Algorithm Developments for an Unstructured Viscous Flow Solver

Pierre Moinier
St Hugh's College

Thesis submitted for the degree of Doctor of Philosophy
at the University of Oxford

Trinity Term 1999

Dedicated to my parents

Andrée Moinier and André Moinier

Oxford, December 1999

Acknowledgements

During the past 4 years, many people have contributed to the progress of this work and I would like now to take this opportunity to thank them all.

I have had the immense chance and great pleasure of working with Mike Giles who has been a constant source of good ideas, patient guidance and a teacher. I am grateful to him for all his hard work on my behalf as also to his kindness for having always provided me the material that I needed and the freedom of travelling.

A special thought would also be for Paul Crumpton who in my first year had the difficult task to introduce me to CFD and to Cindy whereas my level of comprehension was limited to a basic set of English words...and to the making of paper roses...

I would like to thank Nick Birkett for having been (and still being) a great help, not only to solve my Linux problems, but also when I was walking on the edges of a stability limit that required some linear algebra that I had forgotten, and to Leigh Lapworth for his help, particularly during the debugging phase by providing me grids and good advice to interpret the physics of internal flows.

During these years, I have enjoyed the company of many fellow researchers, including Gun, Niles, Mike, Paul, Jens, Mihai, Catherine, Ercilia and Alvaro. Thanks to all of you and also to Bette and Shirley who have made my time at the lab so convivial and fun.

Finally, I am incredibly fortunate to have shared these years with so many wonderful friends. I wish I could write a few words to each of you, without affecting the length of this thesis. I am sure you will recognise yourself here, and let me then thank you all personally... as appropriate!

I am grateful to Professor Eatock Taylor and Dr. Peiró for agreeing to be my examiners and for their comments regarding this dissertation, and finally I would like to acknowledge the funding of the European Community which enabled me to undertake this work.

Abstract

Algorithm Developments for an

Unstructured Viscous Flow Solver

Pierre Moinier — Doctor of Philosophy

St Hugh's College — Trinity Term 1999

An efficient preconditioned multigrid method is developed for both inviscid and viscous flow applications on unstructured hybrid grids. The work builds upon recent breakthroughs in convergence acceleration on structured grids using preconditioning and multigrid. It is motivated by the results obtained with standard multigrid methods with a scalar time step which performs well for Euler calculations but is far less effective for turbulent Navier-Stokes calculations due to the highly stretched cells needed to resolve high Reynolds number boundary layers.

The new scheme provides rapid and robust convergence, and yields computational savings of roughly a factor of three compared to the standard method for a wide range of 2D and 3D inviscid and viscous cases (airfoils, wings, airplane and internal flows). The good performance of the numerical method is explained by analysis of the stability limits of a first order upwind discretisation of the Euler equations, which is a close approximation to the viscous discretisation on the coarser multigrid levels. In the analysis, the preconditioned system is shown to have a field of values which remains inside the stability region of the Runge-Kutta scheme, thereby guaranteeing algebraic stability.

For improved accuracy and convergence for low Mach number applications, the scheme is modified through the additional use of a low Mach number preconditioner. The conclusions of the numerical analysis remain unchanged provided the low Mach number preconditioner is symmetric and positive definite. Analysis of the preconditioned Euler equations also shows the very significant effect of the preconditioning on the effectiveness of boundary conditions in eliminating initial transients.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

A_x, A_y, A_z	inviscid flux Jacobians
$\|A_x\|, \|A_y\|, \|A_z\|$	modulus of A defined by $T\|\Lambda\|T^{-1}$
c	speed of sound
F	discrete flux
i, j, k	mesh indices
M	Mach number
M_∞	free stream Mach number
n	time step index
P	preconditioner
R	spatial residual operator
S_x, S_y, S_z	projection of the surface through which the flux is evaluated in the x, y, z directions
T, T^{-1}	right and left eigenvector matrices of A
t	time coordinate
Δt	local time step
u, v, w	cartesian velocity components
Q	conserved variable vector for Euler and Navier–Stokes equations
x, y, z	cartesian coordinates
$\Delta x, \Delta y, \Delta z$	mesh spacings in Cartesian coordinates
z	variable in the complex plane
V	measure of the control volume
p	pressure
$L()$	pseudo-Lsplacian operator

Greek Symbols

γ	ratio of specific heats
Λ	diagonal eigenvalue matrix of A
$\lvert\Lambda\rvert$	diagonal matrix containing the moduli of the entries in Λ
λ^k	eigenvalue corresponding to the k-th characteristic
μ	molecular viscosity
μ_t	turbulent viscosity
ρ	density
$\tilde{\nu}$	turbulent working variable
ν	kinematic viscosity
$\rho(A)$	spectral radius of A
Ω	control volume
Γ	low Mach number preconditioner

Calligraphic Symbols

$\mathcal{F}$	analytical flux
$\mathcal{R}$	analytical residual operator
$\mathcal{Q}$	analytical conserved variable vector

Miscellaneous Symbols

$\tilde{}$	quantity written using symmetrised variables
$\prime$	perturbation quantity

Chapter 1

Introduction

In the last two decades, a large variety of computational fluid dynamics methods has been developed to solve the compressible Euler and Navier-Stokes equations used in aeronautical applications. Progress in algorithm development (including multigrid methods) and parallel computing techniques have made the prediction of inviscid flow around complex geometries a rapid and robust procedure. Numerical modelling of transonic flow over a complete aircraft using the Euler equations is now reliable and efficient so that the need for further development does not appear overwhelming. However, for an accurate aerodynamic analysis, viscous effects must be considered. To capture these viscous effects, the Reynolds-averaged Navier-Stokes equations are solved with some turbulence model, and transition must also be modelled. The problem is that at high Reynolds numbers typical of aeronautical applications such flows are physically and numerically stiff, and a dramatic degradation of the iterative convergence is observed. The present work focuses entirely on this aspect, and is a continuation of recent breakthroughs in that perspective. The reason why viscous calculations are problematic is the highly stretched computational cells that are needed to efficiently resolve a high Reynolds number boundary layer. These very high aspect ratio cells are required to accurately represent the steep gradients across the boundary layer but increase considerably the size of the problem to solve, in term of storage and computational cost. With the development of computers, the size of these problems will become even larger, therefore, it is important that the

convergence rate of the iterative method used should be grid-independent. To overcome these drawbacks, the solution procedure most commonly adopted is multigrid. The theory of multigrid is very well developed for the case of elliptic problems, and is based on an updating scheme acting as a smoothing operator on each grid level. Roughly speaking, a sequence of successively coarser grids that can represent the smooth error modes of the finer grid is required, as well as some iterative procedure which eliminates the high frequency error modes on each grid. Although the theory is not well developed for hyperbolic p.d.e's, nevertheless excellent convergence rates are achieved for the Euler equations. Therefore, multigrid is the most attractive approach for Navier-Stokes calculations as well.

1.1 The numerical solution of the Navier-Stokes equations

In the present work, attention is focused on developing efficient numerical methods in the context of multigrid for steady viscous flows from nearly incompressible to transonic and supersonic speeds. The steady state is achieved by eliminating the transient behaviour either by damping or by expulsion from the computational domain [60]. The damping is essentially a local process, whereas the propagation is a global one. Consequently, it is the damping properties of the relaxation scheme that are the most critical for insuring insensitivity to problem size. To drive the multigrid algorithm, explicit or implicit relaxation schemes can be used. Explicit schemes, limited by a CFL condition, offer a cheap computational cost, low storage requirements and good parallel capabilities. Alternatively, implicit schemes, theoretically unconditionally stable, require a high operation count, much more memory and are more difficult to parallelise. In the current approach, an explicit scheme is retained. The semi-discrete scheme proposed by Jameson *et al* [72] uses multi-stage Runge-Kutta time-stepping with coefficients chosen to promote rapid damping and propagation of error modes, by ensuring that the amplification factor is small in the region where the eigenvalues corresponding to high frequency modes

are concentrated. Such an algorithm has proven to be highly successful for inviscid calculations, and has shown good multigrid performance. However, when dealing with the Navier-Stokes equations, the multigrid performance is not nearly so good.

In his thesis, Pierce [63] has given a very complete diagnosis of Multigrid Breakdown. The three fundamental causes of poor multigrid performance are (a) stiffness in the discrete system, due to the disparity in the propagative speed of convective and acoustic waves, (b) decoupling of modes which happens when the flow is aligned with the grid causing the convective error mode which is saw-tooth in the cross-flow direction but smooth in the flow direction not to be eliminated on the fine grid, and (c) the highly stretched cells inside the boundary layer. The first two manifest themselves in a identical manner by causing the corresponding eigenvalues of the discrete residual operator to fall near the origin in the complex plane so that they can not be damped efficiently by the multi-stage Runge-Kutta scheme. They are the cause of degraded multigrid performance for inviscid calculations, and persist for Navier-Stokes calculations. The highly stretched cells are a far more serious problem for viscous calculations and lead to acute 'numerical stiffness' problems. This numerical stiffness is related to the fact that the timescale for viscous diffusion across a high Reynolds number boundary layer is much greater than the timescale for the propagation of a pressure wave across the boundary layer. Using explicit solvers restricted by the acoustic timescale, this leads to very slow convergence for the convection/diffusion of streamwise momentum and temperature. There is also numerical stiffness directly related to the source terms in the turbulence model; these will be treated by a point implicit method in order to achieve a satisfactory level of robustness.

To cope with the problem of highly stretched cells, different methods have been proposed, including a semi-coarsening multigrid strategy and the use of a preconditioner. All these methods aim to produce the same effect, which is to damp as efficiently as possible all the error modes. The idea of the semi-coarsening, suggested by Mulder [57], is not to coarsen the mesh in every direction simultaneously, so that each level of the sequence of grids used in the multigrid strategy involves several

grids which can cope separately with each mode.

Alternatively, Allmaras [2] suggests the use of a preconditioner, depending on the multigrid strategy used. For example, he recommended the block-Jacobi matrix preconditioner proposed by Morano et al. [55] with a semi-coarsened strategy. The effect of the preconditioner is to move the eigenvalues away from the origin of the Fourier complex plane providing, within an optimised Runge-Kutta update, a very good damping of the high-frequency error modes.

Recently, Pierce and Giles [65] have analysed different combinations of preconditioner and multigrid method for both inviscid and viscous flow applications. For turbulent Navier-Stokes calculations, a block-Jacobi preconditioner and a semi-coarsening multigrid method provides an effective damping of all modes inside the boundary layer, both in theory and in practice. The preconditioner damps all the convective modes, while the multigrid strategy, in which the grids are coarsened only along the normal to the boundary layer, ensures that all acoustic modes are eliminated efficiently. Thus, they have demonstrated that considerable speed-up can be achieved when using stretched structured meshes.

In this work the same idea is followed, but for unstructured grids. The preconditioner is implemented in a multigrid solver which has proven to be highly successful for inviscid meshes [17, 18], but is modified to treat the highly stretched cells required for high Reynolds number flows [20] so that the equivalent of a semi-coarsening strategy is employed.

1.2 Unstructured grids

Navier Stokes flow solvers on structured grids have been developed to a point where complex flows can be accurately modeled. [83, 40]. However, the required grid for an aircraft configuration is difficult and time consuming to generate; multiple blocks are required to allow for both the geometrical complexity and the disparate length scales of the flowfield. Thus, the use of unstructured mesh techniques has become more popular because of the added flexibility they offer in dealing with complex

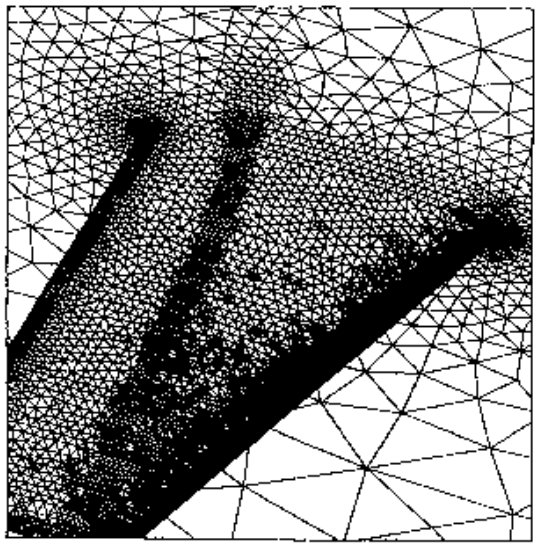


Figure 1.1 Refined grid for Onera M6 wing

geometries, and enabling grid adaptation where extra grid points can simply be added where they are needed (see figure 1.1).

However, a major drawback of such techniques remains their lower efficiency and increased computational cost as compared to structured mesh techniques. This lower computational efficiency is due to several factors; one is that it is much harder to formulate higher order discretisations, and so instead one may need to use more grid points than for a structured grid calculation. Another is the computational cost (and memory requirements) of indirect addressing. A third factor is an increased cost per grid point due to the use of simplex grids (triangles in two dimensions, tetrahedra in three dimension). The additional cost incurred by the use of tetrahedral meshes can be demonstrated by considering a structured hexahedral mesh of N vertices. For an edge-based finite volume scheme, there are N unknowns, and $3N$ fluxes which must be evaluated (effects of the boundaries are neglected). If this hexahedral mesh is now subdivided into a tetrahedral mesh, the equivalent finite-volume scheme consists of N unknowns, as previously, but the evaluation of $7N$ fluxes is now required to construct the discretisation (again neglecting boundary effects). Thus, a tetrahedral mesh discretisation is roughly twice as expensive to

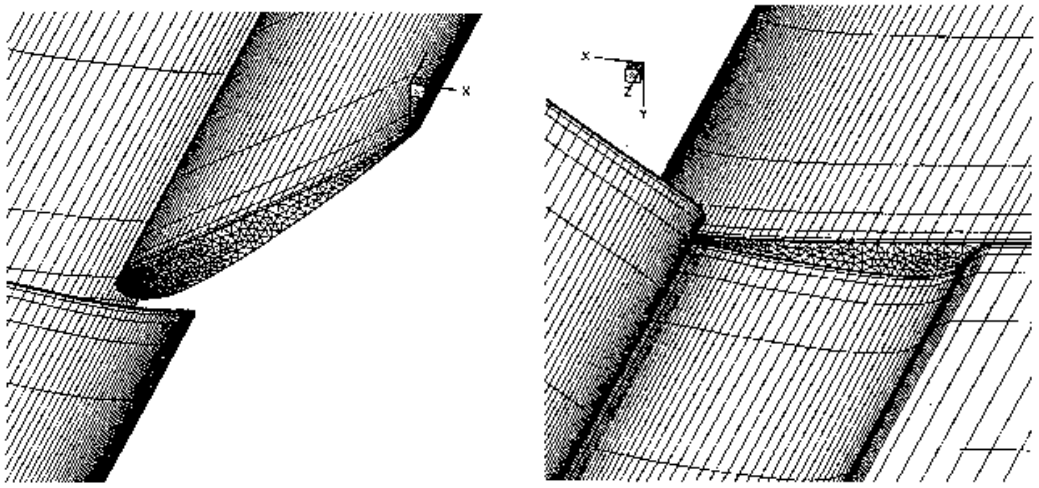


Figure 1.2 Closeup views of the flap edge and the cove for a wing-flap configuration

evaluate as a hexahedral mesh discretisation. One possible solution to reduce the cost of the discretisation is to switch to other types of elements in region of high grid stretching, where their number is very dense. The idea is then to use quadrilaterals out of a pairs of triangles in two dimensions, or prisms, pyramids and hexahedra out of groups of tetrahedra in three dimensions. The use of mixed elements, or hybrid grids, in a unstructured mesh technique offers to some extent the best compromise between mesh quality, efficiency and flexibility. Figure 1.2 show the closeup views of the surface grid near the cove and the flap edge for a wing-flap configuration, where a blend of hexahedra and tetrahedrals are used. Typically these meshes are composed of prismatic elements close to the surface of the geometry being modeled and tetrahedral elements in the far fields. The prisms provide the option to use sufficient grid-clustering in the normal direction as well as flexibility in geometric modelling by using unstructured tesselation, whereas the tetrahedrals are used to fill the outer inviscid region with a gradual transition in grid sizes at the grid interface between prism and tetrahedra [85, 39]. The hybrid approach has already been advocated by several authors[39, 5, 11], and constitutes the framework in which this work is done.

1.3 Low Mach regime capability

Another aspect of this work concerns the capability to solve low Mach number flows. For small Mach number, it can be shown that the incompressible equations approximate the compressible equations. But there are many problems, particularly in turbomachinery, where the flow can vary from low subsonic to supersonic. Also in low speed aerodynamics at high angle of attack most of the flow has a low Mach number, but there are localised regions containing shocks. Consequently, it is appropriate to use the compressible equations even where the Mach number of the flow is small. However, the observed convergence rate gets substantially slower and the solution produced is usually of poor quality, with pressure oscillations visible in contour plots. The slowdown is due to some analytic stiffness arising from the inherent propagative disparities in the limit of vanishing Mach number, where the ratio of the convective speed to the acoustic speed approaches zero. This type of stiffness is often treated using preconditioning techniques [82, 86, 77]. By altering the acoustic speeds of the system such that all eigenvalues become of the same order this difficulty is completely alleviated. In addition, the solution can also be improved by changing the artificial dissipation in the spatial discretisation. Based on the preconditioned system, the relative scaling of different numerical smoothing terms can be improved, and the steady-state solution becomes more accurate. Unfortunately, these benefits are achieved with difficulty because local preconditioners designed for low Mach number performance have poor robustness at stagnation points. Darmofal and Schmid have shown that this lack of robustness is due to unlimited transient amplification of perturbations resulting from a degeneration of the structure of the eigenvectors of the preconditioned equations [23]. These becomes highly non-orthogonal as $M \to 0$. The most common technique to avoid this robustness problem is based on limiting the effect of preconditioning below a multiple of the freestream Mach number. This multiple is typically greater than one [66], and destroys the locality of the preconditioning, since the limit becomes more global. Furthermore, there are problems where a reference Mach number is inappropriate

or non-existent and where this type of limiting is difficult to realize. Examples of these type of flow would be a hypersonic flow around a blunt body (which would contain regions of subsonic flow) or flows in pumps and turbomachinery. A possible way to address this problem is to base this limit on the local Mach number, or on strict local information like the pressure [24].

For non-linear calculations where the local mean flow is altered by the perturbations, the poor eigenvector conditioning can lead to significant transient growth and therefore slow down convergence as transient effects are continuously stimulated by incoming waves. The van Leer and Turkel preconditioners both suffer from significant initial growth rate whereas the block Jacobi preconditioner does not. Although the block Jacobi preconditioner is the best approach to avoid transient effects, it does not accelerate the long wavelength modes as $M \rightarrow 0$ and does not improve the solution quality for nearly incompressible flows since the characteristic speeds are not effectively equalised. However, this stiffness can be fully addressed by integrating a low Mach number preconditioner in the dissipation and hence in the block Jacobi preconditioner. This approach, which is the one followed in this work, is attractive because it does not require any change of variables in the current code.

1.4 Stability analysis

The preconditioned system thus defined and integrated in an appropriate multigrid strategy shows good multigrid performance for any Mach number. Analytic expressions for the preconditioner Fourier Footprints inside an asymptotically stretched boundary layer cell reveal why the preconditioner damps all convective modes and leads to substantial improvements [63]. But there is now a need for the supporting numerical analysis to investigate and give local timestep stability limits. Considering linear differential equations with constant coefficients on structured grids, the Von Neumann method for stability analysis is generally applied. The central idea is that such equations have particular solutions based on a Fourier expansion and to evaluate the conditions for which the amplitude of any error harmonic does not grow in

time. Using unstructured grids and variable coefficients, this approach is not appropriate, and consequently another method is applied. There are two well-documented stability analysis method which can then be used. One involves consideration of the eigenvalues of the matrix representing the discretisation of the spatial differential operator. For a lot of cases, this leads to sufficient conditions for asymptotic stability, but there are well-known examples such as the first order upwinding of the convection equation on a finite 1D domain (e.g., [67], [42], [80]) for which this is not a practical stability criterion because it allows an unacceptably large transient growth before the eventual exponential decay. The other one, which is the one that is used in this work, is the energy method [68] which relies on the construction of a suitable defined "energy" which can be proven to monotonically decrease. In his paper [28], Giles analyses the semi-discrete and fully discrete Navier-Stokes equations arising from a Galerkin discretisation on a tetrahedral grid, and presents two bounding sets for the field of values arising from this discretisation. From these he obtains sufficient time-step stability limits for both time accurate and local timestep computations.

This work also examines and evaluates the limits of the stability region of the scheme, but since the discretisation uses an edge-based data structure, which at the moment is the most common approach [3, 61, 7, 54, 51], the same stability analysis can not be performed. Within the edge-based discretisation, the treatment of the viscous terms appears to be too difficult, and thus, only the Euler equations are considered in a first order upwinding scheme. Although the Navier-Stokes equations are solved, a purely first order numerical dissipation is used on all coarser meshes of the multigrid, and on these coarser levels the viscous terms are not significant. Thus, the conclusions that are drawn from the inviscid stability analysis remain pertinent enough to explain the good behaviour of the Navier-Stokes calculations.

1.5 Summary

The main body of the dissertation is divided into five chapters corresponding to the development of a basic Navier Stokes solver to incorporate preconditioning and a low Mach number capability, to end up with an algorithm capable of dealing with a complete range of Mach numbers for any internal and external flows.

Chapter Two begins with the spatial discretisation followed by the time-stepping scheme, and the edge collapsing multigrid method. Specific attention is focussed on the treatment of the highly stretched cells required for turbulent calculations, in both the discretisation and the edge-collapsing multigrid method. Finally, the turbulence model is introduced, and its treatment described.

To cluster the residual eigenvalues in regions where the multi-stage Runge-Kutta has a rapid damping and propagating effect, the block Jacobi preconditioner is used. Chapter Three describes how to evaluate it, and how to adjust it in order to include a slip boundary condition. Components of the matrix are also given.

The following chapter presents a stability analysis of the resulting method and explains the reasons why good multigrid performance is observed. After first considering applications with periodic boundary conditions, the influence of a slip boundary condition is investigated.

Chapter Five describes the new discretisation that the introduction of a low Mach number preconditioner implies and also investigates the influence of this preconditioner on the boundary conditions. When the solution has almost converged to the steady state, so that only low frequency waves remain, the analysis determines whether an exponential decay of the amplitude of these waves can be expected. Finally, the extension of the stability analysis of the previous chapter is presented, giving a complete study of the method used throughout this work.

All the results in two and three dimensions are gathered in Chapter Six. Going from inviscid to viscous, a complete set of test cases exemplify the resulting method on grids of various complexities.

A concluding chapter summarises the main results of the research and provides

some suggestions for future work.

Chapter 2

Discretisation

The objective of this work is the prediction of steady compressible turbulent flows over complex geometries using hybrid grids. The starting point was a turbulent Navier-Stokes algorithm on tetrahedral grids developed with Dr. Paul Crumpton [20]. This used a fully automatic edge-collapsing multigrid method in a framework allowing parallel computation [10] and included the Spalart-Allmaras turbulence model [76]. Moving to hybrid grids for the reasons outlined in the Introduction required the development by Dr. Jens-Dominic Müller of a modified grid-collapsing strategy [59]. For the flow discretisation there are also some changes due to the use of hybrid grids, but the key features are unchanged. In particular two important aspects of the discretisation remain valid: first the introduction of an anisotropic linear preserving Laplacian and then the addition of some edge derivative terms in the evaluation of the gradient. Both increase the robustness of the algorithm.

This chapter is organised as follows: firstly, the spatial discretisation is described followed by the time-stepping scheme, and then the element-collapsing multigrid method. Specific attention is focussed on the treatment of the highly stretched cells required for turbulent calculations, in both the discretisation and the element-collapsing multigrid method. Finally, the turbulence model is introduced, and its treatment described.

2.1 Governing equations

In Cartesian coordinates (x, y, z), the Navier-Stokes equations are non-dimensionalised using the variables

$$x^* = \frac{x}{L_{ref}}, \quad y^* = \frac{y}{L_{ref}}, \quad z^* = \frac{z}{L_{ref}}, \quad t^* = \frac{t\ U_{ref}}{L},$$

$$u^* = \frac{u}{U_{ref}}, \quad v^* = \frac{v}{U_{ref}}, \quad w^* = \frac{w}{U_{ref}}, \quad \mu^* = \frac{\mu}{\rho_{ref} U_{ref} L_{ref}},$$

$$\rho^* = \frac{\rho}{\rho_{ref}}, \quad p^* = \frac{p}{p_{ref}}, \quad T^* = \frac{T}{T_{ref}},$$

where the reference length scale L_{ref} is 1 metre and the reference density ρ_{ref}, pressure p_{ref} and temperature T_{ref} correspond to atmospheric standard temperature and pressure at sea level. A reference velocity U_{ref} is then defined by $U_{ref} \equiv \sqrt{\frac{p_{ref}}{\rho_{ref}}}$. Omiting the asterisks for clarity, the three-dimensional Navier-Stokes equations then take the form

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial \mathcal{F}_x}{\partial x} + \frac{\partial \mathcal{F}_y}{\partial y} + \frac{\partial \mathcal{F}_z}{\partial z} = 0 \tag{2.1}$$

where

$$\mathcal{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad \mathcal{F}_x = \begin{pmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{yx} \\ \rho uw - \tau_{zx} \\ (\rho E + p)u - u\tau_{xx} - v\tau_{yx} - w\tau_{zx} + q_x \end{pmatrix}$$

$$\mathcal{F}_y = \begin{pmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ \rho vw - \tau_{zy} \\ (\rho E + p)v - u\tau_{xy} - v\tau_{yy} - w\tau_{zy} + q_y \end{pmatrix}$$

$$\mathcal{F}_z = \begin{pmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho w^2 + p - \tau_{zz} \\ (\rho E + p)w - u\tau_{xz} - v\tau_{yz} - w\tau_{zz} + q_z \end{pmatrix}$$

ρ, u, v, w, p, E denote the density, the three Cartesian velocity components, the pressure, and the total internal energy, respectively. To complete the system of equations requires an equation of state for an ideal gas,

$$p = \rho RT = (\gamma - 1)\rho \left(E - \frac{1}{2} \left(u^2 + v^2 + w^2 \right) \right),$$

in which R, T, γ are thr gas constant, temperature and uniform specific heat ratio, respectively, as well as equations defining the heat fluxes,

$$q_x = -k\frac{\partial T}{\partial x}, \qquad q_y = -k\frac{\partial T}{\partial y}, \qquad q_z = -k\frac{\partial T}{\partial z},$$

with the coefficient of thermal conductivity $k = \frac{\gamma\mu}{Pr}$ and Pr the Prandtl number ($Pr = 0.72$ for air). The deviatoric stress are given by:

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{yy} = 2\mu\frac{\partial v}{\partial y} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{zz} = 2\mu\frac{\partial w}{\partial z} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}\right)$$

$$\tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

$$\tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)$$

$$\tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)$$

where the molecular viscosity μ is modeled by Sutherland's law,

$$\mu = \frac{1.461 \times 10^{-6} T^{3/2}}{T + 110.3},$$

and the bulk viscosity λ is defined by invoking Stokes' hypothesis

$$\lambda = -\frac{2}{3}\mu.$$

For turbulent calculations, the Reynolds averaged Navier-Stokes equations [70] are solved using a turbulence model for closure. The one chosem througout this work is the Spalart-Allmaras turbulence model [76]. Following from the Boussinesq hypothesis [8], the averaged equations take the same form as the Navier-Stokes equations if the definitions of the viscosity and thermal conductivity are modified to incorporate both molecular and turbulent contributions. The total viscosities then become

$$\mu_{tot} = \mu + \mu_t, \qquad \lambda_{tot} = -\frac{2}{3}\mu_{tot},$$

and the thermal conductivity is given by

$$k_{tot} = \frac{\gamma\mu}{Pr} + \frac{\gamma\mu_t}{Pr_t},$$

where μ_t is the turbulent eddy viscosity and Pr_t is the turbulent Prandtl number ($Pr_t = 0.9$ for air).

To obtain a well-posed problem, appropriate boundary conditions must be imposed on the domain boundary. For the Euler equations, the appropriate solid wall boundary condition is zero velocity normal to the wall. For the Navier-Stokes equations, both the normal and tangential velocity components are zreo at the wall and either the temperature or the heat flux must be specified at the wall. Calculations are generally performed on a truncated domain, so in practice it is also necessary to introduce boundary conditions at the far field boundary.

2.2 Spatial Discretisation

For clarity, the 3D compressible Reynolds-averaged Navier-Stokes equations are expressed in the more concise way

$$\frac{\partial \mathcal{Q}}{\partial t} + \nabla . \mathcal{F}(\mathcal{Q}, \nabla \mathcal{Q}) = \mathcal{S}(\mathcal{Q}, \nabla \mathcal{Q}). \tag{2.2}$$

$\mathcal{Q}(\mathbf{x})$ is the vector of conserved variables, $(\rho, \rho u, \rho v, \rho w, \rho E)^T$. $\mathcal{F}(\mathcal{Q}, \nabla \mathcal{Q})$ is the total flux and the source term $\mathcal{S}$ is of the form $(0, 0, 0, 0, 0)^T$.

The discretisation described here is appropriate for any hybrid grids. Using the finite volume approach, equation (2.2) is integrated over some control volume, which after the application of the divergence theorem gives the expression

$$\mathcal{R}_j = \frac{1}{V_j} \left(\oint_{\partial V_j} \mathcal{F}(\mathbf{n}, \mathcal{Q}, \nabla \mathcal{Q}) ds - \int_{V_j} \mathcal{S}(\mathcal{Q}, \nabla \mathcal{Q}) dv \right) = 0, \quad \forall j \tag{2.3}$$

where V_j is the measure of the control volume associated with index j. Here the unknowns are stored at the nodes of a given grid, and the control volume is the 'median-dual' [6] which is constructed around each node $\mathbf{x}_j$ of the grid by joining the centroids of the cells surrounding the node with the midpoints of the edges (see figure 2.1). For interior grid points, the flux integration in equation (2.3) is approximated by using pre-computed weights for each edge of the grid, see [56, 38, 59]. These edge weights are anti-symmetric; the contribution of the edge appears

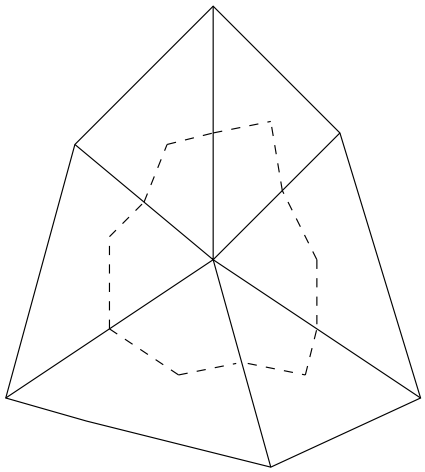


Figure 2.1 Median dual around an internal node.

with opposite signs at either end, ensuring conservation by construction.

$$\oint_{\partial V_j} \mathcal{F}(\mathbf{n}, \mathcal{Q}, \nabla \mathcal{Q}) ds \approx \sum_{i \in E_j} \mathcal{F}(\mathbf{n}_{ij}, \mathcal{Q}, \nabla \mathcal{Q})|_{\mathbf{x}=\frac{1}{2}(\mathbf{x}_i+\mathbf{x}_j)} \triangle s_{ij} \tag{2.4}$$

where E_j is the set of all nodes connected to node j via an edge, $\mathbf{n}_{ij}$ is a unit vector and $\triangle s_{ij}$ an area associated with the edge connecting nodes i and j.

For nodes on a boundary, extra terms from the boundary faces are added, so that the approximation of the flux will be written as

$$\oint_{\partial V_j} \mathcal{F}(\mathbf{n},,) ds \approx \sum_{i \in E_j} \mathcal{F}(\mathbf{n}_{ij},,)|_{\mathbf{x}=\frac{1}{2}(\mathbf{x}_i+\mathbf{x}_j)} \triangle s_{ij} + \sum_{k \in B_j} \mathcal{F}(\mathbf{n}_k,,)|_{\mathbf{x_j}} \triangle s_k \tag{2.5}$$

Here B_j is the set of boundary faces associated with node j (e.g. wall + inflow). $\mathbf{n}_k$ is the corresponding normal and $\triangle s_k$ an area. The edge weight conservation (or closure of the control volumes) implies that

$$\sum_{i \in E_j} \mathbf{n}_{ij} \triangle s_{ij} + \sum_{k \in B_j} \mathbf{n}_k \triangle s_k = 0 \ .$$

Hence, the discrete equivalent to equation (2.3) becomes

$$R_j = \frac{1}{V_j} \left(\sum_{i \in E_j} F_{ij} \triangle s_{ij} + \sum_{k \in B_j} F_k \triangle s_k - S_j V_j \right) \quad \forall j \tag{2.6}$$

where F_{ij} is the numerical flux in the direction $\mathbf{n}_{ij}$ associated with an antisymmetric edge (i, j), and F_k is the one associated with the boundary face k with no viscous contribution, since at the wall the boundary conditions are an adiabatic boundary with a zero relative velocity between the fluid and the solid wall.

It now remains to define the discrete flux functions, and then the spatial discretisation is complete. Since the flux $\mathcal{F}$ can be split into an inviscid and viscous part

$$\mathcal{F}(\mathbf{n}, \mathcal{Q}, \nabla \mathcal{Q}) = \mathcal{F}^I(\mathbf{n}, \mathcal{Q}) + \mathcal{F}^V(\mathbf{n}, \mathcal{Q}, \nabla \mathcal{Q})$$

for any unit normal $\mathbf{n}$, the discrete approximation F of $\mathcal{F}$ will have an inviscid and viscous part. Each of these is presented in the next two sections.

2.2.1 Evaluation of F^I

Basic discretisation

The scheme described here is motivated by the well-known MUSCL [32] approach, in which a functional representation of Q is used within each control volume to arrive at a Riemann problem at the interface between control volumes. Consider the flux F_{ij} to be evaluated at $\frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j)$. Let Q^+ and Q^- be values obtained by reconstruction within volumes i and j, respectively. Then, the flux at the interface is based on the flux-differencing ideas of Roe [69] combining central differencing of the nonlinear inviscid fluxes with a smoothing flux based on one-dimensional characteristic variables. It is expressed as

$$F^I_{ij} = \frac{1}{2}\left(\mathcal{F}^I_{ij}(Q^+) + \mathcal{F}^I_{ij}(Q^-) - |A_{ij}|(Q^+ - Q^-)\right) \tag{2.7}$$

where $\mathcal{F}_{ij} = \mathcal{F}(\mathbf{n}_{ij},,)$ and $A_{ij} = \partial\mathcal{F}^I/\partial\mathcal{Q}$. This approach is often seen in the literature [46, 6, 75]. It is crucially important to note that this characteristic decomposition is performed in a one-dimensional fashion in the direction $\mathbf{n}_{ij}$ at the interface between volumes i and j. With the premise that any inaccuracies associated with this approach are derived from this one dimensional approach, here we endeavour to modify the above to be (a) robust, (b) cheap to evaluate and (c) a good smoother for multigrid.

Since the reconstruction of Q^+ and Q^- is an expensive process, the first step is to approximate equation (2.7) by

$$F^I_{ij} = \frac{1}{2}\left(\mathcal{F}^I_{ij}(Q_i) + \mathcal{F}^I_{ij}(Q_j) - |A_{ij}|(Q^+ - Q^-)\right) \quad .$$

The flux terms now use the nodal variables and so this expression can be more easily interpreted as a central difference (or Galerkin) method with numerical smoothing, as in [56]. The evaluation of this central difference term is computationally cheap since the reconstruction of the values Q^+ and Q^- is not necessary. Since this discretisation is to be used within a 5-stage Runga-Kutta (see section 2.3) where the numerical smoothing will only be evaluated on a few of the steps, and the basic

flux terms will be evaluated on every step, a considerable computational saving has been made.

