Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes

Niles A. Pierce
Christ Church

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

Trinity Term 1997

Dedicated to my parents

Michael K. Pierce and Elizabeth A. Pierce

Oxford, August 1997

Acknowledgments

In a tangible sense, this thesis represents the conclusion of a twenty year episode in my life dedicated almost entirely to formal education. Relaxing amidst the initial sensations of satisfaction, amazement and relief that accompany this unique moment of closure, I feel incredibly fortunate to have shared these experiences with so many wonderful friends, family members and teachers.

During my time at Oxford, I have had the great pleasure of working with Mike Giles, who has been a constant source of good ideas and patient guidance, in addition to extending to me the freedom to research and travel with (at times) equal abandon. I am grateful to him for all his hard work on my behalf during the last four years, including most recently, his detailed reading of this thesis. I also feel honored to have had as my thesis examiners Prof. Charles Hirsch of Vrije Universiteit Brussel and Prof. Bill Morton of Oxford University, who generously agreed to shoulder the burden of performing this important function. Finally, I wish to express my thanks to the Rhodes Trust, who have made my stay in Oxford possible.

From these days at the House, I will miss the conviviality of the many fine meals shared with Paul Waldau and Laurie Claus and the thrill of participating in the annual campaign of the ChCh Football Club. Also missed will be the countless evenings spent lolling in conversation with Siddhartha Mukherjee, Erez Kalir, Peter Beinart, Davide Panagia, Amu Singh and Stephanie Markovits. At the lab, I have enjoyed the company of many fellow researchers, including Frederico Campos, Nick Birkett, Paul Crumpton, Pierre Moinier, Russ Cummings, Fang-Pang Lin, Lyon Lanerolle, Gun Shrinivas, Dave Burgess, Mike Rudgyard, Mark Embree, Nigel Clarke, Paul Houston, Richard Burrows and Lei Chen.

Owing to the extraordinary good will of my friends and colleagues at Princeton, I was fortunate to be able to spend nearly a year of my graduate studies in collaborative efforts there. I am grateful to Luigi Martinelli for the generosity that he has always displayed in sharing his knowledge and for the many occasions on which I have been invited to join his family as he demonstrated his proficiency with the

barbecue. Antony Jameson has been a friend and mentor since I first performed my undergraduate research under his supervision, and continues to be an inspiration and a valued source of encouragement. I will remember fondly the many late-night CFD parties with Juan Alonso and the continual banter during periods of overlap with the day-time crew: Andrey Belov, Scott Sheffer, Chong Kim, Paul Lin, Geoff Cowles, Bob Anderson, James Reuther, Tim Baker and Jo Ann Love.

There are two teachers from my pre-collegiate education to whom I am particularly indebted for their lasting contributions to my life. The first is my mother, who has dedicated herself to providing her children with a Renaissance education, and who painstakingly impressed upon me the value of well-crafted prose. The second is my high school math teacher of three years, Mr. Bob Farrimond, who in addition to mathematics, taught an approach to learning that became a crucial ingredient in my subsequent studies. His untimely passing has deeply saddened the many students who shared an appreciation of his unique sense of humor and gentle personality.

With my friends of many years, Eric Patterson, Jerry Attlan, Dev Lal and Brian Johnson, I have shared many diversely fulfilling experiences, and my connection with each has been a constant source of satisfaction as well as sustenance. To Gillian Harkness, I owe so much of the happiness of these last two years. She has been both my greatest distraction and my staunchest supporter as I grappled with the challenges of the graduate research experience. My final thanks must be reserved for my mother and father, who have nurtured me since birth, and for my siblings, Alice, Lilly Bee and Marshall, who have given me all the joys of being a big brother. I know no better feeling than coming home to family.

Abstract

Preconditioned Multigrid Methods
for Compressible Flow Calculations
on Stretched Meshes

Niles A. Pierce
Christ Church

Doctor of Philosophy
Trinity Term 1997

Efficient preconditioned multigrid methods are developed for both inviscid and viscous flow applications. The work is motivated by the mixed results obtained using the standard approach of scalar preconditioning and full coarsened multigrid, which performs well for Euler calculations on moderately stretched meshes but is far less effective for turbulent Navier–Stokes calculations, when the cell stretching becomes severe. In the inviscid case, numerical studies of the preconditioned Fourier footprints demonstrate that a block-Jacobi matrix preconditioner substantially improves the damping and propagative efficiency of Runge–Kutta time-stepping schemes for use with full coarsened multigrid, providing computational savings of approximately a factor of three over the standard approach. In the viscous case, determination of the analytic expressions for the preconditioned Fourier footprints in an asymptotically stretched boundary layer cell reveals that all error modes can be effectively damped using a combination of block-Jacobi preconditioning and a J-coarsened multigrid strategy, in which coarsening is performed only in the direction normal to the wall. This new approach provides rapid convergence to machine accuracy for turbulent Navier–Stokes calculations in both two and three dimensions, yielding computational savings of roughly an order of magnitude compared to the standard method.

Contents

List of Figures

List of Tables

Nomenclature

Roman Symbols

a	convection coefficient for scalar convection–diffusion equation
A_x, A_y	inviscid flux Jacobians
$\|A_x\|, \|A_y\|$	modulus of A defined by $T\|\Lambda\|T^{-1}$
B_{xx}, B_{yy}, B_{xy}	viscous flux Jacobians
c	speed of sound
C_x, C_y	$1 - \cos\theta_x, 1 - \cos\theta_y$
i, j	mesh indices
$\hat{\imath}$	$\sqrt{-1}$
K	number of multigrid levels
L_t	Runge–Kutta operator
M	Mach number
N	cost of relaxation on the fine mesh
n	time step index
P	preconditioner
P_M	block-Jacobi matrix preconditioner
P_S	scalar preconditioner equivalent to standard local time step
P_*	low-Mach number preconditioner
Pr	Prandtl number
R	spatial residual operator
R_M	spatial residual operator with matrix dissipation
R_S	spatial residual operator with scalar dissipation
$Re_{\Delta x}$	cell Reynolds number

Re_L	free stream Reynolds number based on chord
s_x, s_y	$\sin\theta_x, \sin\theta_y$
T, T^{-1}	right and left eigenvector matrices of A
t	time coordinate
Δt	local time step
u	solution variable for scalar convection–diffusion equation
u, v	Cartesian velocity components
W	conserved variable vector for Euler and Navier–Stokes equations
$\widehat{W}$	temporal Fourier amplitude vector
x, y	Cartesian coordinates
$\Delta x, \Delta y$	mesh spacings in Cartesian coordinates
Z	Fourier symbol of the spatial residual operator R
z	variable in the complex plane

Greek Symbols

γ	ratio of specific heats
θ_x, θ_y	Fourier angles
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
ν	diffusion coefficient for scalar convection–diffusion equation or kinematic viscosity for Navier–Stokes equations
$\rho(A)$	spectral radius of A
σ	Courant number
$\lvert\psi\rvert$	amplification factor of Runge–Kutta scheme
Ω	characteristic variable vector

Chapter 1

Introduction

In broad terms, the field of computational fluid dynamics has developed in response to the need for accurate, efficient and robust numerical algorithms for solving increasingly complete descriptions of fluid motion over increasingly complex geometries. The present work focuses entirely on the efficiency aspects of this pursuit for the two systems of governing equations that have been of principal interest for aeronautical applications during the last two decades: the Euler equations, describing inviscid rotational compressible flow, and the Reynolds Averaged Navier–Stokes equations (supplemented with an appropriate turbulence model), describing viscous turbulent compressible flow. There is a significant disparity in the degree to which existing methods have so far succeeded in efficiently producing solutions to these two systems of equations. On the one hand, techniques developed for the Euler equations are already relatively effective, so that while both the need and opportunity for further improvement are significant, they do not appear overwhelming. On the other hand, the development of efficient numerical methods for solution of the Navier–Stokes equations remains one of the ongoing challenges in the field of computational fluid dynamics. Dramatic improvements over the performance of existing methods will be necessary before this area of research may be considered satisfactorily resolved.

1.1 The Viscous Challenge

The difficulty for viscous calculations stems from the need to use a computational mesh that is highly resolved in the direction normal to the wall in order to accurately represent the steep gradients in high Reynolds number boundary layers. This requirement proves problematic both because the resulting high aspect ratio cells greatly reduce the efficiency of existing numerical algorithms and because the overall increase in the number of mesh points stretches the capability of existing computational hardware for problems of practical aerodynamic interest. The first difficulty suggests that the design of an appropriate numerical algorithm must be based on a careful assessment of the interaction between the discrete method, the computational mesh and the physics of the viscous flow. The second difficulty places a premium on striking the right balance between the operation count, storage requirements and parallel scalability of the method.

Since the relevant problem size will continue to increase as fast as hardware constraints will allow, it is critical that the convergence rate of the method should be insensitive to the number of unknowns. The general solution strategy that appears most promising in this regard is multigrid, for which grid-independent convergence rates have been proven for elliptic operators [20, 5, 53, 10, 23]. Although no rigorous extension of this theory has emerged for problems involving a hyperbolic component, methods based on multigrid have proven highly effective for inviscid calculations with the Euler equations [52, 27, 29] and remain the most attractive approach for Navier–Stokes calculations despite the widely observed performance breakdown in the presence of boundary layer anisotropy.

In the present work, the focus will be placed on developing efficient multigrid methods for steady transonic flow calculations, which are of paramount importance in modeling supercritical flight at cruise conditions. Steady solutions to the Euler and Navier–Stokes equations are obtained by time-marching the unsteady systems until the time-derivative terms have become sufficiently small to ensure the desired degree of steadiness in the solution. Numerically, a steady state is achieved by

eliminating transient error modes either by damping or by expulsion from the computational domain. Since the transient solution is not of interest, multigrid can be employed to accelerate convergence to a steady state without concern for the loss of time-accuracy. Classical multigrid techniques developed for elliptic problems transfer the low frequency errors in the solution to a succession of coarser meshes where they become high frequency errors that are more effectively smoothed by traditional relaxation methods. For the unsteady Euler and Navier–Stokes equations, which exhibit both parabolic and hyperbolic properties in their discrete formulations, the coarse meshes in the multigrid cycle serve the dual role of enhancing both damping and propagation of error modes. Since damping is essentially a local process and error expulsion a global one (requiring disturbances to propagate across the domain to a far field boundary), it is the damping properties of the relaxation scheme that are most critical for ensuring insensitivity to problem size. The propagative efficiency of the relaxation method remains important to the performance of the multigrid algorithm, but is nonetheless a second priority during the investigations that follow.

Efficient multigrid performance hinges on the ability of the relaxation scheme to eliminate on the current mesh all error modes that cannot be resolved without aliasing on the next coarser mesh in the cycle. The choice between an explicit or an implicit relaxation scheme to drive the multigrid algorithm requires consideration of the computational trade-offs, in addition to the relative damping and propagative efficiencies of the approaches. Explicit schemes offer a low operation count, low storage requirements and good parallel scalability but suffer from the limited stability imposed by the CFL condition [15, 63]. Alternatively, implicit schemes theoretically offer unconditional stability but are more computationally intensive, require a heavy storage overhead and are more difficult to parallelize efficiently. In practice, direct inversion is infeasible for large problems due to a high operation count, so that some approximate factorization such as ADI or LU must be employed [22, 7, 62, 37, 38]. The resulting factorization errors effectively limit the convergence of the scheme when very large time steps are employed so that it is not possible to fully capitalize on the potential benefits of unconditional stability [38]. Given these

circumstances, it therefore seems advantageous to adopt an explicit approach if a scheme with suitable properties can be designed.

A popular explicit multigrid smoother is the semi-discrete scheme proposed by Jameson *et al.* [36] which uses multi-stage Runge–Kutta time-stepping to integrate the system of ordinary differential equations resulting from the spatial discretization. In accordance with the requirements for efficient multigrid performance, the coefficients of the Runge–Kutta scheme are chosen to promote rapid damping and propagation of error modes [27, 78]. This is accomplished by ensuring that the amplification factor is small in regions of the complex plane where the residual eigenvalues corresponding to high frequency modes are concentrated, as well as by providing large stability limits along the imaginary and negative real axes. Multigrid solvers based on this approach represent an important schematic innovation in enabling large and complex Euler calculations to be performed as a routine part of the aerodynamic design procedure [27, 29]. However, despite the favorable convergence rates observed for Euler computations, the standard approach does not satisfy all the requirements for efficient multigrid performance. These shortcomings become far more evident when the approach is applied to Navier–Stokes calculations.

1.2 Diagnosis of Multigrid Breakdown

The hierarchy of factors leading to multigrid inefficiency are illustrated in Fig. 1.1. The two fundamental causes of degraded multigrid performance for both the Euler and Navier–Stokes equations are stiffness in the discrete system and decoupling of modes in one or more coordinate directions. These two problems manifest themselves in an 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 be neither damped nor propagated efficiently by the multi-stage Runge–Kutta scheme.

For Euler computations, discrete stiffness results primarily from the inherent disparity in the propagative speeds of convective and acoustic modes. The standard scalar preconditioner (local time step) [40] is fundamentally unable to address this

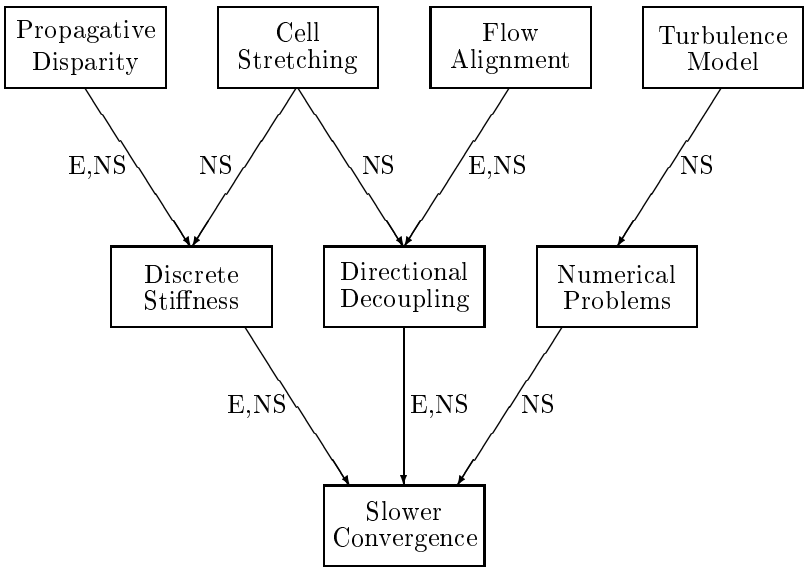


Figure 1.1 Diagnosis of multigrid breakdown for the Euler and Navier–Stokes equations.

type of matrix stiffness but does help to reduce stiffness produced by local variation in scalar quantities such as spectral radius and cell size. This shortcoming is relatively innocuous for inviscid transonic flows, since propagative stiffness is only substantial near the stagnation point, at shocks and across the sonic line. Directional decoupling in Euler computations results primarily from alignment of the flow with the computational mesh, which causes some convective modes to decouple in the transverse direction. In practice, these shortcomings have not prevented the attainment of sufficiently rapid convergence to meet industrial requirements for inviscid transonic flow calculations [32], and although improvements are possible, they do not represent a substantial concern to the CFD community.

For Navier–Stokes computations of high Reynolds number flows, the problems resulting from the disparity in propagative speeds and from flow alignment still persist, but a far more serious source of difficulties is introduced by the high aspect ratio cells inside the boundary layer. These highly stretched cells increase the discrete stiffness of the system by several orders of magnitude so that the entire convective Fourier footprints collapse to the origin while decoupling the acoustic modes from the streamwise coordinate direction. Under these circumstances, the multigrid algorithm is extremely inefficient at eliminating a large fraction of the error modes

which could potentially exist in the solution.

Convergence problems for high Reynolds number Navier–Stokes applications are also compounded by the need to incorporate a turbulence model. Popular algebraic models are notorious for introducing a disruptive blinking phenomenon into the convergence process as the reference distance migrates back and forth between neighboring cells. Alternatively, adopting a one or two-equation model requires the solution of turbulent transport equations that incorporate production and destruction source terms that are both temperamental and stiff. However, recent efforts have demonstrated that turbulent transport equations can be successfully discretized using a multigrid approach without interfering with the convergence process of the flow equations [43, 60].

For three-dimensional calculations, mesh quality can also play a substantial role in determining the convergence rate of a calculation. In particular, single block wing meshes invariably have a topological singularity at the wing tip which can adversely affect both convergence and robustness. Unlike the aforementioned problems, this difficulty arises not from a property of the governing flow equations but from the geometric complexity of the engineering application. The pragmatic challenges of obtaining solutions in the face of poor mesh quality will not be considered in the present work.

1.3 A Unified Approach to Preconditioning and Multigrid

One means of combatting discrete stiffness in the Euler and Navier–Stokes equations is the use of a matrix time step or preconditioner [13, 71, 82, 77, 50, 39, 73]. In the present work, the preconditioner is viewed as a mechanism for overcoming discrete stiffness by clustering the residual eigenvalues away from the origin into a region of the complex plane for which the multi-stage scheme can provide rapid damping and propagation of the corresponding error modes [1, 59]. This is an alternative viewpoint to the one invoked by those who have developed preconditioners for low-Mach

number and incompressible flows [13, 72, 77, 12, 75, 83] where the focus is placed on eliminating analytic stiffness arising from the inherent propagative disparities in the limit of vanishing Mach number. Although the theoretical advantages of matrix preconditioning have been evident for some time, the development of robust numerical implementations has proven challenging [72, 77, 79, 18, 49, 80]. Therefore, the choice of an appropriate matrix preconditioner must be governed by the dual objectives of maximizing the benefits to convergence while minimizing the adverse effects on robustness.

For the problem of directional decoupling, one possible remedy is the use of directional coarsening multigrid algorithms [51]. In certain cases, preconditioning methods can also play a role in alleviating the problems associated with directional decoupling [59, 60, 2]. Under these circumstances, the interaction between the preconditioner and the multigrid coarsening strategy is critical. Therefore, it is imperative that the two components of the scheme are considered simultaneously when attempting to design efficient preconditioned multigrid methods.

Allmaras provided a systematic examination of the damping requirements for relaxation methods used in conjunction with both traditional full coarsened multigrid and the semi-coarsening multigrid algorithm of Mulder [2, 51]. Using full coarsened multigrid in two dimensions, only modes which are low frequency in both mesh directions can be resolved on the coarser grids, so that the relaxation scheme must eliminate all high frequency modes, and also those modes that are high frequency in one mesh direction and low frequency in the other. For use in conjunction with an explicit Runge–Kutta scheme, Allmaras recommends an implicit ADI preconditioner because explicit methods are notoriously poor at damping modes with a low frequency component [2]. Buelow *et al.* [11] and Venkateswaran *et al.* [80] have employed related strategies based on a low-Mach number matrix preconditioner [81] and ADI relaxation.

Alternatively, the semi-coarsening algorithm proposed by Mulder [51] coarsens separately in each mesh direction and therefore reduces the region of Fourier space for which the relaxation scheme on each mesh must successfully damp error modes.

To obtain an $O(N)$ method for a three-dimensional calculation in which N is the cost of relaxation on the fine mesh, Mulder defined a restriction and prolongation structure in which not all grids are coarsened in every direction. For two-dimensional grids that are coarsened separately in both directions, only those modes that are high frequency in both mesh directions need be damped by the relaxation scheme. For this purpose, Allmaras suggests the point-implicit block-Jacobi preconditioner proposed by Morano *et al.* [50] that has previously been demonstrated to be effective in clustering high frequency eigenvalues away from the origin [1]. For grids that are not coarsened in one of the mesh directions, Allmaras proposes using a semi-implicit line-Jacobi preconditioner in that direction [2].

These strategies for preconditioning in the context of both full and semi-coarsened multigrid are well-conceived. The principal drawback of these approaches is that they do not take into account any properties of the viscous flow physics or the computational mesh which could potentially reduce the demands placed on the numerical method. By attempting to provide a completely general approach, both alternatives incur substantial computational overheads. The drawback to implicit preconditioning for full coarsened multigrid is the associated increase in operation count, storage overhead and difficulty in efficient parallelization. The drawback to a semi-coarsened approach is primarily the increase in operation count: for a three-dimensional computation, the costs for full coarsened V and W-cycles are bounded by $\frac{9}{7}N$ and $\frac{4}{3}N$, respectively, while for semi-coarsening, the cost of a V-Cycle is bounded by $8N$ and a W-cycle is no longer $O(N)$ [51].

The purpose of the present work is to analyze and implement two less expensive preconditioned multigrid methods, one of which is designed to perform efficiently for Euler calculations, and the other for turbulent Navier–Stokes calculations. In each case, the intention is to strike an appropriate balance between the additional computational overhead incurred and the degree to which improvement over existing methods can reasonably be expected.

For the Euler equations, existing multigrid solvers employing a standard scalar preconditioner [40] and a full coarsened strategy routinely demonstrate relatively

good convergence rates [32] despite their failure to satisfy all the damping and propagation requirements for efficient multigrid performance. The widespread success using this approach suggests that the computational challenges arising from discrete stiffness and directional decoupling are not particularly severe for Euler calculations. Therefore, it is undesirable to pursue alternative methods that substantially increase the cost and complexity of each multigrid cycle in order to ensure that these efficiency criteria are completely satisfied. Instead, it seems reasonable to view these efficiency requirements as a worthwhile objective to be attained to the highest degree possible while maintaining the desirable cost and complexity properties of a full coarsened approach. The sole responsibility for improving convergence will therefore be placed on the development of an appropriate matrix preconditioning approach.

Numerical studies of the preconditioned Fourier footprints are used to demonstrate that substantial improvements in full coarsened multigrid performance can be achieved by replacing the standard scalar preconditioner with the point-implicit block-Jacobi matrix preconditioner proposed by Morano *et al.* [50] and suggested by Allmaras for use with the more expensive semi-coarsened strategy [2]. Several authors have successfully applied this preconditioner to Euler multigrid calculations on unstructured meshes [50, 19, 54]. For transonic airfoil calculations on fully-resolved structured meshes, the block-Jacobi matrix preconditioner yields computational savings of roughly a factor of three over the standard scalar preconditioner when used in conjunction with full coarsened multigrid [59, 60, 58]. This preconditioned multigrid method has proven robust for inviscid calculations and represents a suitable starting point for addressing the more pressing challenge of improving the convergence of viscous methods.

In the case of turbulent Navier–Stokes calculations, the standard combination of scalar preconditioning and full coarsened multigrid has proven largely inadequate for coping with the problems arising from stretched boundary layer cells. Consequently, some compromise in the simplicity of the standard approach seems justifiable, and modifications to both the preconditioner and the coarsening strategy will be considered. To identify the specific nature of the difficulties and assist in designing

an inexpensive algorithm that does not falter in the presence of boundary layer anisotropy, the present work examines the analytic form of the two-dimensional preconditioned Fourier footprints inside an asymptotically stretched boundary layer cell [59, 60].

This analysis reveals the asymptotic dependence of the residual eigenvalues on the two Fourier angles, thus exposing the clustering properties of the preconditioned algorithm. In particular, it is demonstrated that the balance between streamwise convection and normal diffusion inside the boundary layer enables a point-implicit block-Jacobi preconditioner to ensure that even those convective modes with a low frequency component in one mesh direction are effectively damped [59]. A simple modification of the full coarsened algorithm to a J-coarsened strategy, in which coarsening is performed only in the direction normal to the wall, further ensures that all acoustic modes are damped [60]. The theoretical validity of the approach is then extended to three-dimensional Navier–Stokes calculations by assuming the worst case scenario in which the Fourier footprint is decoupled from the third coordinate direction [61]. This argument suggests that it is unnecessary to resort to either an implicit preconditioner or a complete semi-coarsening algorithm to produce a preconditioned multigrid method that ensures that all error modes are effectively damped inside the boundary layer.

For turbulent Navier–Stokes calculations of transonic airfoil flows on fully-resolved meshes, the combination of block-Jacobi preconditioning and J-coarsened multigrid yields computational savings of roughly an order of magnitude over existing methods that rely on the standard combination of scalar preconditioning and full coarsened multigrid [60, 58]. This new preconditioned multigrid method has been extended to three-dimensional turbulent Navier–Stokes calculations and has been shown to provide essentially the same convergence rates as for two-dimensional flows [61]. The applicability of the new approach extends beyond steady state flow analysis to other areas of active research that rely on a steady state solver as an inner kernel. These include both optimal design by adjoint methods [30, 35] and unsteady methods based on an inner multigrid iteration [31, 57, 56].

