ON TRIANGLES AND FLOW

by

Jens-Dominik Müller

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
1996

Doctoral Committee:

Professor Philip L. Roe, Chairman
Professor Herman Deconinck
Professor Gregory M. Hulbert
Professor Kenneth G. Powell
Professor Bram Van Leer

Preface

"I call our world Flatland, not because we call it so, but to make its nature clearer to you, my happy readers, who are privileged to live in Space.

Imagine a sheet of paper on which straight Lines, Triangles, Squares, Pentagons, Hexagons, and other figures, instead of remainaing fixed in their places, move freely about, on or in the surface, but without the power of rising above or sinking below it, very much like shadows – only hard and with luminous edges – and you will then have a pretty good notion of my country and countrymen. Alas, a few years ago, I should have said "my universe": but now my mind has been opened to higher views of things."

E. A. Abbott, 'Flatland. A Romance of Many Dimensions'

Acknowledgements

Acknowledgements written in a thesis front-page are useless. Nobody reads theses anyways. And those[1] I feel indebted towards know it anyhow. I presume.

[1] Die allermeisten Danksagungen beginnen mit “Zuförderst möchte ich meinen Eltern danken, die ... ”. And then continue with “many thanks also to my friends ... , sans votre soutien ... ”.

Natürlich bin ich meinen Eltern dankbar, ohne sie wäre diese Arbeit nicht geschrieben worden, ohne sie wäre ich weder hier, noch irgendwo. Aber wenn ich ihnen das auf dieser Seite sagen muß, haben sie und ich versagt. Sparen wir's uns.

Et je ne voudrais pas me passer du soutien de la part de Patricia, Kevin, Stella, James, Carrie, Bill and Margaret. Their friendship and support did make a difference. Mais ils ne vont pas lire ces pages, so it is futile to thank them here, and a bottle of Bordeaux is more appropriate.

The special environment at both the von Karman Institute in Rhode-St.-Genèse and the Keck Laboratory for CFD at the University of Michigan does deserve credit in the progress of this thesis. Both places provided me with, apart from bed and bread (which is to be taken literally as well), inspiration and vision. The peers I had at both places, Henri, Robert, Sami, Lisa and Greg, contributed to this with their ideas and criticism. J'ai eu beaucoup de chance de pouvoir combiner le meilleur de l'ancien et du nouveau monde, et cela valait la peine du trajet. But, as my many returns indicate, I never stopped praising both places, so this one here should be left off.

I am also grateful to Prof. Greg Hulbert for showing interest in my work and being on my committee. He is one of the few people who have actually read it before the defense! Prof. Bram Van Leer had his door always ajar for my questions, and I appreciated that very much for soliciting a second opinion at times. Since I am fortunate enough to have both of them on my committee, I can thank them in person.

Ken Powell was instrumental in establishing the collaboration between the CFD-group at VKI and the Keck laboratory, a most successful one by all standards, and helped me come to the Keck. His support started with offering me his living-room couch and never ceased. I do hope though that I can make up for that other than just acknowledging it.

There are two mentors, however, who would make it worthwile to write an acknowledgement page. Without the advice and guidance of Herman Deconinck and Phil Roe, this thesis would not have been written. I have seen and heard of many thesis-supervisors, for whom a doctoral candidate is a cheap laborer rather than a student. Phil and Herman were true advisors in the sense that they generously gave me the space to pursue broad interests in my personal education, explore my own projects, make my own mistakes, and take credit for my own ideas. They did rescue my boat when it was on the brink of being steered into a reef, but they respected my seamanship and trusted me to set my own course when the waters were deep. I will remain indebted to both of you. But, having given up poetry at the age of 18, I am uttely unable to word this. And, besides, I think, both of you know.

Table of Contents

"My point and period will be thoroughly wrought
or well or ill, as this day's battle's fought."

W. Shakespeare, 'King Lear', Act V, Sc. 1

Chapter

List of Figures

Chapter I

Introduction

> “ ‘Oh if I only could write!’ she cried (for she had the odd conceit of those who write that words written are shared)”
>
> V. Woolf, ‘Orlando’.

Computational Fluid Dynamics (CFD) has seen tremendous growth in the past three decades. This is most fundamentally due to the wide variety of phenomena important to our life that involve fluid dynamics, ranging from thunderstorms to blood circulation. The desire to accurately simulate and understand these phenomena is met by a continuing growth of the computer hardware. It is this growth that allows to think about software that might someday be able to simulate the thunderstorm in time and space with all the turbulent growth and decay in it, or the flow of blood around an artificial heart-valve that is opening and closing periodically.

1.1 Treatment of the Multi-Dimensional Euler Equations

Today, CFD is still ‘a couple of breakthroughs’ away from being able to handle these problems. Compared to structural mechanics, where the range of problems of interest is similarly broad, CFD is far from having finished maturing. This is mostly due to the complex mathematical structure of the Euler- and Navier-Stokes equations that are used to describe the motion of the fluid. In the sub-critical steady state these equations are mixed hyperbolic and elliptic in character. Elliptic partial differential equations, that in the Euler equations represent the acoustic propagation in all directions, have a smoothing effect on the field, as can be seen from the behavior of the Cauchy-Riemann equations. The hyperbolic partial differential equations on the other hand allow for the advection and creation of discontinuities and extrema in the solution, as can be seen with Burgers' equation. While excellent algorithms exist when dealing with each type of equation separately, no similarly satisfactory methods exist for the mixed problem.

Pseudo-time-stepping of the unsteady equations has become the method of choice for the time discretization, even when the steady-state solution is sought. Since the unsteady Euler-equations are hyperbolic in the subsonic and supersonic regime, pseudo time-stepping allows to treat the equations as one purely hyperbolic system. Naturally there is a price: the convergence rates decay for low-speed problems or for problems where stagnation regions are an important feature, as the elliptic character of the equations becomes dominant.

In the past decade, algorithms for high-speed flows have seen significant progress, partly due to the funding situation, partly due to the enormous progress brought to CFD by the invention of Approximate Riemann solvers [1–3] used in Godunov-type schemes [4] in the Finite-Volume context. In Finite-Volume schemes the data are constant or vary linearly around the cell average over each cell, thus the data jump at the cell interfaces. In one dimension, approximating the Riemann problem at the discontinuity leads to a diagonalization of the system of equations and scalar advection schemes with an optimal control of the numerical dissipation can be applied. It seemed that the solution of convection dominated flows with highly discontinuous fields is stabilized by the introduction of this discontinuity in the data representation.

A dimension-by-dimension extension of the one-dimensional concepts that considers a set of only one-dimensional Riemann problems normal to cell-interfaces proved highly successful and more robust than any other scheme used for high-speed flows. It was this robustness that made CFD developers dream about automated flow codes that need little user-intervention and that ultimately led to the modern commercial adaptive flow-codes that run even on desktop PCs.

The continuous growth of memory and speed of the hardware have changed CFD in the past years in two major ways: formerly subscale phenomena can now be incorporated in finer meshes and need to be resolved properly, and 'fluid-affected' other than the military and a few aerospace vendors can now afford the necessary number-crunching power to apply CFD to their problems. And these problems turn out to be different.

1.1.1 Rotating the Riemann Solver

It has become clear that for a proper resolution of flow phenomena that are dominated by viscous effects, the numerical dissipation of the Upwind schemes is still too high. As such problems, blunt body separation or intake-manifold turbulence in the automobile industry or the total pressure loss around a turbine blade come to mind. The simple one-dimensional physics finally caught up with the numerical reality. E.g a shear flow oblique to the grid lines, as might be encountered along the streamline leaving a separation point, is interpreted from these schemes as a grid-aligned shear that is preserved and a grid-normal compression to which additional but unnecessary numerical dissipation is applied.

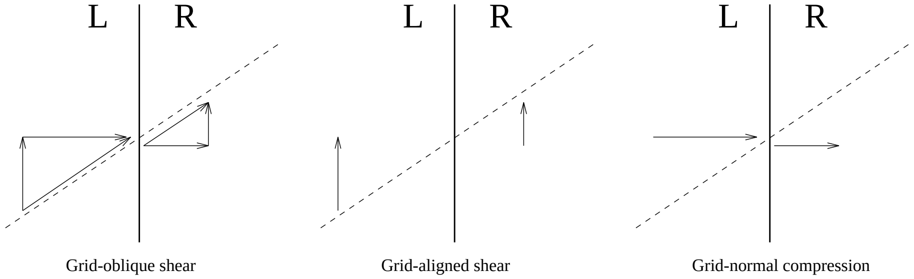


Figure 1.1: Applying one-dimensional concepts to higher-dimensional problems decomposes a grid-oblique shear into a grid-aligned shear and a grid-normal compression.

Numerous attempts have been undertaken to make the solution less dependent on the grid-orientation. In a class of schemes, termed 'rotated Riemann solvers' [5–7], the one-dimensional Riemann problem is solved in a direction that is aligned with the streamlines or the pressure gradient. However, this does not lead to convincing improvements for second order accurate solutions and entails problems of robustness and monotonicity. Clearly, while it is possible to align the 1-D problem with one direction of choice to capture either shears or shocks, it is impossible to deduce a multi-dimensional interpretation of the occurring flow phenomenon from only two states.

This conjecture is made in a more sophisticated fashion in a class of wave-modeling Finite-Volume schemes [8–10]. The fluxes at the interfaces are calculated assuming a two-dimensional wave-pattern of more than three waves. The resulting flux formulae bear a striking similarity to the characteristic flux extrapolation method of [11] that makes the dissipation dependent on an approximate diagonalization of the equations or the multidimensional limiting of Sidilkover [12]. While these schemes show some improvement in the recognition of all flow phenomena, they do not improve in general over second order accurate dimensionally split schemes and, except for the two latter methods, can certainly not rival their robustness.

Multidimensional state vector splitting [13] also makes use of the superposition of a discrete number of waves, with the resulting elementary waves being integrated over all possible directions in [14]. The subsonic performance of these schemes is still unclear, however. Another modification of the Godunov type flux calculation is the genuinely multidimensional Riemann solver [15], where the exact solution to the two-dimensional Riemann problem is computed for fluxes across the interfaces. Not too surprisingly a very costly procedure.

1.1.2 Converging to the Steady State

Another problem of hyperbolic Finite-Volume methods that has become more pressing on larger grids is the poor convergence for low-speed flows. The stability of explicit methods for the unsteady Euler equations depends on the largest wave-speed, $u + c$, the sum of the mean flow speed and the speed of sound. However, some of the information of the flowfield is carried with the slower speeds, u and $u - c$, which vanish in the stagnation point and the sonic point, respectively, thus resulting in a 'stiff' problem. Even with preconditioning and Multigrid techniques, solutions for low-speed flows using compressible Finite-Volume methods can only be obtained at costs far superior to the one of methods designed specifically for the steady incompressible Euler and Navier-Stokes equations.

1.1.3 The Incompressible Approach

One such method, the Finite-Element method that was developed in structural mechanics, has brought a new level of robustness and accuracy to the computation of low-speed flows. The variational formulation yields a rigorous error control for elliptic problems that has been a breakthrough for adaptive mesh refinement schemes. For convection-dominated problems, these schemes are augmented by various forms of streamwise diffusion terms [16, 17] in order to improve their stability for highly discontinuous solutions. These improvements are impressive when applied to scalar equations, and not too surprisingly these SUPG schemes exhibit a close relationship to Upwind schemes when applied to scalar equations. However, their data representation is profoundly different from Finite-Volume schemes: the data are stored at the vertices of the grid and assumed to vary linearly over each cell. Thus, there are no discontinuities in the data.

1.1.4 Fluctuation-Splitting: the Dual View

The dual view of data representation taken by the Fluctuation-Splitting approach [18] brings a new level of unification to the theory of 'Mixed Elements' [19–21]. Fluctuation-Splitting uses a Finite-Element data representation with unknowns at the vertices and a solution that varies linearly over each cell. With the continuous data, the interface Riemann-problem becomes obsolete and it is easier to formulate genuinely multidimensional schemes that depend on physically relevant directions rather than the orientation of the discontinuities at the cell interfaces. Alternatively, the scheme can be viewed as a Finite-Volume scheme that collects fluxes around the edges of the median dual of each vertex. This point of view simplifies the design of schemes that are monotonic, i.e. schemes that do not create overshoots. Many such Fluctuation-Splitting schemes have been found for scalar advection [22, 23] and their design principles are now well understood.

The missing piece is a multidimensional splitting of the flux balance into scalar components that can then be treated well with the multidimensional scalar schemes. Early attempts using simple wave models [24] produced results similar to those of other multidimensional methods: a gain in accuracy paid for by a loss in robustness. It seemed that it was unsolvably hard to choose from the infinite number of propagation directions in the subsonic case.

1.1.5 Splitting the System of Euler Equations

Inspired by the treatment with SUPG schemes of the entire system of equations rather than several scalar components [25], by insights gained from work on preconditioning on the Euler equations [26] and by a re-evaluation of the residual distribution methods by Ni and Hall [27, 28], a new approach has been suggested for the steady Euler equations by Ta'asan [29] and Roe [30] that splits the system in a hyperbolic part with decoupled scalar advection equations that are best discretized with scalar advection schemes and an elliptic part that is treated as a coupled system.

The approach has been successfully demonstrated recently [31–34], achieving very high accuracy with good robustness. Remaining issues are the treatment of the stagnation point where the preconditioning becomes singular and problems with the orthogonality of the eigenvectors arise [35] as well as the surprisingly disappointing rate of convergence of the scheme. It should be expected that the stiffness of the equations is completely removed by the approach, however, it appears to converge significantly more slowly than dimensionally split Finite-Volume schemes. Unfortunately, this phenomenon is not yet understood.

1.2 Unstructured Grid Generation

Unstructured mesh methods are about to become the workhorse for CFD-calculations. Although major automation efforts are still undertaken to reduce the user-time involved when generating multiblocked structured meshes around complex geometries, the user community has voted in favor of unstructured methods for low to moderate Reynolds number flows and the leading software houses account for that.

The flexibility of unstructured grids dramatically reduces the time to generate a computational mesh around a complex geometry from man-months for a structured multiblocked grid to CPU minutes for a tetrahedral unstructured grid. Besides the gains during the mesh generation, even more significant gains during the calculation of the solution are offered by solution-adaptive grid refinement. While adaptivity is intrinsic to the concept of an unstructured mesh, it is a rather cumbersome procedure on structured meshes. An extensive documentation of the state of the art as of 1993 can be found in [36].

1.2.1 Hierarchical Grid Methods

Hierarchical methods based on quad-trees in two dimensions and on octrees in three dimensions are a more recent innovation in grid technology for CFD. They were initially developed for grid generation and simulation in structural mechanics [37] and recently gained interest from the CFD community [38–41]. The interesting idea here is to use the domain partitioning of a quadtree as a computational grid and to use the underlying tree data-structure for communication between the cells and for adaptivity. Moreover, the grid-lines are always orthogonal except for boundary intersections, leading to a favorable error cancellation, at the price of a very rough tessellation of the boundaries. To circumvent the problem of 'cut-cells', the vertices[1] near the boundaries can be warped to form nicer cells and the entire tree can be triangulated, leading to relatively rough grids, however [37]. Alternatively, in order to remove the stiffness associated to the cut-cells, they can be treated specially on the solver level [38].

1.2.2 The Advancing Front Method

In the triangular Advancing Front method (AFM) [42,43], an initial list of frontal edges between boundary nodes is established that represents the boundaries of the domain. The smallest edge from all fronts is taken as the base of a triangle to be formed. A third vertex for a new triangle is either constructed or taken from the list of existing vertices in the front, in accordance with parameters interpolated on a background mesh that the user has to specify, such that the new triangle does not intersect any already existing ones. The front is then updated and the process repeated until all fronts have collapsed, leaving no gaps to be filled. The method is called being 'greedy', since any triangle formed is never removed from the triangulation and the domain is eaten up by the front.

The two-dimensional meshes created by the AFM exhibit a very high degree of regularity after smoothing with a Laplacian filter, although the nodes generated might not be connected in an optimal way. A connection that is optimal, in several senses, as will be shown in section 5.2, can be guaranteed by using a Delaunay triangulation. The AFM has seen widespread use in CFD, but more for a lack of alternative than for its relatively poor performance in terms of CPU time or for its poor robustness.

1.2.3 Delaunay Refinement Methods

A more mathematical concept is the Delaunay triangulation [44–46] that prescribes a unique connectivity between a given set of grid vertices: no circumcircle around three

[1] In the following the term vertex is adopted when referring to a grid-vertex. The distinction is made since a vertex does not necessarily coincide with a node where the solution is located.

vertices that form a triangle contains another vertex. This leads to various desirable optimal characteristics that are briefly reviewed in section 5.1 and more extensively in [47].

The circumcircle criterion can be efficiently used as a mechanism for constructing the Delaunay triangulation [48,49] and for improving it by recursively inserting new vertices into triangles with undesirable shape or size. The general idea shared by all 'Delaunay refinement' methods is to introduce a new vertex at the circumcenter of a non-desirable cell and to retriangulate. This placement guarantees that no vertex is closer than the circumradius of the bad cell from the new vertex. Once a suitable placement has been found in this way, the Delaunay criterion takes care of the optimal connectivity. As the new vertex is placed specifically within the circumcircle of a bad cell, this cell must be retriangulated and its minimum angle will double (fig. 1.2).

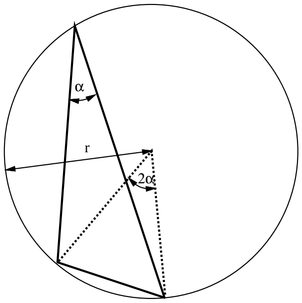


Figure 1.2: Acute cell (thick lines) with a circumcircle and cell with twice the minimum angle (dashed lines) that is created by the insertion of a new vertex at the circumcenter.

Variants of this algorithm are Holmes and Snyder's algorithm, [50], that targets unwanted cells on area and aspect ratio, and Chew's algorithm [51] that targets circumradius. Chew's algorithm actually yields bounds for minimum and maximum angles, provided that the boundary point spacing is within certain bounds, resulting in a triangulation with uniform size.

Ruppert presented a more sophisticated algorithm [52] that introduces a new vertex into the cell with the smallest angle α_{min} found in the triangulation. In the case that the introduction of this vertex is to break a required boundary segment, a new vertex is introduced midways on that segment. Ruppert can prove convergence of his algorithm if all input angles are larger than 90° for any $\alpha_{min} \leq 20°$ and finds convergence in practice for up to 30°.

All three Delaunay refinement methods are successful in achieving good angular properties in the meshes produced. However, they fail badly in regularity, as compared to the Advancing Front method. Also, the user has little control in selecting areas with high mesh density where significant flow features are expected.

1.3 Courses Set, Waters Charted

The following sections begin with a review of the material that is found at the starting point of this work.

Chapter II presents a discussion on conservative, primitive and symmetric forms of the Euler equations and their properties, leading to the preconditioned form of the Euler equations used for the calculations in this thesis that splits the equations into elliptic and hyperbolic parts. The similar splitting of Ta'asan [29] is presented for purposes of illustration, but not evaluated.

Chapter III reviews the various space discretizations pertinent to multi-dimensional advection, including the elliptic system and the time-discretizations employed.

Chapter IV derives the FAS-Multigrid scheme that has been implemented, including aspects of its application to hyperbolic equations and the choice of grid-transfer operators.

These discussions lead to the chapters V and VI where this thesis attempts to contribute.

Chapter V begins with a short description of the Delaunay method including a discussion on grid quality characteristics. This is followed by a detailed description of the novel Frontal Delaunay method. It includes the boundary point discretization, the generation of a background grid for mesh scale interpolation, the generation of the interior point cloud, the generation of coarse vertex-nested grids for Multigrid and estimations on the bounds of key grid-quality parameters.

Chapter VI presents a comprehensive convergence study with the Multigrid method applied to Fluctuation-Splitting discretizations of scalar hyperbolic equations, the Cauchy-Riemann equations and the Euler equations in the sub-critical and trans-critical regime using the Hyperbolic-Elliptic splitting of [30]. The results are discussed and explanations for the convergence behavior is sought.

Chapter VII gives a critical summary of the results with an outlook on paths yet to be embarked upon.

Chapter II

Manipulating the Euler Equations

> "That thou mayst shake the superflux to them
> and show the heavens more just."
>
> W. Shakespeare, 'King Lear', Act III, Sc. 4

> "Wind tunnel experiments suffer from diffusion,
> not appearing in the Euler flow model."
>
> E. van der Maarel, [53]

The Euler equations describe the inviscid flow of a compressible and non-heat-conducting fluid. In two dimensions, the conservative form of the equations can be written as

$$-\frac{\partial \mathbf{U}}{\partial \mathbf{t}} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{y}} \tag{2.1}$$

with the vector of conserved variables **U** and the conservative fluxes **F** and **G** given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \qquad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix}. \tag{2.2}$$

The fluid density is denoted by ρ, the velocities in the x and y directions by u and v, respectively, and the static pressure by p. The total specific energy E is given by

$$E = e + \frac{1}{2}\left(u^2 + v^2\right), \tag{2.3}$$

where e stands for the specific internal energy. The total specific enthalpy H is given by

$$H = E + \frac{p}{\rho} \tag{2.4}$$

In the following the gas is assumed to be perfect, thus

$$p = \rho R T \tag{2.5}$$

with the gas constant R and the absolute temperature T.

Ultimately, the steady Euler equations in conservation form 2.1 are to be solved, since this form allows the weak solutions that occur in trans-critical flows. However, in the course of the analysis other forms of the equations will be more convenient. The explicit time-stepping that will be employed makes use of the time-derivative to march the solution in time, although this time-derivative does not have to be physically meaningful. The transient that is sought is the one that gives the fastest convergence to the steady state.

The Euler equations in quasi-linear form using primitive variables $\mathbf{V}$ are

$$-\frac{\partial \mathbf{V}}{\partial \mathbf{t}} = \hat{\mathbf{A}}\frac{\partial \mathbf{V}}{\partial \mathbf{x}} + \hat{\mathbf{B}}\frac{\partial \mathbf{V}}{\partial \mathbf{y}} \tag{2.6}$$

with

$$\mathbf{V} = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix}, \quad \hat{\mathbf{A}} = \begin{bmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & 1/\rho \\ 0 & 0 & u & 0 \\ 0 & \rho u^2 & 0 & u \end{bmatrix}, \quad \hat{\mathbf{B}} = \begin{bmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & 1/\rho \\ 0 & 0 & \rho v^2 & v \end{bmatrix}. \tag{2.7}$$

In streamwise coordinates ξ, η and using symmetrizing variables $\tilde{\mathbf{Q}}$ [54] the quasi-linear form of the symmetrized Euler equations is obtained:

$$-\frac{\partial \tilde{\mathbf{Q}}}{\partial t} = \tilde{\mathbf{A}}\frac{\partial \tilde{\mathbf{Q}}}{\partial \xi} + \tilde{\mathbf{B}}\frac{\partial \tilde{\mathbf{Q}}}{\partial \eta}. \tag{2.8}$$

with

$$\partial\tilde{\mathbf{Q}} = \begin{pmatrix} \partial p/\rho a \\ \partial \tilde{u} \\ \partial \tilde{v} \\ \partial p - c^2 \partial \rho \end{pmatrix}, \quad \tilde{\mathbf{A}} = \begin{bmatrix} \tilde{u} & 0 & 0 & 0 \\ 0 & \tilde{u} & 0 & 0 \\ 0 & 0 & \tilde{u} & 0 \\ 0 & 0 & 0 & \tilde{u} \end{bmatrix}, \quad \tilde{\mathbf{B}} = \begin{bmatrix} 0 & 0 & c & 0 \\ 0 & 0 & 0 & 0 \\ c & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

with the modulus of the flow speed $\tilde{u} = \sqrt{u^2 + v^2}$. Note that in streamwise coordinates the normal velocity is zero: $\tilde{v} = 0$. However, the flow turning rate $\partial \tilde{v}$ is not.

2.1 From the 1-D Scalar Equation to the 2-D Euler System

A plethora of scalar advection schemes has been designed in the past that satisfy various constraints of optimality. In particular it can be shown that for the linear advection equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{2.10}$$

with the unknown u and the advection speed a, the least dissipative monotonic linear three-point scheme is the first order Upwind scheme,

$$u_i \longleftarrow (1-\nu)u_i + \nu u_{i-1}, \tag{2.11}$$

with the mesh spacing h, the time-step Δt and the CFL number $\nu = \frac{a\Delta t}{h}$. In one space dimension, monotonic Upwind schemes that are third order accurate in smooth regions of the solution have been presented by Van Leer [55, 56].

The extension of these scalar schemes to the 1-D Euler system has been done successfully using various "Approximate Riemann solvers" based on Godunov's method [4]. The most popular flux splittings are Van Leer's Flux-Vector splitting [1], Osher's splitting [2] and Roe's Flux-Difference splitting [3]. However, the straightforward dimension-by-dimension extension of the 1-D algorithm to higher dimensions leads to a loss in accuracy when resolving grid-oblique features like shear layers as shown in figure 1.1. What is needed is a genuinely multidimensional interpretation of the flow phenomena by the scheme.

Various attempts have been made to incorporate the flow direction into the Riemann-problem that is solved across cell interfaces. Early work by Raithby [57] was met with increasing interest [6, 7]. While these modifications do improve the quality of first order accurate solutions dramatically, the improvements over second-order accurate solutions are meager. Moreover, controlling monotonicity is an unresolved issue with these methods.

Recognizing that a major obstacle in devising genuinely multidimensional Upwind schemes are the jumps in the unknowns at the cell interfaces, various scalar advection schemes that employ a continuous variation of the unknowns have been developed in the past few years by Deconinck et al. and Roe et al. [23,24,33,34,58,59] using the Fluctuation splitting approach [18]. These schemes have little dissipation compared to dimensionally split schemes for advection oblique to the grid lines. The basic first order scheme, the N-scheme, can actually be shown to have the minimum dissipation necessary for being a positive scheme [60] and is in this sense the extension of the 1-D Upwind scheme.

2.2 Diagonalizing the Euler Equations

However, applying scalar schemes requires the Euler system to be diagonalized into a decoupled set of scalar equations. This is not possible for the Euler equations in general since the Jacobians **A** and **B** do not commute. Looking at the quasi-linear form of the equations 2.1 and 2.6, one finds

$$-\frac{\partial \mathbf{U}}{\partial \mathrm{t}} = \mathbf{A}\frac{\partial \mathbf{U}}{\partial \mathrm{x}} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \mathrm{y}}$$

$$-\mathbf{A}^{-1}\frac{\partial \mathbf{U}}{\partial \mathrm{t}} = \frac{\partial \mathbf{U}}{\partial \mathrm{x}} + \mathbf{A}^{-1}\mathbf{B}\frac{\partial \mathbf{U}}{\partial \mathrm{y}}$$

and after diagonalizing $\mathbf{A}^{-1}\mathbf{B}$ as $\mathbf{\Lambda} = \mathbf{L}\mathbf{A}^{-1}\mathbf{B}\mathbf{R}$ with the left and right eigenvectors **L** and **R**

$$-\mathbf{A}^{-1}\frac{\partial \mathbf{U}}{\partial \mathrm{t}} = \frac{\partial \mathbf{U}}{\partial \mathrm{x}} + \mathbf{R}\mathbf{\Lambda}\mathbf{L}\frac{\partial \mathbf{U}}{\partial \mathrm{y}}.$$

$$-\mathbf{L}\mathbf{A}^{-1}\mathbf{R}\mathbf{L}\frac{\partial \mathbf{U}}{\partial \mathrm{t}} = \frac{\partial \mathbf{L}\mathbf{U}}{\partial \mathrm{x}} + \mathbf{\Lambda}\frac{\partial \mathbf{L}\mathbf{U}}{\partial \mathrm{y}}.$$

Changing to characteristic variables $\partial \mathbf{W} = \mathbf{L}\partial \mathbf{U}$, one obtains

$$-\mathbf{L}\mathbf{A}^{-1}\mathbf{R}\frac{\partial \mathbf{W}}{\partial \mathrm{t}} = \frac{\partial \mathbf{W}}{\partial \mathrm{x}} + \mathbf{\Lambda}\frac{\partial \mathbf{W}}{\partial \mathrm{y}}. \tag{2.12}$$

or in the steady state

$$\vec{\lambda}_i \nabla W_i = 0, \qquad i = 1, 4. \tag{2.13}$$

In the case of the Euler equations the eigenvalues $\vec{\lambda}_i$, and characteristic variables are most easily obtained from the symmetrized form 2.8 to yield the following system of equations:

$$(\partial_t + q\partial_s)(\partial p - c^2\partial\rho) = 0, \tag{2.14}$$

invariance of entropy along a streamline,

$$(\partial_t + q\partial_s)(\partial p + \rho q\partial q) = 0, \tag{2.15}$$

invariance of total enthalpy along a streamline, and

$$(\partial_t + \frac{\beta}{M}\partial_s + \frac{q}{M}\partial_n)(\beta\partial p + \rho q^2\partial\theta) = 0 \tag{2.16}$$

$$(\partial_t + \frac{\beta}{M}\partial_s - \frac{q}{M}\partial_n)(\beta\partial p - \rho q^2\partial\theta) = 0 \tag{2.17}$$

invariance of the acoustic characteristics along the Mach lines.

The norm of the flow velocity is $q = \sqrt{u^2 + v^2}$, the flow angle is $\theta = v/u$ and $\beta = \sqrt{M^2 - 1}$.

While this treatment fully decouples the Euler system into four scalar equations for supersonic flow, $M > 1$, the acoustic system becomes complex valued and remains coupled in the subsonic case, $M < 1$.

Initial attempts at treating the subsonic Euler equations with scalar advection schemes attempted to fully diagonalize the system which necessitates to pick a few specific directions from the infinite number of possible ones for the acoustic propagation. Various wave models are presented and discussed in [34,61]. All of them suffered from a loss of accuracy compared to the scalar case and/or from a lack of robustness.

2.3 Canonical Variables

Another splitting of the Euler system along similar ideas has been proposed by Ta'asan [29].

$$\left[\begin{array}{cc|cc} \frac{\partial}{\partial_1} & \frac{\partial}{\partial_2} & 0 & 0 \\ -\frac{\partial}{\partial_y} & \frac{\partial}{\partial_x} & -\frac{c^2}{\gamma(\gamma-1)}\frac{\partial}{\partial_3} & \frac{1}{q}\frac{\partial}{\partial_3} \\ \hline 0 & 0 & T\rho Q & 0 \\ 0 & 0 & 0 & \rho Q \end{array}\right] \begin{bmatrix} u \\ v \\ s \\ H \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{2.18}$$

with

$$\frac{\partial}{\partial_1} = \frac{\rho}{c^2}\left((c^2 - u^2)\frac{\partial}{\partial x} - uv\frac{\partial}{\partial y}\right) \tag{2.19}$$

$$\frac{\partial}{\partial_2} = \frac{\rho}{c^2}\left((c^2 - v^2)\frac{\partial}{\partial y} - vu\frac{\partial}{\partial x}\right) \tag{2.20}$$

$$\partial_3 = v\frac{\partial}{\partial x} - u\frac{\partial}{\partial y} \tag{2.21}$$

One finds the invariance along streamlines for total enthalpy and entropy and an elliptic 2×2 kernel. The formulation of the kernel however differs from the one in equations 2.16 and 2.17, which are formulated in terms of the pressure gradient ∂p and the flow turning rate $\partial\theta$. The canonical subsystem 2.18 is formulated in terms of the divergence and the vorticity and has a source term in the vorticity equation that reflects the production of vorticity due to gradients in entropy and enthalpy as given by Crocco's law.

2.4 Windfalls of Local Preconditioning

In the wake of ongoing work on local preconditioning of the Euler equations [26,62], the coincidence of characteristic decoupling and characteristic time-stepping led to the idea of applying the preconditioning matrix to the subsonic system and to treat the acoustic kernel as a coupled elliptic subsystem. This allows to treat the advective equations with a suitable Upwind scheme and to treat the elliptic part with a more centered discretization. Provided that the centered scheme has an elliptic domain of dependence, the unsatisfactory choice of propagation directions for acoustic quantities can be omitted.

Applying a local preconditioner to the time-derivative of 2.12, one finds

$$\begin{aligned}
-\frac{\partial \mathbf{U}}{\partial \mathbf{t}} &= \mathbf{P}(\mathbf{A}\frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial \mathbf{y}}) \\
-\mathbf{A}^{-1}\mathbf{P}^{-1}\frac{\partial \mathbf{U}}{\partial \mathbf{t}} &= \frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \mathbf{A}^{-1}\mathbf{B}\frac{\partial \mathbf{U}}{\partial \mathbf{y}} \\
-\mathbf{L}\mathbf{A}^{-1}\mathbf{P}^{-1}\mathbf{R}\mathbf{L}\frac{\partial \mathbf{U}}{\partial \mathbf{t}} &= \mathbf{L}\frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \mathbf{L}\mathbf{R}\boldsymbol{\Lambda}\mathbf{L}\frac{\partial \mathbf{U}}{\partial \mathbf{y}}. \\
-\mathbf{L}\mathbf{A}^{-1}\mathbf{P}^{-1}\mathbf{R}\frac{\partial \mathbf{W}}{\partial \mathbf{t}} &= \frac{\partial \mathbf{W}}{\partial \mathbf{x}} + \boldsymbol{\Lambda}\frac{\partial \mathbf{W}}{\partial \mathbf{y}}.
\end{aligned} \tag{2.22}$$

In order to achieve full decoupling of the system of equations, the matrix of the time derivative in the supersonic case, $\mathbf{L}\mathbf{A}^{-1}\mathbf{P}^{-1}\mathbf{R}$ has to be diagonal. It can be shown [33,63] that this condition uniquely defines a preconditioner which is the preconditioning matrix of [26]. For the symmetrizing variables it is

$$\mathrm{P} = \begin{bmatrix} \frac{\chi}{\beta^2}M^2 & -\frac{\chi}{\beta^2}M^2 & 0 & 0 \\ -\frac{\chi}{\beta^2}M & \frac{\chi}{\beta^2}+1 & 0 & 0 \\ 0 & 0 & \chi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2.23}$$

with $\chi = \beta / \max(M, 1)$ and $\beta = \sqrt{|M^2 - 1|}$. While the choice of preconditioner is no more unique in the subsonic case, the preconditioner 2.23 still decouples into the 2×2 system of elliptic equations that represent acoustic propagation and two scalar advection equations for enthalpy and entropy. After transformation to characteristic variables 2.14-2.17, one finds

$$-\frac{\partial \mathrm{W}}{\partial t} = \begin{bmatrix} \chi\nu^+ & \chi\nu^- & 0 & 0 \\ \chi\nu^- & \chi\nu^+ & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \frac{\partial \mathrm{W}}{\partial \xi} + \begin{bmatrix} \frac{\chi}{\beta} & 0 & 0 & 0 \\ 0 & -\frac{\chi}{\beta} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \frac{\partial \mathrm{W}}{\partial \eta} \tag{2.24}$$

with $\nu^+ = (M^2 - 1 + \beta^2)/2\beta^2$ and $\nu^- = (M^2 - 1 - \beta^2)/2\beta^2$. The conservative flux balance using characteristic variables can thus be expressed as

$$
\begin{aligned}
&\frac{\partial \mathbf{F}}{\partial x}+\frac{\partial \mathbf{G}}{\partial y}= \\
&\quad \frac{\partial \mathbf{U}}{\partial \mathbf{W}}\mathbf{P}^{-1}\left(\mathbf{r}^1,\mathbf{r}^2\right)\left[\begin{bmatrix} \chi\nu^+ & \chi\nu^- \\ \chi\nu^- & \chi\nu^+ \end{bmatrix}\frac{\partial}{\partial \xi}\begin{pmatrix} W^1 \\ W^2 \end{pmatrix}+\begin{bmatrix} \frac{\chi}{\beta} & 0 \\ 0 & -\frac{\chi}{\beta} \end{bmatrix}\frac{\partial}{\partial \eta}\begin{pmatrix} W^1 \\ W^2 \end{pmatrix}\right]+ \\
&\quad \frac{\partial \mathbf{U}}{\partial \mathbf{W}}\mathbf{P}^{-1}\sum_{l=3}^{4}\vec{\lambda}\cdot\nabla W r^l \qquad (2.25)
\end{aligned}
$$

which lends itself straightforwardly for explicit time-stepping or other forms of collective updates of the unknowns.

The calculations in section 6.3 have been obtained using the preconditioned system since it most readily can be put in conservative form. The singularity in the stagnation point where the flow speed vanishes and the flow direction θ becomes undefined might make the canonical form shown in the previous section 2.3 a more robust choice that has yet to be investigated more thoroughly. Ta'asan has shown impressive iso-enthalpic results [29], but the application to general inviscid flows has not yet been demonstrated.

Chapter III

The Discretization

the wind is a Lady with
bright slender eyes (who

moves) at sunset
and who – touches – the
hills without any reason

(i have spoken with this
indubitable and green person "Are
you the wind?" "Yes" "Why do you touch the flowers
as if they were unalive, as

if they were ideas?" "because, sir
things which in my mind blossom will
stumble beneath a clumsiest disguise, appear
capable of fragility and indecision

– do not suppose these
without any reason and otherwise
roses and mountains
different from the i am who wanders

imminently across the renewed world"
to me said the) wind being a Lady
in a green
dress, who touches: the fields
(at sunset)

e. e. cummings

From the discussions in chapter II the need for two types of discretizations arises: a scalar upstream-biased scheme that is monotonic and exhibits little diffusion to discretize the advection of entropy and enthalpy and a coupled scheme for the 2×2 system that has an elliptic domain of dependence. The added design goal of these schemes is the independence of the updates from the orientation of the cell-interfaces in order to ensure genuine multi-dimensionality.

The 'Fluctuation-Splitting' discretization leading to schemes that satisfy the aforementioned constraints, has been proposed initially by Roe in [18]. Various 1-D and 2-D scalar advection schemes have been derived [22] and the derivation of the ones used here is briefly shown in sections 3.1 and 3.2.

