that $\mathbf{Y}^i$ is represented by

$$\mathbf{Y}^i = \{y_0^i + F_i y^i | y^i \in \mathbf{R}^{n_i}\}, \tag{4.61}$$

where $n_i > 0$. By substituting $y = y_0^i + F_i y^i$ into equation $A_{i+1} y = b_{i+1}$ we obtain

$$A_{i+1} F_i y^i = b_{i+1} - A_{i+1} y_0^i. \tag{4.62}$$

We calculate the SVD of $A_{i+1} F_i = U_{i+1} S_{i+1} V_{i+1}^T$ and find

$$y^{i+1}_{\text{pr}} = V_{i+1} S_{i+1}^{-1} U_{i+1}^T \left(b_{i+1} - A_{i+1} y_0^i\right) \tag{4.63}$$

If $A_{i+1} F_i$ has a nontrivial null space then columns of V_{i+1} related to the zero singular values form the basis of this null space. We shall denote the matrix consisting only of these columns by V_{i+1}^{null}.

If $y^i = y^{i+1}_{\text{pr}} + V_{i+1}^{\text{null}} y^{i+1}$ where $y^{i+1} \in \mathbf{R}^{n_{i+1}}$, then $A_{i+1} F_i y^i = A_{i+1} F_i y^{i+1}_{\text{pr}}$ and so

$$\mathbf{Y}^{i+1} = \{y_0^i + F_i (y^{i+1}_{\text{pr}} + V_{i+1}^{\text{null}} y^{i+1}) | y^{i+1} \in \mathbf{R}^{n_{i+1}}\} \tag{4.64}$$

or

$$\mathbf{Y}^{i+1} = \{y_0^{i+1} + F_{i+1} y^{i+1} | y^{i+1} \in \mathbf{R}^{n_{i+1}}\}, \tag{4.65}$$

where

$$y_0^{i+1} = E_{i+1} \begin{bmatrix} b_1 \\ \vdots \\ b_{i+1} \end{bmatrix}, \qquad F_{i+1} = F_i V_{i+1}^{\text{null}},$$
$$E_{i+1} = \begin{bmatrix} E_i - F_i V_{i+1} S_{i+1}^{-1} U_{i+1}^t A_{i+1} E_i & F_i V_{i+1} S_{i+1}^{-1} U_{i+1}^T \end{bmatrix}. \tag{4.66}$$

If $A_{i+1} F_i$ does not have a nontrivial null space, then $\mathbf{Y}^{i+1}$ contains only y_0^{i+1}, which is the final solution to the set of systems (4.57). In this case it should be $i + 1 = n$, otherwise the subsystems $i + 2, \ldots, n$ do not influence the solution. If $i + 1 = n$ and $A_{i+1} F_i$ has a nontrivial null space, this means that $\text{rank}(A) < n$. In this case we can add an additional subsystem and continue without restarting the algorithm.

We also developed a version of this method that uses an ordinary direct solver and least squares method instead of SVD. It works well and it gives the same result as the method with the SVD. However, it is much more complicated. Stability is also not guaranteed, although in our experiments it was always working. Since our systems are small and do not grow with the mesh, it is advisable to use the SVD.

If there are only two groups of equations and if the equations of the first group are linearly independent, this method is equivalent to the so-called least squares method with constraints, which finds the best solution of a linear system

in the least squares sense, while satisfying some equations (constraints) exactly. However, the equations of the first group may be contradictory. For example, when computing a face-based linear polynomial, the first group consists of the five faces of the two triangles sharing the central face (see 4.3 on the left). The rank of the related subsystem is 4, because two triangles are included, so the divergence is specified twice. Therefore in general it is not possible to match the prescribed normal components exactly, and the least squares method with constraints is not applicable.

4.11 Discretization of momentum equation

In the case of the momentum equation we associate the union Ω_i of two neighboring triangles sharing edge i with each interior edge i, and the neighboring boundary triangle with each boundary edge, and we use these sets as control volumes (see Fig. 4.9). The semi-discretized momentum equation (3.6) is multiplied by $\mathbf{N}_i$ and averaged over these control volumes Ω_i:

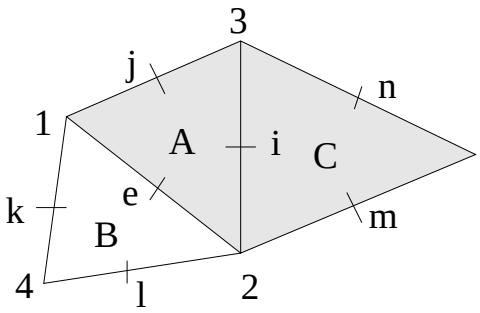

Figure 4.9: Control volume for the momentum equation.

$$\frac{1}{|\Omega_i|}\int_{\Omega_i}\frac{\mathbf{N}_i\cdot(\mathbf{m}^{n+1}-\mathbf{m}^n)}{\Delta t}d\Omega+\frac{1}{|\Omega_i|}\int_{\Omega_i}\nabla\cdot[\mathbf{u}^n(\mathbf{m}^{n+1}\cdot\mathbf{N}_i)]d\Omega=$$
$$=-\frac{1}{|\Omega_i|}\int_{\Omega_i}\mathbf{N}_i\cdot\nabla p^n d\Omega+\frac{1}{|\Omega_i|}N_{\alpha,i}\int_{\Omega_i}(\tau_{\alpha\beta,\beta})^{n+1} \quad (4.67)$$

If the pressure correction method is used, then superscript $n+1$ is replaced by a star $*$.

The control volumes chosen have the property that summation of the integrals of the inviscid momentum equation over the control volumes (specializing to the inviscid case for brevity)

$$\int_{\Omega_i}\frac{\partial\mathbf{m}}{\partial t}d\Omega+\oint_{\partial\Omega_i}[\mathbf{m}(\mathbf{u}\cdot\mathbf{n})+p\mathbf{n}]d\Gamma=0 \quad (4.68)$$

results in three times the integral of this equation over Ω, i.e.

$$3\int_{\Omega}\frac{\partial\mathbf{m}}{\partial t}d\Omega+3\oint_{\partial\Omega}[\mathbf{m}(\mathbf{u}\cdot\mathbf{n})+p\mathbf{n}]d\Gamma=0 \quad (4.69)$$

This shows that at this stage the discretization is conservative. However, due to the multiplication by $\mathbf{N}_i$, linear combination of equations (4.67) will never result in a conservation law integrated over the computational domain, i.e. we cannot prove that the discretization of the momentum equation is conservative. Hence, the Lax-Wendroff theorem does not apply. Nevertheless, as shown in by numerical experiments [37], numerical solutions satisfy the Rankine-Hugoniot conditions. We also illustrate this in Sect. 5.4.4.

In order to fully specify the numerical scheme, the integrals in (4.67) need to be approximated. We proceed term by term.

4.11.1 Viscous term

We apply the divergence theorem:

$$\frac{1}{|\Omega_i|}N_{\alpha,i}\int_{\Omega_i}\tau_{\alpha\beta,\beta}d\Omega=\frac{1}{|\Omega_i|}N_{\alpha,i}\oint_{\partial\Omega_i}\tau_{\alpha\beta}n_\beta d\Gamma=\frac{1}{|\Omega_i|}N_{\alpha,i}\sum_e\tau_{\alpha\beta,e}N_{\beta,e}\overline{l}_e, \tag{4.70}$$

where $\tau_{\alpha\beta,e}=2\mu\left(\frac{1}{2}\left(u_{\alpha,e,\beta}+u_{\beta,e,\alpha}\right)-\frac{1}{3}\Delta\delta_{\alpha\beta}\right)$ is the deviatoric stress tensor averaged over the control volume face e. This term is discretized implicitly. For brevity we excluded the superscript $n+1$. Even if the flow is incompressible, the velocity divergence Δ should still be included when an implicit method is used, because this is found to improve the properties of the global matrix. It is computed as a linear combination of the normal velocities in faces i, j, k and l by applying formula (4.15) to the union of cells A and B (see Fig. 4.9).

Several approximations for the rate of strain tensor $\frac{1}{2}\left(u_{\alpha,e,\beta}+u_{\beta,e,\alpha}\right)$ have been explored. Their relative merit is investigated by application to the following example.

Example 1. The following equations are solved:

$$\tau_{1\beta,\beta}=2\sin x\sin y,\qquad \tau_{2\beta,\beta}=2\cos x\cos y. \tag{4.71}$$

on the square $[-0.5,0.5]\times[-1.3,-0.3]$. We take $\mu=1$. The exact solution

$$u=\sin x\sin y,\qquad v=\cos x\cos y \tag{4.72}$$

is prescribed at the inflow boundaries. At the outflow boundary the exact stress (Neumann) or the exact velocity (Dirichlet) is prescribed. The velocity vectors are shown in Fig. 4.10.

The local truncation error of a discretized equation is the difference between the left hand side and the right hand side when the exact discretized solution (which is the set of face averages of the exact velocity) is substituted. We shall denote the local truncation error of the momentum equation as τ_m.

The (global truncation) velocity error in face e is defined as

$$\varepsilon_{u,e}=u_e-u_{\mathrm{ex},e}, \tag{4.73}$$

where index ex refers to the exact discretized solution.

The l_2 norm or 2-norm of an algebraic vector $x\in\mathbf{R}^n$ is defined as

$$\|x\|_2=\left[\frac{1}{n}\sum_{e=1}^{n}x_e^2\right]^{1/2}, \tag{4.74}$$

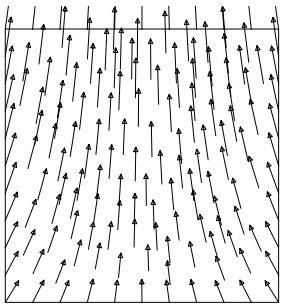

Figure 4.10: Example 1.

whereas the l_∞ or *maximum norm* is defined as

$$\|x\|_\infty = \max_e |x_e|. \tag{4.75}$$

The order of accuracy α is estimated in the following way. It is assumed that some norm of the error depends on h as $\|\varepsilon\| \approx ah^\alpha, \quad h \downarrow 0$, where h is a mesh parameter (in our case, $h = 1/n$, n is the number of edges along each side of the domain), α is the order of accuracy, and a is some constant. This can be written as

$$\frac{\log a}{\log h} + \alpha = \frac{\log \|\varepsilon\|}{\log h}. \tag{4.76}$$

We estimate $\log a$ and α by least squares from a set of results obtained with meshes of the same quality and various h.

This problem was solved on three different types of grids, shown in Fig. 4.11. In order to preserve the properties of the unstructured grids, these grids were not refined by adding points, but random finer grids were created independently.

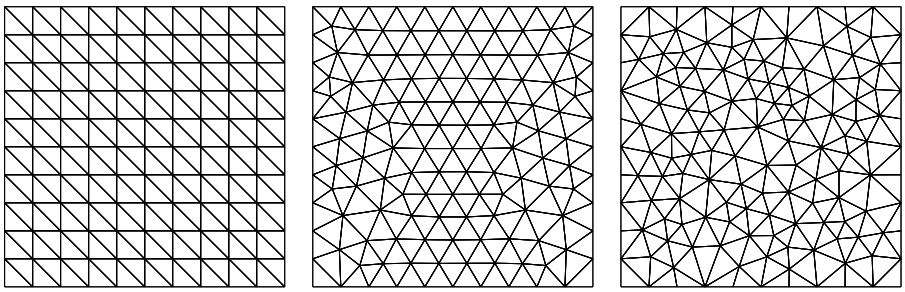

Figure 4.11: Cartesian, regular and irregular grid.

Piecewise constant reconstruction

The rate of strain tensor $\frac{1}{2}(u_{\alpha,e,\beta} + u_{\beta,e,\alpha})$ has to be expressed in terms of the surrounding normal velocity components. In the scheme [36] this is done by constructing a bilinear polynomial based on the velocity vectors in faces i, j, k

and l (see Fig. 4.9). The gradient of this bilinear polynomial is used to compute $u_{\alpha,e,\beta}$ and $u_{\beta,e,\alpha}$. Hence, tangential velocity components are required in faces i, j, k and l.

The missing tangential velocity component in face i $u_i^t = \mathbf{t}_i \cdot \mathbf{u}_i$ is computed as the average of the $\mathbf{t}_i \cdot \mathbf{u}$ velocity component computed in cells A and C. In cell A component $\mathbf{t}_i \cdot \mathbf{u}_A$ is computed from normal velocities u_e and u_j by using formula 4.23, and $\mathbf{t}_i \cdot \mathbf{u}_C$ is computed from u_m and u_n. Such average is called *central reconstruction*. However, this is still a first order reconstruction on an unstructured grid.

This leads to the computational stencil shown in Fig. 4.12. The accuracy of

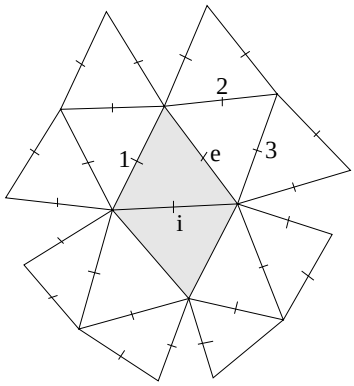

Figure 4.12: Computational stencil for the viscous term.

this method was tested on Example 1. The errors are shown in Fig. 4.13, 4.14 and 4.15. The convergence stagnates on the irregular grid.

Linear reconstruction

For linear reconstruction based in face centers, the computational stencil in the interior of the domain is the same as in the piecewise constant case (see Fig. 4.12).

We apply the resulting scheme to Example 1 with Dirichlet boundary conditions. We solve on three types of grids shown in Fig. 4.11, starting with the irregular one. First we use weights 1 for the central face and for the surrounding four faces, and 10^{-5} for the other faces. We solve the reconstruction systems by SVD, replacing singular values below 10^{-10} by zero. The results are shown in Fig. 4.16. If we use the HLS method with two groups (one for the central five faces or fewer if the boundary is encountered, and one for the rest), we get a very similar result. Convergence when $h \downarrow 0$ is not satisfactory.

Sometimes the local and the global truncation errors jump. A cure for this is to enlarge the reconstruction stencil when an unusually small singular value appears in the reconstruction matrix. If we use weights as described (values 1 and 10^{-5}), matrices arising from reconstruction stencils have two singular values around 10^{-5}. We can enlarge the stencil if a singular value is smaller than $2 \cdot 10^{-6}$. If we use the HLS, we can enlarge the stencil if some singular value in the second group is less than some threshold. Results obtained with

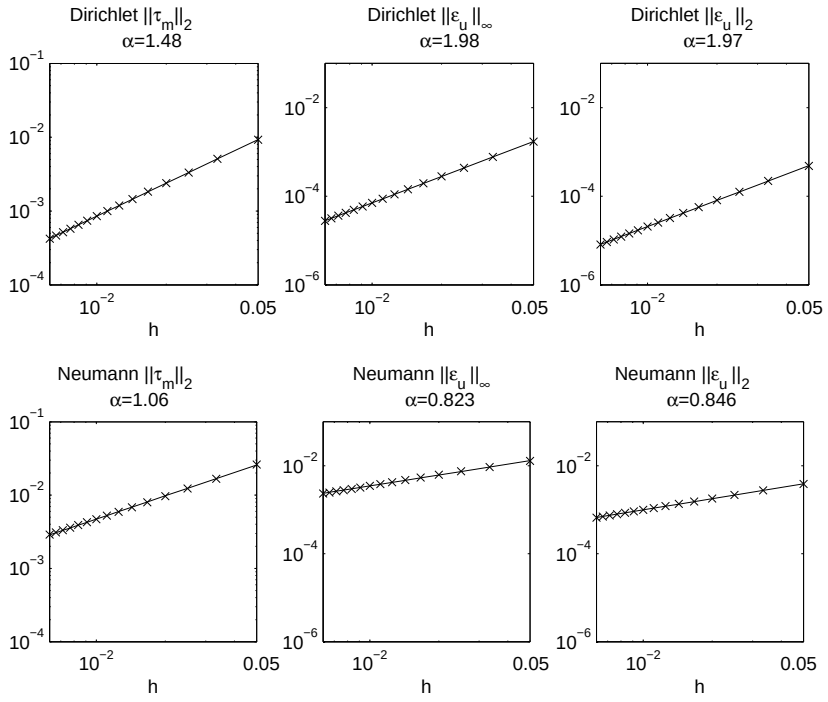

Figure 4.13: Example 1, piecewise constant reconstruction, Cartesian grid.

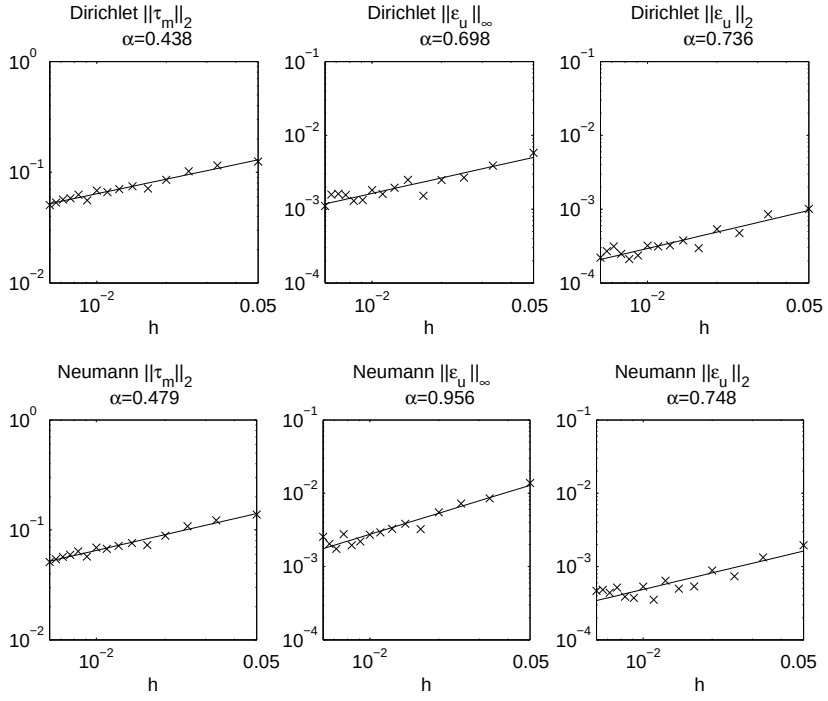

Figure 4.14: Example 1, piecewise constant reconstruction, regular grid.

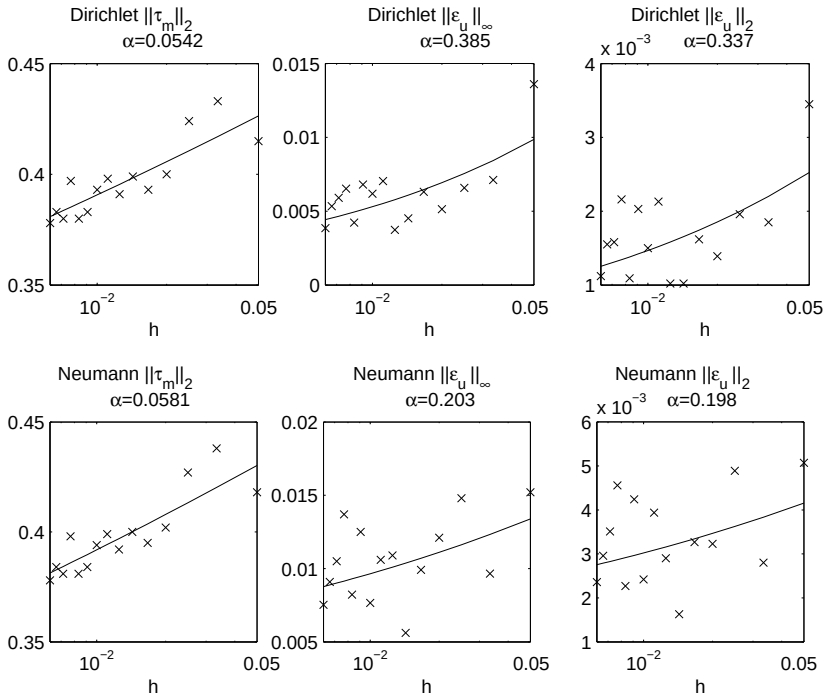

Figure 4.15: Example 1, piecewise constant reconstruction, irregular grid.

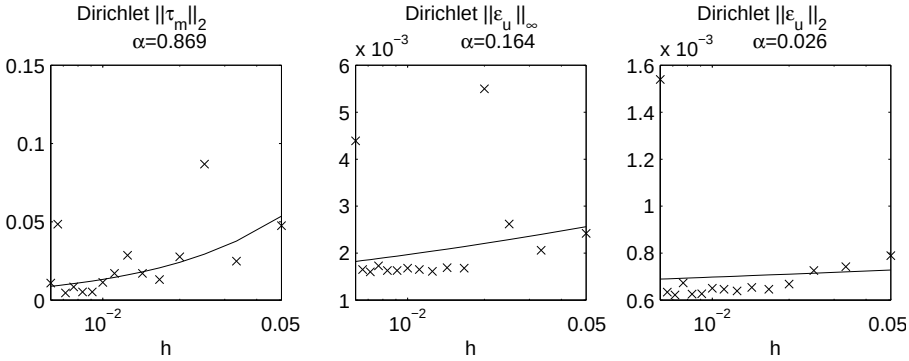

Figure 4.16: Example 1, Dirichlet boundary, linear reconstruction, irregular grid.

the HLS method where threshold is 0.4 are given in Fig. 4.17. The method with weights gives very similar results.

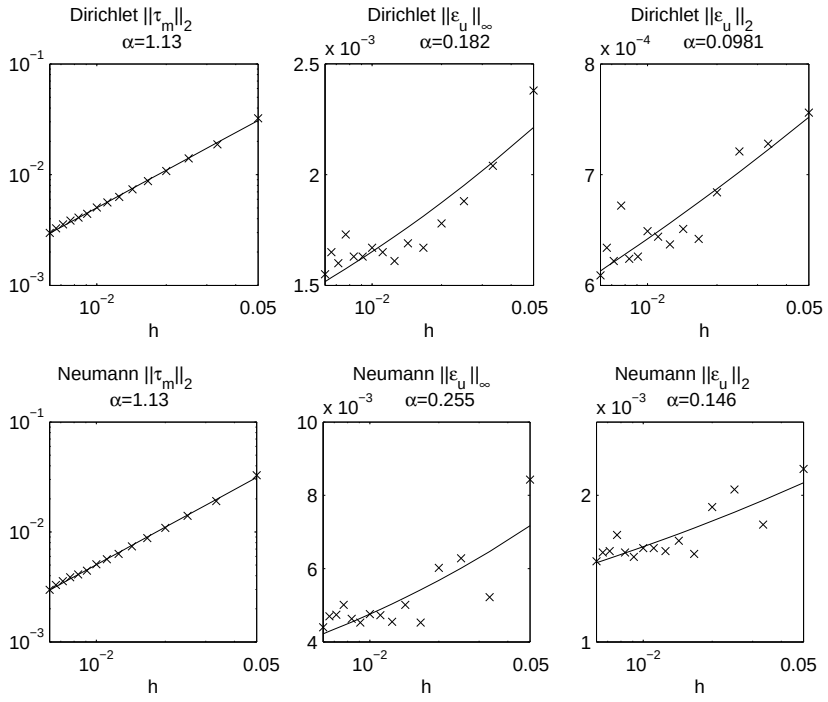

Figure 4.17: Example 1, linear reconstruction, HLS, threshold 0.4, irregular grid.