The next step is to modify the form of the numerical smoothing term. To motivate this, we consider the approach adopted by Lohner [46] in which the values Q_{j^+} and Q_{i^-} are constructed at points $\mathbf{x}_{j^+}$ and $\mathbf{x}_{i^-}$ such that $\mathbf{x}_{j^+}$, $\mathbf{x}_j$, $\mathbf{x}_i$ $\mathbf{x}_{i^-}$ are equi-spaced along a straight line. Consequently, many well-established four point schemes for the evaluation for the flux can be employed, see [32]. In particular, ignoring any limiting that may be necessary, the dissipation term for one such family of methods can be expressed as

$$|A_{ij}|(Q^+ - Q^-) = \frac{1}{2}(1-\kappa)|A_{ij}|\left[(\frac{1}{2}Q_{j^+} - Q_j + \frac{1}{2}Q_i) - (\frac{1}{2}Q_j - Q_i + \frac{1}{2}Q_{i^-})\right] \quad (2.8)$$

where $\kappa \in [0,1]$ represents a one-parameter family of second order schemes for a one-dimensional uniform mesh, with the exception $\kappa = 1/3$ being a third order scheme. The value, $\kappa = 1/2$, is used throughout this work.

With this motivation, we define the numerical inviscid flux F^I_{ij} to be

$$F^I_{ij} = \frac{1}{2}\left(\mathcal{F}^I_{ij}(Q_i) + \mathcal{F}^I_{ij}(Q_j) - \frac{1}{2}(1-\kappa)|A_{ij}|(L_j(Q) - L_i(Q))\right) \quad (2.9)$$

where L is an undivided pseudo-Laplacian with a negative unit central coefficient. Here, this is generalised for unstructured grids by defining L as

$$L_j(Q) = \frac{1}{\#(E_j)}\sum_{i \in E_j}(Q_i - Q_j) \quad (2.10)$$

where $\#(E_j)$ represents the number of elements in set E_j. This scheme is now very similar to some structured grid discretisations [35], and similar algorithms have also been successfully employed on unstructured grids [38].

Another important comment concerns the upwinding treatment. In areas where one of the characteristic speeds passes through zero, the definition of these must be modified to avoid the formation of non-physical expansion shocks and problems with stability. The treatment that is used is based upon the ideas of van Leer [81], and written as

$$|\lambda|_{vL} = max\left(|\lambda|, 2\Delta\lambda\right) \ .$$

In this definition, λ is the eigenvalue, and $\Delta\lambda$ the difference between the corresponding eigenvalues evaluated at each node of one edge. The effect of this fix is to maintain a certain minimum level of numerical smoothing which prevents non physical behaviour but without unnecessary corruption of the physical solution.

Modifications to pseudo-Laplacian

The drawback of using the simple pseudo-Laplacian is outlined in [13]. Broadly speaking, one wants

$$L_j(Q) \sim O(h^2)\, \nabla^2 \mathcal{Q}\big|_{\mathbf{x}=\mathbf{x}_j}$$

so that the dissipation term in equation (2.9) is $O(h^3)$. After being integrated around the control volume in equation (2.6), and divided by the volume, this gives an error which is $O(h^2)$ consistent with the truncation error of the basic central difference Galerkin approximation.

However, a simple Taylor series expansion of $L\mathcal{Q}$ about $\mathbf{x}_j$ reveals that

$$L_j = L_j(\mathbf{x}).\, \nabla \mathcal{Q}|_{\mathbf{x}=\mathbf{x}_j} + O(h^2)$$

where $L_j(\mathbf{x}) = (L_j x, L_j y, L_j z)^T$. Consequently, the local truncation error will not be second order unless the mesh is sufficiently smooth.

Another interpretation is that L will not preserve a linear solution. The basic Galerkin discretisation is exact when $\mathcal{F}(\mathcal{Q})$ is a linear function and $\mathcal{Q}$ varies linearly with $\mathbf{x}$. However, if $L_j(\mathbf{x})$ is not identically zero, $Q_j = \mathcal{Q}(\mathbf{x}_j)$ will not give a zero residual and so it is not a solution of the discrete flow equations.

This has been found to give poor results on general grids [13], so the following modification is made

$$L_j^{lp}(Q) = L_j(Q) - \nabla Q_j . L_j(\mathbf{x})$$

so that $L_j^{lp}(Q)$ will be 'linear preserving' provided ∇Q_j is exact for linears. That is, if $Q = \mathbf{a}.\mathbf{x} + c$ then $L_j(Q) = \mathbf{a}.L_j(\mathbf{x})$ and $\nabla Q_j = \mathbf{a}$, and so $L_j^{lp}(Q) = 0$.

The calculation of ∇Q_j is approximated using the edge weights.

$$\nabla Q_j = \sum_{i \in E_j} \frac{1}{2}(Q_i + Q_j)\mathbf{n}_{ij} \triangle s_{ij} + \sum_{k \in B_j} Q_j\, \mathbf{n}_k \triangle s_k = \sum_{i \in E_j} \frac{1}{2}(Q_i - Q_j)\mathbf{n}_{ij} \triangle s_{ij} \quad . \quad (2.11)$$

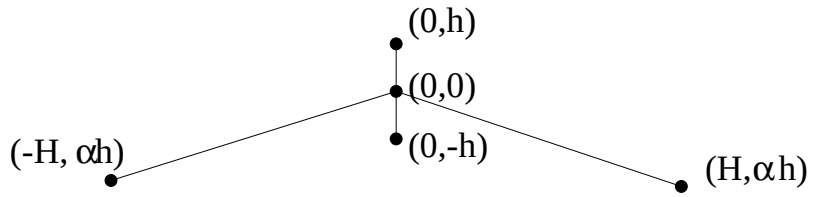


Figure 2.2 Schematic of highly stretched grid

It is worth noting that this discretisation is to be used within a multigrid method (see section 2.4), which requires the damping of high frequency error modes. A local mode analysis of L_j in equation (2.10) on a structured grid reveals good damping properties for high frequency modes [14]. The advantage of this linear preserving operator is the high-frequency damping properties of L^{lp} are identical to those of L since the correction term involves a central difference type of operator, which has minimal effect on high frequency modes. However, at boundary nodes the effect of the linear preserving correction can cause some of the weights of points other than the central one to become negative, threatening the stability. Such problems are avoided by limiting the linear preserving correction to prevent negative weights.

When calculating inviscid flows on unstructured grids which are not highly stretched, the use of the linearly-preserving Laplacian operator $L_j^{lp}(Q)$ gives a method which has been shown to be both accurate and robust [73, 74, 18]. However, a critical modification is required for the highly stretched grids needed to efficiently resolve a high Reynolds number boundary layer.

To exemplify the problems associated with applying L^{lp} on a highly stretched grid, consider the piece of two-dimensional grid in Fig. 2.2. Around the leading edge of an airfoil, the ratio ratio H/h is very large, of the order of 10^3 or greater, and α can also be more than 100. The central difference approximation to ∇Q is given by

$$Q_x = \begin{bmatrix} & 0 & \\ -\frac{1}{2H} & 0 & \frac{1}{2H} \\ & 0 & \end{bmatrix} Q, \qquad Q_y = \begin{bmatrix} & \frac{1}{2h} & \\ 0 & 0 & 0 \\ & -\frac{1}{2h} & \end{bmatrix} Q.$$

The simple pseudo Laplacian $L(Q)$ is given by

$$L(Q) = \frac{1}{4}\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix} Q \quad .$$

and hence $L(\mathbf{x}) = (0, 2\alpha h)$, and therefore

$$\nabla Q.L(\mathbf{x}) = \begin{bmatrix} & \alpha & \\ 0 & 0 & 0 \\ & -\alpha & \end{bmatrix} Q.$$

Consequently, severe difficulties can be expected if $|\alpha| \gg 1$ because the coefficients of the linear preserving correction will then become much larger than those of the basic Laplacian operator L. This loss of diagonal dominance in the smoothing discretisation typically results in numerical instability.

To avoid this problem, the following anisotropic scaling is introduced

$$\hat{L}_i(Q) = \left(\sum_{j \in E_j} \frac{1}{|\mathbf{x}_j - \mathbf{x}_i|} \right)^{-1} \sum_{j \in E_j} \frac{(Q_j - Q_i)}{|\mathbf{x}_j - \mathbf{x}_i|}.$$

In the example above one now gets

$$\hat{L}(Q) = \frac{Hh}{2(H+h)} \begin{bmatrix} & \frac{1}{h} & \\ \frac{1}{H} & -2\frac{H+h}{Hh} & \frac{1}{H} \\ & \frac{1}{h} & \end{bmatrix} Q,$$

and hence

$$\hat{L}(\mathbf{x}) = \frac{Hh}{2(H+h)} \begin{pmatrix} 0 \\ \frac{2\alpha h}{H} \end{pmatrix},$$

and

$$\nabla Q.\hat{L}(\mathbf{x}) = \frac{Hh}{2(H+h)} \begin{bmatrix} & \frac{\alpha}{H} & \\ 0 & 0 & 0 \\ & \frac{-\alpha}{H} & \end{bmatrix} Q \quad .$$

Thus, numerical difficulties will occur only if $|\alpha|h > H$ which is a much more reasonable restriction.

Figure 2.3 demonstrates the effect of the linear preserving modification for an RAE2822 2D airfoil; wiggles in the Mach contours are evident when using $\hat{L}$, but do not appear when $\hat{L}^{lp}$ is used. This extra smoothness in the solution only due to the linear preserving dissipation (the numerical dissipation has not been tuned or modified in any other respect) is thought to be especially important for the turbulence model, which uses highly non-linear functions in the definition of the source and destruction terms in different regions of the boundary layer.

The one disadvantage of this approach is the numerical smoothing operator $\hat{L}^{lp}$ is anisotropic on highly stretched grids, and so will only damp error modes which are high-frequency in the direction of highest grid resolution (i.e. across the boundary layer). This will become crucial to the multigrid strategy that is employed.

Treatment of shocks

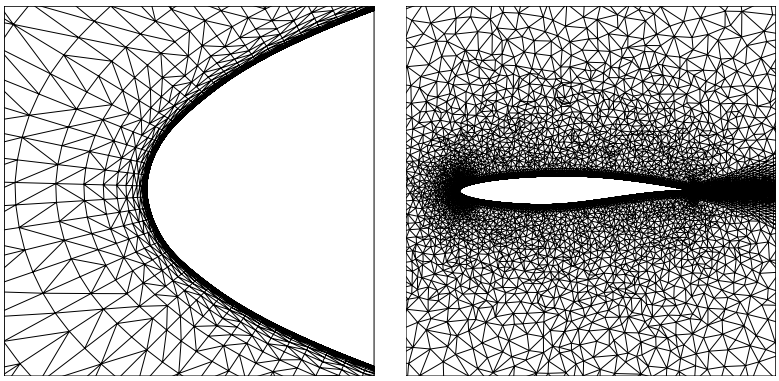
A major challenge is the monotonic resolution of discontinuous or very steep interior layers such as shocks and shear layers, whilst maintaining accuracy where the solution is smooth, for instance in boundary layers or in the far field. For that reason, the artificial dissipation consists of a nonlinear blend of second and fourth-differences, and a limiter is introduced so that the smoothing reverts to first order characteristic upwinding at shocks well known to be monotone and non-oscillatory. This formulation is designed to ape the one used in [35, 19], and is written as

$$F_{ij}^{I} = \frac{1}{2}\left(\mathcal{F}_{ij}^{I}(Q_i) + \mathcal{F}_{ij}^{I}(Q_j) - |A_{ij}|\left(-\frac{1}{3}(1-\Psi)(\hat{L}_i^{lp}(Q) - \hat{L}_j^{lp}(Q)) + \Psi(Q_i - Q_j)\right)\right) \tag{2.12}$$

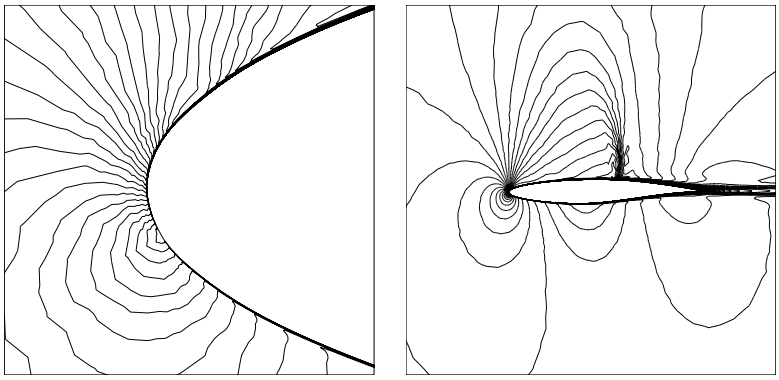
where

$$\Psi = \min\left(\epsilon^{(2)}\left|\frac{p_j - p_i}{p_i + p_j}\right|^2, 1\right) \tag{2.13}$$

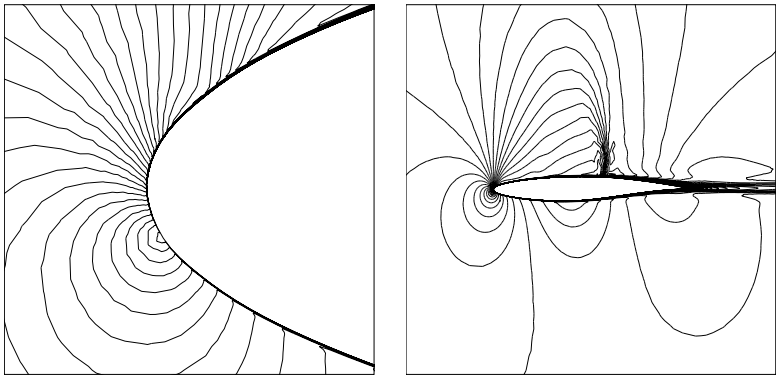
Here $\epsilon^{(2)}$ is a global user defined constant (taken as $\epsilon^{(2)} = 8$) and p the pressure at the corresponding node. As already mentioned in the previous section, the error introduced by the artificial dissipation terms is $O(h^3)$ and consequently, when the solution is smooth and the computation performed on a smoothly varying grid, second-order accuracy of the basic discretisation scheme is therefore observed. The



grid



Mach contours using $\hat{L}$



Mach contours using $\hat{L}^{lp}$

Figure 2.3 Effect of $\hat{L}^{lp}$ on the solution of an RAE2822 airfoil.

role of fourth-difference smoothing is to damp high frequency solution component which is essential for the successful application of a multigrid method to improve convergence to a steady state.

2.2.2 Evaluation of F^V

The viscous fluxes can be evaluated in a number of ways. One possibility is to use the Galerkin finite element approximation (see [38]). This has some difficulties associated with the treatment of the non-linearity within the edge-based data structure, and requires the storage of numerous viscous edge-weights. Another possibility, adopted here, is to approximate the viscous flux half-way along each edge (ie. $\mathcal{F}_{ij}^V$) and then use the usual integration rule around each volume, equation (2.3), thus giving a consistent finite volume treatment of the inviscid and viscous terms. This requires an approximation of ∇Q at the midpoint of each edge. The gradients of the flow variables can be approximated at the nodes using the existing edge-weights, equation (2.11). An approximation at the midpoint of the edge can then be obtained by a straightforward average,

$$\overline{\nabla Q}_{ij} = \frac{1}{2}\left(\nabla Q_i + \nabla Q_j\right) \quad .$$

However, as this is the average of two central differences, it will not damp high frequency modes. Although the inviscid flux includes numerical dissipation terms that will damp these modes, this is insufficient inside the boundary layer where the viscous terms dominate. To remedy this, the component of ∇Q in the direction along the edge is replaced by a simple difference along the edge, giving

$$\nabla Q_{ij} = \overline{\nabla Q}_{ij} - \left(\overline{\nabla Q}_{ij} \cdot \delta \mathbf{s}_{ij} - \frac{(Q_i - Q_j)}{|\mathbf{x}_i - \mathbf{x}_j|}\right) \delta \mathbf{s}_{ij} \tag{2.14}$$

where

$$\delta \mathbf{s}_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|} \quad .$$

In the boundary layer, it is the simple differences along the shortest edges which contribute to the dominant viscous flux terms, and so this formulation damps the high-frequency error modes. Fig. 2.4 illustrates what the stencil of the Laplacian

will be in a 1D representation, when ∇Q is discretised using a central difference scheme or a simple difference along the edge. When applied to a high frequency mode (Fig. 2.5) it appears that in the first case the Laplacian will not have any influence on it, and no damping will occur, which will be different in the other case. In other words, without the modification, a sawtooth mode in the boundary layer will not be damped because the stencil, spread over 5 points, will not see it, whereas with the correction, the 3 adjacent point stencil will. Without the addition of the 'edge-derivative' terms the algorithm failed to converge. Furthermore, this discretisation is still linear preserving, in the sense that a solution Q with linear spatial variation would give identically zero residuals if the function $\mathcal{F}(\mathcal{Q}, \nabla\mathcal{Q})$ were linear.

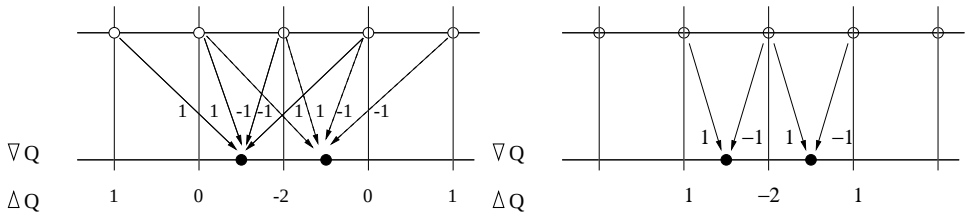


Figure 2.4 Stencils in 1D of ΔQ using a central difference scheme and a simple difference along the edge.

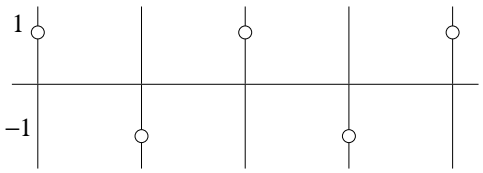


Figure 2.5 Representation of a high frequency mode.

2.2.3 Boundary Conditions

Because a multigrid method is being used, it is important that the residual R_j is well defined for all nodes, including boundary nodes. It is often the case with single-grid pseudo timestepping methods, that boundary conditions are imposed on the update

vector, or even on the solution after the update has taken place. Here a residual with the boundary conditions included is defined, which is appropriate since the FAS multigrid scheme transfers residuals between grids. The differing boundary conditions are summarised below.

slip This condition is imposed on inviscid walls, where boundary layer effects are ignored. The mass flux in the boundary flux F_k in equation 2.6 is set to zero. In addition, to enforce the slip condition the normal momentum components of the residual at all slip boundary nodes are explicitly removed. Thus the resulting update will not change the normal velocity components which are initialised to zero. To ensure that spurious normal velocity components are not introduced by the multigrid process, all nodal normal components are removed during the multigrid transfer operations.

The normals at the boundary nodes are calculated by averaging the face normals arising from all the surrounding faces of each boundary node. If a boundary node lies on the junction between some surface and a symmetry plane, the nodal boundary normal of the surface is projected onto the symmetry plane for consistency.

no-slip This condition is imposed on viscous walls, where the boundary layer is being modelled. All components of the momentum and the turbulence equation of the residual are explicitly set to zero.

free-stream This condition is imposed through the inviscid boundary flux term in equation (2.6), which is evaluated by solving the Riemann problem, that is

$$F_k = \frac{1}{2}\left(\mathcal{F}_k^I(Q_k) + \mathcal{F}_k^I(Q_\infty) - |A_k|(Q_k - Q_\infty)\right)$$

where Q_∞ is a pre-described free-stream state.

periodic Periodic boundaries are gridded such that nodes are matched across periodic boundaries. For example, if the top and bottom of a grid is periodic, then grid nodes on the top and bottom are matched. By summing contributions from the finite volume integrals at matching nodes, a consistent residual can

be constructed. By ensuring the residual is identical at matching nodes, along with Δt, Q and ∇Q the periodic condition is consistently imposed. The multigrid transfer operations are also modified to ensure that periodic conditions remain consistent.

other The basic methodology employed to specify a particular inlet/outlet condition, such as static pressure, enthalpy, etc.. is to modify the free-stream condition, so instead of using Q_∞ in the Riemann solver on the boundary, another state is used with the particular boundary condition imposed.

2.3 Smoothing iteration

The iterative scheme used to converge the discrete residuals to zero is pseudo time-stepping using the 5-stage Runge-Kutta method developed by Martinelli [48]. This can be expressed as

$$\begin{aligned} Q_j^{(0)} &= Q_j^n \\ Q_j^{(k)} &= Q_j^n - \alpha_k \Delta t_j R_j^{(k-1)}, \quad k = 1,2,3,4,5 \\ Q_j^{n+1} &= Q_j^{(5)} \end{aligned} \tag{2.15}$$

where

$$\begin{aligned} R_j^{(k-1)} &= C_j(Q^{(k-1)}) \; - \; B_j^{(k-1)} \\ B_j^{(k-1)} &= \beta_k \, D_j(Q^{(k-1)}) \; + \; (1-\beta_k) \, B_j^{(k-2)} \end{aligned}$$

where $C_j(Q^{(k-1)})$ is the convective contribution to R_j arising from the Galerkin approximation of the inviscid terms in equation (2.2), and $D_j(Q^{(k-1)})$ are the remaining parts due to the source term and the dissipation, both physical and numerical. The coefficients α_k and β_k are

$$\begin{aligned} \alpha_1 &= \tfrac{1}{4} \;\; \alpha_2 = \tfrac{1}{6} \;\; \alpha_3 = \tfrac{3}{8} \;\; \alpha_4 = \tfrac{1}{2} \;\; \alpha_5 = 1 \;, \\ \beta_1 &= 1 \;\; \beta_2 = 0 \;\; \beta_3 = \tfrac{14}{25} \;\; \beta_4 = 0 \;\; \beta_5 = \tfrac{11}{25} \;. \end{aligned}$$

This Runge-Kutta scheme is designed to have a large stability region with a low computational cost, since β_2 and β_4 are zero and so $D_j(Q^{(2)})$ and $D_j(Q^{(4)})$ need not be computed.

Preconditioning will be described later, but if one does not use it, then the standard approach is to use local timesteps. For the Navier-Stokes equations, the local timestep Δt_j is based on a combination of inviscid and viscous timesteps,

$$\frac{1}{\Delta t_j} = \frac{1}{CFL} \max\left(\frac{1}{\Delta t_j^I}, \frac{\epsilon^V}{\Delta t_j^V}\right)$$

where CFL is the inviscid CFL number and $\epsilon^V = 0.5$. The reason for this is that for the 5-stage Runge-Kutta scheme used in the present work, the maximum extent along the negative real axis is roughly twice the extent in either direction along the imaginary axis, suggesting that the parabolic Courant number is twice the hyperbolic one, since these two numbers reflects the extent of the stability region of the Runge-Kutta time-stepping scheme along the negative real and imaginary axes, respectively.

The inviscid time step is based on a spectral radius upper bound on the Jacobians of the discrete inviscid operator.

$$\frac{1}{\Delta t_j^I} = \frac{1}{V_j}\left(\sum_{i\in E_j} \rho(A_{ij}) \triangle s_{ij} + \sum_{k\in B_j} \rho(A_k) \triangle s_k\right).$$

Here $\rho(A)$ is the spectral radius of the matrix $\partial\mathcal{F}^I/\partial\mathcal{Q}$.

The viscous timestep is obtained from the quasi-linear form in term of primitive variables of the viscous terms and based again on the maximum spectral radius of the Jacobian matrices [19]. If $|\mathbf{x_i} - \mathbf{x_j}|$ is chosen as representative of the geometric quantities for each edge (i, j), the viscous timestep is defined by

$$\frac{1}{\Delta t_j^V} = \frac{1}{V_j}\sum_{i\in E_j} \rho(B_{ij}) \frac{1}{|\mathbf{x_i} - \mathbf{x_j}|} \triangle s_{ij}.$$

where $\rho(B)$ denotes the spectral radius of the matrix $\partial\mathcal{F}^V/\partial\mathcal{Q}$.

2.4 Multigrid method

2.4.1 Basic approach

Multigrid has had a major impact on CFD and has become an essential part of any successful steady-state flow algorithm. The fundamental concept behind any multigrid method is to have a sequence of successively coarser grids that can represent the smooth error modes of the finer grid, and some iterative 'smoothing' procedure which eliminates the high frequency error modes on each grid. Thus, all error modes are eliminated. This, along with the transfer operations of restriction (fine to coarse) and prolongation (coarse to fine) defines a multigrid method [87]. On structured grids, a sequence of grids can be constructed trivially by retaining alternate grid points in each direction. On unstructured grids, however, the generation of a sequence of grids becomes a non-trivial task. The four main approaches in the literature are briefly outlined below.

Non–nested Here an independent sequence of grids is produced by some 'black-box' grid generator, and then linked together through the use of efficient searching algorithms. This has been successfully adopted by several authors [62, 49, 15]. This approach allows the flexibility of using any grid-based solver, independent of the data structure used by the solver. However, the generation of the sequence of grids is not automatic, and requires a robust mesh generator which has good control of both the surface and interior point distribution. Often, this involves extensive user interaction. There are also severe problems in viscous applications with curved boundaries in which the boundary layer nodes at one location on one grid can lie wholly outside the other grid.

Agglomeration An increasingly popular approach is agglomeration, which has been successfully applied to very complex problems [84, 50, 51, 53, 52]. Here, coarse grid 'edge-weights' are constructed by fusion of fine grid control volumes. This is completely automatic and very powerful but is totally reliant on having a discretisation dependending only on 'edge-weights', not requiring an underlying grid. It also has the problem that the sum of accuracy of multigrid

restriction and prolongation violates the condition established by Hackbush [30] as being necessary for grid-independent convergence for the Navier-Stokes equations (see section 2.4.3). An *ad hoc* fix has then to be used to overcome it.

Refinement Another strategy is to produce a sequence of fine grids from a coarse grid, preferably in some sort of adaptive refinement procedure [4]. This seems an attractive proposition, however, this requires a strong coupling between the grid refinement and the surface spline definition. There are also difficulties with complex geometries (e.g. aircraft) when certain features (e.g. fairing) may not appear on the coarsest grid.

Grid Collapse The philosophy adopted in this work, as in [17, 18], is to use an automatic point removal algorithm to generate a sequence of coarse grids from an initial fine grid. This is completely automatic, needing no interaction with any grid generation process. The resulting grid sequence can be used by any grid based algorithm, including those which use an edge-based data structure.

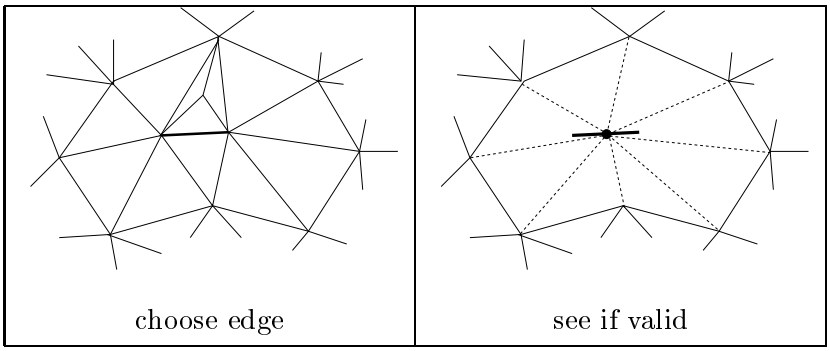


Figure 2.6 An example of a collapsed edge with retriangulation

2.4.2 Point-removal

Initially, the strategy was to replace two nodes connected by an edge, by a single node at the mid-point of the original edge, removing all the elements that were formed with that edge. The cavity created had then to be retriangulated (see Figure 2.6) subject to two constraints; all elements had to have positive volume

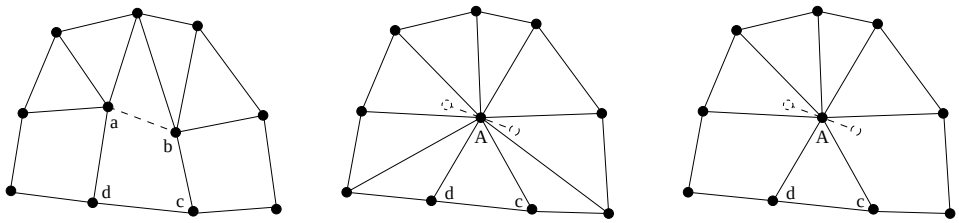


Figure 2.7 Collapsing the edge a-b. In the simplex algorithm, the resulting cavity is retriangulated (center). In the hybrid algorithm the type of all neighboring elements is maintained, except for those that are formed with the edge, element A-c-d (right).

and not more than a specified number of fine grid vertices could collapse into one coarse grid point. This algorithm developed by Crumpton [20] works well for simplex meshes of triangles in two dimensions or tetrahedra in three dimensions, but is not suitable for hybrid grids. The reason is that given a mesh composed of quadrilaterals and triangles, the removal of one edge will create a mesh with only triangles (see Figure 2.7). To be able to coarsen hybrid meshes, maintaining a large number of non-simplex element, Müller [59] has developed a similar algorithm which prevents the neighboring elements that are not formed with the collapsed edge to be triangulated (see Figure 2.7).

Figure 2.8 shows two possible collapsing sequences for a hexahedron when using this edge collapsing strategy. In the first sequence the hexahedron nicely degenerates into a tetrahedron via a prism and a pyramid. All of these elements are nicely shaped. In the second sequence an odd element is formed with three triangular faces and two highly twisted quadrilateral ones. Most likely a higher order discretisation on this element will be unstable and the formation of elements of this quality has to be prohibited. Due to the incremental nature of the collapsing process, prohibiting these elements stalls the edge collapsing algorithm rapidly.

The alternative that produces better shaped elements with an appropriate size on the coarser meshes is the element-collapsing algorithm, presented by Müller in [58]. The basic ingredients are maintained: a collapse is only permissible if (a) the edges are not overly lengthened, (b) the resulting geometry is of good quality and (c) the number of vertices collapsed into a particular one must not exceed a specific

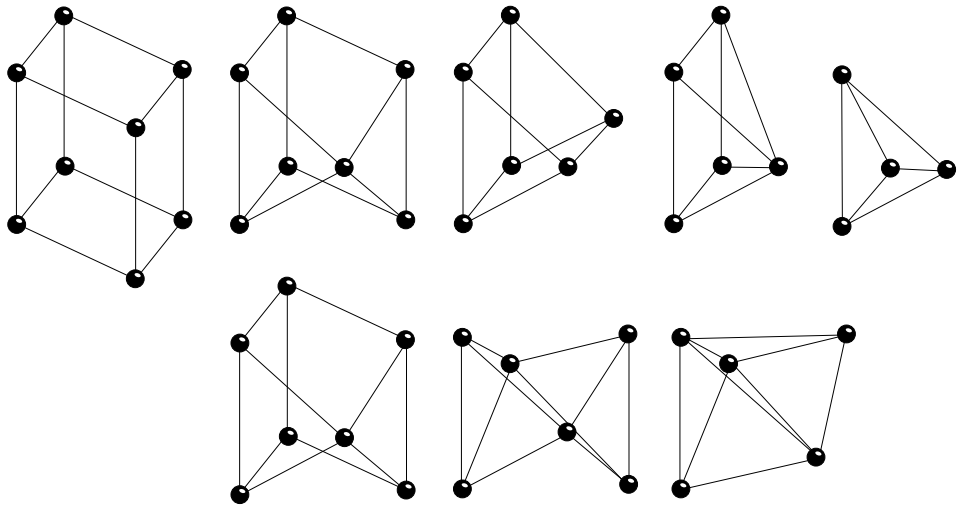


Figure 2.8 Collapsing edges on a hexahedron.

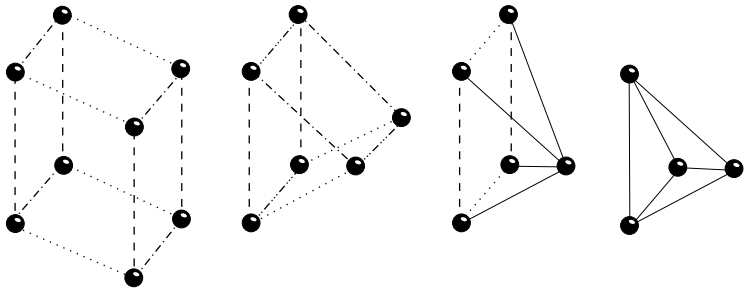


Figure 2.9 Parallel edges on primitive elements (shown in dashed lines). Parallel edges share no vertices, but connect the same faces. If a set of parallel edges is collapsed, the element disappears.

number. The major difference is that instead of collapsing the shortest edge of the mesh, the smallest element of the mesh is collapsed. In order to make an element disappear several edges have to be collapsed . In the current implementation the shortest edge of an element and all of the parallel ones are chosen to be collapsed. These are edges that connect between the same faces (see figure 2.9). An element collapse then happens by two faces of an element falling onto each other.

For isotropic meshes, the implementation of this algorithm is straightforward. Given a fine mesh, each edge is tagged with its length times a growth factor, say 2, as maximum length. The elements are sorted in a heap list for smallest volume and the algorithm tries to collapse the shortest edge and its parallel siblings. Fixing a certain maximum angle for the elements in the collapsed geometry, in the 2D

examples 135°, guarantees a minimum quality of the coarser mesh as well as positive volumes. This test is done by looping over all elements that are formed with any of the collapsed vertices and considering what remains on each element. Other edges on these elements may have been collapsed in earlier steps. E.g. a quadrilateral with one collapsed edge becomes a triangle, a doubly collapsed quadrilateral vanishes. The algorithm terminates once there are no edges left to be collapsed. All remaining elements and nodes are then identified and a coarsened grid is created.

The algorithm has to be modified once more to achieve directional coarsening in stretched layers. All long edges in stretched regions have to be prevented from collapsing. For this we need to identify short edges in stretched regions. A first criterion is that these edges are shorter by a given factor, say 3, compared to the largest neighbouring edge. Additionally it is required that there is at least one other neighbouring edge that is short and points into the same direction. This criterion ensures that single short edges in very irregular unstructured grids do not define a stretched region.

If an element is in a stretched region, all neighbouring long edges of the ones to be collapsed are prevented from any collapse. Once the stretched regions have been directionally coarsened in this way, the isotropic process collapses the rest of the domain. Figures 2.10 show the two first collapses for a hybrid grid around a RAE 2822 airfoil. It can be seen that the stretched part of the grid close to the airfoil remains regular and is coarsened exactly 1:2. The outer part of the structured region which is not stretched loses some regularity and the quadrilaterals collapse into larger quadrilaterals and triangles.