The space discretization is of the 'Cell-Vertex' type where the data are stored at the vertices of the grid and the solution varies linearly over each element. Flux-balances, termed 'fluctuations', are evaluated over each element and distributed to the nodes. The differences in the various Cell-Vertex schemes come about in the distribution of the fluctuation to the nodes of the cell. The most prominent member of this class of schemes, and also its most basic, is Jameson's scheme [64] that is widely used for industrial applications. It mainly excels in its convergence properties. The more sophisticated ancestors of Fluctuation-Splitting are the schemes of Ni [27] and Hall [28]. Both schemes attempt to distribute the fluctuation along concepts of upwinding, however without diagonalizing the Euler equations as is being done here.

3.1 One-dimensional Fluctuation-Splitting for Linear Advection

The Fluctuation-Splitting method assumes its data to vary continuously between nodes, as in Finite-Element methods (fig. 3.1). Conveniently, the data variation is presented as a sum of the linear basis functions $w_i(x)$ (fig. 3.2). This leads to the variation of the data as.

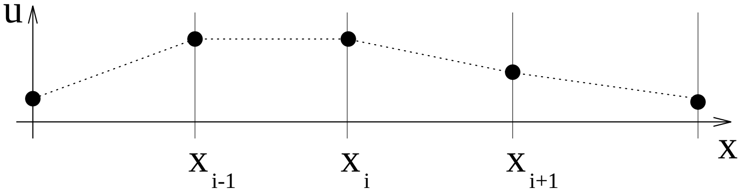


Figure 3.1: The data for one-dimensional Fluctuation-Splitting is assumed to vary linearly between the nodal values.

$$u(x) = \sum_{i=1}^{N} w_i(x) u_i. \tag{3.1}$$

To solve a linear scalar advection equation

$$u_t + a u_x = 0 \tag{3.2}$$

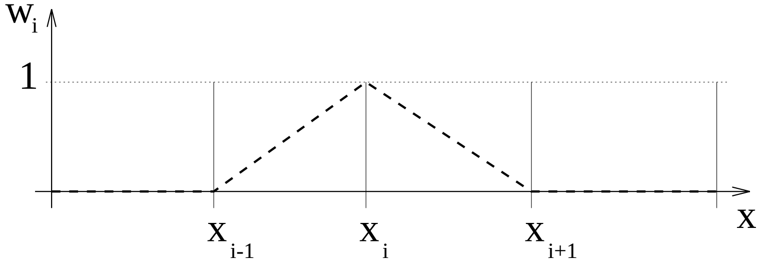


Figure 3.2: The piecewise linear basis function $w_i(x)$ in one dimension.

with the unknown u and the advection speed a, the subscripts indicating derivatives, one can integrate the equation for each cell

$$\int_{x_i}^{x_{i+1}} u_t dx = -a \int_{x_i}^{x_{i+1}} u_x dx = a(u_i - u_{i+1}) = \phi_{i+\frac{1}{2}}. \tag{3.3}$$

to find the fluctuation $\phi_{i+\frac{1}{2}}$ in the cell formed by the nodes i and $i+1$. In order to converge to the solution, the scheme sends a "signal", "an action performed on the data so as to bring it closer to equilibrium" [18]. In the framework of Fluctuation-Splitting the advection equation is discretized for every cell as

$$l_i u_i \longleftarrow l_i u_i + \Delta t \alpha_{i,i+\frac{1}{2}} \phi_{i+\frac{1}{2}} \tag{3.4}$$

$$l_{i+1} u_{i+1} \longleftarrow l_{i+1} u_{i+1} + \Delta t \alpha_{i+1,i+\frac{1}{2}} \phi_{i+\frac{1}{2}} \tag{3.5}$$

with the length $l_i = \frac{1}{2}(x_{i+1} - x_{i-1})$ that weighs each node. Considering the case of $a > 0$, a natural choice to use in cell $i + \frac{1}{2}$ is $\alpha_i = 0$, $\alpha_{i+1} = 1$, leading to the first order Upwind Finite-Volume scheme:

$$\begin{aligned} u_i &\longleftarrow u_i - \frac{a\Delta t}{l_i}(u_i - u_{i-1}), \\ u_i &\longleftarrow \nu u_{i-1} + (1-\nu) u_i. \end{aligned} \tag{3.6}$$

with the CFL number $\nu = \frac{a\Delta t}{l}$. Thus, with little surprise this choice of distribution coefficients recovers the scheme with the optimum amount of dissipation for retaining the positivity of the scheme [60]. However, as clearly demonstrated in [60, 65] and shown in section 3.2, the difference in interpretation of the data leads to different schemes for systems and/or higher dimensions.

3.2 Two-dimensional Fluctuation-Splitting for Linear Advection

The simplest two-dimensional advection problem to consider is the linear advection equation

$$u_t + \mathbf{a}\cdot\nabla u = 0 \tag{3.7}$$

where $\mathbf{a} = (a, b)^T = const.$ in $\mathbb{R}^2$. The integral of u_t over an element T is

$$\iint_T u_t dxdy = -\iint_T \mathbf{a}\cdot\nabla u dxdy = \oint_{\partial T} u\mathbf{a}d\mathbf{n} = \phi_T \tag{3.8}$$

with the boundary of the element ∂T and the inward normal $\mathbf{n}$. This is the definition of the fluctuation on the element ϕ_T.

As in the one-dimensional case, the data u are assumed to vary linearly over each element. The unknown can be expressed as a sum of the two-dimensional linear basis-functions, leading to the variation of the data as shown in figure 3.3.

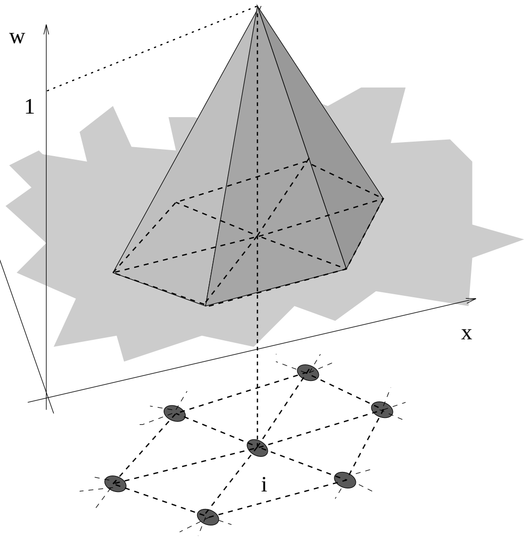


Figure 3.3: The piecewise linear basis function $w_i(x, y)$ in two dimensions, aka. the 'tent'-function.

$$u(x, y) = \sum_{i=1}^{N} w_i(x, y)u_i. \tag{3.9}$$

The fluctuation ϕ_T can be thus be discretized using the trapezoidal rule as

$$\phi_T = \frac{1}{2}(u_1 + u_2)\mathbf{a}\cdot\mathbf{n}_3 + \frac{1}{2}(u_2 + u_3)\mathbf{a}\cdot\mathbf{n}_1 + \frac{1}{2}(u_3 + u_1)\mathbf{a}\cdot\mathbf{n}_2 \tag{3.10}$$

where $\mathbf{n}_i$ is the scaled inward normal to the side i, opposite the node i. Since the sides of the triangle form a closed loop, $\sum \mathbf{n}_i = 0$, equation 3.10 can be rearranged to find (the exercise being left to the reader)

$$\phi_T = -\frac{1}{2}(u_1)\mathbf{a}\cdot\mathbf{n}_1 - \frac{1}{2}(u_2)\mathbf{a}\cdot\mathbf{n}_2 - \frac{1}{2}(u_3)\mathbf{a}\cdot\mathbf{n}_3 = -\sum_{i=1}^{3} k_i u_i \tag{3.11}$$

with

$$k_i = \frac{1}{2}\mathbf{a}\cdot\mathbf{n}_i. \tag{3.12}$$

The following updates are performed at every time step:

$$S_i u_i \longleftarrow S_i u_i + \sum_T \alpha_{i,T}\Delta t \phi_T \qquad \forall T \supset i. \tag{3.13}$$

The area S_i weighing each node is one third of the area of all triangles that are formed with node i (fig. 3.4). Using eqs. 3.11 and 3.13 the following update formulae for one triangle T are obtained:

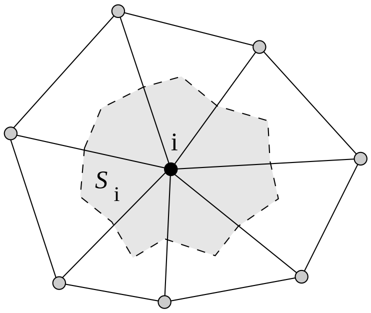


Figure 3.4: Median dual around node i with the surface S_i.

$$S_1 u_1 \longleftarrow S_1 u_1 - \alpha_{1,T}\Delta t\,(k_1 u_1 + k_2 u_2 + k_3 u_3) + TFOT, \tag{3.14}$$
$$S_2 u_2 \longleftarrow S_2 u_2 - \alpha_{2,T}\Delta t\,(k_1 u_1 + k_2 u_2 + k_3 u_3) + TFOT, \tag{3.15}$$
$$S_3 u_3 \longleftarrow S_3 u_3 - \alpha_{3,T}\Delta t\,(k_1 u_1 + k_2 u_2 + k_3 u_3) + TFOT. \tag{3.16}$$

The terms that arise from the contribution of other triangles, $TFOT$, will be omitted in the following as every element will be treated as self-contained.

3.2.1 Three Design Criteria for 2-D Advection Schemes

Fluctuation-Splitting schemes attempt to achieve three design goals: conservation, positivity and linearity preservation.

Conservation

A necessary criterion to be observed by any scheme applied to compressible flow is conservation. Writing out the sum of all updates to all nodes, one finds

$$\sum_{i=1}^{N} S_i u_i \longleftarrow \sum_{i=1}^{N} S_i u_i - \Delta t \sum_{i=1}^{N} \sum_{T} \alpha_{i.T} \sum_{j=1}^{3} k_j u_j \tag{3.17}$$

and after rearranging the summation operators

$$\sum_{i=1}^{N} S_i u_i \longleftarrow \sum_{i=1}^{N} S_i u_i - \Delta t \sum_{T} \sum_{i=1}^{N} \alpha_{i,T} \sum_{j=1}^{3} k_j u_j . \tag{3.18}$$

Provided that the sum of distribution coefficients adds to unity,

$$\forall T : \qquad \sum_{i=1}^{3} \alpha_{i,T} = 1, \tag{3.19}$$

the global sum simplifies to

$$\sum_{i=1}^{N} S_i u_i \longleftarrow \sum_{i=1}^{N} S_i u_i - \Delta t \sum_{T} \phi_T . \tag{3.20}$$

In the linear case discussed here, it is easily seen that the fluctuations ϕ_T telescope, since they are defined by a curve integral around elements that do not overlap,

$$\sum_{T} \phi_T = \sum_{T} \oint_T u \mathbf{a} d\mathbf{n} = \oint_{\partial\Omega} u \mathbf{a} d\mathbf{n}, \tag{3.21}$$

and one is left with boundary terms only,

$$\sum_{i=1}^{N} S_i u_i \longleftarrow \sum_{i=1}^{N} S_i u_i - \oint_{\partial\Omega} u \mathbf{a} d\mathbf{n}. \tag{3.22}$$

The conservative linearization of a nonlinear equation such that property 3.21 holds is discussed in section 3.4.

Positivity

A second desirable aspect is monotonicity of the scheme. It is formulated here following the concept of positivity of Spekreijse [66,67] improving over the TVD criterion proposed by Harten [68] that is too dissipative for higher dimensions. The scheme (3.13) can be rewritten in the following form:

$$u_i^{n+1} = \sum c_k u_k^n. \tag{3.23}$$

A scheme is positive if the values at the new time level $n+1$ are a convex combination of the values at the old time level n, i.e. $c_k \geq 0$. In this case, effectively a weighted average is performed and the scheme observes a maximum principle. That is, u_i^{n+1} is bounded by the minimum and maximum values of u_k^n in its stencil. Condition 3.23 actually provides a time-step constraint for all explicit schemes as shown in section 3.2.2.

Linearity Preservation

A third criterion, the property of preserving an exact steady state, whenever this solution is linear in the spatial coordinates, is an obviously desirable property in the context of linear Finite-Element methods. But this property 'Linearity Preservation' (LP) is not automatically satisfied in the Fluctuation-Splitting context. A test whether a scheme is LP can be performed by looking at how a triangle is treated that is in equilibrium, i.e. with $\phi_T = 0$. If the α_i of equation 3.16 for this case remain finite, no updates to the nodes of this element are sent and the exact solution is preserved. It can be shown [23] that a scheme with the property LP is second order accurate in space on a regular grid for linear advection.

Godunov's Theorem

Godunov's Theorem states the incompatibility between second order accuracy and positivity for linear schemes where the coefficients are independent of the data, $c_k \neq f(u_i)$. It can be generalized to higher dimensions. Writing the scheme in the form (3.13), one needs for linearity preservation either $\alpha_i \neq f(u_i)$ or $\alpha_i = \beta_i/\phi_T$ where β_i are linear functions $f(u_i)$ that sum up to ϕ_T.

$$\phi_T = \sum_{i=1}^{3} \alpha_i \phi_T = \sum_{i=1}^{3} \frac{\beta_i}{\phi_T} \phi_T = \sum_{i=1}^{3} \beta_i \tag{3.24}$$

For LP it is needed that the α_i be bounded, $\beta_i \longrightarrow 0$ as $\phi_T \longrightarrow 0$.

Conditions on positivity for a scheme of the form (3.13) can be derived by looking first at the case of $\alpha_i \neq f(u_i)$. For the coefficients of u_2, u_3 in (3.14) to be non-negative one has to require either $\alpha_1 = 0$ or both k_2, k_3 to have the same sign. Similarly, one needs for u_1, u_3 to have non-negative coefficients in (3.15) $\alpha_2 = 0$ or both k_1, k_3 to have the same sign. As not all k_i can have the same sign for $\mathbf{a} \neq 0$, α_1 or α_2 need to vanish. Taking each pair of eq. (3.14-3.16) in turn (the exercise being left to the reader), one finds that α_1 or α_3 and α_2 or α_3 have to vanish as well. Hence, only one α_i can be non-zero which, without loss of generality, can be chosen to be α_1. Following equation 3.19, it has to be unity for conservation. Thus, the fluctuation is sent to one node,

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t \left(k_1 u_1 + k_2 u_2 + k_3 u_3 \right). \tag{3.25}$$

Positivity is achieved for $S_1 - k_1 \Delta t \geq 0$ and $k_2, k_3 \leq 0$, which holds if Δt is chosen small enough and if the sides 2 and 3 are outflow sides. The entire update then goes to the downstream node 1 as can be seen in figure 3.5. As $\alpha_1 = 1$, the coefficients are bounded and the scheme has the property LP. Thus, for all triangles with one inflow side, sending the entire update to the downstream node is positive and linearity preserving and only this type of updating for these cases will be used, the one-target case.

In the two-target case (cf. fig. 3.5), a linear positive scheme, the N-scheme, can also be derived with the condition $\alpha_i = \beta_i / \phi_T$, as is shown in section 3.2.2. But as β_i is a linear function of u_i, it may be non-zero for $\phi_T = 0$, therefore this scheme does not preserve linearity. A linear scheme can only be locally positive and linearity preserving if and only if all triangles have only one inflow side, which is impossible for flows with zero-divergence. Thus, the two-dimensional extension of Godunov's theorem states that linear monotonic schemes can be at most first order accurate.

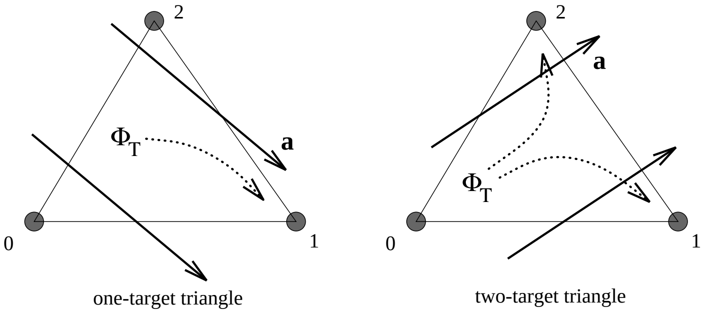


Figure 3.5: Triangle with one inflow side and one updated outflow vertex (left) and a triangle with two inflow sides and two updated outflow vertices (right).

The various Fluctuation-Splitting schemes differ in how the update is distributed in the cases with two inflow sides (figure 3.5) which is being discussed in the following sections. The three scalar Upwind schemes that are used in the testcases are presented in the following sections. A presentation of the Finite-Volume scheme formulated as a Fluctuation-Splitting scheme, a viewpoint initially presented in [24], is included for an illustration of genuine multidimensionality only. Accuracy studies for the different schemes have been presented by the various contributors in [24, 34, 69] and are not be repeated here.

3.2.2 The N-Scheme: a Positive Linear Scheme that is not LP

In the case of a triangle with two inflow sides the fluctuation ϕ_T has to be split and sent to the two downstream nodes which are chosen here to be 1 and 2. As has been proven, the scheme cannot be both positive and linearity preserving. For the N-scheme, positivity is held onto and the property LP is relinquished. Hence, the scheme can be written in the following form:

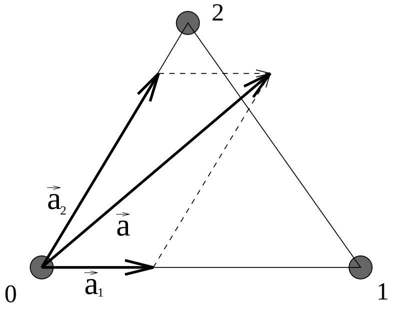


Figure 3.6: The N-scheme can be viewed as two distributions resulting from splitting the advection-vector along the inflow sides.

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t \beta_1, \tag{3.26}$$

$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t \beta_2, \tag{3.27}$$

with the β_i from equation 3.24. It is shown in [60,70] that an optimal scheme is obtained, in the sense of using a minimal stencil and allowing a maximum time-step while minimizing the dissipation, if the advection speed is decomposed into two components along the sides leading downstream to the nodes 1 and 2 as depicted in figure 3.6,

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t k_1 \left(u_1 - u_3\right), \tag{3.28}$$

$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t k_2 \left(u_2 - u_3\right). \tag{3.29}$$

Local positivity is achieved for

$$\Delta t \leq \min\left[\frac{S_1}{k_1}, \frac{S_2}{k_2}\right] \tag{3.30}$$

On a regular rectangular mesh with the diagonals drawn such as to minimize $|k|$, the N-scheme is identical to the streamline Upwind scheme by Rice and Schnipke [71]. The splitting of the advection speed along the sides of the triangle has also been done in the Finite-Element context by Hughes et al. [72]. The name N-scheme has been coined by D. Sidilkover [73] referring to its narrow stencil. In effect, with the diagonals properly drawn, the N-scheme has a stencil of only three points. As expected the scheme is first-order accurate [60,74].

3.2.3 The Dimensionally Split First-Order Upwind Fluctuation-Splitting Scheme

In the one-dimensional case the alternate viewpoint of Fluctuation-Splitting leads to the same first-order Upwind scheme that results from a Finite-Volume approach as has been shown in 3.1. The first-order dimensionally split Upwind Finite-Volume scheme can be cast into a fluctuation form in order to illuminate the differences that arise between the concepts in higher dimensions. The basic Finite-Volume scheme is expressed as

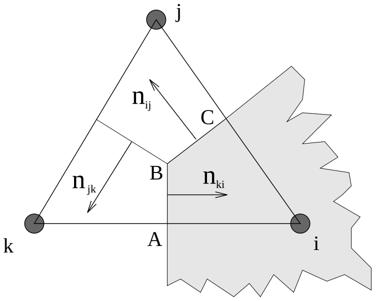


Figure 3.7: Interpretation of the dimensionally split first-order Upwind Finite-Volume scheme based on the vertices as a Fluctuation-Splitting distribution scheme.

$$S_i u_i \longleftarrow S_i u_i - \Delta t \oint_{S_i} u \mathbf{a} d\mathbf{n} \tag{3.31}$$

Looking at one triangle T (figure 3.7) formed with the vertex i and two other vertices j, k in permutation, the contributions of triangle ijk to the flux-integral of i are

$$\int_{AB} u \mathbf{a} d\mathbf{n} = u_k \left(\mathbf{a}\cdot\mathbf{n}_{ki}\right)^+ + u_i \left(\mathbf{a}\cdot\mathbf{n}_{ki}\right)^- \tag{3.32}$$

$$\int_{BC} u \mathbf{a} d\mathbf{n} = u_j \left(\mathbf{a}\cdot\mathbf{n}_{ji}\right)^+ + u_i \left(\mathbf{a}\cdot\mathbf{n}_{ji}\right)^- \tag{3.33}$$

with the standard Upwind definition of

$$\left(\mathbf{a}\cdot\mathbf{n}\right)^+ = \max\left(0, \mathbf{a}\cdot\mathbf{n}\right), \qquad \left(\mathbf{a}\cdot\mathbf{n}\right)^- = \min\left(0, \mathbf{a}\cdot\mathbf{n}\right). \tag{3.34}$$

Since positivity is achieved for (cf. equ. 3.31)

$$\Delta t \leq \frac{S_i}{\sum_T \max\left(0, k_i\right)} \tag{3.35}$$

and the scheme is linear, it can be at most first-order accurate. Rewriting the fluctuation in terms of the interior normals $\mathbf{n}_{ij}$ as shown in figure 3.7

$$\mathbf{n}_i = 2\left(\mathbf{n}_{ij} - \mathbf{n}_{ki}\right) = -\left(\mathbf{n}_j + \mathbf{n}_k\right), \tag{3.36}$$

one obtains

$$\phi_T = -\frac{1}{2}\left(u_i \mathbf{a}\cdot\mathbf{n}_i + u_j \mathbf{a}\cdot\mathbf{n}_j + u_k \mathbf{a}\cdot\mathbf{n}_k\right) \tag{3.37}$$

$$= -\frac{1}{2}\left((u_i - u_k)\mathbf{a}\cdot\mathbf{n}_i + (u_j - u_k)\mathbf{a}\cdot\mathbf{n}_j\right) \tag{3.38}$$

$$= -\frac{1}{2}\left((u_j - u_i)\mathbf{a}\cdot\mathbf{n}_j + (u_k - u_i)\mathbf{a}\cdot\mathbf{n}_k\right) \tag{3.39}$$

$$= -\frac{1}{2}\left((u_k - u_j)\mathbf{a}\cdot\mathbf{n}_k + (u_i - u_j)\mathbf{a}\cdot\mathbf{n}_i\right) \tag{3.40}$$

$$= -\frac{1}{2}\sum_{ijk}(u_i - u_j)\mathbf{a}(\mathbf{n}_i - \mathbf{n}_j) \tag{3.41}$$

and after using interior normals

$$\phi_T = \sum_{ijk}(u_i - u_j)\mathbf{a}\cdot(\mathbf{n}_{i,j}) \tag{3.42}$$

Comparing eqs. 3.42 with 3.31 and 3.33, the distribution scheme that recovers the Finite-Volume scheme is found to be

$$(u_i - u_j)\mathbf{a}\cdot(\mathbf{n}_{ij}) \quad \text{to} \quad \begin{cases} \text{node } i \text{ if } \mathbf{a}\cdot\mathbf{n}_{ij} > 0 \\ \text{node } j \text{ if } \mathbf{a}\cdot\mathbf{n}_{ij} < 0 \end{cases} \quad \forall\{i,j,k\} \tag{3.43}$$

From this it is obvious that the fluctuation is always distributed to two vertices. The one-target case that is positive and linearity preserving is not recovered by the dimensionally split scheme which explains the larger dissipation compared to the optimal N-scheme.

3.2.4 The LDA-Scheme, a Non-Positive Scheme with the Property LP

Rather than losing the property LP and retaining positivity as with the N-scheme, one can also seek schemes of higher order which do not obey a maximum principle. There is a wealth of schemes in this class ranging from Upwind schemes, like the Low Diffusion scheme A (LDA), which is discussed below, and its variation LDB, to central differencing schemes like Lax-Wendroff type schemes or Petrov-Galerkin Finite-Element schemes.

The LDA-scheme is obtained by splitting the fluctuation according to the two areas when the triangle is dissected along the advection vector as given in figure 3.8. Let point 4 be the intersection of the advection vector drawn through the Upwind node 3 and the downstream face 3 between nodes 1 and 2. The α_i are then given as

$$\alpha_1 = \frac{\text{area}342}{\text{area}123} = \frac{\text{length}42}{\text{length}12} = -\frac{k_1}{k_3} \quad ; \quad \alpha_2 = -\frac{k_2}{k_3} \tag{3.44}$$

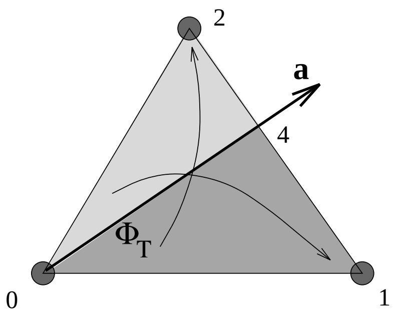


Figure 3.8: In the LDA-scheme the residual is split according to the areas cut by the advection vector.

which leads to the scheme

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t \left[\frac{k_1 k_2}{k_1 + k_2} (u_2 - u_3) + \frac{k_1^2}{k_1 + k_2} (u_1 - u_3) \right] \tag{3.45}$$

$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t \left[\frac{k_2^2}{k_1 + k_2} (u_2 - u_3) + \frac{k_1 k_2}{k_1 + k_2} (u_1 - u_3) \right] \tag{3.46}$$

Linear schemes of the type $\alpha_i \neq f(u_i)$ have some very interesting properties if used on a regular mesh like the one given in figure 3.9. Three of the six contributing cells are cells with one inflow side and the other three are two-inflow side triangles. As the α_i are independent of u_i, they are constant for the three triangles of the same type. Furthermore, imposing the condition that $\sum \alpha_i = 1$ for each triangle, four different α_i can be chosen for the scheme. Let these be α_1, α_2 for for the type with one-inflow side and α_4, α_5 for the type with two-inflow sides.. The most general cubic polynomial that satisfies the steady state is

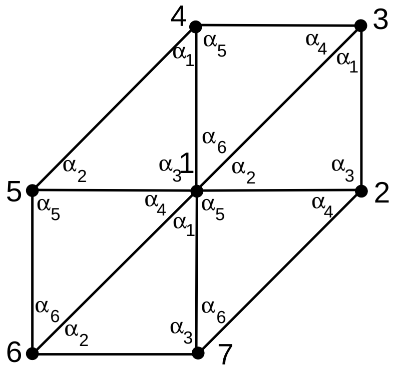


Figure 3.9: Regular mesh of triangles meeting at vertex 1.

$$u(x, y) = e(bx - ay) + f(bx - ay)^2 + g(bx - ay)^3 \tag{3.47}$$

Assembling the contributions from each triangle with a node spacing of $h = 2$ to a Finite-Difference scheme yields the update to node 1 as:

$$\frac{\beta_1}{3g} = \left(\alpha_5 - \alpha_1 + \alpha_4 - \frac{1}{3}\right) ba^3 + \tag{3.48}$$

$$\left(\alpha_1 - \alpha_2 - 2\alpha_4 - \alpha_5 + 1\right) b^2a^2 + \tag{3.49}$$

$$\left(\alpha_2 + \alpha_4 - \frac{2}{3}\right) b^3a. \tag{3.50}$$

As all terms in e and f vanish, an exact quadratic solution is preserved. Thus, any scheme with α_i independent of the data u_i is at least second order accurate in the steady state on a regular mesh.

Considering the case where the diagonals are aligned with the flowfield $b > a > 0$, the three triangles below a diagonal are single target cases, the ones above the diagonals are two target cases. The upwind philosophy requires that $\alpha_2 = \alpha_3 = \alpha_6 = 0$, $\alpha_1 = 1$ and $\alpha_4 + \alpha_5 = 1$. Note that with this choice the stencil for node 1 consists only of five nodes. For β_1 to vanish, it is needed that

$$\alpha_4 = \frac{2}{3} - \frac{1}{3}\frac{a}{b} \quad ; \quad \alpha_5 = \frac{1}{3} + \frac{1}{3}\frac{a}{b} \tag{3.51}$$

This unique third order accurate distribution scheme, however, is not C^0-continuous as can be seen for vanishing a, which in practice would lead do undesirable convergence properties. Note that there is a unique second order scheme using a minimum stencil of four nodes [60]. Hence, the five node-stencil for the Low-Diffusion schemes is the minimum required for a class of second order schemes.

3.2.5 The PSI-Scheme: another Ponlinear Positive Scheme that is LP

As in one-dimensional TVD Finite-Volume schemes, nonlinear schemes have to be resorted to in order to obtain positivity and linearity preservation at the same time.

The basic idea of non-linear Fluctuation-Splitting schemes is the fact that any component $\lambda\mathbf{a}^{\perp}$ can be added normal to ∇u to the advection speed $\mathbf{a}$ without changing the differential equation (fig. 3.10) since

$$\left(\mathbf{a} + \lambda\mathbf{a}^{\perp}\right)\nabla u = \mathbf{a}\cdot\nabla u + \lambda\mathbf{a}^{\perp}\cdot\nabla u = \mathbf{a}\cdot\nabla u. \tag{3.52}$$

But the scheme is affected as the distribution of the fluctuation changes.

Three variations with a slightly different optimization exist, the ‘level-scheme’, the NN-scheme [23] and the PSI-scheme, their results and convergence being almost identical. Only the coefficients of the PSI-scheme however depend continuously on the data and on the advection speed, which is important for robust convergence.

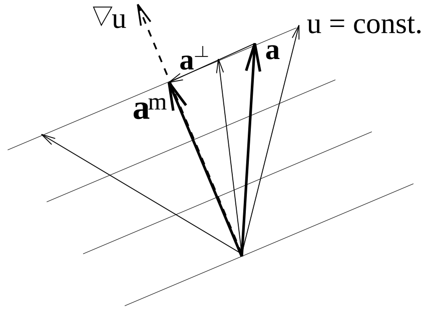


Figure 3.10: Any component $\lambda \mathbf{a}^{\perp}$ can be added normal to the gradient of the solution ∇u without changing the residual. A specific choice is to align the advection speed $\mathbf{a}^m$ with the gradient to obtain the frontal wave speed.

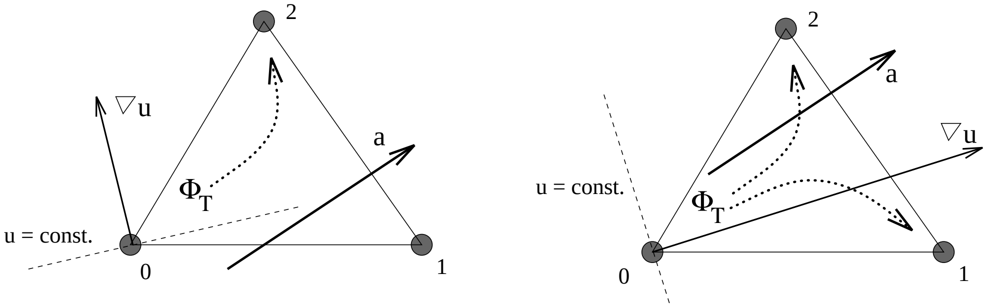


Figure 3.11: The PSI-scheme is one-target if the level-line of the solution does intersect the outflow side, otherwise it is two-target and reverts to the N-scheme.

The PSI-scheme is named for its 'positive streamwise invariance' [24]. In the two-target case chosen on a, the PSI-scheme considers advection with the speed a^m that is aligned with the gradient ∇u, resulting in a one-target scheme as long as ∇u drawn through the upstream vertex does not intersect the outflow side. Otherwise the PSI scheme reverts to the N-scheme:

$$S_1 u_1 \longleftarrow S_1 u_1 + \frac{k_1 (u_1 - u_0)}{k_1 (u_1 - u_0) + k_2 (u_2 - u_0)} \Delta t \phi_T \tag{3.53}$$

$$S_2 u_2 \longleftarrow S_2 u_2 + \frac{k_2 (u_2 - u_0)}{k_1 (u_1 - u_0) + k_2 (u_2 - u_0)} \Delta t \phi_T \tag{3.54}$$

The loss of accuracy due to choosing a first order scheme in this case is barely perceptible in the solutions since at convergence the gradient and advection vectors become normal and the scheme is one-target except in shocks. There, for the sake of monotonicity, the accuracy has to be given up anyhow.

An alternative way of deriving a class of non-linear schemes has been presented by Sidilkover [12] that leads to a very compact and elegant formulation. Recall the definition of the N-scheme in eqs. 3.28 and 3.29:

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t k_1 (u_1 - u_3) = S_1 u_1 - \Delta t \phi_1,$$
$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t k_2 (u_2 - u_3) = S_2 u_2 - \Delta t \phi_2.$$

With a symmetric limiter function Ψ based on the ratio of the two residuals Q,

$$Q = -\frac{\phi_1}{\phi_2} = -\frac{\phi_1}{\phi_2}\frac{\Psi(Q)}{\Psi(Q)}, \tag{3.55}$$
$$\phi_2 \Psi(Q) = -\phi_1 \frac{\Psi(Q)}{Q}, \tag{3.56}$$

a limited contribution of the other residual can be added to the update in order to increase the accuracy of the scheme,

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t(\phi_1 + \Psi(Q)\phi_2), \tag{3.57}$$
$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t(\phi_2 - \Psi(Q)\phi_2), \tag{3.58}$$

and using the symmetry of the limiter for the first equation,

$$S_1 u_1 \longleftarrow S_1 u_1 - \Delta t \left(1 - \frac{\Psi(Q)}{Q}\right) k_1(u_1 - u_3), \tag{3.59}$$
$$S_2 u_2 \longleftarrow S_2 u_2 - \Delta t\,(1 - \Psi(Q))\, k_2(u_2 - u_3). \tag{3.60}$$

The conservation property is obvious from (3.57) and (3.58). Local positivity is given for a choice of limiter function that satisfies

$$0 \leq \Psi(Q) \leq 1, \qquad 0 \leq \frac{\Psi(Q)}{Q} \leq 1 \tag{3.61}$$

which is follows from (3.59) and (3.60). The condition for local positivity can be relaxed if the contributions from all triangles to the node are considered. This 'global positivity' argument allows the use of compressive limiters [75], $\Psi(Q) \leq 2$. Linearity preservation (LP) implies that no update is sent if the fluctuation is zero:

$$\phi = \phi_1 + \phi_2 = 0 \qquad \Leftrightarrow \qquad \phi_1 = -\phi_2 \qquad \Leftrightarrow \qquad Q = 1 \tag{3.62}$$

Thus, LP is achieved for

$$\Psi(1) = 1. \tag{3.63}$$

The PSI scheme is recovered if the limiter is chosen as the MinMod limiter

$$\Psi(Q) = \frac{1}{2}(1 + \mathrm{sgn}(r))\mathrm{min}(r, 1). \tag{3.64}$$

(The exercise of demonstration by computation is left to the reader, or to [34]).

3.3 The Lax-Wendroff Scheme for an Elliptic System

The treatment of the elliptic subsystems in 2.18 and 2.22 requires a scheme with an omnidirectional domain of dependence, since there are no preferred directions to upwind along. A central Galerkin-scheme that distributes equally to all vertices of a cell is marginally stable. While it is suitable for equations with sufficient physical dissipation, it does not exhibit enough dissipation for systems with discontinuous coefficients.

A simple isotropic way to add dissipation is the one chosen by Jameson [64] using second and fourth order isotropic dissipation terms that are added to the Galerkin discretization. A more sophisticated treatment is the "Streamwise-Upwind-Petrov-Galerkin" scheme proposed by Hughes, Johnson et al. [16,17,76], that adds dissipation in the streamwise direction only.

In the following a Lax-Wendroff-type scheme is derived by imposing the equivalent partial differential equation:

$$-\mathbf{U}_t = \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} + \frac{\omega h}{2}\left(\mathbf{A}\frac{\partial}{\partial x} + \mathbf{B}\frac{\partial}{\partial y}\right)\left(\mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y}\right). \tag{3.65}$$

Thus, the added dissipation is scaled with the cell residual and behaves as a Laplacian.

Both decompositions of the Euler equations 2.22 and 2.18 reduce to a form of the Cauchy-Riemann equations, thus

$$\mathbf{U} = \begin{pmatrix} u \\ v \end{pmatrix}, \qquad \mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{3.66}$$

and

$$-\mathbf{U}_t = \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} + \frac{\omega h}{2}\left(\mathbf{I}\mathbf{U}_{xx} + \mathbf{I}\mathbf{U}_{yy}\right) \tag{3.67}$$

$$-\mathbf{U}_t = \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} + \mathbf{B}\frac{\partial \mathbf{U}}{\partial y} + \frac{\omega h}{2}\nabla^2\mathbf{U}. \tag{3.68}$$

Integrating the contribution from a triangle T to one of its forming vertices i (figure 3.12), one obtains

$$\begin{aligned}\int_\Omega \mathbf{U}_t dA = & -\int_\Omega \left(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y\right) dA \\ & + \frac{\omega h}{2}\int_\Omega \mathbf{A}\left(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y\right) dy \\ & + \frac{\omega h}{2}\int_\Omega \mathbf{B}\left(\mathbf{A}\mathbf{U}_x + \mathbf{B}\mathbf{U}_y\right) dx\end{aligned} \tag{3.69}$$

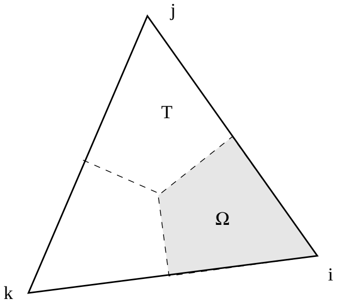


Figure 3.12: The integral $\int \mathbf{U}_t$ over Ω of the triangle T contributes to vertex i.

and with mass-lumping of the time derivative this becomes

$$S_i\mathbf{U}_{t,i} = \frac{S_T}{3}\mathbf{U}_t + \frac{\omega h}{2}\left(\int_\Omega \mathbf{A}dy - \int_\Omega \mathbf{B}dx\right)\mathbf{U}_t \tag{3.70}$$

$$S_i\frac{\Delta\mathbf{U}_i}{\Delta t} = \left(\mathbf{I}\frac{S_T}{3} + \frac{\omega h}{2}(\mathbf{A},\mathbf{B})\cdot\frac{\mathbf{n}_i}{2}\right)\mathbf{U}_t \tag{3.71}$$

or in terms of the distribution coefficients

$$\begin{aligned}\alpha_i =& \mathbf{I}\frac{S_T}{3} + \frac{\omega h}{2}(\mathbf{A},\mathbf{B})\cdot\frac{\mathbf{n}_i}{2}\\ =& \mathbf{I}\frac{S_T}{3} + \frac{\nu_c}{4}\begin{bmatrix} n_x & n_y \\ n_y & -n_x\end{bmatrix}\end{aligned} \tag{3.72}$$

A different way of deriving this scheme is shown in [77–79] starting from a Lax-Wendroff formulation. A second cell-based time-step ν_c is introduced that is equivalent to the ωh term in equation 3.72. Increasing this time-step compared to the nodal time-step increases the dissipation. The stability limits for the time-steps are given by [78] as

$$\nu_c\nu_n \leq .5 \qquad \text{and} \qquad \nu_c \geq \nu_n. \tag{3.73}$$

The definition of ν_T in cell T is chosen as

$$\nu_T = \frac{\Delta t\lambda}{h_i}, \qquad \text{with} \quad h_i = \min|n_i^T|, \quad i = [0,2] \tag{3.74}$$

Although the dissipation does not introduce coupling in the supersonic regime for the acoustic subsystem [34], the control over the dissipation is not optimal as opposed to the one found in the Upwind schemes.