The sharp jump in the maximum error which is visible in Fig. 4.16 does not occur in Fig. 4.17. If we change gradually the threshold in the weighted case from 10^{-10} to $2 \cdot 10^{-6}$, the error does not change much compared to Fig. 4.16 until we cross $1.16 \cdot 10^{-6}$. Then it drops to the value shown in Fig. 4.17. The cause is one reconstruction stencil at the boundary that looks like the one shown in Fig. 4.18 on the left. This is the reconstruction stencil for the face in the middle of it. Two triangles are missing due to the presence of the boundary. When weights are used, the reconstruction matrix has singular values 1.81, 1.49, 0.55, 0.67, $1.38 \cdot 10^{-5}$ and $1.16 \cdot 10^{-6}$. This stencil is close to the one shown in Fig. 4.18 on the right, which leads to a singular reconstruction matrix, as we showed in Sect. 4.6. This is why we have one singular value that is significantly smaller than the weight 10^{-5}, and it could have been even smaller if the stencil was closer to the one shown in Fig. 4.18 on the right. However, if we add a triangle like in Fig. 4.19, then the first four singular values do not change, but the last two become $1.76 \cdot 10^{-5}$ and $7.16 \cdot 10^{-6}$, and the final result is much better, as shown in Fig. 4.17.

Figs. 4.20 and 4.21 give the results of this method on the Cartesian and the regular grids. On the Cartesian grids this method is second order accurate, for both Dirichlet and outflow problems.

However, this method hardly converges on the irregular grids (cf. Fig. 4.17).

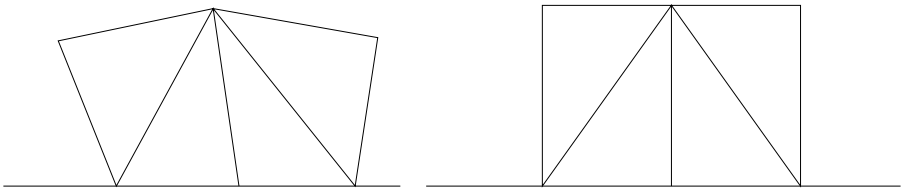

Figure 4.18: Problematic stencil.

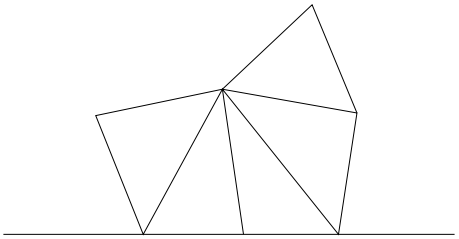

Figure 4.19: Adapted stencil.

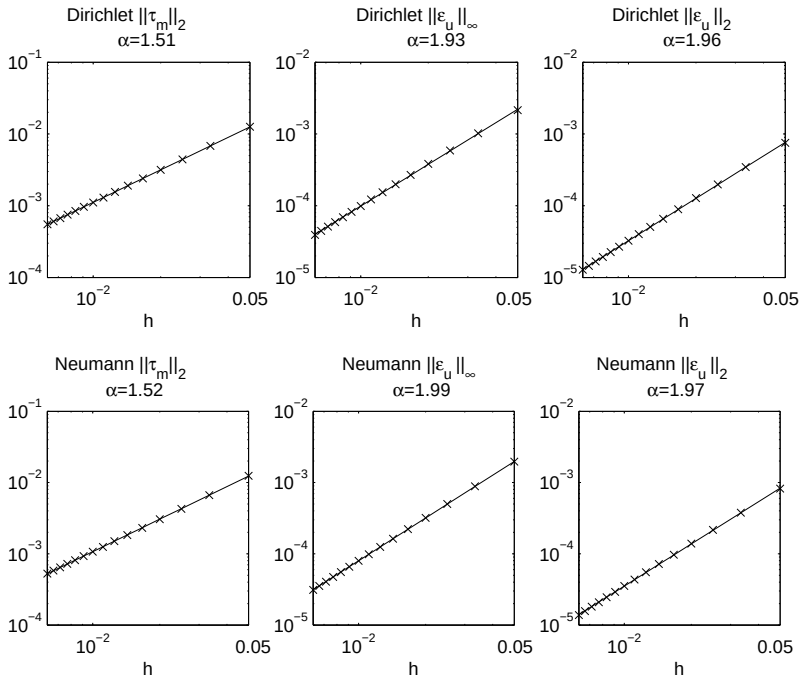

Figure 4.20: Example 1, linear reconstruction, hierarchical method, threshold 0.4, Cartesian grid.

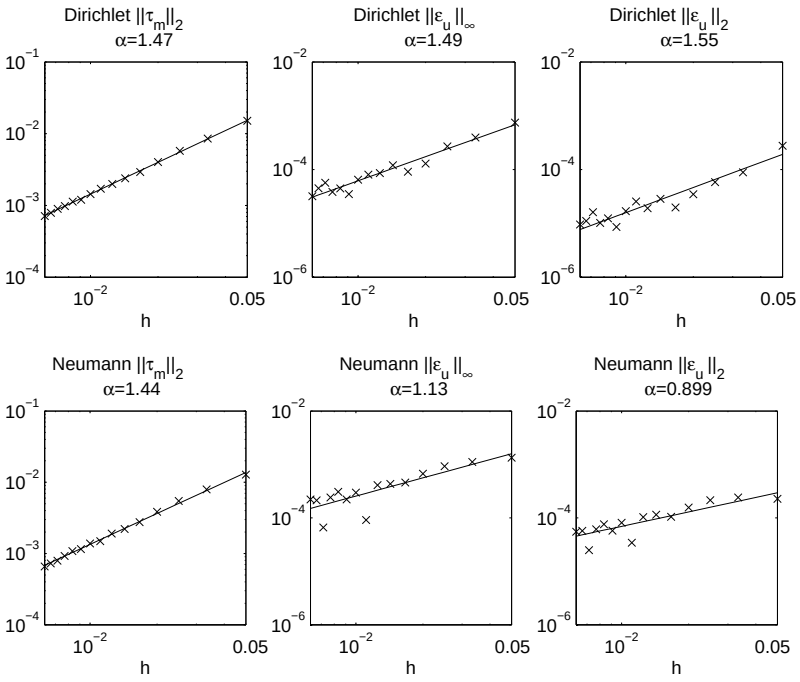

Figure 4.21: Example 1, linear reconstruction, hierarchical method, threshold 0.4, regular grid.

We therefore proceed with another method.

Divergence-free vertex-based linear reconstruction

For the stress in face e (see equation (4.70) and Fig. 4.9) we take the average of the stress in vertices 1 and 2, calculated by using the divergence-free vertex-based reconstruction. The divergence for both vertices is computed from the four faces that surround the control volume face under consideration using formula (4.15). This method leads to a relatively small stencil (see Fig. 4.22).

The results obtained with this scheme are presented in Figs. 4.23, 4.24 and 4.25. Although there are still some problems with the finest irregular grids, the convergence improved considerably.

Sometimes the error increases as we refine the mesh, for example in the case of the unstructured 50×50 grid. The reason is found to be that the mesh contains two neighboring vertices with only four faces meeting in each of them (see Fig. 4.26). Reconstruction stencils for both vertices have to be enlarged. Still, the drop of accuracy is local and limited.

Fig. 4.27 shows the error for the 40×40 unstructured grid. We see that it is not quite chaotic. This error seems to be mainly rotational. This is because the discretization of the divergence is exact.

The question how the prescribed stress should be used in the case of Neumann boundary conditions is very sensitive. Improper discretization can easily lead to an ill-conditioned system or a large drop of accuracy. In Sect. 4.11.4 we

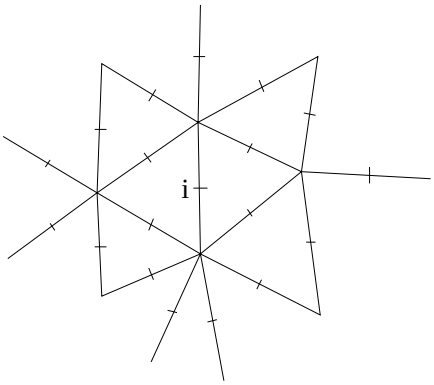

Figure 4.22: Computational stencil for viscous term with the vertex-based divergence-free linear reconstruction.

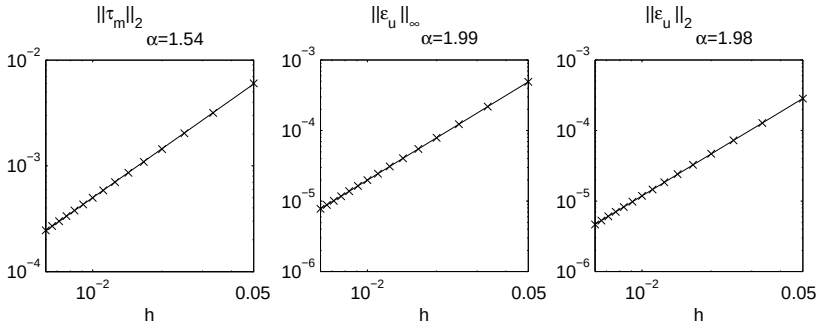

Figure 4.23: Example 1, linear divergence-free reconstruction, Cartesian grid, Dirichlet.

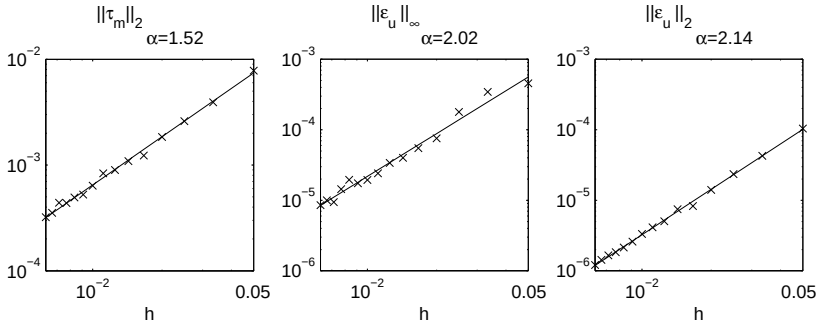

Figure 4.24: Example 1, linear divergence-free reconstruction, regular grid, Dirichlet.

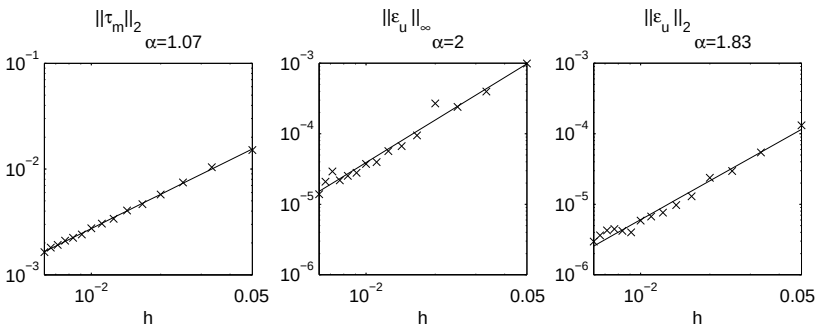

Figure 4.25: Example 1, linear divergence-free reconstruction, irregular grid, Dirichlet.

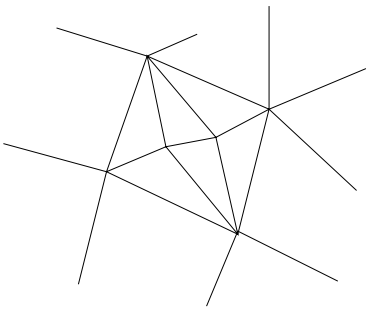

Figure 4.26: A detail of the irregular mesh.

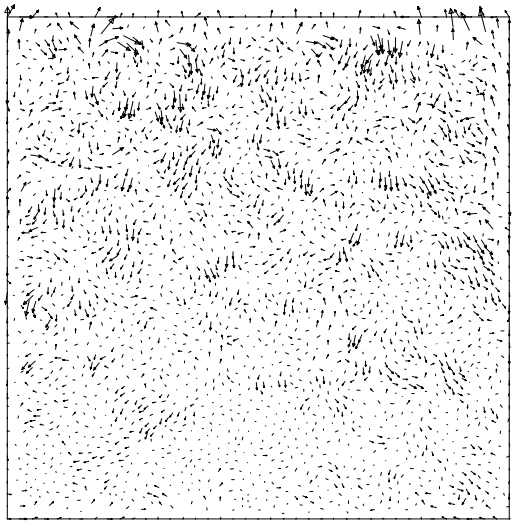

Figure 4.27: The error obtained with the divergence-free linear reconstruction on the irregular mesh.

explain how this should be done.

An open question is: why does the divergence-free reconstruction give so much better results than the non-divergence-free reconstruction? We do not have an answer to this question, but only a vague idea of what may be the reason. In Sect. 4.5 we noticed that the linear reconstruction of scalars does not lead to a working scheme if we try to reconstruct the constant part of the reconstruction polynomial, instead of making the polynomial exact in the upwind cell. This suggests that it is a good idea to make the reconstruction polynomial exact for some of the data used in the reconstruction, instead of being approximate for all the data. In the scalar case the reconstruction polynomial is exact for the value of the scalar in the upwind cell, and in the case of the divergence-free linear reconstruction of a vector field it is exact for the divergence. It is possible to determine a reconstruction polynomial of the staggered vector field that is exact for all three faces of the upwind cell, and not only for the divergence determined from this cell, and such reconstruction might give better results then the presented divergence-free reconstruction. However, it would require six times more memory, because we would have to save two polynomials per face, instead of one polynomial per vertex.

Quadratic reconstruction

If the reconstruction polynomials are face-centered, the stencil becomes huge with quadratic reconstruction. A remedy is not to use finite volume approximation for the viscous term, except for the divergence part. If we suppose that μ is constant over the integration domain, we get

$$\frac{1}{|\Omega_i|}\mathbf{N}_i \cdot \int_{\Omega_i} \nabla \cdot \left[\mu\left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\right)\right] - \frac{2}{3}\nabla\left(\mu\Delta\right) d\Omega_i = \tag{4.77}$$

$$= \frac{1}{|\Omega_i|}\mathbf{N}_i \cdot \int_{\Omega_i} \mu \begin{bmatrix} 2\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x \partial y} \\ 2\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 u}{\partial x \partial y} \end{bmatrix} d\Omega_i - \frac{2}{3}\frac{1}{|\Omega_i|}\mathbf{N}_i \cdot \int_{\Omega_i} \mu \nabla\Delta d\Omega_i,$$

where $\Delta \equiv \nabla \cdot \mathbf{u}$. The second integral is computed as before. For the first one we use quadratic approximation (4.43) of the velocity in a cell with the stencil shown in Fig. 4.7. Integration domain Ω_i is composed of two cells, Ω_1 and Ω_2. In each triangle we have a cell-based reconstruction polynomial, and we approximate the partial derivatives in (4.77) by the coefficients of this polynomial:

$$\begin{array}{cccc} \dfrac{\partial^2 P_x}{\partial x^2} = 2d_1, & & \dfrac{\partial^2 P_x}{\partial x \partial y} = e_1, & \\ \dfrac{\partial^2 P_x}{\partial y^2} = 2f_1, & \dfrac{\partial^2 P_y}{\partial x^2} = 2d_2, & \dfrac{\partial^2 P_y}{\partial x \partial y} = e_2, & \dfrac{\partial^2 P_y}{\partial y^2} = 2f_2 \end{array} \tag{4.78}$$

(see 4.43). This leads to

$$\mathbf{N}_i \cdot \int_{\Omega_1} \mu \begin{bmatrix} 4d_1 + e_2 + 2f_1 \\ 2d_2 + e_1 + 4f_2 \end{bmatrix} d\Omega = \mu\mathbf{N}_i \cdot \begin{bmatrix} 4d_1 + e_2 + 2f_1 \\ 2d_2 + e_1 + 4f_2 \end{bmatrix} |\Omega_1|, \tag{4.79}$$

and the same expression is used to evaluate the integral over Ω_2.

It does not prove to be advantageous to include a given boundary stress in the reconstruction. Instead, we can use the finite volume approach only at the

outflow boundary, and allow our stencils to be larger here by using the quadratic reconstruction based in faces, or calculate the gradient at the control volume faces from the quadratic reconstruction based in the control volume cells. The first of these two approaches leads to a larger stencil at the outflow, but it seems to be a bit more accurate. Actually we could apply the second approach also in the interior of the domain, but this leads to an inaccurate scheme and has no advantages.

We also tried to use cubic reconstruction in the vicinity of a boundary. This does decrease the local truncation error a bit, but it does not improve the final result.

We tried this scheme on Example 1 with Dirichlet boundary conditions and the irregular grid. We used the HLS method with three groups: one for the faces of the central triangle, the other for the faces of the surrounding three triangles, and the third one for all other faces (cf. Fig. 4.7). When the quadratic reconstruction is used, the reconstruction stencil is much larger than in the linear case. Degenerate reconstruction stencils are not an exception as in the linear case, but a rule, especially at the boundary. Without a large threshold, the method does not converge always, and even if it does, the error is very large. It has been noticed that the singular values of the first group in interior stencils are smaller than the ones in the second group, and the ones in the third group are the largest. On the other hand, there can be no parallel faces in a single triangle, so we use 10^{-10} as a threshold for the first group. Experiments show that a threshold of 0.6 in the second group and of 1.5 in the third group give relatively good results, shown in Fig. 4.28 for the irregular grid, and in Fig. 4.29 for the regular grid.

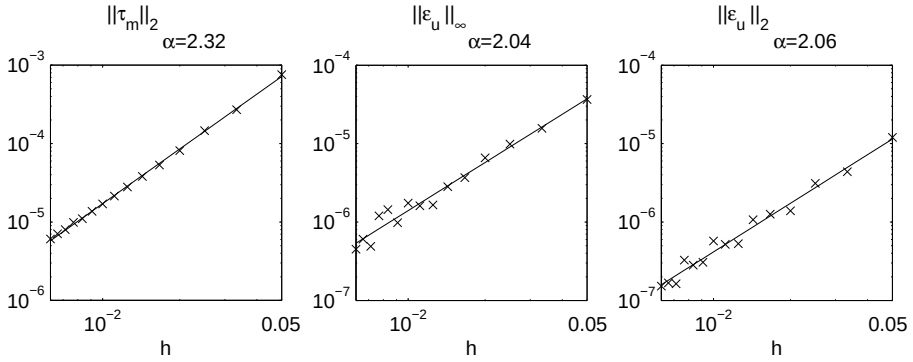

Figure 4.28: Example 1, quadratic reconstruction, irregular grid, thresholds 10^{-10}, 0.6 and 1.5, Dirichlet.

We did not find any weights that would make the results obtained by the weighted SVD comparable to these. It appears that the faces from the second group should have more influence on the reconstruction than those from the third group, which means that they should have higher weights if the weighted method is used. But stencils can also degenerate in the second group, and these bad singular values will still be larger than the ones originating from the third group, where the weight is smaller, so they will be invisible for the threshold. Therefore the HLS method is necessary.

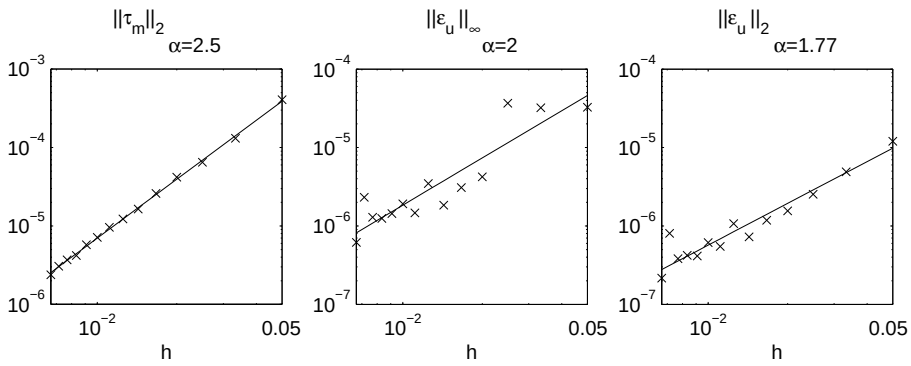

Figure 4.29: Example 1, quadratic reconstruction, regular grid, Dirichlet.

This method is more accurate on the irregular grids. On the 160×160 regular grid it did not converge at all. The reason might be that the regular grid contains many parallel faces, so the rank of the reconstruction matrices might be lost and the stencils must be enlarged too much, although we did not find an example of this. It is possible that this could be corrected by choosing thresholds in a different way, or by choosing a different division of faces into groups.

If the grid is Cartesian, this scheme leads to an ill-conditioned linear system which sometimes cannot be solved by iterative solvers. Even if it can be solved or if we use a direct solver, the accuracy is poor. It has been observed that the divergence of the obtained result is large, whereas the exact solution is divergence-free. Still, the local truncation error is small.

div-curl formulation

The foregoing observation gave us the idea to treat the divergence part of the viscous term in a different way. Some authors, see for example [14], use the following relation, valid only in the incompressible flow case:

$$\nabla \cdot (\nabla \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \operatorname{curl}(\operatorname{curl} \mathbf{u}). \tag{4.80}$$

Also the following equality holds:

$$\nabla \cdot (\nabla \mathbf{u})^T = \nabla(\nabla \cdot \mathbf{u}). \tag{4.81}$$

By combining these two relations we obtain a new expression for $\nabla \cdot \tau$:

$$2\mu \left(\nabla \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) - \frac{2}{3} \nabla (\Delta) \right) = 2\mu \left(\frac{4}{3} \nabla (\Delta) - \operatorname{curl}(\operatorname{curl} \mathbf{u}) \right). \tag{4.82}$$

These relations also hold in the three-dimensional case. We shall discretize the divergence by using formula (4.15). In equation (4.82) we do not substitute $\nabla \cdot \mathbf{u} = 0$, because this is found to make the discretization matrix less well-conditioned. The rotational part is computed from the quadratic cell-based divergence-free reconstruction polynomials. The divergence of the reconstruction polynomials is taken to be zero.

The results are given in Figs. 4.30–4.32. We see that the method is still a bit less accurate on Cartesian grids than on unstructured grids. However, the accuracy improved considerably.