1.4 Summary

The main body of the dissertation is divided into two chapters corresponding to the development of separate preconditioned multigrid methods for Euler and turbulent Navier–Stokes applications. Supporting documentation that is necessary for completeness but that does not contribute substantively to the primary thrust of the discussion is incorporated in a series of appendices.

The chapter on Euler applications begins with a number of preliminary points, including a description of the scheme formulation, definitions of the scalar and matrix preconditioners and a discussion of stability considerations that restrict the possible combinations of preconditioner and numerical dissipation. The Fourier footprint is then described as a mechanism for assessing the effectiveness of preconditioners in clustering residual eigenvalues for rapid damping and propagation by a multi-stage Runge–Kutta scheme. Numerical studies of the two-dimensional preconditioned Fourier footprints are then used to illustrate the improved damping and propagative clustering provided by the block-Jacobi matrix preconditioner. The numerical implementation is then briefly described before a demonstration of robust convergence acceleration is provided for standard transonic airfoil test cases.

The chapter on turbulent Navier–Stokes applications begins by extending the earlier definitions to incorporate the viscous terms. Analytic expressions for the two-dimensional preconditioned Fourier footprints in an asymptotically stretched boundary layer cell are then obtained for both scalar and matrix preconditioners. Numerically generated footprints are used to verify the correctness of the asymptotic procedure and the validity of the asymptotic approximation for addressing problems arising from practical high Reynolds number applications. After exposing the inadequacy of the standard approach of scalar preconditioning and full coarsened multigrid for viscous flow calculations, the analytic footprints are used to motivate an alternative method combining block-Jacobi matrix preconditioning and J-coarsened multigrid. Following an extension of this analysis to the three-dimensional case, the numerical implementation is summarized and the impact on convergence is demon-

strated for both two and three-dimensional turbulent Navier–Stokes calculations.

A concluding chapter summarizes the main results of the research and provides some suggestions for future work. The first appendix then examines the damping and propagative mechanisms for a scalar convection–diffusion equation to demonstrate the manner in which both damping and propagative information can be obtained from Fourier footprints. The remaining appendices document the implementation of the Euler and turbulent Navier–Stokes flow solvers, describing successively the discretization, the preconditioners, the multigrid algorithms, and the turbulence models.

Chapter 2

Euler Applications

To assess the properties of the scalar and matrix preconditioners, Fourier analysis is used to decompose the transient solution errors into modal components which can then be examined individually. This analytic approach is based on a local linearization of the flow on a mesh with constant spacing and periodic boundary conditions. The validity of the analysis then depends on the degree to which the true local behavior of the solution can be modeled under these assumptions. For computational problems not incorporating mesh singularities, numerical results suggest that Fourier analysis does provide a useful indicator of scheme performance characteristics.

2.1 Preliminaries

Some basic definitions must be introduced before proceeding to the examination of preconditioned multigrid methods for inviscid applications. These definitions will be accompanied by a few observations that will influence the focus of the ensuing investigations.

Scheme Description

Inviscid analysis is based on the two-dimensional Euler equations in Cartesian coordinates, linearized with respect to perturbations to a uniform flow,

$$\frac{\partial W}{\partial t} + A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} = 0,$$

where W is the state vector and A_x and A_y are the uniform inviscid flux Jacobians. A preconditioned semi-discrete finite volume discretization of this system appears as

$$L_t W + PR(W) = 0, \tag{2.1}$$

where $R(W)$ is the residual vector of the spatial discretization, L_t represents the multi-stage Runge–Kutta operator and P is the preconditioner, which has the dimension of time and plays the role of a time step. The transient solution is not of interest for steady applications, so the preconditioner and the coefficients of the Runge–Kutta operator can be chosen to promote rapid convergence without regard for time-accuracy. The steady solution admitted by the spatial discretization is unaffected by the choice of preconditioner, as long as P is non-singular, since the system reduces to $R(W) = 0$ when the unsteady terms have vanished.

For the analysis that follows, R is taken to be the standard first order linear operator

$$R = \frac{A_x}{2\Delta x}\delta_{2x} - \frac{|A_x|}{2\Delta x}\delta_{xx} + \frac{A_y}{2\Delta y}\delta_{2y} - \frac{|A_y|}{2\Delta y}\delta_{yy}, \tag{2.2}$$

where numerical dissipation corresponding to upwinding has been introduced using a Roe linearization [64]. Numerical dissipation of this type is a form of matrix dissipation [70], in which each characteristic field is upwinded by introducing dissipation scaled by the associated characteristic speed. A related form of scalar dissipation may be obtained by replacing $|A_x|$ and $|A_y|$ in (2.2) by their spectral radii $\rho(A_x)$ and $\rho(A_y)$, so that the dissipation for each characteristic is instead scaled by the maximum characteristic speed [36]. This approach is computationally less expensive since it avoids matrix operations, but is also less accurate as it introduces unnecessarily large amounts of dissipation into all but one of the characteristic fields. The implications for stability and convergence are compared for these two alternative schemes, but detailed analysis will focus on the more accurate matrix dissipation approach.

Assuming constant γ, the three independent parameters that govern the discrete steady Euler residual are the Mach number, reciprocal cell aspect ratio and flow

angle:

$$M = \frac{\sqrt{u^2 + v^2}}{c}, \qquad \frac{\Delta y}{\Delta x}, \qquad \frac{v}{u}.$$

A Cartesian mesh is assumed to simplify notation, but the theory extends naturally to real applications using either structured or unstructured meshes.

Scalar Preconditioner

The standard scalar preconditioner (local time step) [40] used for Euler calculations is based on the spectral radii of the flux Jacobians

$$P_S^{-1} = \frac{1}{\sigma}\left(\frac{\rho(A_x)}{\Delta x} + \frac{\rho(A_y)}{\Delta y}\right),$$

where the Courant number σ reflects the extent of the Runge–Kutta stability region along the imaginary axis in the complex plane. In comparison with a uniform global time step, this local stability estimate defines a suitable scalar preconditioner for the Euler equations that reduces stiffness resulting from variation in spectral radius and cell size throughout the mesh [40].

Matrix Preconditioner

The block-Jacobi matrix preconditioner is obtained from the discrete residual operator (2.2) by extracting the terms corresponding to the central node in the stencil

$$P_M^{-1} = \frac{1}{\sigma}\left(\frac{|A_x|}{\Delta x} + \frac{|A_y|}{\Delta y}\right).$$

It has been demonstrated [60] that the preconditioner takes a fundamentally similar form for a switched scheme [36] based on the same Roe linearization [64]. This compatibility and related numerical experiments suggest that it is acceptable to base the preconditioner on a first order discretization even when using higher resolution switched and limited schemes.

Stability Considerations

It is essential to note that the choice of either a scalar or matrix preconditioner cannot be made independently from the choice of numerical dissipation. This may

be understood by considering the necessary and sufficient stability condition

$$\Delta t \leq \min\left(\frac{2\nu}{a^2}, \frac{\Delta x^2}{2\nu}\right)$$

for the scalar convection–diffusion equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}$$

discretized using central differences in space and forward differences in time. This discretization can be used to represent first order upwinding of a scalar convection equation if the diffusion coefficient is chosen to correspond to the appropriate form of numerical dissipation $\nu = \frac{|a|\Delta x}{2}$. In this case, the stability requirement then reduces to the standard CFL condition $\Delta t \leq \frac{\Delta x}{|a|}$.

Using characteristic-based matrix dissipation, the linearized Euler equations can be expressed in the corresponding representation

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} = \frac{\Delta x}{2}|A|\frac{\partial^2 W}{\partial x^2}.$$

This one-dimensional system can be decoupled into scalar characteristic equations

$$\frac{\partial \Omega}{\partial t} + \Lambda\frac{\partial \Omega}{\partial x} = \frac{\Delta x}{2}|\Lambda|\frac{\partial^2 \Omega}{\partial x^2},$$

using an eigenvector decomposition of the flux Jacobian $A = T\Lambda T^{-1}$ to produce the characteristic variables $\Omega = T^{-1}W$, where Λ is a diagonal eigenvalue matrix. Applying the convection–diffusion stability requirement separately to each characteristic equation leads to the limit $\Delta t_k \leq \frac{\Delta x}{|\lambda_k|}$ for the k-th characteristic, where λ_k is the corresponding eigenvalue. The scalar preconditioner is stable but sub-optimal since all characteristics evolve with $\Delta t_k = \frac{\Delta x}{\max_k(|\lambda_k|)}$. The matrix preconditioner is stable and also optimal in one dimension, since $\Delta t_k = \frac{\Delta x}{|\lambda_k|}$, and each characteristic field is evolving at its stability limit.

Alternatively, using scalar dissipation, the equivalent expression for the Euler equations becomes

$$\frac{\partial W}{\partial t} + A\frac{\partial W}{\partial x} = \frac{\Delta x}{2}\rho(A)\frac{\partial^2 W}{\partial x^2},$$

where $\rho(A) \equiv \max_k(|\lambda_k|)$ is the spectral radius of the flux Jacobian. This system can still be decoupled into characteristic equations using the same transformation

as above. However, the stability requirement for all characteristics is now just the standard scalar CFL condition, $\Delta t_k \leq \frac{\Delta x}{\rho(A)}$. As a result, the matrix preconditioner is unstable when used in conjunction with scalar dissipation based on the spectral radius and only the scalar preconditioner is appropriate. Out of the four possible combinations of scalar (S) and matrix (M) preconditioners and numerical dissipation schemes, the three stable combinations are denoted $P_S R_M$, $P_M R_M$ and $P_S R_S$. The behavior of the scalar preconditioner applied to scalar dissipation ($P_S R_S$) will only be considered briefly to illuminate the properties of the other two combinations since it produces a different steady state solution and it is undesirable to compromise accuracy for the purposes of convergence acceleration.

Fourier Footprint

In the context of a semi-discrete scheme (2.1), the Fourier footprint of the spatial discretization is critical in revealing the effectiveness of the Runge–Kutta operator in damping and propagating error modes. The footprint is found by substituting a semi-discrete Fourier mode of the form

$$W_{i,j} = \widehat{W}(t) e^{\hat{\imath}(i\theta_x + j\theta_y)}$$

into the spatial residual operator (2.2) where (i, j) are the mesh indices and (θ_x, θ_y) are the Fourier angles in the corresponding directions. The Fourier amplitude $\widehat{W}(t)$ satisfies the evolution equation

$$L_t \widehat{W} + PZ\widehat{W} = 0,$$

where Z is the Fourier symbol of the residual operator

$$\begin{aligned} Z(\theta_x, \theta_y) &= \hat{\imath}\frac{A_x}{\Delta x}\sin\theta_x + \frac{|A_x|}{\Delta x}(1 - \cos\theta_x) \\ &+ \hat{\imath}\frac{A_y}{\Delta y}\sin\theta_y + \frac{|A_y|}{\Delta y}(1 - \cos\theta_y). \end{aligned}$$

The Fourier footprint is defined as the eigenvalues of the matrix PZ, which are functions of the Fourier angles. The distribution of these residual eigenvalues in the complex plane can be compared with the amplification factor contours of the

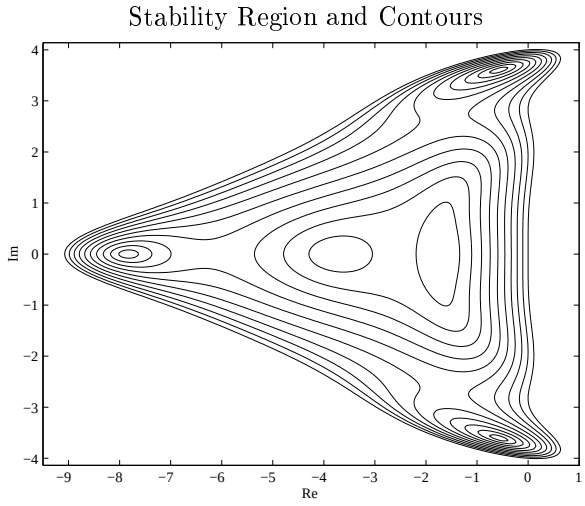


Figure 2.1 Stability region and amplification factor contours defined by $|\psi(z)| = 0.1, 0.2, ..., 1.0$ for a 5-stage Runge–Kutta scheme.

Runge–Kutta scheme to determine the damping properties of the combined spatial and temporal discretization. For a given Runge–Kutta operator, the amplification factor $|\psi(z)|$ is defined by the relationship

$$\widehat{W}^{n+1} = \psi(z)\widehat{W}^{n},$$

where n is the time step index. Stability of the combined discretization requires that the footprint lies within the stability region defined by $|\psi(z)| \leq 1$. The stability region and amplification factor contours for a 5-stage Runge–Kutta scheme due to Martinelli [45] are shown in Fig. 2.1 to provide a realistic context for ensuing examinations of eigenvalue clustering.

In two dimensions, there are four characteristic families representing convective entropy modes, convective vorticity modes and two groups of acoustic pressure modes. From a damping perspective, it is desirable for the residual eigenvalues corresponding to all these modes to be clustered into a region of the complex plane where the amplification factor is significantly less than unity. The primary weakness of explicit time integration using a scalar preconditioner is that a significant fraction of the residual eigenvalues cluster near the origin where the amplification factor is close to unity and the damping of error modes is very inefficient. Since, at the origin, the gradient vector of the amplification factor lies along the negative

real axis, improved damping of these troublesome modes will follow directly from an increase in the magnitude of the real component of the corresponding residual eigenvalues.

Error modes are propagated at the group velocity corresponding to a discrete wave packet of the corresponding spatial frequency. The expression for the group velocity depends on the form of the temporal discretization operator, so it is impossible to determine detailed propagative information from the Fourier footprint. However, for those low frequency modes that are clustered near the origin and are consequently most in need of rapid propagation, the group velocity actually can be obtained by visual inspection of the Fourier footprint, as explained in Appendix A. For Runge–Kutta operators of the type used in the present work, the group velocity of these modes is proportional to the variation in the imaginary components of the residual eigenvalues in the corresponding modal family. Therefore, for rapid propagation, it is desirable for the residual eigenvalues in each family to extend far from the negative real axis.

2.2 Numerical Studies

For full coarsened multigrid to function efficiently, all modes corresponding to the three shaded Fourier quadrants in Fig. 2.2 must be damped by the relaxation scheme since only modes that are low frequency in both mesh directions (L_xL_y) can be resolved without aliasing on the next coarser mesh. Efficient propagation and subsequent expulsion of these error modes will also enhance the performance of the multigrid algorithm. Analysis of eigenvalue clustering will initially focus on modes that are high frequency in both mesh directions (H_xH_y) since point-implicit methods are notoriously poor at eliminating modes with a low frequency component.

2.2.1 High Frequency Modes

Fourier footprints corresponding to high frequency modes for aligned inviscid subsonic flow in a moderately stretched mesh cell are shown for all three stable combi-

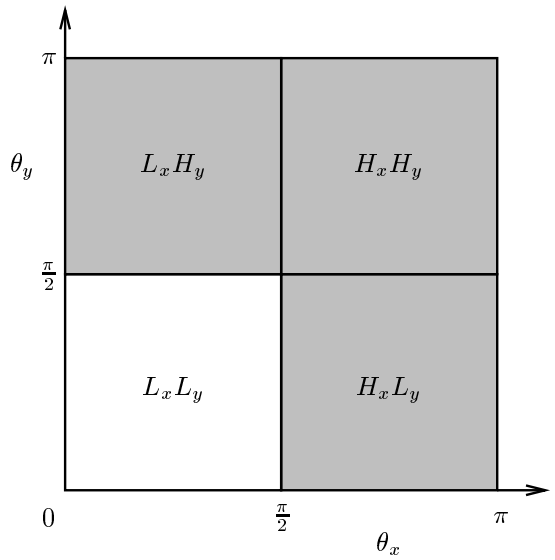
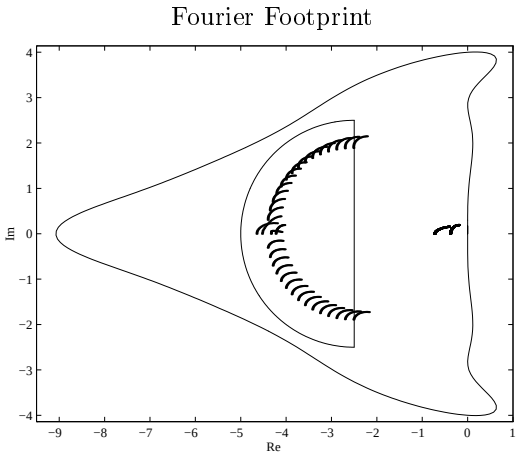


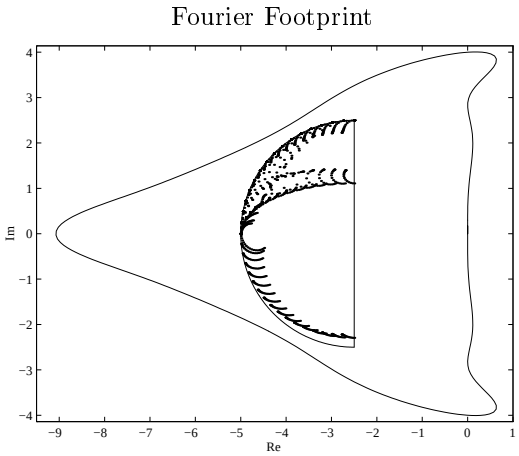
Figure 2.2 Fourier quadrants for which the corresponding error modes must be damped for full coarsened multigrid to function efficiently.

nations of preconditioner and numerical dissipation in Fig. 2.3. The outer solid line in these plots is the stability region of the 5-stage Runge–Kutta scheme. The inner solid line represents the envelope of all possible high frequency footprints arising from a related scalar model problem preconditioned by a suitable scalar time step [1]. Since scalar preconditioning is entirely appropriate for a scalar problem, this boundary represents a useful clustering target for a matrix preconditioner applied to a system of equations. For the purposes of the discussion that follows, this boundary will be considered to define the optimal clustering envelope from a damping perspective. From a propagative viewpoint, only the curved portion of the boundary is optimal.

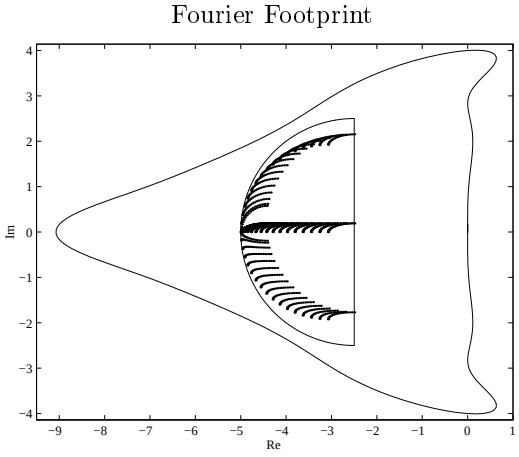
Examining Fig. 2.3a, it is evident that the scalar preconditioner applied to matrix dissipation ($P_S R_M$) is unable to cluster the residual eigenvalues of the two convective modes away from the origin, so that they are neither damped nor propagated efficiently. By contrast, the two acoustic families are clustered nearly inside the optimal envelope and have large imaginary components, so they will be rapidly damped, and low frequency modes from these families will be propagated efficiently. This behavior reflects the better balance that exists between the magnitude of the scalar preconditioner and the dissipative and propagative coefficients of the acoustic



2.3a Scalar preconditioner with matrix dissipation ($P_S R_M$).



2.3b Matrix preconditioner with matrix dissipation ($P_M R_M$).



2.3c Scalar preconditioner with scalar dissipation ($P_S R_S$).

Figure 2.3 Preconditioned Fourier footprint for high frequency modes ($H_x H_y$). $M = 0.5$, $\frac{\Delta y}{\Delta x} = \frac{1}{5}$, $\frac{v}{u} = 0, \sigma = 2.5$.

modes.

The matrix preconditioner applied to matrix dissipation ($P_M R_M$) provides optimal damping clustering for all four modal families. The clustering for the entropy footprint is also optimal from a propagative perspective since it forms an arc on the optimal clustering envelope. The two acoustic footprints have nearly the same radius as the entropy mode, falling one each above and below the real axis, so that low frequency modes from these families will also be propagated efficiently. Only the vorticity footprint, which forms a tongue between the two acoustic footprints, does not approach optimal propagative clustering, though the situation is still far better than with a scalar preconditioner. These excellent damping and propagation properties reflect the delicate balance achieved between the physical characteristic speeds, the magnitude of the corresponding numerical dissipation and the effective time step using the matrix preconditioner.

From a convergence perspective, the scalar preconditioner applied to scalar dissipation ($P_S R_S$) represents an interesting middle ground between the two previous alternatives. The footprints for all four modes are clustered within the optimal damping envelope since both the preconditioner and the numerical dissipation are based on the spectral radii of the flux Jacobians and therefore balance perfectly. However, the propagative properties of this scheme are nearly identical to those of the scalar preconditioner applied to matrix dissipation. The effect of using scalar dissipation has been to slide the eigenvalues along the negative real axis without significantly altering the imaginary components. This behavior, while beneficial in terms of convergence, reflects a corresponding degradation in solution quality since the numerical dissipation no longer scales separately with the individual characteristic speeds. Scalar dissipation has remained popular, despite this drawback, because it does provide superior convergence to matrix dissipative schemes when using a standard scalar preconditioner.

Of the three combinations of preconditioner and numerical dissipation, only the combination of matrix preconditioning and matrix dissipation performs well in terms of damping, propagation and accuracy. Although the $P_S R_S$ combination provides

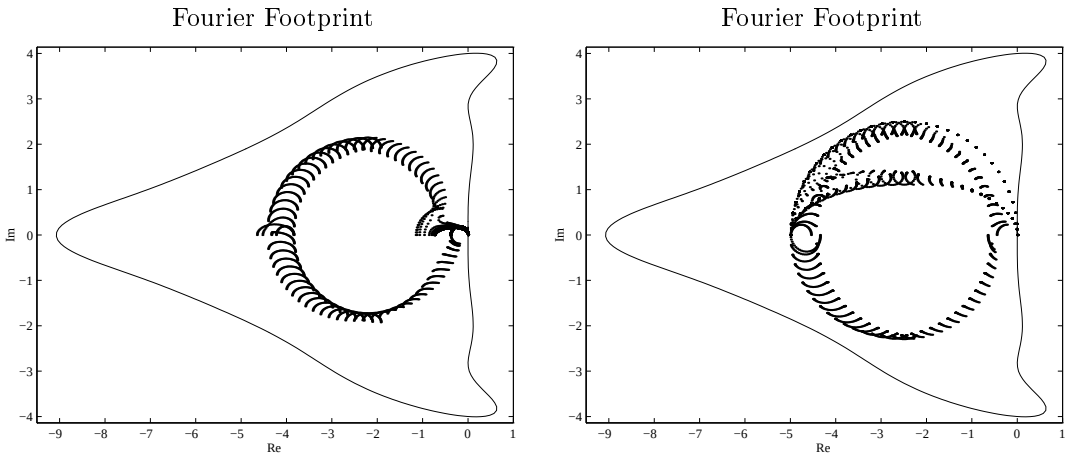
a significant damping improvement over the $P_S R_M$ option, it is highly undesirable to compromise solution accuracy for purposes of convergence, so the remainder of the analysis will focus on scalar and matrix preconditioners applied to matrix dissipation.

2.2.2 High and Low Frequency Modes

We have seen that the matrix preconditioner provides excellent damping and propagative clustering for modes in the $H_x H_y$ quadrant. This result has been shown to hold for a wide range of flow and mesh conditions [1, 59]. The treatment of modes corresponding to the $L_x H_y$ and $H_x L_y$ quadrants must still be accounted for to ensure efficient full coarsened multigrid performance.

For Euler calculations, the cell stretching is typically not severe so that discrete stiffness is chiefly caused by the inherent disparity in propagative speeds and directional decoupling results primarily from flow alignment, as previously indicated in Fig. 1.1. Propagative disparities are most pronounced near the stagnation point, across the sonic line and at shocks, while flow alignment occurs near the airfoil surface when using a body-conforming mesh.