Application of Multigrid methods to the elliptic part favors schemes with strong damping of high frequency modes. The damping rates over the entire frequency spectrum for

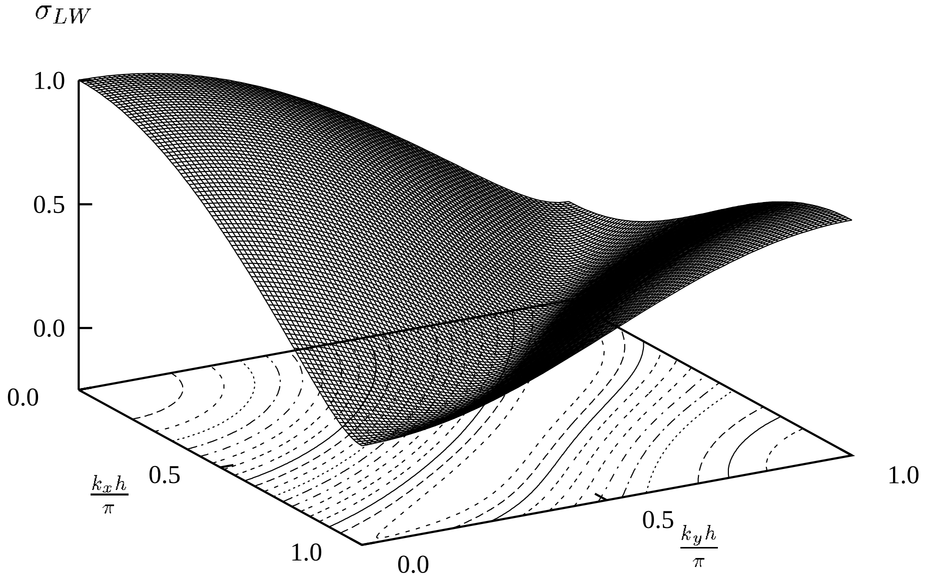


Figure 3.13: Contour plot of the error damping of the Lax-Wendroff scheme over the entire frequency range for the CFL-number combination $\nu_c = 1, \nu_n = .43$, taken from [33].

the above form of the Lax-Wendroff scheme has been analyzed in [33] and a plot of the amplification rate is shown in figure 3.13 for a typical CFL-number combination.

The strongest damping is achieved for

$$\rho(G) = |1 - \frac{24\nu_c^2}{15\nu_c^2 - 1}| \qquad \text{and} \qquad \nu_n = \frac{6\nu_c}{15\nu_c^2 - 1}. \tag{3.75}$$

leading to $\nu_c \to \infty$ and $\nu_n\nu_c \to .4$ with $\rho(G) = .6$. However, this analysis focusing on the damping of errors disregards effects of advection of errors to the boundaries and then out of the domain. In practice, the combination $\nu_c = 1, \nu_n = .5$ resulting in $\rho(G) = .71$ yields the best results when applied to the Cauchy-Riemann equations as will be seen in section 6.2.

3.4 Conservative Linearization

In the preceding discussions the equations were assumed to be linear. In order to be able to deal with nonlinear problems, a linearization has to be performed that remains conservative, that is

$$\phi_T = \oint_{\partial T} \mathbf{F} d\mathbf{n} = \iint_T \nabla \cdot \mathbf{F} dx dy \tag{3.76}$$

In the case of a scalar equation 3.7, this can be achieved efficiently by defining a linearized advection speed

$$\bar{\mathbf{a}} = \frac{1}{S_T} \iint_T \mathbf{a} dx dy \tag{3.77}$$

which leads to

$$\phi_T = \iint_T \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \nabla u dx dy = \bar{\mathbf{a}} \nabla u \iint_T dx dy = S_T \bar{\mathbf{a}} \nabla u \tag{3.78}$$

In a very similar fashion, treating the system case requires to define linearized Jacobians $\bar{\mathbf{A}}, \bar{\mathbf{B}}$ such that the following holds:

$$\Phi_T = \oint_{\partial T} \mathbf{F} dn_x - \mathbf{G} dn_y = \iint_T \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{y}} \right) d\Omega = S_T \left(\bar{\mathbf{A}} \frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \bar{\mathbf{B}} \frac{\partial \mathbf{U}}{\partial \mathbf{y}} \right) \tag{3.79}$$

However the straightforward evaluation of the integral of the Jacobian

$$\bar{\mathbf{A}} = \frac{1}{S_T} \iint_T \mathbf{A} \; dx dy \tag{3.80}$$

is too cumbersome for the Euler equations due to the nonlinear terms it contains. A simpler procedure presented in [80] is obtained by using Roe's parameter vector $\mathbf{Z}$ [3]

$$\mathbf{Z} = \sqrt{\rho}(1, u, v, H)^T. \tag{3.81}$$

With this choice the vector of conservative unknowns $\mathbf{U}$ and the conservative flux vectors $\mathbf{F}$ and $\mathbf{G}$ are quadratic in $\mathbf{Z}$, thus the Jacobians $\frac{\partial \mathbf{U}}{\partial \mathbf{Z}}$, $\frac{\partial \mathbf{F}}{\partial \mathbf{Z}}$ and $\frac{\partial \mathbf{G}}{\partial \mathbf{Z}}$ are linear in $\mathbf{Z}$ and an average gradient can be defined as

$$\nabla \bar{\mathbf{U}} = \frac{1}{S_T} \iint_T \nabla \mathbf{U} dA = \frac{1}{S_T} \iint_T \frac{\partial \mathbf{U}}{\partial \mathbf{Z}} \nabla \mathbf{U} dA = \left. \frac{\partial \mathbf{U}}{\partial \mathbf{Z}} \right|_{\bar{\mathbf{Z}}} \nabla \mathbf{Z} \tag{3.82}$$

and similarly for $\nabla \mathbf{F}$ and $\nabla \mathbf{G}$ using the average parameter state

$$\bar{\mathbf{Z}} = \frac{\mathbf{Z}_i + \mathbf{Z}_j + \mathbf{Z}_k}{3} \tag{3.83}$$

This leads to the fluctuation expressed as

$$\Phi_T = \iint_T \left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{y}} \right) dA = \iint_T \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{y}} \right) dA \tag{3.84}$$

$$= \iint_T \left(\frac{\partial \mathbf{F}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \frac{\partial \mathbf{G}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial \mathbf{y}} \right) dA \tag{3.85}$$

$$= S_T \left(\bar{\mathbf{A}} \frac{\partial \mathbf{U}}{\partial \mathbf{x}} + \bar{\mathbf{B}} \frac{\partial \mathbf{U}}{\partial \mathbf{y}} \right) \tag{3.86}$$

with the average Jacobians evaluated at the average parameter state $\bar{\mathbf{Z}}$

$$\bar{\mathbf{A}} = \left.\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right|_{\bar{\mathbf{Z}}}, \qquad \bar{\mathbf{B}} = \left.\frac{\partial \mathbf{G}}{\partial \mathbf{U}}\right|_{\bar{\mathbf{Z}}}. \tag{3.87}$$

Aspects of linearization are extensively discussed in [34].

3.5 Time Discretization

The steady-state solution of the Euler equations is sought here by applying pseudo-time stepping concepts. The decoupling of the equations using the preconditioning approach 2.22 or the canonical form 2.18 demands two seemingly conflicting optimizations from the time-stepping scheme employed. On the one hand a scheme with good advection properties is needed for the advection-dominated part, namely the entropy and enthalpy equations. Ideally, a space-marching scheme that traces out the characteristics is sought. On the other hand, schemes that emphasize damping are required for the elliptic kernel, especially when opting for Multigrid convergence acceleration. However, both the damping of errors and the advection of errors out of the domain work in common to reach convergence. The analysis of the interaction of both processes has not been undertaken as of 1995.

The trade-off between damping and advection can nicely be illustrated with the optimal multistage schemes for explicit time-stepping [62,64,81]. The optimization is achieved by matching the damping of the time stepping operator to the Fourier footprint of the spatial operator, thus requiring a special set of multistage coefficients for each discretization. The sum of these coefficients is lower than e.g. the same number of Forward-Euler time-steps, thus reducing convection. Multistage time-stepping is widely used for pseudo-time-stepping methods due to its simplicity and stability. The implementation can be done efficiently by first collecting all residuals for all nodes and then performing the update.

For purely elliptic problems in a Multigrid context, Gauß-Seidel relaxation outperforms Jacobi relaxation by a factor of 2 [82] due to its superior smoothing characteristics and is a standard iterative method for such problems. As numerical results for scalar equations reveal (sec. 6.1), Gauß-Seidel can also be an excellent time-stepping scheme for advection equations when combined with an Upwind scheme. For the case of linear advection, a space discretization with the N-scheme combined with Gauß-Seidel time-stepping and a downstream ordering of the cells results in a direct solver. Sidilkover presented results [12] for the Euler equations using multidimensional schemes and Gauß-Seidel time-stepping apparently resulting in fast convergence.

However, the implementation of Gauß-Seidel schemes in the context of a Cell-Vertex discretization is less efficient than the Jacobi scheme. In order to calculate a conservative flux update at each node, the fluctuations of all the cells formed by that node have to be collected. Since the computational cost of the method is dominated by the linearization of the equations, their decomposition and the distribution of the cell residuals, all of which are costs associated to the number of cells, one Gauß-Seidel pass is three times as expensive in CPU as one Forward-Euler pass. Moreover, in general symmetric sweeps have to be applied to equally treat all directions, resulting in a six-fold increase of the work.

All of the convergence studies with Gauß-Seidel and Forward-Euler schemes here are to be taken as preliminary results since the nature and behavior of the coupling of the parts of the non-linear system is not understood. Thus no optimization of any parameters can be suggested and detailed CPU-time comparisons are omitted. Customarily Multigrid work-units are defined for Gauß-Seidel schemes as one symmetric sweep and for multistage schemes as one iteration on the finest grid. This difference in notation is retained here, although it leads to a discrepancy of 6:1 in terms of CPU-time between the schemes.

Chapter IV

Multigrid Convergence Acceleration

"Zeit ist ein Abgrund, Tausend Nächte tief."

Klaus Kinski in Herzogs 'Nosferatu'

A close look at figure 3.13 reveals the main motivation behind applying Multigrid methods to relaxation problems: while the high frequencies are well damped and a relaxation scheme is very effective for these modes, the low frequencies experience very little damping from the scheme. This is characteristic of all relaxation methods.

The essence of the Multigrid concept is then to approximate the lower frequencies on a coarser mesh where they become modes of a higher frequency and accessible to relaxation. Figure 4.1 shows the basic principle. With the wavenumber $k, 0 \leq {}_{ih}k \leq {}_{ih}N$ on a grid of level ih with ${}_{ih}N$ vertices and the non-dimensional frequency $\Theta, 0 \leq {}_{ih}\Theta = {}_{ih}k\pi/{}_{ih}N \leq \pi$, a Fourier-mode is given by

$${}_{ih}w_k = \sin\left(\frac{{}_{ih}k\pi j}{{}_{ih}N}\right) = \sin({}_{ih}\Theta j), \qquad 0 \leq j \leq {}_{ih}N. \tag{4.1}$$

Since smooth modes have to be resolved on a grid twice as coarse, a smooth mode is given by ${}_{ih}\Theta < \frac{\pi}{2}$. Conversely, oscillatory modes are found with ${}_{ih}\Theta \geq \frac{\pi}{2}$. Figure 4.1 shows a smooth level h solution with ${}_{h}\Theta \approx \frac{\pi}{6}$ that remains smooth on level $2h$, ${}_{2h}\Theta \approx \frac{\pi}{3}$. Further coarsening of the solution increases the frequency to ${}_{4h}\Theta \approx \frac{2\pi}{3}$, an oscillatory mode that can be effectively damped with relaxation. Note that taking this oscillatory solution to an even coarser grid level produces a low-frequency mode on the coarse grid that was not present in the oscillatory fine grid solution. Moreover, since it is a low frequency mode it can not effectively be damped with relaxation and is thus to be considered an

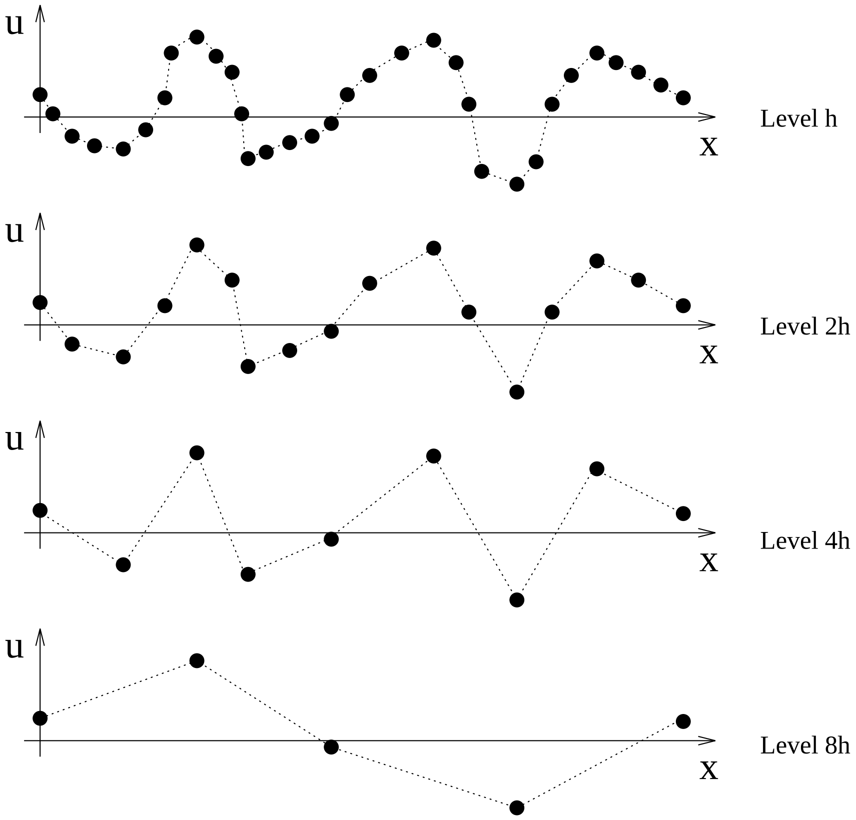


Figure 4.1: The frequency of the same spatial variation of the data increases on a coarser mesh.

expensive mode. This ‘aliasing’ has to be avoided, the solution on the finer grid has to be smoothened in its high frequency contents before restricting it to a coarser mesh.

Once the solution is smoothed on the coarse mesh, the error is “prolongated” back to the fine grid. Thus, the basic procedure is the following:

1. apply smoothing to the fine grid solution until the oscillatory modes have been eliminated,

2. restrict the solution and the residual to the coarse grid,

3. apply smoothing to the coarse grid solution until the oscillatory modes have been eliminated,

4. prolongate the error to the fine grid.

Naturally, the process is applied recursively through a couple of layers of coarser grids to affect all frequency modes in the solution.

4.1 The Full Approximation Storage Scheme

In the "Full Approximation Storage" scheme, (FAS), a full solution is stored on the coarser grid levels. This allows the treatment of nonlinear problems with a Multigrid method,

$$\mathbf{x}_t + \mathbf{N}(\mathbf{x}) = \mathbf{f}, \tag{4.2}$$

with the vector of unknowns $\mathbf{x}$, the nonlinear operator $\mathbf{N}$ and a source term $\mathbf{f}$, here augmented by a [pseudo-]time-derivative, but the steady state of the problem is sought:

$$\mathbf{N}(\mathbf{x}) = \mathbf{f} \tag{4.3}$$

The sign of the residual $\mathbf{r}$ is chosen as

$$\mathbf{x}_t = \mathbf{r} = -\left(\mathbf{N}(\mathbf{x}) - \mathbf{f}\right) \tag{4.4}$$

$$x_i^{n+1} = x_i^n + \frac{\nu \Delta t}{S_i} r_i \tag{4.5}$$

$$\tag{4.6}$$

The discretization of equation 4.3 on the finest mesh, level h, yields

$${}_h\mathbf{N}({}_h\mathbf{x}) = {}_h\mathbf{f} \tag{4.7}$$

which becomes with the definition of the error $\Delta\mathbf{x}$ of the current approximation $\tilde{\mathbf{x}}$

$${}_h\mathbf{x} = {}_h\tilde{\mathbf{x}} + {}_h\Delta\mathbf{x} \tag{4.8}$$

$${}_h\mathbf{N}({}_h\tilde{\mathbf{x}} + {}_h\Delta\mathbf{x}) = {}_h\mathbf{f}. \tag{4.9}$$

The residual of the current solution is

$${}_h\mathbf{N}({}_h\tilde{\mathbf{x}}) - {}_h\mathbf{f} = -{}_h\mathbf{r}. \tag{4.10}$$

Subtracting equation 4.10 from equation 4.9 yields an expression for the residual

$${}_h\mathbf{N}({}_h\tilde{\mathbf{x}} + {}_h\Delta\mathbf{x}) - {}_h\mathbf{N}({}_h\tilde{\mathbf{x}}) = {}_h\mathbf{r}. \tag{4.11}$$

This equation can be restricted to the coarser grid $H = 2h$ to read

$${}_H\mathbf{N}({}_h^H\mathbf{I}\,{}_h\tilde{\mathbf{x}} + {}_H\Delta\mathbf{x}) - {}_H\mathbf{N}({}_h^H\mathbf{I}\,{}_h\tilde{\mathbf{x}}) = {}_h^H\mathbf{I}'\,{}_h\mathbf{r}. \tag{4.12}$$

The restriction operator for the solution ${}_h^H\mathbf{I}$ and the one for the residual ${}_h^H\mathbf{I}'$ don't have to be identical but they are chosen here to be the same as presented in section 4.3 and the prime will be omitted in the following. Defining the coarse grid solution ${}_H\mathbf{x}$ as

$${}_H\mathbf{x} = {}_h^H\mathbf{I}\,{}_h\tilde{\mathbf{x}} + {}_H\Delta\mathbf{x} \tag{4.13}$$

with the coarse grid correction ${}_H\Delta\mathbf{x}$, the equation to be solved on the coarse grid is obtained as

$$ {}_H\mathbf{N}({}_H\mathbf{x}) = {}^H_h\mathbf{I}\,{}_h\mathbf{r} + {}_H\mathbf{N}({}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}) = {}_H\mathbf{f} \tag{4.14} $$

or in an explicit-time-stepping form

$$ \begin{aligned} {}_Hx_i^{n+1} &= {}_Hx_i^n + \frac{\nu\Delta t}{S_i}\left({}^H_h\mathbf{I}\,{}_hr_i + {}_H\mathbf{N}({}^H_h\mathbf{I}\,{}_h\tilde{x}_i^n) - {}_H\mathbf{N}({}_Hx_i^n)\right) \\ &= {}_Hx_i^n + \frac{\nu\Delta t}{S_i}\left({}_Hf_i - {}_H\mathbf{N}({}_Hx_i^n)\right). \end{aligned} \tag{4.15} $$

After solving for the correction ${}_H\Delta\mathbf{x}$ on the coarse grid

$$ {}_H\Delta\mathbf{x} = {}_H\mathbf{x} - {}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}} \tag{4.16} $$

the correction can be “prolongated” to the fine grid by

$$ {}_h\tilde{\mathbf{x}} \longleftarrow {}_h\tilde{\mathbf{x}} + {}^H_h\mathbf{I}\,{}_H\Delta\mathbf{x} \tag{4.17} $$

Note that on convergence of the fine grid solution, the coarse grid error vanishes. That is, the coarse grid solution is driven by the fine grid residual. With

$$ {}_h\mathbf{N}_h\tilde{\mathbf{x}} = {}_h\mathbf{f} \qquad \text{and} \qquad {}_h\mathbf{r} = 0 $$

equation 4.14 becomes

$$ \begin{aligned} {}_H\mathbf{N}({}_H\mathbf{x}) &= {}_H\mathbf{N}\left({}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}\right), \\ {}_H\mathbf{x} &= {}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}, \\ {}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}} + {}_H\Delta\mathbf{x} &= {}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}, \end{aligned} $$

thus

$$ {}_H\Delta\mathbf{x} = 0. $$

Another point of view to be taken can be viewing the fine grid solution as a corrective to the coarse grid solution. Starting from equation 4.14, one finds

$$ \begin{aligned} {}_H\mathbf{N}({}_H\mathbf{x}) &= {}^H_h\mathbf{I}\,{}_h\mathbf{r} + {}_H\mathbf{N}({}^H_h\mathbf{I}\,{}_h\mathbf{x}) \\ &= {}^H_h\mathbf{I}\left({}_h\mathbf{f} - {}_h\mathbf{N}({}_h\tilde{\mathbf{x}})\right) + {}_H\mathbf{N}({}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}) \\ &= {}^H_h\mathbf{I}\,{}_h\mathbf{f} - {}^H_h\mathbf{I}\,{}_h\mathbf{N}({}_h\tilde{\mathbf{x}}) + {}_H\mathbf{N}({}^H_h\mathbf{I}\,{}_h\tilde{\mathbf{x}}) \\ &= {}_H\mathbf{f}^* + \tau \end{aligned} \tag{4.18} $$

Thus, the source term of the coarse grid solution can be viewed as the restriction of the fine grid source term ${}_H\mathbf{f}^*$ and a corrective term that is the difference τ between the restriction of the fine grid residual and the coarse grid residual of the restriction of the fine grid solution. On convergence of the fine grid solution, τ changes the coarse grid solution that it coincides with the fine grid solution. Solution adaptive processes might be able to use τ as an indicator of grid convergence of the solution.

In case the operator $\mathbf{N}$ is linear, $\mathbf{N} = \mathbf{L}$, one recovers the "correction scheme" (CS) where only errors are dealt with on coarser grid levels. Equation 4.14 then simplifies to

$$ {}_H\mathbf{L}_H\mathbf{x} = {}^H_h\mathbf{I}\,{}_h\mathbf{r} +_H \mathbf{L}^H_h\mathbf{I}\,{}_h\mathbf{x} \tag{4.19}$$

$$ {}_H\mathbf{L}\left({}_H\mathbf{x} - {}^H_h\mathbf{I}\,{}_h\mathbf{x}\right) = {}^H_h\mathbf{I}\,{}_h\mathbf{r} \tag{4.20}$$

$$ {}_H\mathbf{L}\left({}^H_h\mathbf{I}\,{}_h\mathbf{x} + {}_H\Delta\mathbf{x} - {}^H_h\mathbf{I}\,{}_h\mathbf{x}\right) = {}^H_h\mathbf{I}\,{}_h\mathbf{r} \tag{4.21}$$

$$ {}_H\mathbf{L}_H\Delta\mathbf{x} = {}^H_h\mathbf{I}\,{}_h\mathbf{r} \tag{4.22}$$

In this form, the fact that the fine grid residual drives the coarse grid solution is most evident.

4.2 Application of Multigrid to Hyperbolic Problems

The previous sections dealt with Multigrid as a concept for solving elliptic problems, that is, the damping of errors was the primary aspect. In the context of the Euler equations however, one is dealing with a system that in the steady state has both elliptic and hyperbolic components.

Previous research has been undertaken to apply Multigrid methods straightforwardly to this mixed problem or even a purely hyperbolic formulation of it. Ground-breaking work has been presented by Jameson [83] and the concept has been carried further by many others on structured [84,85] and unstructured grids [86–88].

The analysis of the Multigrid method applied to hyperbolic problems then focuses more on effects of convection as shown in [89]: errors are advected faster on the coarser grids to be absorbed by the boundaries. Naturally, while the errors are advected on the grid, the high frequencies experience smoothing due to the space-discretization. This is necessary to avoid aliasing.

Although Multigrid provides a worthwhile acceleration of the convergence in these cases, the gains are disappointing for high speed flows: convergence rates for transonic cases of .99 seem to be the accepted performance, compared to .25 in certain elliptic cases. Various methods have been designed to overcome these deficiencies, e.g. by residual over-weighting and defect-correction techniques [90] or by making the transfer operators depend on the flow [91–94].

The fundamental underlying problem however is that for the scalar advection problem that one seeks to reduce the Euler-equations to, Multigrid is not an efficient concept. The method of choice here is of the space-marching type.

Consider the problem of linear advection as presented in section 6.1: the most efficient method that can be conceived has a cost of $O(N)$ associated to it, every node has to be visited once. This is achieved by a space-marching method that orders the nodes according to the streamwise direction and traces the characteristics from the upstream nodes in the domain of dependence toward the node to be updated. During the process of convergence, a front sweeps through the domain that separates the updated, final solution from the initial solution.

A Multigrid method with linear transfer operators ${}_h^H\mathbf{I}$ and ${}_H^h\mathbf{I}$ distributes the correction coming from the coarse grid centrally. That is, even a correctly updated fine grid node will receive a coarse-grid-correction if another node in that is in the restriction stencil of the contributing coarse grid node has an error. While this can be remedied with prolongation operators that distribute only to downstream nodes, the solution convergence will not improve since convergence on the fine grid is obtained only once the characteristics have been traced through the grid in all fine-grid detail.

Naturally, if the problem is mixed hyperbolic-elliptic, accelerating the convergence of the elliptic part might outweigh the the deceleration of the convergence of the hyperbolic part. The Multigrid convergence process of the linear advection problem thus looks very different compared to that of a single-grid method: the main features of the solution appear after one or two cycles, since the solution then already is converged on the coarsest grid, and a gradual sharpening of the features takes place until all details have been resolved on the finest grid (cf. the results in section 6.1).

Using the decompositions presented in section 2.4 and section 2.3, the elliptic problem and the hyperbolic problem become decoupled. This offers the possibility to choose different time and space-discretizations for the advective part and for the acoustic kernel. In the sections 6.1 and 6.2 the suitability of the different time-stepping schemes for the two types of partial differential equations is investigated.

4.3 Restriction and Prolongation Operators

For Cell-Vertex and Finite-Element methods, where the data vary continuously, using the shape function coefficients w_i is a straightforward choice for the restriction and prolongation operators:

$$ {}_H S_i \, {}^H_h \mathbf{I} \, \mathbf{u}_i = \sum_{k=1}^{N} {}_h w_{i,k} \, {}_h S_k \, {}_h \mathbf{u}_k \tag{4.23}$$

$$ {}^H_h \mathbf{I} \, \mathbf{r}_i = \sum_{k=1}^{N} {}_h w_{i,k} \, {}_h \mathbf{r}_k \tag{4.24}$$

$$ {}^h_H \mathbf{I} \, {}_H \Delta \mathbf{x}_i = \sum_{k=1}^{N} {}_H w_{i,k} \, {}_H \Delta \mathbf{x}_k \tag{4.25}$$

This choice satisfies various properties:

- The operators are linear and can be precomputed and stored for efficiency.
- The operators are central.
- The operators accomplish "full weighting", i.e. the integral over the restricted quantity is conserved during the transfer:

$$\int_{{}_H\Omega} {}_h \mathbf{r} d_H \Omega = \int_{{}_h\Omega} {}_h \mathbf{r} d_h \Omega \tag{4.26}$$

  This is important for cases of highly varying coefficients in **N** which means also highly varying **r**.

- The operators are sufficiently accurate. Brandt [82] suggests the condition ${}^H_h m + {}^h_H m > m_N$ with the order of the restriction ${}^H_h m$, the order of the prolongation ${}^h_H m$ and the order m_N of the highest derivative of the nonlinear operator **N**. In the case of full weighting ${}^H_h m +^h_H m = 2 + 2 > m_{Euler} = 1$.
- The restriction operator is the transpose of the prolongation. This condition derives from effects of the aliasing phenomenon: Restricting oscillatory fine grid modes creates smooth coarse grid modes. Conversely, smooth coarse grid modes introduce oscillatory fine grid modes upon prolongation. The magnitude of these undesired modes can be minimized by having both operators being their respective transposes. This is called the Galerkin property. Briggs [95] gives a thorough discussion of the problem. This condition is satisfied by design.

4.4 Cycling

With the transfer operators chosen, the cycling parameters remain to be varied.

The basic Multigrid cycle is the V-cycle (fig. 4.2). This cycle is found naturally when viewing Multigrid as a mere convergence acceleration tool that is applied recursively for

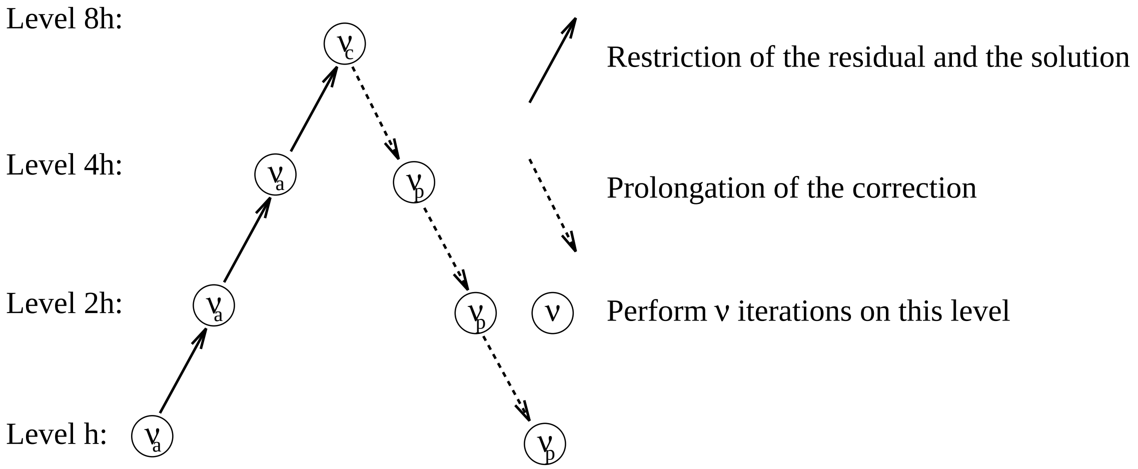

Level 8h:
Level 4h:
Level 2h:
Level h:
ν_c
ν_a
ν_a
ν_a
ν_p
ν_p
ν_p
Restriction of the residual and the solution
Prolongation of the correction
ν
Perform ν iterations on this level

Figure 4.2: The Multigrid V cycle.

several levels of grids. Starting with the current finest-grid solution on level h, ν_a anterior or pre-iterations are performed to smoothen the high frequency content. The solution and the residual are then restricted to the next coarser grid $2h$, and again ν_a iterations are applied to the $2h$-problem with the right-hand-side restriction terms. This is repeated recursively until the coarsest grid is reached where the problem is solved exactly, or nearly so, with ν_c iterations. The error of that solution is prolongated to the next finer grid where ν_p post-iterations are applied to smoothen the high frequencies aliased from low-frequency error-components of the coarse grid. This is done recursively until the finest grid is reached. A cycle with no post-iterations is called a 'saw-tooth' cycle.

Chapter V

Unstructured Triangular Grids

"Triangle man, triangle man,
triangle man hates particle man.
They have a fight.
Triangle wins. Triangle man."

They might be Giants, 'Particle man'.

"On a dit fort bien que, si les triangles se faisoient un dieu,
ils lui donneraient trois côtés."

Montesquieu, 'Lettres persanes'

5.1 The Delaunay Triangulation

The principle of the Delaunay triangulation (DT) [44–46] is beautifully simple, and, as will be shown, amazingly powerful. Given a set of vertices the convex hull around the vertices (imagine the vertices were nails in a board and a rubber band was wrapped around the bunch of nails), the domain is tessellated such that each vertex is assigned the area that is closer to the vertex than to any other vertex. This tessellation is called the Dirichlet tessellation and the set of straight edges that delineate the borders between the different tiles is the Voronoi diagram (fig. 5.1). The rule of connection is to connect those vertices whose regions in the Dirichlet tessellation are adjacent or, in other terms, who share an edge of the Voronoi diagram. These edges are part of the median of the edges in the Delaunay triangulation, and thus for each triangle there exists a point where the three edges of the Voronoi diagram intersect, the Voronoi vertex. This point is, by construction the center of the circle that goes through the three forming vertices of the triangle.

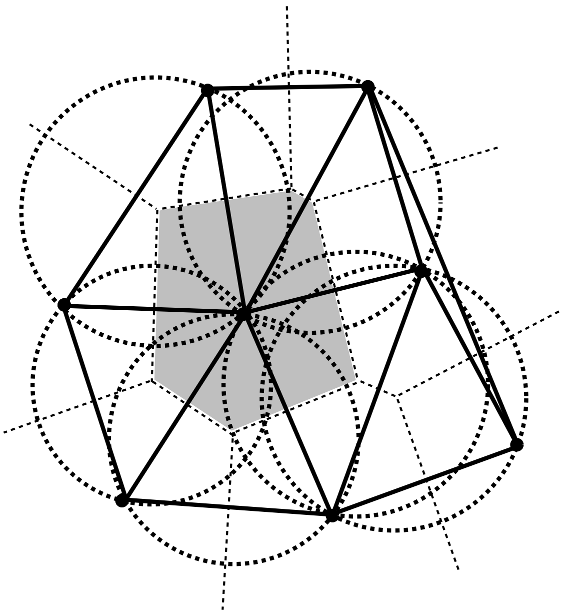


Figure 5.1: A set of vertices with the Dirichlet tessellation (shaded), the Voronoi diagram (dashed lines), the circumcircles (dashed) and the Delaunay Triangulation (solid lines).

5.2 Quality of the Delaunay Triangulation

Let a few geometric properties of the Delaunay triangulation from the literature be recalled (Lee and Schachter give a nice review in [96]):

- the DT is unique, except for degenerate cases of more than three vertices that are cocircular,
- the circumcircle around a triangle does not contain any other vertex,
- the DT is a Max-Min triangulation: it maximizes the minimum angles in the grid,

Among the many interesting mathematical properties (see e.g. Barth [47]) is the fact that of all triangulations of a given point set with random data f_i the Sobolev semi-norm

$$\int_K \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right] dx dy \tag{5.1}$$

is minimized on the Delaunay triangulation as shown by Rippa [97]. That is, in absence of any known directionality of the data or expected solution, a DT will minimize the interpolation error of linear Finite Elements. As to be expected, for strongly non-isotropic fields a modified triangulation with short sides aligned with the strongest second derivatives and long sides normal to them achieves the optimum [98].

These results make it immediately clear that the Advancing Front method (AFM) [42] cannot be optimal in the aforementioned sense as long as the triangulation contains non-DT edges. A few general aspects of grid quality are being highlighted in the next sections.

5.2.1 Maximum Angle Condition

Probably the most classic grid quality criterion has been derived by Babuška and Aziz [99]. They show that the error of interpolation in a linear Finite Element method increases with the maximum angle of the element

$$\| u - Ru \|_{H^1(T)} \leq \Gamma(\alpha) h \| u \|_{H^2(T)} \tag{5.2}$$

where u is the evolution, Ru is a linear projection on the element, $\| \ \|_{H^I(T)}$ are the usual norms in Sobolev spaces, h is the largest edge length of the element and $\Gamma(\alpha)$ is an increasing finite function for $\pi/3 \leq \alpha \leq \pi$. This is to say that the maximum angle of the cell should remain as small as possible.

5.2.2 Regularity

A very interesting result on the effects of regularity has been shown by Roe [100, 101]. By Finite Difference analysis applied to Cell Vertex methods, i.e. the Finite Element type data representation of unknowns at the grid vertices and linear variation inbetween, he showed that the residual $\int \mathbf{a} \cdot \nabla u$ is evaluated with second order accuracy only if the elements connected to a vertex form a regular n-gon with $n = 4$ or $n = 6$. Regular means here that the domain can be tiled with elements of the same shape. In practice this means for triangulations that it is desirable to have six triangles around each vertex with parallel bases on opposite triangles (fig. 5.2).

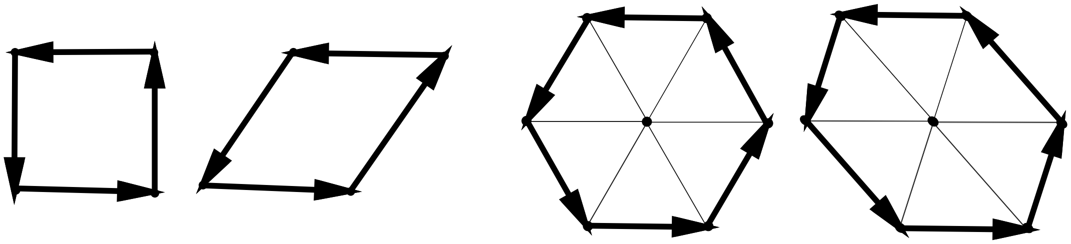


Figure 5.2: A few regular n-gons with n=4,6 that lead to a second order accurate residual calculation with Cell-Vertex schemes.

5.2.3 Size Variation

An intuitive feel for the effect of size variation can be derived by looking at a simple Finite Difference formula, a forward difference of a first derivative.

$$u_x = \frac{u_{i+1} - u_i}{h} \tag{5.3}$$

This gradient approximation is second order accurate only at the midpoint $i + \frac{1}{2}$, hence, one should find the cell interface for the flux calculation exactly at the midpoint between both vertices $i, i+1$. In graded fields variations in element size have to be accepted, but the differences ought to be as small as possible.