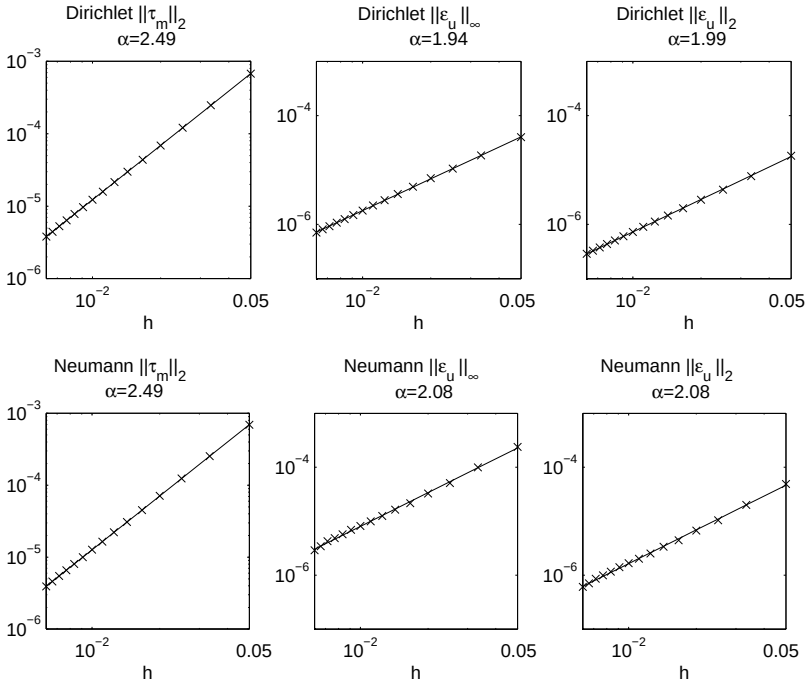

Figure 4.30: Example 1, quadratic divergence-free reconstruction, Cartesian grid.

It follows from (4.80) and (4.81) that

$$\operatorname{curl}(\operatorname{curl}\mathbf{u}) = \nabla\cdot\left((\nabla\mathbf{u})^T - \nabla\mathbf{u}\right) \tag{4.83}$$

This makes it possible to apply the divergence theorem to (4.77) and to separate the divergence part from the rotational part. By doing so we obtain

$$\begin{aligned}\mathbf{N}_i\cdot\int_{\Omega_i}\nabla\cdot\left[\mu\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)\right]-\frac{2}{3}\nabla\left(\mu\Delta\right)d\Omega_i = \\ \mathbf{N}_i\cdot\oint_{\partial\Omega_i}\mu\left(\nabla\mathbf{u}-(\nabla\mathbf{u})^T\right)\cdot\mathbf{n}d\Gamma+\frac{4}{3}\mathbf{N}_i\cdot\oint_{\partial\Omega_i}\mu\Delta\mathbf{n}d\Gamma\end{aligned} \tag{4.84}$$

We can further approximate these integrals along the control volume faces and compute $\nabla\mathbf{u}-(\nabla\mathbf{u})^T$ from a reconstruction polynomial based in one of the central cells. This gives almost the same result as if the finite volume approach was not used. Note that if the divergence part is not separated from the rotational part, such finite volume approach leads to a bad matrix.

At the outflow we cannot use such approach. Although in the incompressible flow case $\nabla\cdot\left(\nabla\mathbf{u}+(\nabla\mathbf{u})^T\right)=\nabla\cdot\left(\nabla\mathbf{u}-(\nabla\mathbf{u})^T\right)$, this does not mean that $\nabla\mathbf{u}+(\nabla\mathbf{u})^T=\nabla\mathbf{u}-(\nabla\mathbf{u})^T$, so we cannot use the given stress at the outflow. Instead we would need a special boundary condition. For this reason we do

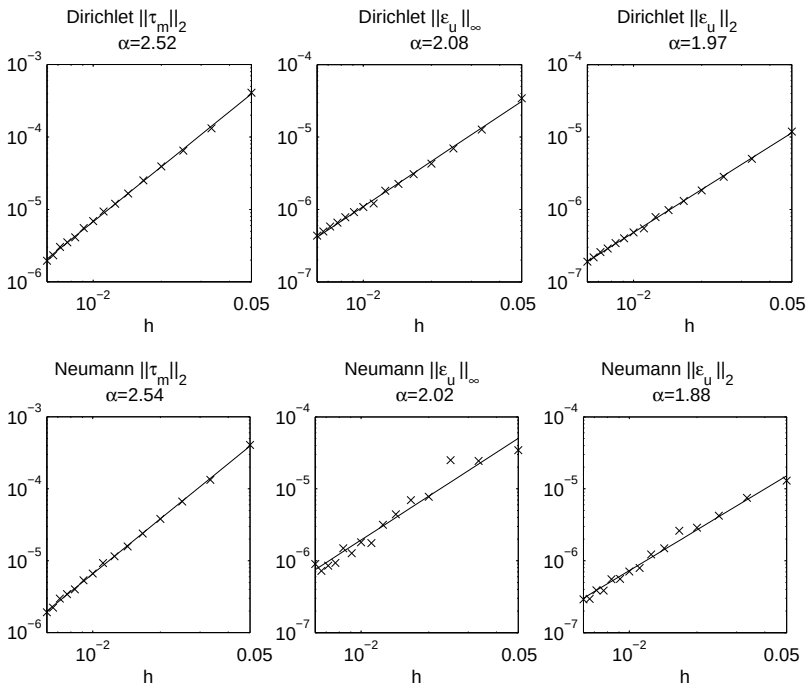

Figure 4.31: Example 1, quadratic divergence-free reconstruction, regular grid.

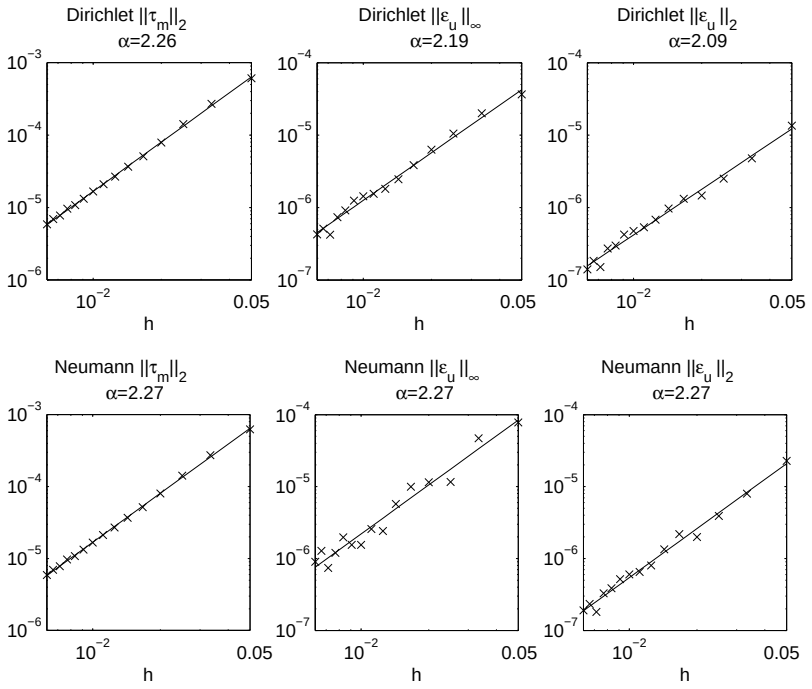

Figure 4.32: Example 1, quadratic divergence-free reconstruction, irregular grid.

not separate the divergence at the outflow. At the faces belonging to outflow control volumes where the stress is not given, we compute $\nabla \cdot \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right)$, add $\frac{4}{3}$ of the divergence to the left hand side to improve the matrix properties, and subtract two times the divergence from the right hand side.

The error is a couple of times larger on the Cartesian grids. However, the method is still second order accurate. We also see that on the irregular grid the oscillations of the error are larger for Neumann boundary conditions then in the case of Dirichlet boundary conditions. Still, this drop of accuracy is local and it occurs at the outflow, which is usually not so important.

Relation (4.82) allows us to make another, much less demanding scheme, that is also second order accurate, based on the divergence-free linear reconstruction. Unfortunately, relation (4.82) can be used only if the viscosity is constant, although a similar relation could be derived assuming piecewise linear viscosity variation. Furthermore, in the compressible flow case we cannot split the divergence from the curl in the viscous term.

Conclusion on viscous term

We have seen that the quadratic reconstruction is connected with numerous difficulties. It is not appropriate for a Cartesian grid, it requires a large stencil, it is difficult to implement the Neumann conditions, it requires HLS to work, and it gives usefull results only in the incompressible flow case. On the other hand, the convergence stalls in the case of the linear non-divergence-free reconstruction. The best results were obtained with the vertex-based divergence-free linear polynomials. Therefore this is the method of choice. It was thoroughly tested and the results are presented in Chapter 5.

4.11.2 Inertia term

The divergence theorem gives

$$\frac{1}{|\Omega_i|}\int_{\Omega_i} \nabla \cdot [\mathbf{u}^n(\mathbf{m}^{n+1}\cdot\mathbf{N}_i)]d\Omega = \frac{1}{|\Omega_i|}\oint_{\partial\Omega_i}(\mathbf{u}^n\cdot\mathbf{n})(\mathbf{m}^{n+1}\cdot\mathbf{N}_i)d\Gamma \approx$$
$$\approx \frac{1}{|\Omega_i|}\sum_e u_e^n(\mathbf{m}_e^{n+1}\cdot\mathbf{N}_i)\bar{l}_e, \qquad (4.85)$$

where $u_e = m_e/\rho_e$ is the average normal velocity in face e. We shall omit the superscripts for brevity.

Scheme [36, 37] uses the piecewise constant reconstruction formula (4.23) to reconstruct $\mathbf{m}_e \cdot \mathbf{N}_i$ in the upwind cell, using the two faces opposite to face e. This discretization is very robust, and the computed velocity is first order accurate, as demonstrated in Sect. 5.2.1, 5.2.2, and 5.2.3, unless the result is spoiled by the artificial viscosity or an improperly placed outflow boundary.

We shall use the linear vertex-based divergence-free polynomials to improve the spatial accuracy of the discretization of the inertia term. The main reason for this choice is that these polynomials exhibited superior performance for viscosity comparing to all other options. It is desirable to use the same polynomials for both inertia and viscous terms in order to save storage space.

Term $\mathbf{m}_e \cdot \mathbf{N}_i$ is represented as a combination of the normal and the tangential momentum in face e:

$$\mathbf{m}_e \cdot \mathbf{N}_i = \mathbf{N}_i \cdot ((\mathbf{N}_e \cdot \mathbf{m}_e)\mathbf{N}_e + (\mathbf{T}_e \cdot \mathbf{m}_e)\mathbf{T}_e) = (\mathbf{N}_i \cdot \mathbf{N}_e) m_e + (\mathbf{N}_i \cdot \mathbf{T}_e) m_e^t. \quad (4.86)$$

We make the following approximation:

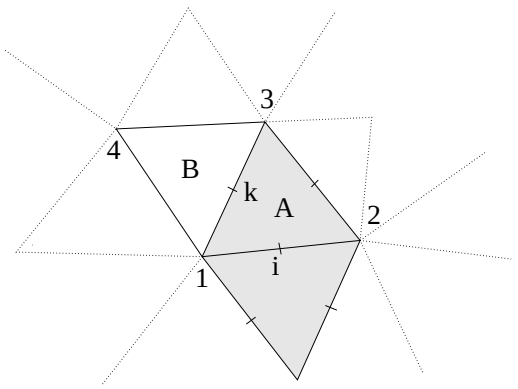

Figure 4.33: Vertex-based reconstruction.

$$m_e^t \approx \mathbf{T}_e \cdot \mathbf{P}_{\text{up}}(\mathbf{r}_e), \quad (4.87)$$

where $\mathbf{P}_{\text{up}}$ is the reconstruction polynomial based in the upwind vertex, and $\mathbf{r}_e$ is the position vector of the center of face e relative to the upwind vertex.

It is also possible to reconstruct the full vector $\mathbf{m}_e$ instead of m_e^t, but the scheme obtained is somewhat less accurate.

The choice of the upwind vertex depends on the direction of the flow, so that an upwind-biased scheme is obtained. The upwind bias is introduced to improve the condition of the matrix and to diminish numerical wiggles. One possibility is to choose the upwind cell A or B depending on the direction of the normal velocity in face k, to compute the divergence in this cell (it is needed for the linear reconstruction because we are using divergence-free polynomials), and to choose one of the vertices of this cell for the center of the polynomial. For instance, we can choose the vertex such that the sum of the inflow fluxes trough the two triangle faces that meet in that vertex is maximal (if the fluid flows out, the inflow flux is negative).

However, better results were obtained when one of the vertices opposite to face k is chosen (i.e. vertex 2 or vertex 4). This is the method of choice. If vertex 2 is used, the divergence is calculated from triangle A, otherwise it is calculated from triangle B. But this vertex can lie at an outflow or a free slip boundary, and then we choose another vertex that is not at the outflow or at the free slip boundary. It has to be one of the vertices of the upwind cell, otherwise the scheme may break up.

Reconstruction stencil for the vertex 4 (see Fig. 4.33) does not belong to the computational stencil of the viscous term shown in Fig. 4.12. This implies that if we want to compute $\mathbf{m}_k \cdot \mathbf{N}_i$ implicitly, we have to enlarge the computational stencil, or to choose some other vertex for reconstruction. We tried to choose

between vertices 1 and 3 depending on the reconstructed tangential velocity in face k, unless this face is at the outflow boundary, in which case we use vertex 2. Resulting scheme did not converge in time. We can also compute $\mathbf{m}_k \cdot \mathbf{N}_i$ explicitly, but this limits the time step that can be used. We also tried to compute this term implicitly while using the normal velocity from the old time level whenever some face does not belong to the computational stencil implied by the viscosity. This also works well and gives the same steady solution, but the time step is still limited as if the explicit scheme was used.

Therefore we choose to enlarge the computational stencil.

For accuracy tests of this reconstruction see Sect. 5.2.1, 5.2.2, and 5.2.3.

4.11.3 Pressure term

Several approximations of the pressure gradient were presented in [36]. Our experience is that the path integral method with six cells gave the best results. This method first appeared in [31] and will be presented here for completeness. It is based on the identity

$$\int_a^b \nabla p \cdot d\mathbf{x} = p_b - p_a, \tag{4.88}$$

where a and b are two points. Therefore the following approximation makes sense:

$$(\nabla p)_{ab} \cdot (\mathbf{x}_b - \mathbf{x}_a) \approx p_b - p_a, \tag{4.89}$$

where $(\nabla p)_{ab}$ is the pressure gradient in some point near the center of the line connecting a and b, and $\mathbf{x}_a$ and $\mathbf{x}_b$ are the position vectors of points a and b, respectively. We apply this formula to three pairs of cells (see Fig. 4.34):

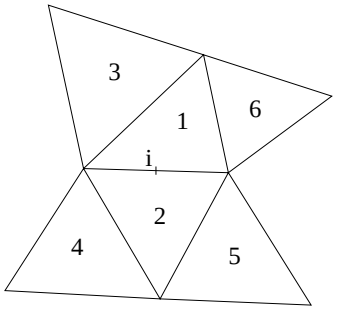

Figure 4.34: Path integral method with six cells.

$$(\nabla p)_i \cdot (\mathbf{x}_2 - \mathbf{x}_1) \approx p_2 - p_1, \tag{4.90}$$

$$(\nabla p)_i \cdot (\mathbf{x}_3 - \mathbf{x}_5) \approx p_3 - p_5, \tag{4.91}$$

$$(\nabla p)_i \cdot (\mathbf{x}_4 - \mathbf{x}_6) \approx p_4 - p_6. \tag{4.92}$$

Here $p_1, \ldots, p_6$ are the pressures averaged over triangles $1, \ldots, 6$ which we use as the primary pressure variables. Using these averages instead of the pointwise pressure in circumcenters is what makes it possible to use non-Delaunay grids.

Points $1, \ldots, 6$ are the centroids of the triangles. By adding (4.91) and (4.92) we obtain

$$(\nabla p)_i \cdot (\mathbf{x}_3 - \mathbf{x}_5 + \mathbf{x}_4 - \mathbf{x}_6) \approx p_3 - p_5 + p_4 - p_6. \tag{4.93}$$

The pressure gradient $(\nabla p)_i$ is determined from (4.90) and (4.93).

We also tried to compute the pressure gradient by fitting a linear or a quadratic polynomial to the pressure in the stencil shown in Fig. 4.34 by using least squares method. Linear polynomial gave almost the same result as path integral method with six cells, and when the quadratic polynomial was used, the method was unstable in time.

4.11.4 Treatment of boundaries

In the subsonic flow case in each boundary face we have two boundary conditions for momentum. We may prescribe the momentum vector (for example at the inflow boundary or in the case of the no-slip condition), the normal and tangential stress (at an outflow boundary), or the normal component of one of these two and the tangential component of the other (as on free slip boundaries or at an outflow boundary).

If the normal stress is prescribed, it is normally assumed that the deviatoric stress tensor is zero, because the outflow should be placed far enough so that the deviatoric stress is indeed zero, and the prescribed stress is used as a prescribed pressure. However, in situations where this is not possible, we prescribe the deviatoric stress and the pressure separately in order to preserve the accuracy. This is mostly useful for testing purposes, because we would not normally know the deviatoric stress at a non-trivial outflow boundary.

The deviatoric stress is only used to discretize the viscous term. Therefore it is ignored in the case of the Euler equations.

In the supersonic flow case the momentum is prescribed at the boundaries in the same way as in the subsonic flow case, except that we do not consider compressible viscous flow, so we do not prescribe the deviatoric stress. The difference with the subsonic flow case is that the pressure is given at inflow boundaries, and not at outflow boundaries.

There is no need to integrate for faces where the normal momentum is given. For boundary faces where the normal momentum is not given a control volume consisting only of one triangle is used.

Inertia

If the tangential momentum m_e^t is given in some of the faces of the control volume, it is used and there is no need to reconstruct for that face. Otherwise the first order scheme uses the one-sided reconstruction of momentum in the boundary cell. In the case of the superlinear scheme we take the vertex belonging to the same boundary triangle as face e which is opposite to this face for the upwind vertex.

If the upwind vertex falls on a boundary where the full momentum was not given, then some other vertex belonging to the upwind cell is chosen for the upwind vertex, except if they all lie on such boundaries. This improves the accuracy a bit.

Viscosity

The term $N_{\alpha,i}\tau_{\alpha\beta,e}N_{\beta,e} = \mathbf{N}_i \cdot \tau_e \cdot \mathbf{N}_e$ needed in (4.70) is represented as

$$\mathbf{N}_i \cdot \tau_e \cdot \mathbf{N}_e = (\mathbf{N}_i \cdot \mathbf{N}_e)\tau_{nn,e} + (\mathbf{N}_i \cdot \mathbf{T}_e)\tau_{nt,e}. \tag{4.94}$$

Here

$$\tau_{nn,e} = \mathbf{N}_e \cdot \tau \cdot \mathbf{N}_e, \qquad \tau_{nt,e} = \mathbf{N}_e \cdot \tau \cdot \mathbf{T}_e \tag{4.95}$$

denote the normal and the tangential stress component, respectively.

If one of the stress components is given in face e, it is used and only the other component is reconstructed. If τ_{nn} and τ_{nt} are given, there is no need to reconstruct.

In the scheme that uses the vertex-based divergence-free reconstruction, the situation where some stress component is given in vertex 1 but is not given in face e (see Fig. 4.35) requires special care. The general rule is that one should never reconstruct the stress which is given, otherwise the scheme becomes unstable.

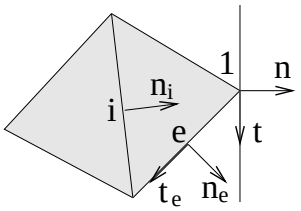

Figure 4.35: Vertex at a boundary.

The deviatoric stress tensor τ is not fully determined by prescribing its normal and tangential component, because it has three degrees of freedom (see Sect. 2). However, we have

$$\tau_{11} + \tau_{22} = \tfrac{2}{3}\mu\Delta, \qquad \tau_{12} = \tau_{21} \tag{4.96}$$

and also

$$\tau_{nn} + \tau_{tt} = \tfrac{2}{3}\mu\Delta, \qquad \tau_{nt} = \tau_{tn}, \tag{4.97}$$

where $\tau_{nn} = \mathbf{n} \cdot \tau \cdot \mathbf{n}$, $\tau_{nt} = \mathbf{n} \cdot \tau \cdot \mathbf{t}$, $\tau_{tn} = \mathbf{t} \cdot \tau \cdot \mathbf{n}$, $\tau_{tt} = \mathbf{t} \cdot \tau \cdot \mathbf{t}$, and $\mathbf{n}$ and $\mathbf{t}$ are the normal and the tangential unit vector at the boundary. Furthermore,

$$\begin{bmatrix} \tau_{nn} & \tau_{nt} \\ \tau_{tn} & \tau_{tt} \end{bmatrix} = R\tau R^{-1}, \qquad R = \begin{bmatrix} n_x & n_y \\ -n_y & n_x \end{bmatrix}, \tag{4.98}$$

and so

$$\tau = R^{-1} \begin{bmatrix} \tau_{nn} & \tau_{nt} \\ \tau_{tn} & \tau_{tt} \end{bmatrix} R = R^{-1} \begin{bmatrix} \tau_{nn} & \tau_{nt} \\ \tau_{nt} & \frac{2}{3}\mu\Delta - \tau_{nn} \end{bmatrix} R. \tag{4.99}$$

If τ_{nn} and τ_{nt} are given in vertex 1, we obtain the divergence from the divergence theorem and use (4.99) to obtain the stress. Then we use it to compute $\mathbf{N}_i \cdot \tau_1 \cdot \mathbf{N}_e$. Hence there is no need for reconstruction at the outflow boundaries.

Of course, in the incompressible flow case one can take $\Delta = 0$. We compute the divergence implicitly.

If one stress component is given in vertex 1, the other is reconstructed.

Pressure

When the pressure is prescribed at a boundary, the derivative of the pressure in direction N_i (see Fig. 4.36) is computed in the following way:

$$(\nabla p \cdot \mathbf{N})_i = \frac{p_{i'} - p_1}{\mathbf{x}_{i'} - \mathbf{x}_1}, \tag{4.100}$$

where point i' is the normal projection of point 1 to the boundary.

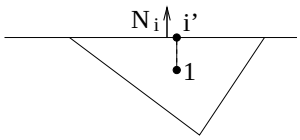

Figure 4.36: Computing the pressure gradient at the outflow.

4.12 Pressure-correction equation in incompressible flow

The discretization of equation (3.8) directly follows from the discretization of the divergence presented in Sect. 4.3 and the discretization of the pressure gradient presented in 4.11.3.

4.13 Mach-uniform pressure-correction equation

The Mach-uniform pressure-correction equation (3.17) is discretized in [36] in a similar way as the continuity equation. It is integrated over the shaded control volume shown in Fig. 4.37.