In regions of strong propagative disparity or perfect flow alignment, neither preconditioner succeeds in clustering all of the residual eigenvalues corresponding to the $L_x H_y$ and $H_x L_y$ quadrants away from the origin. To illustrate the effect of directional decoupling caused by flow alignment, consider the footprints of Fig. 2.4 which contain the residual eigenvalues corresponding to all modes except those in the $L_x L_y$ quadrant for the same subsonic flow conditions as were previously examined. Using either preconditioner, the convective entropy and vorticity footprints form arcs which extend to the origin. This reflects the decoupling of these modal families from the normal coordinate direction so that modes from the $L_x H_y$ quadrant cannot be effectively clustered. However, there remains a significant qualitative difference in the magnitude of this shortcoming for the two preconditioning alternatives. Using the scalar preconditioner, the entire footprints of both convective families are densely clustered near the origin so that damping and propagation are impeded for all modes



2.4a Scalar preconditioner.

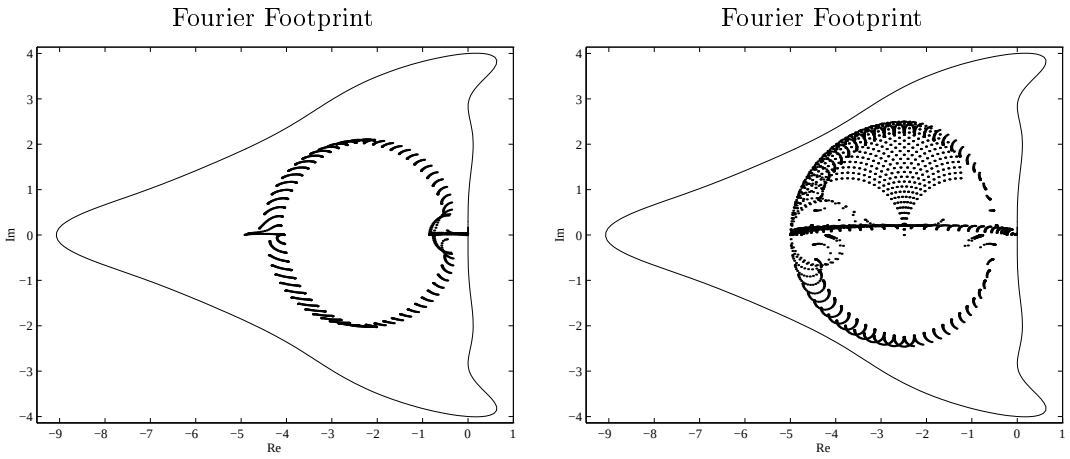
2.4b Matrix preconditioner.

Figure 2.4 Preconditioned Fourier footprints (all modes except $L_x L_y$) illustrating the effects of directional decoupling resulting from flow alignment. $M = 0.5$, $\frac{\Delta y}{\Delta x} = \frac{1}{5}$, $\frac{v}{u} = 0, \sigma = 2.5$.

in these families. With the matrix preconditioner, only the tips of the two convective footprints touch the origin and the rest of the eigenvalues in these families extend far away from both the real and imaginary axes. As a result, nearly all the convective error modes will be damped efficiently and the streamwise group velocity of those modes clustered near the origin will be relatively large.

The ability of the two preconditioners to cope with discrete stiffness arising from the inherent propagative disparity between convective and acoustic modes is illustrated in Fig. 2.5 for low subsonic cross-flow conditions. At a low Mach number of 0.05, the two convective footprints have collapsed nearly to the origin using the scalar preconditioner. Using the matrix preconditioner, the primary effect of the stiffness is the reduction in the imaginary component of the convective vorticity footprint, which now forms a shallow arc above the real axis, extending very close to the origin. On the other hand, the entropy footprint now forms a fan in the upper half-plane and is well clustered away from the origin. This reflects the fact that the diagonal cross-flow has alleviated the directional decoupling which was present in the previous example.

Although the block-Jacobi matrix preconditioner does not entirely eliminate the negative impact of flow alignment and propagative disparity for all modes with a low frequency component, it does significantly improve the prospects for efficient



2.5a Scalar preconditioner.

2.5b Matrix preconditioner.

Figure 2.5 Preconditioned Fourier footprints (all modes except $L_x L_y$) illustrating the effects of discrete stiffness resulting from the inherent propagative disparity between convective and acoustic modes. $M = 0.05$, $\frac{\Delta y}{\Delta x} = \frac{1}{5}$, $\frac{v}{u} = \frac{1}{5}$, $\sigma = 2.5$.

damping and propagation of most error modes. For typical inviscid computations, the impact on convergence of the few troublesome modes that are not well damped is almost certainly insufficient to warrant switching from full coarsened multigrid to a more expensive algorithm.

An interesting possibility for further improving the clustering performance at low Mach numbers has recently been suggested by Turkel [74] and Mavriplis [47], who introduce a low-Mach number preconditioner into the numerical dissipation matrices following Ref. [75]. This modification subsequently alters the form of the block-Jacobi matrix preconditioner to become

$$P_M^{-1} = \frac{1}{\sigma}\left(\frac{P_*^{-1}|P_* A_x|}{\Delta x} + \frac{P_*^{-1}|P_* A_y|}{\Delta y}\right),$$

where P_* is some form of low-Mach number preconditioner [77, 72, 12, 44, 83]. This strategy substantially improves the damping and propagative clustering properties of the block-Jacobi preconditioner as the Mach number approaches zero, but does not yield significant improvements at higher Mach numbers [74]. The approach is therefore best suited for calculations with a low free stream Mach number, but it may still prove worthwhile for transonic calculations due to the subsonic zone at the stagnation point. This idea will not be explored further in the present work, but it represents a potentially fruitful modification of the present scheme that should be

considered in the future.

2.3 Results

This section will demonstrate the relative performance of the scalar and block-Jacobi preconditioners when used in conjunction with full coarsened multigrid for transonic airfoil calculations. Full documentation of the two-dimensional flow solver developed for the present work is provided in Appendices B–D. A brief description of the implementation is provided here for convenience.

The code is based on a conservative cell-centered semi-discrete finite volume approach [36]. Characteristic-based matrix dissipation formed using a Roe linearization [64] provides a basis for the construction of a matrix switched scheme [36, 32] that has been previously validated for inviscid calculations [55, 59]. The solution is computed on a sequence of fine meshes using full coarsened W-cycles in which one time step with a 5-stage Runge–Kutta scheme is performed at each level when moving down the multigrid cycle [36, 27, 45]. The switched scheme is used only on the fine meshes and a purely first order version of the numerical dissipation is used on all coarser meshes. The preconditioner is assembled and inverted before the first stage of each time step for rapid multiplication by the residual updates during each subsequent stage. In the context of preconditioning, an entropy fix serves to prevent the time step from becoming too large near the stagnation point, at shocks and at the sonic line. For Euler calculations, the matrix preconditioner incorporates the same van Leer entropy fix [76] that is used in the numerical dissipation, although numerous tests seem to indicate that this precaution is unnecessary for inviscid calculations.

The test cases used to demonstrate the performance of the proposed preconditioned multigrid method (new) in comparison to the standard approach (standard) are defined in Table 2.1. For the convergence comparisons that follow, the plotted residuals represent the r.m.s. change in density (normalized by the initial residual) during one application of the time-stepping scheme on the finest mesh in the

Test	Geometry	M_∞	α	Mesh	AR_{max}
E1	NACA0012	0.800	1.25°	160×32	2
E2	NACA0012	0.800	1.25°	320×64	2
E3	NACA0012	0.850	1.00°	320×64	2

Table 2.1 Euler test case definitions: airfoil, free stream Mach number, angle of attack, mesh dimensions, maximum cell aspect ratio at the wall.

	Test	Cycles		Rate		CPU Time (s)		Cost Ratio
		Standard	New	Standard	New	Standard	New	
Initial	E1	167	45	.9463	.8120	255.8	79.5	3.22
	E2	213	66	.9576	.8675	1390.0	487.6	2.85
	E3	237	61	.9617	.8532	1550.8	451.3	3.44
Asymptotic	E1	264	48	.9657	.8262	403.2	83.6	4.82
	E2	253	73	.9643	.8804	1646.8	534.8	3.08
	E3	327	71	.9722	.8839	2134.0	520.2	4.10

Table 2.2 Euler results: Initial ($10^0 \rightarrow 10^{-4}$) and asymptotic ($10^{-4} \rightarrow 10^{-8}$) convergence comparisons for scalar preconditioning with full coarsened multigrid (standard) vs. block-Jacobi preconditioning with full coarsened multigrid (new). Categories represent multigrid cycles, convergence rate per cycle, CPU time* and CPU speed-up.

multigrid cycle. Convergence information is also provided in various useful forms in Table 2.2 for both the initial convergence rate between residual levels of 10^0 and 10^{-4} and the asymptotic convergence rate between residual levels of 10^{-4} and 10^{-8}.

The first test case is a standard transonic NACA0012 case with a strong shock on the upper surface and weak shock on the lower surface. Calculations are performed on the 160×32 O-mesh shown in Fig. 2.6, which provides good near-field resolution for inviscid flows but does not introduce significant cell stretching, having a maximum cell aspect ratio of only two. The computed pressure distribution and convergence histories are shown in Fig. 2.7. Both the new and standard methods converge to machine accuracy with very little degradation in asymptotic convergence

*CPU time for an IBM RS6000/590 processor.

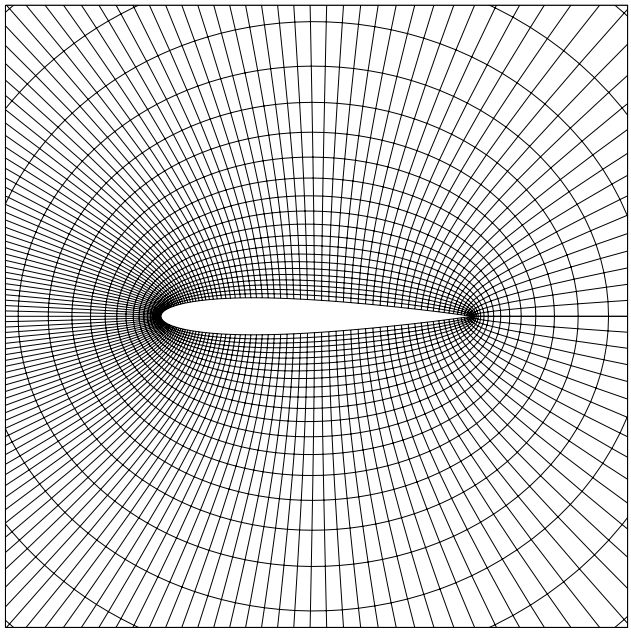


Figure 2.6 160×32 O-mesh for the NACA0012 Airfoil.

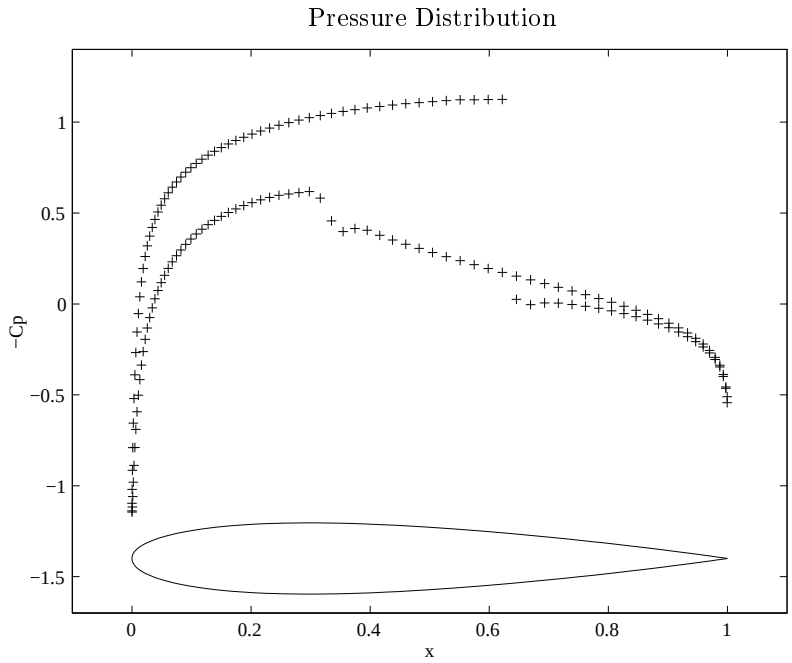
relative to the initial rate, requiring approximately 150 and 700 cycles, respectively. As detailed in Table 2.2, the matrix preconditioned scheme requires 45 cycles to reach a residual level of 10^{-4} at a rate of 0.8120 and an additional 48 cycles to converge the next four orders at a rate of 0.8262. By comparison, the standard scheme using a scalar preconditioner converges four orders in 167 cycles, corresponding to a rate of 0.9463, and then requires an additional 264 cycles to converge the next four orders at a rate of 0.9657. In terms of CPU time, the matrix preconditioner yields computational savings of a factor of 3.22 in initial convergence rate and a factor of 4.82 in asymptotic performance.

Results for the same flow conditions are presented in Fig. 2.8 for a 320×64 O-mesh with twice the resolution of the mesh used for the previous calculation. Using the scalar preconditioner, the number of cycles required to reach machine accuracy increases only slightly to about 720 cycles, while the matrix preconditioner now requires about 220 cycles. The computational savings for this case are a factor of 2.85 in initial convergence and a factor of 3.08 in asymptotic convergence.

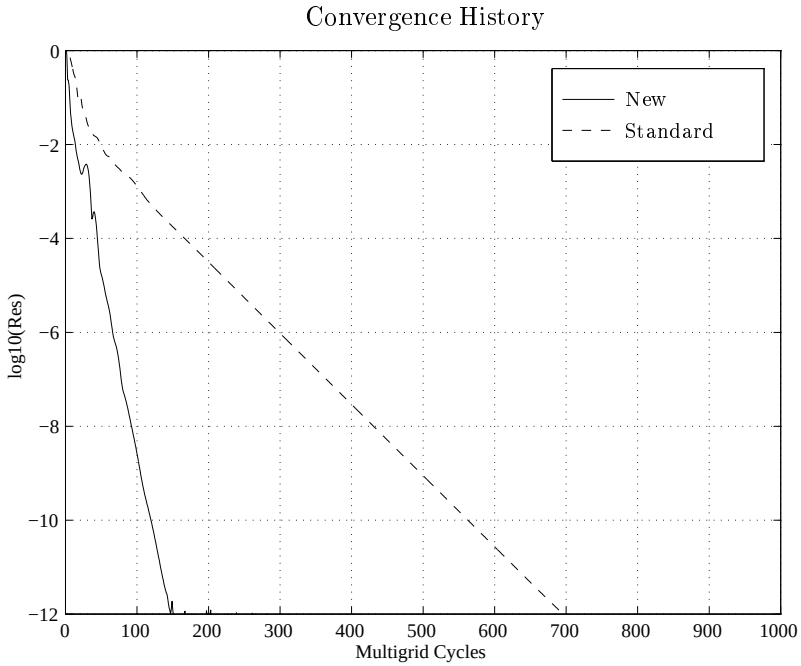
Results for another standard NACA0012 test case with strong shocks on both

upper and lower surfaces are shown in Fig. 2.9 for a calculation performed on the same 320×64 O-mesh. The convergence using the matrix preconditioned scheme is very similar to that of the previous case, while the scalar preconditioned scheme converges somewhat more slowly, so that the initial and asymptotic speed-ups are now 3.44 and 4.10, respectively.

Overall, the scheme using block-Jacobi matrix preconditioning and full coarsened multigrid yields computational savings of roughly a factor of three for convergence to engineering accuracy [59, 60]. Ollivier-Gooch has obtained comparable accelerations using the same matrix preconditioner on unstructured grids [54]. Of principal importance is the fact that this acceleration is achievable without any compromise in the robustness of the flow solver. The groundwork is now in place to consider the more demanding problem of developing an efficient preconditioned multigrid method for turbulent Navier–Stokes calculations.

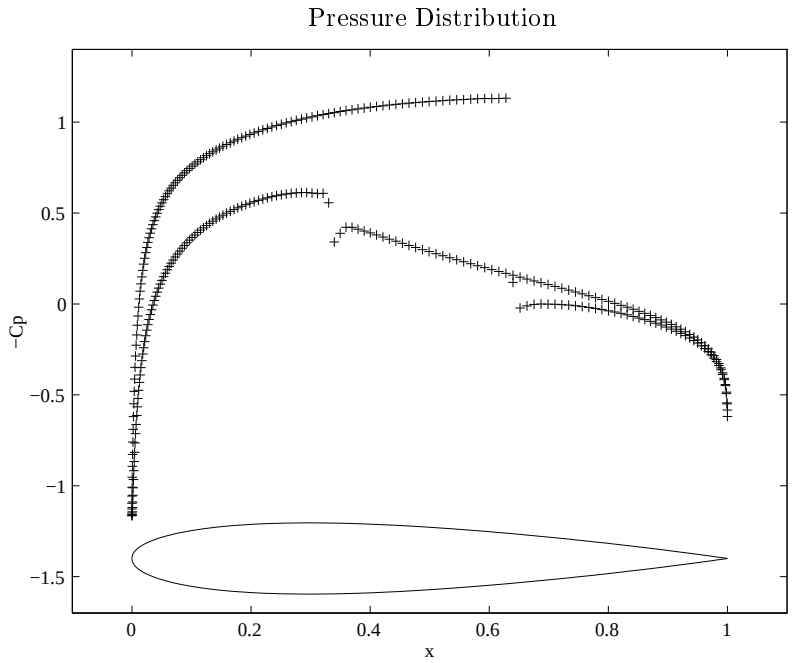


2.7a Coefficient of pressure.
$C_l = .3527, C_d = .0227$.

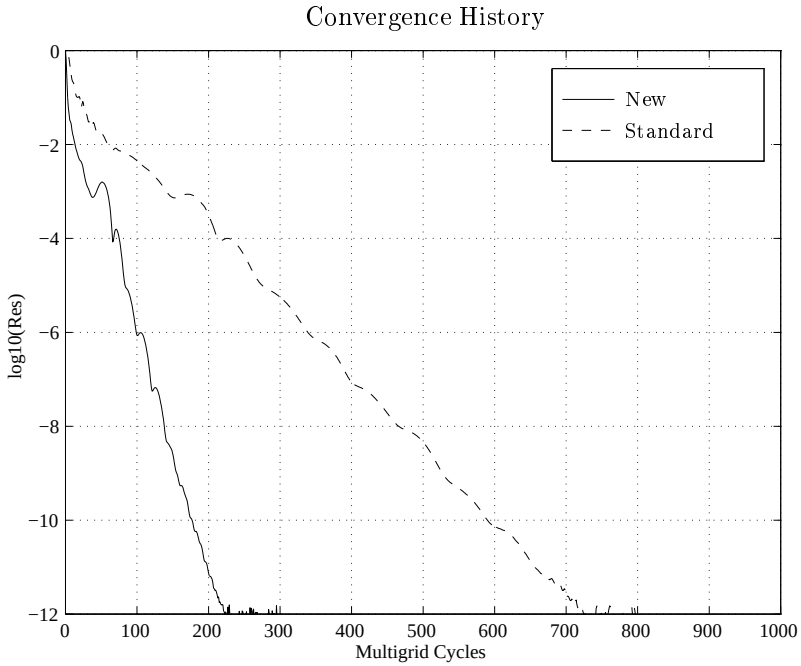


2.7b Convergence comparison.

Figure 2.7 Test E1: Solution and convergence.
NACA0012 Airfoil. $M_\infty = 0.8, \alpha = 1.25$, 160×32 O-mesh.

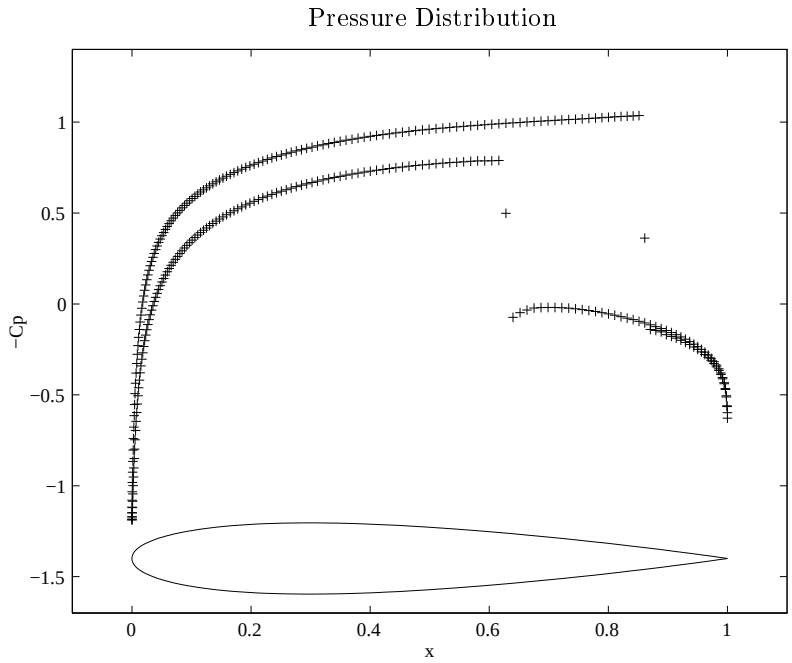


2.8a Coefficient of pressure.
$C_l = .3536, C_d = .0225$.

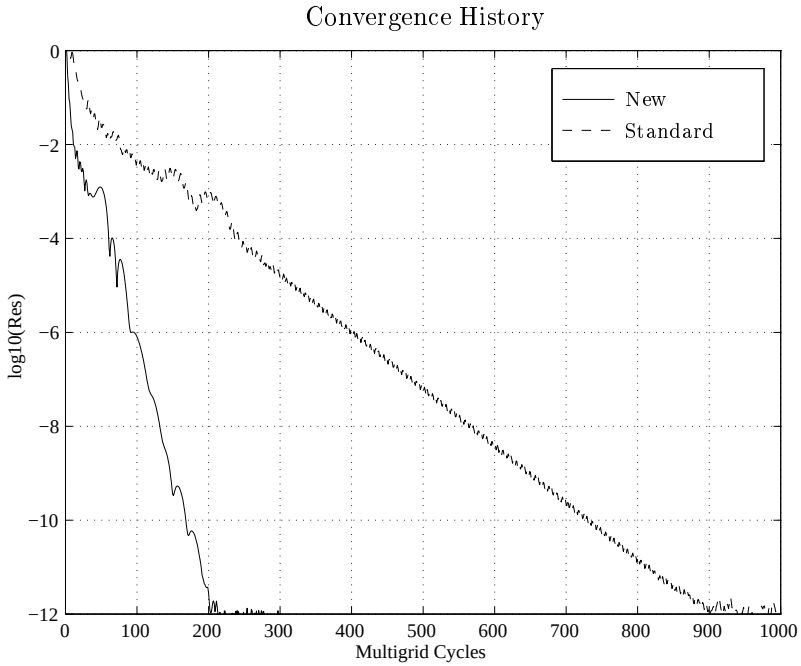


2.8b Convergence comparison.

Figure 2.8 Test E2: Solution and convergence.
NACA0012 Airfoil. $M_\infty = 0.8, \alpha = 1.25$, 320×64 O-mesh.



2.9a Coefficient of pressure.
$C_l = .3721, C_d = .0572$.



2.9b Convergence comparison.

Figure 2.9 Test E3: Solution and convergence.
NACA0012 Airfoil. $M_\infty = 0.85$, $\alpha = 1.0$, 320×64 O-mesh.

Chapter 3

Turbulent Navier–Stokes Applications

For inviscid applications, the problems of discrete stiffness and directional decoupling are not particularly severe. The convergence achieved using block-Jacobi matrix preconditioning and full coarsened multigrid is sufficiently rapid that it is undesirable to adopt a more expensive coarsening strategy to account for the few remaining modes that are not efficiently damped. The situation is much different for turbulent Navier–Stokes calculations, where the highly stretched boundary layer cells significantly exacerbate both the stiffness and decoupling problems. Convergence degrades rapidly as the cell aspect ratios increase, so for viscous applications it is essential to account for the damping of every error mode. Although it is optimal if modes are both rapidly damped and propagated, in the demanding context of severe boundary layer anisotropy, the clustering is deemed successful as long as the residual eigenvalues do not fall arbitrarily close to the origin where the amplification factor is unity.

3.1 Definitions

Before proceeding to the analysis of preconditioned multigrid methods for turbulent Navier–Stokes calculations, each of the basic definitions provided for the Euler

equations must first be restated to incorporate the appropriate viscous terms.