5.3 The Frontal Delaunay Method

The Frontal Delaunay method (FroD) incorporates ideas from the frontal vertex placement strategy of the Advancing Front method (AFM) [42] that achieve the regularity and the smooth size variation that is desired into the Delaunay triangulation (DT), providing the high quality point cloud with the optimal connectivity.

In FroD, vertices are generated and inserted in sets of one row at a time like a tree that adds a ring every year as shown in figures 5.3 and 5.13. This special treatment of each row distinguishes it from the independently developed method of Rebay [102] that is otherwise very similar. After the insertion of each vertex, the DT is reestablished such that there exists a valid DT that covers the entire domain at any stage of the algorithm. However, the Delaunay criterion will not necessarily add the triangles between the new vertex and the frontal edge that they are projected from as proposed by the insertion strategy (the explicit triangles in figure 5.3): if a triangle is not part of the DT, it will

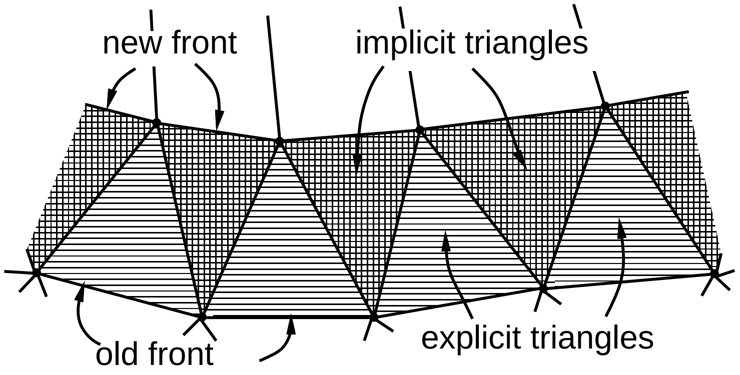


Figure 5.3: Explicit triangles (striped) and implicit triangles (squared) that are formed along the old front and build the new front.

not be formed. Other triangles are formed that close the DT around the newly inserted vertex (the implicit triangles in figure 5.3).

In order to make these implicit triangles well-behaved, a minimum distance between vertices has to be imposed, say a certain fraction ah of the local mesh spacing h. This prevents short edges leading to acute triangles and gives an upper cardinality bound to the triangulation. The required distance is taken from a background grid that specifies the mesh spacing at every location. For a symmetric definition of distance in this non-Euclidean space, the mesh-spacing has to be interpolated at the midpoint between the two points in question. In the case that a new vertex is conflicting with one of a previous row, the new vertex is discarded. In the case that two vertices of the same row are too close, they are merged. This merging makes FroD the only method currently that is not "vertex-greedy".

5.3.1 Generating the Delaunay Triangulation

Delaunay refinement methods insert new vertices recursively into a valid Delaunay triangulation. Thus any method that generates the Delaunay triangulation incrementally is suitable. Many such algorithms have been presented.

Bowyer's algorithm [49] computes and modifies the Voronoi diagram, but is relatively cumbersome to implement.

Another possible choice is Green and Sibson's algorithm [103] or Lawson's algorithm [104] that inserts a new vertex into the containing triangle by connecting it to the three forming vertices or onto an existing edge by connecting it to the four forming vertices of the two triangles that share the edge. Newly created edges are necessarily DT, however adjacent older ones might not and are swapped until the triangulation is again DT.

Watson's algorithm [48] is frequently misquoted as being Bowyer's, but is different and more practical. It exploits the circumcircle criterion directly by finding the region that is covered by all the triangles who contain in their circumcircles the new site that is to be inserted. The cavity can be found by locating a first cell that contains the new vertex and then marching from neighbor to neighbor. All cells of the cavity have to share an edge since the cavity is convex due to the circumcircle criterion. This cavity is retriangulated by removing all the triangles covering it and reconnecting the edges of the cavity with the new vertex to a new valid DT (fig. 5.4). For its simplicity, this is the method that was chosen for FroD. This choice leads to a data structure that stores for each triangle:

- the three forming vertices,
- the three neighboring triangles and
- the circumcenter and -radius.

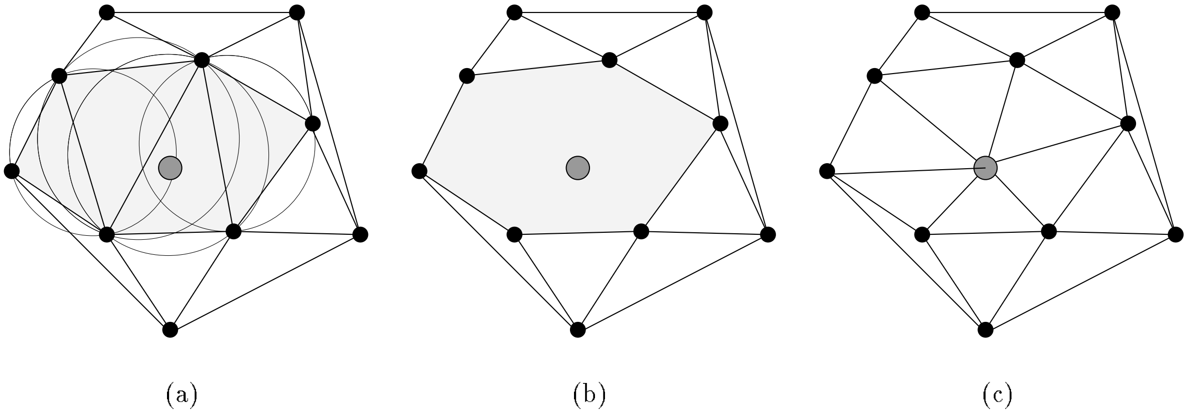


Figure 5.4: Vertex insertion with Watson's algorithm. Part (a) shows the valid Delaunay triangulation with the new vertex that is contained in various circumcircles. The cavity that has to be retriangulated is shown shaded in (b), (c) shows the retriangulated cavity.

All recursive methods have to be started from an initial valid DT, say one or two triangles that cover the entire domain (fig. 5.5(a)). The triangles outside the domain are

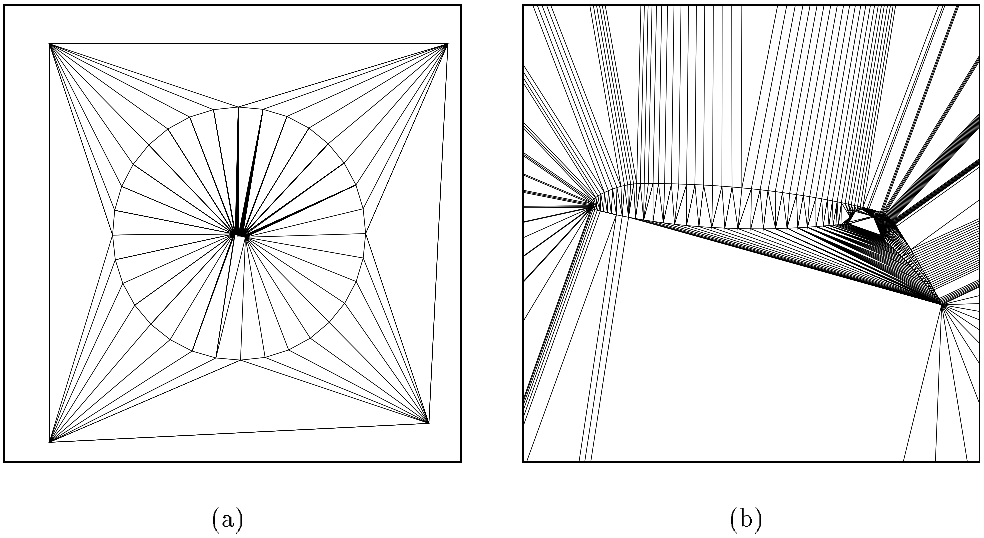


Figure 5.5: The initial triangulation of the boundary vertices is inserted into a valid Delaunay triangulation of four vertices that covers the entire domain (a). The final triangulation of the configuration (b) is shown in figure 5.15.

removed at the end of the triangulation, be they outside the outer boundary or inside of holes in the domain.

5.3.2 The Constrained Delaunay Triangulation

The geometric construction of the Delaunay triangulation does not distinguish between boundary vertices and interior ones and thus does not respect the connectivity between boundary vertices. However, for the purpose of generating computational grids, boundary conformality is indispensable.

A way that ensures boundary conformality is the constrained Delaunay triangulation (CDT). In the CDT, required edges between vertices are added to the set of vertices to be triangulated. These edges truncate the circumcircles in such a way that a vertex that lies on the other side of the edge with respect to the circumcenter is never contained in the circumcircle, thus preventing the formation of an edge that would cross and obliterate the required edge.

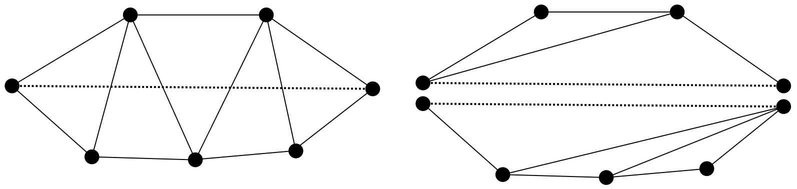


Figure 5.6: A regular triangulation that violates boundary conformality on the left, and the equivalent constrained Delaunay triangulation with a required edge on the right. The required edge blocks the visibility of the circumcircles across the edge, so the two halves of the cavity to either side of the required edge can be triangulated separately.

This is equivalent to finding the cavity that is formed by all cells sharing the edges that cross the required edge, splitting the cavity along the required edge and by retriangulating each part separately (fig. 5.6). Note that the optimal Delaunay properties presented in section 5.2 do not necessarily hold for the CDT.

5.3.3 Searching the Delaunay Triangulation for the Containing Triangle

A frequent search problem that has to be dealt with efficiently in Delaunay refinement methods is finding the cell that contains a certain point. In FroD this has to be done for finding the first triangle of the cavity or for finding the element to interpolate on in the background grid.

A very simple search strategy that does not depend on an overhead tree-structure can be devised by walking on the mesh toward the target. E.g., consider using the cell-based data structure with given vertices and neighbors. Starting form a given cell, the current position traverses to the neighbor that lies in the direction of the target, until it has reached the cell that contains the target. The direction can be determined by calculating the scalar product of the vector from the mid-side of each edge with the outward normal on that edge (fig 5.7). Only in the cell that contains the target, all scalar products are non-positive.

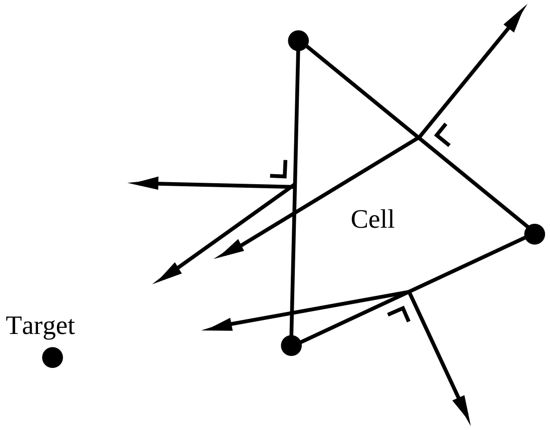


Figure 5.7: The scalar product search uses the scalar product of the outward edge normal with the vector of the mid-side of the edge toward the target to determine in which neighbor to proceed with the search.

Naturally, the average complexity of this method is $O(\sqrt{N})$ with N vertices, which is not good enough for searching large meshes. However, it is a very simple and efficient algorithm if the search starts at a cell that is close to the target and/or if the size of the mesh is small, as is typically the case for background meshes. A suitable application e.g. is the check for the existence of the required boundary edges in the Delaunay triangulation. As the check proceeds from one existing edge to the consecutive edge, an initial guess is always in the vicinity and the search is $O(1)$.

5.3.4 Searching the Voronoi Diagram for the Closest Vertex

A second search problem is finding the closest grid vertex to a given location. This is done for the distance check of new vertices to be inserted. The dual of the DT, the Voronoi diagram (VD) is a natural and most convenient tool to solve the closest-point-problem of the distance check [105]. Every Voronoi vertex is equidistant to the three grid vertices that form the triangle associated with the Voronoi vertex and the three tiles of the Dirichlet tessellation, the Voronoi regions of the three vertices intersect with each other at the Voronoi vertex. The grid vertex that is closest to the target is the one that contains it in its Voronoi region. Similar to the walk procedure on the Delaunay triangulation, a walk

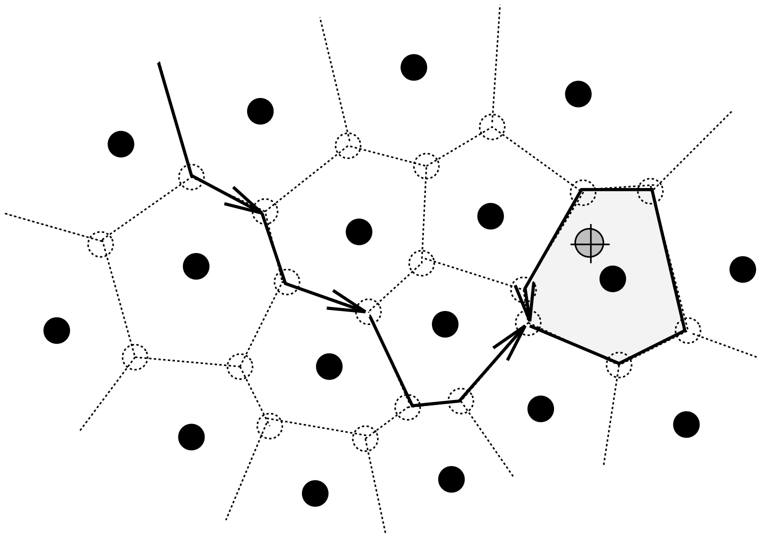


Figure 5.8: Searching for the closest point can be done efficiently on the Voronoi diagram. The bullets mark the grid vertices, the Voronoi vertices are marked with dashed circles, the Voronoi diagram is drawn with dashed lines. The Voronoi region (shaded) that contains the target (marked with a cross) is associated to the closest vertex and can be found by walking on the Voronoi region.

can be done from one Voronoi vertex to the next one leading to the Voronoi region sought. The search starts with an arbitrary grid-vertex and loops over the Voronoi vertices forming the Voronoi region around that grid-vertex. The switch from one grid-vertex to the next occurs when the target switches to the other side of the vector to the next Voronoi vertex, the perimeter vector. In figure 5.8 the loop is executed counterclockwise and the target is kept to the left. When the target switches to the right of the perimeter vector, the grid vertex is switched to the one across the perimeter vector. The procedure is complete when the loop can be executed completely while keeping the target to the left of the perimeter vector.

5.3.5 The Distance Requirement for New Vertices

Checking a new vertex against the other vertices can be done efficiently by using the underlying Voronoi diagram (cf. sec. 5.3.4), provided that the other vertices are part of the current DT. Thus, the new vertices of the current row are introduced into the structure, even though they might have to be removed again during a subsequent collision with another vertex of that row.

As a simplification of the distance check, the new vertex can be considered as properly spaced if a disc with the radius of the required spacing is completely contained in the circumcircle of the containing triangle,

$$r_{circ} > ah + |\mathbf{x}_{new} - \mathbf{x}_{circ}|, \tag{5.4}$$

with the circumradius r_{circ}, the distance tolerance a, the local spacing h and the locations of the new vertex $\mathbf{x}_{new}$ and the circumcenter $\mathbf{x}_{circ}$. Most of the new vertices satisfy the spacing constraint via this condition. It in effect neglects the changes in spacing as h is evaluated at $\mathbf{x}_{new}$.

If the new vertex fails this test, the search on the Voronoi diagram for the closest vertex is executed. If the distance to the nearest vertex is sufficient, the new vertex is introduced into the structure. If the new vertex fails the test conflicting with a vertex from a previous row, the new vertex is discarded. If it fails the test with a vertex on the same row, the other vertex is extracted from the triangulation (cf. sec. 5.3.6), the vertices are averaged and the averaged vertex is introduced. Since the extracted vertex was well spaced and the new vertex was closest to the extracted one, the new vertex inbetween the two locations is, as a simplification, can be assumed to be well spaced.

5.3.6 Extraction of a Vertex from the Delaunay Triangulation

Once the algorithm finds a pair of new vertices on the same row that are too close to each other, as it will be the case when the front is coarsened as it propagates away from the body, the vertex that has already been inserted into the DT has to be extracted and the merged vertex introduced instead. The extraction works as the inverse analogue of Watson's algorithm [48].

1. find the cavity of cells that are formed with the vertex to be extracted,

2. eliminate the triangles of the cavity.

3. retriangulate the cavity in Delaunay.

The first two steps are straightforward, using e.g. the pointers to neighboring cells to find the cavity. The tricky part is how to find the DT for the cavity, and how to do this in such a way that the pointers to the neighbors are generated efficiently.

One possible way to accomplish this is to establish a list of the vertices around the cavity with the neighboring cells that are formed with each pair of the vertices. Looking at a simple 2D cavity (figs. 5.6 or 5.4), it is clear that there has to be at least one cell of the cavity in the DT that is formed with three consecutive vertices on the border of the cavity. This cell can be found by walking along the border and testing each possible candidate for a triangle whether it is convex, i.e. inside the cavity and whether there are no other vertices of the cavity in the circumcircle around it.

If the candidate is accepted, the triangle is formed, the pointers in its neighbors updated and the cavity resized. By induction, the process can continue until one single edge is left. This edge, with the accompanying pointers to the neighbors across the edge, is used to update the last two neighboring pointers.

5.3.7 Generation of the Boundary Point Distribution

The distribution of the boundary vertices is the single most important parameter of the grid to be generated. Naturally, the boundary point density defines how fine the grid will be, but also determines the smoothness of the grid near the boundaries where the grid quality is most important. The effort spent on the distribution of the boundary vertices must remain small, however.

The approach taken here is to define spline curves, however coarse or fine they may be defined, and redistribute the vertices along each boundary segment according to interpolated values of boundary point spacing. The user input for this procedure are the boundary spline points that are usually taken from an archive of curves, are specified analytically or are taken from another grid and a few spacing interpolation stations (fig. 5.9). Vertices are placed on the spline curve using a blended quadratic spline operation due to

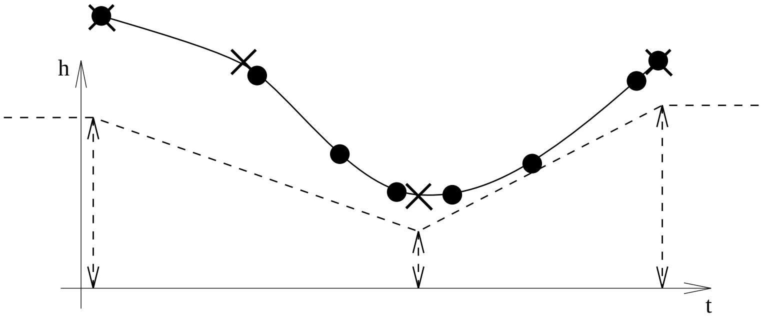


Figure 5.9: Spline curve with four support points (crosses) and boundary point spacing distribution specified by three interpolation stations (double-ended arrows). The vertices (circles) are being put down starting from the beginning of the curve using the local spacing at the last point. The jump in spacing at the end is smoothened by backward Gauß-Seidel relaxation.

Beier [106] that is based on the two spline points to the left and to the right of the interval in question. The spacing is interpolated linearly between the user-specified stations at the position of the last vertex in the curve. This leads to a jump in spacing at the end. A backward-Gauß-Seidel relaxation is then applied to evenly distribute the spacing over the curve, all the while respecting the user-specified spacing distribution. The curve-length t of each boundary vertex is updated such as to place the vertex between two neighbors according to the ratio of spacings $h_{i\pm\frac{1}{2}}$ interpolated at the centers of the two edges that they form,

$$t_i \longleftarrow t_{i-1} + \frac{h_{i-\frac{1}{2}}}{h_{i-\frac{1}{2}} + h_{i+\frac{1}{2}}}(t_{i+1} - t_{i-1}),$$

$$t_i \longleftarrow t_{i+1} - \frac{h_{i+\frac{1}{2}}}{h_{i-\frac{1}{2}} + h_{i+\frac{1}{2}}}(t_{i+1} - t_{i-1}),$$

or in a symmetric expression

$$t_i \longleftarrow \frac{1}{2}(t_{i+1} + t_{i-1}) + \frac{1}{2}\frac{h_{i-\frac{1}{2}} - h_{i+\frac{1}{2}}}{h_{i-\frac{1}{2}} + h_{i+\frac{1}{2}}}(t_{i+1} - t_{i-1}) \qquad (5.5)$$

The free choice of the number and value of interpolation stations makes the procedure highly flexible. Adding a global scaling parameter then allows to vary the total number of grid points at the touch of a scalar.

Note that the resulting boundary-vertex distribution is nor unique, nor symmetric for a spacing distribution that is not constant. The local pointwise evaluation of the spacing makes the relaxation process non-linear. As a simple example, consider a narrow, but extreme dip in the spacing prescription. The minimum will only found and be taken into consideration if the middle of an edge happens to fall on it. Two different curves will result from the same non-constant spacing distribution if the initial placement of the boundary vertices is started from either end.

5.3.8 Generation of the Background Mesh

A constrained Delaunay triangulation of all boundary vertices is computed as an initial triangulation to begin the vertex generation process (figure 5.5). This triangulation provides at no extra cost a background mesh to interpolate a local value of desired distance between vertices at any point. The spacing h is interpolated between the three nodal values of the background triangle that contains the interpolation point. The spacing h at the vertices of the background grid is computed as the average distance to its two neighboring vertices on the boundary.

On grids with a strong spacing gradation, like e.g. grids used for airfoil calculations, a monotonic variation is found between the fine spacing on interior boundaries and the coarse spacing on a far-field boundary if the background triangle connects directly from the interior to the far field boundary. But along concave contours it may happen that the Delaunay criterion connects between finely spaced interior boundaries and the background grid will specify an area with a very low spacing gradient that would produce a too fine grid. Consistent with the philosophy of minimal user input, the algorithm introduces the necessary vertices to break the unwanted connections. Unwanted connections can be

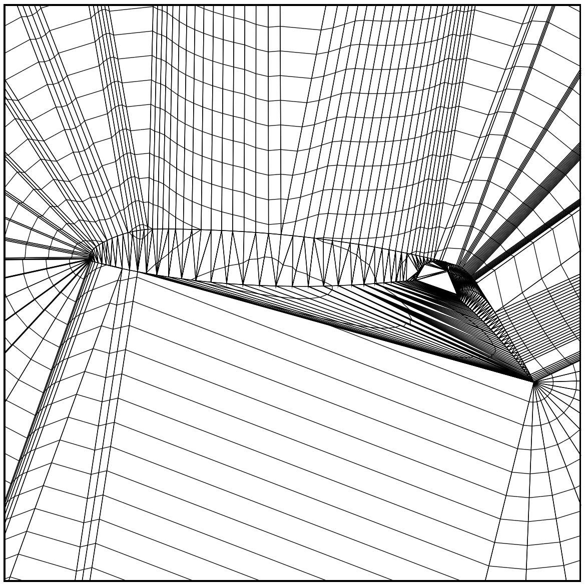


Figure 5.10: The initial triangulation of the boundary vertices with iso-spacing contours superimposed. The spacing exhibits an unwanted spike of fine spacing from the finely discretized trailing edge of the main flap to the underside of the main airfoil.

e.g. connections between components that the user has specified not to be connected or connections between boundary vertices that are non consecutive.

The procedure will be to detect an illicit liaison and place a vertex at the circumcenter of the badly connecting triangle. During a subsequent retriangulation most if not all of the triangles exhibiting an unwanted connection will be broken, thus very few extra vertices suffice. Due to its placement, the new vertex is equidistant from all ill-connected boundaries. The spacing is extrapolated from the most finely discretized boundary using some average spacing gradient of the initial triangulation,

$$\bar{\nabla} h = \frac{\sum_{i=1}^{N} S_T |\nabla h_i|}{\sum_{i=1}^{N} S_T}. \tag{5.6}$$

5.3.9 Placing New Vertices

Since the Delaunay methods are not “greedy”, Tanemura’s rising bubble algorithm being the exception [107], it makes little sense to define a rigid front like in the AFM that separates the triangulated from the untriangulated region. A frontal edge is much better defined here as the interface between a badly shaped cell that is yet to be refined and an acceptable or unrefinable cell.

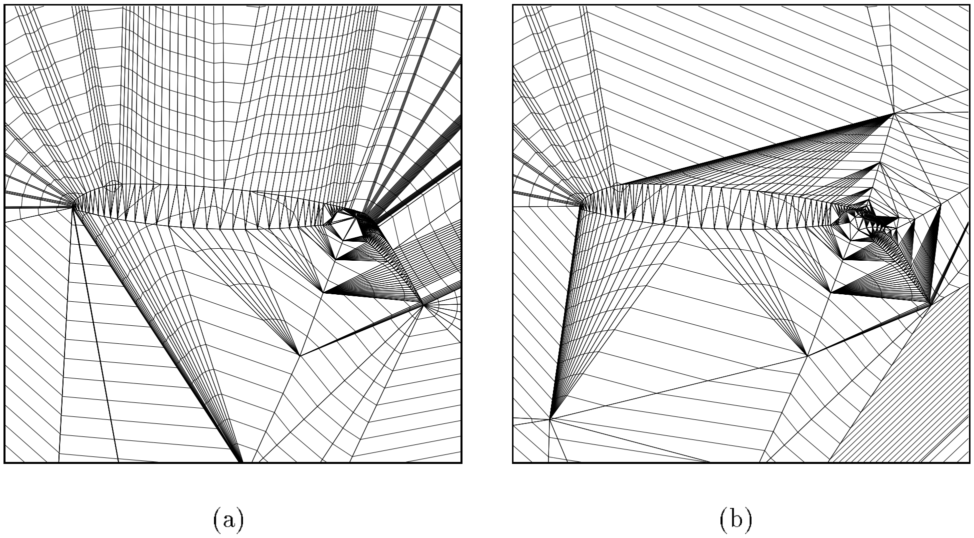


Figure 5.11: Grid (a) has been modified from figure 5.10 by disconnecting the main flap from the main airfoil. Grid (b) has been modified by breaking all connections between non-consecutive boundary vertices. Both modifications effectively and automatically remove the spike of fine spacing.

The ratio of squared side-lengths is used to quantify element quality q,

$$q_i = \frac{\min \mathbf{n}_i\mathbf{n}_i}{\max \mathbf{n}_i\mathbf{n}_i}, \qquad q = \min q_i \tag{5.7}$$

This criterion not only identifies a bad triangle that is either acute or obtuse by evaluating $q < q_{godunov}$, it also identifies the shortest side with $q_i < q_{godunov}$ that might be a frontal edge: a short edge in a cell with disparate edge lengths that is shared by a well shaped or unrefinable cell.

For each frontal edge a new vertex is constructed on the median into the badly shaped cell such that the distances between the new vertex and the two vertices forming the edge approximate h, the isotropic length scale.

The length of the sides l_1 and l_2 opposite vertices 1 and 2 is approximated by $2/\sqrt{3}l$ of the altitude l as found in an equilateral triangle. Requiring that this approximated side-length equals the desired spacing h_4 evaluated midway between vertex 3 and the midpoint of the base M (figure 5.12), one finds

$$h_4 = \frac{2}{\sqrt{3}}|\mathbf{x}_3 - \mathbf{x}_M| = \left(\frac{1}{2}\left(\mathbf{x}_3 - \mathbf{x}_M\right)\nabla h + h_M\right) \tag{5.8}$$

where h_M denotes the desired spacing at M, ∇h is the local gradient of the background spacing and $\mathbf{x}_3$,$\mathbf{x}_M$ are the position vectors of vertex 3 and point M. As the new vertex

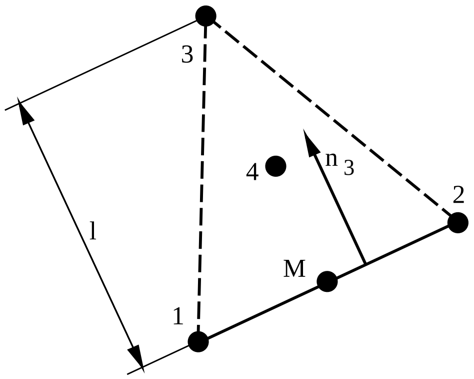


Figure 5.12: Construction of the new vertex. The new vertex 3 is place on the median of the frontal edge 1,2. The spacing is interpolated at the point 4.

is placed along the median, one can write

$$\mathbf{x}_3 - \mathbf{x}_M = l\frac{\mathbf{x}_3 - \mathbf{x}_M}{|\mathbf{x}_3 - \mathbf{x}_M|} = l\mathbf{n}_3. \tag{5.9}$$

where $\mathbf{n}_3$ is the unit normal on the frontal edge pointing towards the triangle to be refined. The altitude for a triangle with counterclockwise sense is thus

$$l = \frac{h_M}{\frac{2}{\sqrt{3}} - \frac{1}{2}\mathbf{n}_3\nabla h} \tag{5.10}$$

Note that in the given form the altitude of the explicit triangle is independent of the length of the frontal edge. This conserves the thickness of the layer of cells introduced even if the length of the faces varies strongly. A partial view of the grid with four rows is shown in figure 5.13.

5.3.10 Summary of the Frontal Delaunay Algorithm

The essential vertex generation loop can be summarized in pseudo-code as:

```
do while new vertices are found
   for each triangle in the grid
      for each face of the triangle
         if this face is frontal
            find a vertex to form a triangle with the face.
         end if
      end for
```

```
    end for
    for each new vertex
        find the closest vertex in the mesh,
        if the new vertex is properly spaced
            introduce the vertex into the structure
        else if the conflicting vertex is on a lower row
            discard the new vertex,
        else
            extract the conflicting vertex from the grid,
            merge it with the new vertex,
            reintroduce the merged vertex.
        end if
    end for
end do
```

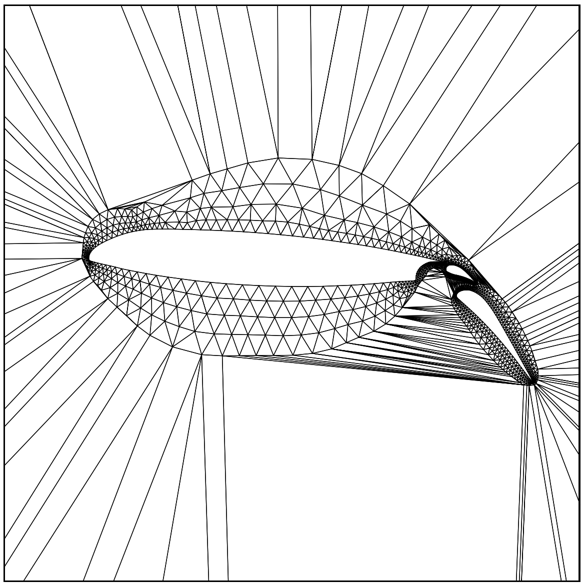


Figure 5.13: View of the grid around a three element airfoil after four rows of vertices have been inserted.The triangles outside the domain have been removed for better visibility. Note the node merging in the second row on the upperside of the airfoil that leads to a coarsening of the front. The thickness of the front is preserved in the fourth row since the node-placement formula is independent of the length of the frontal edge.

The number of operations necessary to introduce a vertex in the isotropic process is of $O(1)$ for the construction and of $O(\log N)$ for the distance check if a tree structure for the searches on the background mesh was used [108]. Hence, the cost of generating a mesh with N vertices, $O(N \log N)$, is asymptotically optimal [105].

5.4 Examples

A generic three-element airfoil configuration is shown in figures 5.14 to 5.16. The regularity of the grid is entirely due to the frontal insertion, no smoothing filter was applied. The cell surface area varies very smoothly from the smallest cells at the trailing edges and corners to the largest cells at the outer boundary with a ratio of nearly 2×10^7. The robustness and the smoothness of the algorithm can be seen in figures 5.15(b) and 5.16(a) where different fronts coalesce between components and the smoothness and regularity of the grid is barely affected.

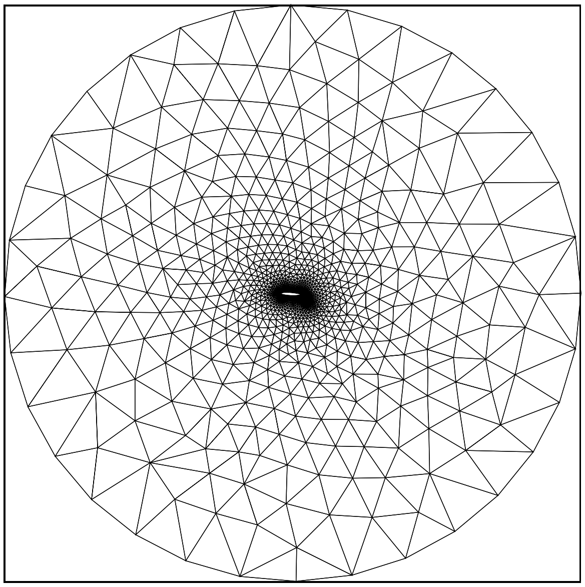


Figure 5.14: Full view of the grid around a three element airfoil.

The triangulation was started from 340 boundary vertices, 133 on the main airfoil, 75 on the vane flap, 104 on the main flap and 33 on the outer boundary. The triangulation of the boundary vertices took .6 sec. on a DEC 5000 workstation. 8 background vertices were inserted to disconnect the main airfoil and the main flap as shown in figure 5.11(a). 39 rows with 3257 interior vertices were inserted in 13 sec. Note that no tree data structure was employed, as would be necessary to obtain optimal $O(N \log N)$ complexity.

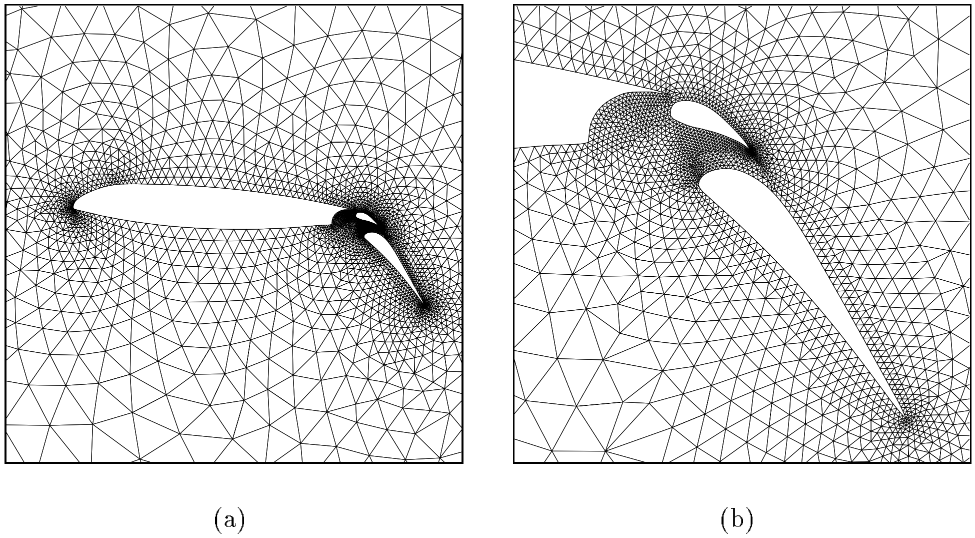


Figure 5.15: View of the grid around the wing section of a three element airfoil (a) and a closeup of the region around the flaps. In this region the fronts emanating from all three elements coalesce smoothly. No smoothing has been applied.

5.5 Laplacian Smoothing

The vertex placement strategy as summarized in section 5.3.10 is an entirely hyperbolic procedure, with the exception of the vertex-merging in the most recent row. Naturally, the Delaunay triangulation has a smoothing effect leading to the various optimal characteristics. It is desirable, however, to improve the smoothness of the point cloud by applying an elliptic procedure, such as Laplacian smoothing

$$\mathbf{x}_i \longleftarrow \mathbf{x}_i + \nu_L \frac{1}{\sum_j 1} \sum_j (\mathbf{x}_j - \mathbf{x}_i) \qquad \forall i, j \subset T \tag{5.11}$$

For the choice of the relaxation factor as $\nu_L = 1$, the vertex $\mathbf{x}_i$ is placed at the average location of all neighboring vertices $\mathbf{x}_j$. The relaxation is done using a point-Gauß-Seidel-update on the numeric order of the vertices. This accelerates the relaxation and simple checks can guarantee that the new location is still in the cavity and no overlapping triangles are created. A point-Jacobi-update cannot guarantee that since all vertices are displaced simultaneously.

As shown in chapter IV, relaxation only affects the solution locally in an efficient way. However, this is precisely a desired property in the context of mesh smoothing: the global vertex distribution must not be affected by too much smoothing.

The grids for the three element airfoil configuration in figures 5.15(b) and 5.16(a) have

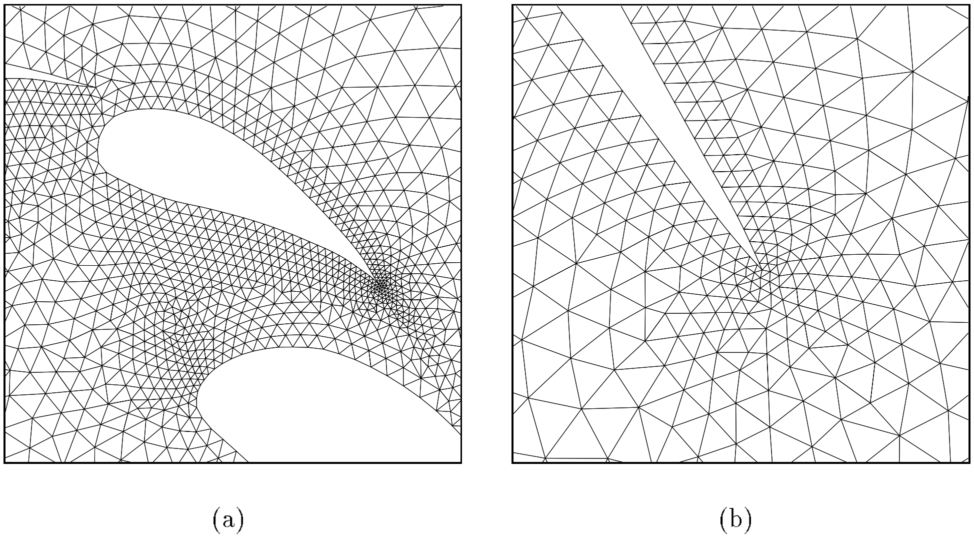


Figure 5.16: View of the grid around the wing section of a three element airfoil around the vane flap (a) and a closeup of the region around the trailing edge of the main flaps (b). The trailing edge is triangulated smoothly indicating the robustness of the algorithm, no smoothing has been applied.

been relaxed with five sweeps at $\nu_L = .5$. The improved grids are shown in figures 5.17, the statistical quality analysis in section 5.7.