Figures 2.11 shows the finest grid and the two first levels of coarsening of a standard Onera M6 wing. The finest level has 147000 elements, and the sequence of coarser grids contains 60400 and 3900 elements respectively, corresponding to a coarsening ratio of 2.4 and 15.5. The low coarsening ratio between the two first level is due to the poor quality of the initial mesh where the strongest influence at this level is the angular tolerance. By looking at the grids, it is worth noting that places like trailing edge, leading edge or junction between wing and symmetry plan can be

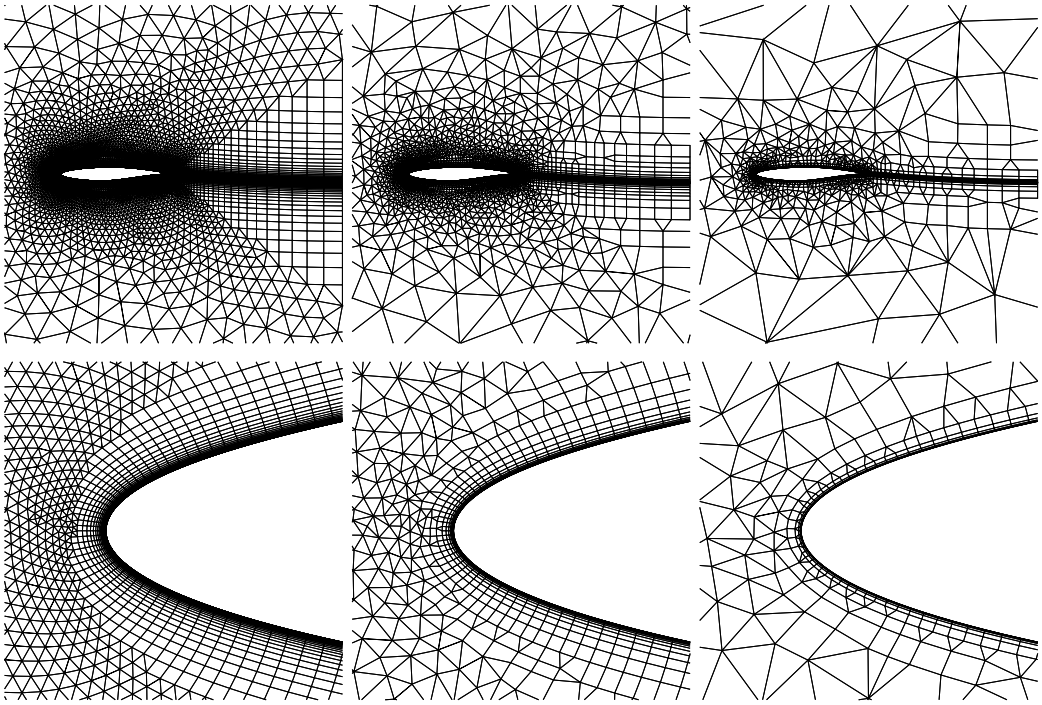


Figure 2.10 Coarsening a hybrid grid around a RAE2822 airfoil. Finest and two successive coarser levels.

more or less severely distorted. However, this is not of paramount importance to convergence to a steady state as it will be shown later.

2.4.3 Transfer operators

The multigrid method used here is the well-known Full Approximation Scheme [9]. Having established the smoother to be used (section 2.3), it remains to define the transfer operators used. With this point removal strategy, every point j on the coarse grid has associated with it the set K_j of points on the fine grid from which it has been derived through repeated application of the element-collapse procedure. Conversely, for each fine grid point, the index of the coarse grid point to which it has been collapsed is easily determined. This latter information is the only grid-to-grid connectivity needed for the transfer operations used here, and thus requires little addition storage.

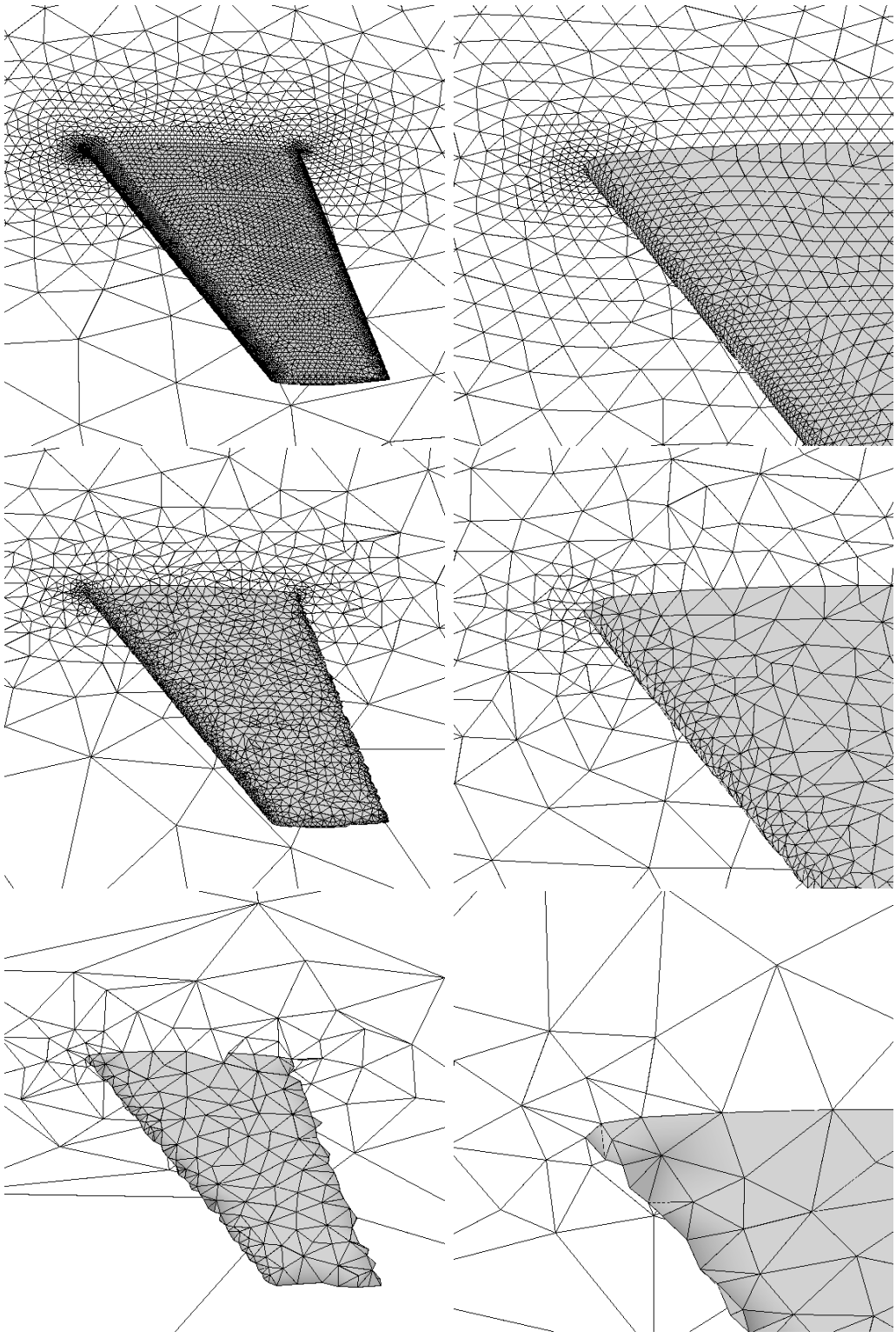


Figure 2.11 Coarsening a tetrahedral grid around an Onera M6 wing. Finest and two successive coarser levels.

To guarantee good convergence rate, multigrid theory [30] requires that

$$\mathcal{O}_P + \mathcal{O}_R > \mathcal{O}_E \tag{2.16}$$

where $\mathcal{O}_P$ and $\mathcal{O}_R$ are defined as the highest degree plus one of the polynomials that are interpolated exactly by the prolongation and restriction operator and $\mathcal{O}_E$ is the order of the differential equation, which equals 2 for the Navier-Stokes equations.

The transfer operators which are used are described below.

Prolongation: transfers corrections ΔQ from the coarse grid to the fine grid. A linear interpolation is used by the reconstruction of the gradients of corrections,

$$\forall i \in K_j \quad \Delta Q_i^h = \Delta Q_j^H + (\mathbf{x}_i^h - \mathbf{x}_j^H).\nabla(\Delta Q^H)_j$$

where ΔQ^H and ΔQ^h are the coarse and fine grid corrections respectively. The gradients of the corrections, $\nabla(\Delta Q^H)$ are evaluated using equation (2.11).

Restriction: transfers residuals from fine to coarse grids. The most obvious choice, analogous to full weighting for the elliptic case, is volume weighting

$$R_j^H = \frac{\sum_{i\in K_j} V_i^h R_i^h}{\sum_{i\in K_j} V_i^h} \quad .$$

This assumes that $V_j^H \approx \sum_{i\in K_j} V_i^h$, which is true for the majority of the grid, however, near boundaries where the surface is constrained, V_j^H can be considerably larger than $\sum_{i\in K_j} V_i^h$. Consequently the following limited volume weighting is used.

$$R_j^H = \frac{\sum_{i\in K_j} V_i^h R_j^h}{\max(V_j^H, \sum_{i\in K_j} V_i^h)} \quad .$$

Throughout this work V-cycles have been employed, along with first order upwinding for the inviscid discretisation (that is $\Psi = 1$ in equation (2.13)) on the coarse grids.

2.4.4 The Full Approximation Scheme (FAS)

The multigrid algorithm follows the Full Approximation Storage (FAS) scheme. If $N(Q) = f$ is a nonlinear system whose solution Q approximates a partial differential

equation, the iterative solver is expressed as

$$Q^{n+1} = Q^n + \mathcal{U}(f - N(Q^n)) \qquad n = 1, 2, ... \tag{2.17}$$

where $\mathcal{U}$ denotes the Runge-Kutta procedure and the quantity $f - N(Q^n)$ the residual operator identically equals to $-R$ in the previous text. $R = 0$ leads to a zero update, and the FAS procedure is then the following [16]:

- Pre-smooth errors on the fine grid by doing μ_1 relaxations:

$$Q^h := Q^h + \mathcal{U}_h(f^h - N^h(Q^h)).$$

- Form a coarse grid right hand side:

  $N(Q) = f$ is a nonlinear system, where Q is the solution vector, and f is a forcing function, the discrete approximation of the system on a grid characterized by spacing h is written as

$$N^h\left(\hat{Q}^h\right) = f^h$$

  where $\hat{Q}^h$ is the exact solution to the discrete system. Let Q^h be the current approximation and now define the error E^h as

$$E^h = \hat{Q}^h - Q^h \ ,$$

  so one can write

$$N^h\left(Q^h + E^h\right) = f^h.$$

  Subtract $N^h(Q^h)$ from both sides to obtain

$$N^h\left(Q^h + E^h\right) - N^h(Q^h) = f^h - N^h(Q^h) = R^h.$$

  Written for the coarse grid, characterised by spacing H, this equation becomes

$$N^H\left(I_h^H Q^h + E^H\right) - N^H\left(I_h^H Q^h\right) = I_h^H\left(f^h - N^h(Q^h)\right).$$

  By rearranging terms the coarse grid forcing function is defined as

$$f^H = I_h^H\left(f^h - N^h(Q^h)\right) + N^H\left(I_h^H Q^h\right).$$

  Solve $N^H(Q^H) = f^H$ using multigrid unless it is the coarsest mesh, in which case n_{cr} iterations of smoother are applied.

- Prolong the coarse grid correction:

$$Q^h := Q^h + I_H^h(Q^H - I_h^H Q^h).$$

- Post-smooth errors on the fine grid by doing μ_2 relaxations:

$$Q^h := Q^h + \mathcal{U}_h(f^h - N^h(Q^h)).$$

The multigrid cycling parameters μ_1 and μ_2 for the smoothing are usually set to 1. n_{cr}, the number of iteration on the coarser grid, is always chosen to be 5.

Usually, three to five grids are used. Multigrid can be applied in different cycling strategies, depending on the number of recursive calls of a coarser level (see figure 2.12). In this work, the multigrid cycle is traversed in a V cycle, which has proven to be the most efficient strategy.

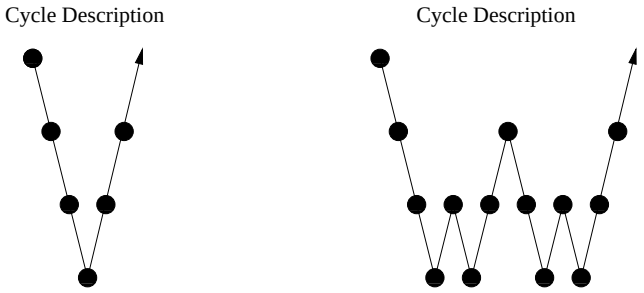


Figure 2.12 Multigrid cycle descriptions: (•) Apply M-stage Runge-Kutta scheme and transfer.

2.4.5 Influence of higher order discretisation on coarse grids

Usually, the switched higher order formulation of the numerical dissipation is used on the fine mesh and the first order treatment is used on all coarser meshes. Tables 2.1 and 2.2 show the influence on the convergence of a high order discretisation on coarse grids.

In a few cases, increasing the order of the numerical dissipation on coarser levels does accelerate the convergence to the steady state. However, this can not be generalised and is (a) grid dependent and (b) flow dependent. Note that by swapping the current pressure switch to a switch based on the velocity, improvements are

	Naca0012 airfoil	
Type of grid	MG parameters	CPU Time (sec)
Triangular grid	5 2 5 1 1 5	833
	5 3 5 1 1 5	799
	5 4 5 1 1 5	733
Quadrilateral grid	5 2 5 1 1 5	2560
	5 3 5 1 1 5	DNC
	5 4 5 1 1 5	DNC

Table 2.1 Inviscid transonic flow calculation. Type of grid, multigrid parameters (# grids, level on/above which first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level), CPU time in seconds. DNC: did not converge

	RAE2822 airfoil	
Type of grid	MG parameters	CPU Time (sec)
Triangular grid	5 2 5 1 1 5	1052
	5 3 5 1 1 5	1126
Hybrid grid	5 2 5 1 1 5	1137
	5 3 5 1 1 5	DNC
	5 4 5 1 1 5	DNC

Table 2.2 Standard RAE test Case 9. Types of grid, multigrid parameters (# grids, level on/above which first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level), CPU time in seconds. DNC: did not converge

achieved for viscous calculations on hybrid grids, since convergence is obtained with a speed up of roughly 30% using second order on the two first level instead of on the finest one only. Although convergence is achieved, the solution appears less accurate, particularly in terms of shock location. Consequently, the standard technique remains the most robust and is retained throughout this work.

2.5 Turbulence Model

2.5.1 Description

The turbulence model used throughout this work is the Spalart-Allmaras turbulence model [76]. It is a one-equation model that takes the form of a scalar convection-diffusion equation with source terms,

$$\frac{\partial \tilde{\nu}}{\partial t} + u\frac{\partial \tilde{\nu}}{\partial x} + v\frac{\partial \tilde{\nu}}{\partial y} + w\frac{\partial \tilde{\nu}}{\partial z} = \frac{1}{\sigma}\left(\nabla.\left[\left(\nu + \tilde{\nu}\right)\nabla\tilde{\nu}\right] + c_{b2}\left(\nabla\tilde{\nu}\right)^2\right) + \mathcal{S}, \tag{2.18}$$

where ν is the molecular kinetic viscosity and $\tilde{\nu}$ is the turbulent working variable. The source terms have the form

$$\mathcal{S} = c_{b1}\tilde{S}\tilde{\nu} - \left(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2 + f_{t1}\triangle u^2, \tag{2.19}$$

which may be divided into production, destruction and trip contributions

$$\mathcal{S} \equiv P(\tilde{\nu}) - D(\tilde{\nu}) + T$$

using the definition

$$\begin{aligned} P(\tilde{\nu}) &= c_{b1}\tilde{S}\tilde{\nu}, \\ D(\tilde{\nu}) &= \left(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right)\left(\frac{\tilde{\nu}}{d}\right)^2, \\ T &= f_{t1}(\triangle u)^2. \end{aligned}$$

The trip term provides a mechanism for triggering transition at a specified location on the geometry.

The equation is put into a non dimensional form by introducing

$$u^* = \frac{u}{\sqrt{p_\infty/\rho_\infty}}, \quad v^* = \frac{v}{\sqrt{p_\infty/\rho_\infty}}, \quad w^* = \frac{w}{\sqrt{p_\infty/\rho_\infty}},$$

$$x^* = \frac{x}{L}, \quad y^* = \frac{y}{L}, \quad z^* = \frac{z}{L}, \quad \tilde{\nu}^* = \frac{\tilde{\nu}}{\sqrt{p_\infty/\rho_\infty}} \quad .$$

The turbulent eddy viscosity is defined by

$$\nu_t = \tilde{\nu} f_{v1},$$

and the auxiliary relations used to construct the production and destruction terms are

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi = \frac{\tilde{\nu}}{\nu}, \qquad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}},$$
$$f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}}, \qquad g = r + c_{w2}(r^6 - r), \qquad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2},$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \qquad S = \sqrt{\left(\frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)^2}.$$

Here, d is the distance to the nearest wall and the closure constants are

$$c_{b1} = 0.1355, \qquad c_{b2} = 0.622, \qquad \sigma = \tfrac{2}{3}, \qquad c_{v1} = 7.1,$$
$$c_{w1} = \tfrac{c_{b1}}{\kappa^2} + \tfrac{1+c_{b2}}{\sigma}, \qquad c_{w2} = 0.3, \qquad c_{w3} = 2, \qquad \kappa = 0.41.$$

The auxiliary relations for the trip terms are

$$\begin{aligned} f_{t1} &= c_{t1}\, g_t\, \exp\left(-c_{t2}\frac{S_t^2}{(\triangle u)^2}\left[d^2 + g_t^2 d_t^2\right]\right), \\ f_{t2} &= c_{t3}\, \exp\left(-c_{t4}\chi^2\right), \\ g_t &= min(0.1, \frac{\triangle u}{S_t \triangle x_t}), \end{aligned}$$

with the additional closure constants given by

$$c_{t1} = 1, \quad c_{t2} = 2, \quad c_{t3} = 1.2, \quad c_{t4} = 0.5.$$

Here, d_t is the distance to the trip point on the wall, S_t is the wall vorticity at the trip, $\triangle u$ is the difference in velocity between the field cell and the trip and $\triangle x_t$ is the grid spacing along the wall at the trip. At a solid wall, the appropriate boundary condition is $\tilde{\nu} = 0$.

2.5.2 Implementation

A first important point to notice is that the trip terms are omitted. These are present to simulate the transition, but for many turbomachinery applications, like the bypass duct of a turbofan engine, the flow is supposed to be fully turbulent. Modelling of the transition is then not necessary and not considered in this work. However it is possible to obtain transition, and the strategy would then consist of

considering a small domain where the transition is expected in which the source terms are multiplied by a factor which increases linearly from 0 to 1. The advection and diffusion terms are not touched by this procedure, so that the transport of the turbulent front is still realized. This approach is much more suitable for 3D problems for which the definition of the transition point becomes tricky and difficult to implement.

The turbulence model requires the normal distance to the nearest viscous wall at each node, which is pre-computed for each grid. It is important that only viscous walls are considered when calculating this distance, in order to avoid the production and destruction of turbulence viscosity at slip walls, where all boundary layer effects are ignored. At slip walls the turbulence model essentially reduces to a simple advection operator, as required.

The diffusion term includes the non-conservative term

$$\nabla\tilde{\nu}.\nabla\tilde{\nu}$$

which is difficult to discretise in a manner that ensures positivity. Thus, the diffusion operator is reformulated in a conservative form [76] by assuming that the molecular viscosity ν is constant, which is a good approximation, since it does not vary significantly. The diffusion operator is then written as

$$\begin{aligned}\frac{1}{\sigma}\left(\nabla.\left(\left(\nu+\tilde{\nu}\right)\nabla\tilde{\nu}\right)+c_{b2}\left(\nabla\tilde{\nu}\right)^2\right) &= \frac{1+c_{b2}}{\sigma}\nabla.\left[(\nu+\tilde{\nu})\nabla\tilde{\nu}\right]-\frac{c_{b2}}{\sigma}(\nu+\tilde{\nu})\nabla^2\tilde{\nu} \\ &= \frac{1}{\sigma}\nabla\left[(\nu+(1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right]-\frac{c_{b2}}{\sigma}\tilde{\nu}\nabla^2\tilde{\nu}.\end{aligned}$$

As far as possible, the turbulence model is discretised in the same way as the Navier-Stokes equations, thereby simplifying the programming implementation. However, because the advective part is not in conservative form, the formulation changes slightly through the approximation

$$\begin{aligned}\int_{V_j}\mathbf{u}.\nabla\tilde{\nu}|_{\mathbf{x}_j}\,dV \approx \sum_{i\in E_j}\frac{1}{2}\left((\mathbf{u}_j.\mathbf{n}_{ij})(\tilde{\nu}_i+\tilde{\nu}_j)-\right. \\ \left.|\mathbf{u}_j.\mathbf{n}_{ij}|((1-\psi)(\hat{L}_j^{lp}(\tilde{\nu}_i)-\hat{L}_j^{lp}(\tilde{\nu}_j))+\psi(\tilde{\nu}_i-\tilde{\nu}_j))\right)\triangle s_{ij}\end{aligned}$$

where ψ is the scalar equivalent of Ψ in equation (2.13). Again the numerical dissipation is a blend of second and fourth differences, but for robustness purposes, ψ is set to 1, enforcing a first order treatment of the turbulence model.

The source terms are sensitive to the method used to compute the distance to the wall, and in particularly in regions which fall below the logarithmic region of the boundary layer, the exact distance is required, consequently, for each vertex of the mesh, the minimum distance from the vertex to the closest viscous solid wall, which implies finding the perpendicular projection onto the wall, is precomputed and stored.

To preserve the positivity of $\tilde{\nu}$, several modifications are made to the Runge-Kutta time integration procedure. The contribution of the turbulent source operator is treated implicitly to limit the rate of exponential decay in the solution. This is equivalent to employing the standard explicit integration procedure with a reduced local time step

$$\Delta t_{imp} = \frac{\Delta t}{1 - \left|\frac{\partial \mathcal{S}}{\partial \nu}\right| \Delta t}$$

where Δt is the explicit time step given, at node j by

$$\frac{1}{\Delta t_j} = \frac{1}{2V_j}\left(\sum_{i \in E_j} |\mathbf{u}_j.\mathbf{n}_{ij}| \triangle s_{ij} + \frac{2(\nu + \tilde{\nu})}{\sigma}\frac{\triangle s_{ij}}{|\mathbf{x_i} - \mathbf{x_j}|}\right).$$

For robustness, the timestep is prevented from becoming too small by using the Harten entropy fix [31] with a minimum cut off value set to $\frac{c}{8}$, where c denotes the local speed of sound.

While the implicit treatment described above provides a useful mechanism for limiting the evolution of the turbulence equation in regions of the flow where rapid decay might otherwise result in negative values of $\tilde{\nu}$, this treatment does not guarantee positivity throughout the Runge-Kutta time-stepping procedure. Therefore, it is important to limit the update to the scheme, whenever a negative value of $\tilde{\nu}$ would result. Defining the standard update by

$$\Delta\tilde{\nu} = \alpha_k \Delta t_{imp} R^{k-1},$$

positivity is guaranteed by the following limited update

$$\Delta\tilde{\nu}_{lim} = \begin{cases} \Delta\tilde{\nu}, & \Delta\tilde{\nu} \leq 0, \\ \frac{(\tilde{\nu}^n - \tilde{\nu}_{min})\Delta\tilde{\nu}}{(\tilde{\nu}^n - \tilde{\nu}_{min}) + \Delta\tilde{\nu}}, & \Delta\tilde{\nu} > 0 \quad . \end{cases}$$

A minimum value ν_{min} is maintained throughout the domain, and in practice, this one is set to the freestream value $\tilde{\nu}_{min} = \tilde{\nu}_\infty = 10 \times \mu_\infty$.

Solved within the same multigrid algorithm, the source terms of the turbulence model are poorly resolved on the coarse levels, and consequently, as in [63], the fine mesh contribution to the forcing function that drives the coarse mesh corrections is deflated in regions of the flow where the source terms are strongly active. This is accomplished using the denominator from the implicit time step, which is equal to unity in inviscid regions and dominated by the Jacobian $\frac{\partial \mathcal{S}}{\partial \nu}$ inside the boundary layer.

The current implementation of the turbulence model has only be tested on a series of airfoil calculations and compared with experimental data. Although it performs reasonably well and seems to be in good agreement with the literature [63, 71, 76], no further investigation and checking concerning the viscous sublayer and log-law region have been made, so that the accuracy of the implementation can be questioned. Nevertheless, it should be emphasised that the main focus of this work concerns convergence acceleration rather than getting accurate prediction of the computed flows.

Chapter 3

Preconditioner

Convergence rates obtained with the standard method described previously happen to be rather poor, particularly for viscous calculations. A way to overcome discrete stiffness in the Euler and Navier-Stokes equations is to use a matrix timestep or preconditioner which seek to improve the convergence rate without affecting the steady-state solution.

Preconditioning techniques belong to two classes. The first class, physical preconditioners, is very successful for the Euler equations on structured meshes [82, 44] but these preconditioners are difficult to extend to unstructured meshes due to the lack of directionality in the unstructured case. The fundamental idea is to equalise the characteristic wave speeds through a computational control volume in order to eliminate analytic stiffness due to the disparity of the propagative speeds.

The other class of preconditioners look at the discretised system rather than the analytical equations. They construct a matrix which has the effect of clustering the eigenvalues of the residual spatial operator in a region of the complex plane where the iterative method has good damping properties. The present work follows this approach and uses the block Jacobi preconditioner which has been successfully applied to the turbulent Navier-Stokes equations on structured grids by Pierce [63] and to the 2D inviscid and laminar viscous flows on unstructured grids by Ollivier-Gooch

[61]. The extension to the 3D turbulent Navier-Stokes equations on unstructured grids is straightforward and it has been incorporated within the multigrid solver described in the previous chapter. In this chapter, the details of the construction of the block Jacobi preconditioner, including crucial modifications at boundaries are presented.

3.1 Approach

Scheme Description

The pre-conditioned semi-discrete equation appears as

$$P^{-1}\frac{dQ}{dt} + R(Q) = 0, \tag{3.1}$$

where Q denotes the set of conservative variables, $R(Q)$ the residual vector of the spatial discretisation and P^{-1} the local preconditioner. The solution is updated via the same multistage scheme described in section 2.3 with the local preconditioner which may be interpreted as a matrix timestep as it will be demonstrated later.

The block-Jacobi preconditioner is based on a local linearisation of the 3D Navier-Stokes equations, and constructed by extracting the terms corresponding to the central node thereby giving a block-diagonal matrix. As the flux can be split into an inviscid and viscous part, the matrix preconditioner will have contributions coming from both.

3.2 The Inviscid Contribution

Using a finite volume approach, the integration of the inviscid terms over some control volume Ω gives, after the application of the divergence theorem,

$$R_j^I = \frac{1}{V_j}\oint_{\partial\Omega} \mathcal{F}^I(\mathbf{n}, \mathcal{Q})dS, \tag{3.2}$$

where V_j is the measure of the control volume associated with index j, and $\mathcal{F}^I(\mathbf{n}, \mathcal{Q})$ is the inviscid flux in the direction of the unit vector $\mathbf{n}$. As explained in section 2.2

the discrete approximation to equation (3.2) is

$$R_j^I = \frac{1}{V_j}\left(\sum_{i\in E_j} F_{ij}^I \triangle s_{ij} + \sum_{k\in B_j} F_k^I \triangle s_k\right), \quad \forall j, \tag{3.3}$$

and the numerical flux takes the form

$$F_{ij}^I = \frac{1}{2}\left(\mathcal{F}_{ij}^I(Q_i) + \mathcal{F}_{ij}^I(Q_j) - |A_{ij}|(-\frac{1}{3}(1-\Psi)(L_i(Q) - L_j(Q)) + \Psi(Q_i - Q_j))\right) \tag{3.4}$$

where $A_{ij} = \dfrac{\partial \mathcal{F}}{\partial \mathcal{Q}}$, and its absolute value $|A_{ij}|$ is defined to be $T|\Lambda|T^{-1}$, with $|\Lambda|$ being the diagonal matrix of absolute eigenvalues, and T the corresponding matrix of right eigenvectors. Linearising locally, the resulting equation becomes

$$F_{ij}^I = \frac{1}{2}\left(A_{ij}Q_i + A_{ij}Q_j - |A_{ij}|(-\frac{1}{3}(1-\Psi)(L_i(Q) - L_j(Q)) + \Psi(Q_i - Q_j))\right).$$

For convenience, the same name is used to denote the flux Jacobian, but it is understood that it is different from the one in equation (3.4), since it results from the linearising procedure. It is listed in detail in Appendix A.

Turning now to the computation of the matrix preconditioner for the Euler part, $\sum_{i\in E_j}(A_{ij}Q_j)$ is identically zero, because it corresponds to the integration of a constant over a closed domain, and consequently, considering only the terms which have a dependence on the central node, one gets

$$\left(P_j^I\right)^{-1} = \frac{1}{2V_j}\left(\sum_{i\in E_j} |A_{ij}|\frac{1+2\Psi}{3} \triangle s_{ij} + \sum_{k\in B_j} |A_k|\frac{1+2\Psi}{3} \triangle s_k\right).$$

It appears that the preconditioner is not identical for a 2nd ($\Psi = 1$) or for a 4th order difference scheme ($\Psi = 0$), but it shows that it is acceptable to base it on a first order discretisation even when using higher order schemes; the resulting timestep will only be underestimated. This is a slight difference from the structured approach where the block-Jacobi preconditioner remains the same for both schemes [64].

Finally, the inviscid contribution is

$$\left(P_j^I\right)^{-1} = \frac{1}{2V_j}\left(\sum_{i\in E_j} |A_{ij}| \triangle s_{ij} + \sum_{k\in B_j} |A_k| \triangle s_k\right). \tag{3.5}$$

3.3 The Viscous Contribution

The integration of the viscous terms follows the usual rule over each volume, equation (3.2), giving a consistent finite volume treatment of the inviscid and viscous fluxes. Consequently, the viscous residual may be written

$$R_j^V = \frac{1}{V_j} \oint_{\partial\Omega} \mathcal{F}^V(\mathbf{n}, \mathcal{Q}, \nabla\mathcal{Q}) dS \ . \tag{3.6}$$

The viscous flux $\mathcal{F}^V$ contains expressions of the form $\mu u \frac{\partial u}{\partial x}$ that can be linearised about a constant state $(\bar{u}, \bar{\mu})$ to become $\bar{\mu}\bar{u}\frac{\partial u}{\partial x}$. Following this procedure, equation (3.6) is approximated using the same pre-computed edge weights as mentioned previously, but without any viscous contribution from the boundary faces for the reasons explained in section 2.2. Thus,

$$R_j^V = \frac{1}{V_j} \sum_{i \in E_j} F_{ij}^V \triangle s_{ij} \quad \forall j \ , \tag{3.7}$$

where F_{ij}^V is the numerical viscous flux in the direction $\mathbf{n}_{ij}$ associated with the edge (i, j). The data structure which is used in representing the unstructured grids is the edge structure. The evaluation of the gradients has to be done over the control volume and consequently it makes direct use of quantities like $\frac{\partial u}{\partial x}$ or $\frac{\partial u}{\partial y}$ difficult. To write the viscous contribution of the matrix preconditioner the following approximations are made:

- All cross derivatives are neglected.
- ∇Q is approximated by $\mathbf{l}.\frac{\partial \mathcal{Q}}{\partial \mathbf{l}}$, where $\mathbf{l}$ is a unit vector for the edge pointing from node i to node j and $\frac{\partial \mathcal{Q}}{\partial \mathbf{l}} = \frac{Q_j - Q_i}{|\mathbf{x}_j - \mathbf{x}_i|}$.

After having rearranged the terms, (3.7) can then be written

$$R_j^V = \frac{1}{V_j} \sum_{i \in E_j} BM^{-1} \frac{Q_j - Q_i}{|\mathbf{x}_j - \mathbf{x}_i|} \triangle s_{ij} \ ,$$

where B is a 5×5 matrix calculated with respect to the set of primitive variables $\mathcal{Q}_p = (\rho, u, v, w, p)^T$, and transformed to the conservative variables using the transformation matrix $M = \dfrac{\partial \mathcal{Q}}{\partial \mathcal{Q}_p}$. Thus, the viscous preconditioner takes the form

$$\left(P_j^V\right)^{-1} = \frac{1}{V_j} \sum_{i \in E_j} BM^{-1} \frac{1}{|\mathbf{x}_j - \mathbf{x}_i|} \triangle s_{ij} \ . \tag{3.8}$$

The full matrix preconditioner is

$$P_j^{-1} = \left(P_j^I\right)^{-1} + \left(P_j^V\right)^{-1}. \tag{3.9}$$

In the Runge-Kutta integration (see equation 2.15), the local timestep Δ_j is replaced by $CFL\ P_j$, where CFL is set to the maximum value that guaranties the field of values to lie inside the stability region. P may then be interpreted as a matrix time step.

3.4 Slip Boundary Condition Adjustment to Preconditioner

To form the block-Jacobi preconditioner, the inviscid and viscous Jacobians need to be calculated at each node of the grid. However, at the wall, the conditions require an adiabatic boundary with zero normal pressure gradient and a zero relative velocity between the fluid and the solid wall. Hence, no viscous contribution must be evaluated since this one reduces itself to zero. In fact, only a no-slip condition has to be satisfied which is achieved by setting all momentum components in the residual to zero.

For Euler calculations, the procedure is slightly different. In addition to the corrections made on the residual, the preconditioner is modified at the wall in order that the condition $\mathbf{u}.\mathbf{n} = \mathbf{0}$ is satisfied; $\mathbf{u}$ and $\mathbf{n}$ denote respectively the velocity vector and the unit normal vector to the wall. This is accomplished by re-evaluating the matrix in the coordinate system (x_n, x_{t_1}, x_{t_2}), by using a rotation matrix T from the original (x, y, z) coordinate system to the new one. x_n is the coordinate in the direction normal to the surface and the other two are mutually orthogonal tangential coordinates. Once done, it is transformed back to the original coordinate system [26]. Generally speaking, the theory is the following:

Let us consider an iterative scheme expressed by

$$P^{-1}\frac{dQ}{dt} = -R, \tag{3.10}$$

where P^{-1}, Q, and R denote respectively the preconditioning matrix, the flow variables, and the residual. Calling by T the transformation matrix, whose expression is

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_x & n_y & n_z & 0 \\ 0 & t_{x_1} & t_{y_1} & t_{z_1} & 0 \\ 0 & t_{x_2} & t_{y_2} & t_{z_2} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

the change in Q over a unit timestep can be written as

$$P^{-1}T^{-1}T\Delta Q = P^{-1}T^{-1}\Delta\bar{Q} = -R,$$

and by multiplying by T,

$$TP^{-1}T^{-1}\Delta\bar{Q} = -TR = -\bar{R}$$

where $\bar{Q}$ denotes the flow variables, and $\bar{R} = [\bar{R}_1, \bar{R}_2, \bar{R}_3, \bar{R}_4, \bar{R}_5]^T$ the residual, both of them in the transformed coordinate system. In this system, the normal momentum equation must be discarded, including the residual $\bar{R}_2$, and replaced by the condition that $u_n = 0$ and hence $\Delta u_n = 0$, where u_n is the normal component of the velocity vector. The first step is achieved by writing

$$MTP^{-1}T^{-1}\Delta\bar{Q} = -M\bar{R} \tag{3.11}$$

where

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

and the second step by adding the condition

$$S\Delta\bar{Q} = 0 \tag{3.12}$$

where

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The modified slip boundary matrix in the transformed coordinate system is obtained by summing (3.12) and (3.11), and in the original system of coordinate, its expression will then be

$$T^{-1}(MTP^{-1}T^{-1} + S)T\Delta Q = -T^{-1}MTR. \tag{3.13}$$

Noting that $M = I - S$, equation (3.13) can be re-written as

$$\left[P^{-1} - T^{-1}ST(P^{-1} - I)\right]\Delta Q = -(I - T^{-1}ST)R, \tag{3.14}$$

where $T^{-1}ST$ only involves the unit normal vector, and is written

$$T^{-1}ST = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & n_x^2 & n_x n_y & n_x n_z & 0 \\ 0 & n_x n_y & n_y^2 & n_z n_y & 0 \\ 0 & n_x n_z & n_y n_z & n_z^2 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This correction to the preconditioner is important. Without it, the algorithm fails to converge because zeroing out the normal momentum residual does not produce a zero change to the normal velocity.