The momentum vector in the time derivative is reconstructed from the three faces of the control volume using least squares approximation. The gradient of the pressure correction is approximated by a least squares fit to the normal pressure gradients in the faces of the control volume, which are computed by the path integral method (see Sect. 4.11.3). If the normal momentum component is prescribed in one of the control volume faces, then it follows from (3.7) that in this face $\nabla \delta p = 0$ because $m^* = m^{n+1}$.

The divergence theorem is applied to the convection and the second order term (second and third row in the formula (3.17)). In [36], convected quantities in the convection term are computed in an upwind manner: density, pressure and the pressure-correction in face e are evaluated in cell 1 if the flow is outward, or in cell 2 otherwise. The momentum vector is reconstructed from faces i and j or from faces k and l, depending on the direction of the flow. Central approximations of momentum, pressure and density in faces are used in the second order term. For details see [36].

In order to obtain higher order accuracy, we modified this scheme by introducing linear reconstruction. Momentum in the time derivative is computed as the average of momenta reconstructed in the vertices of the control volume

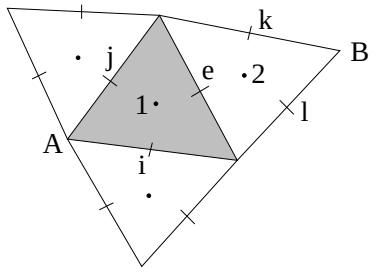

Figure 4.37: Control volume used for the scalar equations.

by using the divergence-free linear reconstruction polynomials based in three vertices of the control volume. The divergence is reconstructed by applying the divergence formula to the control volume. Momentum in the convective term in face e (see Fig. 4.37) is obtained from the reconstruction polynomial based in vertex A if the fluid flows into the control volume and the divergence is computed from cell 1, otherwise vertex B and cell 2 are used. Pressure, pressure-correction and density are reconstructed by using upwind-biased cell-based or vertex-based linear reconstruction, which was presented in Sect. 4.5. The second order term vanishes for the steady solution, and since our scheme is first order accurate in time, there is no need to use linear reconstruction for this term.

4.14 Energy equation

When the fully compressible sequential update procedure is used, the energy equation (3.11) needs to be discretized. The convection term is evaluated in the same way as in the case of the density equation, using ρH instead of ρ. The momentum vector in the kinetic energy term in the time derivative is computed in the same way as in the case of the Mach-uniform pressure-correction equation.

4.15 Equation of state

The equation of state is not used in time stepping when the Mach-uniform pressure-correction method is used. However, it is used in the form (3.12) in the case of the fully compressible sequential update procedure. As in the case of the energy equation and the Mach-uniform pressure-correction equation, the momentum vector in a cell is computed in [36] by least squares from the three faces of this cell, and we compute it as the average of momenta computed in the vertices of this cell using the vertex-based divergence-free linear reconstruction.

4.16 Monotonicity considerations

In the vicinity of steep gradients or discontinuities, spurious oscillations may occur. Not only are these oscillations non-physical, but if they become too large, the density may get close to zero. The convecting velocity is computed

by dividing the momentum by the density, and therefore it may become very large, so the solution procedure breaks up.

These spurious wiggles are a well-known phenomenon, which was investigated by Godunov in [12]. His order barrier theorem shows that linear non-oscillatory schemes are at most first-order accurate. For this reason every higher-order non-oscillatory scheme must be nonlinear.

Several concepts exist that ensure that a scheme is non-oscillatory. One of them is *monotonicity preservation*, a requirement that new extrema cannot appear in the solution. Another one is that the scheme is *total variation diminishing* (TVD), which means that the total variation of the solution cannot grow in time. One which is easier to enforce is that a scheme is *local extremum diminishing* (LED), which means that local maxima cannot increase and local minima cannot decrease. All these requirements indeed hold for the exact solution (for scalar problems only).

Numerous authors have developed methods to enforce these requirements. For a survey, see Sect. 9.4 of [38]. One possible way to ensure that a scheme is LED is flux limiting. The approximation (4.20) is replaced by

$$\psi_e = \begin{cases} \psi_1 + \phi_1 \nabla\psi_{1,e} \cdot (\mathbf{r}_e - \mathbf{r}_1) & \text{if} \quad u_e^n \overline{l}_e \geq 0; \\ \psi_2 + \phi_2 \nabla\psi_{2,e} \cdot (\mathbf{r}_e - \mathbf{r}_2) & \text{if} \quad u_e^n \overline{l}_e < 0, \end{cases} \tag{4.101}$$

where ϕ is so-called *flux limiter*, determined such that

$$\min_j(\psi_j, \psi_i) \leq \psi_i + \phi_i \nabla\psi_{i,e} \cdot (\mathbf{r}_e - \mathbf{r}_i) \leq \max_j(\psi_j, \psi_i), \qquad \forall e, \tag{4.102}$$

where the minimum and the maximum are taken over cell i and the surrounding cells. By *surrounding cells* we mean the three cells that have a common edge with cell i, or less than three, when a boundary is encountered. Quantities that are evaluated at boundaries are not limited. This is a simplified variant of the criterion that Barth uses in [2] and [4]. It is a necessary and, according to [23], close to sufficient condition for a scheme to be LED. We compute the limiter of Barth and Jespersen [2, 3] in the following way:

$$\Theta_{i,e} = \nabla\psi_{i,e} \cdot (\mathbf{r}_e - \mathbf{r}_i), \tag{4.103}$$

$$\phi_{i,e} = \begin{cases} \min\left(1, \frac{\max_j(\psi_j,\psi_i)-\psi_i}{\Theta_{i,e}}\right) & \text{if} \quad \Theta_{i,e} > 0 \\ \min\left(1, \frac{\min_j(\psi_j,\psi_i)-\psi_i}{\Theta_{i,e}}\right) & \text{if} \quad \Theta_{i,e} < 0 \\ 1 & \text{if} \quad \Theta_{i,e} = 0 \end{cases} \tag{4.104}$$

and a unique value of the limiter per cell is taken to be

$$\phi_i = \min_e(\phi_{i,e}). \tag{4.105}$$

In smooth regions where oscillations usually do not appear, the value of ϕ is close to one, while near discontinuities it is close to zero. In this way we switch to the first order scheme when the second order scheme is oscillatory. If only the density equation is solved, the first order scheme is the same as the one presented in [2], and for this one it has been proven that it is non-oscillatory. It is essential that $\nabla\psi_{i,e}$ in the limiter (4.103) is computed in the same way as the gradient in the reconstruction (4.101). If we use the cell-based gradient in

the limiter and the vertex-based gradient in the reconstruction, the oscillations will not be successfully damped, as we have found.

One drawback of the limiter of Barth and Jespersen is that it prevents convergence to a truly steady state. Venkatakrishnan suggests in [33] that the reason for this is the non-differentiability of min and max functions, and proposes the following modification to the limiter of Barth and Jespersen:

$$\phi_{i,e} = \Psi(\Delta_i, \Theta_{i,e}) = \frac{\Delta_i^2 + 2\Delta_i\Theta_{i,e} + \varepsilon^2}{\Delta_i^2 + \Delta_i\Theta_{i,e} + 2\Theta_{i,e}^2 + \varepsilon^2}, \tag{4.106}$$

$$\Delta_i = \begin{cases} \max_j(\psi_j, \psi_i) - \psi_i & \text{for} \quad \Theta_{i,e} > 0, \\ \min_j(\psi_j, \psi_i) - \psi_i & \text{for} \quad \Theta_{i,e} < 0, \\ 1 & \text{for} \quad \Theta_{i,e} = 0, \end{cases} \tag{4.107}$$

where

$$\varepsilon^2 = (Kh)^3 \tag{4.108}$$

is a relaxation factor aimed to prevent limiting of smooth extrema, h is a local mesh parameter, and K is a case-dependent constant. The original limiter of Barth and Jespersen can also be modified to include such a relaxation factor.

However, it is noted in [23] that Venkatakrishnan's limiter also may suffer from stagnation in convergence to steady state, and our experience confirms this. Therefore we use the so-called *historic modification* of the limiter proposed by Delanaye [9]: after a certain number of iterations, the limiter is computed as a minimum of the limiters in the previous and the current time step. This number of iterations is again case dependent, and should be taken large enough. The historic modification leads to the steady state in every case that we tried.

It should be noted that, while it is clear how to limit the convected density in the mass equation, it is not straightforward to decide how to handle the convective term of the energy equation (3.17). If we limit only δp and not p^n, the scheme does not work. The reason might be that it is actually the new pressure that we are computing here, i.e. $p^n + \delta p$. On the other hand, our experiments have shown that limiting the convected kinetic energy term $\frac{1}{2}(\gamma - 1)\mathrm{M}_r^2 \mathbf{m}^* \cdot \mathbf{m}^*/\rho^{n+1}$ does not make any significant difference.

However, the concepts of monotonicity preservation, TVD and LED are not easily extendible to staggered unstructured meshes. It is not clear what an extremum in the discrete momentum field should be, because this field consists of incomparable normal components. This is the reason why it is very hard to design an oscillation detector for the momentum. Furthermore, even the first order upwind scheme [36] is not completely oscillation-free. Very small oscillations in the pressure and the density occur around discontinuities, though they are usually not visible. In the example presented in Sect. 5.4.2 the Mach number obtained by the first order scheme has an oscillation of 0.6% of the jump in Mach.

Fortunately, oscillations typically appear in all variables simultaneously. Therefore we can determine the oscillation indicator ϕ from some scalar quantity. For this purpose we use the density. When strong shocks are present, it may be advantageous to use the inverse of the density, in order to switch to the first order reconstruction when the density becomes small, otherwise the scheme may break down.

Another problem is that we cannot use a formula similar to (4.101) for the momentum because we do not have the momentum vector in any point. Adding

the gradient to a first order reconstruction (4.23) cannot result in a second order reconstruction because there is no point where (4.23) is more than first order accurate. If we remove the gradient part from the vertex-based divergence-free momentum reconstruction, this does not damp the wiggles.

A higher order limited scheme for momentum is obtained as follows. We shall denote the piecewise constant approximation of $\mathbf{m}_e \cdot \mathbf{N}_i$ used in scheme [36] by $(\mathbf{m}_e \cdot \mathbf{N}_i)^1$ and the piecewise linear approximation developed in Sect. 4.11.2 by $(\mathbf{m}_e \cdot \mathbf{N}_i)^2$. The approximation that we use here is a linear combination of these two, i.e.

$$\mathbf{m}_e \cdot \mathbf{N}_i \approx (1-\phi)(\mathbf{m}_e \cdot \mathbf{N}_i)^1 + \phi(\mathbf{m}_e \cdot \mathbf{N}_i)^2, \tag{4.109}$$

Of course, we cannot expect that spurious wiggles will be totally eliminated because the stencil that we use for the reconstruction of momentum is inevitably different from the one that is used to compute the limiter, among other reasons.

Chapter 5

Numerical validation

5.1 Summary

The purpose of the numerical experiments described in this chapter is to show improved accuracy compared to the first order upwind method described in [36, 37].

In Sect. 5.2 we present results obtained for three inviscid incompressible flow problems. Regardless of their simplicity, these examples elucidate some difficulties that can arise in the presence of outflow boundaries and stagnation points.

In Sect. 5.3 the Navier-Stokes equations for incompressible flow are solved for some viscous incompressible flow problems. For the exact Kovasznay solution the behavior of the error as the grid is refined is studied. The backward facing step and circular cylinder flows are computed and comparison is made of some properties of the obtained results with benchmark data.

In Sect. 5.4, the Euler equations are solved for some compressible flow problems with our Mach-uniform scheme. The Ringleb flow is an irrotational transonic flow with known exact solution. The accuracy of the superlinearly convergent scheme with and without limiting is assessed. Flow in a channel with a bump is computed at low and high Mach number. A transonic airfoil flow for which the pressure distribution is fairly well-known is computed. To determine carefully whether Rankine-Hugoniot conditions are satisfied by the scheme, a shocktube problem is computed.

5.2 Incompressible inviscid flow

In this section the accuracy of the scheme is investigated for the Euler equations for the incompressible flow. The pressure-correction method for incompressible flow that is used has been described in Sect. 3.3. For the inertia term, upwind-biased vertex-based divergence-free linear reconstruction is used, which was described in Sects. 4.6.1 and 4.11.2. The pressure term is discretized by using the path integral method presented in Sect. 4.11.3. The order of accuracy is determined by comparison with exact solutions and solutions of benchmark problems. So the accuracy of the scheme is expected to be better than first order.

5.2.1 Converging flow

We consider the same flow as in Example 1 (cf. Fig. 4.10), but this time we solve the incompressible flow Euler problem using the pressure-correction method for incompressible flows described in Sect. 3.3. The exact solution is

$$u = \sin x \sin y, \quad v = \cos x \cos y, \quad p = 1 + \frac{1}{4}(\cos 2x - \cos 2y). \tag{5.1}$$

The domain is the square $[-0.5, 0.5] \times [-1.3, -0.3]$. The three types of grids shown in Fig. 4.11 are used. The exact momentum is prescribed at the inflow boundary and the exact pressure is prescribed at the outflow boundary (see Fig. 4.10). The initial solution is set to zero.

Results for the first order scheme are given in Figs. 5.1, 5.2 and 5.3. Symbols τ and ε denote the local and the global truncation error, as defined in Example 1 in Sect. 4.11.1. For this example, the velocity converges, but convergence of pressure stagnates on the irregular grid.

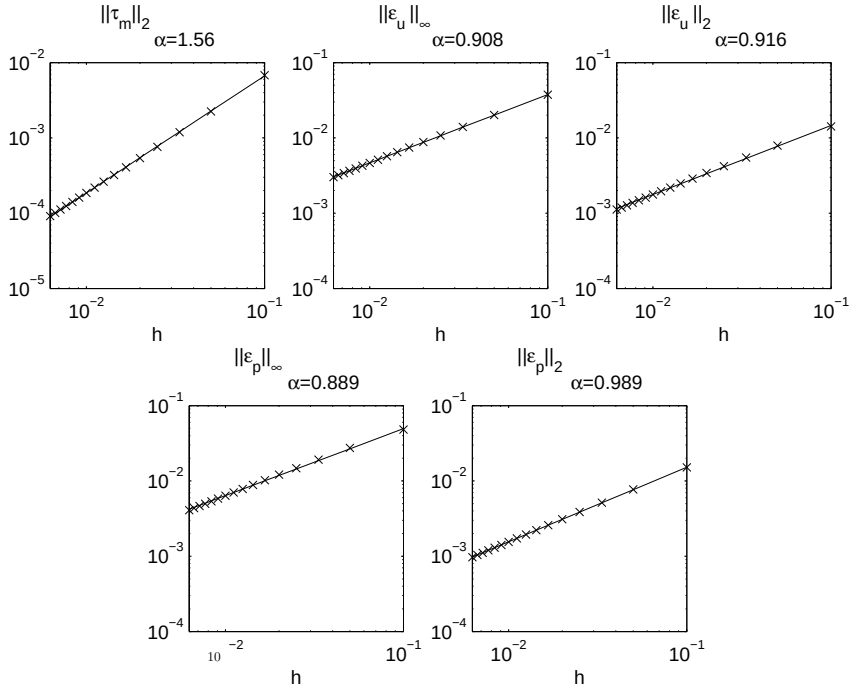

Figure 5.1: Converging flow, Cartesian grids, first order scheme.

The results for the scheme with vertex-based divergence-free linear reconstruction are given in Figs. 5.4, 5.5 and 5.6. Limiting is not applied in these experiments. The estimated order of accuracy on the unstructured grid is 1.91, on the regular grid it is 1.96, and on the Cartesian grid it is 1.79, so convergence is superlinear. The error is not considerably larger at the outflow boundary than inside the domain. But it is a bit larger in the vicinity of the left and the right boundary, where the fluid flows in at very small angle. Wiggles were not detected, presumably because the solution is smooth. Therefore limiting was not necessary.

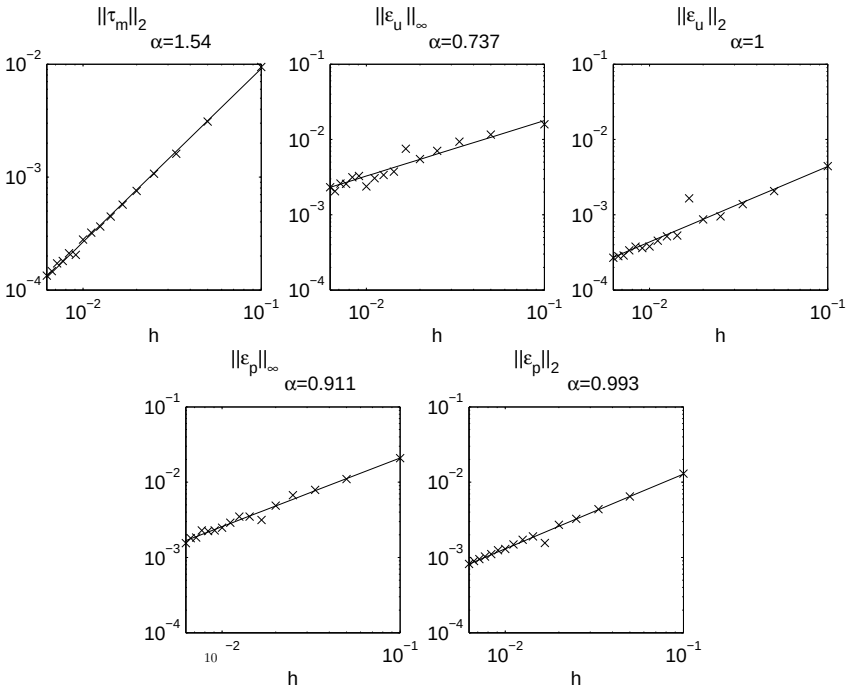

Figure 5.2: Converging flow, regular grids, first order scheme.

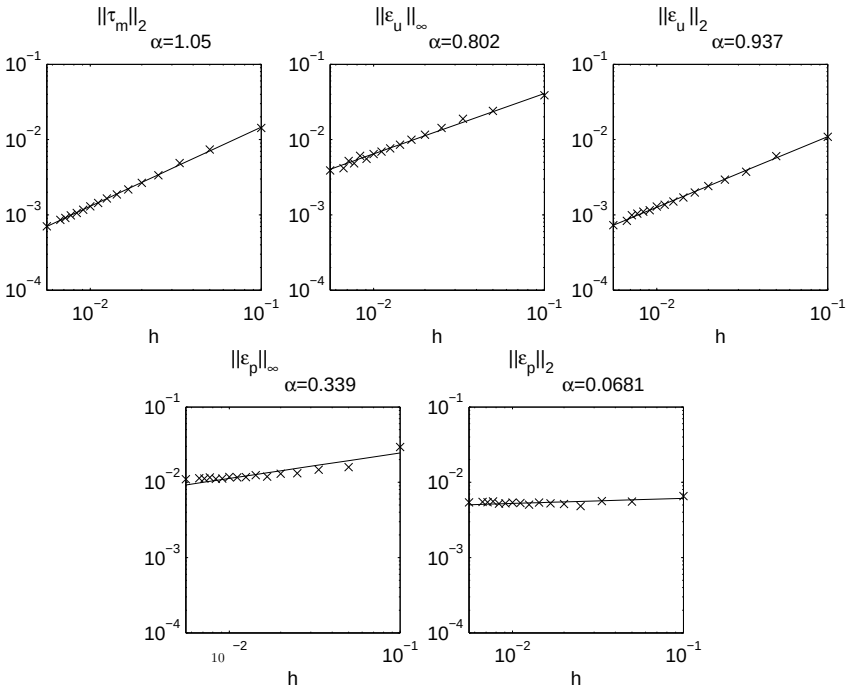

Figure 5.3: Converging flow, irregular grids, first order scheme.

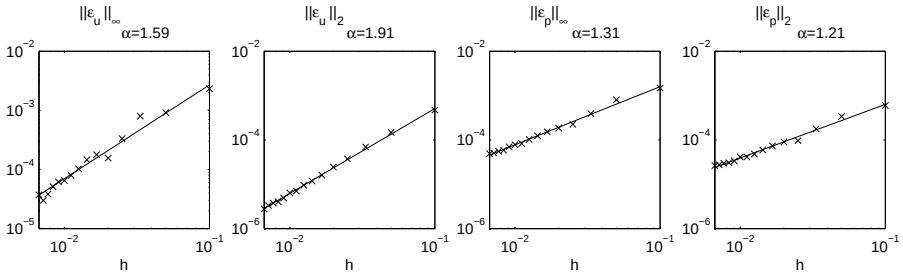

Figure 5.4: Converging flow, irregular grids, linear reconstruction.

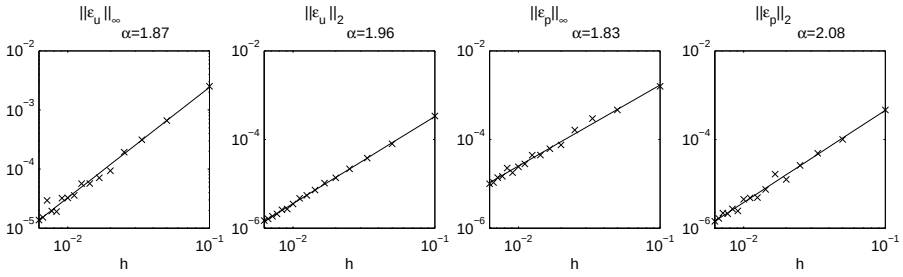

Figure 5.5: Converging flow, regular grids, linear reconstruction.

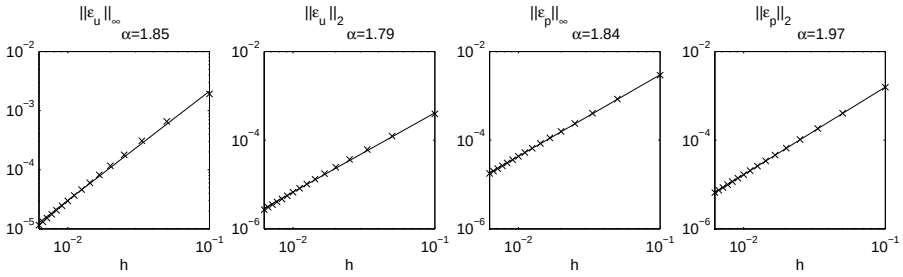

Figure 5.6: Converging flow, Cartesian grids, linear reconstruction.

5.2.2 Solid body rotation

An exact solution of the Euler equations for incompressible flow is solid body rotation around point (x_0, y_0) (see Fig. 5.7):

$$u = y - y_0, \quad v = x_0 - x, \quad p = 1 + \frac{1}{2}\left[(x - x_0)^2 + (y - y_0)^2\right]. \tag{5.2}$$

We take $(x_0, y_0) = (-1/4, -1/4)$, and we solve this problem in the $[0, 1] \times [0, 1]$ square.