5.6 Vertex-Nested Coarsened Meshes for Multigrid Schemes

FroD, like any other frontal mesh generation method, can be modified to produce successively coarser meshes where all vertices of the coarser mesh are present in all finer meshes. This vertex-nestedness is very advantageous in a Multi-Grid context since it dramatically simplifies the transfer operations. Previous Multi-Grid algorithms on unstructured grids either used non-nested sets of meshes [87] that require more costly transfer operations or resorted to agglomeration methods [109] that impose special requirements on the solvers used on the coarsened meshes due to the geometric complexity of the agglomerated elements.

Techniques to generate vertex-nested meshes from triangulations have been presented by Guillard [110] and Richter [111]. Both methods do not employ the frontal character of the mesh generation that the present method is designed for. The duplication of the frontal vertex generation scheme on twice the scale produces element-nested meshes in regions of regular triangulation.

The necessary steps to create a vertex-nested mesh on the next coarser level with FroD

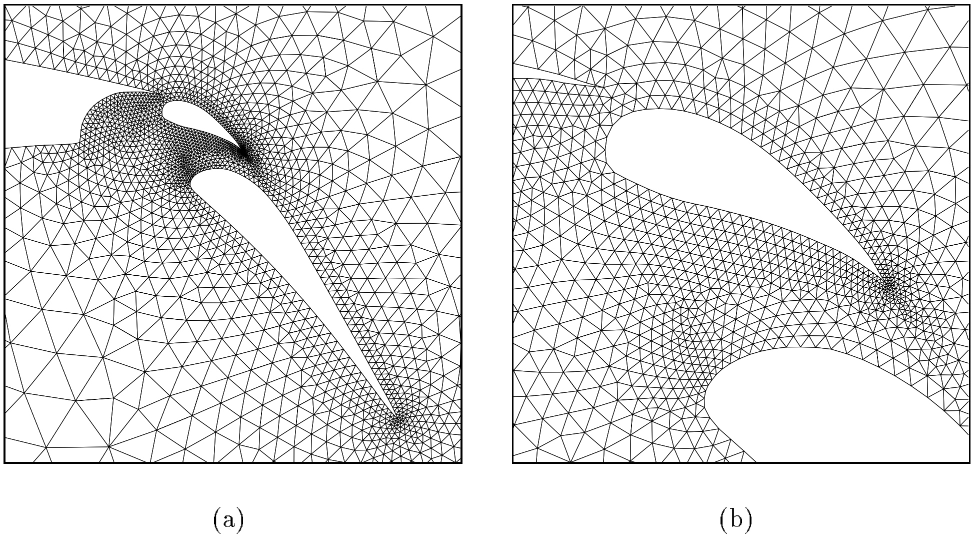


Figure 5.17: View of the grid around the wing section of a three element airfoil in the flap region (a, compares to fig. 5.15(b)) and a closeup of the vane flap (b, compares to fig. 5.16(a)). Laplacian smoothing has been applied.

are as follows:

1. Coarsen the boundaries by skipping every other vertex. If the number of boundary vertices on a boundary segment including the endpoints is even, a pair of vertices has to be skipped. In the current implementation that pair is selected that has the lowest ratio of curve length of the three edges connected to the pair compared to the average spacing of the two vertices adjacent to the pair.

2. Double the spacing values of the background grid.

3. Run a modified frontal process with the coarser set of boundary vertices: the distance check is replaced by selecting an appropriate vertex of the next finer grid and warping the new vertex on the coarse level to that position and introducing it into the coarse grid. An appropriate vertex on the fine gird is close to the proposed location of the new vertex and has no neighboring vertex on the fine level that is already member of the coarse grid. The vertices on the fine mesh are tested for their suitability in the following order:

   (a) the closest vertex,

   (b) the three vertices that form the element that contains the new vertex,

   (c) all neighboring vertices of the closest vertex.

If no such vertex can be found, the new vertex is discarded.

A sequence of vertex-nested grids can be seen in figure 5.18 for the example of the subsonic and transonic GAMM channel [112]. If the fine grid is regular, the coarse grid constructs nested elements as can be observed with the grids of level 5 and 4 in figure 5.18. Note that the method does not guarantee that all possible fine grid vertices are found by the coarse grid frontal process. On very coarse configurations with high gradients it can occur that a possible fine-grid vertex to warp upon is out of the range of search and the coarse grid vertex is rejected. This however seems not to be a problem in practical cases.

5.7 Estimation of Angular Bounds for the Frontal Delaunay Algorithm

As the generation and introduction of vertices in FroD is embedded in the rather rigorous mathematical framework of the Delaunay triangulation, the algorithm can be analyzed mathematically. Ruppert [52] gave an algorithm that guarantees a minimum angle, but produces grids with too few triangles in the field to be suitable for flow calculations. Chew [51] presented an algorithm that guarantees maximum angles of 120° but that does not allow to specify a spacing gradation. Both methods produce meshes that are much more irregular than the ones produced by Rebay's method [102], the AFM [42] or the present method. On the other hand, no derivation of angular bounds is known for the more heuristic AFM. The smoothness of the grids created with the AFM relies on aposteriori smoothing. For a vanishing spacing gradient, Rebay's method [102] reverts to Chew's [51] and the same bounds apply.

In the following, angular bounds for FroD are derived, assuming that the triangulation is Delaunay and that the gradient is constant. Thus, the analysis can fail in the vicinity of boundaries and for configurations that cross edges of the background grid where the spacing gradient exhibits a jump.

5.7.1 Upper Angular Bound

FroD will detect short edges in triangles that are considered too obtuse or too acute and will construct a vertex on the median of that frontal edge in order to refine the triangulation locally and to improve the grid. The new vertex is placed approximately at a distance h from the two vertices that form the frontal edge (cf. sec. 5.3.7). The introduction of the vertex is subject to a distance check: if the new vertex is located too close to any existing vertex, it cannot be introduced and the refinement will not take place. The vertex is too close if the distance to the closest vertex is lower than a tolerance a times the local mesh spacing h that is interpolated at the midpoint of the two vertices in question.

In this way, as a worst case a triangle with non-desirable properties has to be considered that cannot be refined because the local vertex density is already too high. Figure 5.19 shows the limit case: the obtuse triangle ABC with a circumcircle O of radius r. As this triangle is DT, there is no other vertex of the triangulation in O and we can always introduce the new vertex D if its spacing disc S with a radius ah around D is contained in O. Consequently, the worst case occurs when the two short edges of the triangle ABC have the minimum permissible length of ah. With a larger edge length r will increase and the disc around D is included in O already at a smaller angle γ. Hence, given the tolerance a, the maximum angle γ can be calculated for $\nabla h = 0$:

$$\gamma = 2 \arctan \left(s - \frac{1}{4s} \right), \quad s = \frac{\sqrt{3}}{2a} + 1, \tag{5.12}$$

The narrower limit is obtained for $a = 1$ with $\gamma = 120°$. However, this restrictive tolerance will inhibit the insertion of D as D then is too close to C and B. The maximum angle increases monotonically to 139° for $a = .5$ (fig. 5.21).

In the case of $\nabla h \neq 0$, the construction is executed numerically. All orientations of ∇h are tested since the length of the edge CA, the placement of D and the shape of S depend on the gradient. The deformed spacing disc S is traced out and tested for inclusion in O. Three curves for various $\nabla h / h$ are shown in figure 5.21. The curves are drawn for all values a for which S does not contain B or C for any gradient.

5.7.2 Lower Angular Bound

Similarly, a lower angular bound can be derived when refining triangles with one short edge BC opposite the acute angle β by inserting D, as depicted in figure 5.20. As long as the circumcircle O is large enough to contain the spacing disc S around D, refinement of the triangle ABC will always be possible. Again, as β is further increased, O will shrink until the limiting case is reached where O and S become tangent. In this case one finds for $\nabla h = 0$:

$$\beta = 2 \arctan \frac{1}{2s}. \tag{5.13}$$

The smallest minimum angle is found for $a = 1$ as $\beta = 30°$ and decreases monotonically to $\beta = 21°$ for $a = .5$. The construction for $\nabla h \neq 0$ is executed numerically as in the obtuse case. The curves are given in figure 5.21.

5.7.3 Measured Angular Bounds

Maximum and minimum angles for two configurations have been measured for the three-element airfoil and the finest resolution of the Gamm channel for various distance tolerances $.5 \leq a \leq .9$. The number of vertices varies for the three-element airfoil between 2113 and 2927 and for the GAMM channel between 2566 and 3492.

The background grid of the three-element airfoil (fig. 5.15) is very rough with a ratio of the maximum spacing gradient to the minimum spacing of $\nabla h_{max}/h_{min} = 4.9/.034 = 144.$. With this ratio, no useful distance tolerance $a > .5$ is found that keeps the spacing disc inside the circumcircle for all gradient directions. Even when assuming that most of the cells have a gradient ratio of less than twice the average gradient, $2\nabla h_{avg}/h_{min} = .038/.034 = 1.1$, the minimum angles observed fall below the expected minimum (fig. 5.21).

The background grid of the GAMM channel (fig. 5.18, the finer 3400 vertex grid was measured) is much smoother, although the maximum gradient found in only one cell, is still prohibitively large for a bound. However, virtually all of the cells in the background grid are close to the average gradient ratio $\nabla h_{avg}/h_{min} = .058/.086 = .67$ and the observed minima and maxima for all a fall within the predicted bounds (fig. 5.21).

The validity of the proofs could be extended by smoothing the background grids: very easily a procedure could be devised that detects edges with a strong variation of the gradient and introduces vertices at the neighboring circumcircles to improve the quality of the background grid. However, as will be seen in the next section, the actual grid generation process seems very little concerned about these worst cases.

5.7.4 Distribution of angles, surface area ratios and vertex degrees

The analytical bounds for constant gradients suggest using a tolerance a as close to 1 as possible. The results in figure 5.21 show the best values for a tolerance around $a = .8$.

However, distributions of the values for each cell over the entire grid can show the grid quality better than the maximum or minimum values. As the figures 5.22 to 5.24 show, a narrow peak close to the optimum values is found for a rather liberal tolerance $a = .65$ with a very small number of cells with excessive angles $\gamma \geq 90°$ or a surface area ratio larger than 2. The surface area ratio plotted for each cell in figure 5.24 is defined as the larger one of the ratio of the areas of each triangle compared to its neighbor's and the inverse of it. A more stringent tolerance of $a = .8$ produces a broader variation with a larger number of undesirable cells. As opposed to what the actual maximum values from figure 5.21 suggest, any choice of $.5 \leq a \leq .8$ will produce a satisfactory grid. The distribution of the vertex degrees in fig. 5.22 shows an extremely narrow peak at the optimum value of 6 edges formed with each vertex. Very few vertices exhibit a lower or higher vertex degree.

These narrow distributions are improved by applying a few sweeps of a Laplacian filter. This filter also effectively removes the small peak in maximum angles around 90° that appears when two vertices have been merged.

5.8 Summary

A grid generation method has been invented that combines the regularity of frontal vertex-generation schemes with the optimal connectivity of the Delaunay triangulation. The method is robust and efficient and generates grids with minimum user input, such as a boundary point distribution only.

A background grid for the interpolation of the grid spacing is derived automatically from the spacing values implicitly defined with the boundary point distribution. Nodes are placed in rows around frontal boundary segments and are averaged in each row to achieve a smooth distribution when the mesh spacing increases as the front propagates.

The method will not fail but always produce a consistent grid, provided that the prescription of the boundary points is consistent.

Upper and lower bounds for the angle in the triangulation can be shown under certain simplifications and the statistical analysis of the grids shows a high quality of the grids with very few elements approaching those bounds.

A new method has been designed to generate vertex-nested multi-level grids for Multigrid calculations. The generation algorithm exploits the mechanisms of frontal point placement efficiently and can be adapted very easily to work with any frontal node generation scheme.

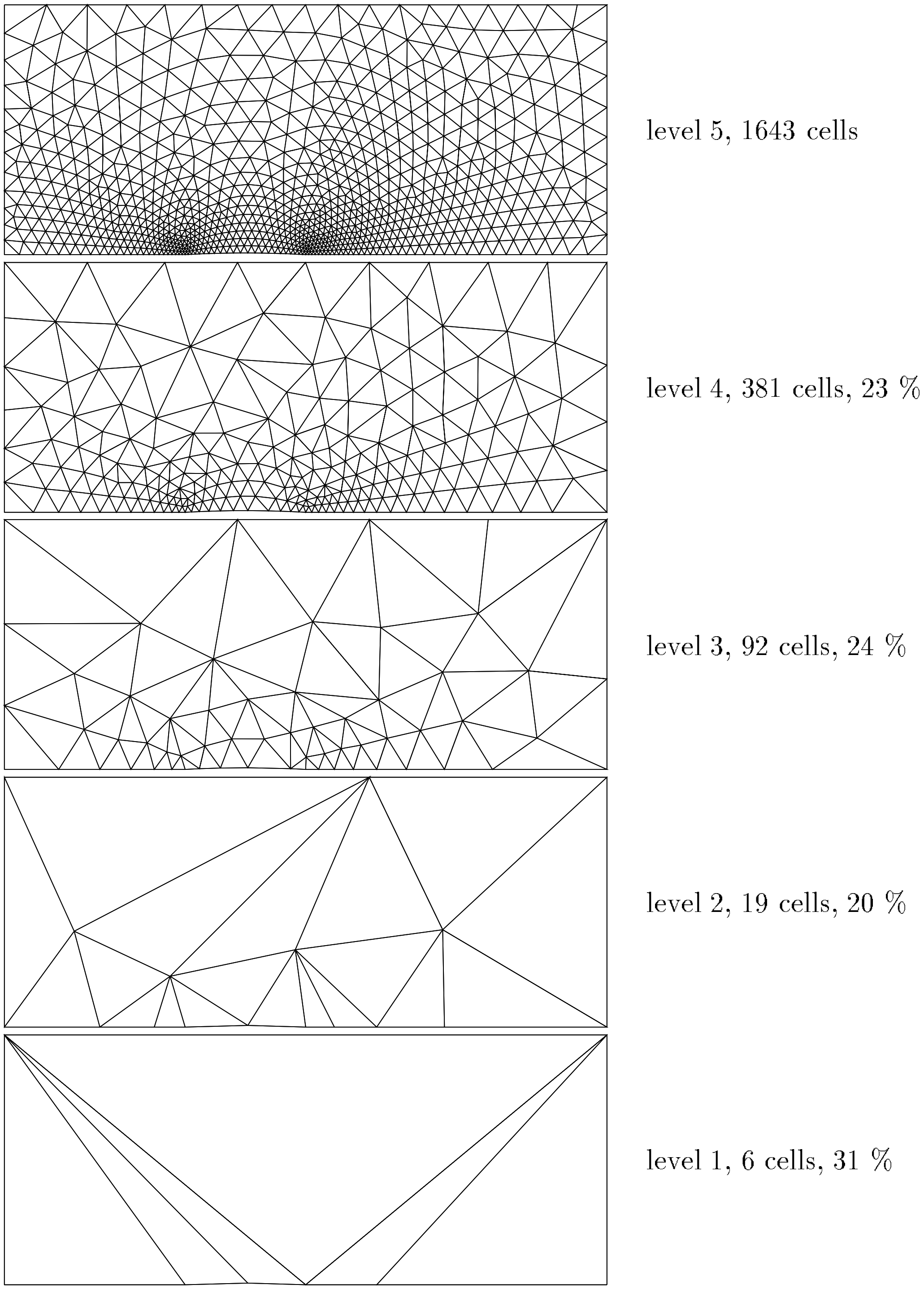


Figure 5.18: Multigrid levels 1-5 for the GAMM channel, 1643 vertices on the finest level. The percentage figure on the right indicates the relative coarsening in cell numbers from the next finer level.

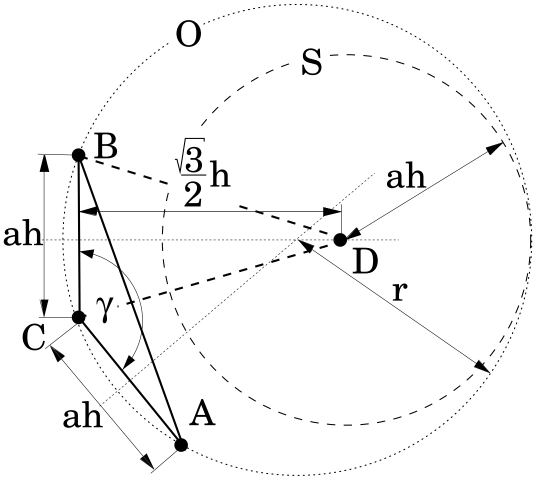


Figure 5.19: Obtuse triangle ABC with maximum angle γ. The circumcircle O contains the spacing disc S around the new vertex D.

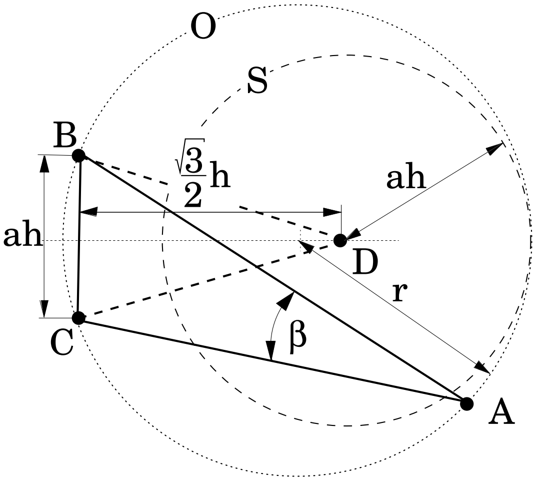


Figure 5.20: Acute triangle ABC with minimum angle β. The circumcircle O contains the spacing disc S around the new vertex D.

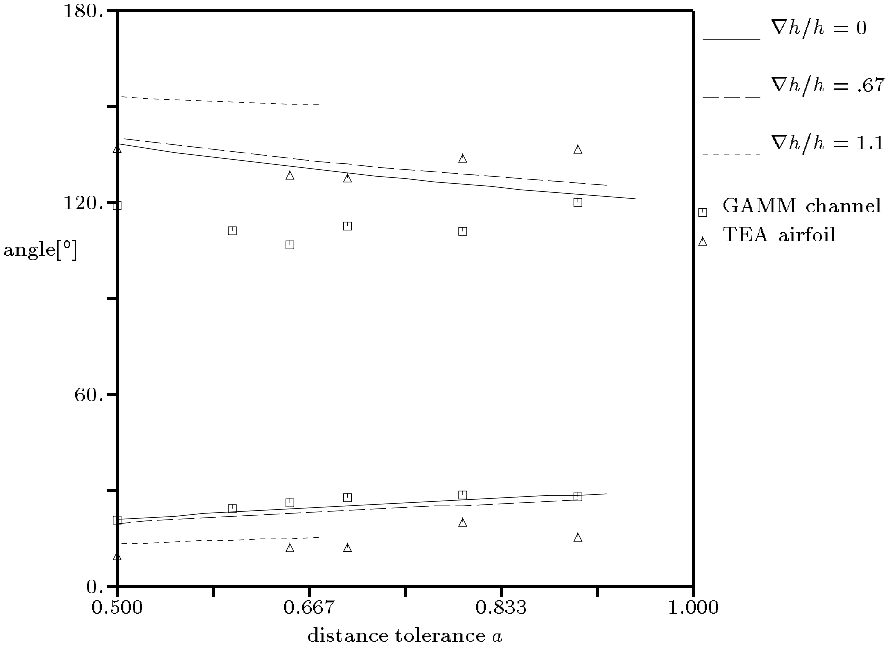


Figure 5.21: Upper and lower angular bounds for different $\nabla h/h$ and measured maximum and minimum angles of two configurations in function of the tolerance a.

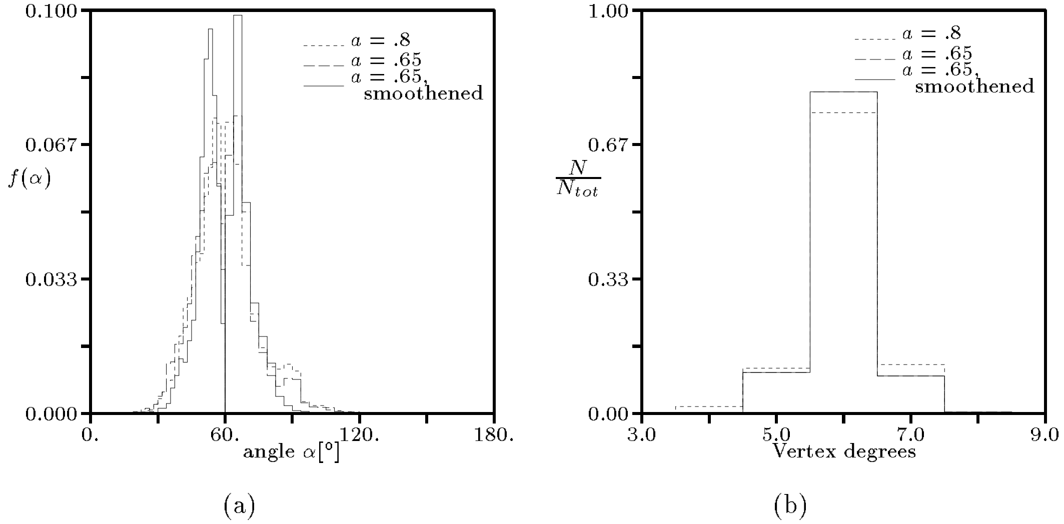


Figure 5.22: Distribution of minimum and maximum angles (a) and vertex degrees (b) for the three airfoil configuration. Shown are distribution functions for the distance tolerance a = .8, for a = .65 and for a = .65 with Laplacian smoothing.

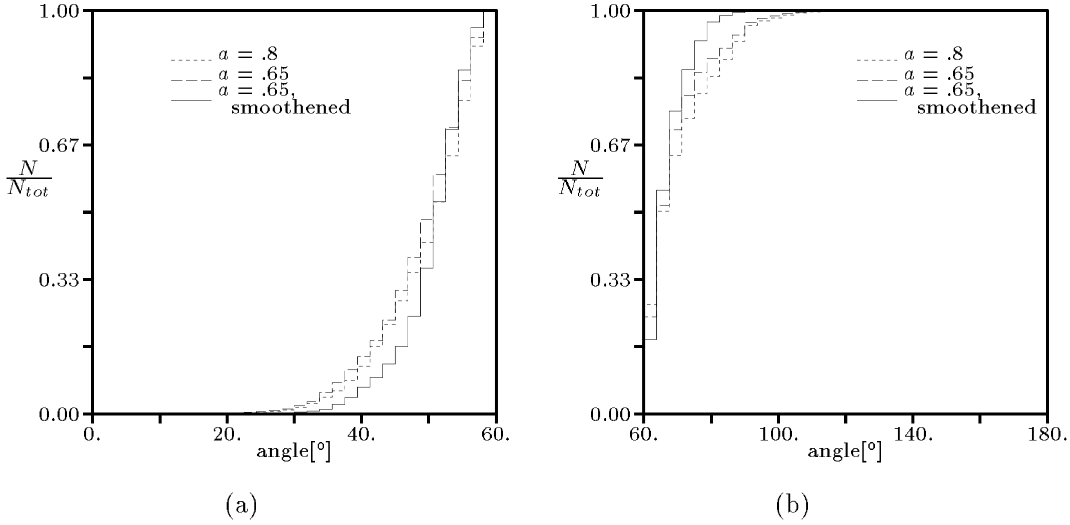


Figure 5.23: Probability densities for minimum (a) and maximum angles (b) for various a and with smoothing.

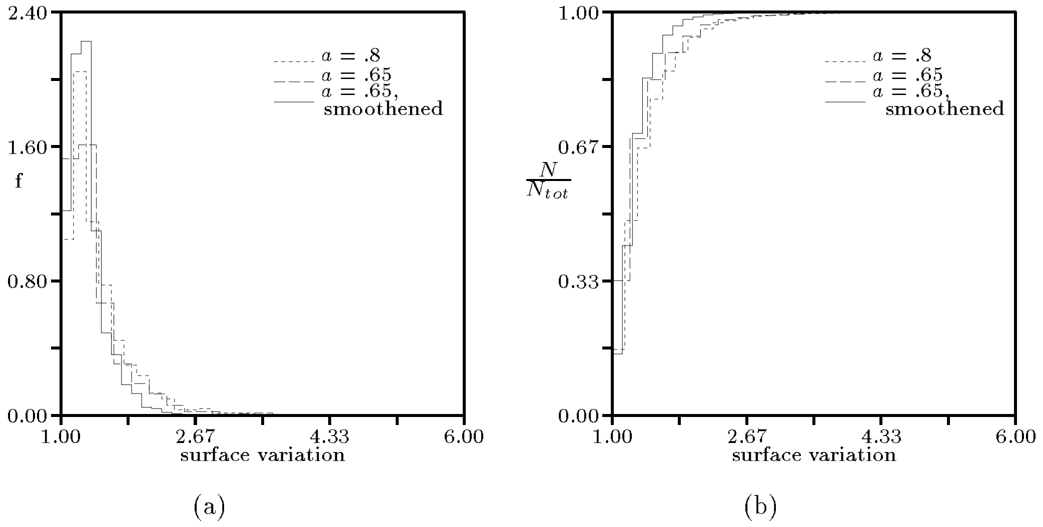


Figure 5.24: Distribution function (a) and probability density (b) of the surface variation between neighboring cells.

Chapter VI

Convergence Studies

> "Qu'est-ce que c'est que cette sorcellerie? Comment échappez-vous à mon pouvoir? Ne pensez pas que cela va continuer. Je trouverai bien la cause de ce désordre. Il y doit avoir quelque chose de rouillé dans le méchanisme et les enchâinements subtils."
>
> Eugène Ionesco, 'Le roi se meurt'

6.1 One-Dimensional Problems

6.1.1 Linear Advection

The simplest testcase to consider as a one-dimensional hyperbolic problem is linear advection of an input profile with a constant advection speed $\mathbf{a}$. Here a jump in the input profile at the origin is chosen that is advected oblique to the grid lines. The input values on the left inflow face $x = 0$ including the origin are set to $u = 1$, the lower inflow values $y = 0$ to $u = 0$ (fig. 6.2). A sample Multigrid solution with four grid levels of regular triangulation is given in figure 6.2 using the PSI scheme.

The inflow values are imposed at the inflow boundaries $x, y = 0$, no boundary condition is specified at the outflow.

Time-stepping methods used are

- Forward-Euler time-stepping with CFL = 1. A work-unit for this time-stepping scheme is equivalent to one relaxation on the finest grid.

- Multistage time-stepping with Catalano's coefficients optimized for high-frequency damping [81]. For purposes of demonstration a three stage scheme was chosen that has coefficients optimized for a second order Upwind scheme that do not depend

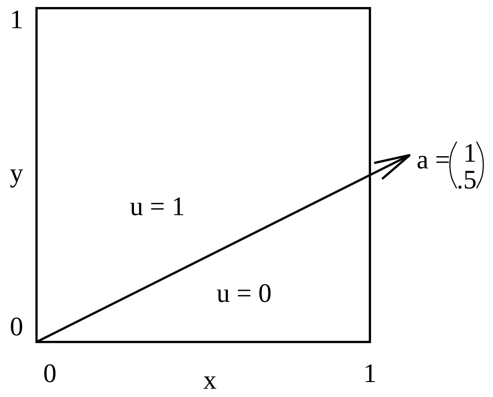


Figure 6.1: Linear advection of a shear oblique to the grid lines.

on the flow direction. The CFL numbers are $\nu_1 = .28\nu$, $\nu_2 = .61\nu$, $\nu_3 = \nu = 1.1$. Catalano gives the damping rate as $\sigma = .42$. A work-unit for this time-stepping scheme is equivalent to one relaxation stage on the finest grid.

- Gauß-Seidel time-stepping. The nodes are ordered downstream according to their distance to a tilted line passing through the origin with direction $\mathbf{s} = (-.01, 1)^T$ for the linear advection case, and $\mathbf{s} = (1, -.01)^T$ for Burgers' equation. All the contributions of the cells formed by a specific node are calculated and the node is updated. The sweeps are done symmetrically with a downstream pass followed by an upstream pass. The work-unit for this time-stepping scheme is equivalent to one symmetric sweep on the finest grid. Considering that the cost of relaxation is mainly associated with the cells, one Gauß-Seidel work-unit is six times as expensive as one Forward-Euler time-step in terms of CPU-time, since there are three nodes per cell and two passes are executed.

The residual is taken as the L_2-norm of the scalar quantity that is being advected, without normalization of any kind.

$$\log_{10}(\mathrm{Res(u)}) = \frac{1}{2}\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N}(\mathrm{Res(u)}_i)^2\right) \tag{6.1}$$

The N-Scheme

Various time-stepping schemes are applied to the shear problem, spatially discretized with the N-scheme in figure 6.3 on the 17x17 grid shown in figure 6.2. Using Gauß-Seidel time-stepping a direct solver is obtained in the single grid case. The 3-node stencil of the N-scheme extends upstream only (cf. section 3.2.2). Thus, by updating in a streamwise ordering, the domain of dependence of each node has been updated once it is reached.

Naturally, adding multi-grid time-stepping to that does not affect the convergence since after the first sweep the residuals on the finest grid have vanished. The associated cost increases with the work done on the coarser levels. The optimal cycle parameters are $\nu_a = 1$ pre and $\nu_p = 0$ post-iterations.

Forward-Euler time-stepping on a single grid exhibits a classic hyperbolic convergence pattern: the front that separates the updated from the initial solution marches through the domain and exits without reflection from the outflow boundaries, resulting in a sudden drop in residual. Applying multigriding to this time-stepping scheme results in a better initial convergence due to the faster wave propagation on the coarser grids. The optimal cycle parameters for the N-scheme and Forward-Euler are $\nu_a = 3$ pre and $\nu_p = 1$ post-iterations. However, convergence to full accuracy is slowed due to the central transfer operators, that in the prolongation step spill errors back upstream into the already converged field (fig. 6.3).

The comparison of convergence histories on various grid sizes in figure 6.4 shows a marked dependence on the grid size, none too surprising after the previous discussion: advection of the solution front through the domain takes longer. However, the difference between the multi grid and the single grid convergence reduces for the larger grids as the speeding-up of waves on coarser grids becomes more pronounced on the larger domains. Since the problem at hand is linear, the characteristics on the coarse grid are the same as the characteristics on the fine grid, allowing the fine grid solution to fully benefit from the fast convergence of the coarse grid solution. This is not the case for general non-linear cases as will be seen in the following section with the nonlinear Burgers' equation.

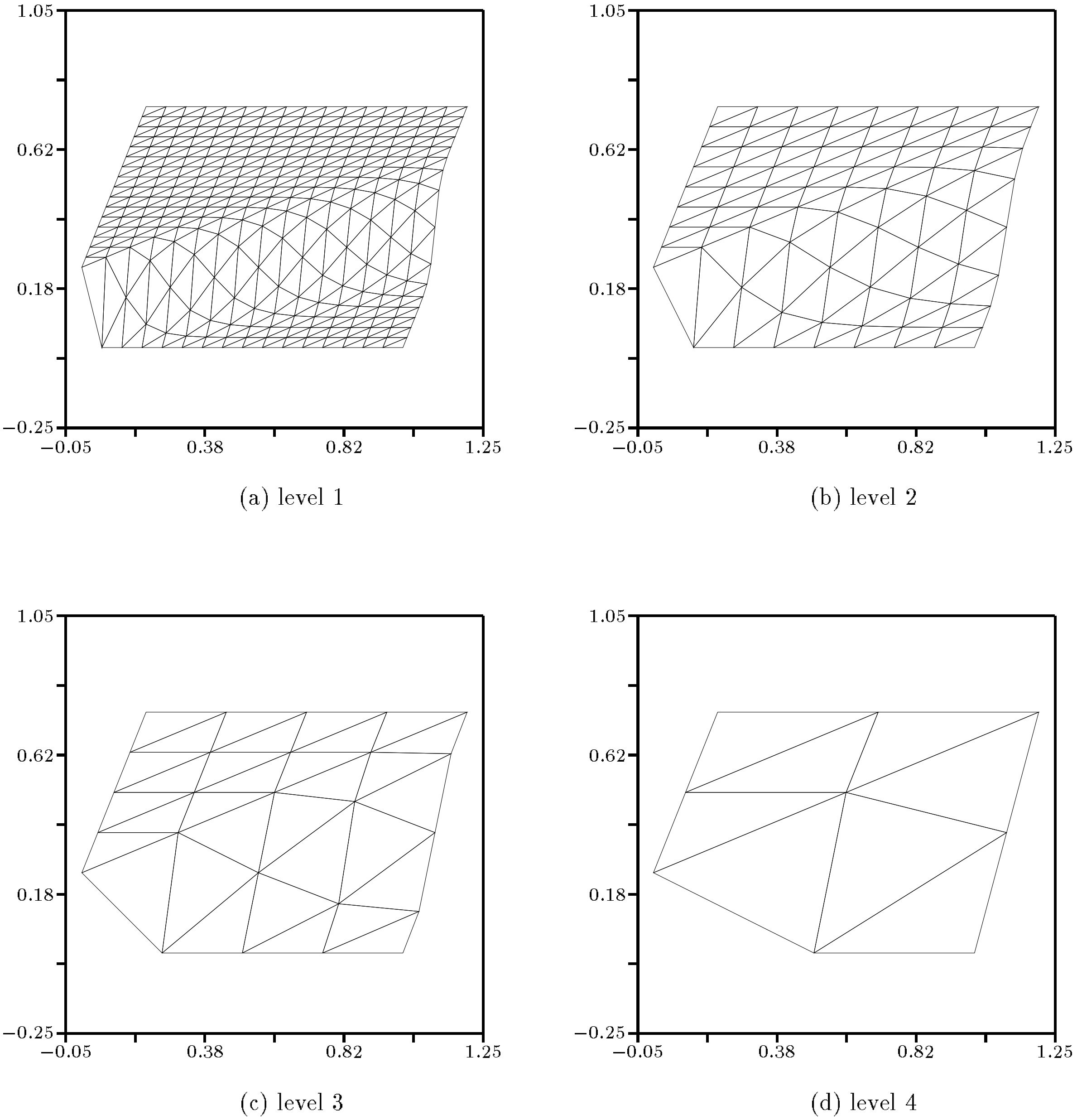
A last comparison for the N-scheme in figure 6.5 compares single stage Forward-Euler time-stepping with multistage time-stepping. Catalano presented multistage coefficients in [81] for various Upwind schemes that attempt to optimally dampen the high-frequencies by proper selection of the stage coefficients. The set of coefficients as given in section 6.1.1 has been derived in [81] for the dimensionally split scalar third order Upwind scheme, $\kappa = \frac{1}{3}$ and used by Paillère in [34]. The improved damping rate however is obtained at the cost of reduced advection: the CFL-numbers are lower than the stability limit CFL=1 employed for Forward-Euler time-stepping. This is clearly demonstrated in figure 6.5. While the four-level multi grid three-stage scheme using Catalano's coefficients (C3lv4) now outperforms the single grid version using the same multistage scheme (C3lv1), it still falls short of the performance of the single-grid Forward-Euler scheme (FElv1). Moreover, the three-stage Multigrid scheme (C3lv4,6) is outperformed even by the Multigrid scheme with Forward-Euler time-stepping (FElv4,6). The benefits of the optimal damping aren't even achieved on the larger 65x65 grid. Advection remains the dominant factor in the convergence process for scalar linear advection.

The PSI-Scheme

For the PSI-scheme, Gauß-Seidel time-stepping no longer results in a direct solver (fig. 6.6). The PSI-scheme uses a five-node stencil, not all of which are updated prior to updating the central node. This is the price that has to be paid for second order accuracy, the unique second order scheme on a regular grid having a stencil of four nodes [60]. Monotonicity has to be bought at the price of adding an additional node to the stencil. Applying Multigrid in conjunction with Gauß-Seidel produces initial benefits, but is detrimental in the latter stages of convergence. Forward-Euler time-stepping doesn't benefit either from multgriding. The optimal cycle parameters are $\nu_a = 10$ pre and $\nu_p = 1$ post-iterations. Although the advection is linear, the scheme applied is non-linear and the benefits realized in the case of the N-scheme cannot be duplicated. Similarly, the improvement of Multigrid performance on larger grids as observed with the N-scheme (fig. 6.4) is not found with the PSI-scheme (fig. 6.7): the single grid scheme always converges faster. The fewer levels used, the better the performance even on the 65x65 grid (fig. 6.8).

The LDA-Scheme

Very similar results to the PSI-scheme are found when using the LDA-scheme with Multigrid (fig. 6.9). In terms of work-units, slight benefits in convergence are realized with Multigrid in the early stages of convergence, but due to the superior advection and the proper domain of dependence, the single-grid in the end converges faster.



(a) level 1

(b) level 2

(c) level 3

(d) level 4

Figure 6.2: Linear advection of a shear oblique to the grid, $\mathbf{a} = (1., .25)^T$, using the PSI-scheme. Shown are the four levels with 17x17 (a), 9x9 (b), 5x5 (c) and 3x3 nodes (d) respectively. Note that the coarse grid solutions telescope on the fine grid solutions due to the right-hand-side forcing function.