3.5 Components and implementation of the Matrix Preconditioner

3.5.1 Components

Inviscid

The absolute value of the conservative Jacobian is [37]

$$|A| = T|\Lambda|T^{-1},$$

and the absolute value of the eigenvalue matrix appears as

$$|\Lambda| = \begin{bmatrix} |q_n| & 0 & 0 & 0 & 0 \\ 0 & |q_n| & 0 & 0 & 0 \\ 0 & 0 & |q_n| & 0 & 0 \\ 0 & 0 & 0 & |q_n + c| & 0 \\ 0 & 0 & 0 & 0 & |q_n - c| \end{bmatrix},$$

where (n_x, n_y, n_z) is unit normal of the surface through which the flux is evaluated, and $q_n = un_x + vn_y + wn_z$.

The right eigenvectors are the columns of

$$T = (R1|R2|R3|R4|R5)$$

where

$$R1 = \begin{pmatrix} n_x \\ un_x \\ vn_x + cn_z \\ wn_x - cn_y \\ \frac{q^2}{2}n_x + c(vn_z - wn_y) \end{pmatrix}, \quad R2 = \begin{pmatrix} n_y \\ un_y - cn_z \\ vn_y \\ wn_y + cn_x \\ \frac{q^2}{2}n_y + c(wn_x - un_z) \end{pmatrix},$$

$$R3 = \begin{pmatrix} n_z \\ un_z + cn_y \\ vn_z - cn_x \\ wn_z \\ \frac{q^2}{2}n_z + c(un_y - vn_x) \end{pmatrix}, \quad R4 = \begin{pmatrix} 1 \\ u + cn_x \\ v + cn_y \\ w + cn_z \\ H + cq_n \end{pmatrix}, \quad R5 = \begin{pmatrix} 1 \\ u - cn_x \\ v - cn_y \\ w - cn_z \\ H - cq_n \end{pmatrix}.$$

and the left eigenvectors are the rows of

$$T^{-1} = \frac{1}{c^2}(L1|L2|L3|L4|L5)^T$$

given by

$$L1 = \begin{pmatrix} \left(c^2 - (\gamma-1)\frac{q^2}{2}\right) n_x + c(wn_y - vn_z) \\ (\gamma-1)un_x \\ (\gamma-1)vn_x + cn_z \\ (\gamma-1)wn_x - cn_y \\ -(\gamma-1)n_x \end{pmatrix},$$

$$L2 = \begin{pmatrix} \left(c^2 - (\gamma-1)\frac{q^2}{2}\right) n_y + c(un_z - wn_x) \\ (\gamma-1)un_y - cn_z \\ (\gamma-1)vn_y \\ (\gamma-1)wn_y + cn_x \\ -(\gamma-1)n_y \end{pmatrix},$$

$$L3 = \begin{pmatrix} \left(c^2 - (\gamma-1)\frac{q^2}{2}\right) n_z + c(vn_x - un_y) \\ (\gamma-1)un_z + cn_y \\ (\gamma-1)vn_z - cn_x \\ (\gamma-1)wn_z \\ -(\gamma-1)n_z \end{pmatrix},$$

$$L4 = \begin{pmatrix} \frac{1}{2}\left((\gamma-1)\frac{q^2}{2} - cq_n\right) \\ -\frac{1}{2}\left((\gamma-1)u - cn_x\right) \\ -\frac{1}{2}\left((\gamma-1)v - cn_y\right) \\ -\frac{1}{2}\left((\gamma-1)w - cn_z\right) \\ \frac{1}{2}(\gamma-1) \end{pmatrix}, \quad L5 = \begin{pmatrix} \frac{1}{2}\left((\gamma-1)\frac{q^2}{2} + cq_n\right) \\ -\frac{1}{2}\left((\gamma-1)u + cn_x\right) \\ -\frac{1}{2}\left((\gamma-1)v + cn_y\right) \\ -\frac{1}{2}\left((\gamma-1)w + cn_z\right) \\ \frac{1}{2}(\gamma-1) \end{pmatrix}.$$

$q^2 = u^2 + v^2 + w^2$ and H is the stagnation enthalpy.

Viscous

Writing each quantity as the sum of a steady uniform value and some perturbation, the matrix B is obtained from the linearisation of perturbations about the uniform flow with the assumptions mentioned in section 3.3 and is given by

$$B = (C1|C2|C3|C4|C5)$$

with

$$C1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{-\bar{p}\gamma}{\bar{\rho}^2}\frac{\frac{\mu}{Pr_l}+\frac{\mu_t}{Pr_t}}{(\gamma-1)}n_* \end{pmatrix}, \quad C2 = \begin{pmatrix} 0 \\ (\mu+\mu_t)\left(\frac{4}{3}n_x l_x + n_y l_y + n_z l_z\right) \\ 0 \\ 0 \\ \bar{u}(\mu+\mu_t)\left(\frac{4}{3}n_x l_x + n_y l_y + n_z l_z\right) \end{pmatrix},$$

$$C3 = \begin{pmatrix} 0 \\ 0 \\ (\mu+\mu_t)\left(\frac{4}{3}n_y l_y + n_x l_x + n_z l_z\right) \\ 0 \\ \bar{v}(\mu+\mu_t)\left(\frac{4}{3}n_y l_y + n_x l_x + n_z l_z\right) \end{pmatrix},$$

$$C4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \bar{w}(\mu+\mu_t)\left(\frac{4}{3}n_z l_z + n_x l_x + n_y l_y\right) \\ \bar{w}(\mu+\mu_t)\left(\frac{4}{3}n_z l_z + n_x l_x + n_y l_y\right) \end{pmatrix}, \quad C5 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\gamma}{\bar{\rho}}\frac{\frac{\mu}{Pr_l}+\frac{\mu_t}{Pr_t}}{(\gamma-1)}n_* \end{pmatrix},$$

with $n_* = n_x l_x + n_y l_y + n_z l_z$.

3.5.2 Implementation

The 5×5 block-Jacobi preconditioner is computed and inverted for each node before the first stage of each time step. The residual vector R is then multiplied by P at each stage of the multistage Runge-Kutta scheme.

To prevent singularities at stagnation points, the matrix preconditioner incorporates an entropy fix which is different from the van Leer entropy fix [81] used in the numerical dissipation, as it does not sufficiently limit the time step. The more severe Harten entropy fix [31] is used with the minimum of the bounding parabola equal to one eighth the speed of the sound.

Chapter 4

Stability Analysis

The purpose of this chapter is to examine and evaluate the limits of the stability region of the scheme. Defining a suitable quantity, the "energy", the way to proceed consists of showing that, considering the semi-discrete equations, this "energy" is monotonically decreasing, giving a sufficient stability condition, and considering the fully discrete equations, to evaluate the limits of the stability region.

The stability analysis is achieved using the set of symmetrising variables of Gustafsson and Sundstrom [29] and Abarbanel and Gottlieb [1], which yields an equation in which the sub-matrices are all symmetric. Consequently, a first step is to prove that the resulting analysis remains valid for the conservative variables. Starting by considering periodic b.c.'s, the influence of a slip boundary condition is then investigated.

4.1 The equivalent symmetrised problem

Linearising with respect to perturbations to a uniform flow, the 3D Euler equations in Cartesian coordinates are

$$\frac{\partial \mathcal{Q}}{\partial t} + A^x \frac{\partial \mathcal{Q}}{\partial x} + A^y \frac{\partial \mathcal{Q}}{\partial y} + A^z \frac{\partial \mathcal{Q}}{\partial z} = 0, \tag{4.1}$$

where A^x, A^y, A^z, are the uniform inviscid flux Jacobian listed in detail in Appendix A. The transformation between the conservative variables and the sym-

metrising variables is accomplished by the matrix [29, 1]

$$L = \begin{bmatrix} \sqrt{\gamma}\frac{\rho}{c} & 0 & 0 & 0 & 0 \\ \sqrt{\gamma}\frac{\rho u}{c} & \rho & 0 & 0 & 0 \\ \sqrt{\gamma}\frac{\rho v}{c} & 0 & \rho & 0 & 0 \\ \sqrt{\gamma}\frac{\rho w}{c} & 0 & 0 & \rho & 0 \\ \sqrt{\gamma}\frac{\rho E}{c} & \rho u & \rho v & \rho w & \sqrt{\frac{\gamma}{\gamma-1}}\frac{p}{c} \end{bmatrix}.$$

Thus, if $\tilde{\mathcal{Q}}$ is the new set of variables, $\mathcal{Q} = L\tilde{\mathcal{Q}}$ and $\tilde{\mathcal{Q}} = L^{-1}\mathcal{Q}$, then equation (4.1) becomes

$$\frac{\partial \tilde{\mathcal{Q}}}{\partial t} + \tilde{A}^x \frac{\partial \tilde{\mathcal{Q}}}{\partial x} + \tilde{A}^y \frac{\partial \tilde{\mathcal{Q}}}{\partial y} + \tilde{A}^z \frac{\partial \tilde{\mathcal{Q}}}{\partial z} = 0, \tag{4.2}$$

where $\tilde{A}^x = L^{-1}A^xL$, $\tilde{A}^y = L^{-1}A^yL$, $\tilde{A}^z = L^{-1}A^zL$.

Using a first order upwinding scheme on a Cartesian grid, the semi discrete equation can then be written

$$\begin{aligned} \frac{d\tilde{Q}}{dt} &= -\frac{\tilde{A}^x}{2\Delta x}\delta_{2x}\tilde{Q}_j + \frac{|\tilde{A}^x|}{2\Delta x}\delta_x^2\tilde{Q}_j - \frac{\tilde{A}^y}{2\Delta y}\delta_{2y}\tilde{Q}_j \\ &\quad + \frac{|\tilde{A}^y|}{2\Delta y}\delta_y^2\tilde{Q}_j - \frac{\tilde{A}^z}{2\Delta z}\delta_{2z}\tilde{Q}_j + \frac{|\tilde{A}^z|}{2\Delta z}\delta_z^2\tilde{Q}_j, \end{aligned} \tag{4.3}$$

with $|\tilde{A}^x| = |L^{-1}\tilde{A}^xL|, |\tilde{A}^y| = |L^{-1}\tilde{A}^yL|, |\tilde{A}^z| = |L^{-1}\tilde{A}^zL|$, and where in one-dimension the δ_{2x} and δ_x^2 operators are defined as

$$\begin{aligned} \delta_{2x}Q_j &= Q_{j+1} - Q_{j-1} \\ \delta_x^2 Q_j &= Q_{j+1} - 2Q_j + Q_{j-1}. \end{aligned}$$

Since $\tilde{A}^x$ and A^x are similar matrices, they have the same eigenvalues; only the eigenvectors are different. The same comment can be made for $\tilde{A}^y$, and for $\tilde{A}^z$. Thus,

$$|\tilde{A}^x| = |L^{-1}A^xL| = L^{-1}T^x|\Lambda^x|T^{x\ -1}L,$$

$$|\tilde{A}^y| = |L^{-1}A^yL| = L^{-1}T^y|\Lambda^y|T^{y\ -1}L,$$

$$|\tilde{A}^z| = |L^{-1}A^zL| = L^{-1}T^z|\Lambda^z|T^{z\ -1}L,$$

where T^α is the matrix of the right eigenvectors of A^α, and Λ^α the corresponding diagonal matrix of eigenvalues.

Let us now discretise first the equation (4.1), and then transform the resulting discretisation in order to use the symmetrising set of variables. The semi-discrete equation on the same Cartesian grid is

$$\begin{aligned}\frac{dQ}{dt} &= -\frac{A^x}{2\Delta x}\delta_{2x}Q_j + \frac{|A^x|}{2\Delta x}\delta_x^2 Q_j - \frac{A^y}{2\Delta y}\delta_{2y}Q_j \\ &\quad + \frac{|A^y|}{2\Delta y}\delta_y^2 Q_j - \frac{A^z}{2\Delta z}\delta_{2z}Q_j + \frac{|A^z|}{2\Delta z}\delta_z^2 Q_j, \end{aligned} \tag{4.4}$$

which can also be written as

$$\begin{aligned}\frac{dQ}{dt} &= -\frac{A^x}{2\Delta x}\delta_{2x}Q_j + \frac{T^x|\Lambda^x|T^{x\ -1}}{2\Delta x}\delta_x^2 Q_j - \frac{A^y}{2\Delta y}\delta_{2y}Q_j \\ &\quad + \frac{T^y|\Lambda^y|T^{y\ -1}}{2\Delta y}\delta_y^2 Q_j - \frac{A^z}{2\Delta z}\delta_{2z}Q_j + \frac{T^z|\Lambda^z|T^{z\ -1}}{2\Delta z}\delta_z^2 Q_j. \end{aligned} \tag{4.5}$$

Transforming now to the symmetrising variables, one gets

$$\begin{aligned}\frac{d\tilde{Q}}{dt} &= -\frac{L^{-1}A^xL}{2\Delta x}\delta_{2x}\tilde{Q}_j + \frac{L^{-1}T^x|\Lambda^x|T^{x\ -1}L}{2\Delta x}\delta_x^2\tilde{Q}_j - \frac{L^{-1}A^yL}{2\Delta y}\delta_{2y}\tilde{Q}_j \\ &\quad + \frac{L^{-1}T^y|\Lambda^y|T^{y\ -1}L}{2\Delta y}\delta_y^2\tilde{Q}_j - \frac{L^{-1}A^zL}{2\Delta z}\delta_{2z}\tilde{Q}_j + \frac{L^{-1}T^z|\Lambda^z|T^{z\ -1}L}{2\Delta z}\delta_z^2\tilde{Q}_j. \end{aligned} \tag{4.6}$$

It appears clearly that both discretisations are equivalent. Consequently, the stability analysis can be done using the symmetrising variables, knowing that it will be also valid for the discretisation using the conservative variables. The same argument can be generalised for discretisations on unstructured grids.

4.2 Semi-discrete equations

In this section, as in the next one, periodic b.c.'s are considered. The domain has then only interior grid points, and consequently, the analysis does not include any boundary treatment.

The starting point is the Euler equations, which may be expressed as

$$\frac{\partial \mathcal{Q}}{\partial t} + \frac{\partial}{\partial x_i}\mathcal{F}(\mathbf{e_i}, \mathcal{Q}) = 0. \tag{4.7}$$

$\mathcal{Q}(\mathbf{x})$ is the vector of conserved variables. Using the same notation and integration rules as in section 3.1, the semi-discrete equivalent to equation (4.7) is

$$V_j\frac{dQ_j}{dt} = -\sum_{i\in E_j} F_{ij} \triangle s_{ij} \quad \forall j, \tag{4.8}$$

where F_{ij} is the numerical flux in the direction $\mathbf{n}_{ij}$ associated with an edge (i,j). The evaluation of F is achieved as explained in section 2.2.1, and with a first order scheme, equation (4.8) becomes

$$V_j \frac{dQ_j}{dt} = -\sum_{i \in E_j} \left(\frac{1}{2}(\mathcal{F}_{ij}(Q_i) + \mathcal{F}_{ij}(Q_j)) - \frac{1}{2}|A_{ij}|(Q_i - Q_j) \right) \triangle s_{ij},$$

where $A_{ij} = \dfrac{\partial \mathcal{F}}{\partial \mathcal{Q}}$.

Linearising locally and transforming to the symmetrising variables $\tilde{Q}$, the resulting equation is

$$V_j \frac{d\tilde{Q}_j}{dt} = -\sum_{i \in E_j} \left(\frac{1}{2}\tilde{A}_{ij}(\tilde{Q}_i + \tilde{Q}_j) - \frac{1}{2}|\tilde{A}_{ij}|(\tilde{Q}_i - \tilde{Q}_j) \right) \triangle s_{ij}, \tag{4.9}$$

where $\tilde{A}_{ij} = \tilde{A}^x \mathbf{n}_x + \tilde{A}^y \mathbf{n}_y + \tilde{A}^z \mathbf{n}_z$ is a symmetric matrix, and the matrices $\tilde{A}^x, \tilde{A}^y, \tilde{A}^z$ are related to the edge (i, j).

Considering now the whole mesh, and calling $\tilde{U}$ the vector of unknowns, at a given time level, of dimension $5N$, where N is the number of nodes, the system of o.d.e.'s may be written as

$$V \frac{d\tilde{U}}{dt} = -\left(\mathbb{A}\tilde{U} + \mathbb{D}\tilde{U} \right).$$

$\mathbb{A}$ and $\mathbb{D}$ are $N \times N$ block matrices, in which each block is a 5×5 matrix. Their expressions are

$$\mathbb{A} = \sum_{edges} \mathbb{A}_{ij} = \sum_{edges} \begin{bmatrix} & 0 & +\frac{1}{2}\tilde{A}_{ij} \triangle s_{ij} & \\ & & & \\ & -\frac{1}{2}\tilde{A}_{ij} \triangle s_{ij} & 0 & \end{bmatrix}, \tag{4.10}$$

and

$$\mathbb{D} = \sum_{edges} \mathbb{D}_{ij} = \sum_{edges} \begin{bmatrix} & \frac{1}{2}|\tilde{A}_{ij}| \triangle s_{ij} & -\frac{1}{2}|\tilde{A}_{ij}| \triangle s_{ij} & \\ & & & \\ & -\frac{1}{2}|\tilde{A}_{ij}| \triangle s_{ij} & \frac{1}{2}|\tilde{A}_{ij}| \triangle s_{ij} & \end{bmatrix}, \tag{4.11}$$

where an edge connects node j to node i. In these expressions the four block elements correspond to entries (j,j), (j,i), (i,j) and (i,i); all other elements are zero.

The diagonal blocks in $\mathbb{A}$ are zero because $\sum_{i\in E_j} \frac{1}{2}\mathcal{F}_{ij}(Q_j) = 0$ due to the closed control volume around node j. In addition, $\mathbb{A}$ is anti-symmetric because $\tilde{A}$ is symmetric and hence, the (i,j) element is the opposite of the (j,i) one.

Concerning the matrix $\mathbb{D}$ a first remark has to be made: if $\tilde{A}$ is a symmetric matrix, then $|\tilde{A}|$ is symmetric. Indeed, since $\tilde{A}$ is symmetric, it is diagonalisable by an orthogonal similarity transformation $\tilde{A} = T\Lambda T^T$, $T^T T = I$, which implies that $|\tilde{A}|$ is symmetric.

Looking now at the full matrix $\mathbb{D}$, it is clear that it is also a symmetric matrix, since the contribution of node j to node i and vice versa is the same. In addition, considering equation (4.11) for any complex vector W of length $5N$ and its Hermitian W^*,

$$W^*\mathbb{D}W = \frac{1}{2}\sum_{edges}(w_i - w_j)^*|\tilde{A}_{ij}| \triangle s_{ij}(w_i - w_j).$$

The matrices $|\tilde{A}_{ij}|$ are real positive definite symmetric, and the quantities $\triangle s_{ij}$ are positive real numbers. Consequently, $W^*\mathbb{D}W$ is a sum of non-negative real numbers. Therefore $\mathbb{D}$ is positive semi-definite.

The pre-conditioned semi-discrete equation may be written as

$$\mathbb{P}^{-1}\frac{d\tilde{U}}{dt} = -(\mathbb{A}+\mathbb{D})\tilde{U} \tag{4.12}$$

where $\mathbb{P}^{-1}$ is a block diagonal matrix whose dimension is the same as $\mathbb{A}$ and $\mathbb{D}$, and with the j^{th} block diagonal defined as $\frac{1}{2}\sum_{i\in E_j}|\tilde{A}_{ij}| \triangle s_{ij}$. The application of an entropy fix [31] to the eigenvalues ensures that each block diagonal is symmetric positive definite, and so is $\mathbb{P}^{-1}$.

Defining now the "energy" as $E = \frac{1}{2}\tilde{U}^*\mathbb{P}^{-1}\tilde{U}$,

$$\begin{aligned}
\frac{dE}{dt} &= \frac{1}{2}\left(\frac{d\tilde{U}^*}{dt}\mathbb{P}^{-1}\tilde{U} + \tilde{U}^*\mathbb{P}^{-1}\frac{d\tilde{U}}{dt}\right) \\
&= \frac{1}{2}\left(-\tilde{U}^*(\mathbb{A}+\mathbb{D})^*\tilde{U} - \tilde{U}^*(\mathbb{A}+\mathbb{D})\tilde{U}\right) \\
&= -\tilde{U}^*\mathbb{D}\tilde{U} \leq 0.
\end{aligned}$$

From this result, the energy is clearly non-increasing. Since $\mathbb{P}^{-1}$ is symmetric and positive definite, this in turn implies stability for the semi-discrete equation.

4.3 Fully discrete equations

Now that a sufficient stability condition has been obtained, it is necessary to investigate and give local timestep stability limits. Starting with equation (4.12) and defining $\mathbb{C} = -(\mathbb{A} + \mathbb{D})$, this is rearranged by setting a new variable $W = \mathbb{P}^{-1/2}\tilde{U}$ to become

$$\mathbb{P}^{-1/2}\frac{dW}{dt} = \mathbb{C}\mathbb{P}^{1/2}W$$

$$\Longleftrightarrow \frac{dW}{dt} = \mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}W.$$

Using Runge-Kutta time integration, the fully discrete equations are

$$W^{(n+1)} = L\left(k\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right)W^{(n)}, \tag{4.13}$$

where $L(z)$ is the Runge-Kutta polynomial with stability region S, and k the global timestep. (4.13) can also be written

$$W^{(n+1)} = \left(L\left(k\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right)\right)^n W^{(0)}.$$

The necessary and sufficient condition for absolute stability as n $\rightarrow \infty$ is that there are no discrete solutions which grow exponentially with n. This requires that $|L\left(k\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right)| \leq 1$, or equivalently that the matrix eigenvalues lie in S. Theoretically, this is enough to get asymptotic convergence, but it is not sufficient in practice because if the matrix is not normal (i.e. its eigenvectors are non-orthogonal), it allows the possibility of a very large transient growth and can lead to arithmetic overflow. In the analysis, it is then important to find sufficient conditions to eliminate this possibility. Ideally, one would hope to prove strong stability, which using the L_2 vector norm is expressed as

$$||W^{(n)}|| \leq \gamma||W^{(0)}||,$$

where γ is a constant not only independent of n but is also a uniform bound applying to all matrices in the family of spatial discretisations for different mesh spacings h and with the timestep k being a function of h. However, in practice it is often not possible to prove strong stability. Indeed what can be more easily proved is a weaker form of stability called algebraic stability [67, 45, 41, 28]. This allows, at worst, a linear growth in the transient solution of the form

$$||W^{(n)}|| \leq \gamma n ||W^{(0)}||,$$

where γ is a constant. As shown in [28], a sufficient condition for algebraic stability is that

$$\tau\left(k\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right) \subset S,$$

where the field of values τ is defined as

$$\tau\left(k\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right) = \left\{ k\frac{W^*\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}W}{W^*W} : W \neq 0 \right\}.$$

The first step is to prove that when $k = 1$ the field of values is bounded by a unit circle centred on $z = -1$. Writing $V = \mathbb{P}^{1/2}W$, consider

$$\tau\left(\mathbb{P}^{1/2}\mathbb{C}\mathbb{P}^{1/2}\right) = \left\{ \frac{V^*\mathbb{C}V}{V^*\mathbb{P}^{-1}V} : V \neq 0 \right\} = \left\{ -\frac{V^*(\mathbb{A}+\mathbb{D})V}{V^*\mathbb{P}^{-1}V} : V \neq 0 \right\}.$$

Looking at the contributions from a single edge (i, j), and diagonalising $\frac{1}{2}\tilde{A}_{ij} \triangle s_{ij}$ as $\frac{1}{2}\tilde{A}_{ij} \triangle s_{ij} = H^T \Lambda H$ we obtain

$$V^*\mathbb{P}_{ij}^{-1}V = (HV)^* \begin{bmatrix} |\Lambda| & 0 \\ 0 & |\Lambda| \end{bmatrix} HV = \sum_m \left|\lambda^{(m)}\right| \left(|v_i^{(m)}|^2 + |v_j^{(m)}|^2\right),$$

$$V^*\mathbb{D}_{ij}V = (HV)^* \begin{bmatrix} |\Lambda| & -|\Lambda| \\ -|\Lambda| & |\Lambda| \end{bmatrix} HV = \sum_m - \left|\lambda^{(m)}\right| \left(v_i^{(m)*}v_j^{(m)} + v_j^{(m)*}v_i^{(m)}\right)$$
$$+ \left|\lambda^{(m)}\right| \left(|v_i^{(m)}|^2 + |v_j^{(m)}|^2\right),$$

and

$$V^* \mathbb{A}_{ij} V = (HV)^* \begin{bmatrix} 0 & +\Lambda \\ -\Lambda & 0 \end{bmatrix} HV = \sum_m \lambda^{(m)} \left(v_i^{(m)*} v_j^{(m)} - v_j^{(m)*} v_i^{(m)} \right),$$

where (m) denotes the m-th characteristic (i.e. the m^{th} component of the vector HV). Hence

$$\begin{aligned} \left| V^*(-\mathbb{A}_{ij} - \mathbb{D}_{ij})V + V^* \mathbb{P}_{ij}^{-1} V \right| &= \sum_m \left| \left(-\lambda^{(m)} + |\lambda^{(m)}| \right) v_i^{(m)*} v_j^{(m)} \right. \\ &\qquad \left. + \left(\lambda^{(m)} + |\lambda^{(m)}| \right) v_j^{(m)*} v_i^{(m)} \right| \\ &\leq \sum_m 2|\lambda^{(m)}| \; |v_i^{(m)}| \; |v_j^{(m)}| \\ &\leq \sum_m |\lambda^{(m)}| (|v_i^{(m)}|^2 + |v_j^{(m)}|^2), \end{aligned}$$

$$\Longrightarrow \quad \left| V^*(-\mathbb{A}_{ij} - \mathbb{D}_{ij})V + V^* \mathbb{P}_{ij}^{-1} V \right| \leq V^* \mathbb{P}_{ij}^{-1} V \quad .$$

Summing over all edges then gives the desired result.

$$\left| V^* \sum_{edges} (-\mathbb{A}_{ij} - \mathbb{D}_{ij})V + V^* \sum_{edges} \mathbb{P}_{ij}^{-1} V \right| \leq V^* \sum_{edges} \mathbb{P}_{ij}^{-1} V \; ,$$

$$\Longrightarrow \quad \left| \frac{V^*(-\mathbb{A} - \mathbb{D})V}{V^* \mathbb{P}^{-1} V} + 1 \right| \leq 1.$$

When $k \neq 1$ the field of values is bounded by a circle of radius k centred on $z = -k$. Choosing the largest such circle lying inside the stability region S gives a timestep k which is guaranteed to be algebraically stable. Numerical results will later establish that this is close to being a necessary condition for stability as well as sufficient.

4.4 Slip boundary condition

Thus far, only periodic b.c.'s have been considered, but the analysis can be extended to include the effect of an inviscid flow tangency condition at a solid wall.

Taking the wall to be flat, for simplicity, the tangency requirement can be written as

$$NQ = 0,$$

where N is a symmetric projection matrix which obtains the normal component of the momentum at the wall nodes. A general vector Q can be decomposed into components $Q_{\|}$ and $Q_{\perp}$ defined by

$$Q = Q_{\perp} + Q_{\|}, \quad Q_{\perp} = NQ, \quad NQ_{\|} = 0.$$

As shown in section 3.4, the modified form of the preconditioner, expressed in symmetric variables, is

$$\left[(I-N)\,\mathbb{P}^{-1} + N\right]\frac{d\tilde{Q}}{dt} = (I-N)\,\mathbb{C}\tilde{Q},$$

where $N = T^{-1}ST$. It is re-written in symmetrised form as

$$\left[(I-N)\,\mathbb{P}^{-1}\,(I-N) + N\right]\frac{d\tilde{Q}}{dt} = (I-N)\,\mathbb{C}\,(I-N)\,\tilde{Q}.$$

This form does not change the tangency condition and consequently is equivalent to the previous one; $\frac{d\tilde{Q}_{\perp}}{dt} = 0$ and the flow tangency $\tilde{Q}_{\perp} = 0$ is undisturbed. Applying the stability analysis we obtain

$$\begin{aligned} |V^*(I-N)\mathbb{C}(I-N)V \quad &+ \quad V^*\left[(I-N)\,\mathbb{P}^{-1}\,(I-N) + N\right]V\big| \\ &= \left|V_{\|}^*\mathbb{C}V_{\|} + V_{\|}^*\mathbb{P}^{-1}V_{\|} + V_{\perp}^*V_{\perp}\right| \\ &\leq V_{\|}^*\mathbb{P}^{-1}V_{\|} + V_{\perp}^*V_{\perp} \\ &= V^*\left[(I-N)\,\mathbb{P}^{-1}\,(I-N) + N\right]V \end{aligned}$$

and hence

$$\left|\frac{V^*(I-N)\mathbb{C}(I-N)V}{V^*\left[(I-N)\,\mathbb{P}^{-1}\,(I-N) + N\right]V} + 1\right| \leq 1$$

and the field of values lies within the unit circle centred on $z = -1$, as before.

4.5 Some Runge-Kutta stability curves

If one was solving the system of linearised equations using a first order upwinding, as analysed above, a simple Euler forward time integration with a global time step equal to 1 would be enough, since the stability regions would perfectly coincide. Unfortunately, the equations which are actually solved are non-linear, and as mentioned in section 2.2.1 a second order numerical dissipation is used, which has the consequence of "flattening out" the boundary of the field of values along the imaginary axis (the "flattened" boundary is compared with the unit circle in Fig. 4.1). To cope with this new shape, a Runge-Kutta time integration has to be used, giving a wider stability region.

Let us consider a system of o.d.e.'s of the form

$$\frac{dQ}{dt} = CQ,$$

where C is a real square matrix. Using a Runge-Kutta time integration with timestep k,

$$Q^{(n+1)} = L(kC)Q^{(n)},$$

where $L(z) = \sum_{m=0}^{p} a_m z^m$ ($a_0 = a_1 = 1$, $a_p \neq 0$) is the Runge-Kutta polynomial function with stability region S. The following figures show, for four popular multistage integration schemes, the stability region S within which $|L| \leq 1$. They also show the largest circle which lies inside S and which corresponds to the sufficient stability limits of the scheme being analysed. Its radius r_c equals the maximum timestep for which the analysis gives a sufficient condition for algebraic stability.

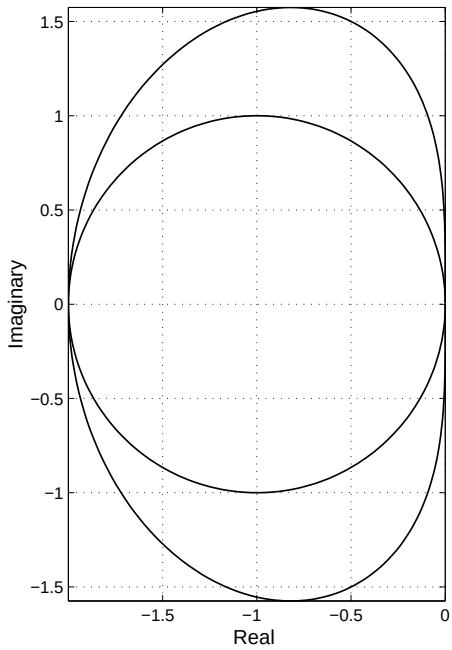


Figure 4.1 Effect of a second order dissipation on the field of values for the one dimensional linear convection equation with a Courant number set to 1.

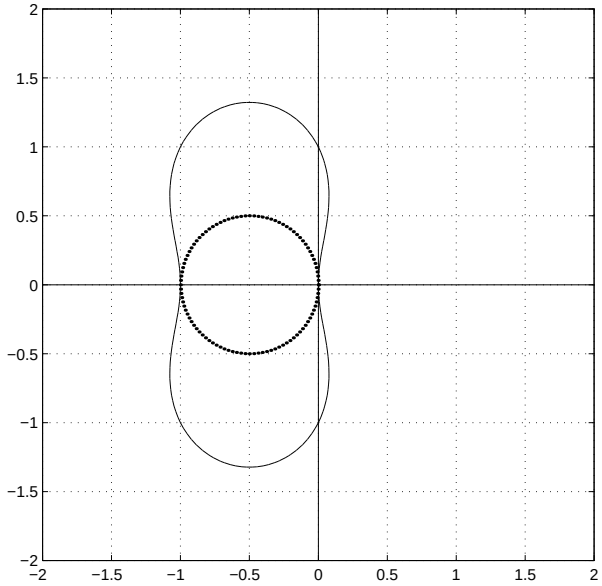


Figure 4.2 Predictor-corrector. $r_c = 0.5$.

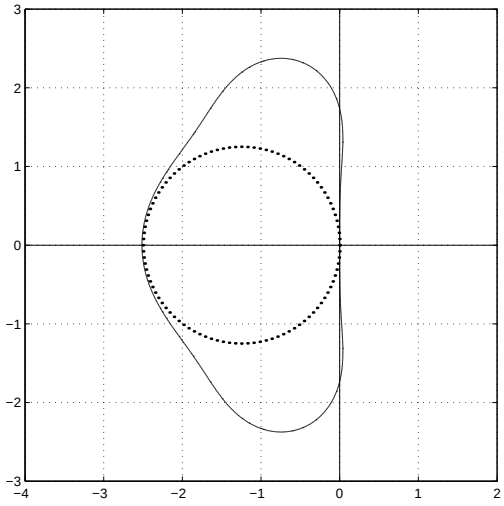


Figure 4.3 Three-stage scheme. $r_c = 1.25$.

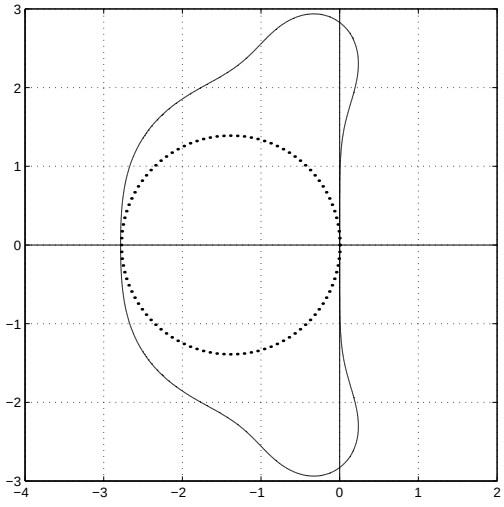


Figure 4.4 Four-stage scheme. $r_c = 1.39$.