Scheme Description

Viscous analysis is based on the two-dimensional linearized Navier–Stokes equations in Cartesian coordinates

$$\frac{\partial W}{\partial t} + A_x \frac{\partial W}{\partial x} + A_y \frac{\partial W}{\partial y} = B_{xx} \frac{\partial^2 W}{\partial x^2} + B_{yy} \frac{\partial^2 W}{\partial y^2} + B_{xy} \frac{\partial^2 W}{\partial x \partial y},$$

where B_{xx}, B_{yy}, and B_{xy} are the viscous flux Jacobians. The preconditioned semi-discrete scheme has the same basic components as in the inviscid case (2.1), but the residual vector of the spatial discretization now has the form

$$\begin{aligned} R &= \frac{A_x}{2\Delta x}\delta_{2x} - \frac{|A_x|}{2\Delta x}\delta_{xx} + \frac{A_y}{2\Delta y}\delta_{2y} - \frac{|A_y|}{2\Delta y}\delta_{yy} \\ &- \frac{B_{xx}}{\Delta x^2}\delta_{xx} - \frac{B_{yy}}{\Delta y^2}\delta_{yy} - \frac{B_{xy}}{4\Delta x \Delta y}\delta_{2x2y}, \end{aligned} \tag{3.1}$$

where, once again, upwinding of the inviscid terms is accomplished by a Roe linearization [64].

Assuming constant Pr and γ, the four independent parameters that govern the discrete steady Navier–Stokes residual are the cell Reynolds number, Mach number, reciprocal cell aspect ratio and flow angle:

$$Re_{\Delta x} = \frac{u\Delta x}{\nu}, \qquad M = \frac{\sqrt{u^2+v^2}}{c}, \qquad \frac{\Delta y}{\Delta x}, \qquad \frac{v}{u}.$$

As in the inviscid case, a Cartesian mesh is assumed to simplify notation, but the theory extends naturally to real applications using either structured or unstructured meshes.

Scalar Preconditioner

A conservative time step estimate for the Navier–Stokes equations is based on the purely hyperbolic and parabolic time steps formed using the spectral radii of the flux Jacobians [45],

$$P_S^{-1} = \Delta t_h^{-1} + \Delta t_p^{-1}, \tag{3.2}$$

where the hyperbolic time step is given by

$$\Delta t_h^{-1} = \frac{1}{\sigma_h}\left(\frac{\rho(A_x)}{\Delta x} + \frac{\rho(A_y)}{\Delta y}\right)$$

and the parabolic time step is

$$\Delta t_p^{-1} = \frac{1}{\sigma_p}\left(\frac{4\rho(B_{xx})}{\Delta x^2} + \frac{4\rho(B_{yy})}{\Delta y^2} + \frac{\rho(B_{xy})}{\Delta x \Delta y}\right).$$

The factor of four in the parabolic time step arises from considering the scenario of a checkerboard mode, for which the coefficients of the second-difference stencil reinforce each other in both directions. The hyperbolic and parabolic Courant numbers, σ_h and σ_p, reflect the extent of the stability region of the Runge–Kutta time-stepping scheme along the imaginary and negative real axes, respectively. For the 5-stage Runge–Kutta scheme used in the present work, the fact that the maximum extent along the negative real axis is roughly twice the extent in either direction along the imaginary axis (see Fig. 2.1) suggests the definition $\sigma_p = 2\sigma_h$, so that only the hyperbolic Courant number need be specified and the subscripts may be dropped.

Matrix Preconditioner

In the viscous case, the block-Jacobi matrix preconditioner based on the spatial residual operator (3.1) becomes

$$P_M^{-1} = \frac{1}{\sigma}\left(\frac{|A_x|}{\Delta x} + \frac{|A_y|}{\Delta y} + \frac{2B_{xx}}{\Delta x^2} + \frac{2B_{yy}}{\Delta y^2}\right). \tag{3.3}$$

Fourier Footprint

As before, the Fourier footprint is defined by the eigenvalues of the matrix PZ where Z is the Fourier symbol of the spatial residual operator

$$\begin{aligned} Z(\theta_x, \theta_y) &= \hat{\imath}\frac{A_x}{\Delta x}\sin\theta_x + \frac{|A_x|}{\Delta x}(1-\cos\theta_x) \\ &+ \hat{\imath}\frac{A_y}{\Delta y}\sin\theta_y + \frac{|A_y|}{\Delta y}(1-\cos\theta_y) \\ &+ \frac{2B_{xx}}{\Delta x^2}(1-\cos\theta_x) + \frac{2B_{yy}}{\Delta y^2}(1-\cos\theta_y) \\ &+ \frac{B_{xy}}{\Delta x \Delta y}\sin\theta_x \sin\theta_y. \end{aligned}$$

3.2 Asymptotic Analysis

An effective means of illuminating the causes of multigrid breakdown for high Reynolds number calculations is an examination of the preconditioned Fourier footprints that arise in a highly stretched boundary layer cell. As for the preceding inviscid studies, these footprints could be evaluated numerically to reveal the clustering behavior of the residual eigenvalues for various flow and mesh conditions. However, the current interest in high Reynolds number applications suggests that more detailed information may be available by considering the asymptotic form of the preconditioned Fourier footprints as the cell Reynolds number increases to infinity. In this limit, the structure of the footprints becomes sufficiently simple that analytic expressions for their asymptotic behavior can be deduced.

These analytic footprints are valuable because they identify the asymptotic dependence of each modal family on the two Fourier angles, thus exposing the clustering properties of the preconditioned algorithm for modes from each of the Fourier quadrants. In general, asymptotic dependence on a Fourier angle corresponds to effective damping of modes with a high frequency component in the corresponding mesh direction since the residual eigenvalues will be prevented from collapsing to the origin. To establish the relevance of conclusions drawn from these asymptotic results, it will be important to verify that the asymptotic expressions closely approximate the footprints obtained for finite cell Reynolds numbers representative of practical aerodynamic calculations.

3.2.1 Procedure

In the present work, analytic expressions for the preconditioned Fourier footprints are obtained for the important set of asymptotic limits summarized in Table 3.1. Cases E1 and E2 represent the inviscid flows corresponding to the viscous conditions of cases NS1 and NS2, and are provided to illustrate the importance of viscous coupling across the boundary layer in determining the appropriate course of action. Cases E1 and NS1 represent a stretched cell with perfect flow alignment while

Case E1	$Re_{\Delta x} = \infty$	$\frac{\Delta y}{\Delta x} \to 0$	$\frac{v}{u} = 0$
Case E2	$Re_{\Delta x} = \infty$	$\frac{\Delta y}{\Delta x} \to 0$	$\frac{v}{u} = \frac{\Delta y}{\Delta x}$
Case NS1	$Re_{\Delta x} \to \infty$	$\frac{\Delta y}{\Delta x} = Re_{\Delta x}^{-1/2}$	$\frac{v}{u} = 0$
Case NS2	$Re_{\Delta x} \to \infty$	$\frac{\Delta y}{\Delta x} = Re_{\Delta x}^{-1/2}$	$\frac{v}{u} = \frac{\Delta y}{\Delta x}$

Table 3.1 Asymptotic limits for which analytic expressions for the preconditioned Fourier footprints are obtained.

Case	$\text{eig}(P_S Z_M)$	$\text{eig}(P_M Z_M)$
E1	0	$C_x + \hat{\imath} s_x$
	0	$C_x + \hat{\imath} M s_x$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$
E2	0	$\frac{1}{2}(C_x + C_y) + \frac{\hat{\imath}}{2}(s_x + s_y)$
	0	$\frac{1}{1+M} C_x + \frac{M}{1+M}[C_y + \hat{\imath}(s_x + s_y)]$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$
NS1	0	$\frac{2}{2+Pr} C_y + \frac{Pr}{2+Pr}(C_x + \hat{\imath} s_x)$
	0	$\frac{1}{1+2M} C_x + \frac{2M}{1+2M}(C_y + \frac{\hat{\imath}}{2} s_x)$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$
NS2	0	$\frac{1}{1+Pr} C_y + \frac{Pr}{(1+Pr)}[\frac{1}{2}(C_x + C_y) + \frac{\hat{\imath}}{2}(s_x + s_y)]$
	0	$\frac{1}{1+3M} C_x + \frac{3M}{1+3M}[C_y + \frac{\hat{\imath}}{3}(s_x + s_y)]$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$

Table 3.2 Analytic expressions for the preconditioned Fourier footprints of scalar and matrix preconditioners applied to first order upwind matrix dissipation for the cases described in Table 3.1. The modal families are listed in the order: entropy, vorticity, acoustic, acoustic.

Cases E2 and NS2 correspond to the same stretched cell with diagonal cross-flow. For the viscous cases, the cell aspect ratio is scaled to reflect the physical balance between streamwise convection and normal diffusion, so that

$$\frac{u}{\Delta x} = \frac{\nu}{\Delta y^2},$$

which leads to the relation

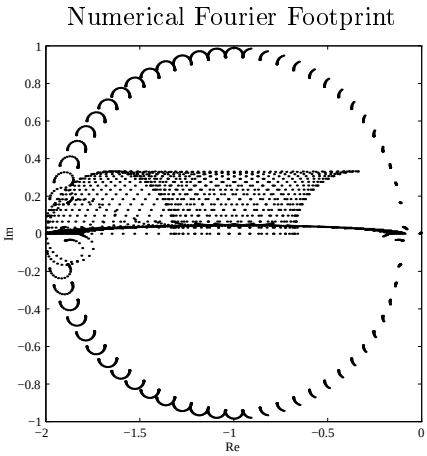
$$\frac{\Delta y}{\Delta x} = Re_{\Delta x}^{-1/2}.$$

The Mach number is held fixed during the limiting procedure so that it appears in the analytic expressions for the Fourier footprints displayed in Table 3.2 for first order upwind matrix dissipation. Here, the notation $s_x \equiv \sin\theta_x$, $s_y \equiv \sin\theta_y$, $C_x \equiv 1 - \cos\theta_x$, $C_y \equiv 1 - \cos\theta_y$ is adopted for brevity.

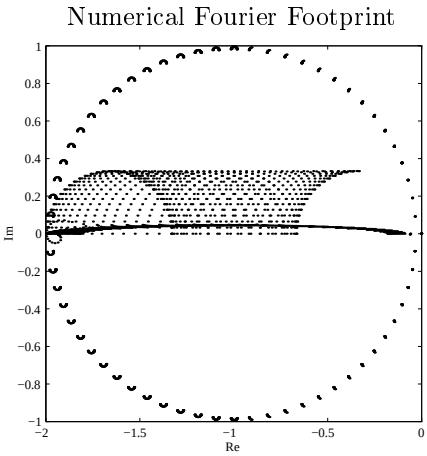
To assist in obtaining these expressions, a symbolic manipulation package [84] was used to expand all the terms in the complete expression for the matrix PZ. For the inviscid cases, the simplified asymptotic form of the matrix was then identified by eliminating all but the dominant terms as $\frac{\Delta y}{\Delta x}$ tended to zero. For the viscous cases, a similar procedure was applied in the limit as $Re_{\Delta x}$ tended to infinity. The analytic expressions for the preconditioned residual eigenvalues of the asymptotic matrix were then obtained using the same symbolic package.

3.2.2 Validation

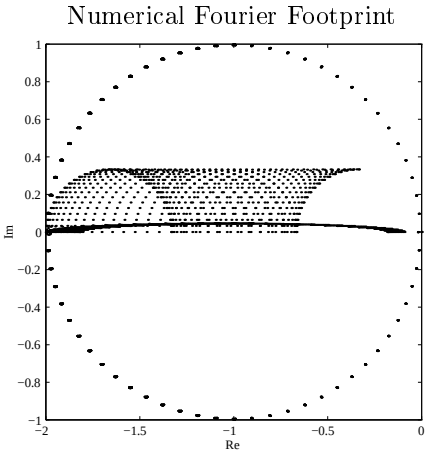
Before considering the clustering implications of the expressions in Table 3.2, it is first important to verify both that the analytic results correctly describe the asymptotic behavior of the complete expressions for the footprints and also that the asymptotic approximation is appropriate for Reynolds numbers typical of aerodynamic applications. For this purpose, numerically generated Fourier footprints corresponding to matrix preconditioning for the aligned viscous flow of Case NS1 are shown for cell Reynolds numbers of 10^2, 10^3 and 10^4 in Fig. 3.1. Comparison of these footprints with a plot of the corresponding analytic asymptotic footprint displayed in Fig. 3.3a reveals that the analytic expressions do accurately reflect the



3.1a $Re_{\Delta x} = 10^2$.



3.1b $Re_{\Delta x} = 10^3$.



3.1c $Re_{\Delta x} = 10^4$.

Figure 3.1 Numerical validation of analytic asymptotic Fourier footprints. Case NS1 for a range of cell Reynolds numbers with $M = 0.05$.

true asymptotic behavior of all four modal families as the cell Reynolds number increases.

To establish the applicability of the asymptotic approximation to analysis motivated by practical aerodynamic applications, it is necessary only to demonstrate that the cell Reynolds numbers typically achieved near the wall in turbulent calculations are sufficiently large. The cell Reynolds number and the free stream Reynolds number based on chord are related by the expression

$$Re_{\Delta x} = Re_L \, \frac{u}{u_\infty} \, \frac{\Delta x}{L}.$$

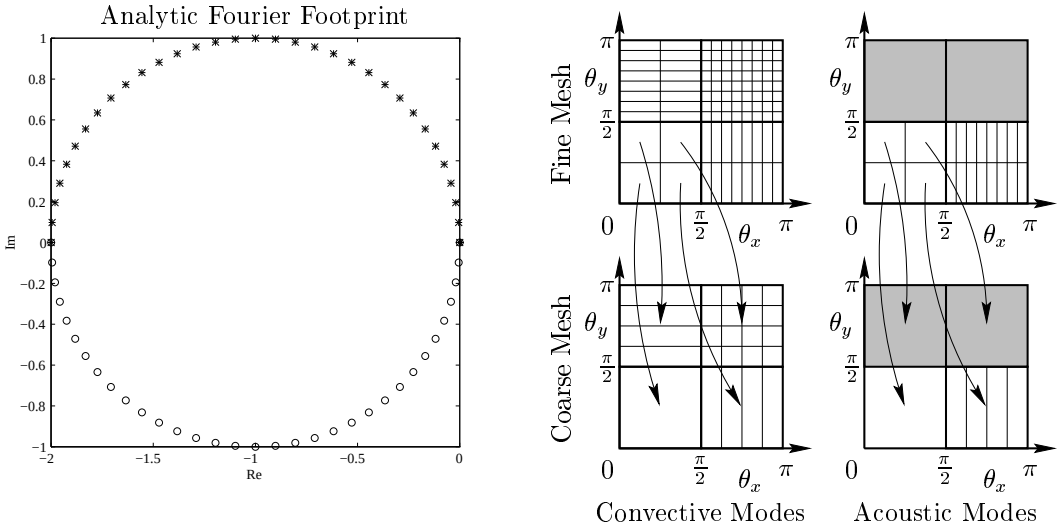
For flow and mesh conditions typical of transonic turbulent aerodynamic calculations, a good estimate of the minimum cell Reynolds number achieved in the boundary layer is given by

$$Re_{\Delta x} \approx O(10^7) \; O(10^{-2}) \; O(10^{-2}) = O(10^3),$$

which falls well within the range for which asymptotic footprints have been demonstrated to closely approximate the full Fourier footprint expressions.

3.2.3 Scalar Preconditioner and Full Coarsened Multigrid

The analytic footprints of Table 3.2 may now be used to identify the inadequacies of the standard combination of scalar preconditioning and full coarsened multigrid for high Reynolds number applications. Using the scalar preconditioner, the asymptotic footprints are identical for all four cases and are displayed in Fig. 3.2a for all modes except those in the L_xL_y quadrant, which need not be damped on the fine mesh in a full coarsened multigrid context. The *entire* footprints of both convective families collapse to the origin so that neither damping nor propagation of these modes is possible and the system will not converge. The situation is not as bad for the acoustic families, which depend asymptotically on θ_y so that modes with a high frequency component in the y direction will be damped effectively. However, acoustic modes that are low frequency in the y direction will be poorly damped, and in the worst case, the eigenvalue for a "washboard" acoustic mode that is uniform in the y



3.2a Fourier footprint for all modes except L_xL_y.

3.2b Damping schematic for full coarsened multigrid.

Figure 3.2 Clustering provided by the scalar preconditioner and implications for the performance of full coarsened multigrid in highly stretched boundary layer cells. Footprint symbols: entropy (+), vorticity (·), acoustic (∗, ∘).

direction and high frequency in the x direction will fall exactly on the origin. These H_xL_y modes will still propagate transversely across the boundary layer since the dependence on θ_y produces a large variation in the imaginary components of the acoustic footprints. Although beneficial, this error propagation is far from sufficient to yield efficient elimination of these modes in a multigrid context, and it remains essential to ensure that all error modes can be eliminated locally by the damping of the Runge–Kutta scheme.

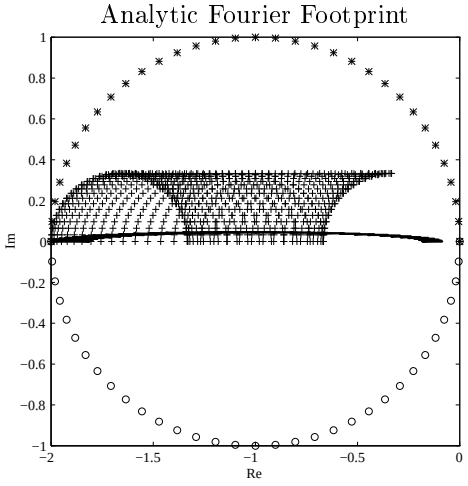
Given this scenario, the expectations for the performance of full coarsened multigrid with scalar preconditioning, which is the strategy in widespread use throughout the CFD community, are illustrated schematically in Fig. 3.2b. The shaded regions represent Fourier quadrants for which the corresponding modes are effectively damped and the other hatchings are stylized depictions of the modes that cannot be damped and therefore prevent or impede convergence. There is no mechanism for damping convective modes in any quadrant or acoustic modes in the H_xL_y quadrant. It is not surprising that poor convergence is observed when using this algorithm for viscous computations with highly stretched boundary layer cells.

3.2.4 Matrix Preconditioner and Full Coarsened Multigrid

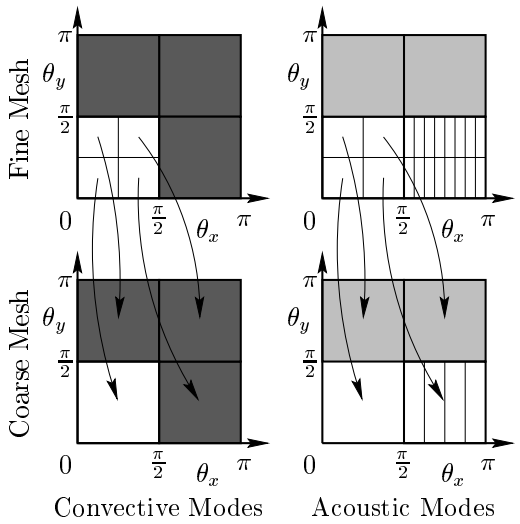
Developing an understanding for the behavior of the block-Jacobi matrix preconditioner requires a careful examination of the expressions in Table 3.2. For the aligned inviscid flow of Case E1, the convective modes are dependent only on θ_x, and the acoustic modes are dependent only on θ_y, so that each modal family is damped effectively in only two Fourier quadrants. By comparison, the viscous results of Case NS1 reveal that the balance between streamwise convection and normal diffusion has caused the two convective families to become dependent on both Fourier angles, so that all quadrants except L_xL_y will be damped effectively. For the entropy family, this property is independent of the Mach number, while for the vorticity family, this behavior exists except in the case of vanishing Mach number. For both inviscid and viscous results, the effect of introducing diagonal cross-flow in Cases E2 and NS2 is to improve the propagative performance for the convective modes by introducing a dependence on both Fourier angles in the imaginary components. Notice that the matrix preconditioner has no effect on the footprints for the acoustic modes, which are identical to those using the scalar preconditioner.

The scenario for full coarsened multigrid using the matrix preconditioner is illustrated by the Fourier footprint and schematic damping diagram of Fig. 3.3. The footprint depicts all modes except L_xL_y for the perfectly aligned viscous flow of Case NS1 with $M = 0.05$. This Mach number represents a realistic value for a highly stretched boundary layer cell near the wall [58]. Fig. 3.3a reveals that the entropy footprint is clustered well away from the origin for all modes. The vorticity footprint remains distinctly clustered away from the origin even at this low Mach number. Propagative clustering of the vorticity mode away from the real axis improves if either the Mach number or the flow angle increases.

This beneficial effect on the clustering of the convective eigenvalues has a profound influence on the outlook for the performance of full coarsened multigrid as described in Fig. 3.3b. Darker shading is used to denote the Fourier quadrants for which damping is facilitated by use of a matrix preconditioner. The full coarsened algorithm will now function efficiently for all convective modes. However, the foot-



3.3a Footprint for all modes except L_xL_y. Case NS1 with $M = 0.05$.



3.3b Damping schematic for full coarsened multigrid.

Figure 3.3 Clustering provided by the block-Jacobi matrix preconditioner and implications for the performance of full coarsened multigrid in highly stretched boundary layer cells. Footprint symbols: entropy (+), vorticity ($\cdot$), acoustic ($*, \circ$).

prints for the acoustic modes still approach the origin when θ_y is small, so the only remaining impediments to efficient performance are the acoustic modes corresponding to the H_xL_y quadrant.

3.2.5 Matrix Preconditioner and J-Coarsened Multigrid

The fact that the block-Jacobi preconditioner provides effective clustering of convective eigenvalues in all but the L_xL_y quadrant provides the freedom to modify the multigrid coarsening strategy with only the damping of H_xL_y acoustic modes in mind. One possibility that avoids the high cost of a complete semi-coarsening stencil and takes advantage of the damping properties revealed in the present analysis is a J-coarsened strategy in which coarsening is performed only in the direction normal to the wall. Using this approach, only modes corresponding to the shaded Fourier quadrants in Fig. 3.4 need be damped by the relaxation scheme on the current mesh since all modes that are low frequency in θ_y can now be resolved on the next coarser mesh. The implications for multigrid performance with this strategy are summarized in Fig. 3.5. The Fourier footprint is plotted for the diagonal cross-flow

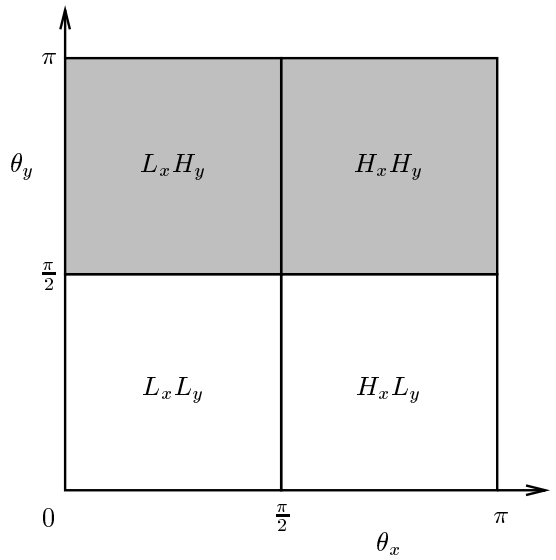
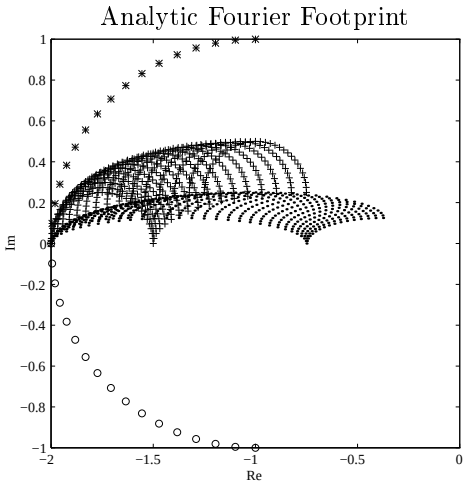


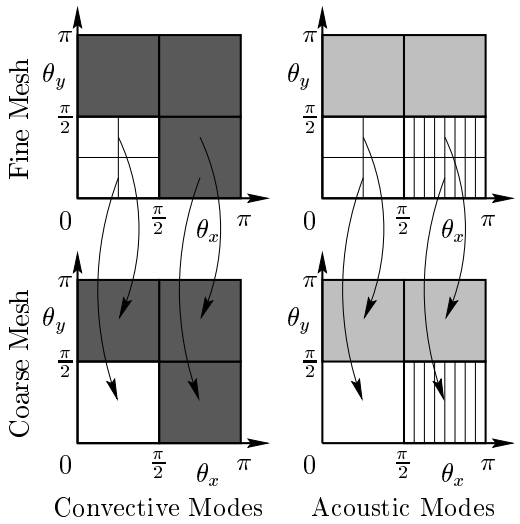
Figure 3.4 Fourier quadrants for which the corresponding error modes must be damped for J-coarsened multigrid to function efficiently.

of Case NS2 with $M = 0.2$ to demonstrate the rapid improvement in the clustering of the convective eigenvalues as the flow angle and Mach number increase above the extreme conditions shown in Fig. 3.3a. Only residual eigenvalues corresponding to modes in the L_xH_y and H_xH_y Fourier quadrants are displayed in Fig. 3.5a since modes from the other two quadrants can be resolved on the next coarser mesh. The residual eigenvalues are now effectively clustered away from the origin for all families.