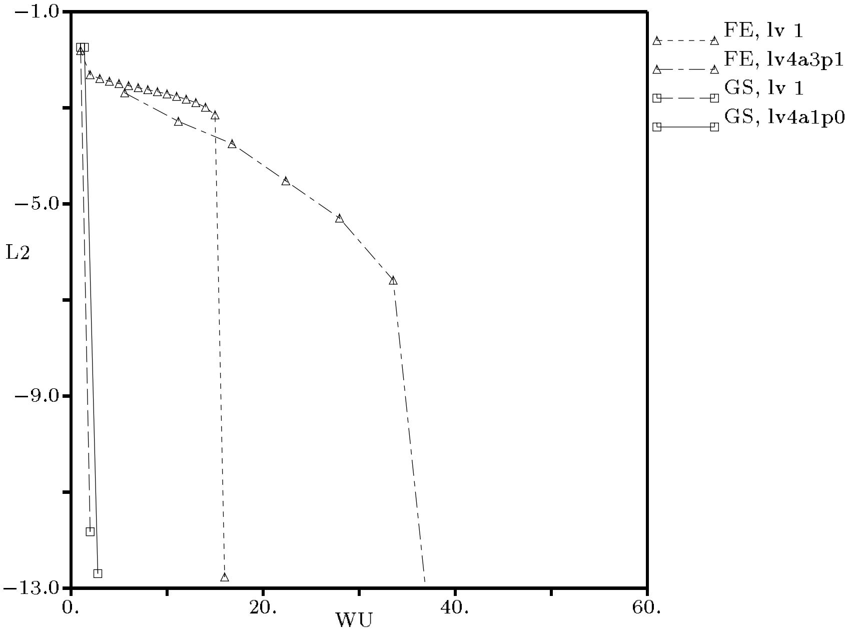


Figure 6.3: Convergence histories of the N-scheme for linear advection oblique to the grid, $\mathbf{a} = (1., .5)^T$, comparing Forward-Euler (FE) and Gauß-Seidel (GS) time-stepping on a single grid (lv1) and on 4 grid levels (lv4) with 3 pre and 1 post-iterations.

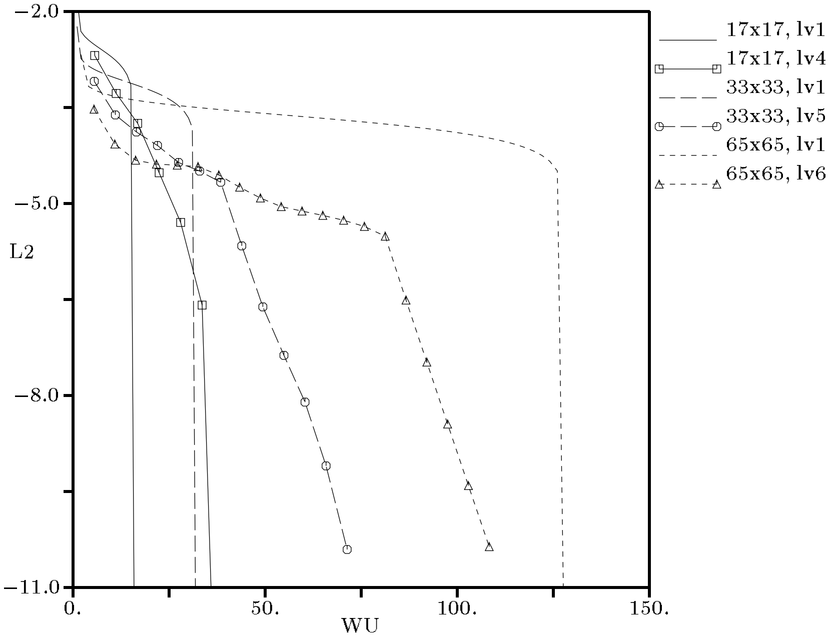


Figure 6.4: Convergence histories of the N-scheme for linear advection oblique to the grid, $\mathbf{a} = (1., .5)^T$, comparing grid sizes 17x17, 33x33 and 65x65, respectively. Forward-Euler time-stepping on 4,5 and 6 grid levels with 3 pre and 1 post-iterations is used.

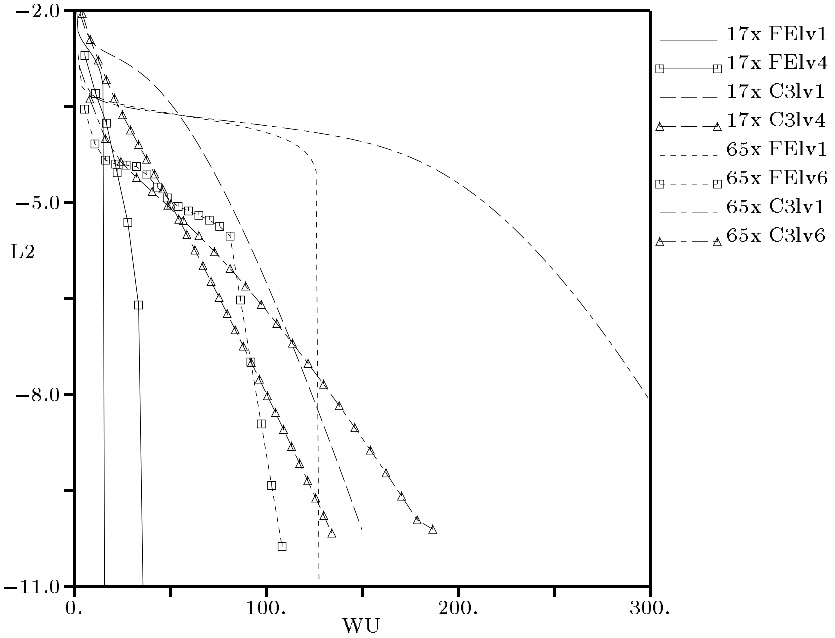


Figure 6.5: Forward-Euler and 3-stage multistage time-stepping with the N-scheme for linear advection on a 17x17 node and a 65x65 node grid. Shown are convergence histories for the single grid scheme (lv1, lines only) and the multi grid scheme (lv4,5,6, lines with symbols).

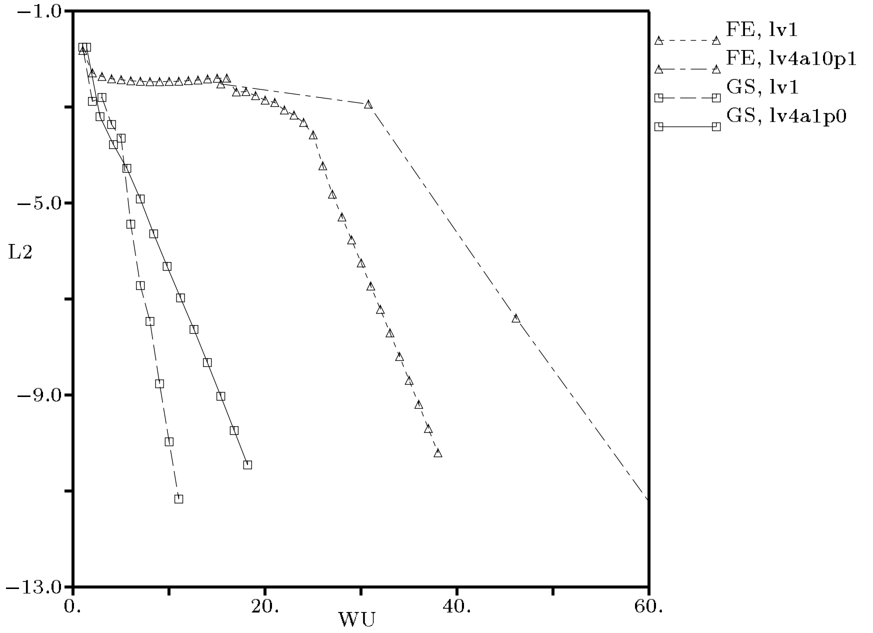


Figure 6.6: Convergence histories of the PSI-scheme on a 17x17 grid for linear advection oblique to the grid lines, $\mathbf{a} = (1., .5)^T$, using Forward-Euler (FE) and Gauß-Seidel (GS) time-stepping on a single grid (lv1) and on 4 Multigrid levels (lv4).

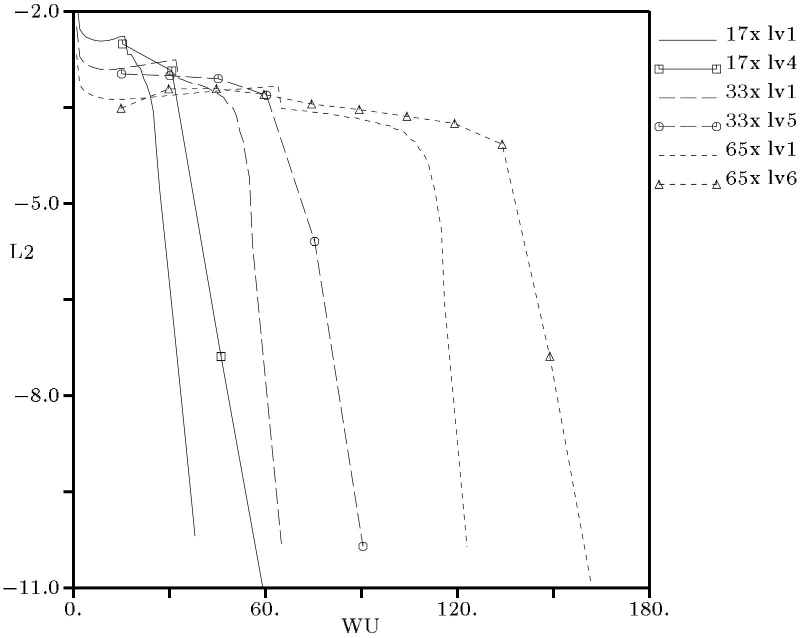


Figure 6.7: Convergence histories of the PSI-scheme for linear advection oblique to the grid, $\mathbf{a} = (1.,.5)^T$, using Forward-Euler time-stepping on a single grid and on 4,5,6 Multigrid levels on grid sizes 17x17, 33x33 and 65x65.

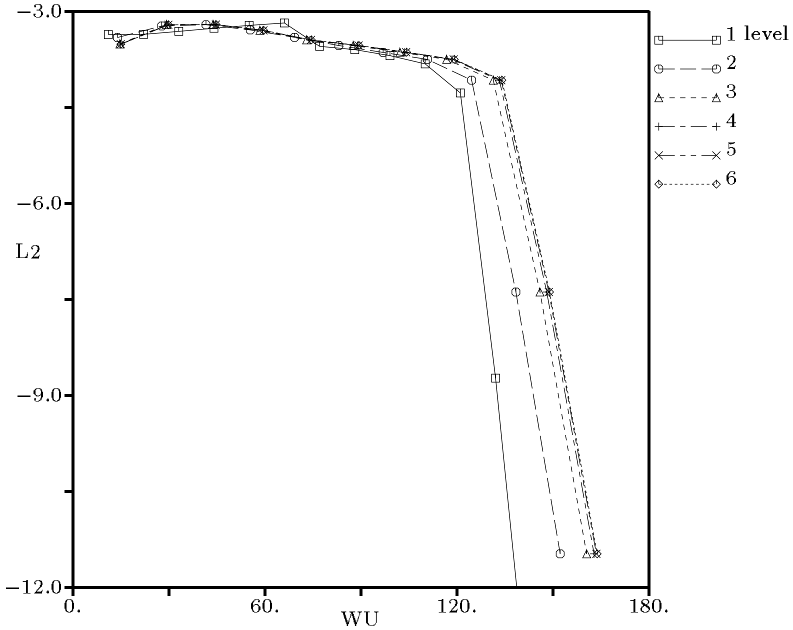


Figure 6.8: Convergence histories of the PSI-scheme for linear advection oblique to the grid, **a** $= (1.,.5)^T$, using Forward-Euler time-stepping on a 65x65 grid comparing 1 to 6 Multigrid levels.

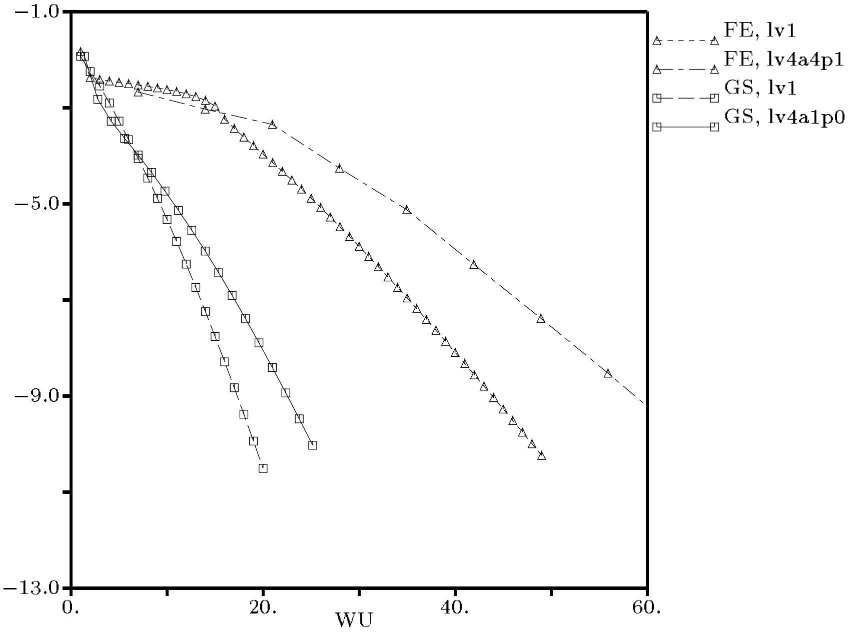


Figure 6.9: Convergence histories of the LDA-scheme for linear advection oblique to the grid, $\mathbf{a} = (1., .5)^T$, using Forward-Euler (FE) and Gauß-Seidel (GS) time-stepping on a single grid (lv1) and on 4 grid levels (lv4) with 4 pre and 1 post-iterations.

6.1.2 Burgers' Equation

As a second testcase for scalar advection, the nonlinear 2-D inviscid form of Burgers' equation is considered

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\frac{u^2}{2} + \frac{\partial u}{\partial y} = 0 \tag{6.2}$$

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \tag{6.3}$$

The nonlinear advection speed is

$$\mathbf{a} = \frac{\partial f}{\partial u}\vec{1}_x + \frac{\partial g}{\partial u}\vec{1}_y = u\vec{1}_x + \vec{1}_y \tag{6.4}$$

On the unit square, the exact solution features a compression fan that focuses into an oblique shock for the following boundary conditions (fig. 6.10):

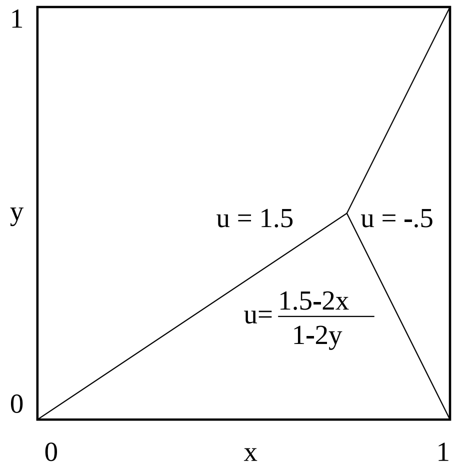


Figure 6.10: Solution for the inviscid Burgers' equation with an oblique shock on the unit square.

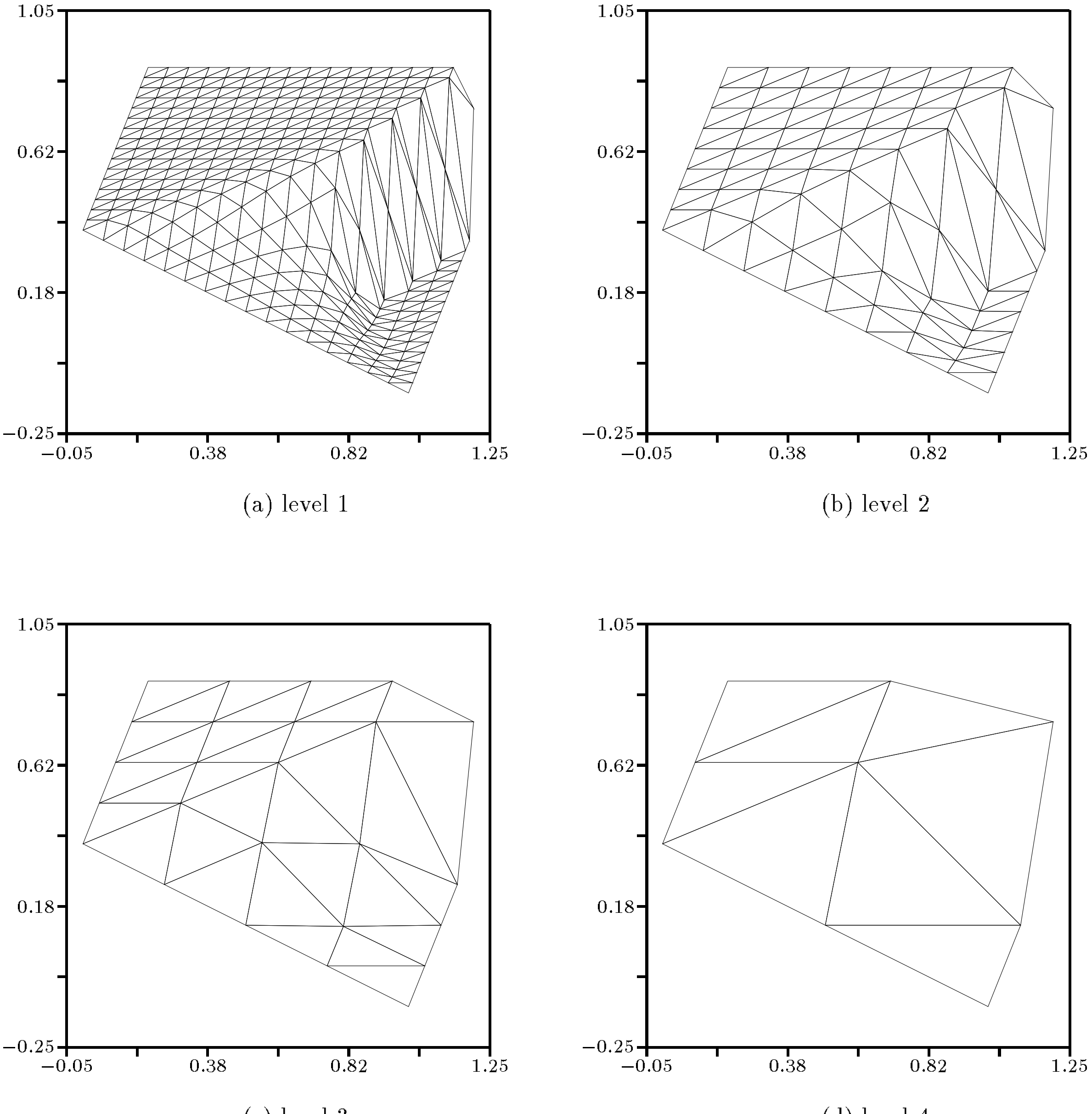
$$u = \begin{cases} 1.5 & \text{for x=0,} \\ 1.5 - 2x & \text{for y=0,} \\ -.5 & \text{for x=1.} \end{cases}$$

No boundary condition is imposed at outflow, $y = 1$. A sample solution obtained with the PSI-scheme on a 17x17 grid with four Multigrid levels is presented in figure 6.11. The schemes applied to this testcase are the N-scheme, retained for its outstanding performance with Gauß-Seidel time-stepping, and the PSI-scheme for its improved solution accuracy.

Figure 6.12(a) shows that Gauß-Seidel time-stepping converges rapidly for both schemes, but is not a direct solver with the N-scheme as it is in the linear case. Due to the converging characteristics in the vicinity of the shock, the N-scheme has a stencil that is larger than 3 nodes and not all nodes in the domain of dependence are updated when ordering the nodes downstream. Moreover, the simple sweep pattern, sweeping lexicographically from left to right in the inner loop and from bottom to top in the outer, biases the solution process toward the right-running characteristics. The backward sweep that would treat the left-running characteristics is done against the time-like direction and thus ineffective. This case clearly shows the limits for lexicographic Gauß-Seidel sweeps for general and complex flowfields. The single grid Gauß-Seidel convergence of the PSI scheme is roughly 50% slower compared to the linear case.

As in the linear case, adding Multigrid to Gauß-Seidel time-stepping does not affect the convergence dramatically. While on the coarse 17×17 grid only the N-scheme with Gauß-Seidel and on the finer 65×65 grid all but the PSI-scheme with Forward-Euler time-stepping benefit from the Multigrid scheme, the gains (or losses) are very small. In the best case, the N-scheme on the fine grid with Gauß-Seidel converges twice as fast with Multigrid, which is a far cry from the order of magnitude commonly seen with elliptic problems.

However, the results present themselves somewhat differently (fig. 6.13)when not the final fully converged solution, but just a very good approximation is sought. Such a case could arise e.g. when a relatively educated guess for a linearization is needed. A relatively good solution in terms of ‘plotting accuracy’ is obtained with after two Multigrid cycles. Thus, in a scheme for a system with both hyperbolic and elliptic components it can make sense to apply Multigrid to the hyperbolic parts.



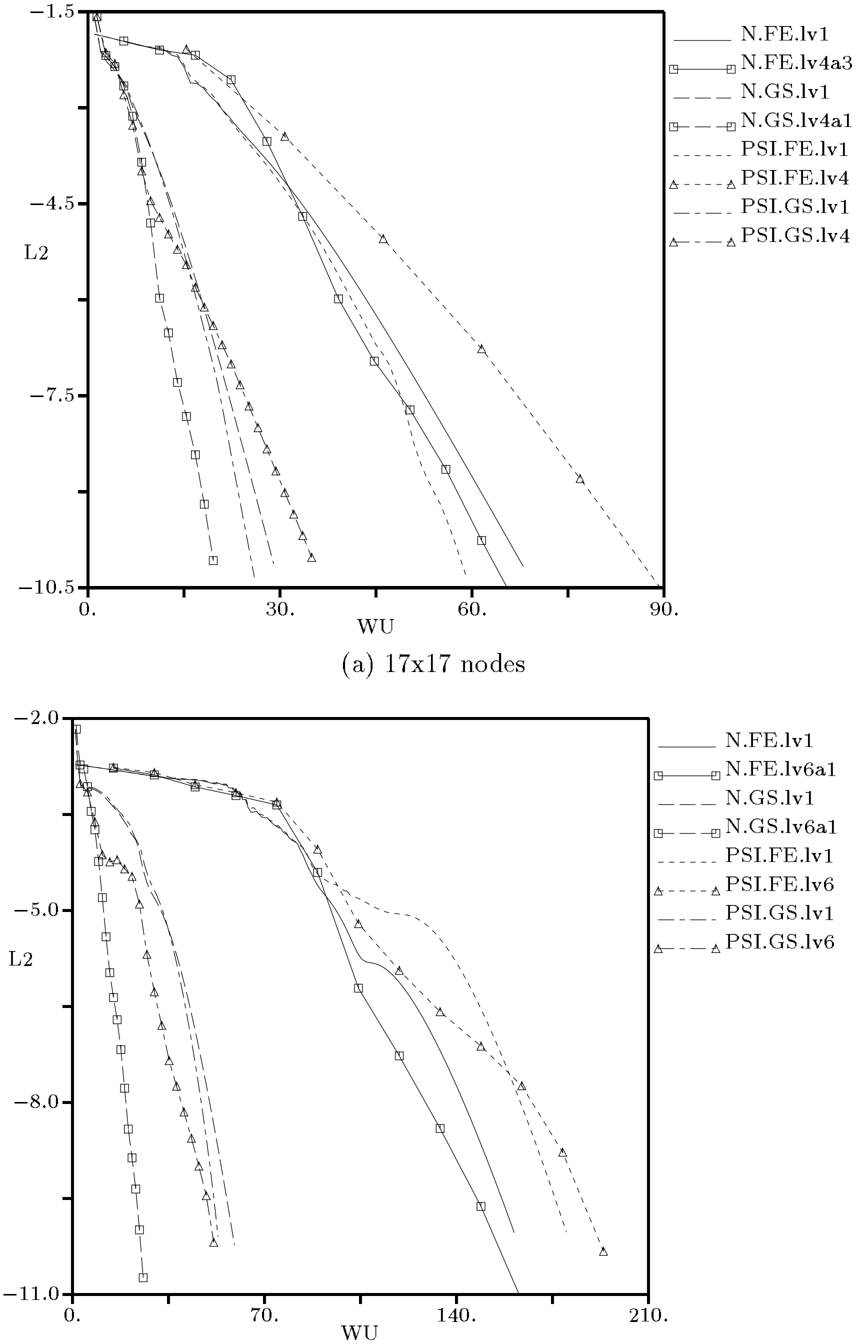
(a) level 1

(b) level 2

(c) level 3

(d) level 4

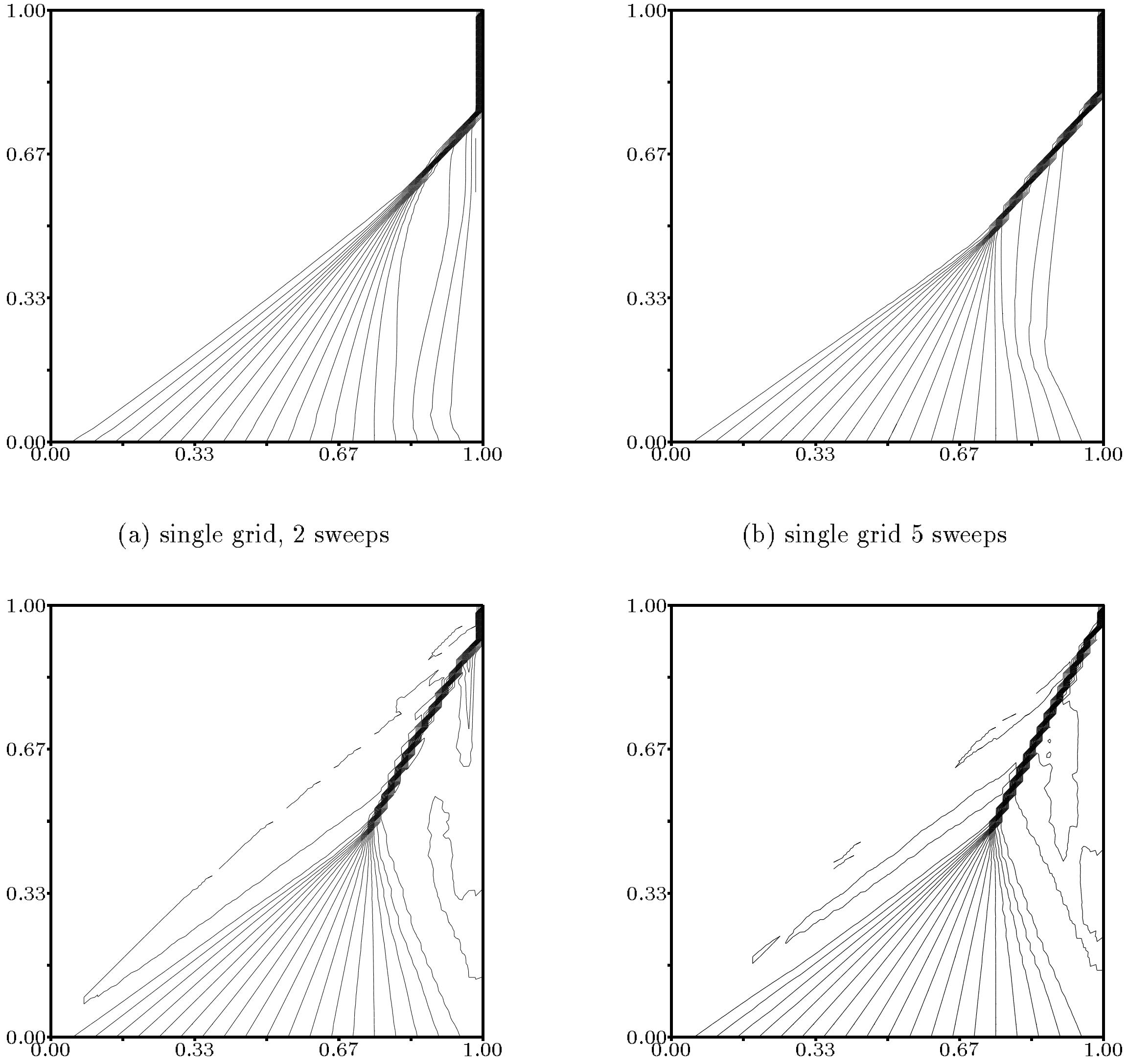
Figure 6.11: Oblique shock case for the inviscid Burgers' equation. Shown are the four Multigrid levels with 17x17 (a), 9x9 (b), 5x5 (c) and 3x3 nodes (d) respectively. Note that the coarse grid solutions telescope on the fine grid solutions due to the right-hand-side forcing function.



(a) 17x17 nodes

(b) 65x65 nodes

Figure 6.12: Convergence plots for the PSI and N-schemes applied to Burgers' oblique shock problem on a 17x17 grid (a) and a 65x65 grid (b). The plots compare Gauß-Seidel time-stepping (GS, curves on the left) to Forward-Euler time-stepping (FE, curves to the right). The single grid cases carry no symbol, Multigrid cases were obtained with 4, resp. 6 levels and carry a symbol.



(a) single grid, 2 sweeps

(b) single grid 5 sweeps

(c) Multigrid, 1 cycle

(d) Multigrid, 2 cycles

Figure 6.13: Burgers' equation solved with the PSI-scheme and Gauß-Seidel time-stepping after one (a) and two (b) Multigrid cycles with $\nu_a = \nu_p = 1$ and after work-unit equivalent 2 (c) and 5 (d) sweeps on a single grid.

6.2 The Cauchy-Riemann Equations

Both ways of diagonalizing the Euler equations presented in sections 2.4 and 2.3, the van Leer-Lee-Roe preconditioning and the canonical form, produce in the subsonic case a 2×2 system of coupled equations that is of the form of the Cauchy-Riemann equations. In the case of van Leer-Lee-Roe preconditioning, the acoustic system is defined in terms of the characteristic variables and can be transformed into the Cauchy-Riemann equations. In case of the canonical splitting, the acoustic kernel actually does work in terms of the velocity components and the Cauchy-Riemann equations are obtained for a vanishing vorticity source term of enthalpy and entropy gradients.

In order to assess the convergence properties of the proposed Lax-Wendroff scheme that treats the system, stagnation flow in a corner and the flow over a bump for the Cauchy-Riemann equations are considered.

Writing the system for $\mathbf{u} = (u, v)^T$ in the following form,

$$\frac{\partial \mathbf{u}}{\partial t} + \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \frac{\partial \mathbf{u}}{\partial x} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \frac{\partial \mathbf{u}}{\partial y} = 0, \tag{6.5}$$

produces a hyperbolic system with real eigenvalues. For the unit square, analysis of the eigenvectors yields the following characteristic boundary conditions:

$$\begin{aligned} u + v &= \mathit{const.} && \text{for} \quad y = 0, \\ v &= \mathit{const.} && \text{for} \quad x = 1, \\ u - v &= \mathit{const.} && \text{for} \quad y = 1 \quad \text{and} \\ u &= \mathit{const.} && \text{for} \quad x = 0. \end{aligned} \tag{6.6}$$

Soft boundary conditions are imposed for the bump cases. For each boundary edge, three ghost cells with vanishing thickness and three reflected or imposed boundary states $\mathbf{u}^{\mathrm{B}}$ are calculated as shown in figure 6.14. At inflow, $u = u_\infty$, at outflow $v = v_\infty$ are imposed as the boundary states $\mathbf{u}^{\mathrm{B}}$. At a solid wall three reflected velocities are imposed, $\mathbf{u}_i^{\mathrm{B}} = \mathbf{u}_i - 2\frac{\mathbf{u}_i\mathbf{n}_3}{|\mathbf{n}_3|}$, with $\mathbf{u}_i$ being $\mathbf{u}_1$, $\mathbf{u}_2$ and $\frac{1}{2}(\mathbf{u}_1 + \mathbf{u}_2)$, respectively. The resulting fluctuations evaluated over each ghostcell are distributed to the forming nodes of each cell using the Lax-Wendroff scheme. The fictive area that enters the cell-time-step calculation is set to $\mathbf{n}_3\mathbf{n}_3$, which is the maximum allowable cell-time-step for the given nodal time-steps of the forming nodes 1 and 2 (cf. equ. 3.74). Convergence is monitored as the L_2-norm of the u-velocity component.

6.2.1 Stagnation Flow in a Corner

The flow in a stagnation corner has the exact solution

$$u = 2x, \qquad v = -2y. \tag{6.7}$$

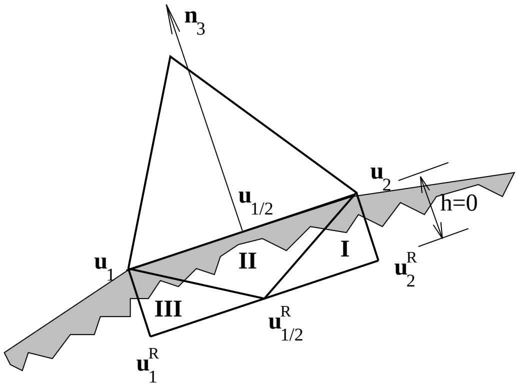


Figure 6.14: Ghostcell geometry for weak boundary conditions for the Cauchy-Riemann equations with three reflected states and three boundary ghost cells. The triangulation is contiguous at the boundary.

It is solved here on the unit square 6.15. Note that although the flowfield exhibits vanishing speeds, the modulus of the characteristic speed remains unity for all directions. The grids employed are the same regular triangulation as used for the scalar testcases in section 6.1.

Forward-Euler time-stepping, even when combined with Multigrid, converges disappointingly slow for this testcase. Using the optimal cycling parameters of $\nu_a = 10$ pre and $\nu_p = 1$ post-iterations, the convergence rate falls way short of the rate of $.7^{11} \approx .02$ per cycle predicted by local analysis (cf. sec. 3.3). Moreover, the Multigrid convergence rate is highly grid dependent and does not improve over the single grid convergence rate. Looking at the residual patterns, washboard modes that are aligned with the lower-left

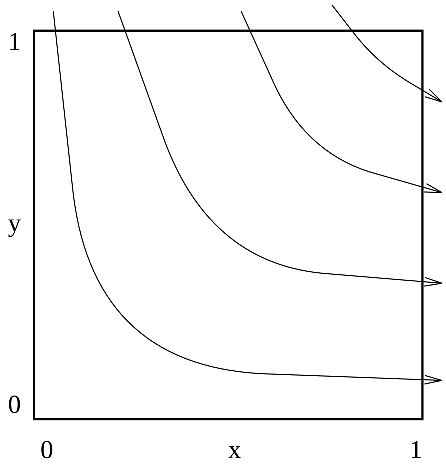


Figure 6.15: Stagnation flow in a corner with the Cauchy-Riemann equations.

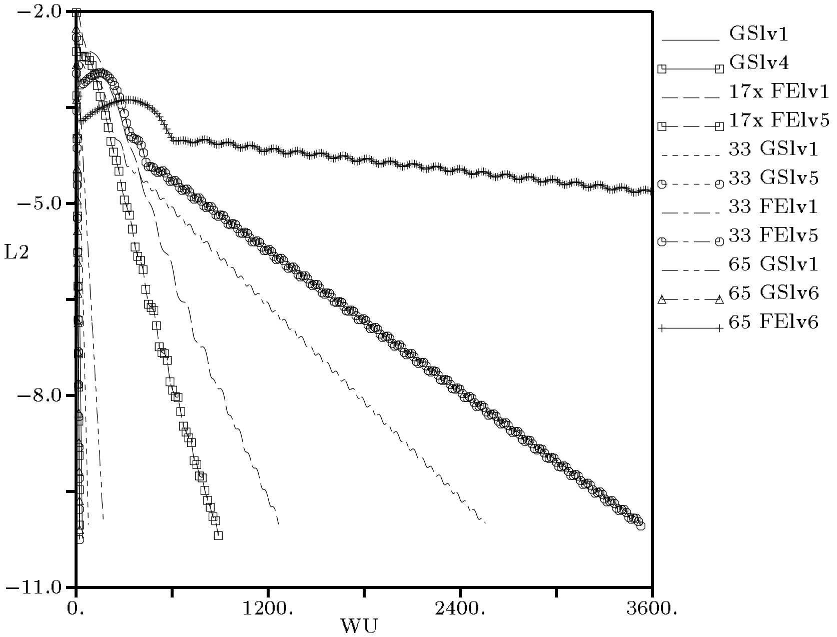


Figure 6.16: Convergence of the Lax-Wendroff-scheme for the stagnation corner in the unit square with 17x17, 33x33 and 65x65 nodes. The plots compare Gauß-Seidel time-stepping (GS, curves on the left) to Forward-Euler time-stepping (FE, curves to the right). The single grid cases carry no symbol, Multigrid cases were obtained with 4, 5, resp. 6 levels and carry a symbol.

to upper-right diagonal can be identified and that are not damped by the Lax-Wendroff scheme with Forward-Euler time-stepping on this configuration.

Gauß-Seidel time-stepping applied to the stagnation corner 6.16 produces a convergence rate of $\sigma = .3$ with the optimal cycle parameters of $\nu_a = 1$ pre and $\nu_p = 0$ post-iterations, thus improving over the rate predicted for Forward Euler. The convergence rate is virtually grid-independent. The rate even exceeds the convergence rate of a staggered grid formulation for the Cauchy-Riemann equations given in [82] of $\sigma = .5$ for lexicographic ordering. This indicates that the enhanced damping of errors by the Multigrid scheme cooperates favorably with the advection of errors by the Lax-Wendroff scheme.