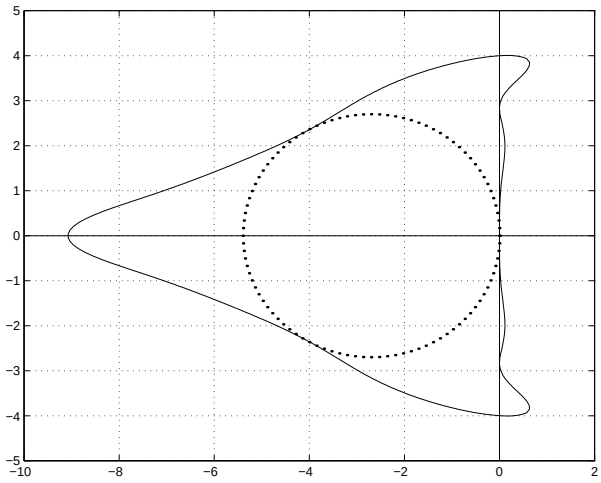


Figure 4.5 Five-stage scheme. $r_c = 2.7$.

Chapter 5

Preconditioner for low Mach number flows

Now that an efficient preconditioned multigrid method for both inviscid and viscous flows has been introduced, the aim is to solve problems for a range of flow conditions from nearly incompressible to transonic and supersonic. However, at low Mach number the disparity between the acoustic and convective wave speeds cannot be adequately handled by the current approach, and a slowdown of the convergence is observed. Furthermore, the numerical solution produced is often of poor quality with significant errors in the pressure distribution due to the relative scaling of the different numerical smoothing terms. To address these difficulties, a low Mach number preconditioner can be incorporated into the numerical dissipation and hence into the block-Jacobi preconditioner.

The first part of this chapter describes the new discretisation that the introduction of a low Mach number preconditioner implies. A second part will investigate the influence of this preconditioner on the boundary conditions: when the solution has almost converged to the steady state, and hence only low frequency waves remain, the analysis will determine whether an exponential decay of the amplitude of these waves can be expected. Finally, the extension of the stability analysis presented in the previous chapter is presented, giving a complete study of the method used throughout this work.

5.1 Preconditioned numerical dissipation

5.1.1 1D preliminary

The part of the equation concerned by the preconditioning is the inviscid part. Consequently, only the Euler equations are here considered and preconditioned for low Mach number applications by an invertible matrix Γ, which is expressed in symmetrised variables, to become

$$\frac{\partial \tilde{Q}}{\partial t} + \Gamma \tilde{A} \frac{\partial \tilde{Q}}{\partial x} = 0 \ ,$$

which in a semi discrete form is written as

$$\frac{d\tilde{Q}}{dt} + \frac{1}{2}\Gamma \tilde{A} \delta_{2x} \tilde{Q} - \frac{1}{2}|\Gamma \tilde{A}| \delta_x^2 \tilde{Q} = 0 \ , \tag{5.1}$$

and re-written as

$$\Gamma^{-1}\frac{d\tilde{Q}}{dt} + \frac{1}{2}\tilde{A} \delta_{2x} \tilde{Q} - \frac{1}{2}\Gamma^{-1}|\Gamma \tilde{A}| \delta_x^2 \tilde{Q} = 0 \ . \tag{5.2}$$

Using M and N the transformation matrices from primitive variable to conservative variables, and from symmetrized variables to primitive variables, respectively, (5.2) becomes then the conservative semi discrete equations for low Mach number flow and read as

$$MN\ \Gamma^{-1}\ N^{-1}M^{-1}\frac{dQ}{dt} + \frac{1}{2}A\delta_{2x}Q - \frac{1}{2}MN\Gamma^{-1}|\Gamma \tilde{A}|N^{-1}M^{-1}\delta_x^2 Q = 0 \ . \tag{5.3}$$

5.1.2 3D generalisation

Following the 1D approach, the flux function defined in equation (2.12) is modified to incorporate the low Mach number preconditioner, becoming

$$\begin{aligned} F_{ij}^I &= \frac{1}{2}\left(\mathcal{F}_{ij}^I(Q_i) + \mathcal{F}_{ij}^I(Q_j)\right. \\ &- \left. M_{ij}N_{ij}\Gamma_{ij}^{-1}|\Gamma_{ij}\tilde{A}_{ij}|N_{ij}^{-1}M_{ij}^{-1}\left(-\frac{1}{3}(1-\Psi)(\hat{L}_i^{lp}(Q) - \hat{L}_j^{lp}(Q)) + \Psi(Q_i - Q_j)\right)\right) \ . \end{aligned}$$

In this formulation, only the dissipation has changed, which makes the implementation attractive because it does not require any change of variables in the current

code.

In [**?**] Lee gives a broad overview of the current state of preconditioning. As demonstrated in [86], and because of their highly non-normal feature for low Mach number, many local preconditioners can transiently amplify perturbations by a factor of $1/M$ as $M \to 0$. Taking this fact into account, the preconditioner used here is the same as that used by Darmofal and Siu [24], for which the transient-growth can be limited by a careful treatment of the parameter ϵ. This preconditioner is

$$\Gamma = \begin{bmatrix} \epsilon & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

It is identical to the preconditioner of Weiss and Smith [86], and is expressed in the symmetrised variables $\left[dp/\rho c, du, dv, dw, dp - c^2 d\rho\right]^T$. ϵ is a free parameter whose role is to equilibrate the eigenvalues. As it can be seen further down, this preconditioner alters only the eigenvalues relative to the acoustic waves. At a Mach number of unity, the sign of one of these two eigenvalues must change to give a identical sign to all eigenvalues for supersonic flows, since information cannot flow upstream. The stiffness at a sonic point is not removed with this preconditioner: a singularity shows up and the eigenvalues cannot be isotropic. This preconditioner is only for use at low Mach number, and consequently, switched off when not appropriate: it reduces to the identity matrix, and the original system of equations is recovered.

Choosing $\epsilon = O(M^2)$ ensures that the convective and acoustic wave speeds are of a similar magnitude, proportional to the flow speed. Very often, it is required that ϵ be greater than some multiple of the square of the freestream Mach number [78, 79, 66]. Although this approach has proved to work well, it cannot be used for internal flows where the freestream Mach number is usually unknown. Consequently, a different approach is followed here. Looping over the edges, the biggest Mach number between two nodes connected by an edge is evaluated and kept for both

nodes. Repeating the procedure four times defines small regions with a common maximum value. This evaluation stays local, and provides a smoother behaviour of the limiter than if this was only based on the nodal Mach number. In the end, the determination of ϵ is implemented as follows:

$$\epsilon = min \ \left[1, \eta M_{max}^2\right] \ ,$$

where η is a free parameter set to 3.0 [66]. With this preconditioner, the following eigenvalues are obtained:

$$\begin{aligned} \lambda_1 &= \frac{1}{2}\left(1+\epsilon\right)Q - \frac{1}{2}\tau \\ \lambda_2 &= \frac{1}{2}\left(1+\epsilon\right)Q + \frac{1}{2}\tau \\ \lambda_3 &= \lambda_4 = \lambda_5 = Q \end{aligned}$$

with

$$\tau = \sqrt{\left(1-\epsilon\right)^2 Q^2 + 4\epsilon c^2} \ .$$

In the expression of $M N \Gamma^{-1} |\Gamma \tilde{A}| N^{-1} = M N \Gamma^{-1} L |\Lambda| L^{-1} N^{-1}$, $M N \Gamma^{-1} L$ is given by

$$M N \Gamma^{-1} L = \left(R_1 | R_2 | R_3 | R_4 | R_5\right)^T$$

where

$$R_1 = \begin{pmatrix} \frac{s^+}{2c\epsilon} \\ \frac{s^-}{2c\epsilon} \\ n_x \\ n_y \\ n_z \end{pmatrix} \qquad R_2 = \begin{pmatrix} \frac{us^+ - 2\,c^2 nx\,\epsilon}{2c\epsilon} \\ \frac{us^- + 2\,c^2 nx\,\epsilon}{2c\epsilon} \\ un_x \\ vn_y + cn_z \\ un_z + cn_y \end{pmatrix}$$

$$R_3 = \begin{pmatrix} \frac{vs^+ - 2\,c^2 ny\,\epsilon}{2c\epsilon} \\ \frac{vs^- + 2\,c^2 ny\,\epsilon}{2c\epsilon} \\ vn_x + cn_z \\ vn_y \\ vn_z - cn_x \end{pmatrix} \qquad R_4 = \begin{pmatrix} \frac{ws^+ - 2\,c^2 nz\,\epsilon}{2c\epsilon} \\ \frac{ws^- + 2\,c^2 nz\,\epsilon}{2c\epsilon} \\ wn_x - cn_y \\ wn_y - cn_x \\ wn_z \end{pmatrix}$$

$$R_5 = \begin{pmatrix} \frac{Hs^+ - 2\,c^2\,Q\,\epsilon}{2c\epsilon} \\ \frac{Hs^- + 2\,c^2\,Q\,\epsilon}{2c\epsilon} \\ \frac{1}{2}Q^2 n_x + c(vn_z - wny) \\ \frac{1}{2}Q^2 n_y + c(wn_x - unz) \\ \frac{1}{2}Q^2 n_z + c(un_y - vnx) \end{pmatrix}$$

with

$$\begin{aligned} s^+ &= \tau + (1-\epsilon)Q \\ s^- &= \tau - (1-\epsilon)Q \end{aligned}$$

and, $L^{-1}N^{-1}$ by

$$L^{-1}N^{-1} = (C_1|C_2|C_3|C_4|C_5)$$

where

$$C_1 = \begin{pmatrix} 0 \\ 0 \\ n_x \\ n_y \\ n_z \end{pmatrix} \qquad C_2 = \begin{pmatrix} \frac{-\rho s^- n_x}{2c\tau} \\ \frac{\rho s^+ n_x}{2c\tau} \\ 0 \\ \frac{-\rho n_z}{c} \\ \frac{\rho n_y}{c} \end{pmatrix}$$

$$C_3 = \begin{pmatrix} \frac{-\rho s^- n_y}{2c\tau} \\ \frac{\rho s^+ n_y}{2c\tau} \\ \frac{\rho n_z}{c} \\ 0 \\ \frac{-\rho n_x}{c} \end{pmatrix} \qquad C_4 = \begin{pmatrix} \frac{-\rho s^- n_z}{2c\tau} \\ \frac{\rho s^+ n_z}{2c\tau} \\ \frac{-\rho n_y}{c} \\ \frac{\rho n_x}{c} \\ 0 \end{pmatrix} \qquad C_5 = \begin{pmatrix} \frac{1}{c\tau} \\ \frac{1}{c\tau} \\ -\frac{n_x}{c^2} \\ -\frac{n_y}{c^2} \\ -\frac{n_z}{c^2} \end{pmatrix} .$$

Modification of the artificial dissipation implies automatically modification of the block-Jacobi preconditioner. The adjustment is straitforward, and only concerns the inviscid part. Thus equation (3.5) becomes

$$\begin{aligned} \left(P_j^I\right)^{-1} &= \frac{1}{2V_j}\left(\sum_{i\in E_j} M_{ij}N_{ij}\Gamma_{ij}^{-1}|\Gamma_{ij}\tilde{A}_{ij}|N_{ij}^{-1}M_{ij}^{-1} \triangle s_{ij}\right. \\ &+ \left.\sum_{k\in B_j} M_j N_j \Gamma_j^{-1}|\Gamma_j \tilde{A}_j|N_j^{-1}M_j^{-1} \triangle s_k\right). \end{aligned} \tag{5.4}$$

5.2 Boundary condition

When the solution has almost converged to the steady state, the residual is due to low frequency waves which propagate up and down the domain and are not affected by the numerical viscosity. These can only be dissipated through the interaction with the inflow and outflow boundary conditions. In general, when they arrive at one boundary, these waves are reflected and propagated in the other direction until they reach the other boundary, and so on. Ideally, one would like to have perfectly non-reflecting boundary conditions, absorbing these low frequency waves and resulting in a much faster convergence rate, but in two or three dimensions, these do not exist and consequently one only can expect an exponential decay of the amplitude of these waves. In [27] Giles has examined this process for the subsonic one-dimensional Euler equations by deriving the exact eigenmodes and eigenfrequencies of the initial boundary value problem and by determining the exponential decay rates for perturbations under different sets of boundary conditions. In a similar way the same analysis is done here, but this time, looking at the one-dimensional preconditioned Euler equations for low Mach number flows. Although it is not obvious to what extent the conclusions of the following model are valid for more general flows, it provides insight concerning the effect of a low Mach number preconditioner on the boundary conditions.

5.3 Analysis

When expressed in the symmetrised variables, the preconditioner reduces itself into a diagonal matrix. Consequently the analysis starts with the one-dimensional unsteady linearised Euler equations using symmetrised variables which are written as

$$\frac{\partial \tilde{Q}}{\partial t} + \tilde{A}\frac{\partial \tilde{Q}}{\partial x} = 0 \tag{5.5}$$

where

$$\tilde{A} = \begin{bmatrix} \bar{u} & \bar{c} & 0 \\ \bar{c} & \bar{u} & 0 \\ 0 & 0 & \bar{u} \end{bmatrix},$$

and $d\tilde{Q} = \left[\frac{dp'}{\bar{\rho}\bar{c}}, du', dp' - \bar{c}^2 d\rho'\right]^T$. ρ', u', p' are the perturbation density, velocity and pressure, and $\bar{\rho}, \bar{u}, \bar{p}$ are the steady uniform values.

To reduce propagation speed stiffness and to improve low Mach number accuracy, they are preconditioned and become

$$\frac{\partial \tilde{Q}}{\partial t} + \Gamma \tilde{A} \frac{\partial \tilde{Q}}{\partial x} = 0, \tag{5.6}$$

where

$$\Gamma = \begin{bmatrix} \epsilon & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Γ is the one dimensional version of the preconditioner due to Weiss and Smith [86]. Defining the transformation matrix M_p as $M_p = \frac{\partial Q_p}{\partial \tilde{Q}}$ where $Q_p = [\rho, u, p]^T$, (5.6) becomes

$$\frac{\partial Q_p}{\partial t} + M_p^{-1} \Gamma \tilde{A} M_p \frac{\partial Q_p}{\partial x} = 0. \tag{5.7}$$

The analysis is greatly simplified by defining the following non-dimensional variables

$$\rho^* = \rho/\bar{\rho}, \quad u^* = u/\bar{c}, \quad p^* = p/\bar{\rho}\bar{c}^2, \quad x^* = x/L, \quad t^* = t\bar{c}/L,$$

where $\bar{c} = [\gamma \bar{p}/\bar{\rho}]^{1/2}$ is the speed of sound. L is the physical length of the domain considered, so in the non-dimensional domain, the subsonic inflow is at $x = 0$ and the outflow is at $x = 1$. Omitting the asterisks for clarity, the resulting non-dimensional equation of (5.7) is

$$\frac{\partial Q_p}{\partial t} + A \frac{\partial Q_p}{\partial x} = 0, \tag{5.8}$$

with

$$A = \begin{bmatrix} M & \epsilon & M(\epsilon - 1) \\ 0 & M & 1 \\ 0 & \epsilon & M\epsilon \end{bmatrix}.$$

Notice that when $\epsilon = 1$, Γ reduces itself into the identity matrix, and equation (5.8) is the non-dimensional version of the unpreconditioned equations.

The boundary conditions for subsonic flow require two inflow quantities and one outflow quantity to be specified. The inflow boundary conditions can be expressed as

$$C_{in} Q_p(0,t) = 0, \tag{5.9}$$

where C_{in} is a 2×3 matrix depending on the specific choice of inflow conditions. Similarly, the single outflow boundary condition can be expressed as

$$C_{out} Q_p(1,t) = 0, \tag{5.10}$$

where C_{out} is a 1×3 matrix depending on the specific choice of outflow condition. Equations (5.8), (5.9), and (5.10) represent the initial boundary value problem, and an eigenmode of the initial boundary value problem is given by

$$Q_p = e^{-i\omega t} \left[\alpha_1 e^{i(\omega/\lambda_1)x}\ r_1 + \alpha_2 e^{i(\omega/\lambda_2)x}\ r_2 + \alpha_3 e^{i(\omega/\lambda_3)x}\ r_3 \right], \tag{5.11}$$

where the constants α_i are the strengths of each eigenmode, and the r_i and λ_i ($i = 1,2,3$) are the right eigenvectors and eigenvalues, respectively, of A. Symbolically, these are

$$\lambda_1 = \frac{1}{2}\left[(\epsilon+1)M + \tau\right], \quad \lambda_2 = \frac{1}{2}\left[(\epsilon+1)M - \tau\right], \quad \lambda_3 = M,$$

with $\tau = \sqrt{(1-\epsilon)^2 M^2 + 4\epsilon}$, and

$$r_1 = \left[1, -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon}, 1\right]^T,\ r_2 = \left[1, -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon}, 1\right]^T,\ r_3 = [1,0,0]^T.$$

The eigenfrequency ω and the constants α_i are determined by the boundary conditions. For the inflow boundary, substitution of equation (5.11) into equation (5.9) leads to

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0,$$

where

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \end{bmatrix} = C_{in}\left(r_1, r_2, r_3\right).$$

As described by Giles [27], a necessary condition for the well-posedness of the initial boundary value problem is that the incoming characteristics, α_1 and α_2, can be determined as functions of the outgoing characteristic, α_3. This requires that the 2×2 matrix

$$\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

is non-singular and therefore invertible.

For the outflow boundary, substitution of equation (5.11) into equation (5.10) leads to

$$\begin{pmatrix} b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0,$$

where

$$\begin{pmatrix} b_{31} & b_{32} & b_{33} \end{pmatrix} = C_{out}\left(e^{i(\omega/\lambda_1)}\ r_1, e^{i(\omega/\lambda_2)}\ r_2, e^{i(\omega/\lambda_3)}\ r_3\right).$$

In this case, well-posedness of the initial boundary value problem requires that the incoming characteristic, α_3, can be determined as a function of the outgoing characteristics α_1 and α_2. Thus, b_{33} must be non-zero. Combining the inflow and outflow boundary conditions leads to

$$B(\omega) \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0. \tag{5.12}$$

In order for a non-trivial solution of the initial boundary value problem to exist, a non-zero vector, $(\alpha_1, \alpha_2, \alpha_3)^T$, must exist which satisfies equation (5.12). This is possible for values of ω for which,

$$\det B(\omega) = 0. \tag{5.13}$$

Separating the eigenfrequency into its real and imaginary parts, $\omega = \omega_r + \omega_i$, the amplitude of the eigenmodes grows as $\exp(-\omega_i t)$. Thus, in order for the eigenmodes to decay, $\omega_i > 0$ for all possible values of ω.

5.4 Examples

5.4.1 Stagnation enthalpy and total pressure at inflow; pressure at outflow

The non linear boundary conditions are:

$$\begin{array}{ll} x = 0 & \left\{ \begin{array}{rcl} \frac{\gamma-1}{2}u'^2 + \frac{\gamma p'}{\rho'} & = & \frac{\gamma-1}{2}\bar{u}^2 + \frac{\gamma\bar{p}}{\bar{\rho}} \\ p'\left(1 + \frac{\gamma-1}{2\gamma}\frac{\rho' u'^2}{p'}\right) & = & \bar{p}\left(1 + \frac{\gamma-1}{2\gamma}\frac{\bar{\rho}\bar{u}^2}{\bar{p}}\right) \end{array} \right. \\ x = 1 & p' = \bar{p} \end{array}$$

where the prime quantities are the unsteady physical variables which are the sum of the steady state and unsteady perturbation variables.

The corresponding linearised non-dimensionalised equations are

$$\begin{array}{lll} x = 0 & \left[\begin{array}{ccc} \frac{M^2}{2} & M & \frac{\gamma}{(\gamma-1)} \\ -1 & (\gamma-1)M & \gamma \end{array} \right] \left[\begin{array}{c} \rho \\ u \\ p \end{array} \right] & = \quad 0 \\ x = 1 & \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right] \left[\begin{array}{c} \rho \\ u \\ p \end{array} \right] & = \quad 0 \end{array} \tag{5.14}$$

giving

$$\begin{aligned} C_{in} &= \left[\begin{array}{ccc} \frac{M^2}{2} & M & \frac{\gamma}{(\gamma-1)} \\ -1 & (\gamma-1)M & \gamma \end{array} \right], \\ C_{out} &= \left[\begin{array}{ccc} 0 & 0 & 1 \end{array} \right]. \end{aligned}$$

At $x = 0$ substitution of the eigenvector definitions into the eigenmode definition yields

$$\left[\begin{array}{c} \rho \\ u \\ p \end{array} \right] = e^{-i\omega t} \left[\begin{array}{ccc} 1 & 1 & 1 \\ -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon} & -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon} & 0 \\ 1 & 1 & 0 \end{array} \right] \left[\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{array} \right],$$

with $\tau = \sqrt{(1-\epsilon)^2 M^2 + 4\epsilon}$. Substitution of this equation into equation (5.14) ($x = 0$) produces the characteristic inflow boundary condition

$$e^{-i\omega t}\begin{bmatrix} \frac{M^2}{2} & M & \frac{\gamma}{(\gamma-1)} \\ -1 & (\gamma-1)M & \gamma \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \\ -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon} & -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon} & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0. \tag{5.15}$$

Similarly, at $x = 1$

$$\begin{bmatrix} \rho \\ u \\ p \end{bmatrix} = e^{-i\omega t}\begin{bmatrix} 1 & 1 & 1 \\ -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon} & -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon} & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha_1 \ \exp(i\omega/\lambda_1) \\ \alpha_2 \ \exp(i\omega/\lambda_2) \\ \alpha_3 \ \exp(i\omega/\lambda_3) \end{bmatrix},$$

and substitution into equation (5.14) ($x = 1$) produces the characteristic outflow boundary condition

$$e^{-i\omega t}\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \\ -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon} & -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon} & 0 \\ 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha_1 \ \exp(i\omega/\lambda_1) \\ \alpha_2 \ \exp(i\omega/\lambda_2) \\ \alpha_3 \ \exp(i\omega/\lambda_3) \end{bmatrix} = 0. \tag{5.16}$$

Together equations (5.15) and (5.16) define the matrix B

$$B = \begin{bmatrix} \frac{1}{2}\left(M^2 - \frac{M(M(\epsilon-1)-\tau)}{\epsilon}\right) + \frac{\gamma}{\gamma-1} & \frac{1}{2}\left(M^2 - \frac{M(M(\epsilon-1)+\tau)}{\epsilon}\right) + \frac{\gamma}{\gamma-1} & \frac{1}{2}M^2 \\ -1 - \frac{1}{2}\frac{(\gamma-1)M(M(\epsilon-1)-\tau)}{\epsilon} + \gamma & -1 - \frac{1}{2}\frac{(\gamma-1)M(M(\epsilon-1)+\tau)}{\epsilon} + \gamma & -1 \\ \exp(i\omega/\lambda_1) & \exp(i\omega/\lambda_2) & 0 \end{bmatrix}.$$

The eigenfrequencies are given by

$$\begin{aligned} \det B \ &= (2 + M^2\gamma - M^2)\ \big(\exp(i\omega/\lambda_1)(\epsilon M^2 - M^2 + \tau M - 2\epsilon) \\ &\qquad - \exp(i\omega/\lambda_2)(\epsilon M^2 - M^2 - \tau M - 2\epsilon)\big)/(4\epsilon) = 0. \end{aligned}$$

The eigenfrequencies which result in a zero determinant are

$$\begin{aligned} \omega_r &= \frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} 2n\pi \quad \forall n \\ \omega_i &= -\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1}\ \log\left(\frac{\epsilon M^2 - M^2 - \tau M - 2\epsilon}{\epsilon M^2 - M^2 + \tau M - 2\epsilon}\right) \end{aligned}$$

where

$$\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} = -\frac{1}{4\tau}\left((\epsilon+1)^2 M^2 - \tau^2\right).$$

Defining the decay rate σ_n

$$\sigma_n = -\mathrm{Im}(\omega) \ ,$$

for this example

$$\sigma_n = \frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1}\log\left(\frac{\epsilon M^2 - M^2 - \tau M - 2\epsilon}{\epsilon M^2 - M^2 + \tau M - 2\epsilon}\right),$$

The amplitude of the eigenmode grows, or decays, as $\exp(-\sigma t)$, so the requirement for all eigenmodes to decay is $\sigma_n > 0$ for every n. However, the quantity of interest is actually the decay per timestep $\sigma_n \Delta t$. Taking the CFL restriction into consideration, $\mathrm{CFL} = \frac{\lambda_{max}\Delta t}{\Delta x}$, one then is concerned about $\frac{\sigma_n}{\lambda_{max}}$.
The definition which is retained for ϵ is $\epsilon = \eta M^2$, where η is a free parameter. Consequently, as $M \to 0$, $\tau \to \sqrt{(1+4\eta)}M$,

$$\frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1} \to \frac{\eta M}{\sqrt{(1+4\eta)}} \quad > 0,$$

and

$$\frac{\epsilon M^2 - M^2 - \tau M - 2\epsilon}{\epsilon M^2 - M^2 + \tau M - 2\epsilon} \to \frac{4\eta^2}{((-1-2\eta) + \sqrt{(1+4\eta)})^2} \quad > 1.$$

$\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1}$ is $O(M)$, as λ_{max}, and consequently the decay per timestep remains finite. $\sigma_n > 0$ is satisfied $(\forall\eta)$ so any initial disturbance at $t = 0$ will decay exponentially.

5.4.2 Euler Riemann boundary conditions: Entropy and the appropriate Riemann invariant at inflow; the other Riemann invariant at outflow

$$x = 0 \quad \begin{cases} \frac{p'}{\rho'^\gamma} & = & \frac{\bar{p}}{\bar{\rho}^\gamma} \\ u' + \frac{2}{\gamma-1}c' & = & \bar{u} + \frac{2}{\gamma-1}\bar{c} \end{cases}$$

$$x = 1 \quad u' - \frac{2}{\gamma-1}c' = \bar{u} - \frac{2}{\gamma-1}\bar{c}$$

Linearisation and non-dimensionalisation of these boundary conditions give

$$C_{in} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & (\gamma-1) & \gamma \end{bmatrix},$$

and

$$C_{out} = \begin{bmatrix} 1 & (\gamma-1) & -\gamma \end{bmatrix}.$$

The resultant matrix B is then

$$B = \begin{bmatrix} 0 & 0 & -1 \\ -1 - \frac{(\gamma-1)(M(\epsilon-1)-\tau)}{2\epsilon} + \gamma & -1 - \frac{(\gamma-1)(M(\epsilon-1)+\tau)}{2\epsilon} + \gamma & -1 \\ \left(1 - \frac{(\gamma-1)(M(\epsilon-1)-\tau)}{2\epsilon} - \gamma\right)\exp(i\omega/\lambda_1) & \left(1 - \frac{(\gamma-1)(M(\epsilon-1)+\tau)}{2\epsilon} - \gamma\right)\exp(i\omega/\lambda_2) & \exp(i\omega/\lambda_3) \end{bmatrix}.$$

The determinant of B is

$$\begin{aligned} \det B &= -\frac{(\gamma-1)^2}{4\epsilon^2}\left[(-2\epsilon - M + \epsilon M - \tau)(2\epsilon - M + \epsilon M + \tau)\exp\left(\frac{i\omega}{\lambda_2}\right) - \right. \\ &\quad \left. (-2\epsilon - M + \epsilon M + \tau)(2\epsilon - M + \epsilon M - \tau)\exp\left(\frac{i\omega}{\lambda_1}\right)\right], \end{aligned}$$

and the eigenfrequencies obtained by solving $\det B = 0$ are

$$\begin{aligned} \omega_r &= \frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} 2n\pi \quad \forall n \\ \omega_i &= -\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1}\log\left(\frac{(2\epsilon+M-\epsilon M-\tau)(2\epsilon-M+\epsilon M-\tau)}{(2\epsilon+M-\epsilon M+\tau)(2\epsilon-M+\epsilon M+\tau)}\right) \\ &= -\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1}\log\left(\frac{\epsilon+1+\tau}{\epsilon+1-\tau}\right). \end{aligned}$$

In contrast to the unpreconditioned Euler equations the Riemann invariant boundary conditions are reflective, since ω_i is not $-\infty$. Again, looking at the decay per timestep, one finds that this one tends to zero when $M \to 0$, indicating that disturbances will not decay and that convergence to a steady state will never be reached.

5.4.3 Entropy, stagnation enthalpy at inflow; pressure at outflow

Another common set of boundary conditions for subsonic, internal flows is the specification of entropy and stagnation enthalpy at the inflow and pressure at the outflow:

$$\begin{aligned} x = 0 \quad & \begin{cases} \frac{p'}{\rho'^\gamma} &= \frac{\bar{p}}{\bar{\rho}^\gamma} \\ \frac{\gamma-1}{2}u'^2 + \frac{\gamma p'}{\rho'} &= \frac{\gamma-1}{2}\bar{u}^2 + \frac{\gamma\bar{p}}{\bar{\rho}} \end{cases} \\ x = 1 \quad & p' = \bar{p} \end{aligned}$$

For these boundary conditions, C_{in} and C_{out} are,

$$C_{in}\begin{bmatrix} -1 & 0 & 1 \\ -1 & (\gamma-1)M & \gamma \end{bmatrix}$$

$$C_{out}\begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

From C_{in} and C_{out}, the matrix B becomes

$$B = \begin{bmatrix} 0 & 0 & -1 \\ -1 - \frac{1}{2}\frac{(\gamma-1)M(M(\epsilon-1)-\tau)}{\epsilon} + \gamma & -1 - \frac{1}{2}\frac{(\gamma-1)M(M(\epsilon-1)+\tau)}{\epsilon} + \gamma & -1 \\ \exp(i\omega/\lambda_1) & \exp(i\omega/\lambda_2) & 0 \end{bmatrix}.$$

and hence, the eigenfrequencies which result in a zero determinant of B are

$$\begin{aligned} \omega_r &= \frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} 2n\pi \quad \forall n \\ \omega_i &= -\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} \log\left(\frac{\epsilon M^2 - M^2 - \tau M - 2\epsilon}{\epsilon M^2 - M^2 + \tau M - 2\epsilon}\right) \end{aligned}$$

These are the same as for the example 5.4.1, where stagnation enthalpy and total pressure were imposed at the inflow. This is an expected result, because in both cases, the same boundary conditions are enforced: setting stagnation enthalpy and total pressure automatically set the entropy, since the three quantities are related to each other.

5.4.4 Velocity, temperature at inflow; pressure at outflow

The final set of boundary conditions considered here is setting velocity and temperature at the inflow and pressure at the outflow. These conditions are fairly common in low speed viscous flow applications. Specifically,

$$x = 0 \quad \begin{cases} \frac{p'}{\rho'} &= \frac{\bar{p}}{\bar{\rho}} \\ u' &= \bar{u} \end{cases}$$

$$x = 1 \quad p' = \bar{p}$$

For these boundary conditions, C_{in} and C_{out} are,

$$C_{in} \begin{bmatrix} -1 & 0 & \gamma \\ 0 & 1 & 0 \end{bmatrix},$$

$$C_{out} \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

From C_{in} and C_{out}, the matrix B becomes

$$B = \begin{bmatrix} \gamma - 1 & \gamma - 1 & -1 \\ -\frac{1}{2}\frac{M(\epsilon-1)-\tau}{\epsilon} & -\frac{1}{2}\frac{M(\epsilon-1)+\tau}{\epsilon} & 0 \\ \exp(i\omega/\lambda_1) & \exp(i\omega/\lambda_2) & 0 \end{bmatrix},$$

and hence, the eigenfrequencies which result in a zero determinant of B are

$$\begin{aligned} \omega_r &= \frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1} 2\,(n+1)\,\pi \quad \forall n \\ \omega_i &= -\frac{\lambda_1\lambda_2}{\lambda_2-\lambda_1}\,\log\left|\frac{(\epsilon-1)M-\tau}{(\epsilon-1)M+\tau}\right| . \end{aligned}$$

When $M \to 0$, $\left|\frac{(\epsilon-1)M-\tau}{(\epsilon-1)M+\tau}\right| \to \frac{4\eta}{\left(\sqrt{1+4\eta}-1\right)^2}$. As in 5.4.1, the decay per timestep remains finite, and for all η, any disturbance at $t = 0$ will decay exponentially.

5.4.5 Remarks

The analysis of the effect of local preconditioning on boundary conditions for the subsonic, one dimensional Euler equations shows that care must be taken concerning the physical boundary condition which must be applied for a specific problem: Riemann invariant boundary conditions are reflective with preconditioning, and, at low Mach numbers, disturbances do not decay, whereas for the unpreconditioned Euler equations, these are non-reflective, with the unsteady perturbations becoming zero after the finite time it takes for all three characteristic waves to cross the domain once [27]. A similar conclusion has been reached by Darmofal [21] considering the one-dimensional van Leer-Lee-Roe preconditioner [**?**], and can probably be generalised to more preconditioners. As mentioned in [22] an interesting possibility implied by this analysis would be to design a preconditioner to address these problems, so that the boundary conditions are less reflective.

5.5 Stability analysis for Low Mach Number Preconditioning.

For low Mach number applications, the numerical dissipation and hence the block Jacobi preconditioner need to be modified, and a question of interest concerns the implications of the modification on the timestep stability limits presented in the previous chapter. Writing the Euler equations in symmetrised variables as

$$\frac{\partial\tilde{Q}}{\partial t} + \tilde{A}^x\frac{\partial\tilde{Q}}{\partial x} + \tilde{A}^y\frac{\partial\tilde{Q}}{\partial y} + \tilde{A}^z\frac{\partial\tilde{Q}}{\partial z} = 0, \tag{5.17}$$

where $\tilde{A}^x, \tilde{A}^y, \tilde{A}^z$ are the same as in section 4.1, the preconditioned form is

$$\Gamma^{-1}\frac{\partial \tilde{Q}}{\partial t} + \tilde{A}^x\frac{\partial \tilde{Q}}{\partial x} + \tilde{A}^y\frac{\partial \tilde{Q}}{\partial y} + \tilde{A}^z\frac{\partial \tilde{Q}}{\partial z} = 0. \tag{5.18}$$

Performing a change of variables to $V = \Gamma^{-\frac{1}{2}}\tilde{Q}$ gives

$$\frac{\partial V}{\partial t} + \Gamma^{\frac{1}{2}}\tilde{A}^x\Gamma^{\frac{1}{2}}\frac{\partial V}{\partial x} + \Gamma^{\frac{1}{2}}\tilde{A}^y\Gamma^{\frac{1}{2}}\frac{\partial V}{\partial y} + \Gamma^{\frac{1}{2}}\tilde{A}^z\Gamma^{\frac{1}{2}}\frac{\partial V}{\partial z} = 0, \tag{5.19}$$

which is very similar to the original symmetrised Euler equations in that the three coefficient matrices are symmetric.

The numerical discretisation including the characteristic smoothing and block-Jacobi preconditioning proceeds from this point in exactly the same way as with the Euler equations. Therefore the entire stability analysis remains the same.

In general the requirement for the analysis to remain valid is that the low Mach number preconditioner Γ must be symmetric and positive definite; $\Gamma^{\frac{1}{2}}$ is then well-defined. The van Leer-Lee-Roe matrix satisfies this condition [82], but those due to Turkel [77] and Lee [**?**] do not. It is the lack of symmetry in the matrix which is responsible for the very large transient growth analysed by Darmofal and Schmidt [23]. The analysis in this chapter shows that such a large growth will not occur for symmetric preconditioners.

Chapter 6

Numerical Results

This chapter gathers numerical results for a complete set of applications. 2D and 3D inviscid/viscous test cases have been run on grids of various complexities for a complete range of flow conditions. For viscous calculations, the main focus is on convergence acceleration rather than accuracy of the solution, since the implementation of the turbulence model has not been fully checked for the case of a flat plate. Nevertheless, compared to results coming from the literature [63, 71, 76], the Spalart and Allmaras turbulence model seems to perform reasonably well.