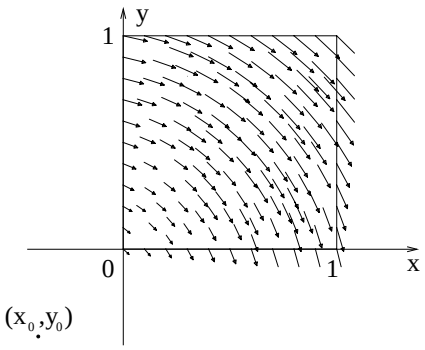

Figure 5.7: Solid body rotation.

The exact momentum is prescribed at the inflow (the upper and the left) boundary and the exact pressure is prescribed at the outflow (the lower and the right) boundary. The initial solution is set to zero.

The 2-norm of the local truncation error τ_m and maximum and 2-norms of the global truncation error of the first order method are shown in Figs. 5.8, 5.9 and 5.10. In the case of the irregular grid the solution stops improving after the 70×70 case, although the local truncation error keeps decreasing. On the regular and irregular grids, the accuracy is less than for the preceding converging flow example. We think this is due to the fact that the exact solution varies more at the outflow boundaries and that the length of the outflow boundaries is larger than in the preceding example. In many practical applications, outflow boundaries are placed in regions of nearly uniform flow, so the influence of outflow is negligible. We include this example to demonstrate that also here the scheme with linear reconstruction brings significant improvement, as shown in Figs. 5.11–5.13

The estimated order of accuracy measured in 2-norm on a irregular grid is 2.31, on regular grid it is 2.52, and on a structured grid it is 1.85. The error is somewhat larger at the outflow boundary. The exact momentum is a linear function, so our reconstruction is exact, although the face integrals of quadratic convection are not. For this reason this might not be a good benchmark for the accuracy of this method.

5.2.3 Stagnation point flow

The exact solution is

$$u = x, \quad v = -y, \quad p = 1.5 - \frac{1}{2}\left(x^2 + y^2\right). \tag{5.3}$$

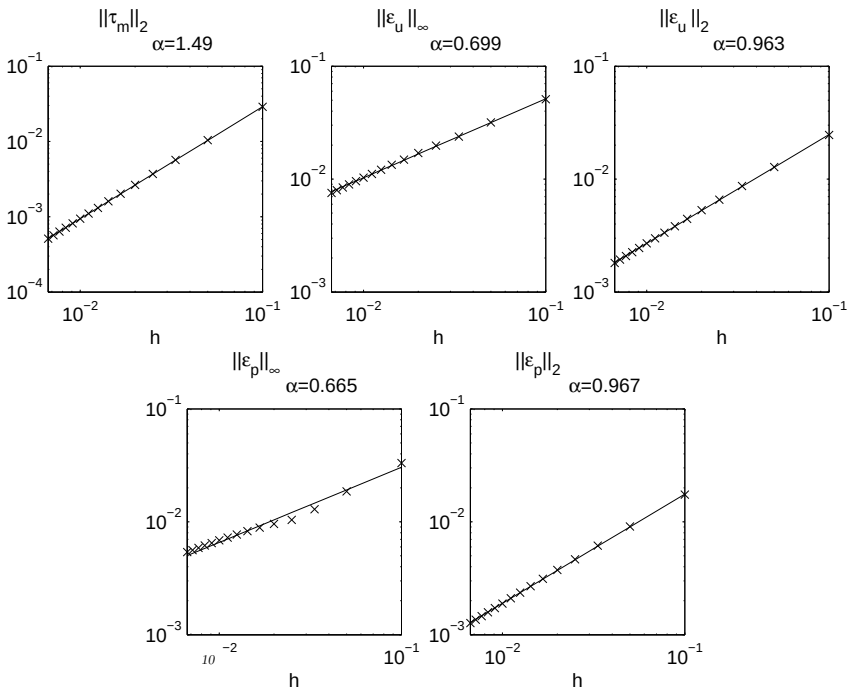

Figure 5.8: Solid body rotation, Cartesian grids, first order scheme.

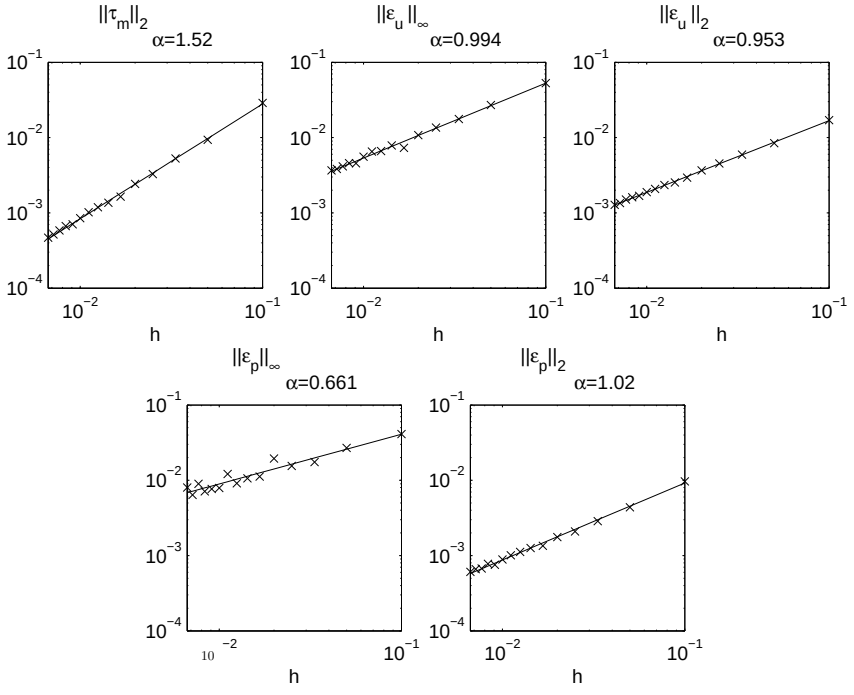

Figure 5.9: Solid body rotation, regular grids, first order scheme.

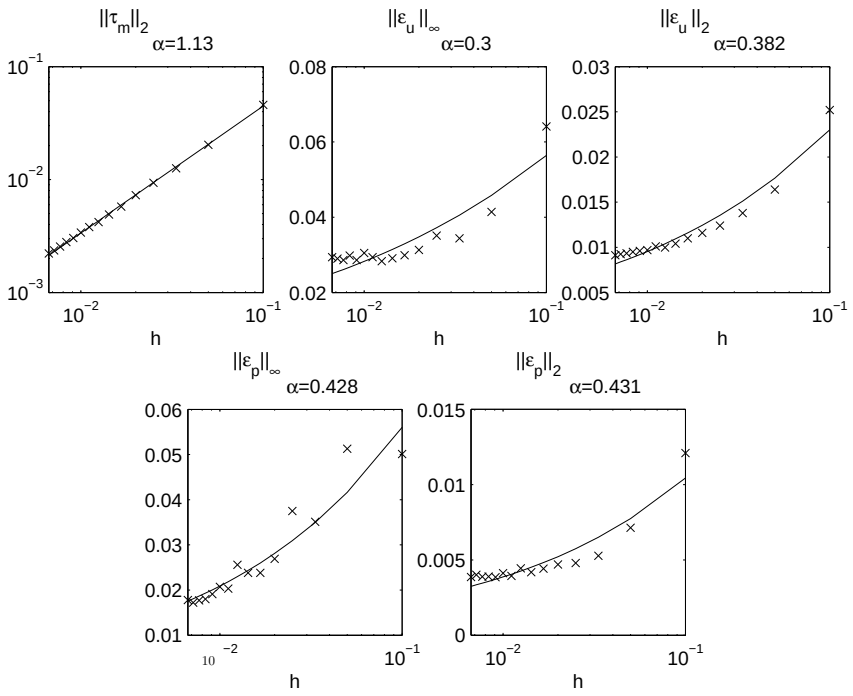

Figure 5.10: Solid body rotation, irregular grids, first order scheme.

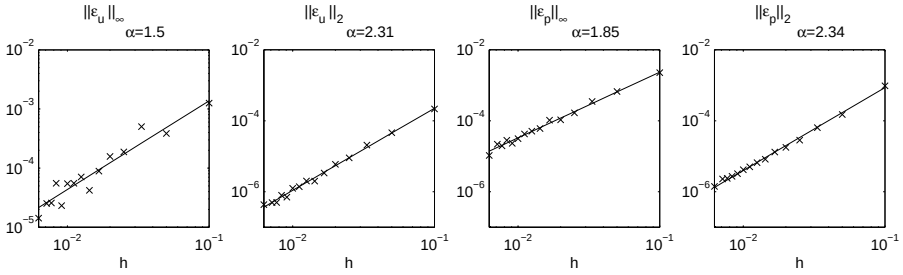

Figure 5.11: Solid body rotation, linear reconstruction, irregular grids.

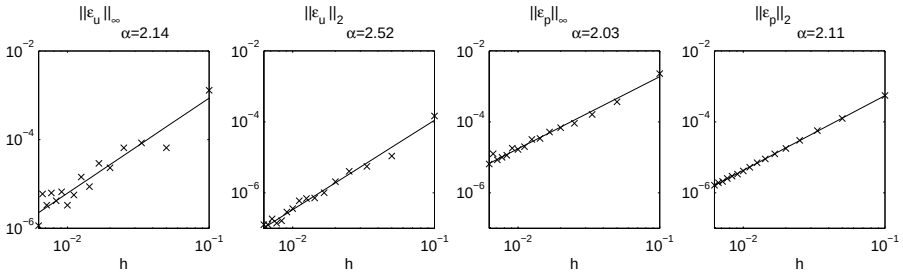

Figure 5.12: Solid body rotation, linear reconstruction, regular grids.

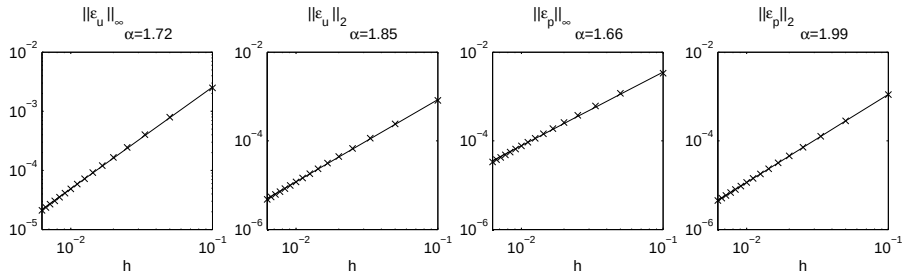

Figure 5.13: Solid body rotation, linear reconstruction, Cartesian grids.

The domain is the unit square. The solution is shown in Fig. 5.14.

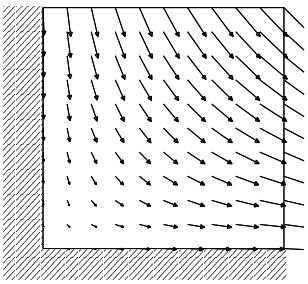

Figure 5.14: Stagnation point flow

This example was solved on the three types of grids shown in Fig. 4.11. Momentum is prescribed at the inflow (upper) boundary, pressure is prescribed at the outflow (right) boundary, and the normal momentum is set to zero at the the left and the lower boundary. The exact solution is used as initial solution. The errors of the solutions obtained with the first order scheme are shown in Figs. 5.15, 5.16 and 5.17.

Even on the Cartesian grid the order of convergence is much less than one, and the situation is not better on the unstructured grids. Fig. 5.18 shows the error of the obtained solution. It was computed by reconstructing the error of the velocity vector in the vertices from the errors of the normal components in the faces. First the tangential components of the error are computed in faces by using the central reconstruction formula. Then the error in the vertices is computed by using the following formula:

$$\boldsymbol{\varepsilon}_{\text{vertex}} = \frac{\sum_j \omega_j \boldsymbol{\varepsilon}_{m,j}}{\sum_j \omega_j}, \tag{5.4}$$

where summations run over the faces connected to the vertex. Weight ω_j is the inverse of the distance between the vertex and the midpoint of face j, and $\boldsymbol{\varepsilon}_{m,j}$ is the error vector in face j that consists of the error $\varepsilon_{m,j}$ and the reconstructed tangential component of the error.

This error is seen to have the structure of a vortex.

We think that the cause of this error is the artificial viscosity. It is known that for certain problems (for example the flow around a circular cylinder, see

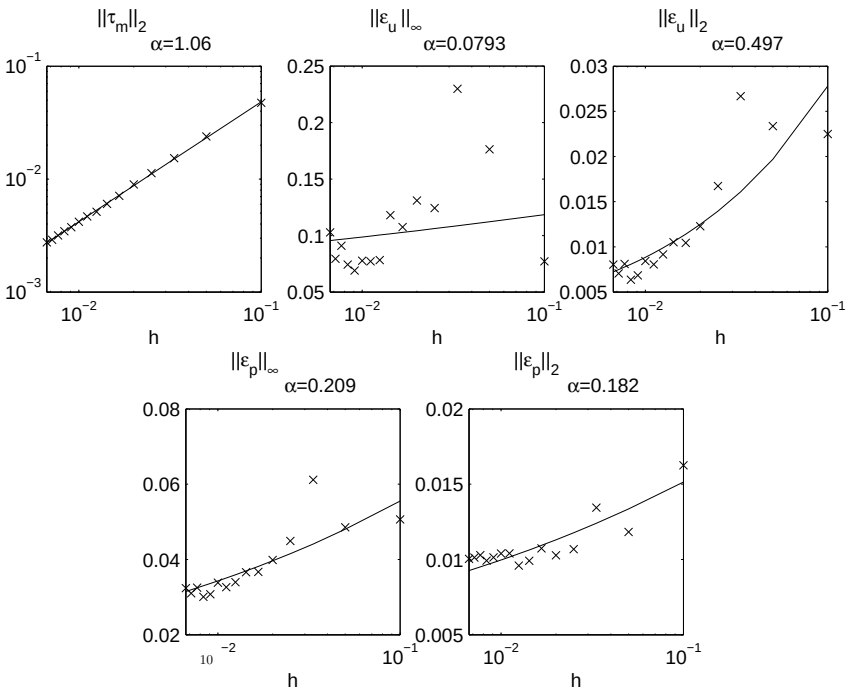

Figure 5.15: Stagnation point flow, irregular grids, first order scheme.

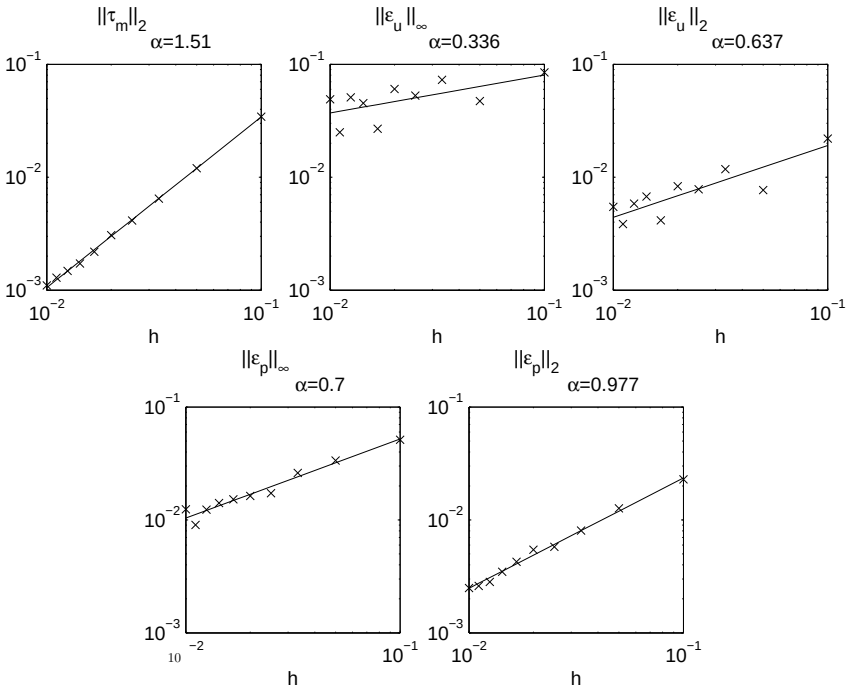

Figure 5.16: Stagnation point flow, regular grids, first order scheme.

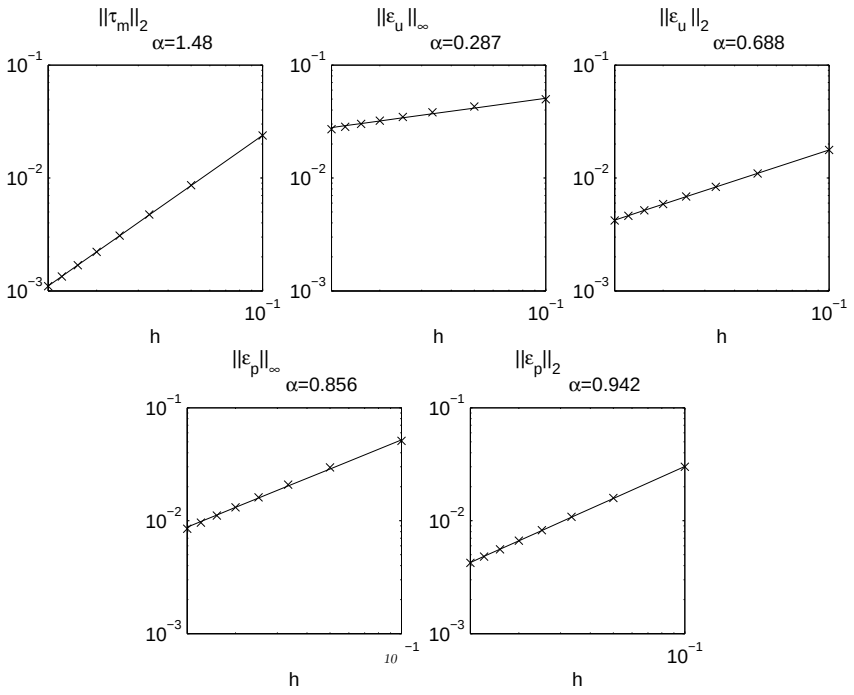

Figure 5.17: Stagnation point flow, Cartesian grids, first order scheme.

Example 5.3.3) the solution when $\mu \to 0$ does not converge to the solution of the inviscid flow problem. This problem is similar to the situation behind the cylinder.

The results obtained with the vertex-based divergence-free linear reconstruction are given in Figs. 5.19 and 5.20. As in the first order case, the error takes a form of a vortex (see Fig. 5.18). The accuracy is significantly improved. If the grid is Cartesian the error is much higher, for example in the 100×100 case the maximal error in velocity is 0.015, and the maximal error in pressure is 0.0019. We think that the reason is that parallel faces exist in reconstruction stencils, which may reduce the rank of the reconstruction matrix.

5.3 Incompressible viscous flow

We now turn to the numerical solution of the Navier-Stokes equations for incompressible flow. As in the case of the incompressible inviscid flow (see Sect. 5.2), the pressure-correction method for incompressible flow was used for time-stepping, the upwind-biased vertex-based divergence-free linear reconstruction was used in the inertia term, and the pressure term was approximated using the path integral method. The viscous term was approximated using the vertex-based divergence-free linear interpolation, presented in Sections 4.6.1 and 4.11.1.

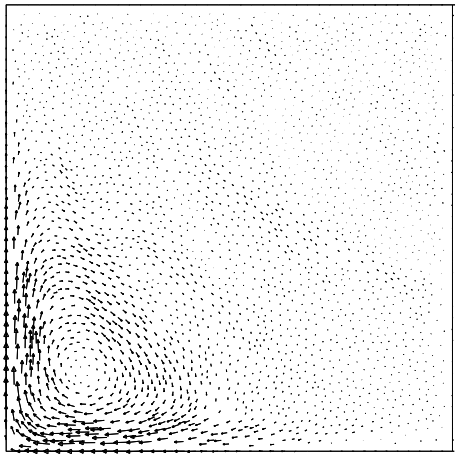

Figure 5.18: The error takes the form of a vortex.

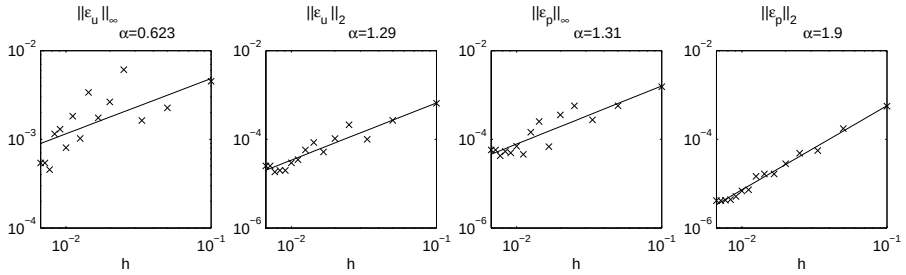

Figure 5.19: Stagnation point flow, irregular grids, linear reconstruction.

5.3.1 Kovasznay flow

This is a laminar flow behind a two-dimensional grid presented in [20] (see Fig. 5.21). The exact solution is

$$v(x,y) = \frac{\lambda}{2\pi}\,\mathrm{e}^{\lambda x}\sin(2\pi y), \qquad u(x,y) = 1 - \mathrm{e}^{\lambda x}\cos(2\pi y),$$
$$p(x,y) = c - \tfrac{1}{2}\,\mathrm{e}^{2\lambda x}, \qquad \lambda = \frac{\mathrm{Re}}{2} - \left(\frac{\mathrm{Re}^2}{4} + 4\pi^2\right)^{\frac{1}{2}}, \tag{5.5}$$

where $\mathrm{Re} = 1/\mu$. We solved this problem for $\mathrm{Re} = 10$ using three types of grids show in Fig. 5.22 on the domain $[0.1, 5] \times [-0.5, 0.5]$ shaded in Fig. 5.21. The exact stress and the pressure were prescribed on the outflow boundary, and the exact velocity was prescribed on the other parts of the boundary. In order to make the tests more severe, the finer grids were not created from the coarser grids by subdividing their triangles, but were created from scratch. Unnecessarily irregular and stretched grids were used to test the scheme severely.

The 1-norm, 2-norm and ∞-norm of τ_m, ε_m and ε_p are shown in Figs. 5.23-5.25 as functions of a mesh parameter h for the schemes with piecewise constant and piecewise linear reconstruction.

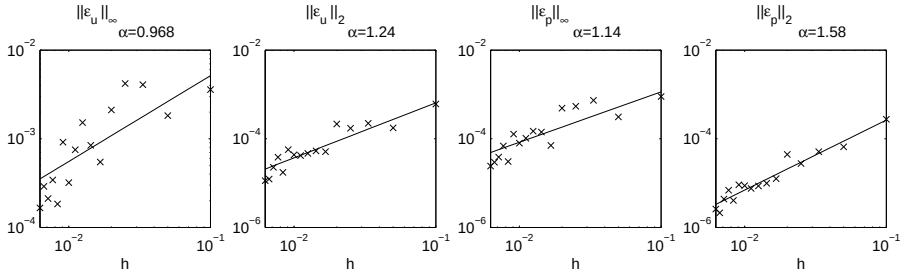

Figure 5.20: Stagnation point flow, regular grids, linear reconstruction.