6.2.2 Channel Flow with a Circular Bump

The geometry of the GAMM channel bump [112] is shown in figure 6.21. The bump height for the Cauchy-Riemann testcases was chosen as .014. A sample set of Multigrid-levels is shown in figure 5.18. The coarsening ratio of cells between adjacent Multigrid levels, ideally 25 %, is slightly exceeded for the initial grids, since the coarsening algorithm

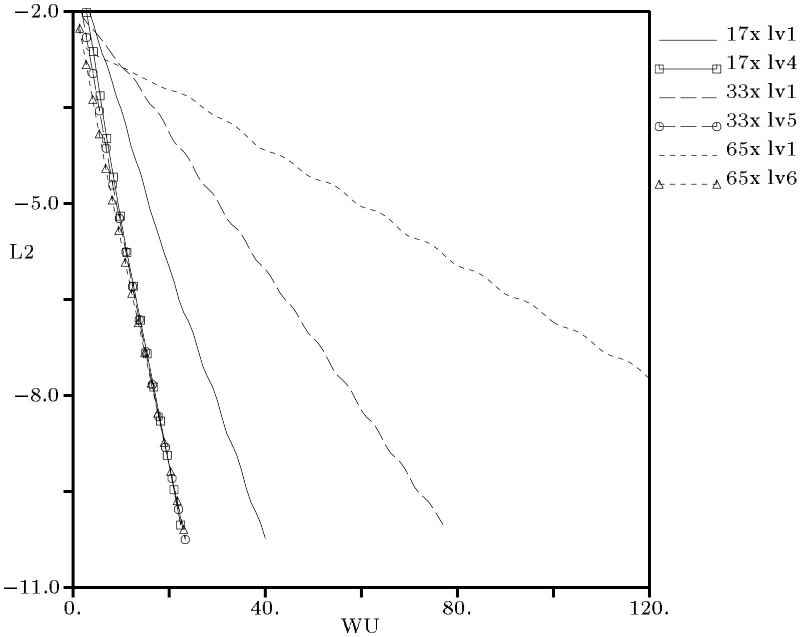


Figure 6.17: Convergence of the Lax-Wendroff-scheme using Gauß-Seidel time-stepping for the stagnation corner in the unit square with 17x17, 33x33 and 65x65 nodes. Multigrid cases were obtained with 4, 5, resp. 6 levels and carry a symbol.

always chooses to over-coarsen a boundary segment with an even number of boundary nodes (cf. sec. 5.6). For the coarsest level 1 the rate drops as some boundary segments do not allow further coarsening. This effect is even more pronounced on the coarsened grids below the the 851 node grid for the same configuration that is used in the calculations for figure 6.19.

Contours of total velocity of the solution on the finest level of the 1643 node grid are given in figure 6.18.

The spurious mode that prevented fast convergence on the regular unit-square grid with aligned diagonal is not present on the isotropic grids created with FroD (cf. chapter V). The convergence rate with the optimal cycle parameters $\nu_a = 8$ pre and $\nu_p = 1$ post-iterations, is found to be $\sigma_{c,FE} = .35$ per cycle, which should be achieved with only three iterations of the Lax-Wendroff scheme at $\sigma_{LW} = .7$.

Gauß-Seidel time-stepping converges roughly three times as fast, producing a convergence rate of $\sigma_{c,GS} = .57$ per cycle. Since in terms of CPU-time one Gauß-Seidel work-unit equals roughly six Forward-Euler work-units, the total cost is slightly in favor of Forward-Euler time-stepping with a total of nine iterations per level, $\sigma_{c,GS} \cdot 6/9 = .38 > \sigma_{c,FE}$. The reduction in convergence rate from the stagnation case of nearly 50% has to be attributed

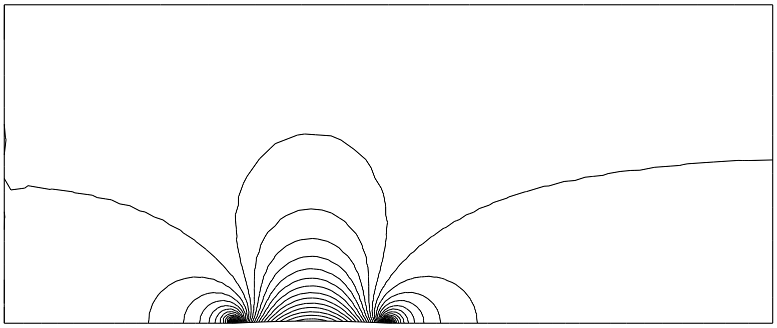


Figure 6.18: Solution for the GAMM-channel using the Cauchy-Riemann equations on a 1643 node grid. Shown are isolines of total velocity.

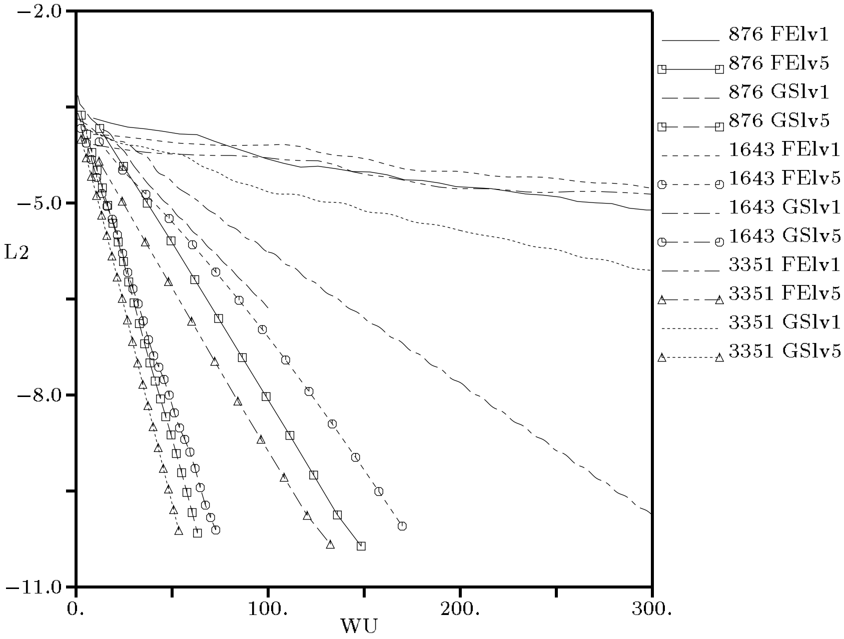


Figure 6.19: Convergence of the Lax-Wendroff-scheme for the GAMM channel.

to the irregularity of the unstructured grid: the coarsened cells do not telescope on fine grid cells as they do in the regular grids (figs. 6.2 and 5.18).

A look at the dependence of the Multigrid convergence rate on the grid size shows the interesting fact that the finest grid with 3351 nodes actually converges fastest in terms of work-units. This is due to the fact that the finer grid maintains a more optimal coarsening ratio throughout its levels and thus the Multigrid scheme is more efficient.

Although tremendous acceleration with Multigrid is achieved for the Lax-Wendroff applied to the Cauchy-Riemann equations, the results still fall short of simple staggered-grid schemes as presented in [82]. A better formulation than the arbitrary choice than the one of equation 6.5 that relates u_t to the divergence and v_t to the curl of the velocity field might be found.

However, when compared to convergence rates found in state-of-the-art dimensionally split Upwind schemes the convergence with the Lax-Wendroff scheme is excellent. Were it possible to extend these convergence rates to the Euler system, a vast improvement could be achieved.

6.3 The Euler Equations

Solving the subsonic and transonic Euler equations accurately and efficiently is the prime objective of this work. It has been shown that a Fluctuation-Splitting space discretization (chap. III) used with the hyperbolic-elliptic spitting (sec. 2.4) yields a very accurate solution with good robustness [31, 32]. However, their results exhibit a strikingly slow convergence of such schemes, and a Fourier analysis is yet to be done. A likely hypothesis is that the elliptic part is responsible for the slower convergence of the scheme with hyperbolic-elliptic splitting since it is the one modeled differently from purely hyperbolic Finite-Volume Euler schemes. In this section Multigrid will be applied to the split system, most importantly to treat the elliptic system.

At first glance, recalling the poor results on the performance of Multigrid with hyperbolic equations of section 6.1.2, a scheme that treats the hyperbolic part with Forward-Euler time-stepping scalar advection schemes and that treats the elliptic part with Multigrid seems very attractive. Such an approach has been taken by Ta'asan [113] (cf. sec. 2.3), first updating enthalpy and entropy and then solving the acoustic elliptic system. In a Cell-Vertex framework however, the cost of one iteration will roughly double if the updates are done consecutively since each cell has to be visited twice.

On the other hand, the results in section 6.1.2 indicate that there are benefits of applying Multigrid to a purely hyperbolic problem if a 'very educated guess' is sought rather than convergence to machine precision. This is precisely the case in the Hyperbolic-Elliptic splitting, where a good approximation to the entropy and enthalpy field will increase the accuracy of the splitting. Since the same time-stepping and Multigrid schemes work well with both parts of the solution, the extra effort that is incurred by splitting the updates seems not to be balanced by any benefits of using different time-steppings. Thus, the same Multigrid scheme that is used for the elliptic part will be applied here to the advective part and all unknowns are updated simultaneously using the same time-stepping scheme.

The optimal cycle parameters for the Euler equations were in general found to be $\nu_a = 8$ pre and $\nu_p = 1$ post-iterations for Forward-Euler time-stepping and $\nu_a = 1$ pre and $\nu_p = 0$ post-iterations for Gauß-Seidel time-stepping.

The nodal preconditioning [26] that is used improves the stability of the scheme. Rather than the stability limit of $\nu_c \nu_n \leq .5$ that is predicted by Fourier analysis of the Lax-Wendroff scheme on triangles [33], a practical limit of $\nu_c \nu_n \leq 1$ is observed and used in the calculations. The residual is monitored as the L_2-Norm of the residual of the density (cf. equ. 6.1).

The boundary conditions are applied as overlapping ghostcells as described in [61] and shown in figure 6.20. The fluctuation in triangle III vanishes in the limit of $h \to 0$. Node 1 receives contributions from triangles I and II. Since the ghostcells for node 2 overlap with the ones of node 1, only scalar upwind schemes can be used in this configuration

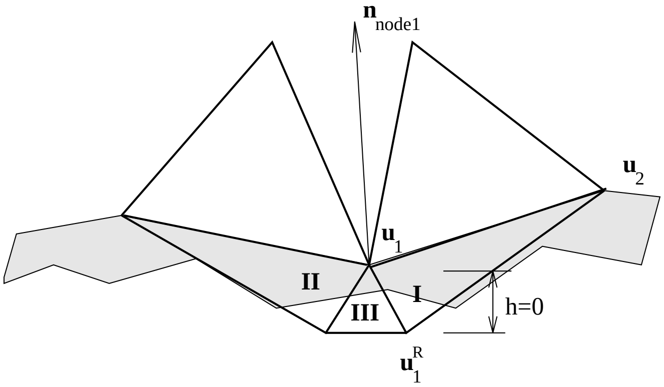


Figure 6.20: Ghostcell geometry for weak boundary conditions for the Euler equations with one reflected state and three boundary ghost cells. The triangulation is not contiguous at the boundary.

that become single target for the case that the gradient is normal to the face. At the boundary a normal is defined to choose the wave direction and any consistent wave-model that reverts to a 1-D splitting can be used. Paillère [34] suggests to use Roe's six-wave model D [114]. The ghoststate is reflected for a solid wall boundary condition. For a subsonic inlet the total pressure and for a subsonic outlet the static pressure are imposed.

6.3.1 Subsonic Flow over a Circular Bump

Subsonic flow over the GAMM channel [112] has been chosen as a subsonic testcase. The geometry of the channel with the circular bump has been proposed for the GAMM workshop [112] as shown in figure 6.21.

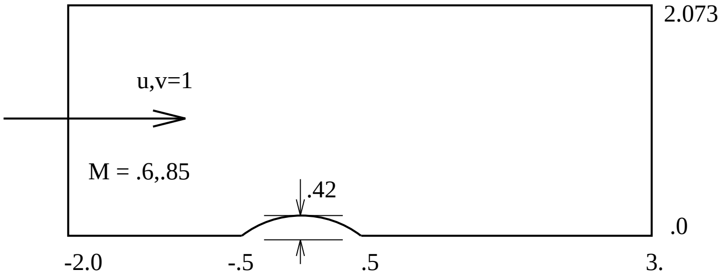
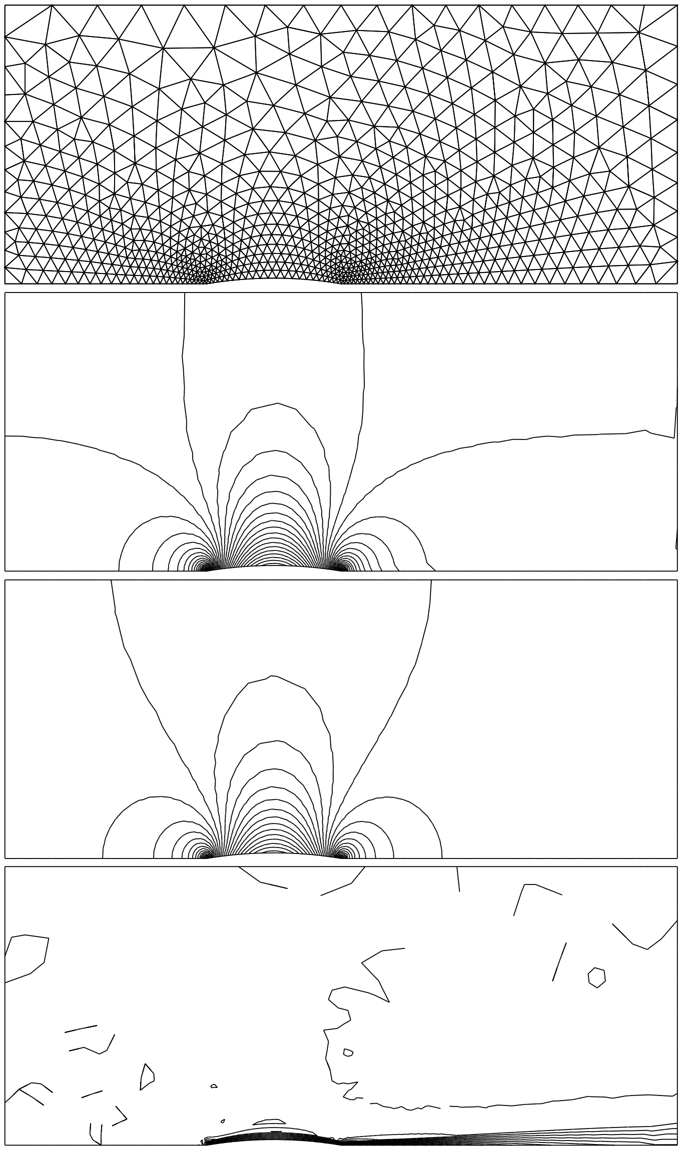


Figure 6.21: The GAMM channel for subsonic and transonic flows. The height of the bump is exaggerated to make it visible.

With the available coarse resolutions, the stagnation regions at the ends of the circular arc are very underresolved and present no difficulties to the preconditioning. The solution shown in figure 6.22 is very accurate considering that the grid has only 863 nodes. Very little entropy is produced along the bump.



Mach number:

$$\begin{aligned} M_{min} &= 0.4889 \\ M_1 &= 0.4900 \\ \Delta M &= 0.0050 \\ M_{max} &= 0.6936 \end{aligned}$$

pressure:

$$\begin{aligned} p_{min} &= 1.834 \\ p_1 &= 1.840 \\ \Delta p &= 0.010 \\ p_{max} &= 2.148 \end{aligned}$$

entropy:

$$\begin{aligned} s_{min} &= \text{-.706E-04} \\ s_1 &= \text{-.600E-04} \\ \Delta s &= \text{0.200E-04} \\ s_{max} &= \text{0.400E-03} \end{aligned}$$

Figure 6.22: Solution for the subsonic GAMM channel, free-stream Mach-number .6 with the hyperbolic-elliptic splitting. The solution shows very little entropy production over the bump.

The optimal Multigrid parameters were determined by numerical experiments. The results for various values of pre and post-iterations ν_a, ν_p are shown for Gauß-Seidel in figure 6.23(a) and indicate that there is no marked influence of these parameters. A sawtooth-cycle with $\nu_a = 1, \nu_p = 0$ is used for Gauß-Seidel time-stepping. A similar insensitivity to the cycling parameters is found for Forward-Euler suggesting as values $\nu_a = 8, \nu_p = 1$.

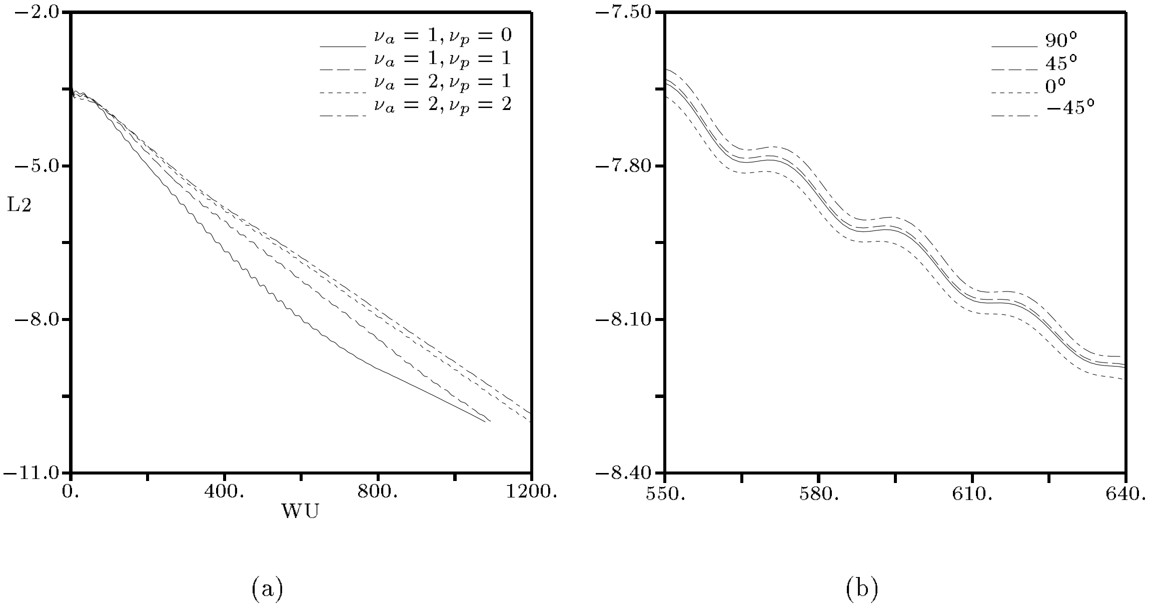


Figure 6.23: Convergence for for the subsonic GAMM channel, 863 node grid. Fig. a shows the variation of the convergence with the cycling parameters, fig. b shows a close-up of the convergence with varying orientation of the tilted line that orders the cells.

Fig 6.23(b) shows the influence of the cell ordering for the Gauß-Seidel sweeps. The cells are ordered with increasing distance to a tilted line. E.g. a choice of a line with $\Delta x = -.001, \Delta y = 1.$ through the origin, thus nearly 90° from the horizontal, will order the cells of the GAMM channel in a inner loop from bottom to top and in the outer loop from left to right. Four orientations of the tilted line are shown in the figure with virtually identical results down to the small wave-reflections present in the sinusoidal oscillations of the convergence. This is a surprising result since the advection of enthalpy and entropy should benefit from a streamwise ordering with a 90° line. Apparently the convergence of the subsonic case is determined by the elliptic system with the enthalpy and entropy fields converging very quickly.

The convergence behavior shows a pattern that remains the same for all subsonic and transonic calculations presented here (fig. 6.24). The slowest convergence is obtained to no surprise with a Forward-Euler single grid scheme. Applying Gauß-Seidel on the single grid amounts to a tremendous acceleration of the convergence. For the first drop of two orders of magnitude of the residual, the single grid Gauß-Seidel scheme performs as well as the 5-level Multigrid Forward-Euler scheme. The best performance as measured in work-

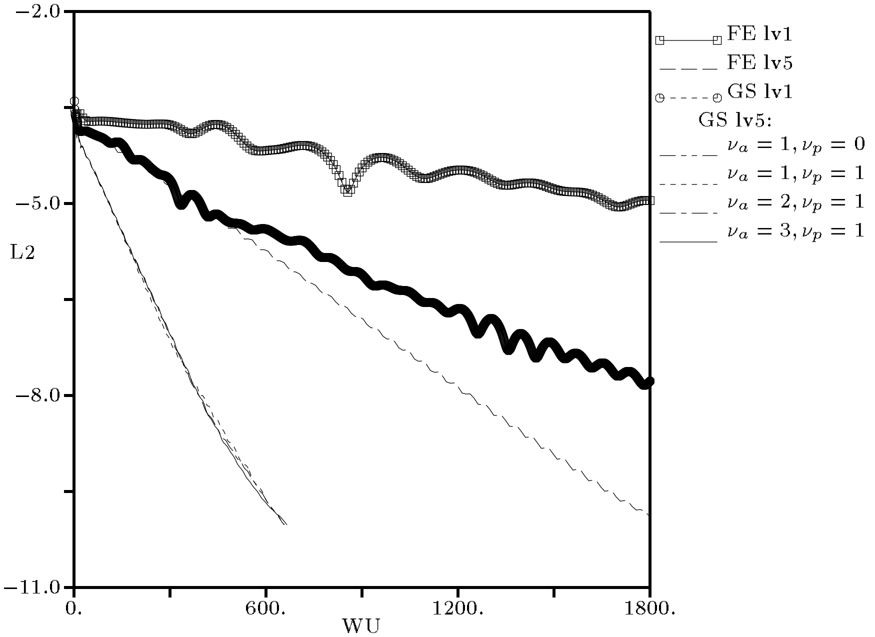


Figure 6.24: Convergence curves for the subsonic GAMM channel, 863 node grid. Shown are single-grid (marked with a symbol) and Multigrid convergence histories (lines only) for Gauß-Seidel (GS) and Forward-Euler (FE) time-stepping.

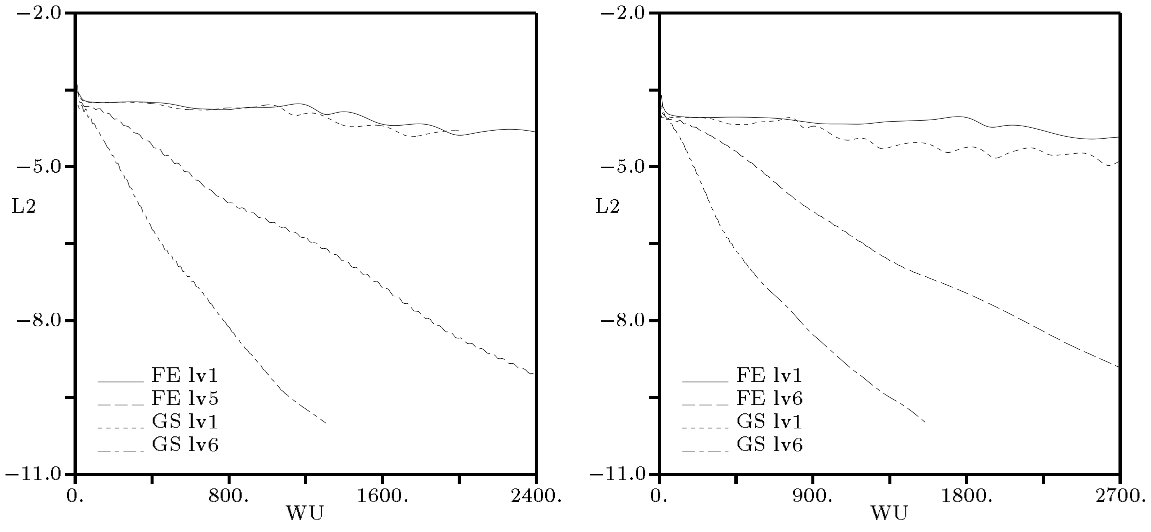
units is obtained from the Multigrid Gauß-Seidel scheme. Varying the cycle parameters actually seems to have very little influence on the convergence of that scheme.

Note however, that in the current implementation, one Gauß-Seidel work-unit takes roughly 6 times the CPU time of one Forward-Euler work-unit. Thus, the work-unit ratio being found around 1:3, Gauß-Seidel is, in the current implementation, twice as expensive as Forward Euler for subsonic cases.

This finding is repeated on the finer grids. Gauß-Seidel single grid compares less favorably to Multigrid Forward Euler with increasing grid size. The ratio of 1:3 of convergence rates in favor of Gauß-Seidel when measuring CPU-time is again found on the finer grids.

Grid independence of the convergence is good, all curves tend to show little decay in the convergence rate for increased grid size. Note however, that the sizes tested are still relatively small and that for the smallest grid size a penalty for poor cell-nesting is incurred as explained in the context of section 6.2.2. The convergence rate of the Gauß-Seidel scheme tends to decay a little in the later stages of convergence and the effect becomes more pronounced with increasing grid size. Analysis of the temporal behavior of the coupling between elliptic and hyperbolic parts might shed some light on this phenomenon.

Convergence of about 7 orders of magnitude is achieved within 800 work-units for Gauß-Seidel and around 2000 work-units for Forward-Euler. While these values do not



(a) (b)

Figure 6.25: Convergence curves for the subsonic GAMM channel, 1592 nodes (a) and 3498 nodes (b).

indicate remarkable performance, it is a vast improvement over the single grid rate.

However, performance is impressive when not the convergence to machine precision is sought, but to an intermediate level where all flow features are resolved to 'plotting precision'. Such an approach has been taken in [12]. The solutions to the subsonic GAMM channel on a coarse 863 node grid are shown in figure 6.27 after 10, 20, 30 and 40 cycles, respectively.

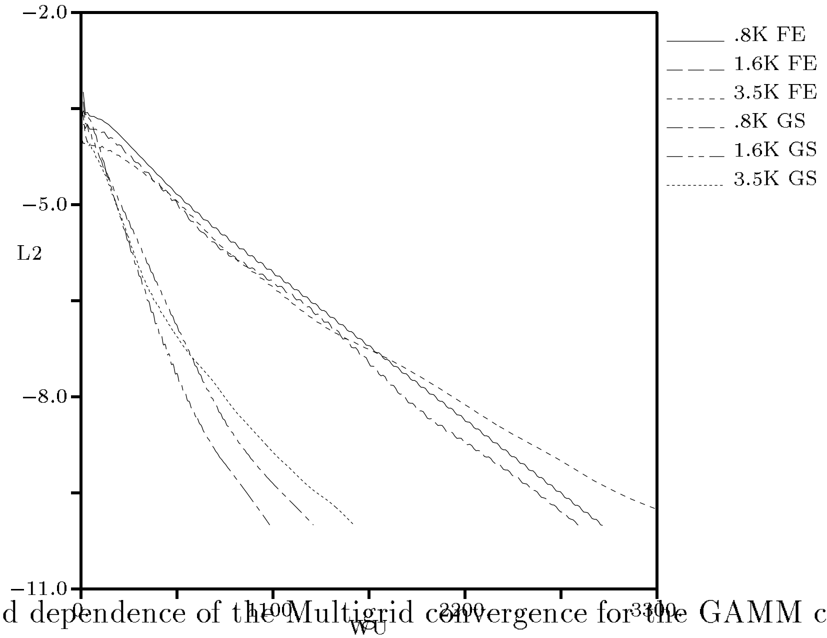


Figure 6.26: Grid dependence of the Multigrid convergence for the GAMM channel, grid sizes compared are 863, 1564 and 3498 nodes. Shown are Forward Euler ($\nu_a = 8$, $\nu_p = 1$) and Gauß-Seidel ($\nu_a = 1$, $\nu_p = 1$) convergence histories.

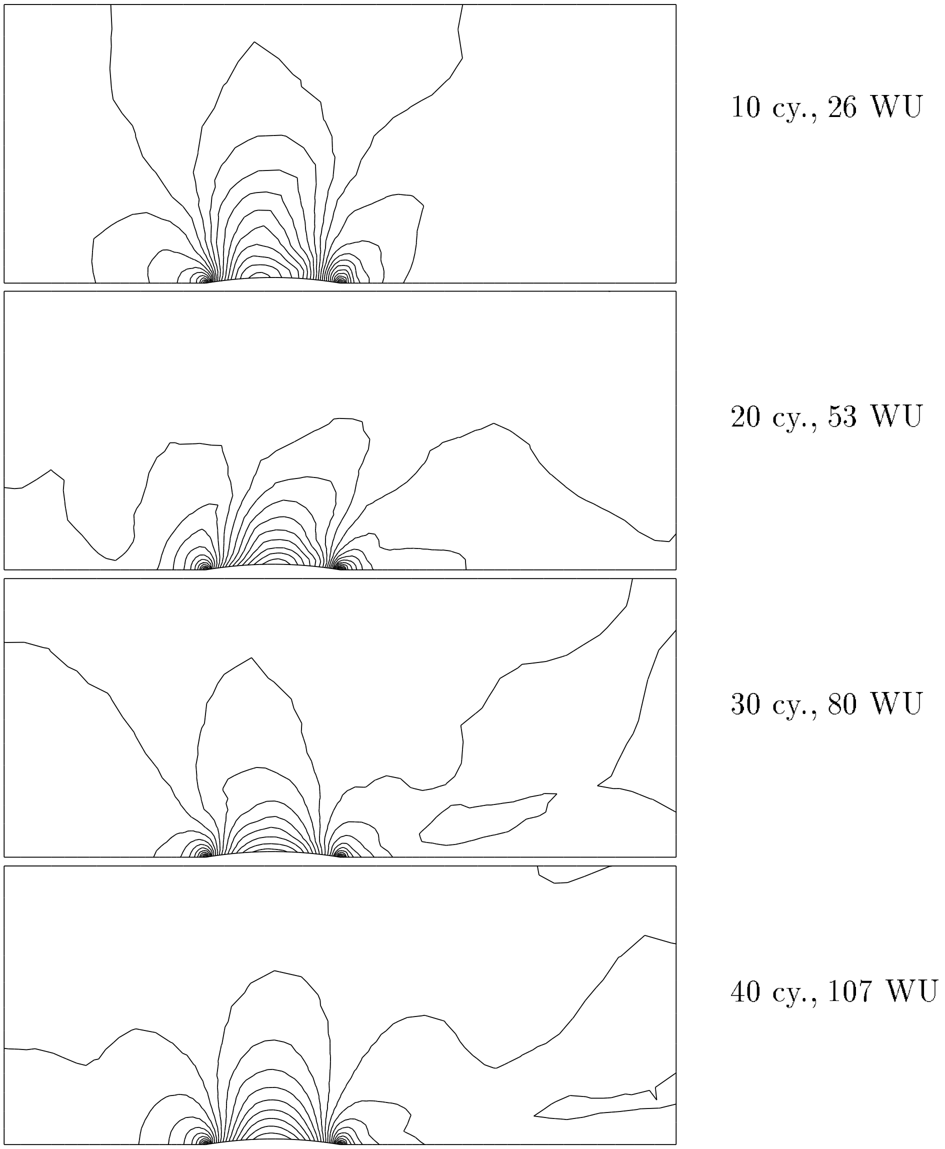


Figure 6.27: Mach number contours of an Euler solution after a few Multigrid cycles. Solution for the subsonic GAMM channel, 863 nodes, free-stream Machnumber .6 with the Hyperbolic-Elliptic splitting, PSI-LW schemes and nodal preconditioning. Cycle parameters were $\nu_a = \nu_p = 1$.

6.3.2 Transonic Flow over a Circular Bump

The GAMM channel geometry is simulated with an inflow Mach number of .85 for a transonic flow through the channel. Of major concern is the dissipation applied in the shock region. Unfortunately, the Lax-Wendroff scheme applied is not very sophisticated in its choice of artificial dissipation: only the scalar cell-time-step applied to the elliptic system can be raised to increase the dissipation. However, since the two time-steps are coupled, $\nu_c \nu_n \leq 1$, more dissipation means a loss in convection and a slowing down of the convergence.

Three CFL-number combinations have been tested and the Mach number contours are shown in figure 6.28. The solution with $\nu_c = 4$ does show improved monotonicity in the region around the foot of the shock. However, the entropy layer after the bump indicates that excessive dissipation was applied globally to the solution. Pressure and entropy contours for the $\nu_c = \nu_n = 1$ solution are given in figure 6.29.

The convergence rate for the 863 node grid in figure 6.30 shows a nearly linear dependence on the nodal CFL-number, little benefit is derived for Gauß-Seidel Multigrid from the increased smoothing rate of the Lax-Wendroff scheme with an increased cell-CFL-number. The single grid Gauß-Seidel scheme does not compare in performance with the Multigrid Forward-Euler scheme for the first orders of convergence as it did in the subsonic case (fig. 6.24). A comparison of the two single grid schemes in larger scales than those of figure 6.30 shows a better convergence of the Gauß-Seidel in work-units with a factor of three.

Multigrid Forward-Euler still converges slower than Gauß-Seidel in terms of work-units, however, the gap narrowed from 1:3 in the subsonic to 1:2 in the transonic case, making Forward-Euler a third as expensive in terms of CPU.

Similar observations can be made studying the convergence of the schemes on the finer grids (fig. 6.31). A remarkably poor convergence is shown by the Forward-Euler Multigrid scheme. The oscillations around the shock completely inhibited convergence for the cycle parameters $\nu_c = 8, \nu_n = 1$. Only with more smoothing by using $\nu_c = 12, \nu_n = 1$ slow oscillatory convergence could be achieved. The Gauß-Seidel scheme seemed unaffected.

The comparison of convergence histories for all three grid sizes are shown in figure 6.32. All methods show little grid dependence in the transonic case.

Comparing convergence for the subsonic case (fig. 6.26) to the transonic case (fig. 6.32) shows that the convergence is reduced for the transonic bump to 1/2 for the Gauß-Seidel scheme and to 2/3 for Forward-Euler. Transonic convergence is definitely not as good as might be expected from our proper characteristic treatment, but the improvements over the single grid performance is dramatic.

6.4 Summary

A two-dimensional Multigrid scheme for vertex-centered solution algorithms has been implemented, using the benefits of the vertex-nested grids available through the Frontal Delaunay method presented in section 5.3. The Multigrid scheme has been applied to a Fluctuation-Splitting discretization for scalar equations, the Cauchy-Riemann equations and the Euler equations. Gauß-Seidel time-stepping, Forward-Euler time-stepping and multistage time-stepping have been implemented and compared.

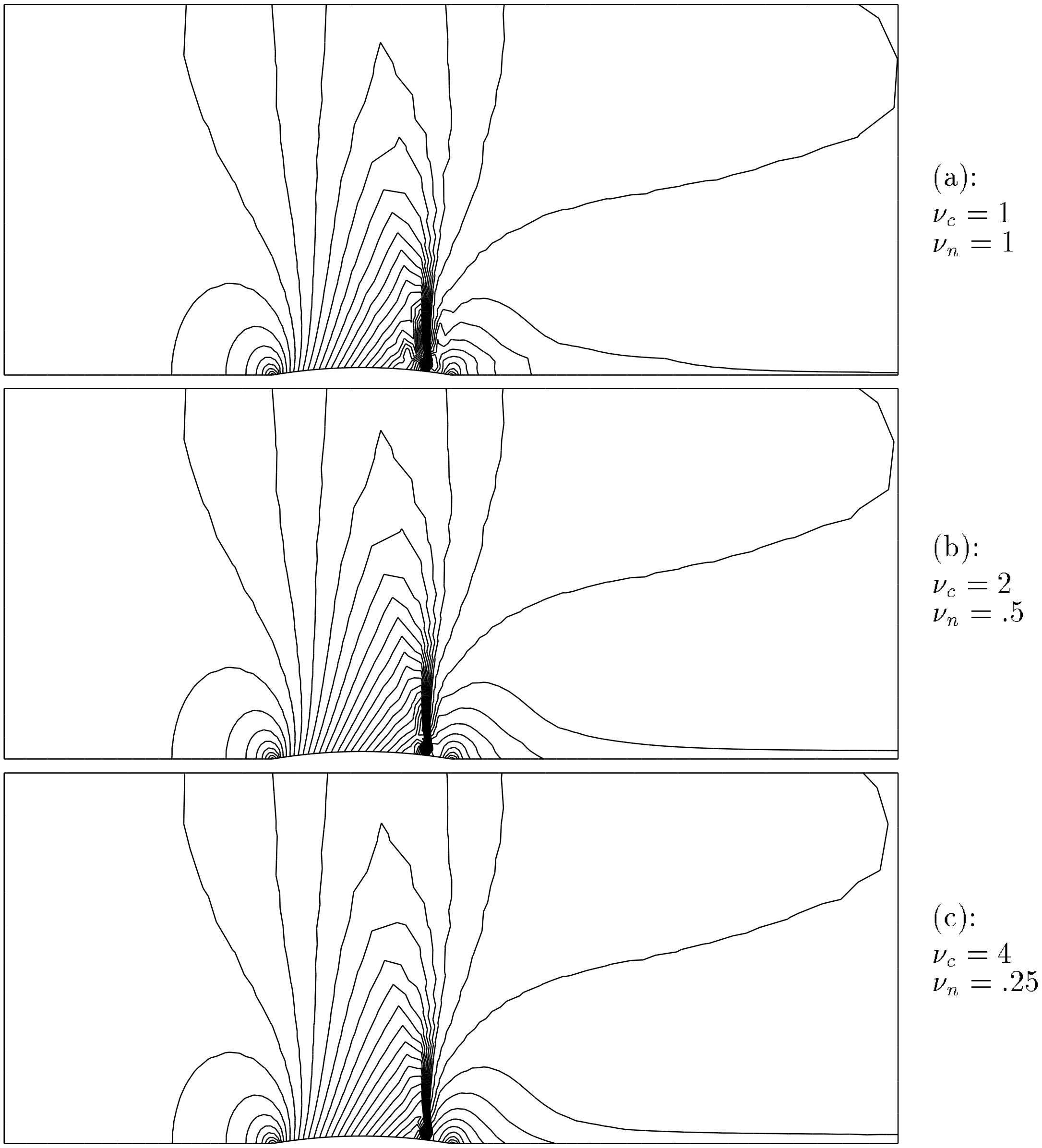
Not too surprisingly, the Multigrid scheme offers little for the scalar equation. Scalar hyperbolic problems are best treated with a space-marching scheme that follows the characteristics. The genuinely multidimensional first order scheme, the N-scheme, becomes such a direct solver when applied to linear problems with Gauß-Seidel time-stepping in the characteristic direction. Higher-order genuinely multidimensional Upwind schemes like the PSI scheme benefit less from Gauß-Seidel: with the constraint of compactness, the stencil has to be extended laterally, making it impossible to find a sweep ordering that updates all nodes in the domain of dependence. Multigrid actually more often slows convergence rather than accelerating it for these schemes when considering the convergence to machine precision. However, the initial convergence is accelerated by Multigrid, which is a desired property in the context of seeking a decently converged state for a linearization. For the non-linear PSI-scheme Gauß-Seidel and Forward-Euler time-stepping perform equally well in terms of CPU time consumed.