6.1 Inviscid

This section will demonstrate the performance of the scalar and block-Jacobi preconditioners when used in conjunction with full coarsened multigrid for transonic airfoil calculations.

The code is constructed on a conservative edge based semi-discrete finite volume approach where the flux discretization is based on the flux-differencing ideas of Roe, combining central differencing of the nonlinear fluxes with a smoothing flux based on one-dimensional characteristic variables. The solution is computed on a sequence of 5 grids using full coarsened V-cycles in which one time step with 5-stage Runge-Kutta scheme is performed when moving up and down the multigrid cycle.

Test	Geometry	M_∞	α	Mesh	Nnode
TE1	NACA0012	0.8	1.25	Triangular	5766
QE2	NACA0012	0.8	1.25	Quadrilateral	20800

Table 6.1 Euler test case definitions: airfoil, free stream Mach number, angle of attack, type of grid, number of vertices.

The switched scheme is used only on the fine meshes and a purely first order numerical dissipation is used on all coarser meshes. The preconditioner is assembled and inverted before the first stage of each time step and an entropy fix prevents the time step from becoming too large near the stagnation point, at shocks and at the sonic line. Whereas the van Leer entropy fix [81] is used in the numerical dissipation the more severe Harten fix [31] is employed in the matrix preconditioner, providing better robustness.

6.1.1 2D Naca0012 airfoil

The test cases used to demonstrate the performance of the preconditioned multigrid method (Jacobi) in comparison to the standard approach (scalar) are defined in Table 6.1. For the convergence comparisons that follow, the L_2 norm of the residual vector (normalised by the initial residual) during one application of the time-stepping scheme on the finest mesh in the multigrid cycle is plotted. Convergence information is also provided in Tables 6.2 and 6.3 for the initial convergence rate between residual levels of 10^0 and 10^{-4} and the asymptotic convergence rate between residual levels of 10^0 and 10^{-10}.

The test case is a standard transonic NACA0012 case with a strong shock on the upper surface and a weak shock on the lower surface. Calculations are first performed on a sequence of triangular grids, and then using a sequence generated from a 320×64 O-mesh. The grid sequences used can be seen in figures 6.1 and 6.2, respectively; the sizes of the grids are tabulated below, with the number in

Test	MG Cycle		CPU Time(sec)		MG Parameters	
	scalar	Jacobi	scalar	Jacobi	scalar	Jacobi
TE1	267	112	269	114	5 2 5 1 1 5	5 2 5 1 1 5
QE2	180	108	1072	382	5 2 5 2 2 5	5 2 5 1 1 5

Table 6.2 Euler results: Initial ($10^0 \rightarrow 10^{-4}$) convergence comparisons for scalar preconditioning vs. block-Jacobi preconditioning, both with full coarsened multigrid. Multigrid cycles, CPU time, multigrid parameters (# grids, level on/above first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level).

Test	MG Cycle		CPU Time(sec)		MG Parameters	
	scalar	Jacobi	scalar	Jacobi	scalar	Jacobi
TE1	876	347	833	336	5 2 5 1 1 5	5 2 5 1 1 5
QE2	485	338	2560	1038	5 2 5 2 2 5	5 2 5 1 1 5

Table 6.3 Euler results: Asymptotic ($10^0 \rightarrow 10^{-10}$) convergence comparisons for scalar preconditioning vs. block-Jacobi preconditioning, both with full coarsened multigrid. Multigrid cycles, CPU time, multigrid parameters (# grids, level on/above first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level).

parentheses being the ratio of fine to coarse nodes.

	number of nodes			
base grid	1st collapse	2nd collapse	3rd collapse	4th collapse
5766	2180 (2.6)	893 (2.4)	403 (2.2)	214 (1.8)
20800	9496 (2.2)	3767 (2.5)	1825 (2.1)	603 (3.0)

For the triangular grid, the far field boundary is located at 50 chords away from the airfoil, and 30 chords away for the 320×64 mesh. It is worth noting that the leading edge of the airfoil can become severely distorted on the coarsest grid, while the trailing edge remains well defined (Fig. 6.1), and also that the coarsening procedure maintains the general topology of the domain. This is particularly obvious in the Fig. 6.2, where the coarse grids remain mainly composed of quadrilaterals. The

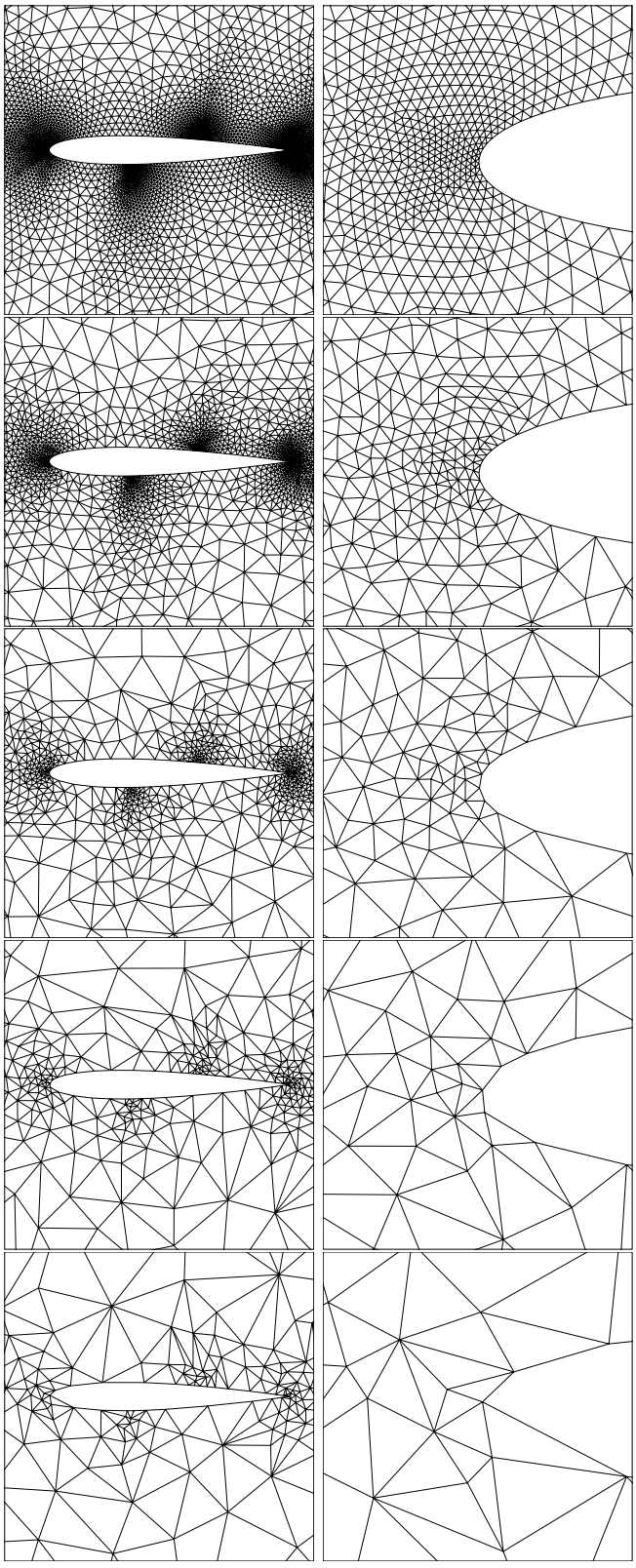


Figure 6.1 Element-collapse grids for the triangular NACA0012 airfoil. 5 levels.

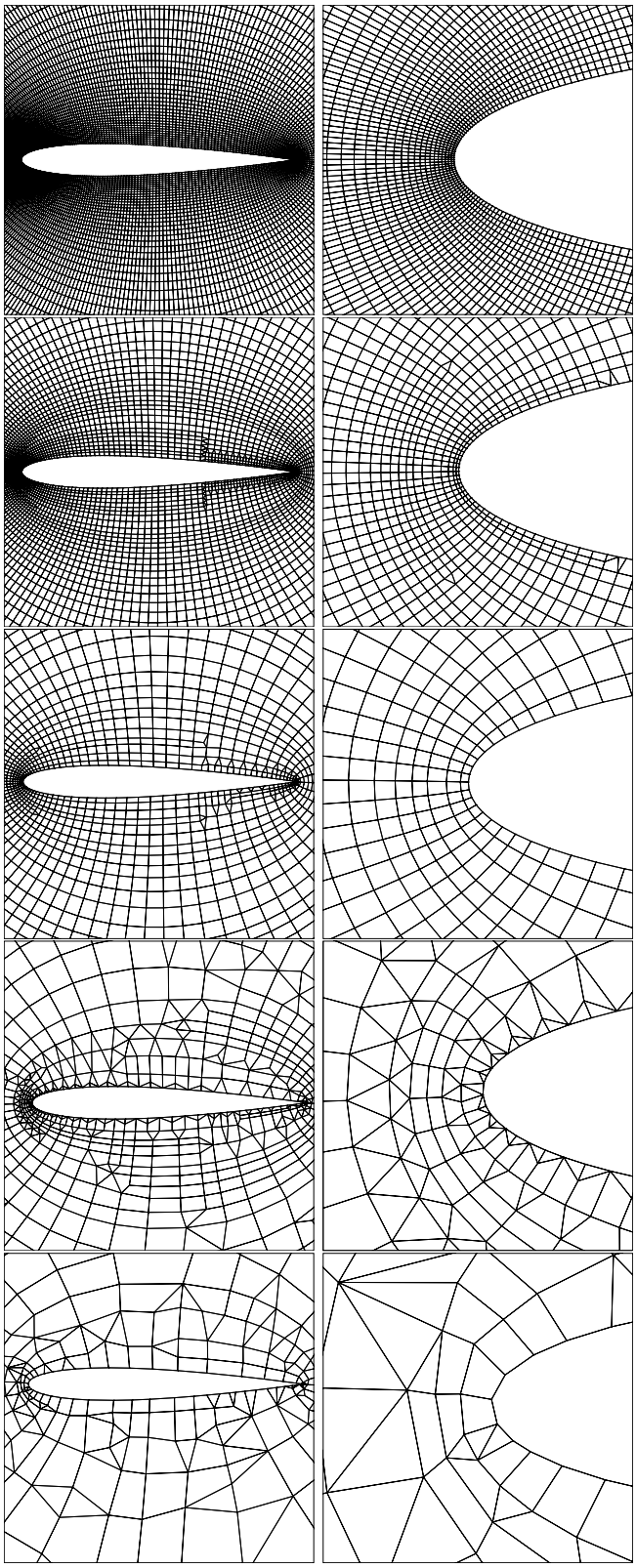


Figure 6.2 Element-collapse grids for the quadrilateral NACA0012 airfoil. 5 levels.

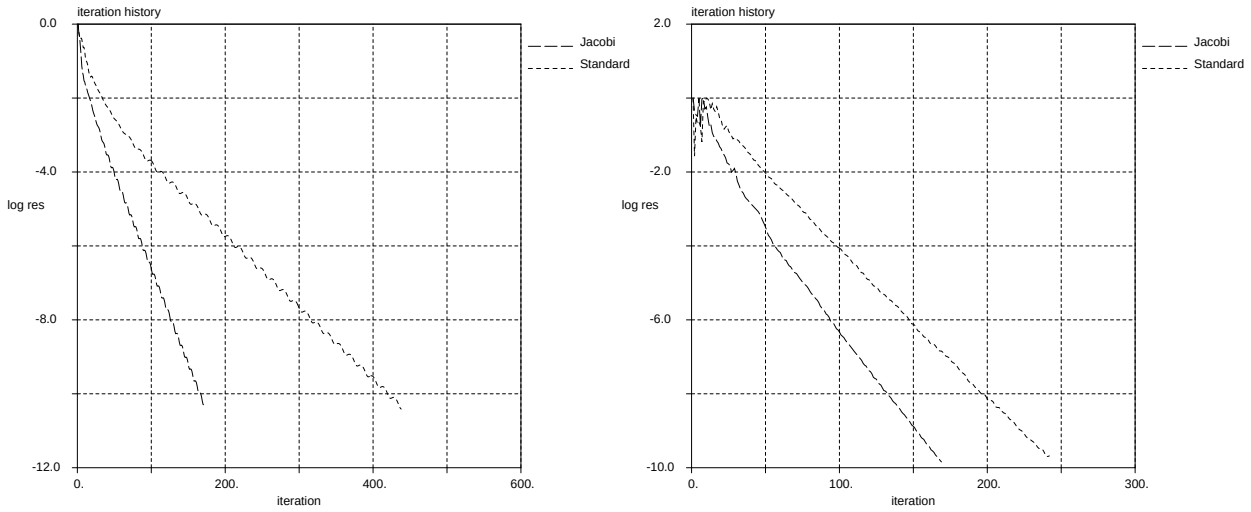


Figure 6.3 Tests TE1 and QE2: Convergence comparisons. NACA0012. $M_\infty = 0.8$, $\alpha = 1.25$ triangular mesh, quadrilateral mesh.

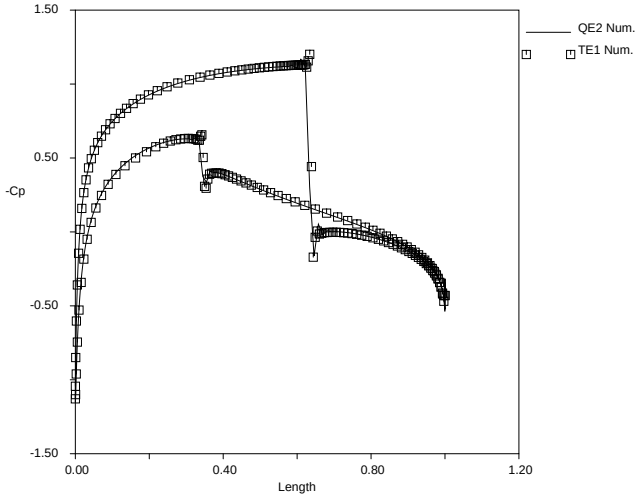


Figure 6.4 Solution: Comparison of the coefficients of pressure for the cases TE1 and QE2. NACA0012. $M_\infty = 0.8$, $\alpha = 1.25$.

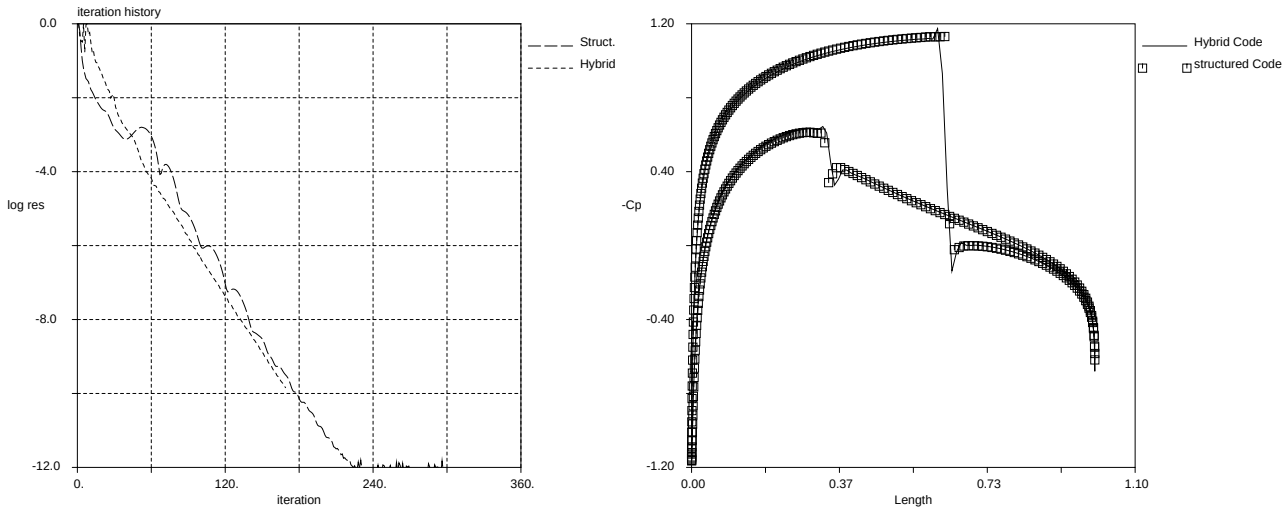


Figure 6.5 Comparisons between the structured approach [63] and the hybrid one. Convergence history and coefficient of pressure.

convergence histories are shown in Fig. 6.3. On each grid, both methods converge to machine accuracy with little degradation in asymptotic convergence requiring, on the triangular grid, 347 iterations with the Jacobi preconditioner and 876 iterations with the standard approach and 339 and 485 iterations, respectively on the quadrilateral mesh. In term of CPU time, the matrix preconditioner yields computational savings of a factor of 2.36 and 2.88, respectively, in initial convergence rate, and a factor of 2.48 and 2.44, respectively, in term of asymptotic convergence. The computed pressure distribution is shown in Fig. 6.4. The two plots are exactly the same, except across the strong shock where triangles generate a bigger overshoot.

Overall, the acceleration is achieved without any compromise in the robustness or accuracy of the flow solver. The accelerations are comparable to those obtained by Ollivier-Gooch [61] on unstructured grids, and Pierce [63] on structured meshes. The latest is underlined in Fig. 6.5 where the performance of the collapsing multigrid with the hybrid approach is compared with the results obtained by Pierce.

The scheme using block-Jacobi matrix preconditioning and full coarsened multigrid yields computational savings of roughly a factor of 2.5 for convergence to engineering accuracy.

6.1.2 Onera M6 Wing

The next example is a 3D tetrahedral grid around the standard Onera M6 wing located in the middle of half a cylinder whose length is 20 times the length of the longest chord and of a radius of roughly 13 chords. The grid has 147000 elements at the finest level, and three coarser levels are used in the multigrid, containing respectively, 68800, 12000 and 1365 elements. The collapsing algorithm is based on several criterion driving the collapsing procedure. In this case, the low coarsening ratio between the finest and the first coarser mesh (1.8) is due to the poor quality of the initial mesh which has elements with a dihedral angle of more than 180^o. The four grids are presented in Fig. 6.6, and the test case considered is $M_\infty = 0.84$, $\alpha = 3.06$. Figures 6.7 and 6.8 show the convergence history and the Mach contour

plot. Convergence to machine accuracy is achieved in 142 iterations using Jacobi preconditioning with a total CPU time of 1674 seconds and 237 iterations with the standard approach in 2784 seconds. In term of CPU time, the computational saving is roughly of a factor of 1.7.

6.1.3 Falcon Business Jet

The last example presented is over a geometry of increased complexity. It involves the solution of inviscid transonic flow over a Falcon jet (Avions Marcel Dassault, France). The geometry consists of a half complete aircraft configuration bounded by a symmetry plane. The airplane is located in the middle of half a cylinder whose length is roughly 5 times the length of its body and whose radius is roughly twice the same length. The fine grid has 156000 vertices and 847000 tetrahedra. Two coarser grids are derived by the element collapsing algorithm and contain respectively, 58500 and 9800 grid points. Again, the low coarsening ratio between the finest and the first coarser mesh is due to the same reason as in the previous section. The three grids used for the multigrid are presented in Fig. 6.9. The freestream conditions are $M_\infty = 0.85$ and $\alpha = 2^o$. Figures 6.10 and 6.11 show the convergence history and the Mach contour plot where the shock patterns are evident. Convergence to machine accuracy is achieved in 249 iterations using Jacobi preconditioning with a total CPU time of 18670 seconds and 813 iterations with the standard approach in 60298 seconds. In term of CPU time, the computational saving is roughly of a factor of 3.2.

6.2 Viscous

This section will demonstrate the performance of the scalar and block-Jacobi preconditioners when used in conjunction with semi coarsening multigrid for two and three dimensional turbulent Navier-Stokes calculations on various meshes, all highly stretched.

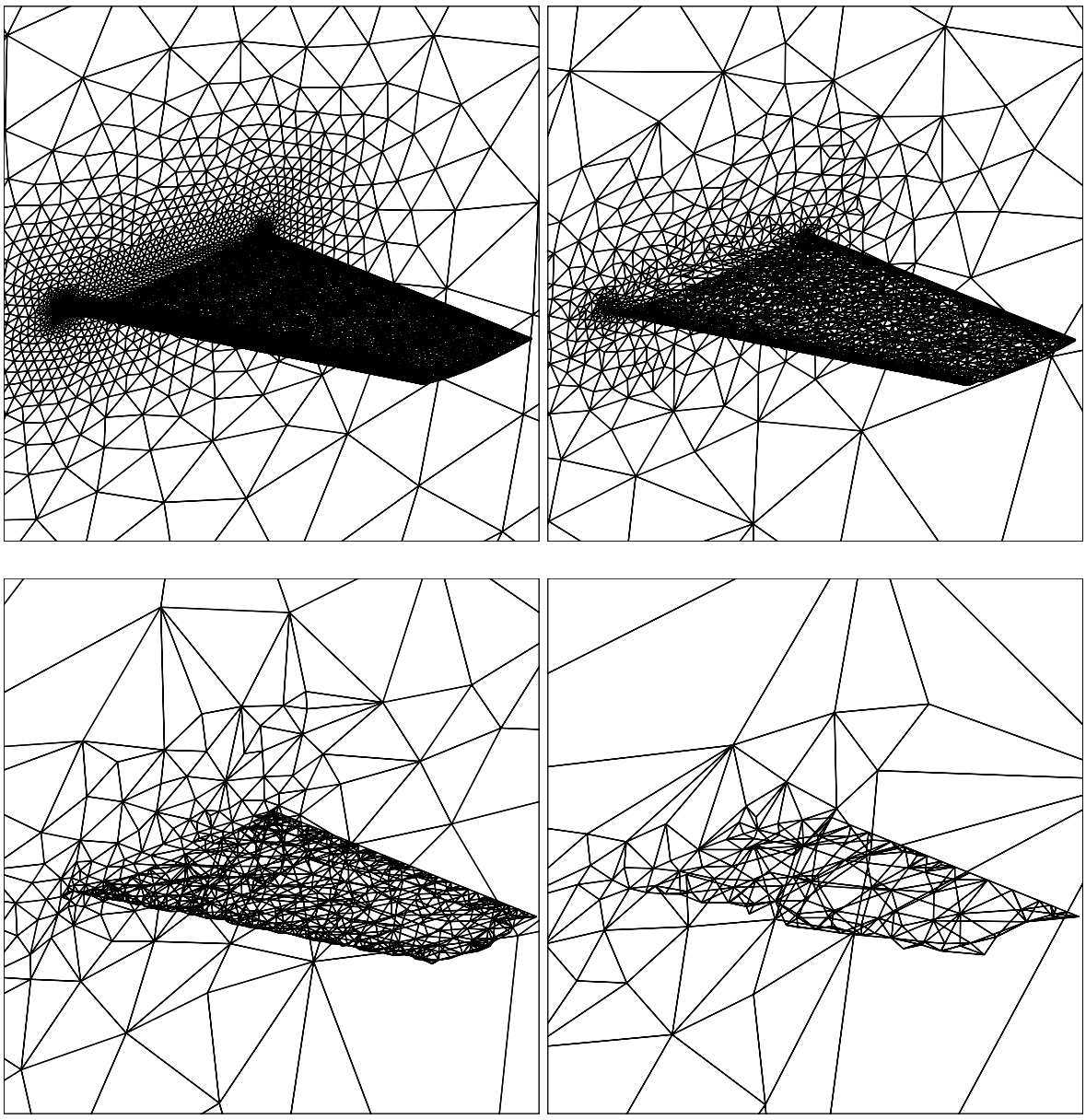


Figure 6.6 Element-collapse grids for the Onera M6 wing. 4 levels.

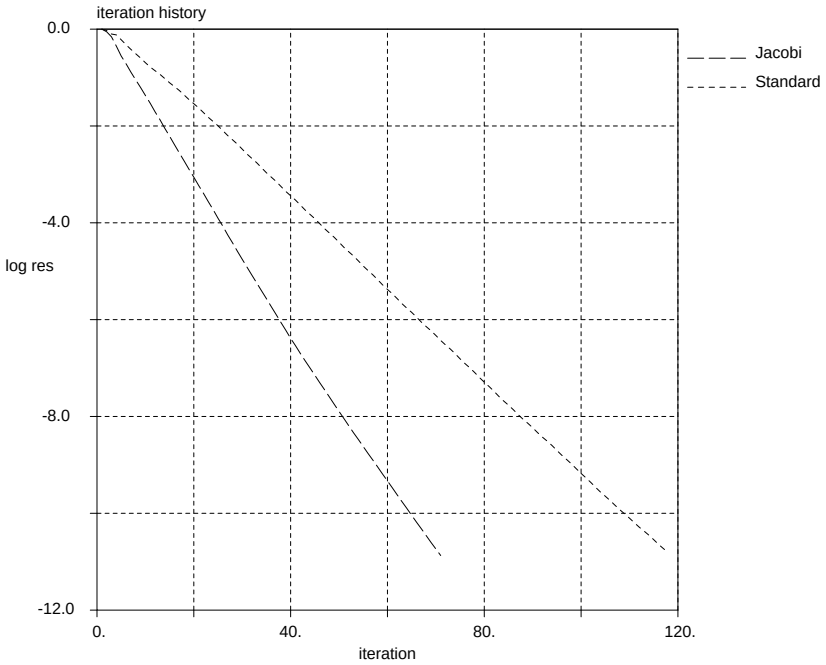


Figure 6.7 Onera M6 wing, convergence history

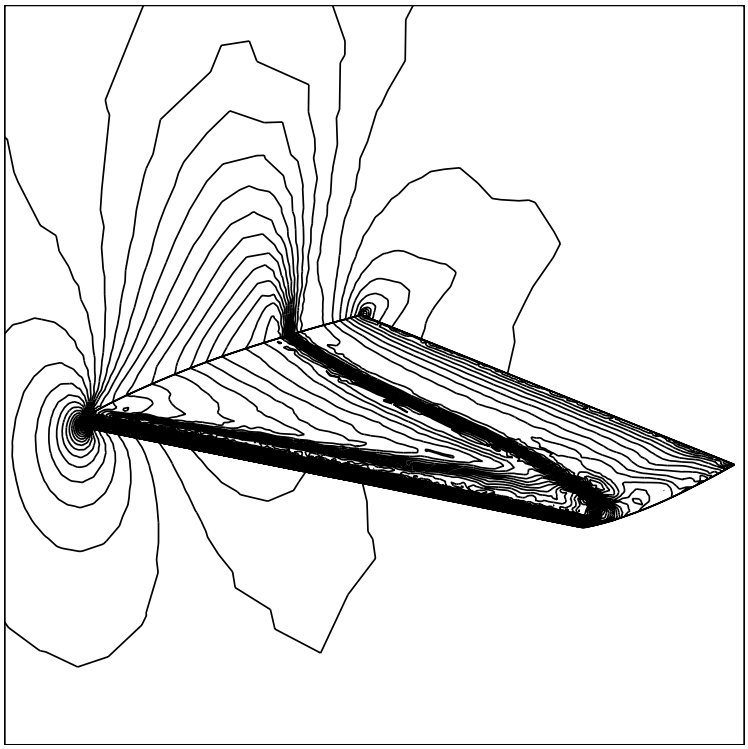


Figure 6.8 Onera M6 wing, Mach number contours

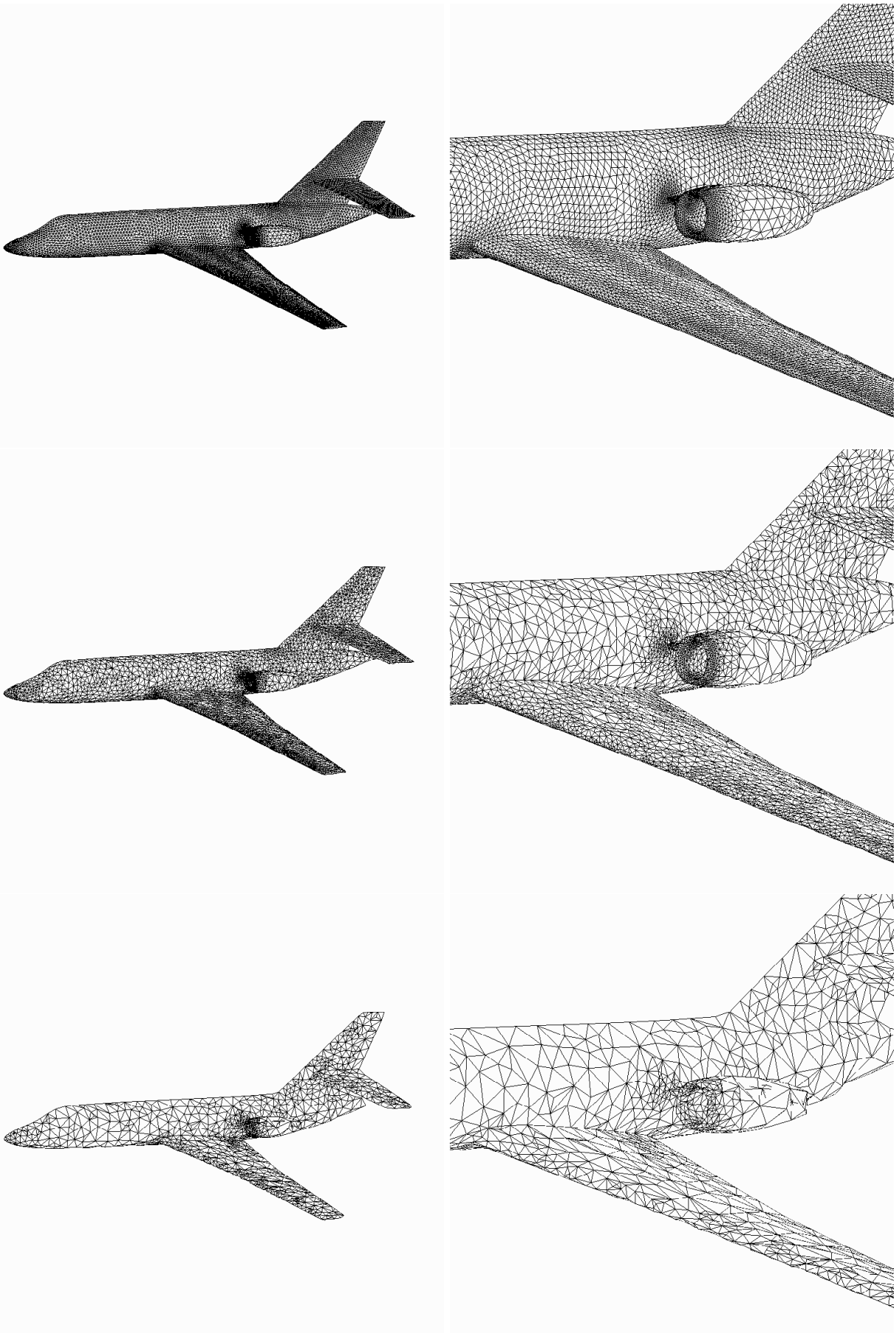


Figure 6.9 Element-collapse grids for the business jet. 3 levels.

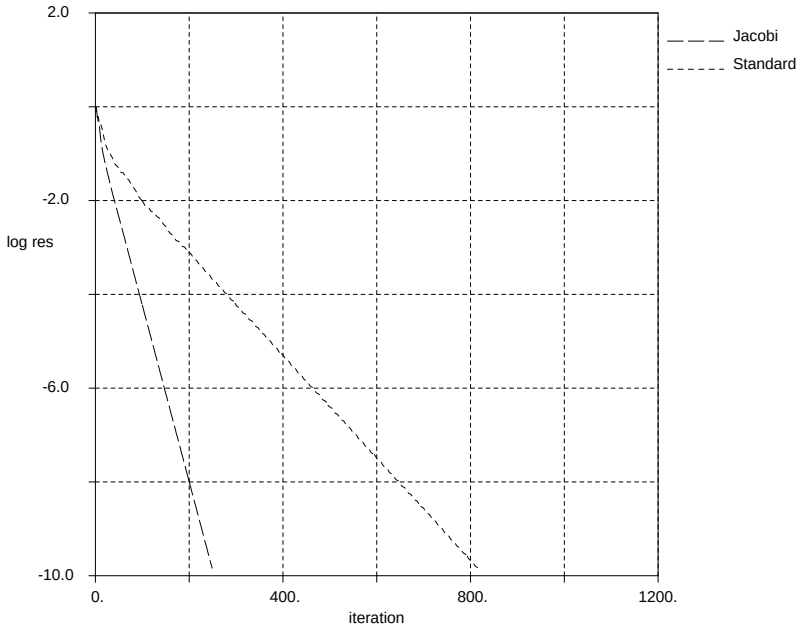


Figure 6.10 Business jet, convergence history

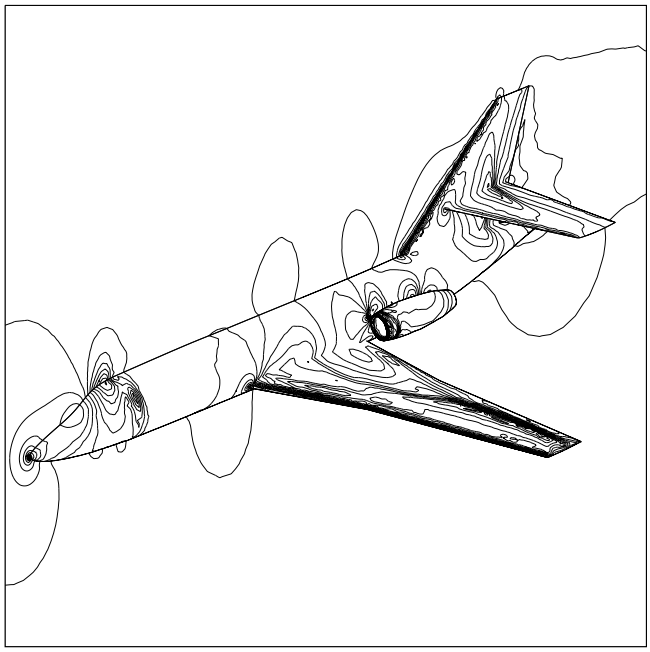


Figure 6.11 Business jet, Mach number contours

The discretization of the inviscid fluxes is identical to that described for the Euler solver. Special attention is paid to the treatment of the highly stretched grids in both the discretization and the element collapse multigrid method. The viscous flux is approximated half-way along each edge by a straightforward average of two central differences, which will not damp high frequency modes. Although the inviscid flux includes numerical dissipation terms that will damp these modes, this is insufficient inside the boundary layer where the viscous terms dominate. To remedy this, the component of the gradient in the direction along the edge is replaced by a simple difference along the edge. The element collapsing procedure removes elements to coarsen a given fine grid, essentially following a semi-coarsening strategy in the stretched regions [59]. An isotropic process collapses the rest of the domain.

For turbulent Navier-Stokes calculations, solutions are computed on a sequence of meshes using semi coarsened V-cycles on each mesh. The first point nearest to the wall is fixed for each grid so that $y^+ < 3$, where $y^+ = yU_*/\nu$ with y, U_* and ν the normal distance from the wall, the wall shear velocity and the kinematic viscosity. The iterative scheme used to converge the discrete residual to zero is pseudo time-stepping using the 5-stage Runge-Kutta method developed by Martinelli [48] with a CFL number of 2.5 on each mesh, except for the hybrid grids on which computations have been performed with a CFL of 2.3. The switched formulation of the numerical dissipation is used on the fine meshes and a first order version is used on all coarser meshes. Again, the entropy fix in the matrix preconditioner prevents the timestep from becoming too big near the wall.