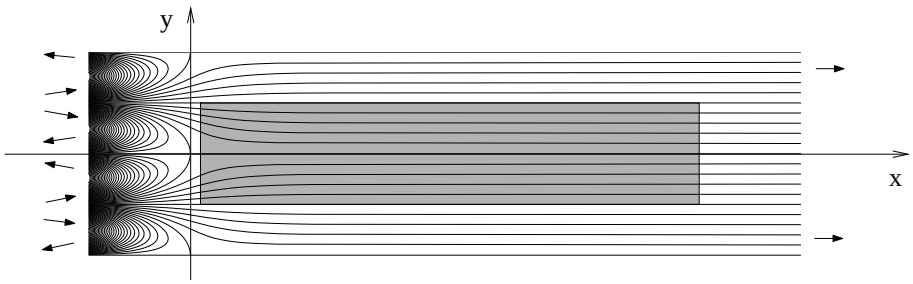

Figure 5.21: Kovasznay flow - streamlines and the shaded computational domain.

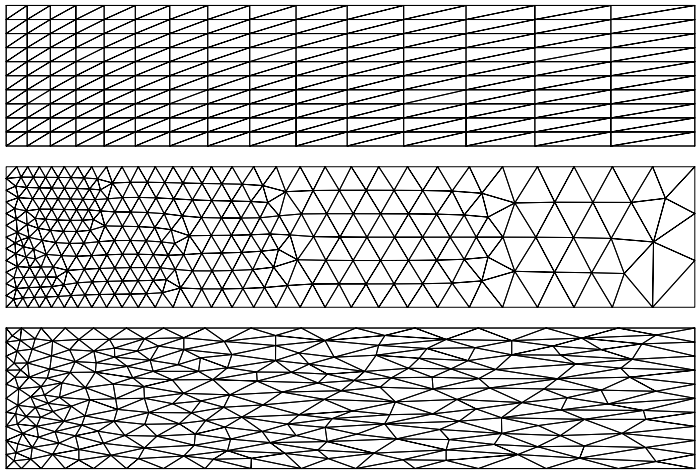

Figure 5.22: Three types of grid used for the Kovasznay flow: the stretched Courant grid, the unstructured grid, and the stretched unstructured grid.

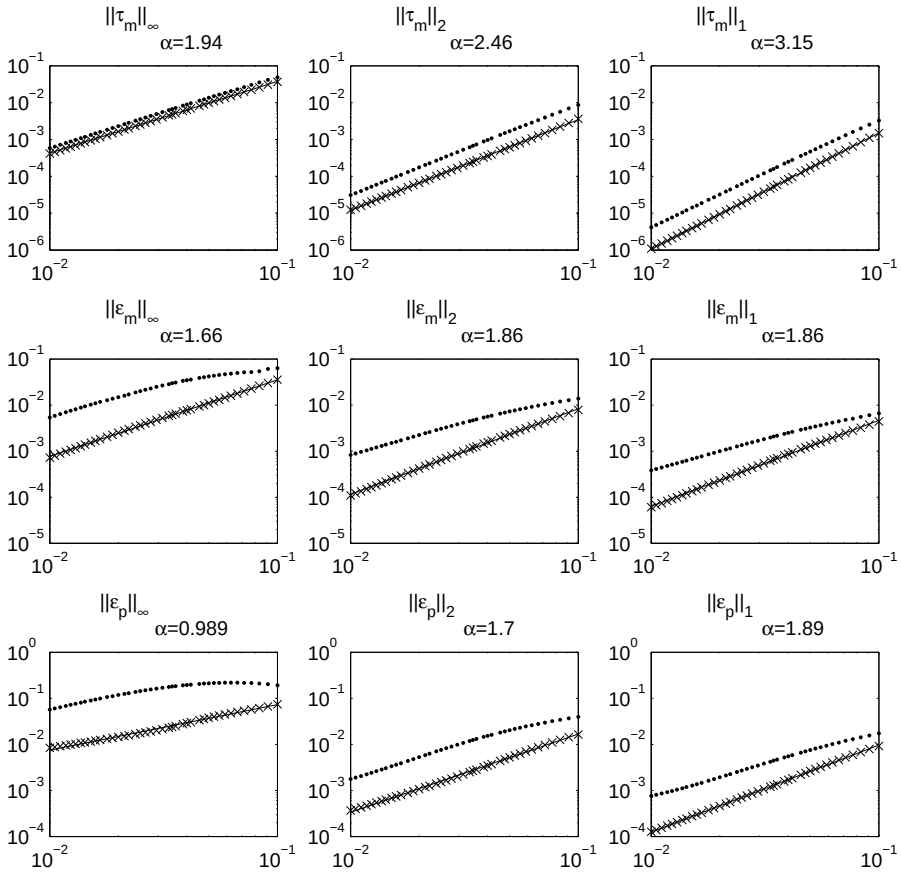

Figure 5.23: Accuracy of the computed Kovasznay flow - stretched Courant grid. "▪": first order scheme; "×": scheme with linear reconstruction.

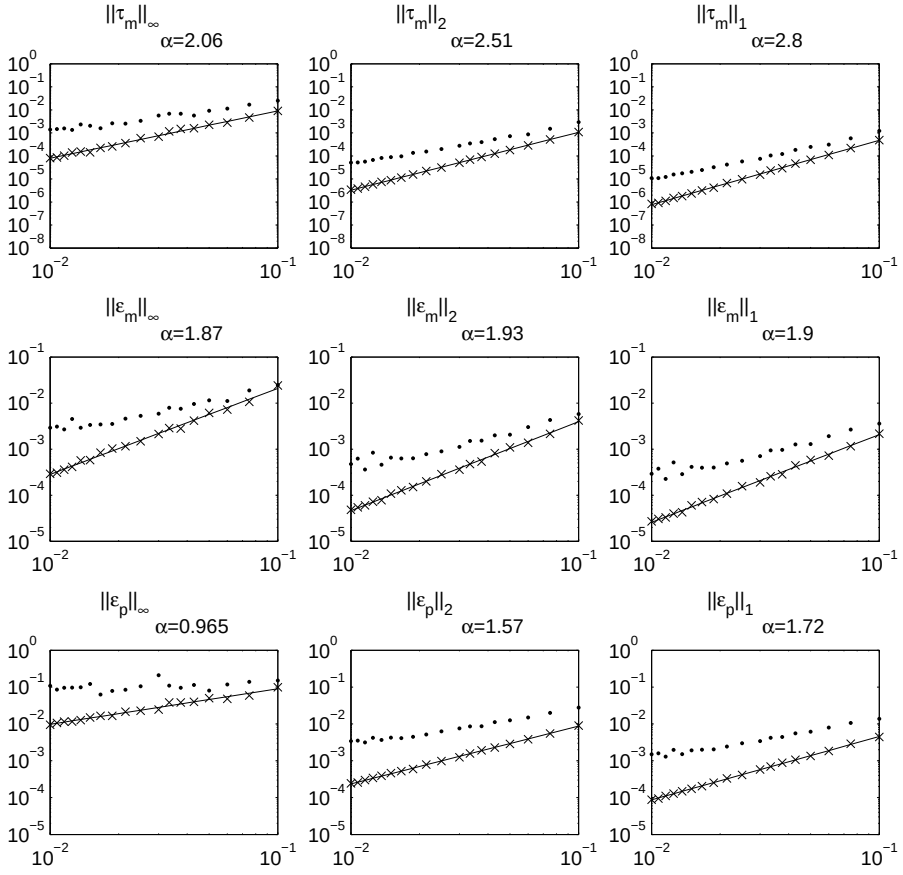

Figure 5.24: Accuracy of the computed Kovasznay flow - unstructured grid. "▪": first order scheme; "×": scheme with linear reconstruction.

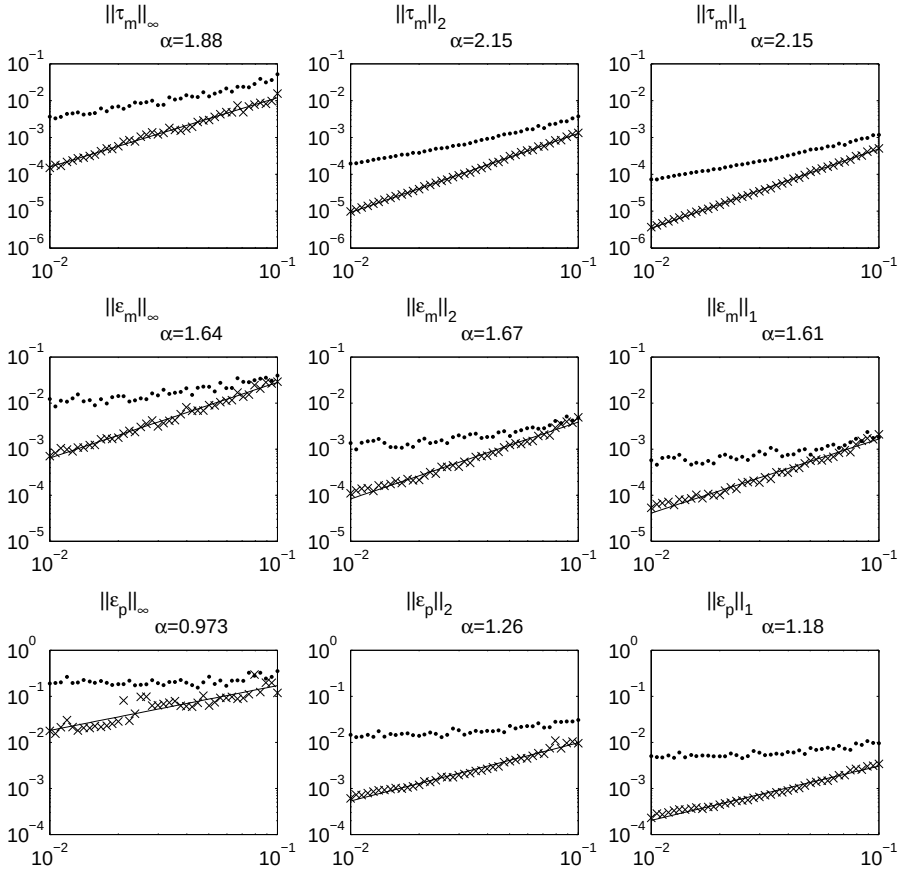

Figure 5.25: Accuracy of the computed Kovasznay flow - stretched unstructured grid. "▪": first order scheme; "×": scheme with linear reconstruction.

The accuracy of the scheme is seen to be less than second order, but convergence is clearly superlinear. As usual for schemes for the incompressible Navier-Stokes equations, pressure is approximated with less accuracy than momentum.

We also tested the accuracy of the vertex-based divergence-free reconstruction. The momentum is reconstructed in each vertex using the divergence free linear reconstruction. The vector of differences between these reconstructed momenta and the exact momenta in vertices is denoted by δ_m. The vector of differences between the gradients of the reconstruction polynomials and the exact momentum gradients in vertices is denoted by $\delta_{\nabla m}$. This is shown in Figs. 5.26–5.28.

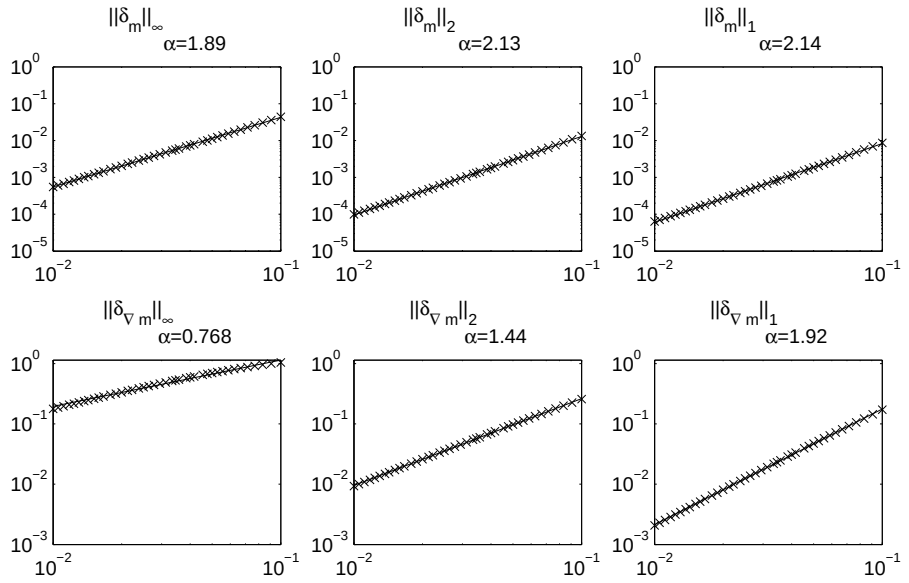

Figure 5.26: Accuracy of the vertex-based divergence-free reconstruction of the Kovasznay flow - stretched Courant grid.

We see in these figures that the reconstruction is a bit less accurate on the Courant grid than on the other two types of grid. This is probably because of the rank reduction due to parallel faces in the reconstruction stencils (see Sect. 4.6). This loss of accuracy is reflected in the accuracy of the computed flow, see Figs. 5.23-5.25.

5.3.2 Backward facing step

The flow over a backward facing step has been computed by many authors. The length of the primary recirculation zone depends sensitively on the Reynolds number. Underprediction of this length is a consequence of artificial numerical dissipation. Furthermore, somewhere between Re = 400 and Re = 800 a secondary separation zone appears at the upper wall. Schemes that have too much artificial numerical dissipation do not resolve this secondary separation zone.

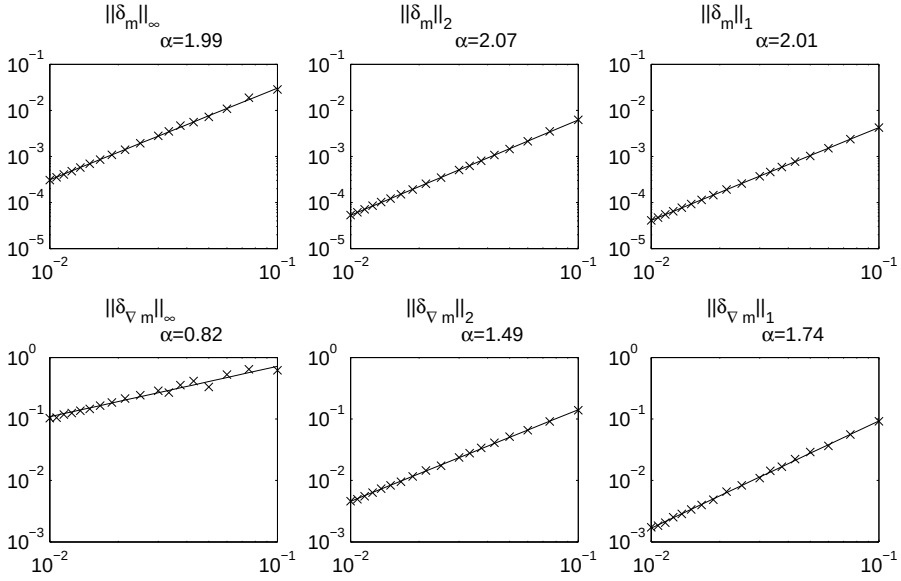

Figure 5.27: Accuracy of the vertex-based divergence-free reconstruction of the Kovasznay flow - unstructured grid.

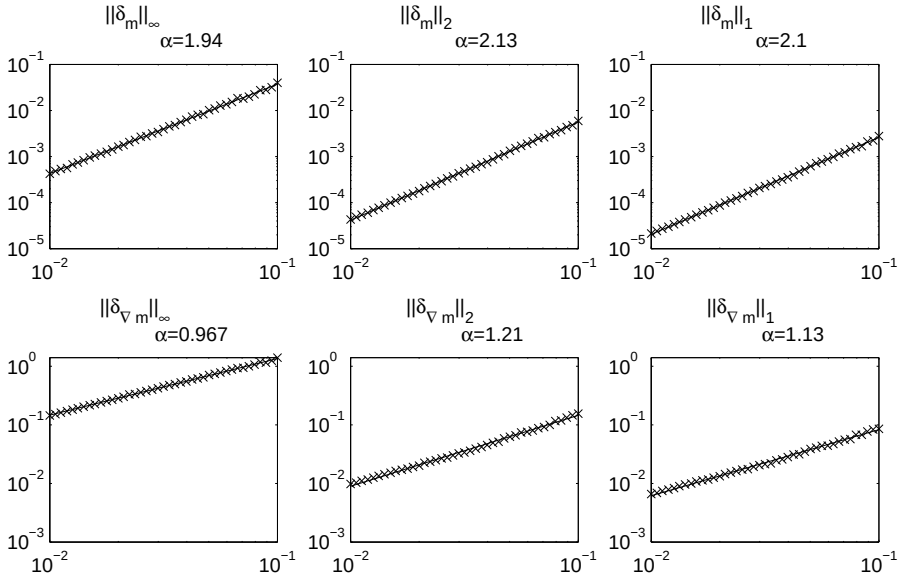

Figure 5.28: Accuracy of the vertex-based divergence-free reconstruction of the Kovasznay flow - stretched unstructured grid.

The domain shown in Fig. 5.29 is 18.87 units long (not including the inlet) and 1 unit wide. The inlet is 0.5 units wide and 0.5 units long. The initial solution is zero. The inflow velocity profile is parabolic and given by $24t(1-y)(y-0.5)$ when time $t < 1$ and $24(1-y)(y-0.5)$ for $t \geq 1$, so that the inflow is switched on gradually.

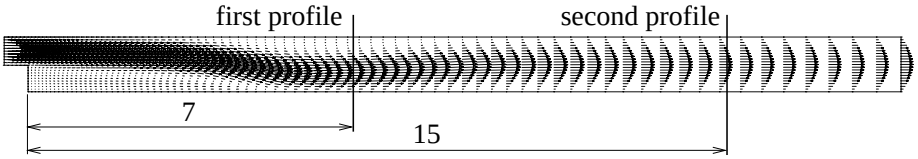

Figure 5.29: Backward facing step, Re = 800.

We found the steady solution to this problem using the vertex-based divergence-free linear reconstruction and various resolutions of two types of grid (called regular and distorted) shown in Fig. 5.30 for two different Reynolds numbers: 400 and 800. We define the Reynolds number as $\rho LU/\mu$, where L is the width of the channel, U is the average velocity at the inflow boundary for $t \geq 1$, μ is the viscosity. Fig. 5.29 shows the steady solution for Re = 800 obtained on a grid consisting of 3320 triangles.

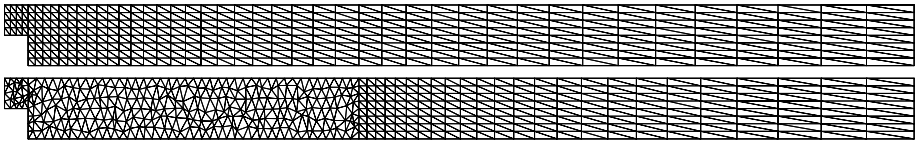

Figure 5.30: Grids used for the backward facing step problem. Upper: "regular", lower: "distorted".

Velocity profiles at distances 7 and 15 from the step obtained for Re = 800 with the finest regular grid that we used are given in Figs. 5.31 and 5.32, in comparison with the benchmark data published in [11]. The accuracy in vertical velocity in Fig. 5.31 is not as bad as it seems, because this quantity is close to zero in magnitude.

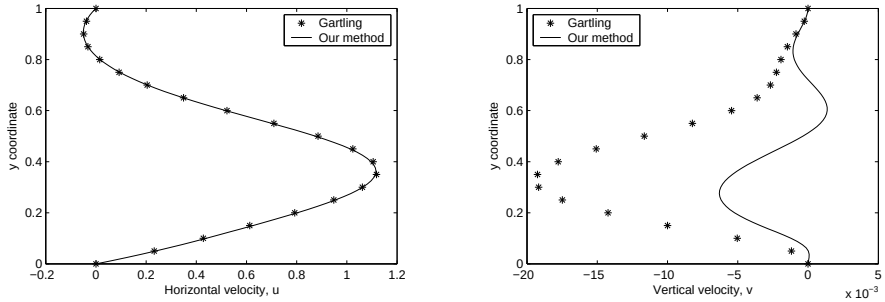

Figure 5.31: Velocity profiles at distance 7 from the step, Re = 800.

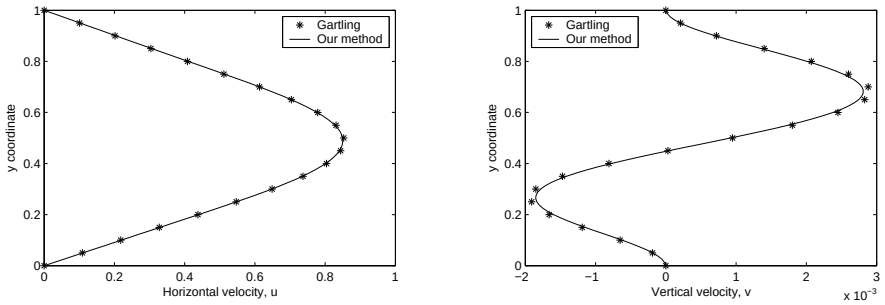

Figure 5.32: Velocity profiles at distance 15 from the step, Re = 800.

We calculated the position of the primary reattachment point as a function of a mesh parameter, and compared it to the experimental data published in [1] and to the numerical prediction of [11] and [19]. Results are given in Fig. 5.33. For Reynolds number 400 our scheme gives a very good estimate of this

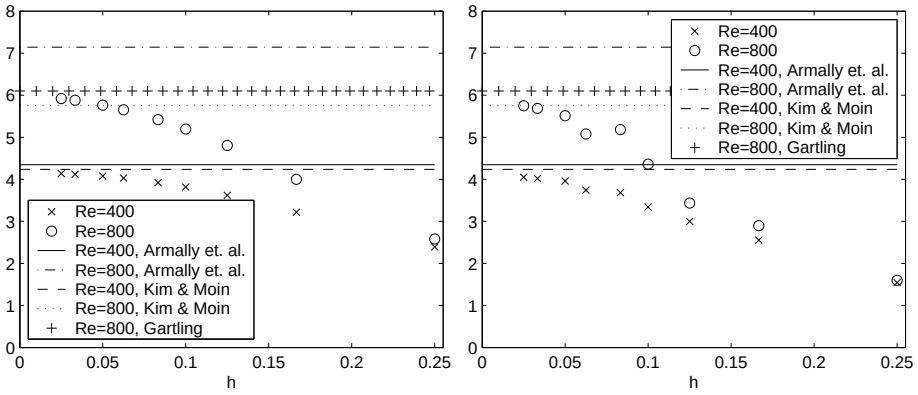

Figure 5.33: Position of the primary reattachment point for regular (left) and distorted grids (right).

parameter, which is known to be very sensitive to numerical dissipation.