For the elliptic system of Cauchy-Riemann equations Multigrid proves most efficient. The convergence rates with the Lax-Wendroff scheme are quite adequate and compare acceptably to other methods. A better formulation for the hyperbolic time-dependent system might be sought that reduces to a proper elliptic procedure, a div-curl formulation comes to mind. As in the case of the scalar advection, Gauß-Seidel and Forward-Euler time-stepping yield very comparable efficiency in terms of CPU time for lexicographic ordering.

Both temporal discretization methods, Gauß-Seidel and Forward-Euler seem equally well suited to treat the sub- and transonic Euler-system that is mixed hyperbolic-elliptic in the steady state.

Since the elliptic system receives a treatment that reflects the mathematical behavior of the steady state, a convergence rate comparable to the one of incompressible methods should be possible. However, in the present case the sum is less than its parts: the overall performance falls way short of the convergence results of the scalar advection schemes or the Cauchy-Riemann equations. The insensitivity of the convergence results to the sweep direction in Gauß-Seidel suggest that the hyperbolic part is not dominant for the convergence behavior.

Apparently, the inherent assumption has to be revised that the coupling terms between the two parts of the system are negligible in their influence on the convergence. These terms do appear as variable coefficients in the Cauchy-Riemann like acoustic system in the hyperbolic-elliptic splitting.

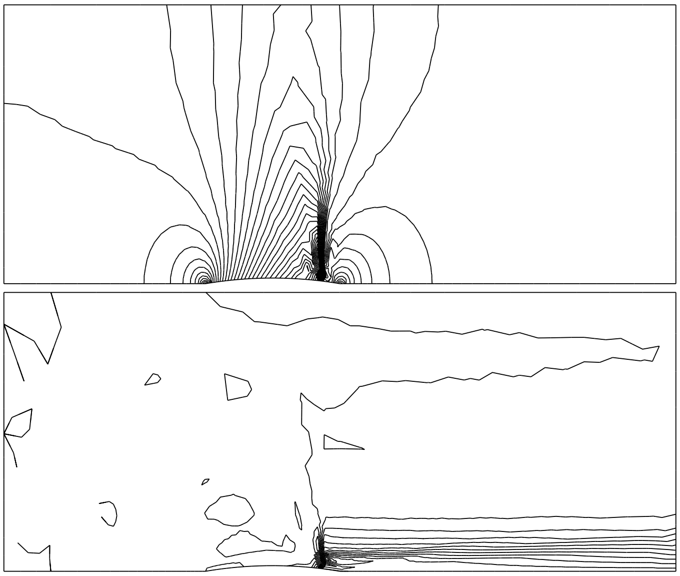


(a):
$\nu_c = 1$
$\nu_n = 1$

(b):
$\nu_c = 2$
$\nu_n = .5$

(c):
$\nu_c = 4$
$\nu_n = .25$

Figure 6.28: Solution for the transonic GAMM channel, free-stream Mach-number .85. The solutions are not monotonic for the cell-CFL-number $\nu_c = 1, 2$. Only for $\nu_c = 4$ the solution becomes nearly monotonic, however the globally added dissipation creates a stronger entropy layer.



pressure:

$$p_{min} = 0.493$$
$$p_1 = 0.500$$
$$\Delta p = 0.025$$
$$p_{max} = 1.213$$

entropy:

$$s_{min} = -0.0039$$
$$s_1 = -0.0030$$
$$\Delta s = 0.001$$
$$s_{max} = 0.0134$$

Figure 6.29: Pressure and entropy contours for the transonic GAMM channel, free-stream Mach-number .85 with $\nu_c = 1, \nu_n = 1$.

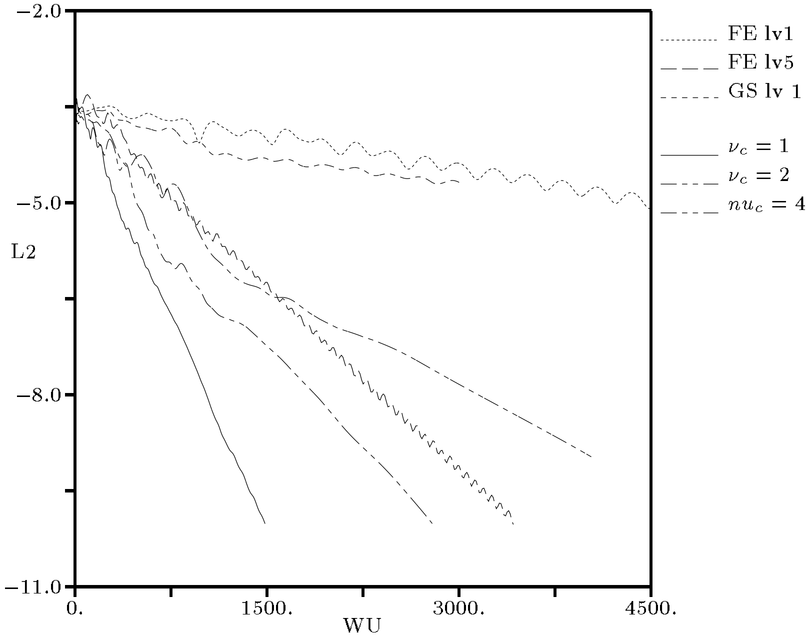


Figure 6.30: Convergence curves for the transonic GAMM channel, 863 nodes.

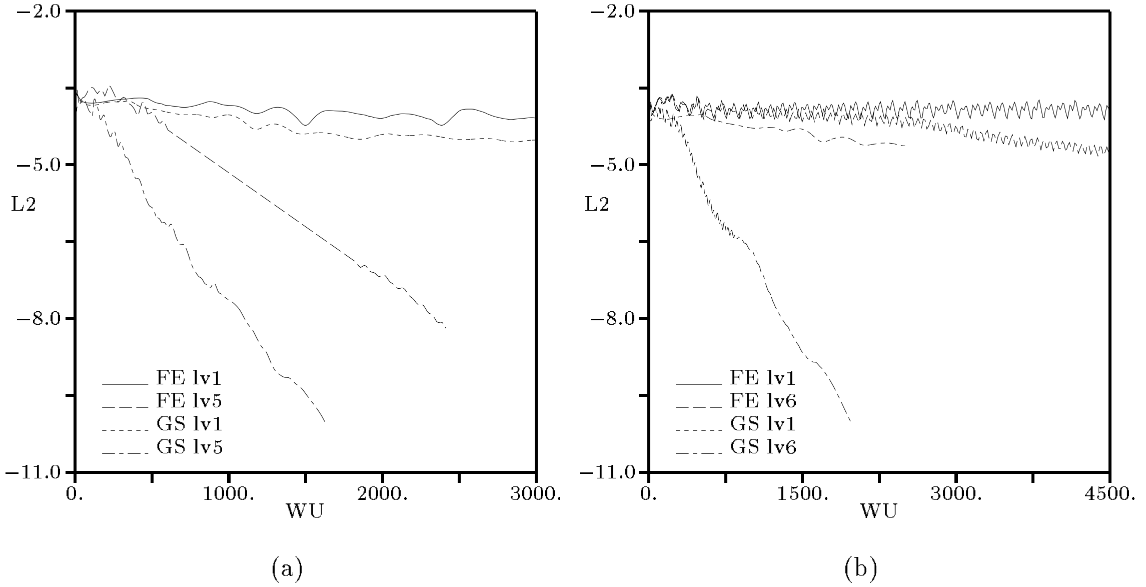


Figure 6.31: Convergence curves for the transonic GAMM channel, 1592 nodes (a) and 3498 nodes (b), $\nu_c = 1$.

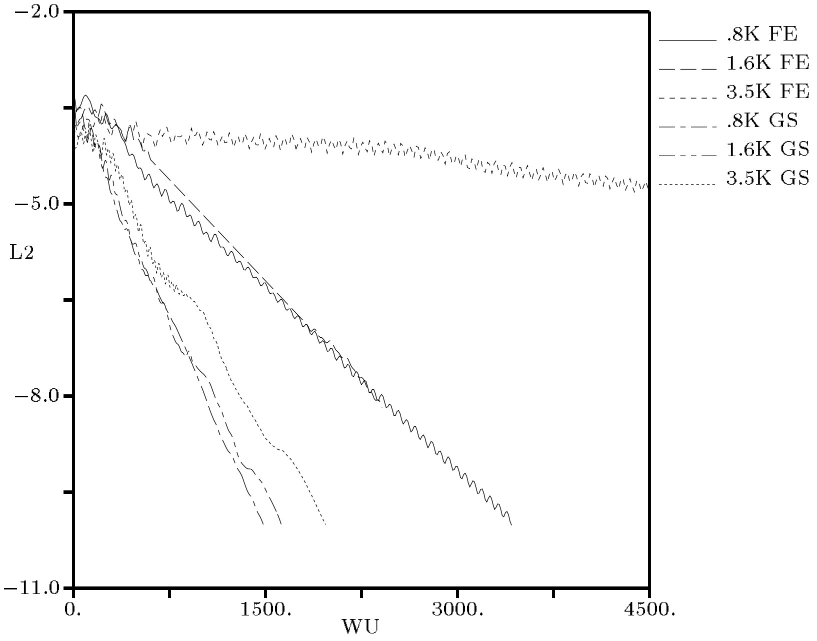


Figure 6.32: Grid dependence of the Multigrid convergence for the GAMM channel, grid sizes compared are 863, 1564 and 3498 nodes. Shown are Forward Euler ($\nu_a = 8$, $\nu_p = 1$) and Gauß-Seidel ($\nu_a = 1$, $\nu_p = 1$) convergence histories. The cell-CFL-number is taken as $\nu_c = 1$.

Chapter VII

Waters to Chart, Lands to Visit

Harmlose Schwestern der Heilsarmee
sind die Sirenen,
verglichen mit dieser Todesgrazie,
die ihre Opfer zerstückelt.

Cyrus Atabay, 'Die Erkenntnis'

[...]
Ich lachte kein klein wenig, gar nicht ma sœur,
Der ich spielend dunklem Schicksal entgegenging –
Während schon die Gesichter hinter mir
Langsam im Abend des blauen Walds verblaßten

Alles war schön an diesem einzigen Abend, ma sœur,
Nachher nie wieder, und nie zuvor —
Freilich: mir blieben nur mehr die großen Vögel
Die abends im dunklen Himmel Hunger haben."

Bertold Brecht, 'Ich habe Dich nie je so geliebt'

"Now I gotta get up
early in the morning."

Robert Palmer, 'Heavy Nova'

For two-dimensional isotropic triangulations, the novel Frontal Delaunay scheme leaves little to be desired (cf. section 5.8). An extension to viscous grids has been designed [115], but has not been as successful as the isotropic scheme documented here. Alternative approaches [116] using hybrid grids seem more promising.

The need for non-isotropic grids even in inviscid flow calculations and the fact that not all of the nice Delaunay properties extend from two dimensions let the Delaunay triangulation shine a lot less brightly in three dimensions. For one, the property that the discretization of a Laplacian guarantees a maximum principle on a Delaunay triangulation does not hold in three dimensions [47]. For another, the Delaunay triangulation in three dimensions allows the creation of a 'sliver', a very flat tetrahedron of four nearly equidistant and nearly coplanar points. Most proofs of grid quality in two dimensions, however, base the derivation of angular bounds on criteria of equidistance. Thirdly, the fact that the grid quality of frontal Delaunay methods only exhibits very few elements close to these bounds is based on the regularity of triangulations: a triangle does fill the plane in a regular fashion. A tetrahedron doesn't fill space regularly, and a look at the irregularity of three-dimensional tetrahedral grids demonstrates this strikingly. As a last remark, filling space with tetrahedra as opposed to hexahedra creates 4 times as many elements with the same number of vertices. Experiments in two dimensions suggest that the increased number of flux-calculations or residual-distributions does not lead to an increased accuracy of the solution, since the number of degrees of freedom remains the same [117].

Thus, the grid-generation in three dimensions for the coming years will be as hexahedral as possible. Only where hexahedral mesh generation, unstructured hexahedral approaches included, fail or are too cumbersome to use, prismatic and tetrahedral volume-meshing tools will be chosen. For this choice, Delaunay with it's robustness and speed will still outclass the Advancing Front method. Efforts in generating high-quality 3-D grids will have to be invested in hybrid grid matching, not in tuning Delaunay methods.

—

Multigrid convergence acceleration has been applied to the Euler equations using a Hyperbolic-Elliptic splitting and a Fluctuation-Splitting space discretization. Convergence is dramatically accelerated, but does not attain the rate of purely elliptic systems (cf. section 6.4).

Various efforts have yet to be undertaken in order to reap the benefits of the outstanding accuracy of the hyperbolic elliptic splitting. First and foremost, the temporal behavior of the coupling terms between the elliptic and the hyperbolic parts of the solution need to be better understood in order to obtain convergence results that are comparable with purely elliptic or purely hyperbolic problems. Monotonic compact schemes for elliptic problems with highly varying coefficients are needed that have good high-frequency smoothing properties and allow an optimal control of the dissipation as in scalar Upwind schemes. Some promising recent progress has been reported [118].

A proper solution to the singularity of the preconditioning in stagnation points has to be found to obtain a method that is as robust as dimensionally split Finite-Volume Upwind methods. A possible alley of research could be a formulation of the equations as presented by Ta'asan.

A more efficient time-stepping method for Cell-Vertex methods than the current lexicographic implementation of Gauß-Seidel has to be used, coloring schemes for unstructured grids such as red-black come to mind.

—

What has been presented here does not offer the solution to the convergence problems of compressible Euler and Navier-Stokes calculations. If the material shown has opened more questions than it has answered, if it is a brick in the road toward accurately simulating and understanding the flow in the artery and the thunderstorm, then this voyage will be have been a worthwhile undertaking.

Bibliography

[1] B. Van Leer. Towards the ultimate conservative difference scheme. III. Upstream centered Finite-Difference schemes for ideal compressible flow. *Journal of Computational Physics*, 23:263–275, 1977.

[2] S. Osher and F. Solomon. Upwind schemes for hyperbolic systems of conservation laws. *Mathematics and Computation*, 38, 1982.

[3] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43(2), 1981.

[4] S. K. Godunov. A Finite-Difference method for the numerical computation and discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47:271–306, 1959.

[5] S. F. Davis. A rotationally biased Upwind difference scheme for the Euler equations. *Journal of Computational Physics*, 56:65–92, 1984.

[6] D. Levy, K. G. Powell, and B. Van Leer. Implementation of a grid-independent Upwind scheme for the Euler equations. *AIAA-paper*, 1989. 9th Computational Fluid Dynamics Conference.

[7] A. Dadone and B. Grossmann. A rotated Upwind scheme for the Euler equations. *AIAA-paper 91-0635*, 1991.

[8] K. G. Powell, T. J. Barth, and I. H. Parpia. A solution scheme for the Euler equations based on a multi-dimensional wave model. *AIAA-paper 93-0065*, 1993.

[9] I. H. Parpia and D. J. Michalek. Grid-independent Upwind scheme for multidimensional flow. *AIAA-Journal*, 31(4):646–651, 1993.

[10] C. Rumsey, B. Van Leer, and P. L. Roe. A multidimensional flux function with applications to the Euler and Navier-Stokes equations. *Journal of Computational Physics*, 105:306–323, 1993.

[11] P. Van Ransbeeck and C. Hirsch. Multidimensional Upwind dissipation models for the 2-D Navier-Stokes equations. In *Proceedings of the 2nd European CFD Conference*, Stuttgart, Sept 1994. Wiley.

[12] D. Sidilkover. Multidimensional upwinding and multigrid. *AIAA-paper 95-1795*, 1995.

[13] H. Oksuzoglu. State vector splitting for the Euler equations of gasdynamics. *AIAA-paper 92-0326*, 1992.

[14] M. Fey and R. Jeltsch. A simple multidimensional Euler scheme. Technical Report 92-10, ETH Zürich, July 1992.

[15] R. Abgrall. Approximation of the multidimensional Riemann problem in compressible fluid mechanics by a Roe-type method. *SIAM Journal on Numerical Analysis*, 1994. submitted.

[16] T. J. R. Hughes. Recent progress in the development and understanding of SUPG methods with special reference to the compressible Euler and Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 7:1261–1275, 1987.

[17] C. Johnson. Finite-Element methods for compressible and incompressible flows. In *20th VKI Lecture Series on Computational Fluid Dynamics 1990-03* [120].

[18] P. L. Roe. Fluctuations and Signals - A Framework for Numerical Evolution Problems. *Numerical Methods for Fluid Dynamics, Academic Press, Inc*, 1982.

[19] G. Hauke and T. J. R. Hughes. A unified approach to compressible and incompressible flows. *Comput. Meth. Appl. Mech. Eng.*, 113:389–395, 1984.

[20] E. Oñate and S. R. Idelsohn. A comparison between Finite-Element and Finite-Volume methods in cfd. In C. Hirsch et. al., editor, *Computational Fluid Dynamics '92*, volume 1. Elsevier, 1992.

[21] H. Guillard. Mixed element volume methods in computational fluid dynamics. In *26th VKI Lecture Series on Computational Fluid Dynamics 95-02*, Rhode-St.-Genèse, Belgium, March 1995. Von Karman Institute for Fluid Dynamics.

[22] H. Deconinck, R. Struijs, and P. L. Roe. Fluctuation-Splitting for multidimensional convection problems: an alternative to Finite-Volume and Finite-Element methods. In *20th VKI Lecture Series on Computational Fluid Dynamics 1990-03* [120].

[23] H. Deconinck, R. Struijs, G. Bourgois, and P.L. Roe. High resolution shock capturing Cell-Vertex advection schemes on unstructured grids. In *25th VKI Lecture Series on Computational Fluid Dynamics 94-05* [121].

[24] R. Struijs. *A multi-dimensional Upwind discretization method for the Euler equations on unstructured grids.* PhD thesis, Technische Universiteit te Delft, 1994.

[25] T. J. R. Hughes, M. Mallet, Y. Taki, T. Tezduyar, and R. Zanutta. A one-dimensional shock capturing Finite-Element method and multi-dimensional generalizations. In *Workshop on Numerical Methods for the Compressible Euler Equations, INRIA, Roquencourt, December 1983*, Philadelphia, 1985. SIAM.

[26] B. Van Leer, W.-T. Lee, and P.L. Roe. Characteristic time-stepping or local preconditioning of the Euler equations. *AIAA-paper 91-1552*, 1991. 10th Computational Fluid Dynamics Conference 1991.

[27] R. H. Ni. A multiple-grid scheme for solving the Euler equations. *AIAA-Journal*, 20:1565–1571, 1981.

[28] M. G. Hall. Cell-Vertex multigrid schemes for solution of the Euler equations. In K. W. Morton and M. J. Banes, editors, *Numerical Methods for Fluid Dynamics II.* Oxford University Press, 1986.

[29] S. Ta'asan. Canonical forms of multidimensional inviscid flows. Technical Report 93-34, ICASE, 1993.

[30] P.L. Roe. Multidimensional upwinding – motivation and concepts. In *25th VKI Lecture Series on Computational Fluid Dynamics 94-05* [121].

[31] L.M. Mesaros and P.L. Roe. Multidimensional Fluctuation-Splitting schemes based on decomposition methods. *AIAA-paper 95-1699*, 1994.

[32] H. Paillère, H. Deconinck, and P.L. Roe. Towards the ultimate Upwind residual distribution scheme for the 2d steady compressible flow equations. *AIAA-paper 95-1700*, 1995.

[33] L. M. Mesaros. *A Multi-Dimensional Fluctuation-Splitting Scheme for the Euler Equations on Unstructured Grids.* PhD thesis, University of Michigan, 1995. to be defended.

[34] H. Paillère. *Multidimensional Upwind Residual Distribution Schemes for the Euler And Navier-Stokes Equations on Unstructured Grids.* PhD thesis, Université Libre de Bruxelles, June 1995.

[35] D. Darmofal and P. Schmidt. The importance of eigenvectors for local preconditioning of the Euler equations. *AIAA-paper 95-1655*, 1995.

[36] J. F. Thompson and N. P. Weatherill. Aspects of numerical grid generation: Current science and art. *AIAA invited paper*, August 9-11 1993. 11th AIAA Applied Aerodynamics Conference.

[37] M.A. Yerry and M.S. Shephard. Automatic three-dimensional mesh generation by the modified octree-technique. *International Journal for Numerical Methods in Engineering*, 20:1965–90, 1984.

[38] D. De Zeeuw and K. G. Powell. An adaptively-refined Cartesian mesh solver for the Euler equations. *Journal of Computational Physics*, 104(1):56–68, 1993.

[39] D. De Zeeuw. *A Quadtree-based adaptively-refined Cartesian-grid Algorithm for Solution of the Euler Equations.* PhD thesis, The University of Michigan, 1993.

[40] M. J. Berger and R. J. LeVeque. An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries. In *AIAA 9th Computational Fluid Dynamics Conference*, 1989.

[41] J. Quirk. *An Adaptive Grid Algorithm for Computational Shock Hydrodynamics.* PhD thesis, Cranfield Institute of Technology, 1991.

[42] J. Peraire, M. Vahdati, K. Morgan, and O. C. Zienkiewicz. Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, 72:449–466, 1987.

[43] R. Löhner and P. Parikh. Three-dimensional grid generation by the Advancing Front Method. *Int. J. Num. Meth. Fluids*, 8:1135–1149, 1988.

[44] G. L. Dirichlet. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. *Z. Reine Angew. Math.*, 40(3):209–227, 1850.

[45] G. Voronoi. Novelles applications des paramètres continus à la théorie des formes quadratiques. *J. Reine Angew. Math.*, 134, 1908.

[46] B. Delaunay. Sur la sphère vide. *Bull. Acad. Science USSR VII: Class. Sci. Mat.-Nat. 793-800*, 1934.

[47] T.J. Barth. Aspects of unstructured grids and Finite-Volume solvers for the Euler and Navier-Stokes equations. In *Special Course on Unstructured Grid Methods for Advection Dominated Flows* [119]. Report 787.

[48] D. F. Watson. Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes. *The Computer Journal*, 24(2):167–171, 1981.

[49] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

[50] D.G. Holmes and D.D. Snyder. The generation of unstructured triangular meshes using Delaunay triangulation. *Proceedings of the Second Conference on Grid Generation in Computational Fluid Dynamics, Pineridge Press, Swansea,*, 1988.

[51] P. Chew. Guaranteed-quality triangular meshes. Technical Report TR-89-98993, Cornell University, 1989.

[52] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. Techn. Rep. UCB/CSD 92/694, University of California at Berkeley, June 1992.

[53] E. van der Maarel. *A local Grid Refinement Method for the Euler Equations.* PhD thesis, Universiteit van Amsterdam, 1993.

[54] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics*, 72:277–298, 1987.

[55] B. Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23, 1977.

[56] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32, 1979.

[57] G. D. Raithby. Skew upstream differencing schemes for problems involving fluid flow. *Comp. Meth. Appl. Mech. Eng.*, 9:153–164, 1976.

[58] H. Deconinck, K. G. Powell, P.L. Roe, and R. Struijs. Multi-dimensional schemes for scalar advection. *AIAA-paper 91-1532*, May, 1991.

[59] H. Deconinck, R. Struijs, G. Bourgois, H. Paillère, and P.L. Roe. Multidimensional Upwind methods for unstructured grids. In *Special Course on Unstructured Grid Methods for Advection Dominated Flows* [119]. Report 787.

[60] P. L. Roe and D. Sidilkover. Optimum positive linear schemes for advection in two and three dimensions. *SIAM Journal on Numerical Analysis*, 1993.

[61] H. Paillère, H. Deconinck, P.L. Roe R. Struijs, L.M. Mesaros, and J.-D. Müller. Computations of inviscid compressible flows using Fluctuation-Splitting on triangular meshes. *AIAA-paper 93-3301*, 1993.

[62] J. F. Lynn. *Multigrid Solution of the Euler Equations With Local Preconditioning.* PhD thesis, University of Michigan, 1995.

[63] B. Van Leer, L. Mesaros, C. H. Tai, and E. Turkel. Local preconditioning in a stagnation point. *AIAA-paper 95-1654*, 1995.

[64] A. Jameson. Numerical solution of the Euler equations for compressible inviscid fluids. In F. Angrand, A. Dervieux, J. A. Désidéri, and R. Glowinski, editors, *Numerical Methods for the Euler Equations of Fluid Dynamics*. SIAM, 1985.

[65] B. Van Leer. Progress in multi-dimensional Upwind differencing. Technical Report 92-43, ICASE, 1992.

[66] S. P. Spekreijse. *Multigrid solution of the steady Euler equations*. PhD thesis, Technische Universiteit te Delft, 19897.

[67] S. P. Spekreijse. Multigrid solutions of monotone second order discretizations of hyperbolic conservation laws. *Math. Comp.*, 49:268–284, 1988.

[68] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.

[69] J.-D. Müller and P.L. Roe. Experiments on the accuracy of some advection schemes on unstructured and partly structured grids. In Mickle Vogt, editor, *23rd Annual Pittsburgh Conference on Modeling and Simulation*. University of Pittsburgh, Pittsburgh, PA, April 1992.

[70] H. Deconinck, R. Struijs, and P. L. Roe. A 2-D generalization of Roe's approximate Riemann solver. In *21st VKI Lecture Series on Computational Fluid Dynamics 1991-02*, Rhode-St.-Genèse, Belgium, March 1991. Von Karman Institute for Fluid Dynamics.

[71] J. Rice and R. Schnipke. A monotone streamline Upwind method for convection-dominated problems. *Comp. Meth. in Appl. Mech. and Eng*, 48:313–327, 1985.

[72] T. J. R. Hughes. A new finite element formulation for CFD II, beyond SUPG. *Comp. Meth. in Appl. Mech. and Eng.*, 84(3), 1986.

[73] David Sidilkover. *Numerical Solution to Steady-State Problems with Discontinuities*. PhD thesis, Weizmann Institute of Science, 1989.

[74] B. Perthame, Y. Qiu, and B. Stoufflet. Sur la convergence des schémas Fluctuation-Splitting pour l'advection et leur utilisation en dynamique des gaz. *C. R. Acad. Sc. Paris*, t.319, série I:283–288, 1994.

[75] D. Sidilkover and P. L. Roe. Unification of some advection schemes in two dimensions. Report 95-10, ICASE, 1995.

[76] A. Mizukami and T. J. R. Hughes. A Petrov-Galerkin Finite-Element method for convection dominated flows: an accurate Upwinding technique for satisfying the maximum principle. *Comp. Meth. Appl. Mech. and Eng.*, 50:181–194, 1985.

[77] K. W. Morton, M. A. Rudgyard, and G. J. Shaw. Upwind iteration methods for the Cell-Vertex scheme in one dimension. *Journal of Computational Physics*, 114:209–226, 1994.

[78] M. Rudgyard. Cell-Vertex methods for steady inviscid flow. In *24th VKI Lecture Series on Computational Fluid Dynamics 93-04*, Rhode-St.-Genèse, Belgium, March 1993. Von Karman Institute for Fluid Dynamics.

[79] M. Vanmaele, K. W. Morton, E. Süli, and A. Borzi. Analysis of the Cell-Vertex Finite Volume method for the Cauchy-Riemann equations. Technical Report 94/5, Oxford University Computing Laboratory, Numerical Analysis Group, Oxford, England, June 1994.

[80] H. Deconinck, P. L. Roe, and R. Struijs. A multi-dimensional generalization of Roe's flux difference splitter for the Euler equations. *Journal of Computers and Fluids*, 22(2/3):215–222, 1993.

[81] L. A. Catalano and H. Deconinck. Two-dimensional optimization of smoothing properties of multi-stage schemes applied to hyperbolic equations. In *Proceedings of the Third European Conference on Multigrid Methods*, 1990.

[82] A. Brandt. Multigrid techniques: 1984 guide. In *14th VKI Lecture Series on Computational Fluid Dynamics 1984-04*, Rhode-St.-Genèse, Belgium, March 12-16 1984. Von Karman Institute for Fluid Dynamics.

[83] A. Jameson. Solution of the Euler equations for two-dimensional transonic flow by a multigrid method. Technical Report MAE 1613, Princeton University, Department of Mechanical and Aerospace Engineering, 1983.

[84] B. Koren. *Multigrid and Defect correction for the Steady Navier-Stokes equations.* PhD thesis, Centrum voor Wiskunde en Informatica, Amsterdam, 1989.

[85] P. W. Hemker and B. Koren. Defect correction and nonlinear multigrid for the steady Euler equations. Technical Report Note NM-N8801, Centrum voor Wiskunde en Informatica, Amsterdam, 1988.

[86] D. J. Mavriplis. Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes. Technical Report 88-40, ICASE, 1988.

[87] D. Mavriplis. Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes. *AIAA-Journal*, 26(7):824–31, July 1988.

[88] D. J. Mavriplis. Three dimensional unstructured multigrid for the Euler equations. Technical Report 91-41, ICASE, 1991.

[89] B. Gustafsson and P. Lödtstedt. Analysis of the multigrid method applied to hyperbolic equations. In Hackbusch and Trottenberg [122], pages 253–263. Proceedings of the Third European Multigrid Conference.

[90] A. Brandt and I. Yavneh. Improved coarse-grid correction for high-Reynolds flows. In P.W. Hemker and P. Wesseling, editors, *Multigrid methods IV*, volume 116 of *International Series of Numerical Mathematics*, pages 1–23. Birkhäuser Verlag, Basel, 1994. Proceedings of the Fourth European Multigrid Conference, Amsterdam, July 6-9, 1993.

[91] E. Katzer. Multigrid methods for hyperbolic equations. In Hackbusch and Trottenberg [122], pages 253–263. Proceedings of the Third European Multigrid Conference.

[92] M. P. Leclercq and B. Stoufflet. Characteristic multigrid method: Application to solve the Euler equations with unstructured and unnested grids. Technical report, Avions Marcel Dassault, 1991.

[93] B. Koren and P. W. Hemker. Damped, direction-dependent multigrid for hypersonic flow copmutations. *Appl. Num. Math.*, 1990.

[94] Y. P. Marx. Multigrid solution of the advection-diffusion equation with variable coefficients. *Comm. in Appl. Num. Meth.*, 8:633–650, 1992.

[95] W. L. Briggs. *A Multigrid Tutorial.* Society for Industrial and Applied Mathematics, Philadelphia, Pennsylavania, 1987.

[96] D.T. Lee and B.J. Schachter. Two algorithms for constructing a Delaunay triangulation. *Int. Journal of Comp. and Inform. Sciences*, 9(3):219–41, 1980.

[97] S. Rippa. Minimal roughness property of the Delaunay triangulation. *CAGD*, 7(6):489–479, 1990.

[98] S. Rippa. Long and thin triangles can be good for linear interpolation. *SIAM J. Num. Anal.*, 29:257–270, 1992.

[99] I. Babuška and A.K. Aziz. On the angle condition in the Finite-Element method. *SIAM J. Num. Anal.*, 13(2):214–26, 1976.

[100] P.L. Roe. Error estimates for Cell-Vertex solutions of the compressible Euler equations. Technical Report 87-6, ICASE, 1987.

[101] P.L. Roe. The influence of mesh quality on solution accuracy. In *19th VKI Lecture Series on Computational Fluid Dynamics 1989-04*, Rhode-St.-Genèse, Belgium, March 1989. Von Karman Institute for Fluid Dynamics.

[102] S. Rebay. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. *Journal of Computational Physics*, 106(1):125–38, 1993.

[103] P. J. Green and R. Sibson. Computing the Dirichlet tesselation in the plane. *The Comp. J.*, 21(2):168–173, 1977.

[104] C. L. Lawson. Properties of n-dimensional triangulations. *CAGD*, 3:231–246, 1986.

[105] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, 1985.

[106] K.-P. Beier. Computer-aided hull design and production. Informal Lecture Notes, Dept. of Nav. Arch. and Marine Eng., The University of Michigan, 1995.

[107] M. Tanemura, T. Ogawa, and N. Ogita. A new algorithm for three-dimensional Voronoi tesselation. *Journal of Computational Physics*, 51:191–207, 1983.

[108] D.E. Knuth. *The Art of Computer Programming - Sorting and Searching*, volume 3. Addison Wesley, Reading, MA, 1973.

[109] M.H. Lallemand, H. Steve, and A. Dervieux. Unstructured multigridding by volume agglomeration: Current status. *Computers and Fluids*, 21:397–433, 1992.

[110] H. Guillard. Node nested multigrid with Delaunay coarsening. Technical Report 1898, INRIA, 1993.

[111] R. Richter. *Schémas de capture de discontinuités en maillage non-structuré avec adaptation dynamique: applications aux écoulements de l'aérodynamique.* PhD thesis, Ecole Polytechnique Fédérale, Lausanne, Switzerland, 1993.

[112] A. Rizzi and H. Viviand, editors. *Numerical methods for the computation of inviscid transonic flows with shock waves*, volume 3 of *Notes on Numerical Fluid Mechanics*, Braunschweig, 1981. Vieweg.

[113] S. Ta'asan. Canonical-variables multigrid method for steady state euler equations. Technical Report 94-14, ICASE, 1994.

[114] P. L. Roe. Discrete models for the numerical analysis of time-dependent multidimensional gas dynamics. *Journal of Computational Physics*, 63, 1986.

[115] J.-D. Müller. Proven angular bounds and stretched triangulations with the Frontal Delaunay method. *AIAA-paper 93-3347*, 1993.

[116] D. Banks, J.-D. Müller, P. Vankeirsbilck, and H. Deconinck. An object-oriented approach to hybrid structured/unstructured grid generation. *AIAA-paper 96-0032*, 1996.

[117] M. Aftosmis, D. Gaitonde, and T. Sean Tavares. On the accuracy, stability and monotonicity of various reconstruction algorithms for unstructured meshes. *AIAA-paper 94-0415*, 1994.

[118] E. van der Weide and H. Deconinck. A new N-scheme for systems of equations. Private communication, Sept. 1995.

[119] AGARD. *Special Course on Unstructured Grid Methods for Advection Dominated Flows*, Neuilly Sur Seine, France, May 1992. Report 787.

[120] Von Karman Institute for Fluid Dynamics. *20th VKI Lecture Series on Computational Fluid Dynamics 1990-03*, Rhode-St.-Genèse, Belgium, March 5-9 1990.

[121] Von Karman Institute for Fluid Dynamics. *25th VKI Lecture Series on Computational Fluid Dynamics 94-05*, Rhode-St.-Genèse, Belgium, March 1994.

[122] W. Hackbusch and U. Trottenberg, editors. *Multigrid Methods III*, volume 98 of *International Series of Numerical Mathematics*, Basel, 1991. Birkhäuser Verlag. Proceedings of the Third European Multigrid Conference.

Errata

“Someone had blundered.”

V. Woolf, ‘To the Lighthouse’

Ancestors of the Hyperbolic-Elliptic splitting

The work by El Dabaghi et. al. should have been mentioned. See e.g.: F. El Dabaghi, O. Pironneau, J. Periaux, G. Poirier, “2-D/3-D Finite-Element Solution of the Steady Euler Equations for Transonic Lifting Flow by Stream Vector Correction”, Int. J. Num. Meth. Fluids, Vol. 7, pp. 1191-1209, 1987.

Also the very similar work of Hughes et. al. in [25], although not further pursued by the authors, should have been linked more clearly to the splitting used here. (March, 1996)

Accuracy of the PSI-scheme

In section 3.2.5 on page 30 it is claimed that the PSI-scheme loses accuracy when it reverts to the N-scheme in the two-target case. When looking at the PSI-scheme from the view of limiters, one can actually show that the distribution coefficients remain bounded in this case. Thus, the N-scheme is linearity preserving for the one-target case and for the two-target case when the gradient vector intersects the outflow side. And consequently no loss of accuarcy occurs for the PSI-scheme in this case. (March, 1996)

Consistent gradient

Equation 3.82 should have $\nabla \mathbf{Z}$ in the second integral rather than $\nabla \mathbf{U}$.

$$\nabla \bar{\mathbf{U}} = \frac{1}{S_T} \iint_T \nabla \mathbf{U} dA = \frac{1}{S_T} \iint_T \frac{\partial \mathbf{U}}{\partial \mathbf{Z}} \nabla \mathbf{Z} dA = \left. \frac{\partial \mathbf{U}}{\partial \mathbf{Z}} \right|_{\bar{\mathbf{Z}}} \nabla \mathbf{Z}$$

(March, 1996)

Abstract

ON TRIANGLES AND FLOW

by
Jens-Dominik Müller

Chairman: Phillip L. Roe

An accurate and efficient solution to the two-dimensional Euler equations on complex domains is sought.

For this purpose, a novel unstructured grid generation method based on the Delaunay triangulation is presented that combines frontal vertex-generation methods with the concept of the Delaunay triangulation. The method works on minimal user input such as boundary vertices only and generates its own background grid for spacing interpolation. Upper and lower angular bounds are estimated and the grid quality is analyzed statistically. A novel eway of obtaining within the framework of frontal grid generation methods sets of coarsened multilevel grids that are vertex-nested is described and demonstrated.

Convergence of the scheme using Multigrid with Forward-Euler and Gauß-Seidel timestepping is examined for scalar advection equations and the Cauchy-Riemann equations, as well as Euler calculations for subsonic and transonic bump cases.

The Euler equations are discretized in their conservative form using Fluctuation-Splitting schemes and a "hyperbolic-elliptic" decomposition that splits the equations in the subsonic case into two decoupled advection equations for entropy and enthalpy and a coupled 2×2 system similar to the Cauchy-Riemann equations. In the supersonic case the equations become a fully decoupled set of scalar advection equations.

It is found that when considered separately, Forward-Euler as well as Gauß-Seidel are suitable time-stepping schemes for hyperbolic and elliptic problems when used in conjunction with Multigrid. Although the final convergence of purely hyperbolic equations does not benefit from Multigrid, initial convergence to a useful estimate after few cycles does. The efficiency found on the purely hyperbolic and elliptic problems is, however, not recovered with for the full set of Euler equations, but a significant acceleration of convergence compared to the single-grid scheme is found.