The turbulence model implemented is the one equation Spalart-Allmaras turbulence model [76]. It is solved using a first order spatial discretization and 5-stage Runge-Kutta time integration with implicit treatment of the source terms within the same multigrid algorithm as used for the flow equations. Precautions are taken to ensure that at any moment negative turbulent viscosity is introduced. It has never been necessary to freeze the turbulent viscosity after a certain level of convergence

Test	Geometry	M_∞	α	Re	Mesh	Nnode
TNS1	RAE2822	0.725	2.4	6.5×10^6	Triangular	11298
TNS2	RAE2822	0.73	2.8	6.5×10^6	Triangular	11298
HNS1	RAE2822	0.725	2.4	6.5×10^6	Hybrid	19126
HNS2	RAE2822	0.73	2.8	6.5×10^6	Hybrid	19126

Table 6.4 Two-dimensional turbulent Navier-Stokes test case definitions: airfoil, free stream Mach number, angle of attack, Reynolds number, type of grid, number of vertices.

to get convergence of the flow equations. For the convergence comparisons that follow, the L_2 norm of the Navier-Stokes residual vector (normalised by the initial residual) during one application of the time-stepping scheme on the finest mesh in the multigrid cycle is plotted.

6.2.1 2D RAE2822 airfoil

The two-dimensional turbulent Navier-Stokes test cases used for comparisons are defined in Table 6.4 and correspond to RAE2822 AGARD Cases 6 and 9 [12]. Solutions are computed on two different grids, one composed of triangles, the other one, hybrid. The two multigrid sequences used are depicted in Fig. 6.12 and 6.13. Convergence information is also provided in Tables 6.5 and 6.6 for the initial convergence rate between residual levels of 10^0 and 10^{-4} and the asymptotic convergence rate between residual levels of 10^0 and 10^{-8}.

Results are shown in Figs. 6.14, 6.15, 6.16, and 6.17. The computed pressure distributions compare well with the experimental data [12], although the turbulence model produces a shock location forward of the experimental location, behaviour which has been previously observed [76, 63]. The turbulence model predicts a shock in better agreement with the measurements for Case 9. The shock induces a small region of separation bubble measuring about 5% of chord.

Test	MG Cycle		CPU Time(sec)		MG Parameters	
	scalar	Jacobi	scalar	Jacobi	scalar	Jacobi
TNS1	311	122	839	341	5 2 5 1 1 5	5 2 5 1 1
TNS2	565	171	1510	472	5 2 5 1 1 5	5 2 5 1 1
HNS1	234	78	1563	534	5 2 5 2 2 5	5 5 2 2 5
HNS2	297	170	1982	1149	5 2 5 2 2 5	5 5 2 2 5

Table 6.5 Two-dimensional turbulent Navier-Stokes results: Initial ($10^0 \rightarrow 10^{-4}$) convergence comparisons for scalar preconditioning vs. block-Jacobi preconditioning, both with semi-coarsened multigrid. Multigrid cycles, CPU time, multigrid parameters (# grids, level on/above first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level).

Test	MG Cycle		CPU Time(sec)		MG Parameters	
	scalar	Jacobi	scalar	Jacobi	scalar	Jacobi
TNS1	947	240	2522	649	5 2 5 1 1 5	5 2 5 1 1 5
TNS2	1233	361	3293	1052	5 2 5 1 1 5	5 2 5 1 1 5
HNS1	1788	918	11956	6152	5 2 5 2 2 5	5 2 5 2 2 5
HNS2	2655	1697	17645	11377	5 2 5 2 2 5	5 2 5 2 2 5

Table 6.6 Two-dimensional turbulent Navier-Stokes results: Asymptotic ($10^0 \rightarrow 10^{-8}$) convergence comparisons for scalar preconditioning vs. block-Jacobi preconditioning, both with semi-coarsened multigrid. Multigrid cycles, CPU time, multigrid parameters (# grids, level on/above first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level).

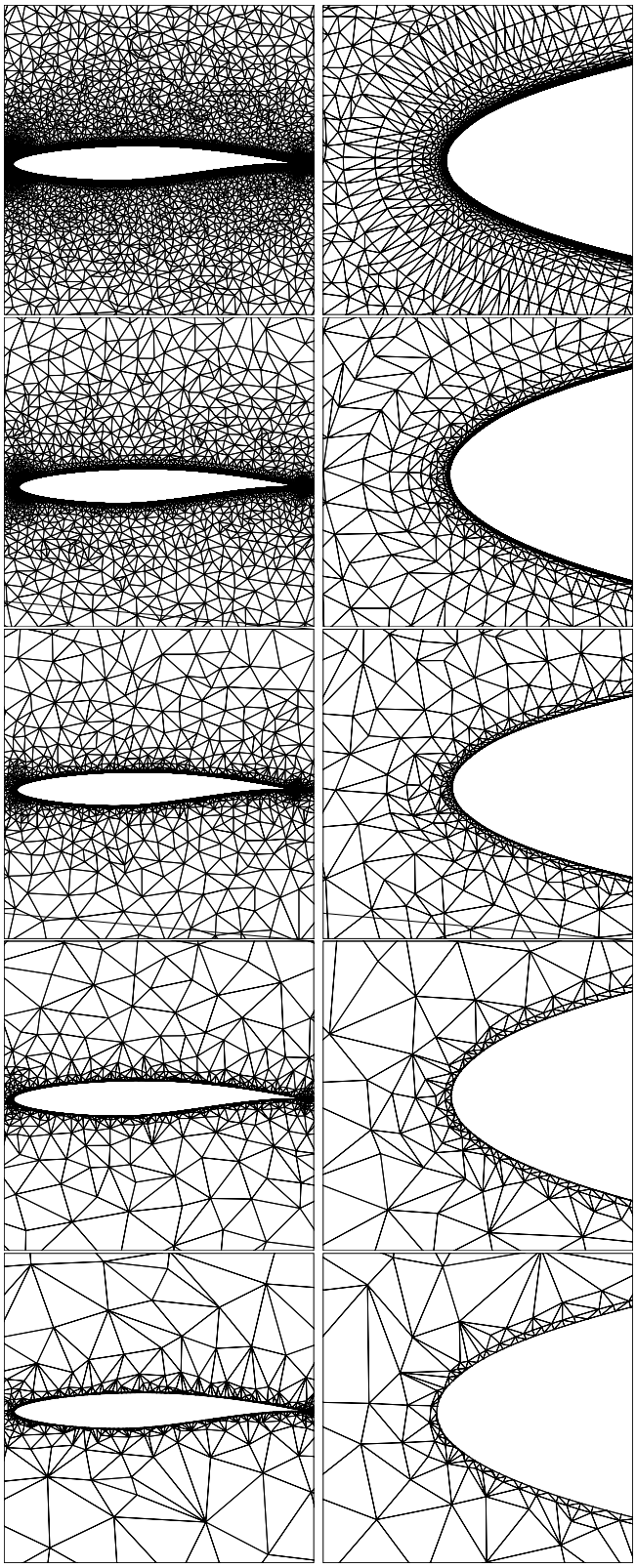


Figure 6.12 Edge-collapse grids for the triangular RAE2822 airfoil. 5 levels.

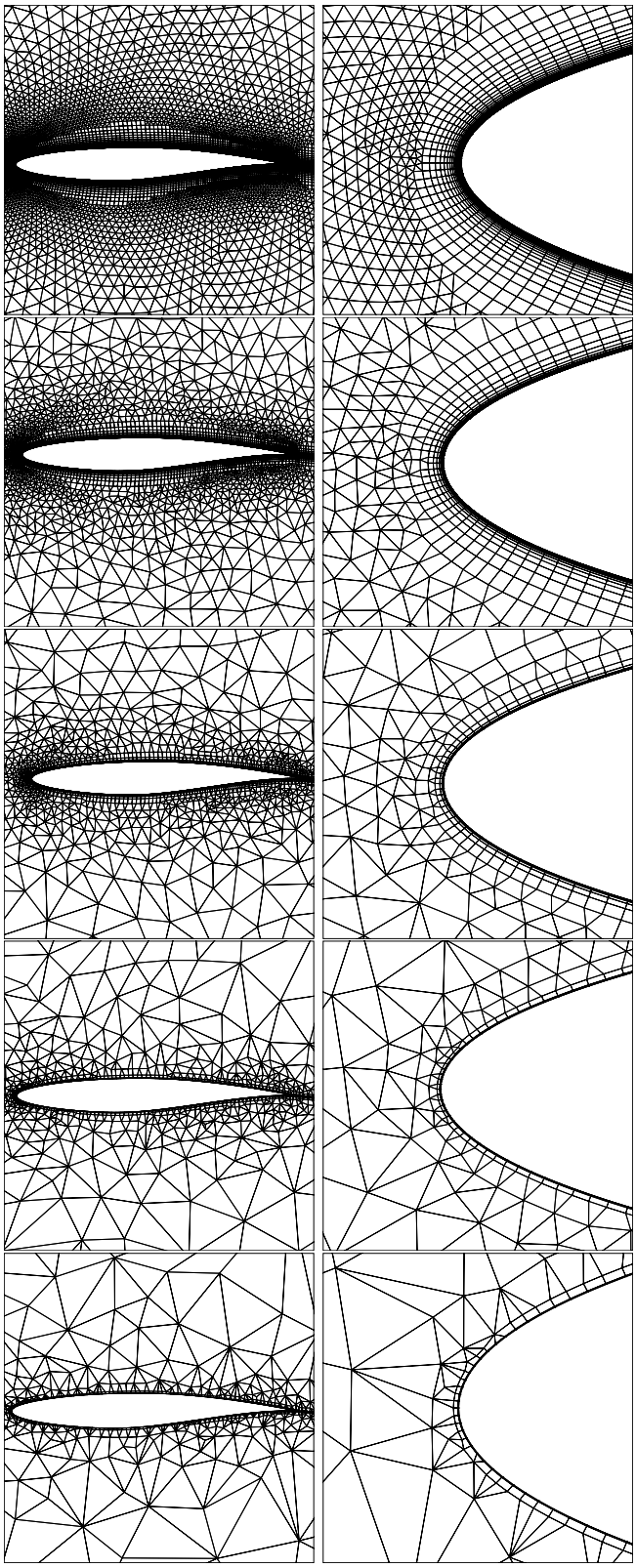


Figure 6.13 Edge-collapse grids for the hybrid RAE2822 airfoil. 5 levels.

Convergence histories are shown for each grid, in Figs. 6.14, 6.15, 6.16, and 6.17, for the block-Jacobi preconditioning and the standard approach of the scalar preconditioning, both with semi-coarsened multigrid. In all cases, both methods converge to machine accuracy, along with the turbulence model. The Jacobi approach converges quite smoothly and rapidly to engineering accuracy on triangular and hybrid grids. This remains true on triangles for asymptotic convergence, but not on the hybrid mesh. After 4 orders of magnitude, a severe degradation in convergence occurs, making this method only as efficient as the standard one. It is difficult to explain this dramatic loss, but one factor, based on Pierce work, could be related to the semi coarsening strategy. He observed that [63]:

> [...], the dominant effect of semi-coarsening [...] is to change the shape of the convergence history by dramatically improving the asymptotic convergence rate using either preconditioner so that the "elbow" at three orders of magnitude is eliminated.

The matrix preconditioner damps the convective modes very effectively, and these are dominant in the initial stages of convergence. Then, acoustic modes become significant, and semi coarsening yields improvements throughout the convergence process. It looks as if the semi-coarsening strategy in the edge collapsing procedure has not given meshes with a perfect regularity, where only the mesh interval in the normal direction is allowed to double. In the unstructured approach, this is not guaranteed.

In any case, the matrix preconditioner yields computational savings of a factor of 2.46 and 2.92, in initial convergence rate, for Case 6 on triangular grid and hybrid grid, respectively (factor of 3.2 and 1.73, for Case 9), and of a factor 3.88 and 3.13 in asymptotic convergence on the triangular grid for Cases 6 and 9, respectively (comparisons on the hybrid grid are here irrelevant).

In general, the convergence rate is significantly enhanced, and the improvement ranges from a factor 3 for engineering accuracy to 3-4 for asymptotic convergence.

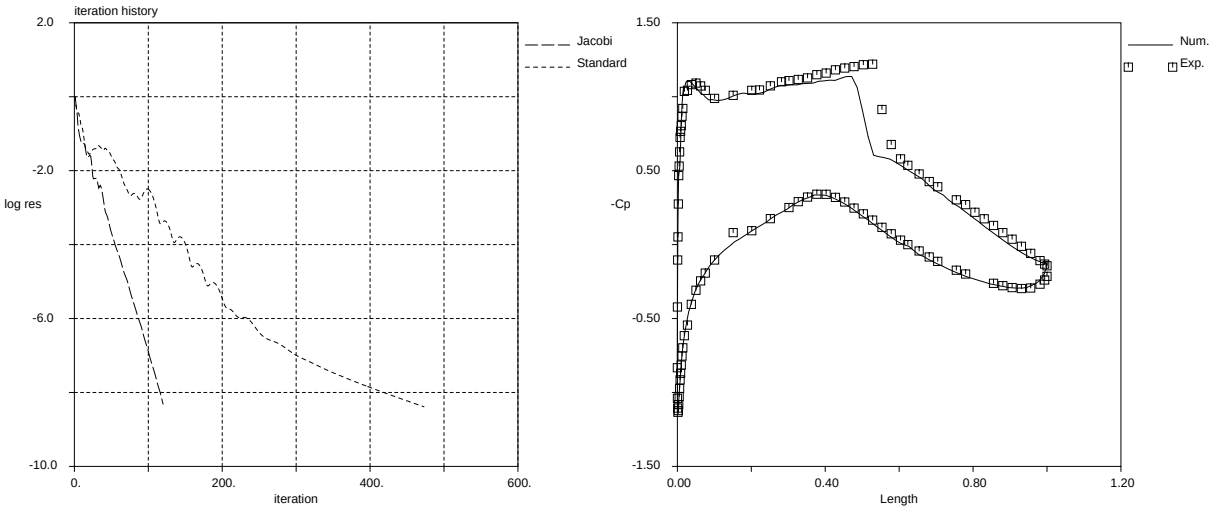


Figure 6.14 Test TNS1: Convergence comparison and Coefficient of pressure. RAE2822 AGARD Case 6. $M_\infty = 0.725$, $\alpha = 2.4$, $Re = 6.5 \times 10^6$ triangular mesh.

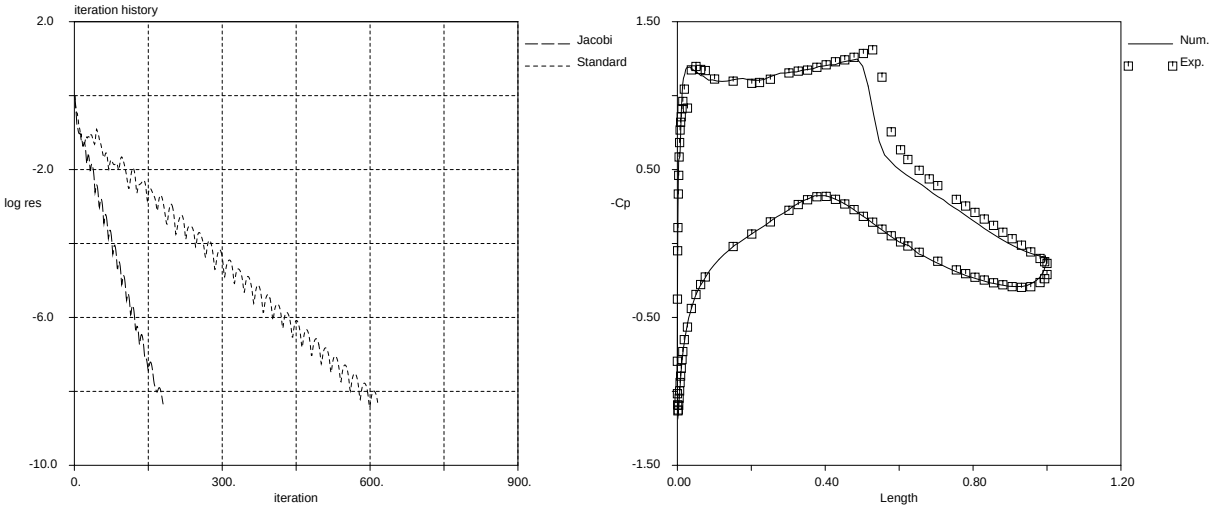


Figure 6.15 Test TNS2: Convergence comparison and Coefficient of pressure. RAE2822 AGARD Case 9. $M_\infty = 0.73$, $\alpha = 2.8$, $Re = 6.5 \times 10^6$ triangular mesh.

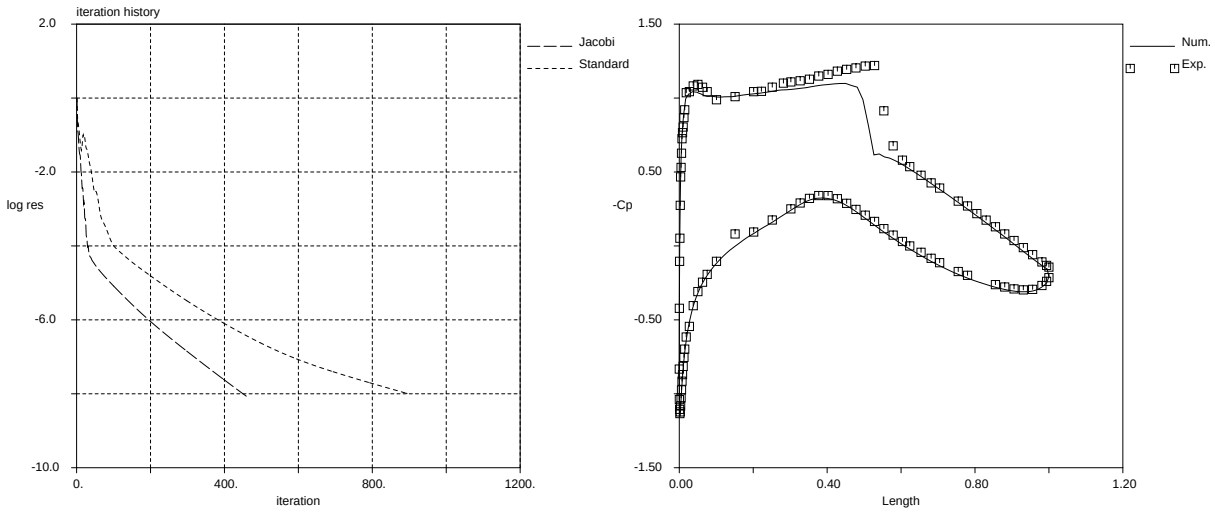


Figure 6.16 Test HNS1: Convergence comparison and Coefficient of pressure. RAE2822 AGARD Case 6. $M_\infty = 0.725$, $\alpha = 2.4$, $Re = 6.5 \times 10^6$ hybrid mesh.

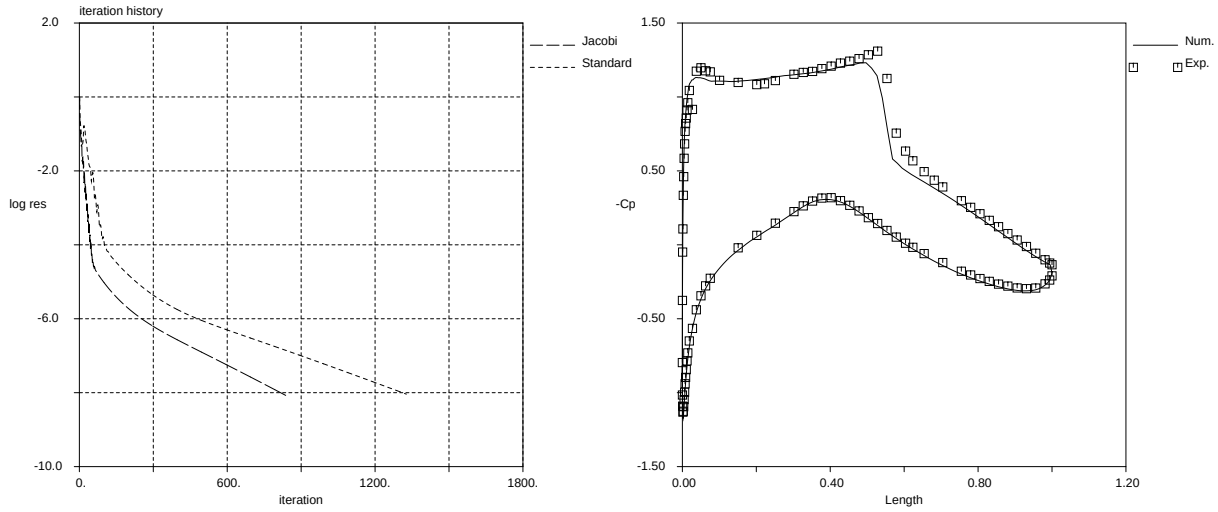


Figure 6.17 Test HNS2: Convergence comparison and Coefficient of pressure. RAE2822 AGARD Case 9. $M_\infty = 0.73$, $\alpha = 2.8$, $Re = 6.5 \times 10^6$ hybrid mesh.

6.2.2 3D bypass duct

The final example is the flow through the 3D bypass duct of a turbofan engine. The geometry is composed of ten struts and a pylon. The fine grid has 274000 grid

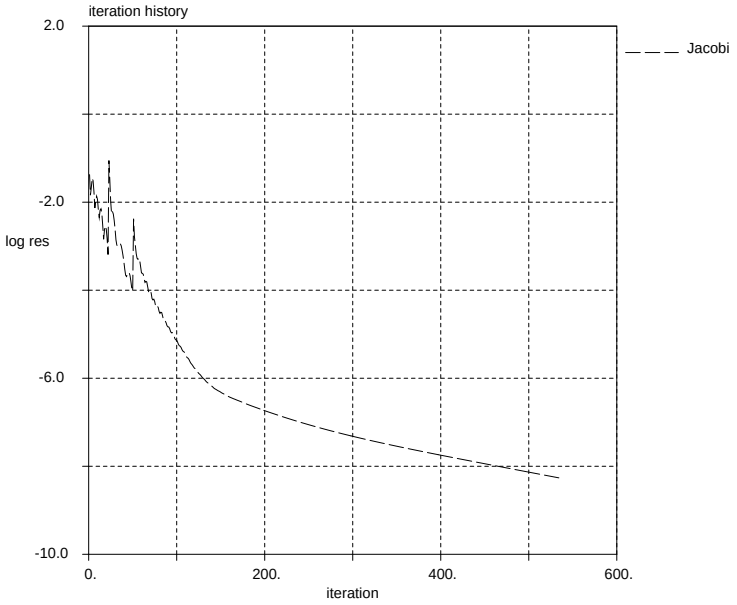


Figure 6.18 3D bypass duct; convergence history.

points and is constructed by stacking a sequence of 2D grids. Convergence history and Mach contours can be seen in Figures 6.18 and 6.19. From the fine grid, 2 coarser grids are produced containing respectively 138000 and 79300 vertices. The coarsening ratio is low because the multigrid semi-coarsening strategy is essentially only removing points in one-dimension in the areas of high stretching, which is both through the boundary layer and radially. The radial stretching is a consequence of the grid being composed of stacked 2D grids with a fixed radial step. This leaves a high aspect ratio in the radial direction in all regions of the 2D grid that have a much smaller mesh spacing than the radial step. For an inflow Mach number of 0.55, with zero incidence and a Reynolds number of 6 million around the struts, convergence to 6 orders of magnitude is reached in 250 multigrid cycles (the pylon is here treated as inviscid, because the purpose of studying this geometry did not require the pylon boundary layer to be resolved).

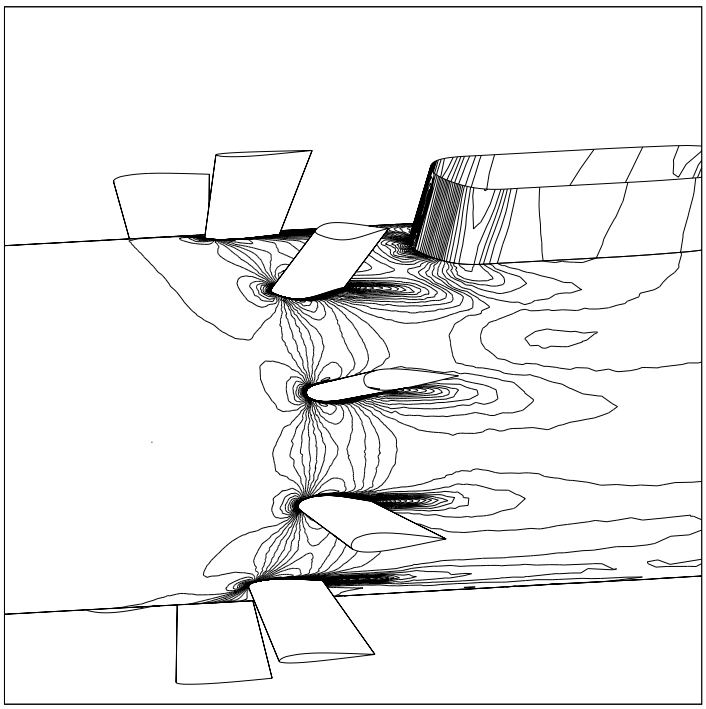


Figure 6.19 3D bypass duct; Mach number contours.

6.3 Low Mach number flow

6.3.1 Inviscid 2D Naca0012 airfoil

Based on the same sequence of grid as shown in Fig. 6.1, four cases are investigated, as listed in Table 6.7. Convergence histories are plotted together in Fig. 6.21, and additional information is also provided in Table 6.8. The results presented have been obtained with the matrix timestep, and it appears as expected that convergence is Mach independent. In Fig. 6.20, plots of the coefficient of pressure contours for each case are depicted. For Mach numbers smaller than 0.1, an unpreconditioned code is much slower (see Fig. 6.21) and produces, when it converges, solutions of poor quality with oscillations in the pressure and density contour plot (Fig. 6.22). These problems are fully addressed with a low Mach number preconditioner which preserves accuracy in the incompressible limit.

Test	Geometry	M_∞	α	Mesh	Nnode
TELM1	NACA0012	0.1	0.0	Triangular	5766
TELM2	NACA0012	0.01	0.0	Triangular	5766
TELM3	NACA0012	0.001	0.0	Triangular	5766
TELM4	NACA0012	0.0001	0.0	Triangular	5766

Table 6.7 Euler test case definitions for low Mach number: airfoil, free stream Mach number, angle of attack, type of grid, number of vertices.

6.3.2 Accuracy preservation by preconditioning

For an incompressible, non-viscous fluid, the velocity field and hence the pressure coefficient can be obtained by solving the potential flow via the Schwartz-Christoffel conformal mapping [47]. To demonstrate the accuracy discussed in the previous section, the computed pressure coefficient is here compared with that obtained with the Schwartz-Christoffel tool-box written by Toby Driscoll [25]. In Fig. 6.23 the comparison is presented for each case, proving an undeniable accuracy of the low Mach number preconditioner in the incompressible limit.

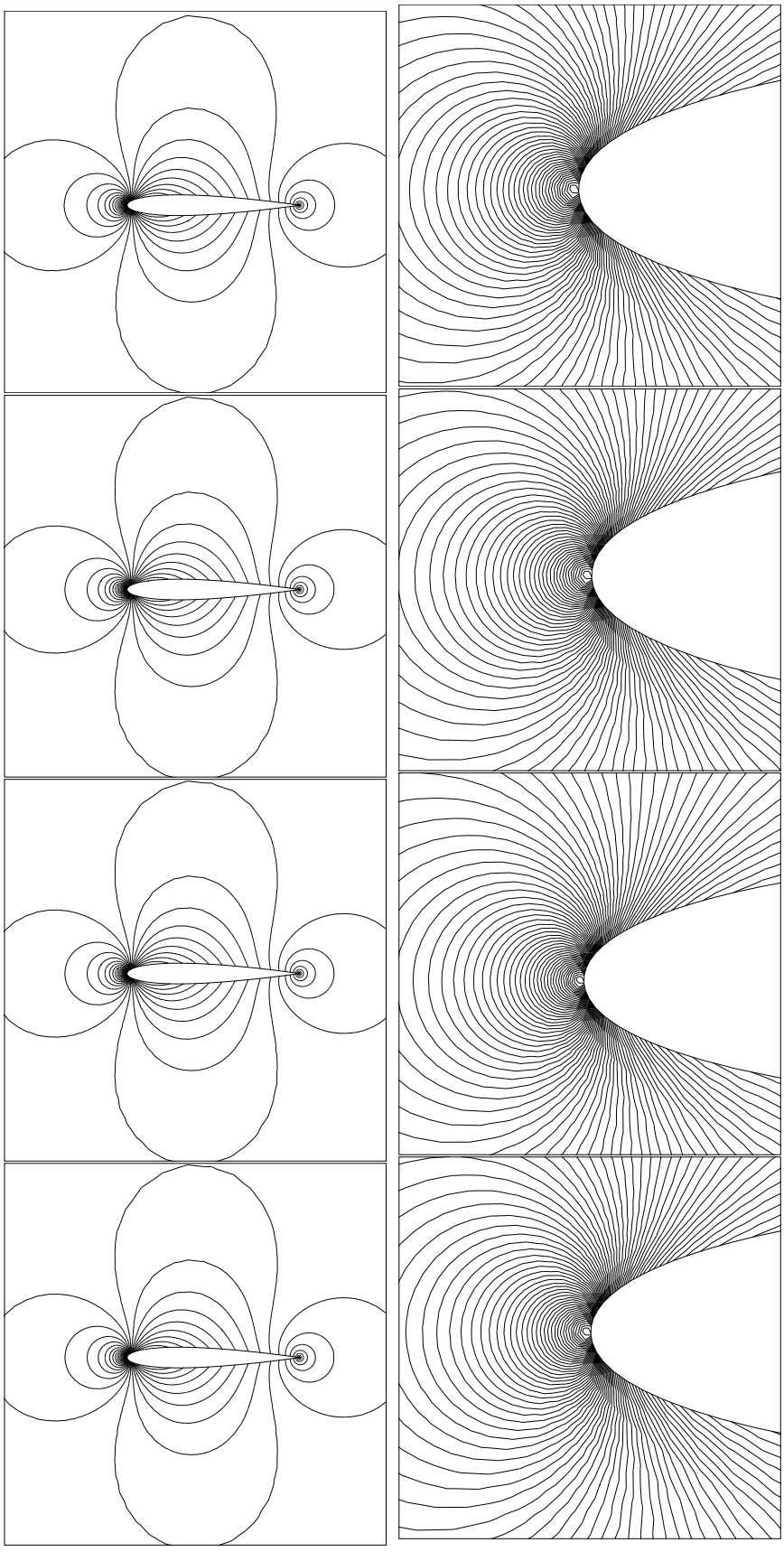
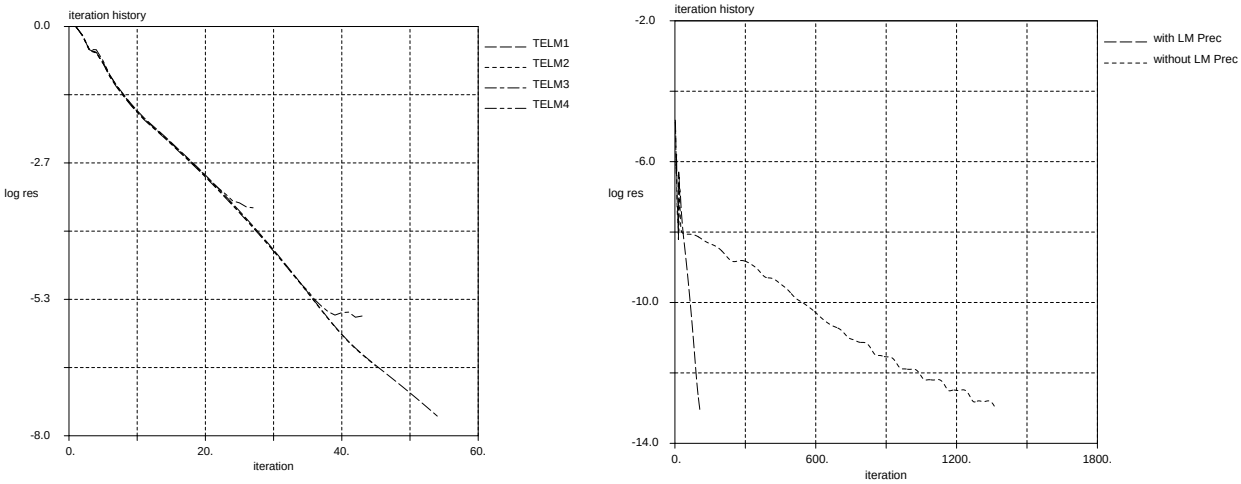


Figure 6.20 Computed pressure coefficient contours around a Naca0012 airfoil, overview and zoom around the leading edge. $M_\infty = 0.1, 0.01, 0.001, 0.0001$. Triangular mesh. $\alpha = 0$.



(a)Tests TELM1, TELM2, TELM3, TELM4. (b) Low Mach number preconditioner effect

Figure 6.21 Convergence history comparison for low Mach number flow around a Naca0012 airfoil

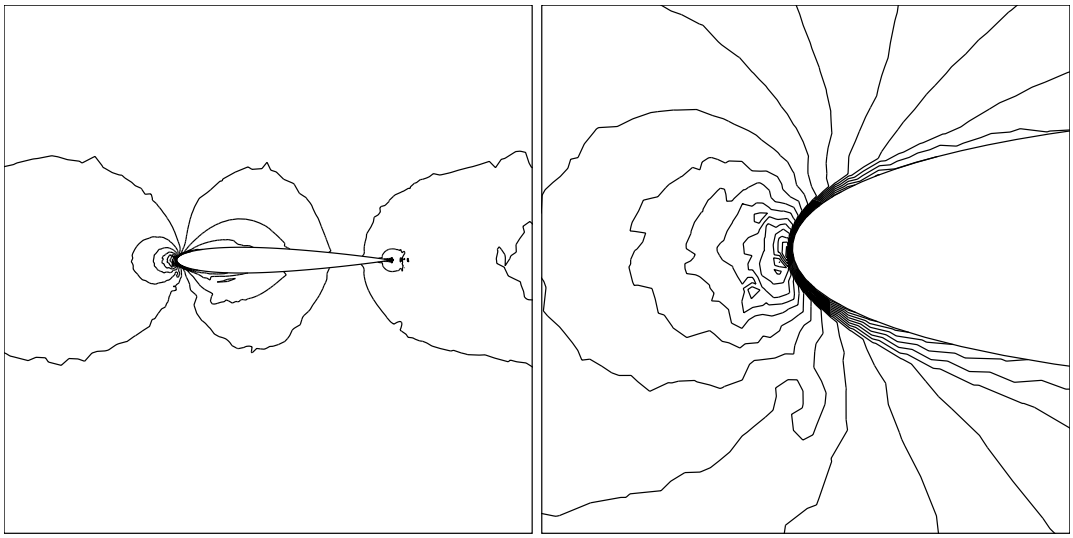


Figure 6.22 Computed pressure plot for $M_\infty = 0.01$ without low Mach number preconditioner. $\alpha = 0$.