Reynolds number 800 is very high for an upwind-biased scheme, because the artificial viscosity becomes more influential. There is a noticeable discrepancy between the experimental results [1] and the numerical prediction [19]. For a coarse grid the reattachment point computed by our method is way off. However, Fig. 5.33 shows that for fine grids our method gives a more accurate prediction than [19]. Fig. 5.29 suggests and Fig. 5.31 shows that our scheme resolves the secondary recirculation zone.

The observed quality of the prediction of the length of the primary recirculation zone and the resolution of the secondary recirculation zone are characteristic of a higher order scheme.

5.3.3 Flow around a circular cylinder

Flow past a circular cylinder is a classical example of an unsteady flow. For Re > 47 vortex shedding occurs, and a so-called Karman street of vortices is formed in the wake of the cylinder. Schemes with too much artificial dissipation show onset of unsteadiness at higher Re, or not at all. For Reynolds numbers higher than about 180 the flow becomes turbulent (see [39]). The vortex shedding process is periodic in time. The dimensionless frequency is called the Strouhal number, defined as $\mathrm{S} = D/(uT)$, with D the cylinder diameter, u the upstream velocity and T the period. Williamson and Brown in [39] give the following relation:

$$\mathrm{S} = 0.2665 - \frac{1.018}{\sqrt{\mathrm{Re}}}, \qquad \mathrm{Re} = \frac{\rho u D}{\mu}, \tag{5.6}$$

which is in good agreement with the experimental data.

We solved this problem using the vertex-based divergence-free linear reconstruction for various Reynolds numbers on six grids, one of which is shown in Fig. 5.34, the other has twice as many vertices along each boundary, the third one has twice as many vertices at the cylinder as the second one, and the other three were made by distorting the first three grids (see Fig. 5.35). Uniform velocity $u = 1$, $v = 0$ was prescribed along the left, the upper and the lower boundary, pressure was prescribed at the outflow (right) boundary, and the velocity was set to zero along the cylinder. An instantaneous pressure distribution for Re = 100 is shown in Fig. 5.36. Obtained Strouhal numbers in comparison

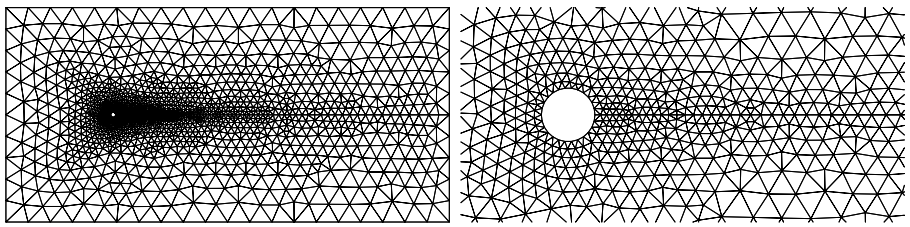

Figure 5.34: Regular mesh used for the flow around a cylinder; full domain (left) and a detail (right).

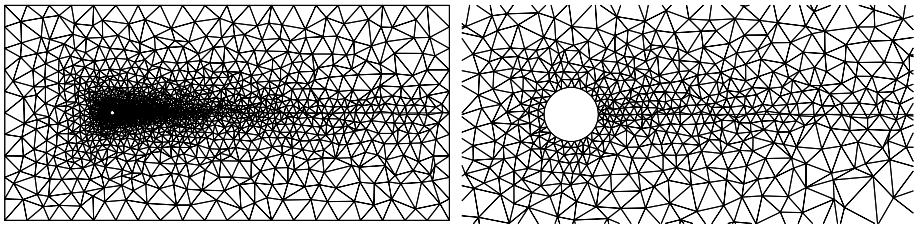

Figure 5.35: Distorted mesh used for the flow around a cylinder; full domain (left) and a detail (right).

with formula (5.6) are given in Fig. 5.37. Agreement is quite satisfactory on the medium and fine grids, both regular and distorted.

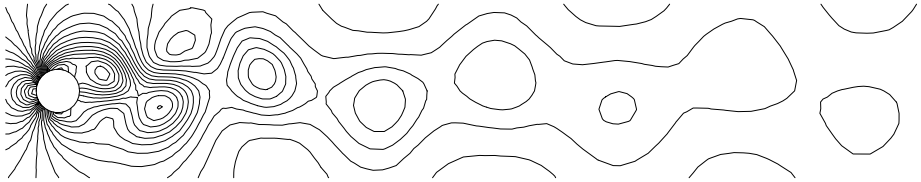

Figure 5.36: Isobars in flow around a cylinder, Re = 100, coarse regular grid.

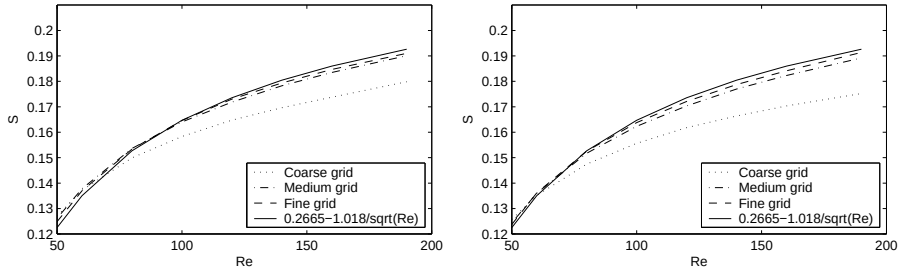

Figure 5.37: Strouhal numbers obtained on the regular (left) and the distorted grids (right).

5.4 Compressible flow

In this section, numerical tests of the Mach-uniform method described in Sect. 3.5 are presented. The momentum equation was discretized in the same way as in the incompressible flow case, see Sect. 5.3. The discretization of the density and the pressure-based energy equation, as well as the linear reconstruction of scalar data used here, was presented in Sects. 4.4, 4.5 and 4.13. We also used flux limiting, which was described in Sect. 4.16. The first test concerns an exact solution for irrotational shock-free transonic flow. The second test concerns flow in a channel with a bump at various Mach numbers. The third test is the flow around the NACA 0012 airfoil. The fourth test is Sod's shocktube problem.

5.4.1 Ringleb flow

For this transonic flow in a curved duct the analytical solution of the potential flow equations is known. It was first published in [29], and a description may be found in [8]. This is a valuable test case to investigate the accuracy of numerical schemes. The flow is directed upwards in Fig. 5.38. There is a small supersonic flow region in the middle of the right wall, but there is no shock. The maximum Mach number is 1.09. The problem was solved on regular and distorted grids (see Figs. 5.38 and 5.40) using the first order and the second order method. The exact Mach isolines are circular arcs. The exact velocity and the density were prescribed along the inlet (lower boundary), normal velocity was set to zero along the left and the right boundary, and the pressure was prescribed along the outflow (upper) boundary. The grids with 10-100 cells along the inlet were generated independently from each other, preserving the quality of the grids

shown in Figs. 5.38 and 5.40. Fig. 5.39 shows the error of the solution obtained on the regular grid. The independent parameter along the horizontal axis is the mesh parameter, one over the number of cells along the inlet boundary.

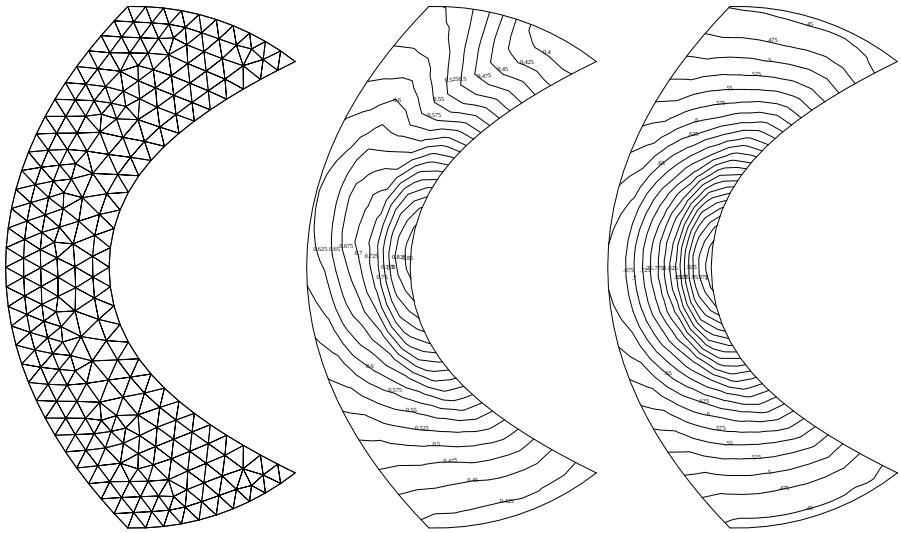

Figure 5.38: Ringleb flow, regular mesh (left) and Mach isolines obtained with the first order (middle) and the second order unlimited (right) method.

Fig. 5.39 gives local (τ) and global (ε) truncation error norms of the first order scheme and of three variants of the scheme with linear reconstruction, described in Sect. 4.5. The best results were obtained with the vertex-based scalar reconstruction in the density equation, and the cell-based reconstruction in the pressure equation. If the vertex-based reconstruction is used in both equations, the results are less accurate, probably because more artificial dissipation is introduced. If the cell-based reconstruction is used in the density equation, the scheme does not converge in time when strong shocks are present (see Sect. 5.4.2), therefore we do not consider this variant. We see that the limited scheme is a bit less accurate than the unlimited one, i.e. the limiter is not completely inactive. If a non-zero relaxation factor is used (see (4.16)), the difference in accuracy is smaller.

On distorted grids (see Figs. 5.40 and 5.41) the scheme still works, but it is less accurate than on the regular ones. However, the linear reconstruction improves the accuracy considerably.

5.4.2 Channel with a bump

The flow in a channel with a circular arc bump is used to evaluate the presented method for computations of steady state solutions. We present the incompressible and the supersonic flow case. In the incompressible flow case the height of the bump is 10% of the channel height, while in the supersonic flow case it is 4%. This benchmark was proposed in [10].

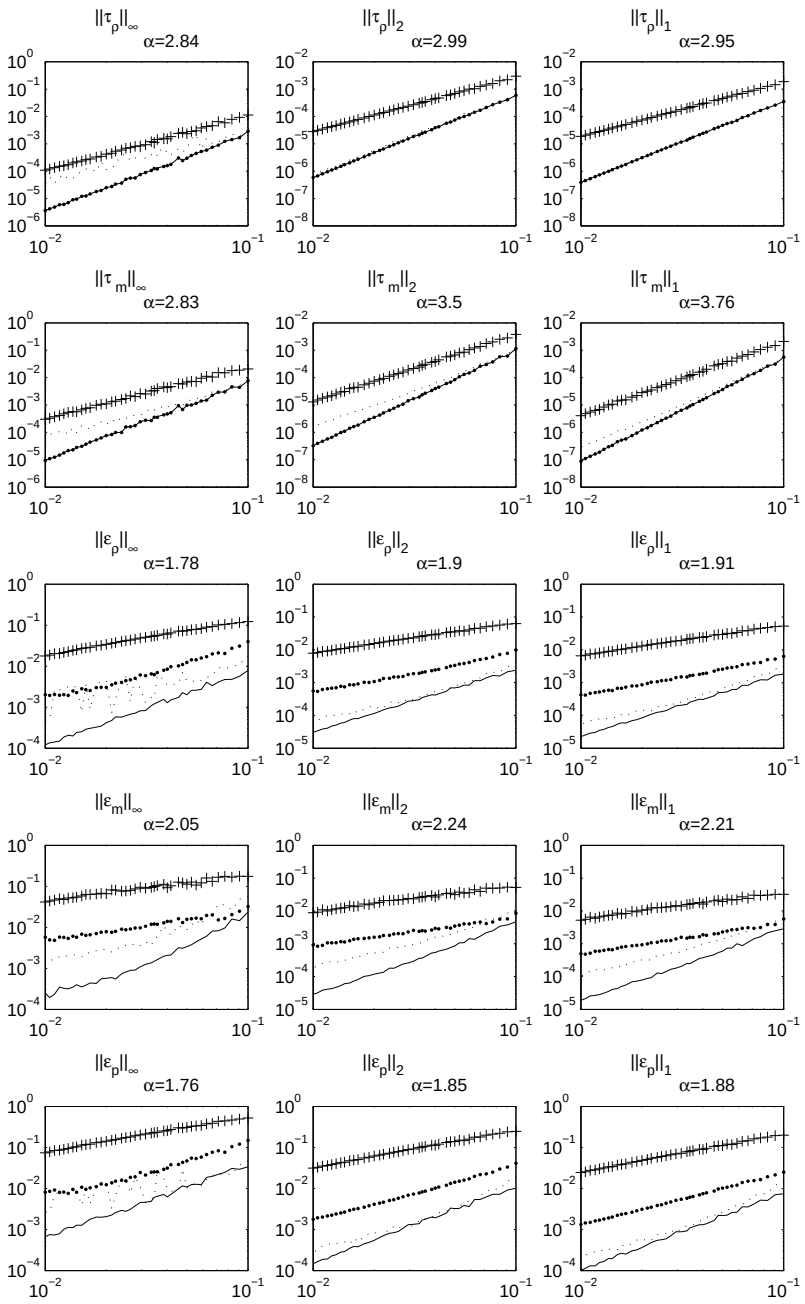

Figure 5.39: Ringleb flow, regular grids; norms of local and global truncation errors. "+": first order scheme; "▪": vertex-based reconstruction of all convected variables, no limiter; "-": vertex-based reconstruction of the density, cell-based reconstruction of the pressure, no limiter; ".": vertex-based reconstruction of the density, cell-based reconstruction of the pressure, unrelaxed Venkatakrishnan limiter. The estimate of the rate of convergence α refers to the results marked "-".

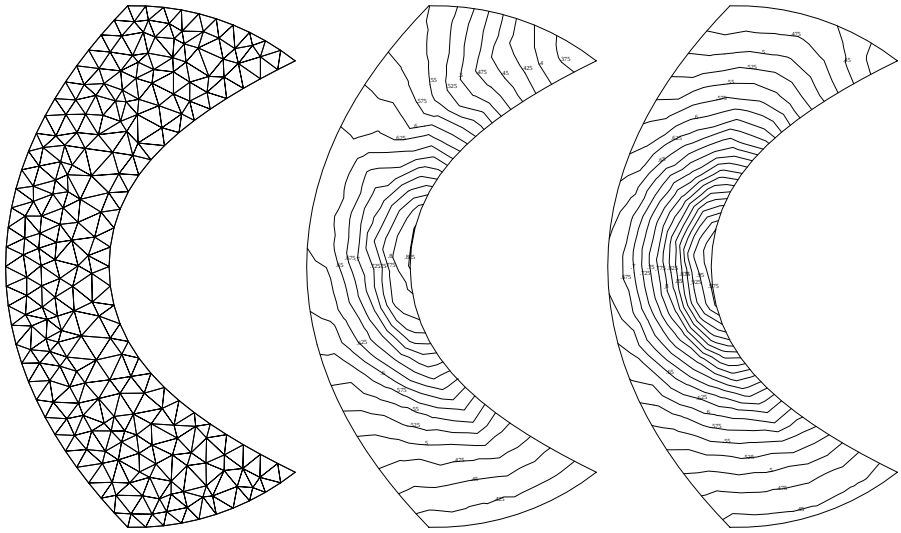

Figure 5.40: Ringleb flow, distorted mesh (left) and Mach isolines obtained with the first order (middle) and the second order unlimited (right) method.

Incompressible flow case

For M = 0 we want to demonstrate that the solution obtained with the Mach-uniform method matches the solution obtained with the incompressible method. The mesh is shown in Fig. 5.42. The isobars and the pressure distribution along the lower wall obtained with the method for incompressible flow and with the Mach-uniform method are shown in Fig. 5.43. There is no visible difference between these two solutions. Actual differences are of order 10^{-5}, which we consider reasonable since we are comparing results of different codes.

Supersonic flow case

The mesh used in this case is shown in Fig. 5.44. The inflow Mach number is 1.65. The Mach number obtained by the first order scheme has an oscillation before the first shock on the lower wall of about 0.6% of the jump in Mach.

If the vertex-based reconstruction is used for the density equation and the cell-based reconstruction is used for the pressure equation, and limiting is applied, the oscillations are not much larger than in the first order case, and the shock resolution is better. This especially holds for the reflected shock, which is completely blurred in the case of the first order scheme. If the cell based reconstruction is used for the density, the scheme does not converge in time. The results obtained are in good agreement with [10]. The obtained solution is less accurate at the outflow boundary than in the interior (see the left side of Fig.5.46), but this is also visible in [10].

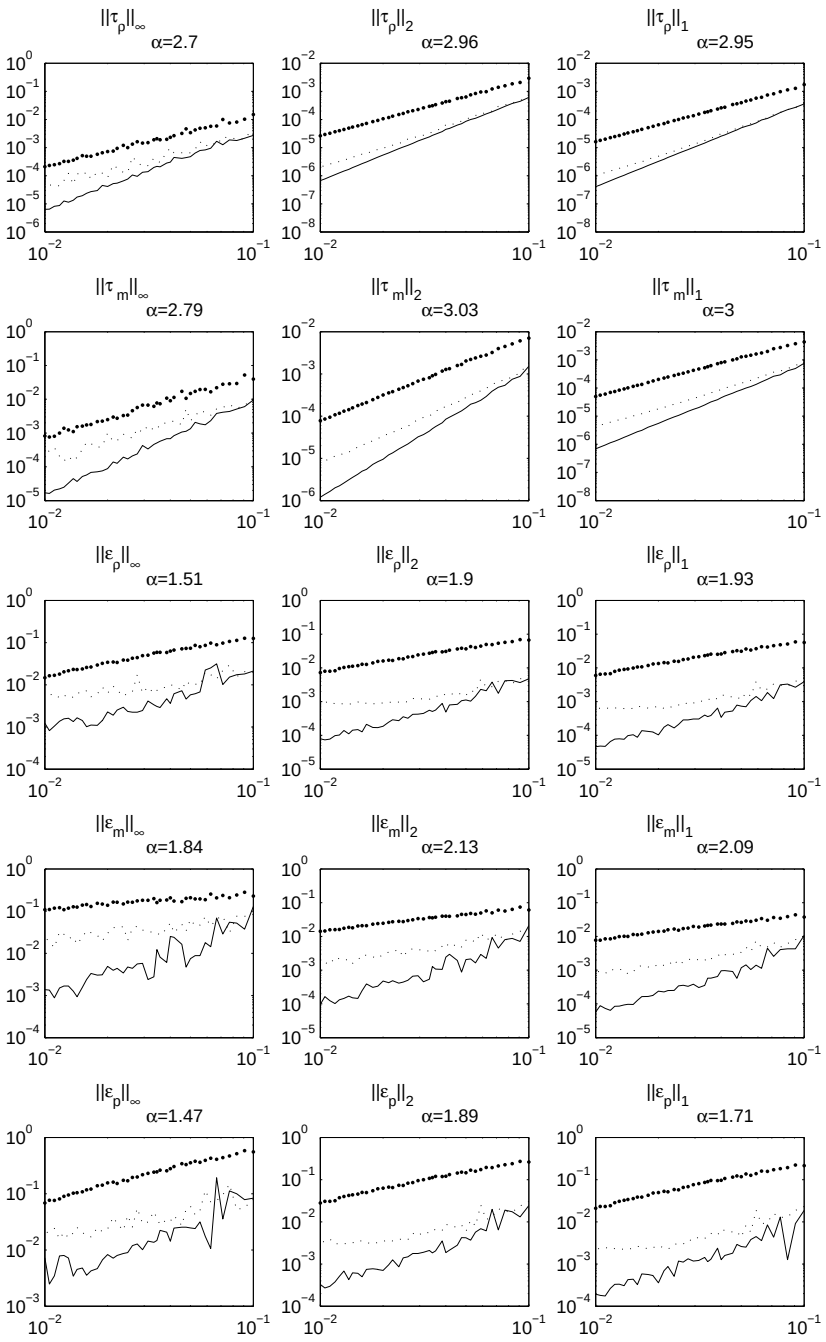

Figure 5.41: Ringleb flow, distorted grids. The errors of solutions. "▪": first order scheme; "-": vertex-based reconstruction of the density, cell-based reconstruction of the pressure, no limiter; ".": vertex-based reconstruction of the density, cell-based reconstruction of the pressure, unrelaxed Venkatakrishnan limiter. The estimate of the rate of convergence α refers to the results marked "-".

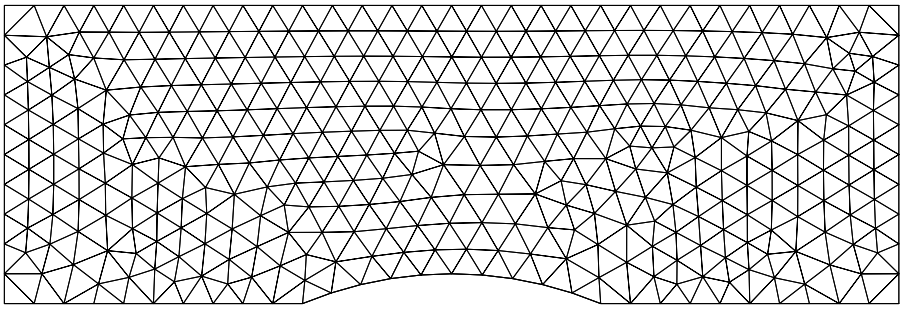

Figure 5.42: Channel with a bump, mesh used in the incompressible flow case.

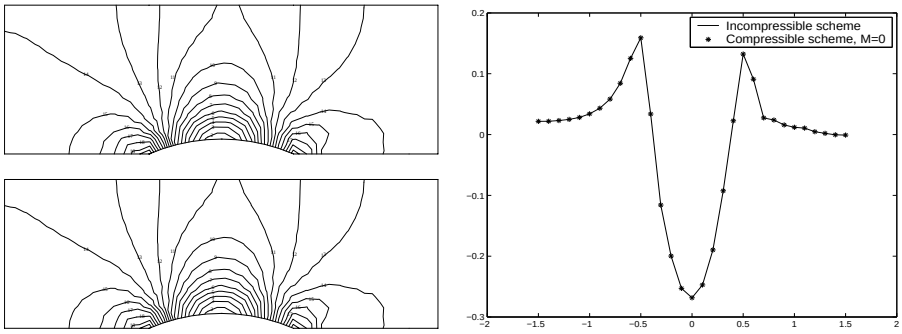

Figure 5.43: Channel with a bump, incompressible flow case. Pressure isolines obtained with the incompressible flow (left upper) and the Mach-uniform (left lower) method, and the pressure along the lower wall.