The schematic of Fig. 3.5b demonstrates that the combination of block-Jacobi preconditioning and J-coarsened multigrid accounts for the damping of *all* error modes inside highly stretched boundary layer cells. This result holds even for the perfectly aligned flow of Case NS1 as long as the Mach number does not vanish. The requirement on Mach number emphasizes the point that the methods developed in this paper are not intended for preconditioning in the limit of incompressibility. For typical viscous meshes, the Mach number remains sufficiently large, even in the cells near the wall, that the tip of the vorticity footprint remains distinguishable from the origin as in Fig. 3.3a. For most boundary layer cells, the Mach number is large enough that even the vorticity footprint is clustered well away from the origin as in Fig. 3.5a. The interaction between the preconditioner and multigrid algorithm



3.5a Footprint for L_xH_y and H_xH_y modes. Case NS2 with $M = 0.2$.



3.5b Damping schematic for J-coarsened multigrid.

Figure 3.5 Clustering provided by the block-Jacobi matrix preconditioner and implications for the performance of J-coarsened multigrid in highly stretched boundary layer cells. Footprint symbols: entropy (+), vorticity ($\cdot$), acoustic ($*, \circ$).

is critical, since the preconditioner is chiefly responsible for damping the convective modes and the coarsening strategy is essential to damping the acoustic modes.

The use of a low-Mach number preconditioner within the block-Jacobi construction [74, 47], previously described in an inviscid context on p. 25, could potentially prove most beneficial for viscous applications, where large regions of low-subsonic flow exist near the wall even for transonic calculations. On the basis of the analysis presented above, it appears that this modification to the present approach would be particularly attractive if improved damping and propagative clustering of the convective vorticity family could be demonstrated for the case of large cell aspect ratios. This possibility is not explored further in the present work but should be considered as a subject for future investigation.

Cost bounds for full and J-coarsened cycles are presented in Table 3.3, where N is the cost of relaxation on the fine mesh and K is the number of multigrid levels. The cost of J-coarsened multigrid is independent of the number of dimensions since coarsening is performed in only one direction. For a V-cycle, the cost of J-coarsening is 80% more than full coarsening in two dimensions and 133% more in

2D	Full	J
V	$\frac{5}{3}N$	$3N$
W	$2N$	KN

3.3a 2D multigrid cost bounds.

3D	Full	J
V	$\frac{9}{7}N$	$3N$
W	$\frac{4}{3}N$	KN

3.3b 3D multigrid cost bounds.

Table 3.3 Cost comparisons for V and W-cycles using full and J-coarsened multigrid.

three dimensions. Use of a J-coarsened W-cycle is inadvisable since the cost depends on the number of multigrid levels. While there is a significant overhead associated with using J-coarsened vs. full coarsened multigrid, subsequent demonstrations will show that the penalty is well worthwhile for turbulent Navier–Stokes calculations.

The implementation of J-coarsening for structured grid applications is straightforward for single block codes but potentially problematic for multi-block solvers. Coarsening directions will not necessarily coincide in all blocks so that cell mismatches would be produced at the block interfaces on the coarse meshes. This problem can be avoided if an implementation is developed in which hanging nodes are permitted so that local coarsening directions can be specified independently for each block, or even locally within each block [25]. The difficulty of mismatches on the coarse meshes can also be circumvented by the overset grid approach in which interpolation is performed between the overlapping blocks [8].

Since the J-coarsened approach is only beneficial inside the boundary layer, those blocks in the inviscid region of the flow should employ a full coarsened strategy, while continuing to use the block-Jacobi preconditioner for improved eigenvalue clustering [59, 60]. Assuming that half the mesh cells are located in blocks outside the boundary layer, this has the effect of decreasing the cost of the multigrid cycle to the average of the full and J-coarsened bounds.

Although the J-coarsened approach is described in the present work using structured mesh terminology, the method also fits very naturally into unstructured grid applications. In this case, it is no longer necessary to specify a global coarsening direction since edge collapsing [16, 17], agglomeration [66, 48] or graph-based overset meshing [47] procedures can be employed to provide directional coarsening across

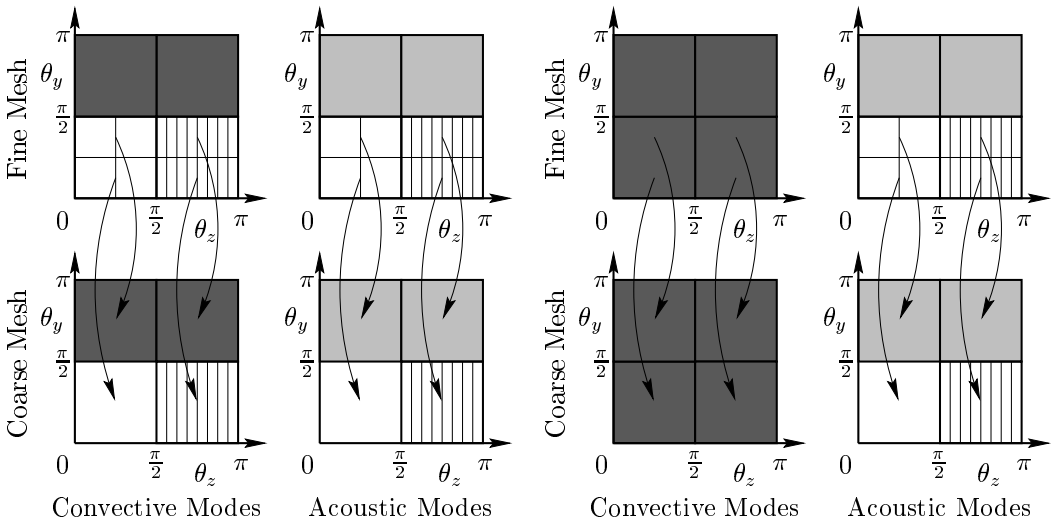
the boundary layer and full coarsening in the inviscid regions.

3.2.6 Extension to Three Dimensions

In three dimensions, the situation is complicated by the addition of one more convective vorticity family and one more space dimension in which error modes for all five modal families must be efficiently damped. It is not possible to obtain analytic asymptotic results for the three-dimensional preconditioned Fourier footprints, since determination of the eigenvalues requires solution of a quintic equation*. Sidestepping this intractability, it is still possible to proceed in assessing the performance of the two proposed methods by assuming the worst case scenario: that none of the preconditioned footprints are asymptotically dependent on θ_z. This is true for the cases previously examined if $\Delta z = O(\Delta x)$ and the flow is aligned with the x direction.

In three dimensions, a complete multigrid damping schematic must display information for all eight Fourier octants. This is most easily accomplished by displaying two slices through Fourier space, one for low values of the third Fourier angle and one for high values. One possibility is to take the slices for low and high values of θ_z. Assuming that there is no coupling in the z direction, these two slices would each be identical to the two-dimensional schematic of Fig. 3.5. In other words, the performance of the scheme is unchanged by the addition of a third coordinate direction. To see why this is the case, it is illuminating to display slices for low and high values of θ_x so that the interaction between the z direction and the direction of coarsening is exposed. Fig. 3.6 illustrates that the behavior for all acoustic modes is identical to the two-dimensional case. The principal development in three dimensions is that the preconditioner no longer ensures that all convective modes are damped since the convective eigenvalues for the $L_xL_yH_z$ octant are assumed not to be dependent on θ_z. However, using a J-coarsened strategy, these modes are transferred to a coarser mesh where they become $L_xH_yH_z$ modes that can then be effectively damped. The

*Abel proved in 1824 that the method of radicals, which can be employed to find the roots of equations of degree ≤ 4, cannot be extended to the solution of quintics [69].



3.6a Damping Schematic for Low θ_x. $(0 \leq \theta_x \leq \frac{\pi}{2})$

3.6b Damping Schematic for High θ_x. $(\frac{\pi}{2} \leq \theta_x \leq \pi)$

Figure 3.6 Implications for the performance of J-coarsened multigrid with block-Jacobi matrix preconditioning in three dimensions.

coarsening strategy therefore assumes the extra responsibility of eliminating some of the convective modes in addition to the normal role of eliminating all the acoustic modes with a low frequency component in θ_y.

Although the use of J-coarsened multigrid avoids the excessively high cost of using a complete semi-coarsening algorithm, the cost of a J-coarsened V-cycle is still more than double the cost of full coarsening for three-dimensional applications. Therefore, it may prove worthwhile to consider a rival combination of preconditioner and coarsening strategy that strikes a moderately different balance between cost, storage and scalability demands. Asymptotic analysis indicates that the combination of a J-Jacobi preconditioner that is line-implicit in the direction normal to the wall, together with an IJ-coarsened multigrid strategy that coarsens in both the normal direction and predominate streamwise direction, will also provide effective damping of all error modes inside the boundary layer [61]. Subsequent investigations may even demonstrate that this assessment is overly conservative and that full coarsened multigrid may be employed with impunity. In either case, this scheme has the advantage that the beneficial multigrid cost bounds are substantially recovered. The use of an implicit preconditioner in only one mesh direction need not inhibit the

parallel efficiency of the method since the approach is only appropriate inside the boundary layer and can therefore be applied in only those blocks adjacent to the wall. These improvements are obtained at the cost of increased storage requirements and an increased cost of relaxation. This method has not yet been thoroughly explored but details of the analysis motivating the approach can be found in Ref. [61].

Using a hybrid approach, in which line-implicit preconditioning is combined with directional coarsening normal to the wall, Mavriplis [47] has recently demonstrated good convergence rates for turbulent Navier–Stokes calculations on unstructured grids. Rather than opting for a full coarsened strategy, the improved low frequency damping properties of the line-implicit preconditioner are exploited by increasing the ratio of the directional coarsening procedure from 2:1 to 4:1 so that desirable multigrid cost bounds can once again be attained for both V and W-cycles.

3.3 Results

This section will demonstrate the acceleration provided by the proposed preconditioned multigrid method of block-Jacobi preconditioning and J-coarsened multigrid (new) in comparison to the standard approach of scalar preconditioning and full coarsened multigrid (standard) for two and three-dimensional turbulent Navier–Stokes calculations on highly stretched meshes.

Two-dimensional calculations were performed using a flow solver developed for the present work and three-dimensional calculations were performed with the solver FLO107 written by Jameson and Martinelli [34]. This code was modified by the author to include matrix preconditioning and J-coarsening as well as to operate on parallel platforms. Full documentation of the two-dimensional implementation is provided in Appendices B–E and details are provided whenever the extension to three dimensions is not straightforward. The discretization of the inviscid portion of the algorithm is identical to that used for the Euler solver, as described briefly in Section 2.3. In discretizing the viscous terms, it is critical to use a compact formulation that does not admit odd/even oscillatory modes, as these modes

can significantly impede convergence. In two dimensions, this is accomplished by evaluating the gradients for the stress components using auxiliary control volumes centered on the cell faces [45]. In three dimensions, each cell is associated with three faces but only one vertex, so it is more economical to evaluate the gradients and stress components at the vertices and then average to the cell faces before computing the viscous flux balance. This implementation produces a large stencil which admits odd/even spurious modes, but the discretization can be approximately converted to the compact stencil of the face-centered approach by applying a correction stencil to the velocity gradients computed at the vertices [33, 43, 61].

For turbulent Navier–Stokes calculations, solutions are computed on a sequence of fine meshes using either full or J-coarsened V-cycles on each mesh. A single Runge–Kutta time step is computed at each level when moving both up and down the multigrid cycle. Again the switched formulation of the numerical dissipation is used on the fine meshes and a first order version is used on all coarser meshes. In viscous calculations, the principal role of the entropy fix in the matrix preconditioner is to bound the hyperbolic contribution to the time step near the wall. For high aspect ratio cells, the van Leer entropy fix used in the numerical dissipation does not sufficiently limit the time step to provide robustness [76] so a more severe Harten entropy fix [24] is employed.

Both the algebraic Baldwin–Lomax (BL) turbulence model [6] and the one-equation Spalart-Allmaras (SA) turbulence model [68] are implemented. The turbulent transport equation for the SA model 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 must be taken to ensure that neither the time integration procedure nor the coarse grid corrections introduce negative turbulent viscosity values into the flow field. Even so, it is sometimes necessary to freeze the turbulent viscosity after a certain initial level of convergence has been achieved, to prevent the turbulence models from adversely affecting the convergence of the flow equations. For the convergence comparisons that follow, the plotted residuals represent the r.m.s. change in density

or turbulent viscosity (normalized in each case by the corresponding initial residual) during one application of the time-stepping scheme on the finest mesh in the multigrid cycle.

3.3.1 Two Dimensions

The two-dimensional turbulent Navier–Stokes test cases used for the present work are defined in Table 3.4 and correspond to RAE2822 AGARD Cases 6 and 9 [14]. Initial ($10^0 \rightarrow 10^{-4}$) and asymptotic ($10^{-4} \rightarrow 10^{-8}$) convergence information for these calculations is provided in Table 3.5. All but one of these calculations converged smoothly to machine accuracy without having to freeze the turbulent viscosity. The only exception was for AGARD Case 9 using the SA turbulence model and the standard preconditioned multigrid method, when the turbulence field was frozen after the density had converged by four orders of magnitude. These calculations were performed on a 288×64 C-mesh with 224 cells on the surface of the airfoil as shown in Fig. 3.7. The maximum cell aspect ratio on the airfoil surface is 2500 and the average and maximum y^+ values at the first cell height are about one and two, respectively.

Results for RAE2822 AGARD Case 6 using both the Spalart–Allmaras [68] and Baldwin–Lomax [6] turbulence models are shown in Figs 3.8 and 3.9. The computed pressure distributions compare well with the experimental results [14] as shown in Fig. 3.8a. The Spalart–Allmaras turbulence model produces a shock location somewhat forward of the experimental location as has been previously observed [68].

Convergence histories of the density and turbulent viscosity residuals using the SA model are shown in Fig. 3.8b for both the new approach of block-Jacobi preconditioning with J-coarsened multigrid and the standard approach of scalar preconditioning with full coarsened multigrid. Using the new approach, both quantities converge to machine accuracy in under 500 cycles, while the standard approach converges rapidly at first and then experiences the widely observed degradation in convergence after about three orders, eventually reaching machine accuracy after

Test	Geometry	M_∞	α	Re_L	Mesh	$\mathrm{AR}_{\max}$	$y^+_{\mathrm{ave/max}}$
NS1	RAE2822	0.725	2.40°	6.5×10^6	288×64	2500	1.02/2.12
NS2	RAE2822	0.730	2.79°	6.5×10^6	288×64	2500	0.97/1.83

Table 3.4 Two-dimensional turbulent Navier–Stokes test case definitions: airfoil, free stream Mach number, angle of attack, Reynolds number, mesh dimensions, maximum cell aspect ratio at the wall, average and maximum y^+ at the first cell height.

about 35,000 cycles. From Table 3.5 it is evident that the new approach converges four orders of magnitude in 113 cycles at a rate of .9205 while the standard approach requires 2212 cycles at a rate .9958, yielding computational savings of 10.49 in initial convergence. The standard scheme then requires an additional 13,109 cycles to converge the next four orders while the new approach requires only 163, corresponding to a computational speed-up of 42.93 in asymptotic performance.

To demonstrate the individual roles that the preconditioners and coarsening strategies play in determining convergence properties, residual histories generated using the Baldwin–Lomax turbulence model for the same AGARD Case 6 test case are shown for all four combinations of preconditioner and coarsening strategy in Fig. 3.9a. These schemes are designated $\mathrm{P_M MG_J}$, $\mathrm{P_S MG_J}$, $\mathrm{P_M MG_{Full}}$ and $\mathrm{P_S MG_{Full}}$, where the first and last combinations correspond to the schemes otherwise referred to as "new" and "standard". First, it is worth mentioning that over-plotting the results for the new and standard schemes with the previously described results obtained using the SA turbulence model reveals that the convergence histories are virtually identical all the way to machine accuracy. This demonstrates that the solution of the one-equation SA turbulence model can be obtained using multigrid without any negative effects on the convergence of the flow equations [60, 58].

Returning to the discussion of the four combinations of preconditioners and coarsening strategies, it is evident from Fig. 3.9a that in comparison to the scalar preconditioner, the block-Jacobi matrix preconditioner has the effect of improving both the initial and asymptotic convergence rates using either coarsening strategy, but does not influence the shape of the convergence history. In particular, the re-

	Test	Turb Model	Cycles		Rate		CPU Time (s)		Cost Ratio
			Standard	New	Standard	New	Standard	New	
Initial	NS1	SA	2212	113	.9958	.9205	17,747.1	1692.4	10.49
		BL	2262	114	.9959	.9208	13,310.3	1234.0	10.79
	NS2	SA	2273	110	.9960	.9196	18,150.8	1640.9	11.06
		BL	2467	111	.9963	.9175	14,576.6	1202.3	12.12
Asymptotic	NS1	SA	13,109	163	.9993	.9456	104,086.0	2424.3	42.93
		BL	12,508	162	.9993	.9455	73,606.6	1747.2	42.13
	NS2	SA	13,827	174	.9993	.9485	110,164.3	2576.1	42.76
		BL	17,190	161	.9995	.9463	101,553.9	1737.4	58.45

Table 3.5 Two-dimensional turbulent Navier–Stokes results: Initial ($10^0 \rightarrow 10^{-4}$) and asymptotic ($10^{-4} \rightarrow 10^{-8}$) convergence comparisons for scalar preconditioning with full coarsened multigrid (standard) vs. block-Jacobi preconditioning with J-coarsened multigrid (new). Categories represent multigrid cycles, convergence rate per cycle, CPU time* and CPU speed-up.

sults using the matrix preconditioner and full coarsened multigrid ($P_M MG_{Full}$) still exhibit a significant degradation in convergence at around three orders of magnitude. On the other hand, the dominant effect of J-coarsening in comparison to the standard full coarsened strategy 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. These results suggest that for turbulent Navier–Stokes calculations on highly stretched meshes, the initial convergence is limited by the convective modes, while the asymptotic convergence is limited by the acoustic modes. Re-examining Fig. 3.9a, it is evident that J-coarsening actually has no effect on the initial convergence using the scalar preconditioner since the convective error modes are still dominant. On the other hand, when employing the matrix preconditioner, the convective modes are being effectively damped so the acoustic modes become significant even in the initial stages of convergence and J-coarsening yields improvements throughout the convergence process.

*CPU time for an IBM RS6000/590 processor.

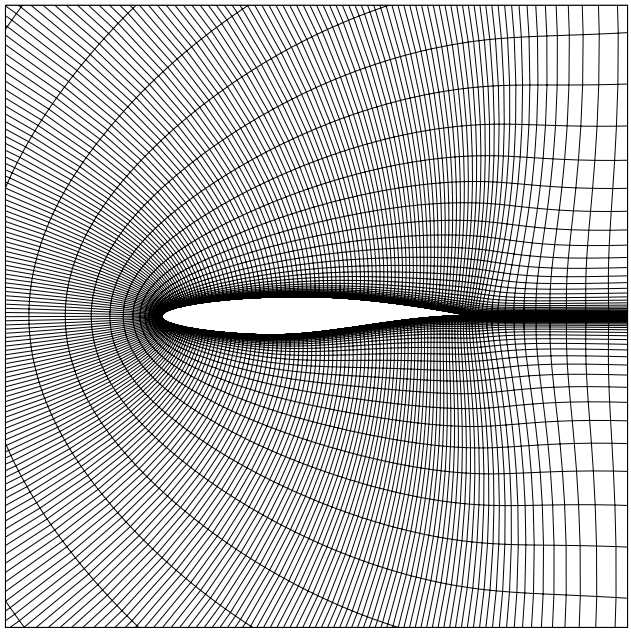


Figure 3.7 288×64 C-mesh for the RAE2822 Airfoil.

When comparing these four schemes, it is important to take into account the actual computational expense of each type of preconditioned multigrid cycle. For this purpose, the entire convergence histories for the four calculations are plotted as a function of CPU time in Fig. 3.9b. To reach a residual level of 10^{-4}, the new approach ($P_M MG_J$) requires 114 cycles and 1234.0 s, while the standard approach ($P_S MG_{Full}$) requires 2262 cycles and 13,310.3 s. The intermediate scheme using scalar preconditioning and J-coarsened multigrid ($P_S MG_J$) requires 589 cycles and 5664.2 s while the other intermediate scheme using matrix preconditioning and full coarsened multigrid ($P_M MG_{Full}$) requires 723 cycles and 4763.8 s. Although the first of the intermediate schemes requires fewer multigrid cycles than the second, the lower cost per cycle makes the second intermediate approach more efficient at the level of engineering accuracy. Compared to the standard method, the CPU speed-up using this second intermediate scheme is a factor of 2.79 in initial convergence, which is roughly the same degree of acceleration observed using the identical approach for the Euler equations. For situations in which it is infeasible to implement J-coarsening, it is therefore still beneficial to adopt the matrix preconditioner

when using full coarsened multigrid due to the substantial improvement in initial convergence rate. The use of J-coarsening in conjunction with the matrix preconditioner then yields further savings of a factor of 3.86 for a total savings over the standard approach of 10.79.

Results for RAE2822 AGARD Case 9 are shown for the new and standard schemes in Fig. 3.10 for both the SA and BL turbulence models. As before, the BL model predicts a stronger shock somewhat aft of that predicted by the SA turbulence model, though in this case the SA result is in better agreement with the experimental measurements [14]. Using the SA turbulence model, the shock induces a very small region of separation measuring roughly 0.5% of chord while the stronger shock predicted by the BL model produces a separation bubble that measures about 5% of chord. From Fig. 3.10b it is evident that the new and standard schemes converge at rates similar to those observed for Case 6. Once again, the new approach yields convergence to machine accuracy in just under 500 cycles while the standard approach exhibits the usual degradation in convergence after about three orders of magnitude. The computational savings at a residual level of 10^{-4} are 11.06 and 12.12 using the SA and BL turbulence models, respectively. The CPU speed-up for asymptotic performance is 42.76 using the SA turbulence model, which is nearly identical to the results for Case 6. The asymptotic convergence rate of the standard scheme is somewhat slower using the BL model, increasing the asymptotic speed-up to 58.45.

3.3.2 Three Dimensions

To compare the performance of the new and standard approaches for three-dimensional turbulent Navier–Stokes computations, a series of test cases are chosen to systematically introduce the influence of the third coordinate direction into the flow:

- straight wing with spanwise periodicity,
- swept wing with spanwise periodicity,
- twisted wing with spanwise periodicity,

Wing	Section	AR	SW	TW	Mesh	M_∞	α	Re_L	Span b.c.
Straight	RAE2822	4.0	0.0°	0.0°	288×64×16	0.725	2.4°	6.5×10^6	Periodicity
Swept	RAE2822	4.0	30.0°	0.0°	288×64×16	0.800	2.8°	6.5×10^6	Periodicity
Twisted	RAE2822	4.0	0.0°	2.0°	288×64×16	0.725	3.4°	6.5×10^6	Periodicity
Fully-3D	RAE2822	8.0	30.0°	3.5°	288×64×16	0.800	4.0°	6.5×10^6	Symmetry

Table 3.6 Three-dimensional turbulent Navier–Stokes test case definitions: wing name, airfoil section, aspect ratio, sweep, twist, mesh dimensions, free stream Mach number, angle of attack, Reynolds number, and type of boundary condition in the spanwise direction.

- fully-3D swept twisted wing with symmetry planes at the root and tip.

The RAE2822 airfoil section is used to construct all test cases and the details of the geometry and flow conditions for each case are provided in Table 3.6. The first case is essentially a two-dimensional flow corresponding to AGARD Case 6 [14] and provides an opportunity for direct comparison with the two-dimensional flow solver [60, 58]. The second case introduces a constant spanwise velocity into the flow and the third case introduces a periodically varying spanwise velocity into the flow. The final case provides a fully three-dimensional wing flow. Symmetry planes are placed at both the root and tip sections to avoid the issue of mesh singularities emanating from the wing tip. This geometry therefore resembles a wind tunnel model with end plates. For each of the test geometries, the flow conditions were modified so as to produce a pressure distribution with a shock strength resembling that of Case 6.