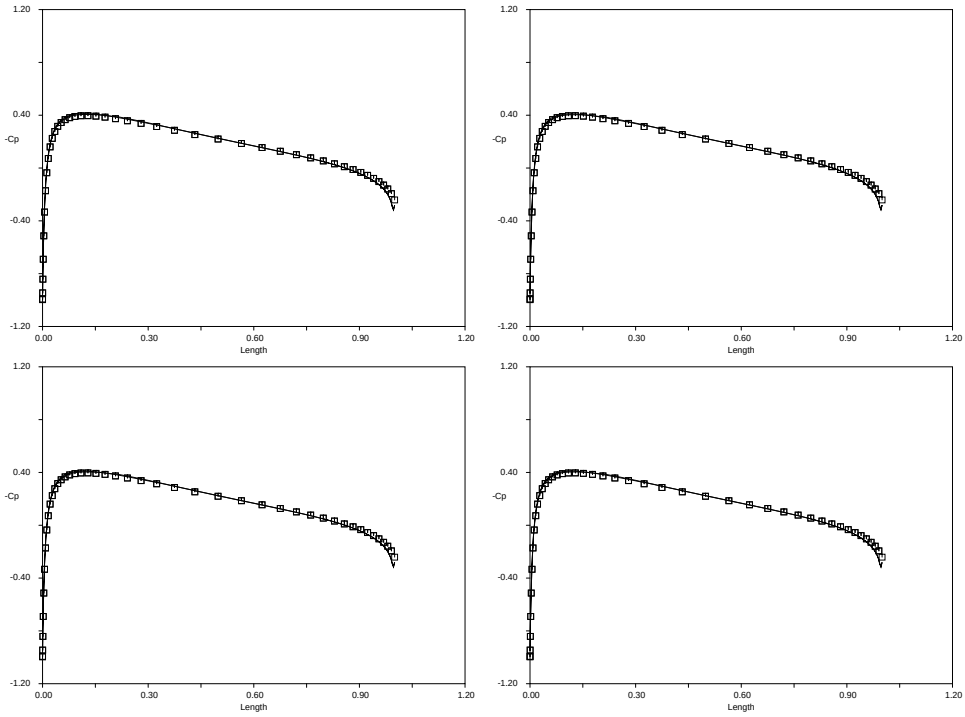


Figure 6.23 Computed pressure coefficient contours around a Naca0012 airfoil. Comparison with the potential flow solution. $M_\infty = 0.1, 0.01, 0.001, 0.0001$. $\alpha = 0$.

Test	MG Cycle	CPU Time(sec)	MG Parameters
TELM1	109	156	5 2 5 1 1 5
TELM2	91	133	5 2 5 1 1 5
TELM3	86	124	5 2 5 1 1 5
TELM4	57	81	5 2 5 1 1 5

Table 6.8 Convergence history using block-Jacobi preconditioning with full-coarsened multigrid. Multigrid cycle, CPU time, multigrid parameters. (# grids, level on/above first order smoothing is used, level for full multigrid startup, # pre and post smoothing iterations, # iterations on coarsest level)

6.3.3 A 3D example: the U-bend duct

The next example is the numerical computation of a flow inside a sharp U-bend used as an internal cooling passage inside the turbine rotor blades to maintain the operating temperature of the blades down to safe levels. The geometry is shown in Fig. 6.24 and the mesh containing only hexahedrals has 200000 vertices. The calculation is performed on a single grid since no sequence of coarser meshes was available at the moment of the writing of this document, and the convergence history is presented in Fig. 6.27. The purpose of this example is to check the 3D capabilities of the implementation of the low Mach number preconditioner. The Reynolds number defined by $\frac{U_m D}{\nu}$, where U_m is the bulk mean velocity and D the duct width is set to 100000. Figure 6.25 and 6.26 provide a picture of the overall flow development through the U-bend for a stationary condition. The boundary conditions are such that at the inflow, the total pressure and the total temperature are both set to 1 with an outflow static pressure equal to 0.986. The results indicate that there is a reasonably symmetric developing flow approaching the bend entry with a strong flow acceleration along the inner wall and deceleration along the outer wall which becomes stronger near the corner between the top and outer walls. As the flow progresses through the bend, the fluid begins to move faster over the outer wall. Along the inner wall, by the 90^o location the flow has separated near the

Figure 6.24 Geometry of a U-bend.

symmetry plane while, close to the top wall, the flow is still attached. By the bend exit, the separation region accros the pipe has grown to approximately 20% of the duct diameter, while near the top wall the reverse motion is weaker and confined to a narrower region. Over the downstream region, even though the reverse motion is stronger near the symmetry plane than near the top wall, reattachment occurs earlier along the symmetry plane. All these observations seem to be in a relative good agreement with the experiments [33, 34], but more investigations need to be done to fully validate the accuracy of the solution.

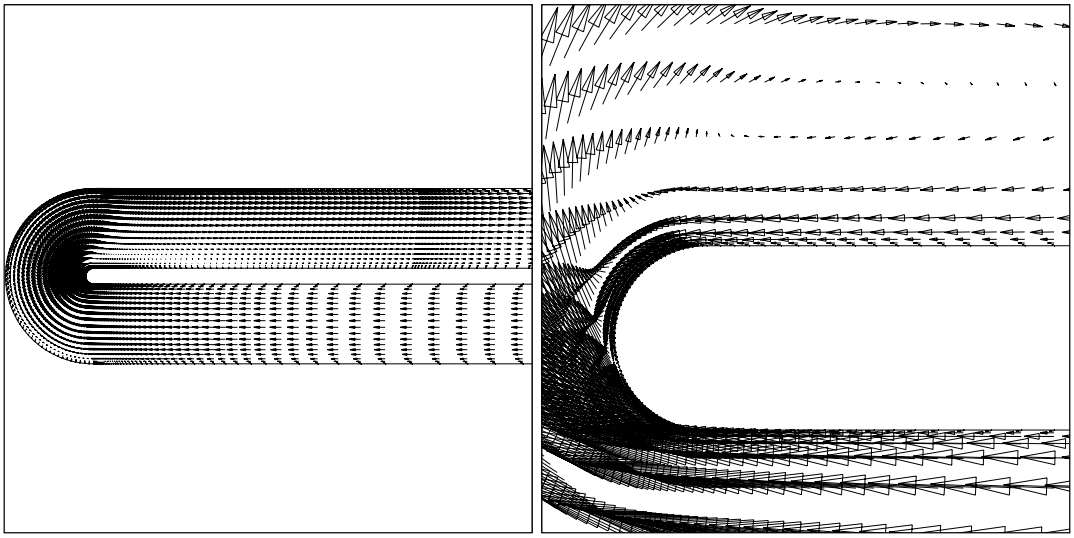


Figure 6.25 U-bend; Velocity field along the symmetry plane. Overview and zoom around the bend.

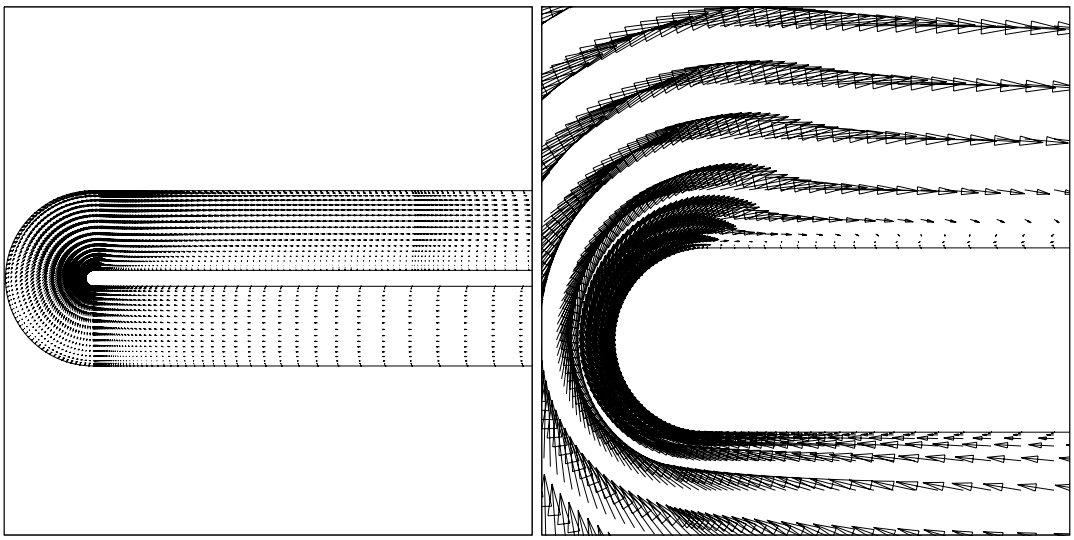


Figure 6.26 U-bend; Velocity field along the near-wall plane. Overview and zoom around the bend.

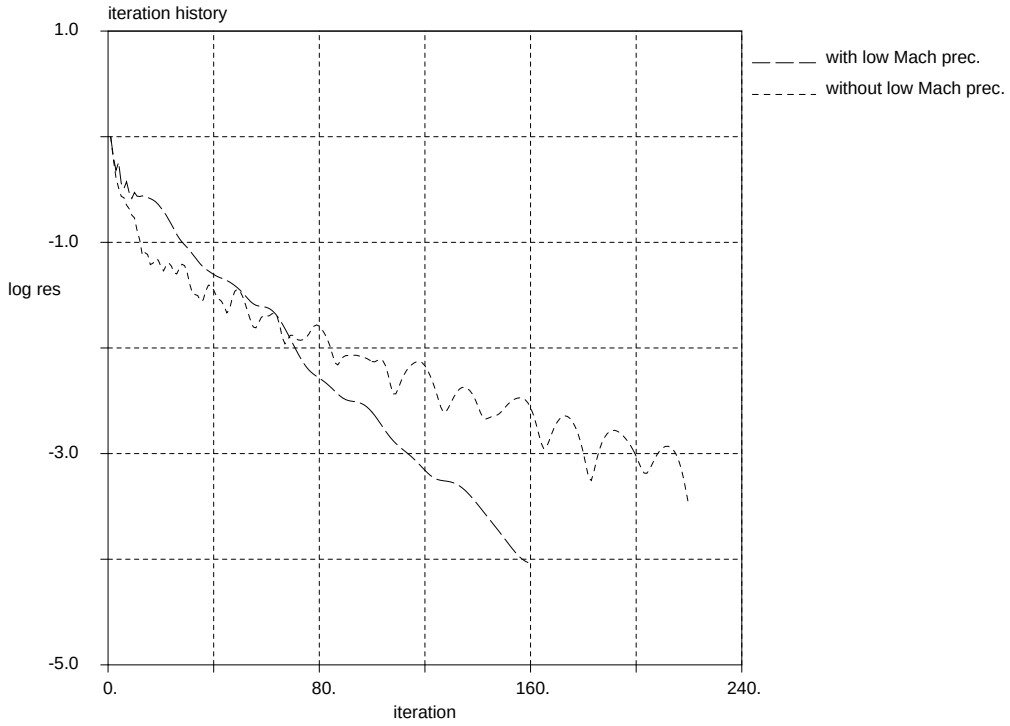


Figure 6.27 U-bend; convergence history.

Chapter 7

Conclusions and future work

7.1 Conclusion

In a continuation of recent breakthroughs in convergence acceleration, an efficient preconditioner working for a wide range of Mach numbers has been developed and implemented in a new multigrid algorithm for hybrid meshes. The standard scheme in common use employs local time step based on the largest eigenvalues of the system. When used in conjunction with multigrid, this approach works relatively well for Euler calculations, but appears to be much less effective for turbulent Navier-Stokes calculations due to the highly stretched cells needed to efficiently resolve a high Reynolds number boundary layer.

For Euler calculations, the block-Jacobi matrix preconditioner improves the damping of the convective modes of Runge-Kutta schemes when used with full coarsened multigrid. In comparison to the standard approach, the computational savings using this approach is between a factor of two and three, for both convergence to engineering accuracy and asymptotic convergence.

For turbulent Navier-Stokes flows, the same matrix preconditioner with a semi-coarsened multigrid strategy in which coarsening is performed only in the direction normal to the wall yields a computational saving, relative to the standard approach, of a factor of three. The new scheme provides rapid and robust convergence for two and three dimensions and for all kinds of geometric configuration. The good

behaviour of the numerical computations is explained by examining and evaluating the limits of the stability region of the scheme considering the Euler equations in a first order upwinding scheme. This is a close approximation of what is solved on the coarse levels of the multigrid, since a purely first order numerical dissipation is used and the viscous terms are not significant on these coarse levels. The preconditioned system has a field of values which remains bounded by the largest circle lying inside the stability region of the Runge-Kutta scheme guaranteeing algebraic stability. A slip boundary condition does not effect the analysis which is extended to include it.

For low Mach number applications, a low Mach number preconditioner is incorporated into the numerical dissipation and hence into the block-Jacobi preconditioner to improve the damping of error modes. Convergence to the steady state is enhanced, and the quality of the numerical solution improved for inviscid and turbulent Navier-Stokes calculations. Extension of the stability analysis draws the same conclusion, i.e. an algebraic stability of the scheme as long as the low Mach number preconditioner is symmetric and positive definite. The analysis of the preconditioned Euler equations shows the quite significant effect of the preconditioning on the effectiveness of boundary conditions in eliminating initial transients. For example, boundary conditions based on the Riemann invariants of the Euler equations are found to be reflective in conjunction with preconditioning, whereas they are non-reflecting at the inflow without it.

To take advantage of multi-processor computers, the three dimensional code is fully parallelised and all of the theory is backed up by an extensive selection of two and three dimensional test cases simulating internal and external turbulent flows.

7.2 Further work

Substantial further work will be necessary to validate in more detail the implementation of the turbulence model. Calculations over a flat plate and comparison of the structure of the flow in the inner layer of the boundary layer with experimental data should be done. Also it will be necessary to go through more validations on

three dimensional hybrid grids involving other types of element than those used in this work, - e.g. prisms, pyramids. This will require the generation of meshes using academic or industrial grid generation software. Although there has been excellent progress in the development of such software in the last few years, it nevertheless still requires expert use to produce good grids, especially for viscous flows around complex geometries.

On the stability analysis side, some additional work could also be done to investigate the eventual possibility of integrating in the analysis the second order treatment of the scheme.

The work described in this thesis is a description of the nonlinear code called HYDRA, which is also the basis for additional codes modelling linear unsteady flows and solving adjoint equations for design: the nonlinear flow code is used to obtain the approximate sensitivity of the flow field to small perturbations to the geometry, or used in the analysis of the numerical error in integral functionals such as lift and drag which are two quantities of primary concern on an aircraft for example. The adjoint solution defines the relationship between the error in the functional of interest and the finite volume residual error, which is the extent to which the finite volume solution is not the solution of the original analytic problem. In that context, and since the use of hybrid grids is well suited to the inclusion of adaption, an estimate of the adjoint solution together with the local finite volume residual error could lead to an optimal grid adaptation to obtain the most accurate prediction of a particular functional.

Also another interesting capability would be the modelling of the unsteady aeroelastic behaviour of complex configurations, such as a complete aircraft. Following the approach of Jameson [36] this could be achieved by using an implicit time discretisation, with the resulting equations being solved using the multigrid method.

Finally, since the methodology and numerical analysis is now well established, the extension to other application areas - e.g. acoustics, electromagnetics could also be done.

Bibliography

[1] S. Abarbanel and D. Gottlieb. Optimal time splitting for two- and three-dimensional Navier-Stokes equations with mixed derivatives. *J. of Computational Physics*, 35:1, 1981.

[2] S.R. Allmaras. Analysis of a local matrix preconditioner for the 2-D Navier-Stokes equations. *AIAA 93-3330-CP*, 1993.

[3] W.K. Anderson, R. Rausch, and D. Bonhaus. Implicit multigrid algorithms for incompressible turbulent flows on unstructured grids. *AIAA Paper 95-1740-CP*, 1995.

[4] T. Barth. Randomized multigrid. AIAA Paper 95–0207, 1995.

[5] T.J. Barth. On unstructured grids and solvers. In *VKI Lecture series VKI-LS 1990-03*, March 1990.

[6] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. VKI lecture series, von Karman Institute for Fluid Dynamics, Belgium, 1994.

[7] T.J. Barth and S.W. Linton. An unstructured mesh Newton solver for compressible fluid flow and its parallel implementation. *AIAA Paper 95-0221*, January 1995.

[8] J. Boussinesq. Theorie de l'ecoulement tourbillonant. *Comptes-Rendus de l'Academie des Sciences, 23:46-50*, 1877.

[9] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation*, 31:333–390, 1977.

[10] D.A. Burgess, P.I. Crumpton, and M.B. Giles. A parallel framework for unstructured grid solvers. In K.M. Decker and R.M. Rehmann, Eds., *Programming Environments for Massively Parallel Distributed Systems*, pp. 97–106. Birkhauser, 1994.

[11] P.N. Childs, J.A. Shaw, A.J. Peace, and J.M. Georgala. SAUNA: A system for grid generation and flow simulation using hybrid structured/unstructured grids. In *Computational Fluid Dynamics'92*, pp. 875–882. Hirsch, 1992.

[12] P. Cook, M. McDonald, and M. Firmin. Airfoil RAE 2822 - Pressure distributions, and boundary layer wake measurements. *AGARD AR-138*, p. A6, 1979.

[13] P.I. Crumpton. A cell vertex method for 3d Navier Stokes solutions. Technical Report NA93/09, Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD., 1993.

[14] P.I. Crumpton. An efficient cell vertex method for unstructured tetrahedral grids. Technical report, Oxford University Computing Laboratory, Numerical Analysis Group, 1995.

[15] P.I. Crumpton and M.B. Giles. Aircraft computations using multigrid and an unstructured parallel library. AIAA Paper 95-0210, 1995.

[16] P.I. Crumpton and M.B. Giles. Aircraft computations using multigrid and an unstructured parallel library. Technical report, Oxford University Computing Laboratory, Numerical Analysis Group, 1995.

[17] P.I. Crumpton and M.B. Giles. Implicit time accurate solutions on unstructured dynamic grids. *International Journal for Numerical Methods in Fluids*, 25:1285–1300, 1997.

[18] P.I. Crumpton, M.B. Giles, and G. N. Shrinivas. Design optimisation for complex geometries. 15th ICNMFD Conference, Monterey, USA, June 1996.

[19] P.I. Crumpton, J.A. Mackenzie, and K.W. Morton. Cell vertex algorithms for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 109(1):1–15, 1993.

[20] P.I. Crumpton, P. Moinier, and M.B. Giles. An unstructured algorithm for high Reynolds number flows on highly-stretched grids. In C. Taylor and J. T. Cross, Eds., *Numerical Methods in Laminar and Turbulent Flow*, pp. 561–572. Pineridge Press, 1997.

[21] D.L. Darmofal. Eigenmode analysis of boundary conditions for the one-dimensional preconditioned Euler equations. *AIAA Paper 99-3352*, 1999.

[22] D.L. Darmofal, P. Moinier, and M.B. Giles. Eigenmode analysis of boundary conditions for the one-dimensioned Euler equations. *Journal of Computational Physics*, 1999. submitted.

[23] D.L. Darmofal and P.J. Schmid. The importance of eigenvectors for local preconditioners of the Euler equations. *Journal of Computational Physics*, 127(2), September 1996.

[24] D.L. Darmofal and K. Siu. A robust multigrid algorithm for the Euler equations with local preconditioning and semi-coarsening. *Journal of Computational Physics*, 151:728–756, 1999.

[25] T.A. Driscoll. A Matlab toolbox for Schwartz-Christoffel mapping. *ACM Trans. Math. Software*, pp. 168–186, 1996.

[26] J. Elliott. *Aerodynamic optimization based on the Euler and Navier-Stokes equations using unstructured grids.* PhD thesis, MIT Dept. of Aero. and Astro., 1998.

[27] M.B. Giles. Eigenmode analysis of unsteady one-dimensional Euler equations. ICASE NASA Contract No. NAS1-17130, 1983.

[28] M.B. Giles. Stability analysis of Galerkin/Runge-Kutta Navier–Stokes discretisations on unstructured grids. *Journal of Computational Physics*, 132:201–214, 1997.

[29] B. Gustafsson and A. Sundström. Incompletely parabolic problems in fluid dynamics. *J. of Computational Physics*, 35(2):343–357, 1978.

[30] W. Hackbusch. Multi–grid convergence theory. In W. Hackbusch and U. Trottenberg, Eds., *Multigrid Method Proceedings of the Conference Held at Köln–Porz, November 23–27, 1981*, pp. 177–219. Springer–Verlag, 1981.

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

[32] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 2. John Wiley & Sons, 1990.

[33] H. Iacovides, D.C. Jackson, G. Kelemenis, B.E. Launder, and Y-M Yuan. Recent progress in the experimental investigation of flow and local wall heat transfer in internal cooling passages of gas-turbine blades. In *2nd EF Conference on Turbulent Heat Transfer*, Manchester,UK, 1998.

[34] H. Iacovides and M. Raisee. Recent progress in the computation of flow and heat transfer in internal cooling passages of turbine blades. In *2nd EF Conference on Turbulent Heat Transfer*, Manchester,UK, 1998.

[35] A. Jameson. Transonic aerofoil calculations using the Euler equations. In P.L. Roe, Ed., *IMA Conference on Numerical Methods in Aeronautical Fluid Dynamics*, pp. 289–308, London, 1982. Academic Press.

[36] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils, wings, and helicopter rotors. *AIAA Paper 91-1596*, 1991.

[37] A. Jameson. Analysis and design of numerical scheme for gas dynamics,2: Artificial diffusion and discrete shock structure. *comp. Fluid Dyn.*, 5:1–38, 1995.

[38] A. Jameson and D. Mavriplis. Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *AIAA Journal*, 24(4):611–618, Apr 1986.

[39] Y. Kallinderis, A. Khawaja, and H. McMorris. Hybrid prismatic/tetrahedral grid generation for complex geometries. *AIAA Paper 95-0211*, 1995.

[40] T.J. Kao, T.Y. Su, and N.J. Yu. Navier-Stokes calculations for tranport wing-body configurations with nacell and struts. *AIAA Paper 93-2945*, 1993.

[41] J.F.B.M. Kraaijevanger, H.W.J. Lenferink, and M.N. Spijker. Stepsize restrictions for stability in the numerical solution of ordinary and partial differential equations. *Journal of Computational and Applied Mathematics*, 20:67–81, Nov 1987.

[42] H.O. Kreiss and L. Wu. On the stability definition of difference approximations for the initial boundary value problem. *Applied Numerical Mathematics*, 12:213–227, 1993.

[43] D. Lee. *Local Preconditioning of the Euler Equations.* PhD thesis, University of Michigan, 1996.

[44] D. Lee and B. van Leer. Progress in local preconditioning of the Euler and Navier-Stokes equations. *AIAA Paper 93-3328-CP*, July 1993.

[45] H.W.J. Lenferink and M.N. Spijker. On the use of stability regions in the numerical analysis of initial value problems. *Mathematics of Computation*, 57(195):221–237, 1991.

[46] H. Luo, J. D. Baum, and R. Lohner. Edge-based finite element scheme for the Euler equations. *AIAA Journal*, 32(6), 1994.

[47] J.E. Marsden and M.J. Hoffman. *Basic Complex Analysis.* W.H. Freeman, 1999.

[48] L. Martinelli. *Calculations of Viscous Flows with a multigrid method.* PhD thesis, Dept. of Mech. and Aerospace Eng., Princeton University, 1987.

[49] D. Mavriplis and A. Jameson. Multigrid solution of the Euler equations on unstructured and adaptive meshes. In S. McCormick, Ed., *Proceedings of the Third Copper Mountain Conference on Multigrid Methods: Lecture Notes in Pure and Applied Mathematics.* Marcel Dekker Inc, 1987.

[50] D. J. Mavriplis and V. Venkatakrishnan. Agglomeration multigrid for viscous turbulent flows. AIAA Paper 94-2332, 1994.

[51] D.J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes. *AIAA Paper 97-1952*, 1997.

[52] D.J. Mavriplis. On convergence acceleration techniques for unstructured grids. Technical report, ICASE, 1998. Report No. 98-44.

[53] D.J. Mavriplis and V. Venkatakrishnan. A unified multigrid solver for the Navier-Stokes equations on mixed element meshes. *International Journal of Computational Fluid Dynamics*, 8:247–263, 1997.

[54] P. Moinier and M.B. Giles. Preconditioned Euler and Navier Stokes calculations on unstructured meshes. In M. J. Baines, Ed., *Numerical Methods for Fluid Dynamics VI.* Oxford University Computing Laboratory, 1998.

[55] E. Morano, M.H. Lallemand, M.P. Leclercq, H. Steve, B. Stoufflet, and A. Dervieux. Local iterative upwind methods for steady compressible flows. In *Third European Conference on Multigrid Methods*, p. 227. Springer-Verlag, Berlin 1991, 1990.

[56] K. Morgan, J. Peraire, J. Peiró, and O. Hassan. The computation of three dimensional flows using unstructured grids. *Computer Methods in Applied Mechanics and Engineering*, 87:335–352, 1991.

[57] W.A. Mulder. A new multigrid approach to convection problems. *Journal of Computational Physics*, 83:303–323, 1989.

[58] J.-D. Müller. Coarsening 3-d hybrid meshes for multigrid methods. In *9th Copper Mountain Multigrid Conference*, 1999.

[59] J.-D. Müller and M.B. Giles. Edge-based multigrid scheme for hybrid grids. In M. J. Baines, Ed., *Numerical Methods for Fluid Dynamics VI*. Will Print, Oxford, 1998.

[60] R.H. Ni. Multigrid convergence acceleration techniques for explicit Euler solvers and applications to Navier-Stokes calculations. VKI lecture series, von Karman Institute for Fluid Dynamics, Belgium, 1986.

[61] C. Ollivier-Gooch. Towards problem-independent multigrid convergence rates for unstructured mesh methods I: Inviscid and laminar flows. In *Proceedings of the 6th International Symposium on CFD*, Lake Tahoe, September 1995.

[62] J. Peraire, J. Peiró, and K. Morgan. Finite element multigrid solution of Euler flows past installed aero-engines. *Computational Mechanics*, 11:433–451, 1993.

[63] N.A. Pierce. *Preconditioned Multigrid Methods for Compressible Flow Calulations on Stretched Meshes*. PhD thesis, Oxford University, 1997.

[64] N.A. Pierce and M.B. Giles. Preconditioning compressible flow calculations on stretched meshes. AIAA Paper 96-0889, 1996.

[65] N.A. Pierce and M.B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. *J. of Computational Physics*, 136:425–445, 1997.

[66] D. Jesperson T. Pulliam and P. Buning. Recent enhancements to OVERFLOW. *AIAA Paper 97-0644*, 1997.

[67] S.C. Reddy and L.N. Trefethen. Stability of the method of lines. *Numerische Mathematik*, 62:235–267, 1992.

[68] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial-Value Problems*. Wiley-Interscience, 2nd edition, 1967. Reprint edn (1994) Krieger Publishing Company, Malabar.

[69] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.

[70] M.W. Rubesin and W.C. Rose. The turbulent mean-flow, Reynolds-stress and heat-flux equations in mass-averaged dependent variables. Technical report, NASA-TM-X-62248, 1973.

[71] C. L. Rumsey and V. N. Vatsa. A comparison of the predictive capabilities of several turbulence models using upwinding and central-difference computer codes. *AIAA 93-0192*, 1993.

[72] A. Jameson W. Schmidt and E. Turkel. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA Journal 81-1259*, 1981.

[73] G.N. Shrinivas and M.B. Giles. Application of sensitivity analysis to the redesign of OGV's. In *Proceedings of the IMECE Conference*, 1995.

[74] G.N. Shrinivas and M.B. Giles. OGV tailoring to alleviate pylon-OGV-fan interaction. *ASME Paper 95-GT-198*, 1995.

[75] T. Sonar. On the design of an upwind scheme for compressible flow on general triangulations. *Numerical Algorithms*, 4:135–149, 1993.

[76] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aérospatiale*, 1:5–21, 1994.

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

[78] E. Turkel. Preconditioning-squared methods for multidimensional aerodynamics. *AIAA Paper 97-2025*, 1997.

[79] E. Turkel, V. Vatsa, and R. Radespiel. Preconditioning methods for low-speed flows. *AIAA Paper 96-2460*, 1996.

[80] van J.L.M. Dorsselaer, J.F.B.M. Kraaijevanger, and M.N. Spijker. Linear stability analysis in the numerical solution of initial value problems. *Acta Numerica*, pp. 199–237, 1993.

[81] B. van Leer, W-T. Lee, and K.G. Powell. Sonic–point capturing. AIAA Paper 89-45-CP, The University of Michigan, Department of Aerospace Engineering, 1989.

[82] B. van Leer, W.T. Lee, and P.L. Roe. Characteristic time-stepping or local preconditioning for the Euler equations. *AIAA paper 91-1552-CP*, June 1991.

[83] V.N. Vatsa, M.D. Sanetrick, and E.B. Parlette. Development of a flexible and efficient multigrid-based multi-block flow solver. *AIAA Paper 93-0677*, 1993.

[84] V. Venkatakrishnan and D.J. Mavriplis. Agglomeration multigrid for the three-dimensional Euler equations. ICASE Report, 1994.

[85] S. Ward and Y. Kallinderis. Hybrid prismatic/tetrahedral grid generation for complex 3-D geometries. *AIAA Paper 93-0669*, 1993.

[86] J. Weiss and W. Smith. Preconditioning applied to variable and constant density flows. *AIAA Journal*, 33(11):2050–2057, 1995.

[87] P. Wesseling. *An introduction to multigrid methods.* John Wiley, 1992.

Appendix A

Matrices for stability analysis

The conservative form of the Euler equations in three dimensions is given by

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \tag{A.1}$$

where

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}$$

$$G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix} \quad H = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix}.$$

ρ, u, v, w, p, E denote the density, the three cartesian components of velocity, the pressure, and the total specific energy. Also we have,

$$p = (\gamma - 1)\rho\left(E - \frac{q^2}{2}\right), \quad \text{with} \quad q^2 = u^2 + v^2 + w^2.$$

The system (A.1) may be written in the quasi-linear non-conservative form

$$\frac{\partial Q}{\partial t} + A^x \frac{\partial Q}{\partial x} + A^y \frac{\partial Q}{\partial y} + A^z \frac{\partial Q}{\partial z} = 0, \tag{A.2}$$

by defining the flux Jacobians in the x, y, and z directions $A^x = \frac{\partial F}{\partial Q}$, $A^y = \frac{\partial G}{\partial Q}$, and $A^z = \frac{\partial H}{\partial Q}$. The structure of the Jacobian matrices is quite complicated. Calculation and decomposition are most easily accomplished by transforming to an intermediate set of primitive variables, which are defined for three dimensions by

$$V = \begin{pmatrix} \rho \\ u \\ v \\ w \\ p \end{pmatrix}.$$

By defining the Jacobian matrix of the transformation from conservative to non-conservative variables, as

$$\begin{aligned} M &= \frac{\partial Q}{\partial V} = \left[\frac{\partial Q}{\partial \rho}, \frac{\partial Q}{\partial u}, \frac{\partial Q}{\partial v}, \frac{\partial Q}{\partial w}, \frac{\partial Q}{\partial p}\right] \\ &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ \frac{q^2}{2} & \rho u & \rho v & \rho w & \frac{1}{(\gamma-1)} \end{bmatrix}, \end{aligned}$$

we can rewrite (A.2) as

$$M\frac{\partial V}{\partial t} + A^x M\frac{\partial V}{\partial x} + A^y M\frac{\partial V}{\partial y} + A^z M\frac{\partial V}{\partial z} = 0.$$

If now we multiply by M^{-1}, we then obtain

$$\frac{\partial V}{\partial t} + (M^{-1}A^x M)\frac{\partial V}{\partial x} + (M^{-1}A^y M)\frac{\partial V}{\partial y} + (M^{-1}A^z M)\frac{\partial V}{\partial z} = 0.$$

$$\Leftrightarrow \frac{\partial V}{\partial t} + \bar{A}^x\frac{\partial V}{\partial x} + \bar{A}^y\frac{\partial V}{\partial y} + \bar{A}^z\frac{\partial V}{\partial z} = 0.$$

The primitive Jacobians are related to the conserved variable Jacobians by the similarity transformation $A^x = M\bar{A}^x M^{-1}$, $A^y = M\bar{A}^y M^{-1}$, $A^z = M\bar{A}^z M^{-1}$, with

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ w & 0 & 0 & \rho & 0 \\ \frac{u^2+v^2+w^2}{2} & \rho u & \rho v & \rho w & \frac{1}{(\gamma-1)} \end{bmatrix},$$

and

$$\bar{A}^x = \begin{bmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & 0 & u \end{bmatrix},$$

$$\bar{A}^y = \begin{bmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho c^2 & 0 & v \end{bmatrix},$$

$$\bar{A}^z = \begin{bmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & \frac{1}{\rho} \\ 0 & 0 & 0 & \rho c^2 & w \end{bmatrix}.$$

In a non-conservative form, the system of the Euler equation may be written , with the primitive variable V as

$$\frac{\partial V}{\partial t} + (\hat{A}.\vec{\nabla})V = 0,$$

with

$$\hat{A} = \begin{bmatrix} Q & \rho S_x & \rho S_y & \rho S_z & 0 \\ 0 & Q & 0 & 0 & \frac{S_x}{\rho} \\ 0 & 0 & Q & 0 & \frac{S_y}{\rho} \\ 0 & 0 & 0 & Q & \frac{S_z}{\rho} \\ 0 & \rho S_x c^2 & \rho S_y c^2 & \rho S_z c^2 & Q \end{bmatrix},$$

where c denotes the speed of the sound, and $Q = uS_x + vS_y + wS_z$. The eigenvalues of $\hat{A}$ are obtained as solution of $det(\hat{A} - \lambda I) = 0$, and thus, we get the diagonal matrix

$$\Lambda = \begin{bmatrix} Q & 0 & 0 & 0 & 0 \\ 0 & Q & 0 & 0 & 0 \\ 0 & 0 & Q & 0 & 0 \\ 0 & 0 & 0 & Q + cS & 0 \\ 0 & 0 & 0 & 0 & Q - cS \end{bmatrix},$$

where $S^2 = S_x^2 + S_y^2 + S_z^2$. Also, the right eigenvectors of $\hat{A}$ are the columns of

$$L = \begin{bmatrix} n_x & n_y & n_z & 1 & 1 \\ 0 & -cn_z & cn_y & cn_x & -cn_x \\ cn_z & 0 & -cn_x & cn_y & -cn_y \\ -cn_y & cn_x & 0 & cn_z & -cn_z \\ 0 & 0 & 0 & c^2 & c^2 \end{bmatrix},$$

where n_x, n_y, and n_z are the components of the unit normal

$$n_x = \frac{S_x}{S}, \quad n_y = \frac{S_y}{S}, \quad n_z = \frac{S_z}{S}.$$

The left eigenvectors of $\hat{A}$ are the rows of

$$L^{-1} = \begin{bmatrix} n_x & 0 & \frac{n_z}{c} & -\frac{n_y}{c} & -\frac{n_x}{c^2} \\ n_y & -\frac{n_z}{c} & 0 & \frac{n_x}{c} & -\frac{n_y}{c^2} \\ n_z & \frac{n_y}{c} & -\frac{n_x}{c} & 0 & -\frac{n_z}{c^2} \\ 0 & \frac{n_x}{2c} & \frac{n_y}{2c} & \frac{n_z}{2c} & \frac{1}{2c^2} \\ 0 & -\frac{n_x}{2c} & -\frac{n_y}{2c} & -\frac{n_z}{2c} & \frac{1}{2c^2} \end{bmatrix}.$$

Finally, $\hat{A}$ can be decomposed as

$$\hat{A} = L\Lambda L^{-1}.$$