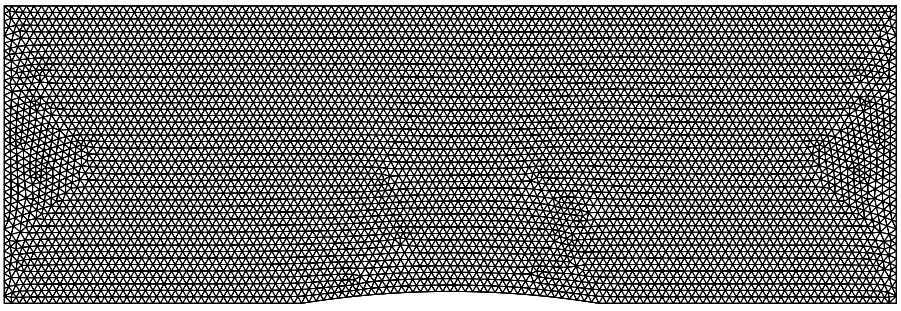

Figure 5.44: Channel with a bump, mesh used in the supersonic flow case.

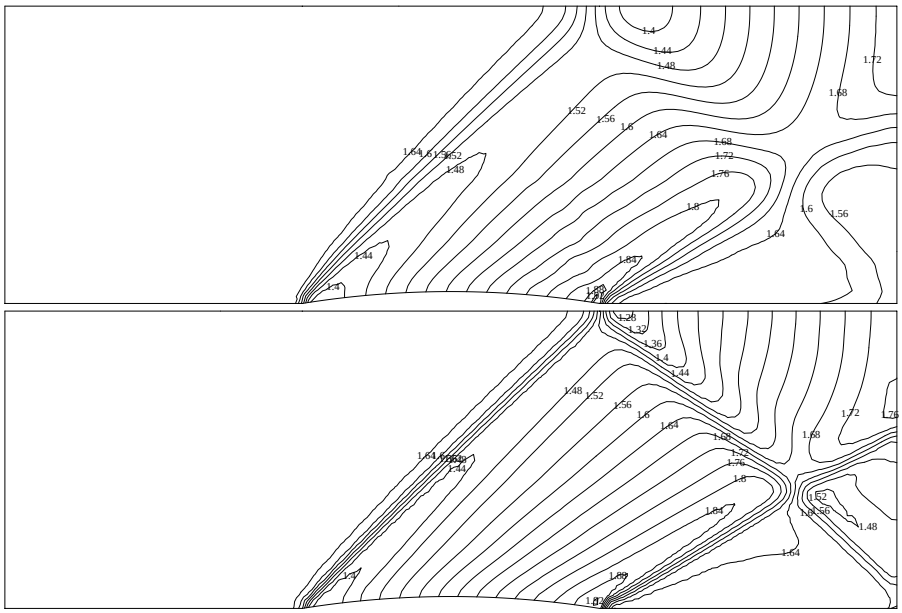

Figure 5.45: Channel with a bump, supersonic flow case. Mach number isolines. First order scheme (top) and the scheme with linear reconstruction (bottom). Vertex-based reconstruction was applied in the density and in the momentum equation, and the cell based reconstruction was used in the energy equation.

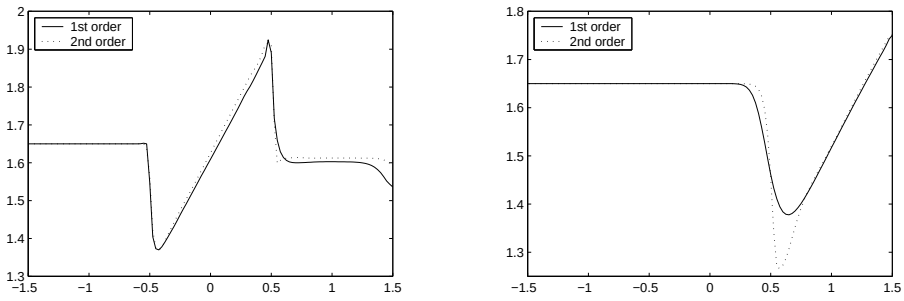

Figure 5.46: Channel with a bump, supersonic flow case. Mach number distribution along the lower (left) and the upper wall (right).

5.4.3 NACA 0012 airfoil

We have computed the flow around the NACA 0012 airfoil for M = 0.8 and angle of attack $\alpha = 1.25°$. This transonic benchmark problem was presented in [40]. Mach isolines obtained with the first order scheme and the superlinear

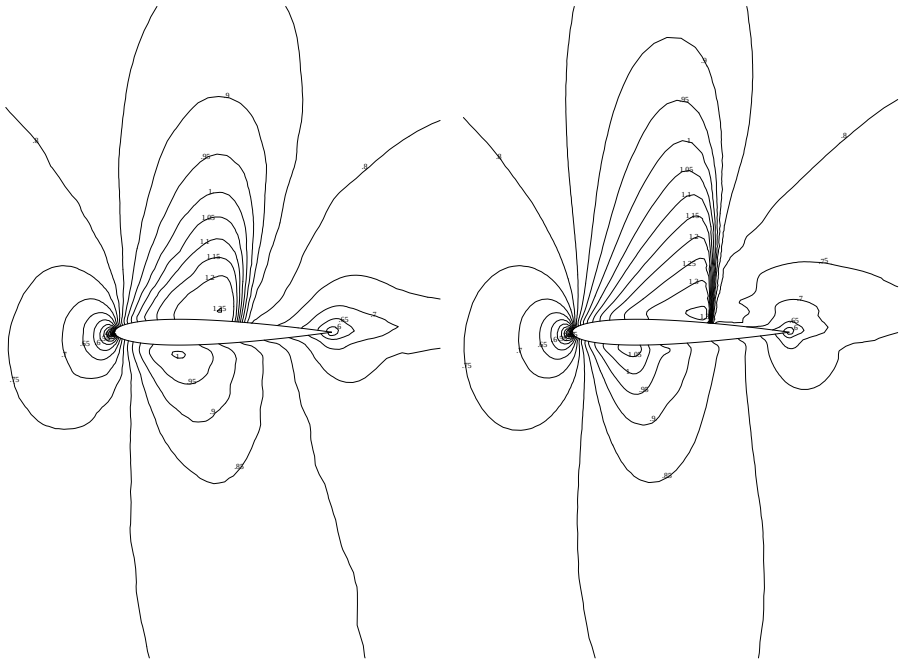

Figure 5.47: NACA 0012 airfoil, the Mach isolines obtained with the first order scheme (left) and with the superlinear scheme (right).

scheme on the grid shown in Fig. 5.48 is shown in Fig. 5.47. Vertex-based scalar reconstruction was used in the density equation, and cell-based scalar reconstruction was used in the Mach-uniform pressure-correction equation. As expected, the shock on the upper surface of the airfoil is sharper in the case of the superlinear scheme.

The Mach distribution along the airfoil is shown in Fig. 5.48 on the right, in comparison with the AGARD benchmark results published in [40]. Parameter n is the number of grid points along the upper surface of the airfoil. The weak shock on the lower surface of the airfoil is particularly hard to capture. The first order scheme does not see it. But the second order scheme gives an indication of this shock even for a twice less fine grid than the one shown in Fig. 5.48.

5.4.4 Sod's shocktube problem

In order to verify that our numerical method yields the correct entropy solution and that Rankine-Hugoniot conditions are satisfied, we solved Sod's shocktube problem. The domain is a channel of length 1 and width 0.16. The unstructured mesh that we used is shown in Fig. 5.49. The initial velocity is zero. In the left half of the domain the initial pressure is 1 and the initial density is also 1.

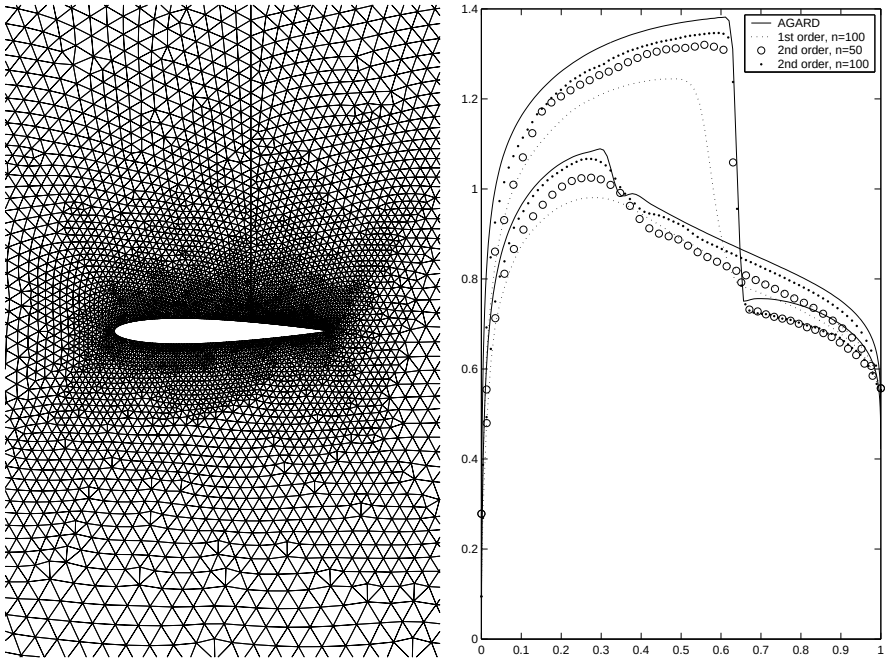

Figure 5.48: NACA 0012 airfoil, the finest mesh used, and the Mach distribution along the airfoil.

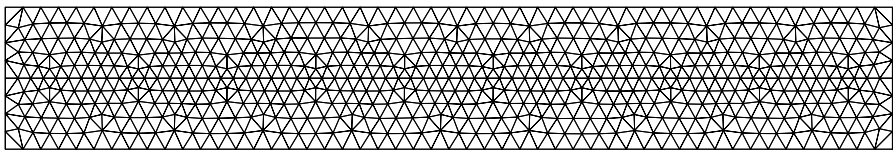

Figure 5.49: The mesh used with Sod's shocktube problem.

In the right half the initial pressure is 0.1 and the initial density is 0.125. The results obtained along the central horizontal line at time 0.15 are shown in Fig. 5.50 in comparison with the exact solution. We see that the the discontinuities

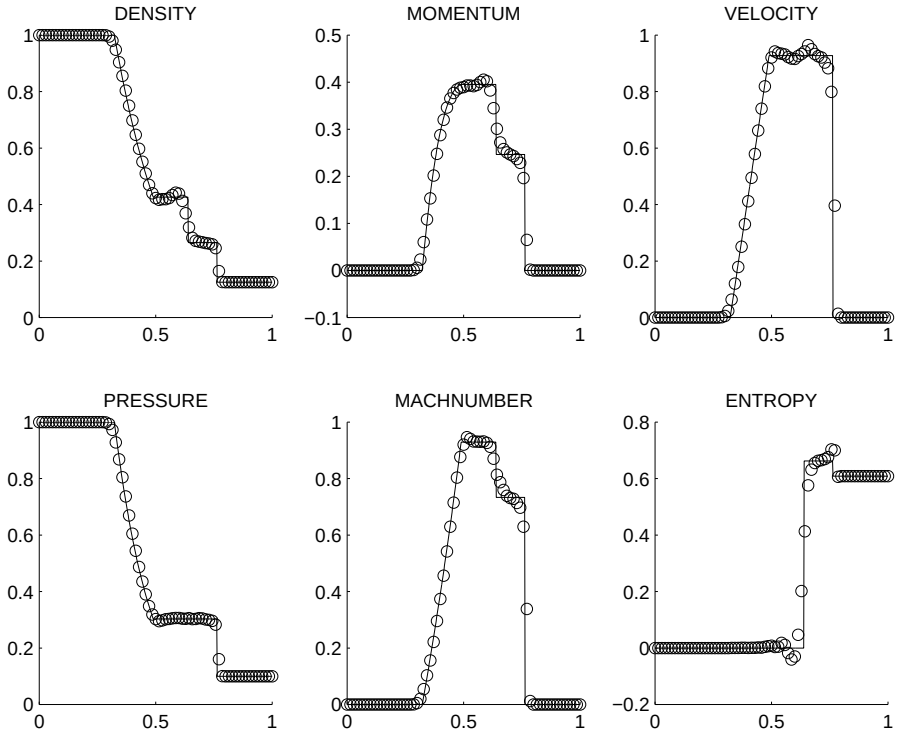

Figure 5.50: Sod's shocktube problem. Comparison of the numerical results with the exact solution at $t = 0.15$.

move at the correct speed and that the entropy solution was obtained. For a detailed explanation of this problem and the exact solution see [38].

Chapter 6

Conclusion

6.1 Possible extensions

6.1.1 Momentum reconstruction on three-dimensional tetrahedral grids

In three dimensions we approximate the momentum field $\mathbf{m}$ in the vicinity of a mesh vertex by a linear polynomial equivalent to (4.25):

$$\mathbf{m}(\mathbf{r}) \approx \mathbf{P}(\mathbf{r}) = \mathbf{a} + \mathbf{b}x + \mathbf{c}y + \mathbf{d}z, \qquad \mathbf{a}.\mathbf{b}, \mathbf{c}, \mathbf{d} \in \mathbb{R}^3. \tag{6.1}$$

This polynomial can be determined from the linear system

$$\mathbf{N}_e \cdot \mathbf{P}(\mathbf{r}_e) = m_e, \qquad \text{for each face } e \text{ in the reconstruction stencil.} \tag{6.2}$$

The term *faces* refers to the two-dimensional surfaces of the cells (triangles), and *edges* are line segments connecting the vertices. Vector $\mathbf{N}_e$ is one of the two possible unit normal vectors in face e, and normal momentum m_e is actually the average of the momentum component $\mathbf{N}_e \cdot \mathbf{m}$ over triangle e.

In the same manner as in the one-dimensional case, we integrate the divergence of this polynomial over some tetrahedron Θ and apply the divergence theorem:

$$b_1 + c_2 + d_3 = \nabla \cdot \mathbf{P} = \frac{1}{|\Theta|} \int \nabla \cdot \mathbf{P} d\Theta = \frac{1}{|\Theta|} \sum_e m_e |\Omega_e|, \tag{6.3}$$

where Ω_e are the faces of the tetrahedron. If the momentum is prescribed in all faces of one tetrahedron, the divergence of the linear polynomial is determined. Therefore, if the faces of more than one tetrahedron are included in the reconstruction stencil, the rank of the system (6.2) may be reduced.

To prevent the rank reduction due to the divergence theorem we base the polynomial in a vertex and try to include only the faces that meet in this vertex in the reconstruction stencil. We reconstruct only the divergence-free part $\overline{\mathbf{P}}$ of the linear polynomial $\mathbf{P}$:

$$\overline{P}_1 = P_1, \quad \overline{P}_2 = P_2, \quad \overline{P}_3 = P_3 + (d - b_1 - c_2 - d_3)z. \tag{6.4}$$

There are only 11 free parameters left.

In the interior points of typical tetrahedral meshes the number of faces meeting in a vertex is usually around 30. This is clearly more than enough to determine the divergence free polynomial. Of course, as in the two-dimensional case, additional rank reduction may occur.

6.1.2 Non-triangular and non-tetrahedral grids

It is highly desirable that a divergence-free polynomial can be determined from the faces meeting in one vertex only. Since only four faces meet in each internal vertex of a quadrilateral grid, they do not determine a divergence-free linear polynomial. For the application of our scheme it is therefore advisable to convert a quadrilateral or a mixed-element grid into a triangular grid by subdividing each non-triangular polygon into triangles.

There are 12 faces that meet in each internal vertex of a three-dimensional rectangular grid. They all belong to three mutually orthogonal planes, four faces in each plane. Restriction of a normal component of a linear polynomial to one of these planes is a scalar linear function of a two-dimensional argument, and therefore it has three free parameters. Since there are four conditions imposed to this linear function, one for each face, equations for these four faces must be linearly dependent. This means that the rank of the related linear system will be at most 9, and a divergence-free linear polynomial cannot be determined. As in the two-dimensional case, to avoid stencil enlargement it is best to divide non-tetrahedral polyhedra into tetrahedra.

6.2 Closing remarks

We have elucidated difficulties that arise from polynomial reconstruction of staggered representations of vector fields, and proposed a divergence-free linear reconstruction method that is robust and accurate.

Staggered schemes on unstructured grids for incompressible, compressible, and Mach-uniform flows with superlinear convergence have been presented. These schemes employ the proposed reconstruction technique. Our reconstruction method can also be used to improve the order of accuracy of other staggered schemes, which, as noted in [27], is at present generally first order.

Bibliography

[1] 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.

[2] T. J. Barth. *Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations.* 25th Computational Fluid Dynamics Lecture Series, Von Karman Institute, 1994.

[3] T.J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. AIAA Paper 93-0668, 1993.

[4] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366, 1989.

[5] G.K. Batchelor. *An Introduction to Fluid Dynamics.* Cambridge University Press, Cambridge, UK, 1967.

[6] H. Bijl. *Computation of flow at all speeds with a staggered scheme.* PhD thesis, Delft University of Technology, The Netherlands, February 1999.

[7] 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.

[8] G. Chiocchia. Exact solutions to transonic and supersonic flows. AGARD advisory report AR-211, AGARD, Neuilly-sur-Seine, France, 1985.

[9] M. Delanaye. *Polynomial reconstruction finite volume schemes for the compressible Euler and Navier-Stokes equations on unstructured adaptive grids.* PhD thesis, University of Liege, Belgium, September 1996.

[10] S. Eidelman, P. Colella, and R.P. Shreeve. Application of the Godunov method and its second-order extension to cascade flow modelling. *AIAA J.*, 22:1609–1615, 1984.

[11] 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.

[12] 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).

[13] 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.

[14] 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.

[15] F.H. Harlow and A.A. Amsden. Numerical calculation of almost incompressible flows. *J. Comp. Phys.*, 3:80–93, 1968.

[16] F.H. Harlow and A.A. Amsden. A numerical fluid dynamics calculation method for all flow speeds. *J. Comp. Phys.*, 8:197–213, 1971.

[17] 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.

[18] Stephen Hawking. *A Brief History of Time*. Bantam Books, New York, 1988.

[19] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, 59:308–323, 1985.

[20] L. Kovasznay. Laminar flow behind a two-dimensional grid. Proc. Cambridge Philos. Soc. 44, 1948.

[21] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37:141–158, 1981.

[22] P. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure and Appl. Math.*, 13:217–237, 1960.

[23] I. Lepot. *A parallel high-order implicit finite volume method for three-dimensional inviscid compressible flows on deforming unstructured meshes*. PhD thesis, University of Liege, Belgium, January 2004.

[24] T. Liszka, C. Duarte, and W. Tworzydlo. hp-Meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139:263–288, 1996.

[25] 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.

[26] 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.

[27] B. Perot. Conservation properties of unstructured staggered mesh schemes. *J. Comp. Phys.*, 159:58–89, 2000.

[28] 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.

[29] F. Ringleb. Exakte Lösungen der Differentialgleichungen einer adiabatischen Gasströmung. *ZAAM*, 20:185–198, 1940.

[30] 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.

[31] 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.

[32] 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*, 40:521–529, 2002.

[33] V. Venkatakrishnan. Perspective on unstructured grid flow solvers. *AIAA Journal*, 34:533–547, 1996.

[34] D. Vidović, A. Segal, and P. Wesseling. A superlinearly convergent finite volume method for the incompressible Navier-Stokes equations on staggered unstructured grids. *Journal of Computational Physics*, 198:159–177, 2004.

[35] D. Vidović, A. Segal, and P. Wesseling. A superlinearly convergent Mach uniform finite volume method for the Euler equations on staggered unstructured grids. *Journal of Computational Physics*, 2005. Submitted.

[36] I. Wenneker, A. Segal, and P. Wesseling. A Mach-uniform unstructured staggered grid method. *International Journal for Numerical Methods in Fluids*, 40:1209–1235, 2002.

[37] I. Wenneker, A. Segal, and P. Wesseling. Conservation properties of a new unstructured staggered scheme. *Computers and Fluids*, 32:139–147, 2003.

[38] P. Wesseling. *Principles of Computational Fluid Dynamics*. Springer, Heidelberg, 2001.

[39] C.H.K. Williamson and G.L. Brown. A series in $1/\sqrt{Re}$ to represent the Strouhal-Reynolds number relationship of the cylinder wake. *Journal of Fluids and Structures*, 12:1073–1085, 1998.

[40] H. Yoshihara and P. Sacher. Test cases for inviscid flow field methods. AGARDograph No. 211, AGARD, Neuilly-sur-Seine, France, 1985.

Curriculum vitae

- 2000-present: Delft University of Technology, The Netherlands, PhD student in Numerical Analysis, supported by the Netherlands Technology Foundation (STW), subject: *Unified computing methods for compressible and incompressible flows on staggered unstructured grids*, primary adviser: Prof.dr.ir. P. Wesseling, Department of Applied Mathematical Analysis, Faculty of Electrical Engineering, Mathematics and Computer Science
- 1999-2000: Delft University of Technology, The Netherlands, research project in Neural Networks, subject: *Rule extraction from a trained neural network and inverting the network reasoning function*, primary adviser: Dr. L. Rothkrantz, Group of Knowledge Based Systems, Faculty of Electrical Engineering, Mathematics and Computer Science
- 1995-1999: Belgrade University, Serbia, research project in Numerical Analysis, subject: *A boundary value problem with irregular boundary conditions*, primary advisor: L. Ivanović, Group of Numerical Analysis and Optimization, Mathematical Faculty, teaching assistant at the Group of Technical Mathematics, Faculty of Technology and Metallurgy.
- 1988-1994: Belgrade University, Serbia, BSc in Numerical Mathematics and Optimization, graduation project: FORTRAN Implementation of Revised Simplex Method of Linear Programming.
- 1984-1987: Mathematical Gymnasium, Belgrade, Serbia.
- Born on 28-01-1969 in Banja Luka, Bosnia.

List of publications

- D. Vidović, A. Segal and P. Wesseling A Superlinearly Convergent Mach Uniform Finite Volume Method for the Euler Equations on Staggered Unstructured Grids. *Journal of Computational Physics*, 2005. Submitted.
- D. Vidović, A. Segal and P. Wesseling A Superlinearly Convergent Finite Volume Method for the Incompressible Navier-Stokes Equations on Staggered Unstructured Grids. *Journal of Computational Physics* 198:159-177, 2004.
- D. Vidović, A boundary value problem with irregular boundary conditions. Paper 421 in Graham de Vahl Davis and Eddie Leonardi (eds.): CHT'01: Advances in Computational Heat Transfer II. Proceedings of a Symposium organized by the International Centre for Heat and Mass Transfer IHMT. 20-25 May 2001, Palm Cove, Queensland, Australia. Begell House Inc., New York. CDROM, ISBN 1-56700-158-0
- D. Vidović and M. Todorović, Efficient Portfolio: Analytic Solution and Economical Meaning. Proceedings of the International Symposium On Operational Research, Belgrade, Serbia, 1999.

Frontispiece: *An unstructured grid.* Photograph by D. Vidović of Duisburg Meiderich iron mill, Germany.