Initial ($10^0 \rightarrow 10^{-4}$) and asymptotic ($10^{-4} \rightarrow 10^{-8}$) convergence information for all the test cases is provided in various useful forms in Table 3.7 . The asymptotic rate of the standard approach is too slow to permit convergence of the solution to a residual level of 10^{-8}, so the asymptotic rates for this approach are extrapolated based on the convergence during the first 500 cycles after 10^{-4} has been reached. This estimate is generous since some degradation in the asymptotic convergence rate continues to occur beyond this point in the convergence history. For three-dimensional calculations, the turbulent viscosity was generally frozen after the density had converged by four orders of magnitude, although in a few cases it was

	Wing	Turb Model	Cycles		Rate		WC Time (s)		Cost Ratio
			Standard	New	Standard	New	Standard	New	
Initial	Straight	SA	2120	109	.9957	.9187	44,188	5000	8.84
		BL	1935	113	.9953	.9213	24,407	3257	7.49
	Swept	BL	1855	113	.9950	.9190	23,409	3340	7.01
	Twisted	SA	2113	124	.9957	.9268	46,362	5672	8.17
	Fully-3D	BL	5579	131	.9984	.9321	54,559	2904	18.79
Asymptotic	Straight	SA	9013	199	.9990	.9548	186,780	9185	20.34
		BL	8386	191	.9989	.9528	104,200	5450	19.12
	Swept	BL	6572	196	.9986	.9554	82,693	5605	14.75
	Twisted	SA	9555	207	.9990	.9568	198,160	9436	21.00
	Fully-3D	BL	27,100	213	.9997	.9575	266,710	4685	56.93

Table 3.7 Three-dimensional turbulent Navier–Stokes results: Initial ($10^0 \rightarrow 10^{-4}$) and asymptotic ($10^{-4} \rightarrow 10^{-8}$) convergence comparisons for scalar preconditioning with full coarsened multigrid (standard) vs. block-Jacobi preconditioning with J-coarsened multigrid (new): multigrid cycles, convergence rate per cycle, wall clock time*, wall clock time speed-up. Asymptotic convergence information for the standard method is extrapolated based on the first 500 cycles after reaching 10^{-4}.

necessary to freeze the turbulence field after only three orders.

Straight Wing

The initial convergence rates for the density and turbulent viscosity using the Spalart–Allmaras turbulence model are displayed in Fig. 3.11a. The convergence histories of the two quantities are identical using the new approach and the turbulent viscosity converges somewhat faster than the flow equations using the standard approach. As summarized in Table 3.7, the density residual requires 109 cycles at a rate of .9187 per cycle to reach four orders of magnitude using the new method while the standard approach requires 2120 cycles at a rate of .9957 to reach the same level of convergence. Since each multigrid cycle is more than twice as expensive using the new approach, it is important to take into consideration the actual computational

*Wall clock time for an IBM SP2 with RS6000/590 nodes and a high performance switch.

effort involved in each of these calculations. To ensure that the communication costs of the parallel implementation are accounted for, all cost comparisons are made on the basis of wall clock time. A wall clock time cost comparison is shown for the present calculations in Fig. 3.11b. The new approach yields savings of a factor of 8.84 in reaching a residual level of 10^{-4} and a factor of 20.34 in asymptotic performance.

The convergence rate of the new approach using the 3D code is compared to the convergence history on the same mesh using the 2D code in Fig. 3.12a. These calculations were performed with both Spalart–Allmaras and Baldwin–Lomax turbulence models to demonstrate that the new approach is insensitive to the choice of turbulence model. The initial convergence of the 2D and 3D codes is nearly identical, with the 2D code eventually converging to machine accuracy in about 450 cycles and the 3D code requiring about 500 cycles. This slight decrease in asymptotic convergence for the 3D implementation results from freezing the turbulent viscosity after four orders of magnitude in order to prevent the turbulence models from interfering even more substantially with the convergence process. The corresponding solutions are displayed in Fig. 3.12b, where the two implementations of the SA model yield almost identical results, while the two BL implementations produce slightly different shock locations that are both in better agreement with the experimental location [14]. The computational savings of the new approach are slightly less when using the Baldwin–Lomax turbulence model for this test case, yielding a factor of 7.49 in initial convergence and 19.12 in asymptotic convergence.

Swept Wing

The convergence histories for a periodically swept wing with constant spanwise velocity are shown in Fig. 3.13. Both the Mach number and angle of attack were increased for this flow, as the spanwise relief would otherwise eliminate the shock on the upper surface. Using the Baldwin–Lomax turbulence model, the new approach converges four orders of magnitude in 113 cycles at a rate of .9190 and the standard approach requires 1855 cycles at a rate of .9950, corresponding to computational

savings of 7.01 in initial convergence and 14.75 in asymptotic rate. As in the case of the straight wing with two-dimensional flow, the new method converges to machine accuracy in about 500 cycles.

Twisted Wing

Convergence results for the periodically twisted wing are shown in Fig. 3.14 for a calculation employing the Spalart–Allmaras turbulence model. The twist produces incidence angles ranging between $\pm 1^\circ$ of the standard AGARD Case 6 value of 2.4°. For the case of non-uniform spanwise velocity, the new approach now requires 124 multigrid cycles to reach four orders of magnitude compared to 2113 using the standard approach, which represents a computational savings of 8.17. The computational savings in terms of asymptotic convergence rate are now a factor of 21.00. The number of cycles required for the new approach to converge to machine accuracy increases slightly to about 550.

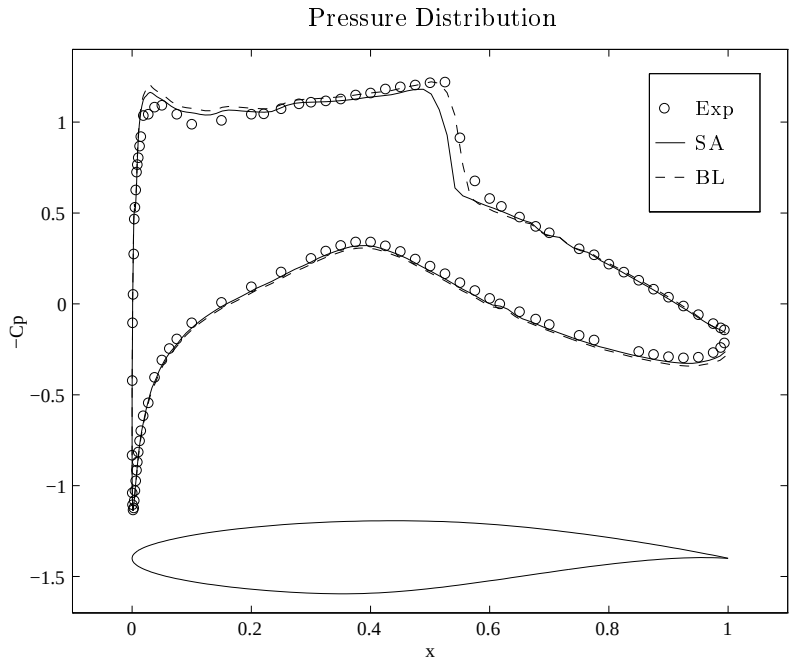
Fully-3D Wing

This geometry was constructed using a swept planform with constant chord and a linear twist distribution along the span. The pressure distributions at six stations along the wing are shown in Fig. 3.15. The presence of the end plate at the tip does not substantially alter the flow field over the majority of the wing, as is evident from the typical root and central pressure distributions. However, the plate does have a significant effect on the flow near the tip, where the lack of three-dimensional relief tends to introduce a very strong shock. The choice of the maximum twist angle was therefore largely governed by the desire to keep the tip flow from separating. Note that the small wiggles in the pressure distribution are not numerical oscillations but rather the effect of imperfections in the measured experimental coordinates of the original RAE2822 test section [14].

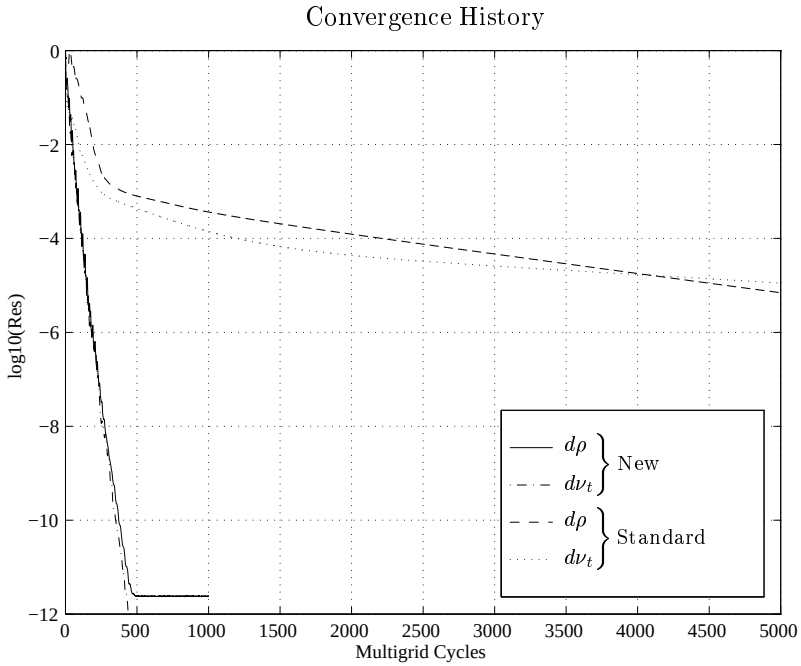
The convergence results for the new and standard methods are displayed in terms of multigrid cycles and wall clock time in Figs 3.16a and 3.16b. The introduction of full three-dimensionality into the flow field has very little effect upon the convergence

rate of the new approach. Four orders of convergence are achieved in 131 cycles at a rate of .9321 and the next four orders require only an additional 213 cycles at a rate of .9575. As in the case of periodic twist, the solution converges to machine accuracy in about 550 cycles. The convergence of the standard approach is substantially slower than for the previous cases, requiring 5579 cycles to reach four orders at a rate of .9984 and an extrapolated requirement of 27,100 additional cycles to converge the next four orders. The new approach therefore provides computational savings in terms of wall clock time of 18.79 in initial convergence and 56.93 in asymptotic convergence.

It remains to extend the method to treat exposed wing tips and other complex configurations which introduce mesh singularities that do not fall within the assumptions of the theoretical analysis used to motivate this approach. Extensive testing has not yet been performed for wings with exposed tips, but preliminary studies suggest that the void produced at the wing tip using a C-H topology does reduce the robustness of the approach.

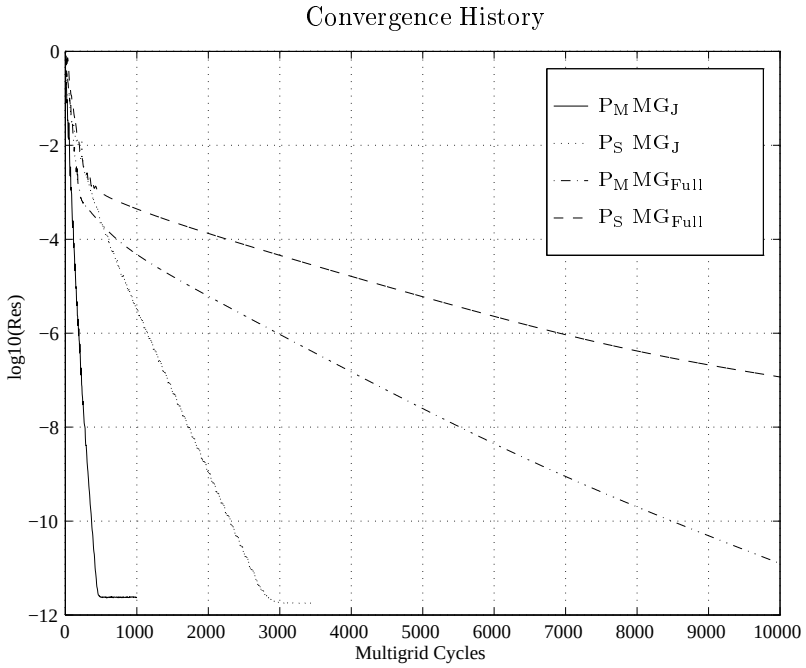


3.8a Coefficient of pressure.

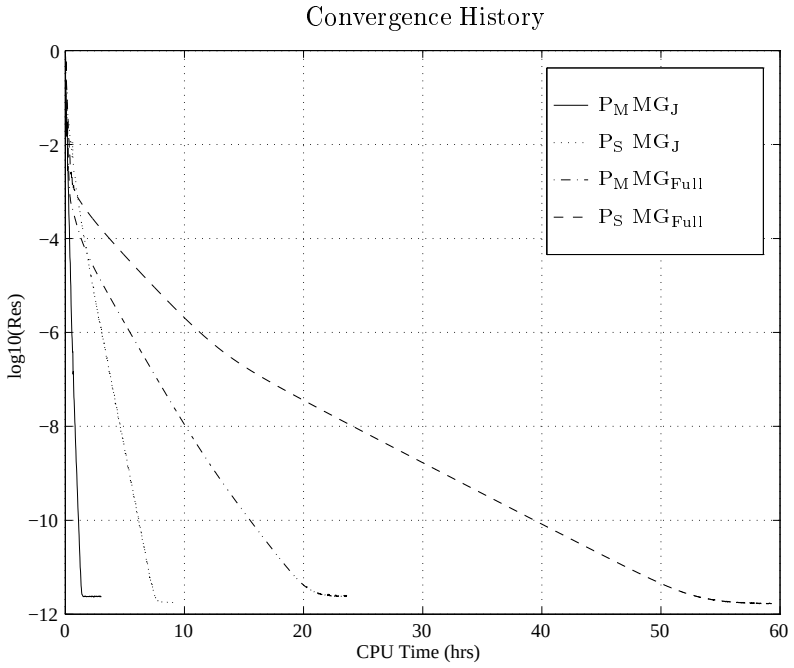


3.8b Convergence comparison using the SA model.

Figure 3.8 Test NS1: Solution and convergence comparisons.
RAE2822 AGARD Case 6. $M_\infty = 0.725$, $\alpha = 2.4$, Re $= 6.5\times10^6$, 288×64 C-mesh.

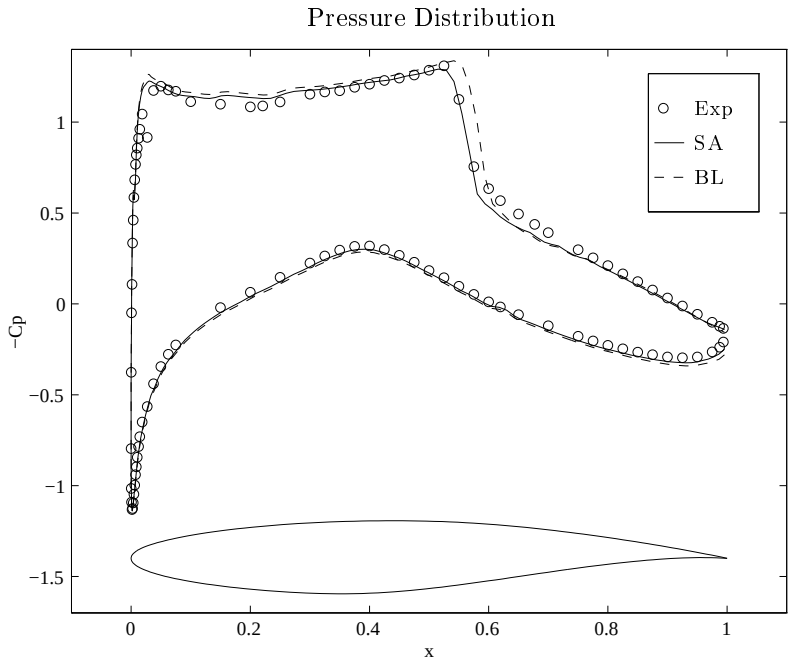


3.9a Convergence comparison using the BL model.

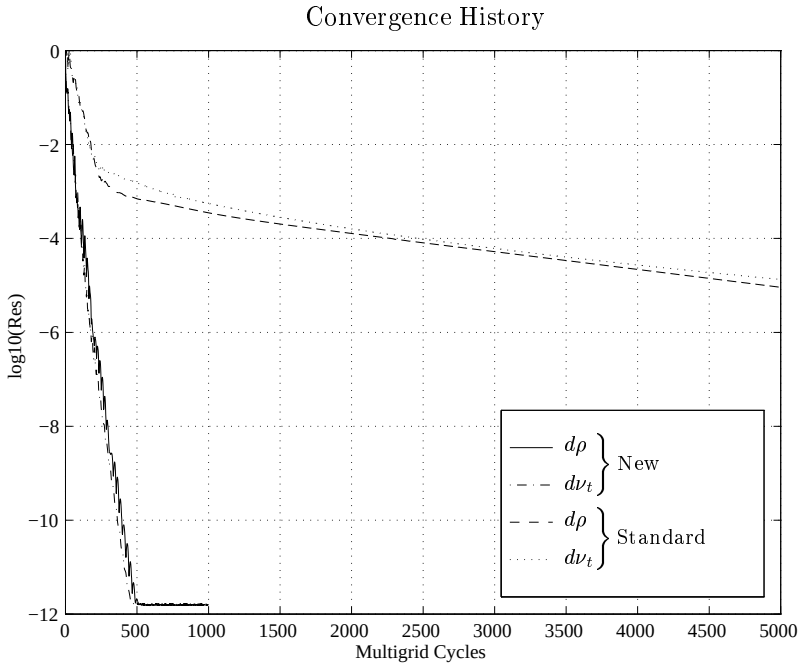


3.9b CPU cost comparison using the BL model.

Figure 3.9 Test NS1: Convergence and cost comparisons.
RAE2822 AGARD Case 6. $M_\infty = 0.725$, $\alpha = 2.4$, Re $= 6.5\times10^6$, 288×64 C-mesh.

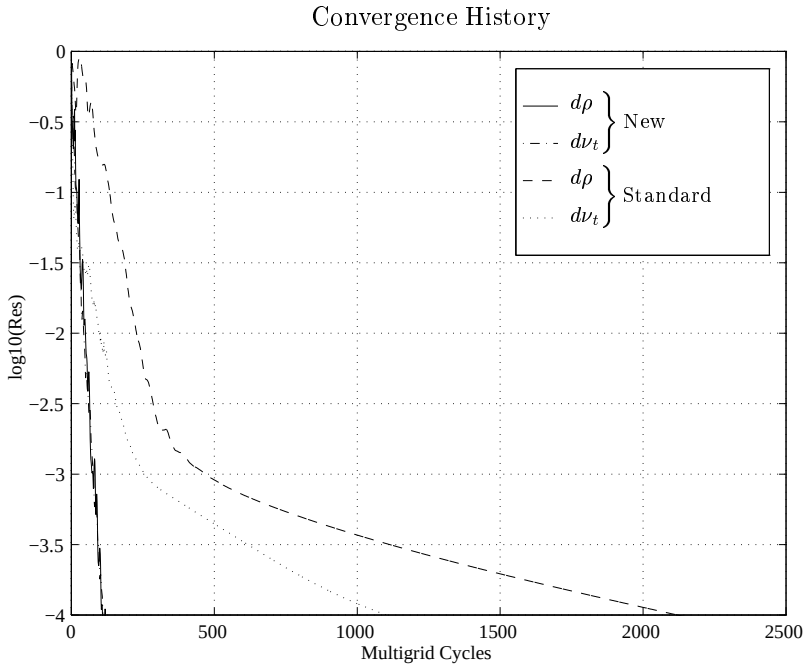


3.10a Coefficient of pressure.

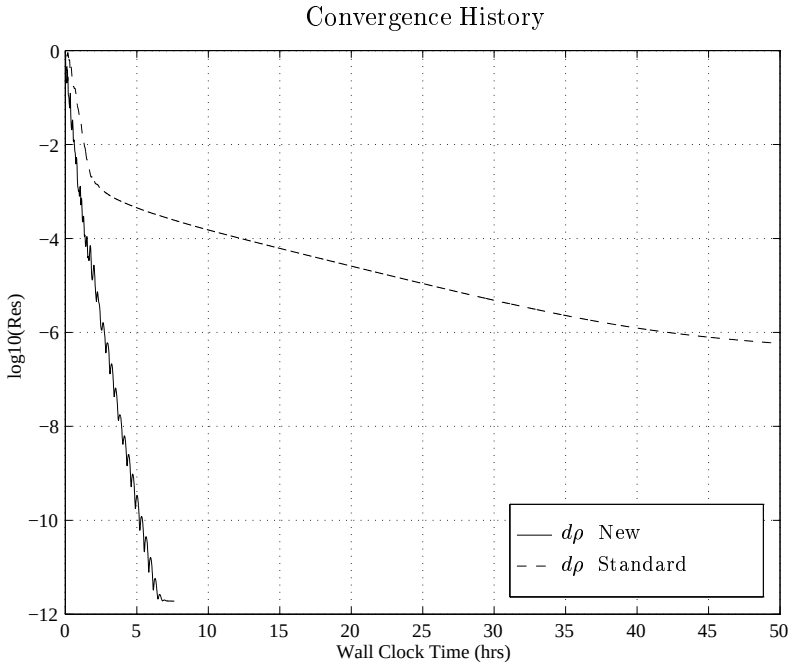


3.10b Convergence comparison using the SA and BL models.

Figure 3.10 Test NS2: Solution and convergence comparisons.
RAE2822 AGARD Case9. $M_\infty = 0.73, \alpha = 2.79$, Re $= 6.5\times10^6$, 288×64 C-mesh.

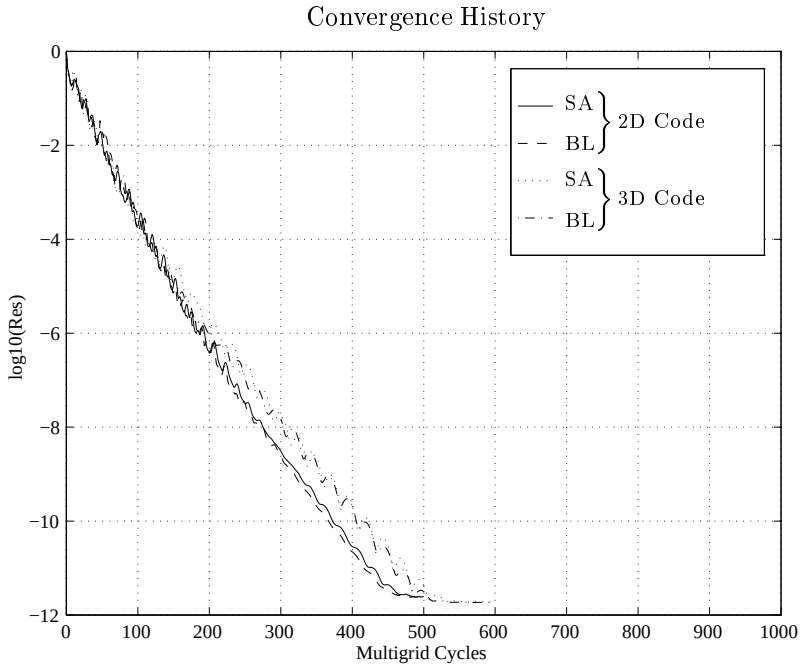


3.11a Initial convergence of the density and turbulent viscosity residuals.

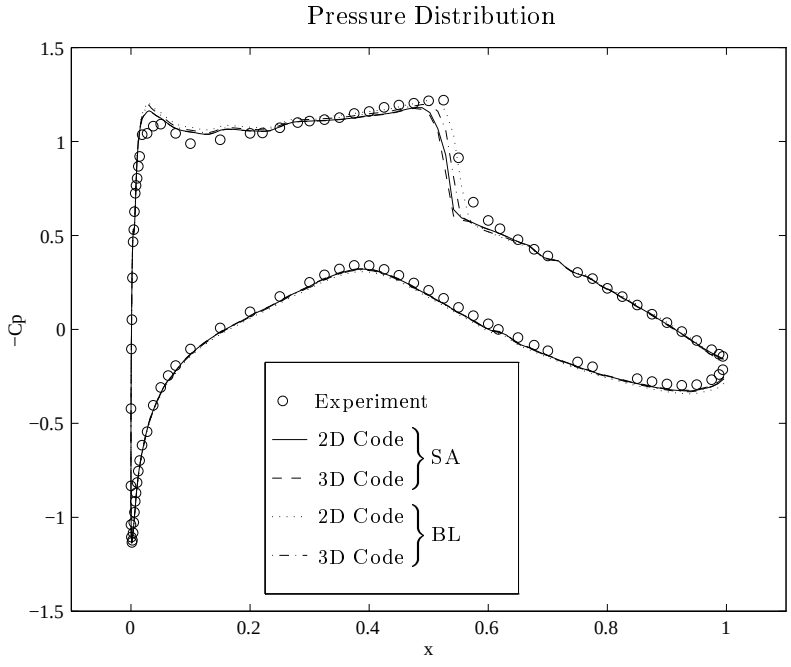


3.11b Wall clock time cost comparison.

Figure 3.11 Straight Wing: Convergence and cost comparisons using the SA model.



3.12a Convergence of the density residual.



3.12b Coefficient of pressure.

Figure 3.12 Straight Wing: Comparison of the 2D and 3D codes using the SA and BL models.

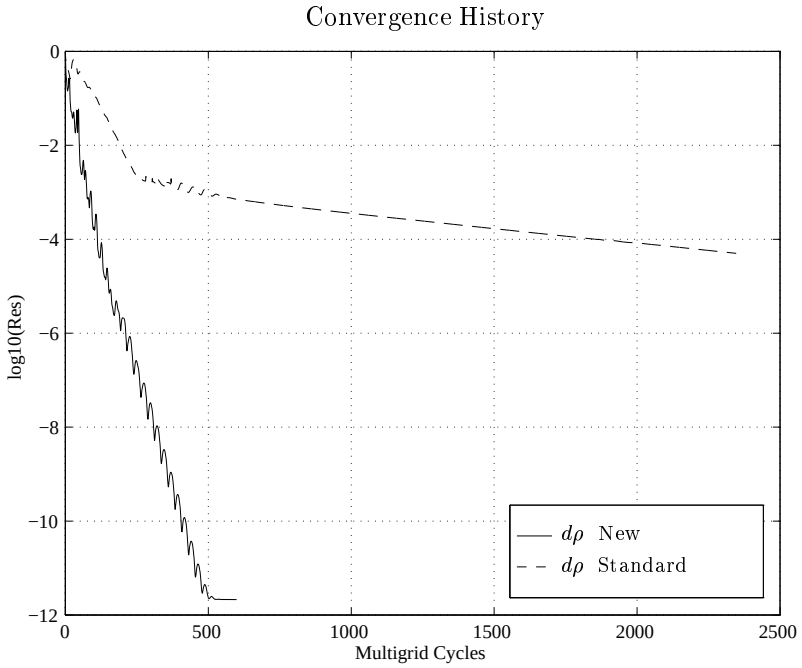


Figure 3.13 Swept Wing: Convergence comparison using the BL model.

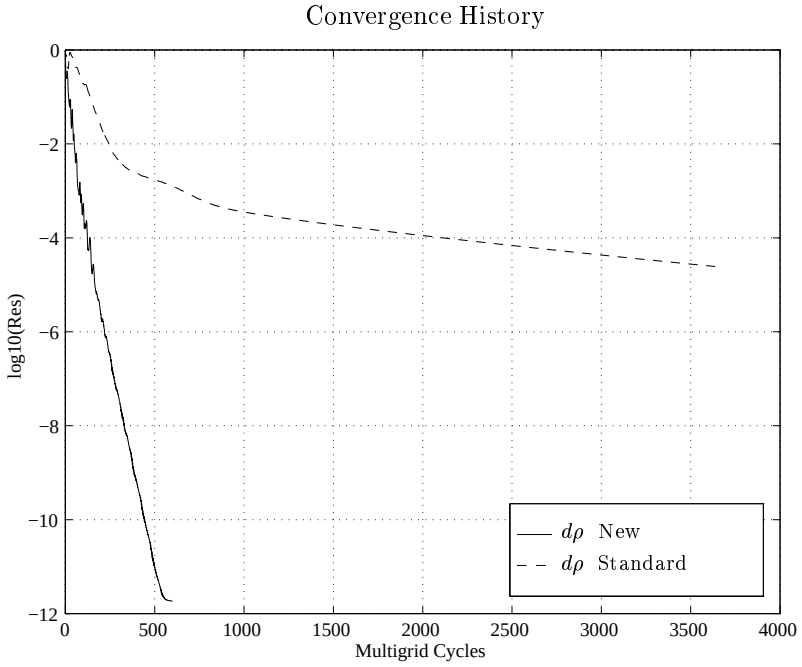
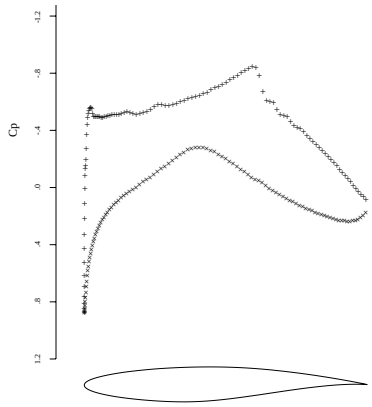
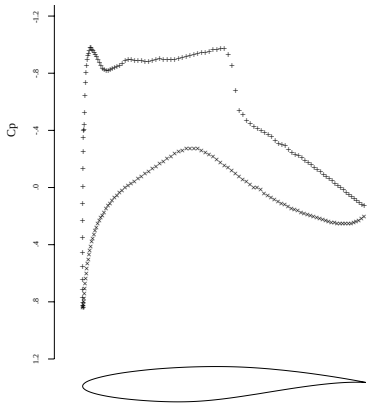


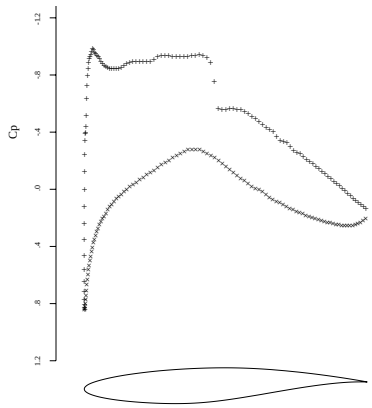
Figure 3.14 Twisted Wing: Convergence comparison using the SA model.



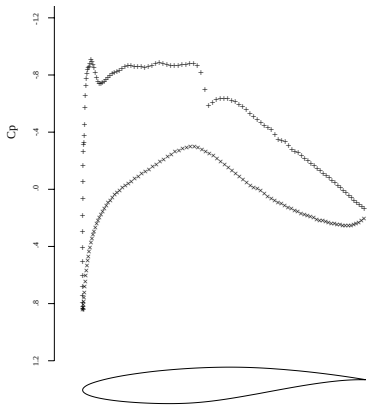
3.15a C_p at span station $z = 0.25$.



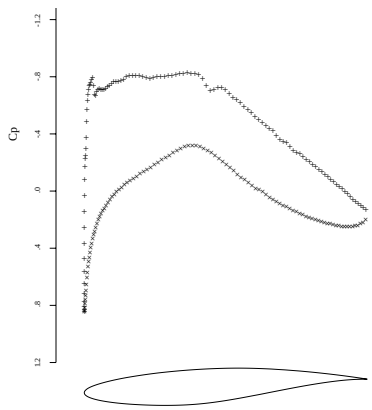
3.15b C_p at span station $z = 1.75$.



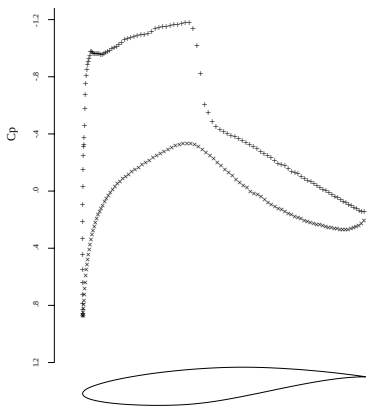
3.15c C_p at span station $z = 3.25$.



3.15d C_p at span station $z = 4.75$.

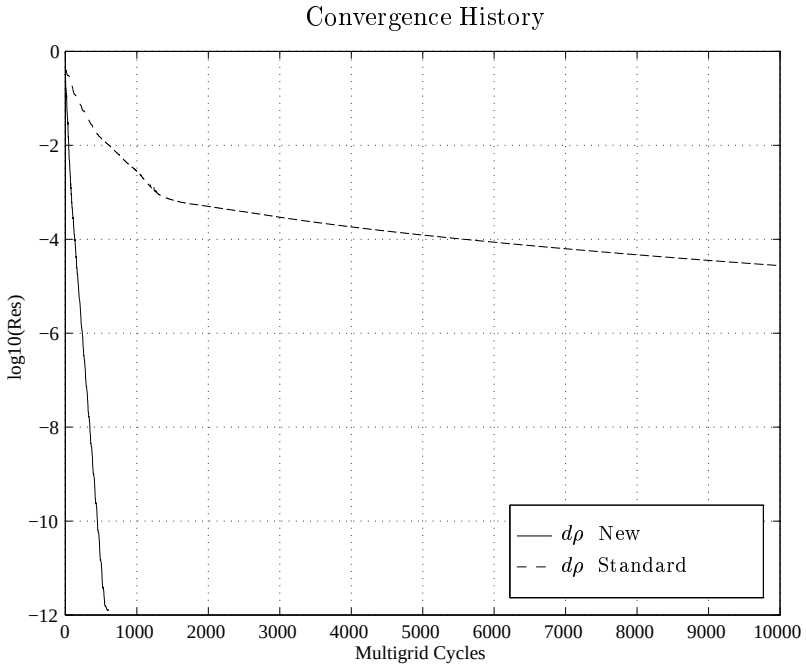


3.15e C_p at span station $z = 6.25$.

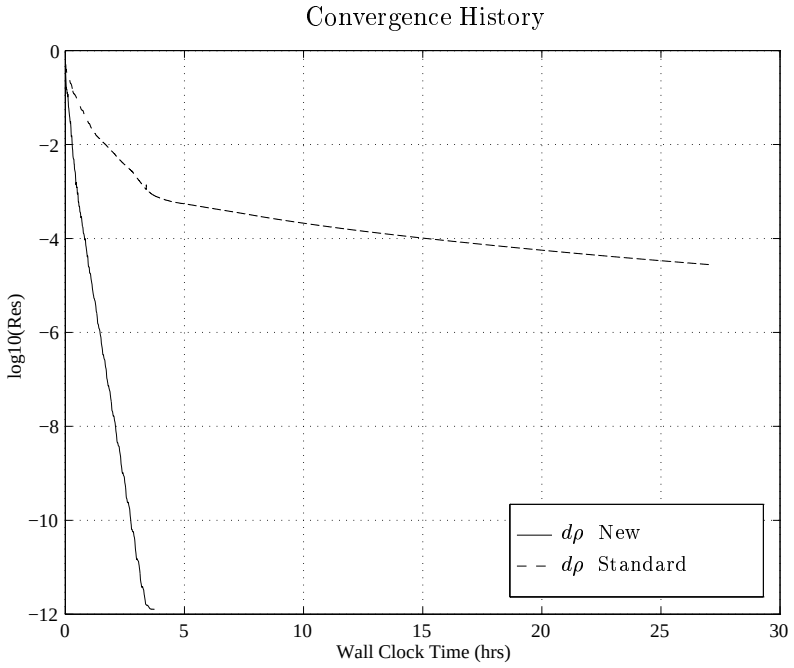


3.15f C_p at span station $z = 7.75$.

Figure 3.15 Fully-3D Wing: Pressure distributions for a wing with endplates. RAE2822 Airfoil, Aspect Ratio = 8, Sweep = 30°, Twist = 3.5°, 288×64×16 C-H mesh. $M = 0.8, \alpha = 4.0°, Re = 6.5{\times}10^6$.



3.16a Convergence comparison.



3.16b Wall clock time cost comparison.

Figure 3.16 Fully-3D Wing: Convergence and cost comparisons using the BL model.

Chapter 4

Conclusions

Efficient preconditioned multigrid methods have been proposed, analyzed and implemented for both inviscid and viscous flow applications. The standard scheme currently in widespread use employs a scalar preconditioner (local time step) with full coarsened multigrid. This approach works relatively well for Euler calculations but is less effective for turbulent Navier–Stokes calculations as a result of the discrete stiffness and directional decoupling introduced by the highly stretched cells in the boundary layer.

For Euler calculations on moderately stretched meshes, numerical studies of the preconditioned Fourier footprints demonstrate that a block-Jacobi matrix preconditioner substantially improves the damping and propagative efficiency of Runge–Kutta time-stepping schemes for use with full coarsened multigrid. In comparison to the standard method, the computational savings using this approach are roughly a factor of three for convergence to engineering accuracy and between a factor of three and five for asymptotic convergence.

For turbulent Navier–Stokes flows, a new scheme based on block-Jacobi preconditioning and J-coarsened multigrid is shown to provide effective damping of all error modes inside the boundary layer. Analytic expressions for the preconditioned Fourier footprints inside an asymptotically stretched boundary layer cell reveal that the balance between streamwise convection and normal diffusion enables the preconditioner to damp all convective modes. Adoption of a J-coarsened strategy, in

which coarsening is performed only in the direction normal to the wall, then ensures that all acoustic modes are damped. The new scheme provides rapid and robust convergence to machine accuracy for turbulent Navier–Stokes calculations in both two and three dimensions. The computational savings relative to the standard approach are roughly an order of magnitude for engineering accuracy and much larger in terms of asymptotic performance. The analysis and computational results indicate that the initial convergence of the standard approach is primarily limited by convective error modes while the notoriously poor asymptotic performance of the method results principally from acoustic pressure modes that are uniform across the boundary layer and rapidly varying in the streamwise direction.

Substantial further work will be necessary before the new preconditioned multigrid method can be incorporated in production flow solvers for complex configurations. The first priority is the demonstration of robust convergence in the presence of topological mesh singularities such as those produced at the wing tip of a single block mesh. For applicability to calculations involving complex configurations, additional work will be required to develop a flexible multi-block implementation that allows local coarsening directions in each block. Alternatively, the method can be implemented in the context of unstructured grids. To improve the damping and propagation of error modes at low Mach numbers, the advantages of incorporating a low-Mach number preconditioner into the numerical dissipation and hence into the block-Jacobi preconditioner should be further explored. Additional research is also required into the possibility of using a line-implicit preconditioner either in lieu of, or possibly in conjunction with, directional coarsening.

Bibliography

[1] S. Allmaras. Analysis of a local matrix preconditioner for the 2-D Navier–Stokes equations. AIAA Paper 93-3330-CP, 11th Computational Fluid Dynamics Conference, Orlando, FL, 1993.

[2] S. Allmaras. Analysis of semi-implicit preconditioners for multigrid solution of the 2-D compressible Navier–Stokes equations. AIAA Paper 95-1651-CP, 12th Computational Fluid Dynamics Conference, San Diego, CA, 1995.

[3] J.J. Alonso, T.J. Mitty, L. Martinelli, and A. Jameson. A two-dimensional multigrid-driven Navier-Stokes solver for multiprocessor architectures. In *Proceedings of the Parallel CFD '94 Conference*, Kyoto, Japan, 1994.

[4] J.D. Anderson. *Fundamentals of Aerodynamics, 2nd Ed.* McGraw-Hill, Inc., 1991.

[5] N.S. Bakhvalov. On the convergence of a relaxation method with natural constraints of the elliptic operator. *Zh. Vychisl. Mat. Mat. Fiz.*, 6(5):861–885, 1966. *(USSR Comput. Math. Math. Phys.*, 6:101-135, 1966).

[6] B. Baldwin and H. Lomax. Thin layer approximation and algebraic model for separated turbulent flows. AIAA Paper 78-257, 1978.

[7] R.M. Beam and R.F. Warming. An implicit finite-difference algorithm for hyperbolic systems in conservation-law form. *J. Comput. Phys.*, 22:87–110, 1976.

[8] J.A. Benek, P.G. Buning, and J.L. Steger. A 3-D Chimera grid imbedding technique. AIAA Paper 85-1523, 7th Computational Fluid Dynamics Conference, Cincinatti, OH, 1985.

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

[10] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31(138):333–390, 1977.

[11] P.E.O. Buelow, S. Venkateswaran, and C.L. Merkle. Effect of grid aspect ratio on convergence. *AIAA Journal*, 32(12):2401–2408, 1994.

[12] Y.-H. Choi and C.L. Merkle. The application of preconditioning in viscous flows. *J. Comput. Phys.*, 105:207–223, 1993.

[13] A.J. Chorin. A numerical method for solving incompressible viscous flow problems. *J. Comput. Phys.*, 2:12–26, 1967.

[14] P.H. Cook, M.A. McDonald, and M.C.P. Firmin. Aerofoil RAE2822 pressure distributions, boundary layer and wake measurements. AGARD Advisory Report No. 138, 1979.

[15] R. Courant, K.O. Friedrichs, and H. Lewy. Über die partiellen differenzengleichungen der mathematischen physik. *Math. Ann.*, 100:32–74, 1928. *(IBM J.*, pp. 215–234, 1967).

[16] P.I. Crumpton and M.B. Giles. Implicit time accurate solutions on unstructured grids. AIAA Paper 95-1671, 12th Computational Fluid Dynamics Conference, San Diego, CA, 1995.

[17] P.I. Crumpton, P. Moinier, and M.B. Giles. Calculation of turbulent flow on unstructured grids with high aspect ratio. In *Proc. 10th International Conference on Numerical Methods for Laminar and Turbulent Flow*, Swansea, UK, July, 1997.

[18] D.L. Darmofal and P.J. Schmid. The importance of eigenvectors for local preconditioners of the Euler equations. *J. Comput. Phys.*, 127:346–362, 1996.

[19] E. Dick and K. Riemslagh. Multi-stage Jacobi relaxation as smoother in a multigrid method for steady Euler equations. In *Proc. ICFD Conference on Numerical Methods for Fluid Dynamics*, pp. 533–539, 1993.

[20] R.P. Fedorenko. The speed of convergence of one iterative process. *Zh. Vychisl. Mat. Mat. Fiz.*, 4(3):559–564, 1964. *(USSR Comput. Math. Math. Phys.*, 4:227-235, 1964).

[21] M.B. Giles and W.T. Thompkins, Jr. Propagation and stability of wavelike solutions of finite difference equations with variable coefficients. *J. Comput. Phys.*, 58(3):349–360, 1985.

[22] A.R. Gourlay and A.R. Mitchell. A stable implicit difference method for hyperbolic systems in two space variables. *Num. Math.*, 8:367–375, 1966.

[23] W. Hackbusch. On the multi-grid method applied to difference equations. *Computing*, 20:291–306, 1978.

[24] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[25] C. Hirsch. Private communication, 1997.

[26] C. Hirsch. *Numerical Computation of Internal and External Flows: Vol. 2, Computational Methods for Inviscid and Viscous Flows.* John Wiley & Sons, 1990.

[27] A. Jameson. Solution of the Euler equations by a multigrid method. *Appl. Math. Comput.*, 13:327–356, 1983.

[28] A. Jameson. Transonic flow calculations for aircraft. In F. Brezzi, Ed., *Lecture Notes in Mathematics, Numerical Methods in Fluid Dynamics*, pp. 156–242. Springer–Verlag, 1985.

[29] A. Jameson. Multigrid algorithms for compressible flow calculations. In W. Hackbusch and U. Trottenberg, Eds., *Lecture Notes in Mathematics, Vol. 1228: Multigrid Methods II*, pp. 166–201. Springer–Verlag, 1986. Second European Conference on Multigrid Methods, Cologne, 1985.

[30] A. Jameson. Aerodynamic design via control theory. *J. Sci. Comput.*, 3:233–260, 1988.

[31] A. Jameson. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. AIAA Paper 91-1596, 10th Computational Fluid Dynamics Conference, Honolulu, HI, 1991.

[32] A. Jameson. Analysis and design of numerical schemes for gas dynamics 1: Artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence. *Int. J. Comput. Fluid Dyn.*, 4:171–218, 1995.

[33] A. Jameson and D.A. Caughey. A finite volume method for transonic potential flow calculations. AIAA Paper 77-635, 1977.

[34] A. Jameson and L. Martinelli. Mesh refinement and modeling errors in flow simulation. AIAA Paper 96-2050, 1996.

[35] A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. AIAA Paper 97-0101, 35th Aerospace Sciences Meeting, Reno, NV, 1997. (*J. Theor. Comput. Fluid Mech.*, to appear).

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

[37] A. Jameson and E. Turkel. Implicit schemes and LU decompositions. *Math. Comp.*, 37(156):385–397, 1981.

[38] A. Jameson and S. Yoon. Lower-upper implicit schemes with multiple grids for the Euler equations. *AIAA J.*, 25(7):929–935, 1987.

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

[40] C.P. Li. Numerical solution of the viscous reacting blunt body flows of a multicomponent mixture. AIAA Paper 73-202, 1973.

[41] J. Lighthill. *Waves in Fluids.* Cambridge University Press, 1978.

[42] F. Liu and A. Jameson. Multigrid Navier–Stokes calculations for three-dimensional cascades. AIAA Paper 92-0190, 1992.

[43] F. Liu and X. Zheng. A strongly coupled time-marching method for solving the Navier–Stokes and k-ω turbulence model equations with multigrid. *J. Comput. Phys.*, 128:289–300, 1996.

[44] J.F. Lynn and B. van Leer. A semi-coarsened multigrid solver for the Euler and Navier–Stokes equations. AIAA Paper 95-1667-CP, 1995.

[45] L. Martinelli. *Calculations of Viscous Flows with a Multigrid Method.* PhD thesis, Princeton University, 1987.

[46] L. Martinelli and A. Jameson. Validation of a multigrid method for the Reynolds averaged equations. AIAA Paper 88-0414, 26th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1988.

[47] D.J. Mavriplis. Multigrid strategies for viscous flow solvers on anisotropic unstructured grids. AIAA Paper 97-1952, 13th Computational Fluid Dynamics Conference, Snowmass, CO, 1997.

[48] D.J. Mavriplis and V. Venkatakrishnan. Multigrid techniques for unstructured meshes. ICASE Report No. 95-30, 1995.

[49] B.T. McCann and D.L. Darmofal. Evaluation of local preconditioners for multigrid solutions of the compressible Euler equations. AIAA Paper 97-2028, 13th Computational Fluid Dynamics Conference, Snowmass, CO, 1997.

[50] E. Morano, M.-H. Lallemand, M.-P. Leclercq, H. Steve, B. Stoufflet, and A. Dervieux. Local iterative upwind methods for steady compressible flows. In *GMD-Studien Nr. 189, Multigrid Methods: Special Topics and Applications II*, pp. 227–238. Springer–Verlag, 1991. 3rd European conference on multigrid methods, Bonn, 1990.

[51] W.A. Mulder. A new multigrid approach to convection problems. *J. Comput. Phys.*, 83:303–323, 1989.

[52] R.-H. Ni. A multiple-grid scheme for solving the Euler equations. *AIAA J.*, 20(11):1565–1571, 1982.

[53] R.A. Nicholaides. On the l^2 convergence of an algorithm for solving finite element equations. *Math. Comp.*, 31:892–906, 1977.

[54] C.F. Ollivier-Gooch. Towards problem-independent multigrid convergence rates for unstructured methods I: Inviscid and laminar viscous flows. In *Sixth International Symposium on Computational Fluid Dynamics*, pp. 913–918, Lake Tahoe, NV, 1995.

[55] N.A. Pierce. Characteristic-based dissipative schemes and characteristic time stepping for the Euler equations. Qualifying dissertation for DPhil status, Oxford University, 1994.

[56] N.A. Pierce and J.J. Alonso. Efficient computation of unsteady viscous flows by an implicit preconditioned multigrid method. *AIAA J.*, in revision.

[57] N.A. Pierce and J.J. Alonso. A preconditioned implicit multigrid algorithm for parallel computation of unsteady aeroelastic compressible flows. AIAA Paper 97-0444, 35th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1997.

[58] N.A. Pierce and M.B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. *J. Comput. Phys.*, to appear.

[59] N.A. Pierce and M.B. Giles. Preconditioning on stretched meshes. Oxford University Computing Laboratory Technical Report 95/10, Presented at the 12th AIAA Computational Fluid Dynamics Conference, San Diego, CA, 1995.

[60] N.A. Pierce and M.B. Giles. Preconditioning compressible flow calculations on stretched meshes. AIAA Paper 96-0889, 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1996.

[61] N.A. Pierce, M.B. Giles, A. Jameson, and L. Martinelli. Accelerating three-dimensional Navier–Stokes calculations. AIAA Paper 97-1953, 13th Computational Fluid Dynamics Conference, Snowmass, CO, 1997.

[62] T.H. Pulliam and J.L. Steger. Recent improvements in efficiency, accuracy and convergence for implicit approximate factorization algroithms. AIAA Paper 85-0360, 23rd Aerospace Sciences Meeting and Exhibit, Reno, NV, 1985.

[63] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems.* Wiley–Interscience, 1967.

[64] P.L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[65] 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.

[66] W.A. Smith. Multigrid solution of transonic flow on unstructured grids. In O. Baysal, Ed., *Recent Advances and Applications in Computational Fluid Dynamics*, 1990. Proceedings of the ASME Winter Annual Meeting.

[67] P.R. Spalart and S.R. Allmaras. A one-equation turbulence model for aerodynamic flows. AIAA Paper 92-0439, 30th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1992.

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

[69] I. Stewart. *Galois Theory.* Chapman & Hall Mathematics, 1994.

[70] R.C. Swanson and E. Turkel. On central-difference and upwind schemes. *J. Comput. Phys.*, 101:292–306, 1991.

[71] E. Turkel. Fast solutions to the steady state compressible and incompressible fluid dynamics equations. In *Lecture Notes in Physics, Vol. 218*, pp. 571–575. Springer–Verlag, 1984. Ninth International Conference on Numerical Methods in Fluid Dynamics, France, 1984.

[72] E. Turkel. Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comput. Phys.*, 72:277–298, 1987.

[73] E. Turkel. Review of preconditioning methods for fluid dynamics. *Appl. Num. Math.*, 12:257–284, 1993.

[74] E. Turkel. Preconditioning-squared methods for multidimensional aerodynamics. AIAA Paper 97-2025, 13th Computational Fluid Dynamics Conference, Snowmass, CO, 1997.

[75] E. Turkel, A. Fiterman, and B. van Leer. Preconditioning and the limit of the compressible to the incompressible flow equations for finite difference schemes. In D.A. Caughey and M.M. Hafez, Eds., *Frontiers of Computational Fluid Dynamics*, pp. 215–234. John Wiley & Sons, 1994.

[76] B. van Leer, W.-T. Lee, and K.G. Powell. Sonic-point capturing. AIAA Paper 89-1945-CP, 9th Computational Fluid Dynamics Conference, 1989.

[77] B. van Leer, W.-T. Lee, and P.L. Roe. Characteristic time-stepping or local preconditioning of the Euler equations. AIAA Paper 91-1552-CP, 1991.

[78] B. van Leer, W.-T. Lee, P.L. Roe, and C.-H. Tai. Design of optimally smoothing multi-stage schemes for the Euler equations. *J. Appl. Num. Math.*, 1991.

[79] B. van Leer, L. Mesaros, C.-H. Tai, and E. Turkel. Local preconditioning in a stagnation point. AIAA Paper 95-1654-CP, 12th Computational Fluid Dynamics Conference, San Diego, CA, 1995.

[80] S. Venkataswaran, P.E.O. Buelow, and C.L. Merkle. Development of linearized preconditioning methods for enhancing robustness and efficiency for Euler and Navier–Stokes computations. AIAA Paper 97-2030, 1997.

[81] S. Venkataswaran, J.M. Weiss, C.L. Merkle, and Y.-H. Choi. Propulsion-related flowfields using the preconditioned Navier–Stokes equations. AIAA Paper 92-3437, 1992.

[82] H. Viviand. Pseudo-unsteady systems for steady inviscid flow calculations. In F. Angrand *et al.*, Eds., *Numerical Methods for the Euler Equations of Fluid Dynamics*, pp. 334–368. SIAM, 1985.

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

[84] S. Wolfram. *Mathematica: A system for doing mathematics by computer.* Addison–Wesley, 1988.

Appendix A

Damping and Propagative Mechanisms

Transient errors in the discrete solution are eliminated by a combination of damping and propagation produced by the interaction between the spatial and temporal discretization operators. To facilitate the examination of these properties for errors arising locally in the flow domain, it is useful to view each disturbance as a discrete wave packet of the type shown in Fig. A.1, where the amplitude of the oscillating error is nonzero in only a small region of the domain. The wave number is assumed to be uniform within each packet so that a given disturbance represents only a single slice of the spatial Fourier spectrum. The packet is eliminated either by damping the amplitude to zero or by propagation out of the computational domain.

In general, the crests within the packet travel with a phase velocity c that differs in magnitude (and possibly direction) from the group velocity c_g at which the packet itself travels. As a result, new oscillations will continuously arise and disappear at opposite ends of the packet as it propagates through the domain. From the point of view of error expulsion, the motion of these individual crests within the packet provides no benefit. Rather, it is the motion of the packet itself that is of interest. Consequently, the relevant quantity for examining propagative effectiveness is the group velocity associated with a given error disturbance. The corresponding measure of damping effectiveness is the the amplification factor, χ, which represents

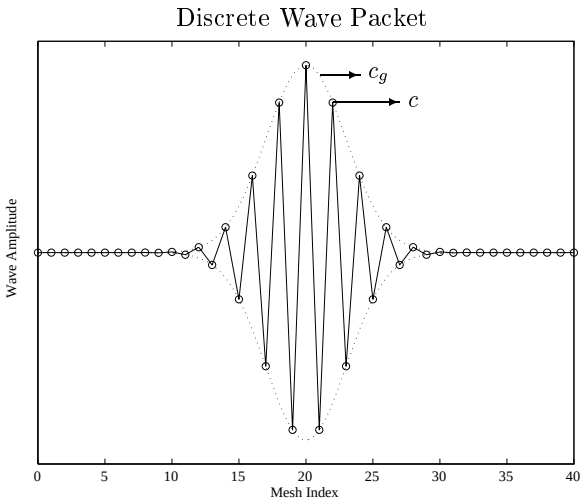


Figure A.1 Discrete wave packet model for a local error with uniform wave number.

the reduction in the packet amplitude from one time level to the next.

A.1 Analytic Properties

To examine the structure of the damping and propagative mechanisms, it is convenient to consider the behavior of solutions to a scalar convection–diffusion equation defined by the linear differential operator,

$$\mathcal{L}(u) \equiv \frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} - \nu\frac{\partial^2 u}{\partial x^2} = 0. \tag{A.1}$$

The analytic solutions are damped sinusoids of the form

$$u = u_0 e^{-\varphi t} e^{\hat{\imath}(kx-\omega t)}, \tag{A.2}$$

where φ is the damping parameter, k is the wave number and ω is the frequency. Substituting (A.2) into (A.1) produces the damping and dispersion relationships

$$\begin{aligned} \mathrm{Re}[\mathcal{L}(u)] = 0 \quad &\Rightarrow \quad \varphi = \nu k^2, \\ \mathrm{Im}[\mathcal{L}(u)] = 0 \quad &\Rightarrow \quad \omega = ak. \end{aligned} \tag{A.3}$$

The amplification factor describing the reduction in amplitude of the wave per unit time is then defined by

$$\chi = e^{-\varphi} = e^{-\nu k^2},$$

and the phase velocity at which the wave crests travel is defined by

$$c = \frac{\omega}{k} = a.$$

The analytic solution is therefore non-dispersive since the phase velocity is independent of the wave number. For a wave packet produced by a linear combination of waves of differing wave number, Lighthill [41] has shown that the analytic group velocity is given by

$$c_g = \frac{\partial \omega}{\partial k}. \tag{A.4}$$

For wave packets composed of non-dispersive analytic solutions of the form (A.2), the group velocity is thus equal to the phase velocity.

A.2 Discrete Properties

As a discrete approximation to (A.1), consider the general linear homogeneous finite difference equation

$$Lu_i^n = 0, \tag{A.5}$$

where i is the mesh index, n is the time level, and the discretization operator L has the form

$$L \equiv \sum_{m,p} C_{mp} S_{mx} S_{pt}.$$

Here, C_{mp} are the coefficients of the difference stencil and the spatial and temporal shift operators are defined by $S_{mx}u_i^n = u_{i+m}^n$ and $S_{pt}u_i^n = u_i^{n+p}$ [21]. To examine the evolution of a discrete wave packet governed by this difference equation, the solution is written as the product of a damped sinusoid and an amplitude that is a function of both space and time

$$u_i^n = A(i,n)e^{-n\phi}e^{\hat{\imath}(i\theta - n\Omega)}, \tag{A.6}$$

where A is the amplitude, ϕ is the damping parameter, θ is the wave number and Ω is the frequency*. Following immediately from this representation, the amplification

*The main body of the dissertation deals exclusively with a semi-discrete representation so there is little call to use the word "frequency" in a temporal sense. It is therefore conveniently appropriated and used to refer to spatial variation in accordance with common usage (e.g. "high frequency modes").

factor describing the reduction in magnitude during each time step is given by

$$\chi = e^{-\phi},$$

and the phase velocity at which the wave crests travel may be expressed as

$$c = \frac{\Omega}{\theta}.$$

The appropriate definition for the discrete group velocity is not readily apparent and requires some further observations.

Following the approach of Giles and Thompkins [21], the amplitude is assumed to be slowly varying in both space and time so that A may be accurately represented within the discretization stencil using a first order Taylor series approximation. Substitution of (A.6) into (A.5) then produces

$$LU_i^n \approx e^{-n\phi} e^{\hat{\imath}(i\theta - n\Omega)} \sum_{m,p} C_{mp} e^{-p\phi} e^{\hat{\imath}(m\theta - p\Omega)} \left[A_{i,n} + p\frac{\partial A}{\partial n} + m\frac{\partial A}{\partial i} \right] = 0.$$

This expression requires that the amplitude satisfies

$$a_0(\phi,\theta,\Omega)A + a_1(\phi,\theta,\Omega)\frac{\partial A}{\partial n} + a_2(\phi,\theta,\Omega)\frac{\partial A}{\partial i} = 0, \tag{A.7}$$

where

$$\begin{aligned} a_0(\phi,\theta,\Omega) &= \sum_{m,p} C_{mp} e^{-p\phi} e^{\hat{\imath}(m\theta - p\Omega)}, \\ a_1(\phi,\theta,\Omega) &= \hat{\imath}\frac{\partial a_0}{\partial \Omega}\bigg|_{\phi,\theta}, \\ a_2(\phi,\theta,\Omega) &= -\hat{\imath}\frac{\partial a_0}{\partial \theta}\bigg|_{\Omega,\phi}. \end{aligned}$$

The coefficient $a_0(\phi,\theta,\Omega)$ represents the substitution of a damped sinusoid into the discretization stencil, so that by analogy with (A.3), the discrete damping and dispersion relationships are found by equating the real and imaginary parts to zero

$$\begin{aligned} \mathrm{Re}[a_0(\phi,\theta,\Omega)] = 0 \quad &\Rightarrow \quad \phi = \phi(\theta), \\ \mathrm{Im}[a_0(\phi,\theta,\Omega)] = 0 \quad &\Rightarrow \quad \Omega = \Omega(\theta). \end{aligned} \tag{A.8}$$

The remaining terms of (A.7) may then be written in the form

$$\frac{\partial A}{\partial n} + (a_2/a_1)\frac{\partial A}{\partial i} = 0,$$

where the group velocity at which the packet amplitude propagates is defined by

$$c_g = \mathrm{Re}[a_2/a_1].$$

This expression for the group velocity may be evaluated by differentiating a_0 with respect to θ

$$\left.\frac{\partial a_0}{\partial \phi}\right|_{\theta,\Omega} \frac{\partial \phi}{\partial \theta} \quad + \quad \left.\frac{\partial a_0}{\partial \theta}\right|_{\Omega,\phi} \quad + \quad \left.\frac{\partial a_0}{\partial \Omega}\right|_{\phi,\theta} \frac{\partial \Omega}{\partial \theta} = 0$$

and noting that

$$\left.\frac{\partial a_0}{\partial \phi}\right|_{\theta,\Omega} = -\hat{\imath}\, \left.\frac{\partial a_0}{\partial \Omega}\right|_{\phi,\theta}$$

to yield

$$\frac{a_2}{a_1} = \frac{\partial \Omega}{\partial \theta} - \hat{\imath}\frac{\partial \phi}{\partial \theta}.$$

The group velocity for the propagation of a discrete wave packet is therefore given by

$$c_g = \frac{\partial \Omega}{\partial \theta},$$

which is analogous to the analytic expression (A.4).

A.3 Semi-discrete Interpretation

Having determined general expressions for the amplification factor and phase and group velocities in the fully-discrete case, it is now desirable to examine the behavior of these quantities for semi-discrete schemes of the type being used in the present work [36]. With a semi-discrete approach, the spatial approximation is performed first and then the resulting o.d.e.'s are integrated in time. Applied to a scalar model problem, the scheme appears as

$$L_t u + \Delta t R(u) = 0, \tag{A.9}$$

where L_t is the Runge–Kutta operator, Δt is the time step and R is the spatial residual operator.

The solution is once again represented in terms of damped sinusoids, but following the semi-discrete paradigm, only the spatial variation is initially introduced so

that the semi-discrete solution has the form

$$u_i(t) = \widehat{u}(t)e^{\hat{\imath}(i\theta)},$$

where $\widehat{u}(t)$ is an unspecified time-varying Fourier amplitude. Substituting this expression into (A.9) produces an evolution equation for $\widehat{u}$

$$L_t\widehat{u} = \Delta t Z\widehat{u}, \tag{A.10}$$

where Z is the complex spatial Fourier operator

$$Z(\theta) = X(\theta) + \hat{\imath}Y(\theta),$$

and the Fourier footprint is the scalar quantity ΔtZ.

Temporal integration of (A.10) then produces

$$\widehat{u}^{n+1} = \psi(\Delta tZ)\widehat{u}^n \tag{A.11}$$

where ψ is also complex

$$\psi = U(X, Y, \Delta t) + \hat{\imath}V(X, Y, \Delta t).$$

Specifying the time varying amplitude to have the form

$$\widehat{u}(t) = u_i^0 e^{-n(\phi+\hat{\imath}\Omega)},$$

and substituting this expression into (A.11) then yields

$$e^{-(\phi+\hat{\imath}\Omega)} = \psi.$$

Taking the logarithm of both sides and equating the real and imaginary parts to zero as for (A.8) then produces the discrete damping and dispersion relationships corresponding to the spatial and temporal discretization operators R and L_t,

$$\begin{aligned} \phi &= -\ln|\psi|, \\ \Omega &= -\tan^{-1}\frac{V}{U}. \end{aligned}$$

The amplification factor is then given by

$$\chi = e^{-\phi} = |\psi| = (U^2 + V^2)^{1/2}, \tag{A.12}$$

the phase velocity by

$$c = \frac{\Omega}{\theta} = \frac{-\tan^{-1}\frac{V}{U}}{\theta}, \tag{A.13}$$

and the group velocity by

$$c_g = \frac{\partial \Omega}{\partial \theta} = \frac{V\frac{\partial U}{\partial \theta} - U\frac{\partial V}{\partial \theta}}{U^2 + V^2}. \tag{A.14}$$

One noticeable difference between the damping and propagative quantities is that the amplification factor $|\psi|$ does not depend explicitly on θ and is thus uniquely defined at each locus in the complex plane. As a result, contours of the amplification factor can be evaluated for a given temporal operator L_t independently from the choice of spatial operator R. The damping effectiveness for a particular R can then be determined by over-plotting the corresponding spatial Fourier footprint $\Delta t Z$. In this sense, the damping properties can be conveniently analyzed within the context of a semi-discrete approach where the spatial and temporal discretizations are treated separately. On the other hand, the phase and group velocities depend explicitly on θ and must be evaluated for the combined spatial and temporal discretization. Since group velocity contours for a given temporal operator are not uniquely defined in the complex plane, it is impossible to deduce detailed propagative information from the Fourier footprint. However, the next section will demonstrate that footprints can still be used to deduce some important propagative features of the combined discretization.

A.4 Evaluation for Runge–Kutta Schemes

Using order of magnitude arguments, the behavior of both the Runge–Kutta scheme and the Fourier footprint can be determined near the origin in the complex plane where the amplification factor is close to unity. For this purpose, consider a discretization of the convection–diffusion equation (A.1) with the diffusion coefficient taken to be

$$\nu = \Phi\frac{|a|\Delta x}{2}, \qquad \Phi \geq 1.$$

Choosing $\Phi = 1$ corresponds to upwinding of a convection equation while setting $\Phi > 1$ corresponds to more diffusive conditions typical of viscous flows. The spatial

residual operator then takes the form

$$R = \frac{a}{2\Delta x}\delta_{2x} - \Phi\frac{|a|}{2\Delta x}\delta_{xx}, \tag{A.15}$$

so that the real and imaginary parts of the Fourier footprint are

$$X(\theta) = -\Phi\frac{|a|}{\Delta x}(1-\cos\theta), \qquad Y(\theta) = -\frac{a}{\Delta x}\sin\theta,$$

and the time step is given by

$$\Delta t^{-1} = \frac{\Phi}{\sigma}\frac{|a|}{\Delta x},$$

where σ is the Courant number. For Runge–Kutta schemes of the class defined in Section B.4, the consistency requirements on the coefficients ensure that to first order, the real and imaginary components of ψ are approximated near the origin by

$$U \approx 1 + \Delta t X, \qquad V \approx \Delta t Y.$$

The portion of the Fourier footprint near the origin corresponds to small θ, so to first order

$$\Delta t X \approx 0, \qquad \Delta t Y \approx -\frac{\sigma}{\Phi}\theta,$$

and the discrete frequency is thus found to be

$$\Omega = -\tan^{-1}\frac{V}{U} \approx \frac{\sigma}{\Phi}\theta.$$

The phase and group velocities are therefore approximately equivalent

$$c \approx c_g \approx \frac{\sigma}{\Phi}, \tag{A.16}$$

indicating that the discretization is non-dispersive for modes that vary slowly in space. Moreover, for these low wave number modes, the group velocity is approximately equal to the amplitude of the sinusoid that defines the imaginary component in the Fourier footprint.

This information is useful because it provides a means of evaluating the group velocity for those modes that are least efficiently damped simply by visual inspection of the Fourier footprint. For example, Fig. A.2 depicts three different Fourier footprints corresponding to various choices of Φ and σ for the spatial operator defined

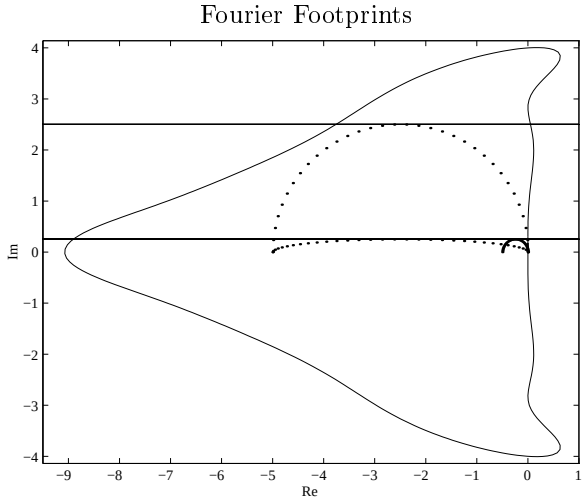


Figure A.2 Visual determination of the group velocity for errors with low wave number.

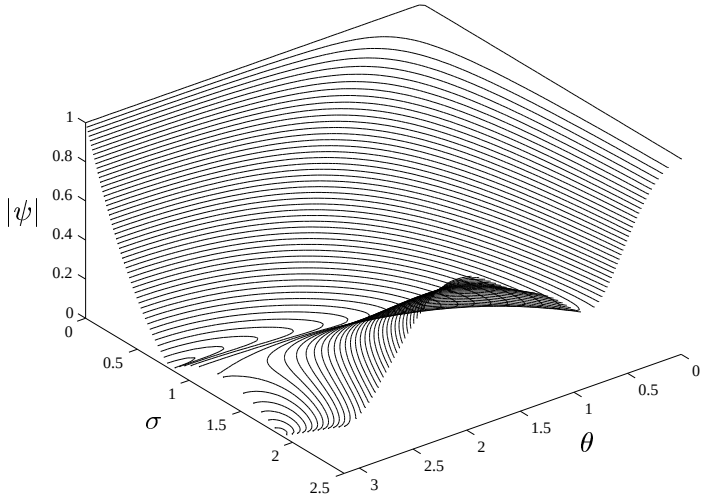
by (A.15). The stability region for the 5-stage Runge–Kutta scheme [45] used in the present work is also included to provide a frame of reference. Without actually knowing the analytic dependence of Ω on θ for these combinations of R and L_t, it is now possible to observe that the low wave number modes from the footprint forming a large circular arc will be propagated quite well ($c_g \approx 2.5$) while those from the other two footprints will be propagated equally poorly ($c_g \approx 0.25$).

To obtain detailed propagative information for every mode represented in the Fourier footprint, it is necessary to resort to numerical evaluation of the group velocity for specific choices of R and L_t. As a relevant example, consider the spatial operator (A.15) with $\Phi = 1$ (corresponding to upwinding of a convection equation) and the same 5-stage Runge–Kutta time integration scheme. After much tedious manipulation, it is possible to obtain analytic expressions for U, V, $\frac{\partial U}{\partial \theta}$ and $\frac{\partial V}{\partial \theta}$ and hence to form expressions for ψ, c and c_g using equations (A.12) to (A.14). Sampling these expressions numerically on the intervals $(0 \leq \theta \leq \pi)$ and $(0 \leq \sigma \leq 2.5)$ yields the contours shown in Fig. A.3. In the complex plane, this sampling corresponds to the Fourier footprint shown in Fig. A.4*, where for example, the arched line corresponds to $\sigma = 2.5$ for the range of θ and the ray from the origin to the center

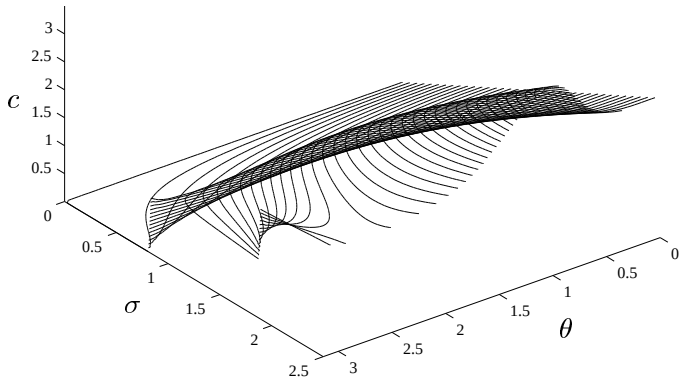
*For visual simplicity, the convention is adopted of plotting Fourier footprints with a reflected imaginary component so that residual eigenvalues corresponding to a positive phase velocity fall in the upper half-plane.

of the arc corresponds to $\theta = \frac{\pi}{2}$ for the range of σ.

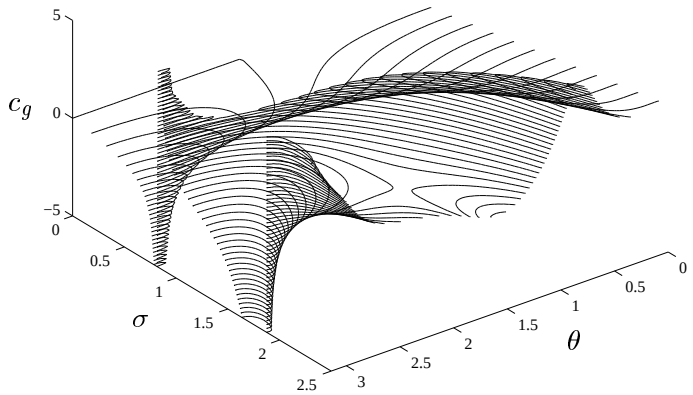
The amplification factor contours in Fig. A.3a provide an alternative viewpoint to the more familiar complex plane representation of Fig. 2.1. For the reasons explained in Section A.3, the phase and group velocities cannot be uniquely defined in the complex plane, so they must instead be evaluated in the (θ, σ) coordinate system. The phase velocity contours in Fig. A.3b exhibit the expected linear behavior of equation (A.16) for small values of θ and vary smoothly elsewhere except at $\theta = \pi$, where there are step discontinuities as U switches sign for $V = 0$. While the phase velocity is non-negative everywhere, the group velocity produced by the discrete method does become negative for certain modes, as shown in Fig. A.3c. As expected, the group velocity is identical to the phase velocity for small values of θ. This can be more clearly seen in Fig. A.5a, where slices of all three contour plots are shown for $\theta = \frac{\pi}{16}$. It is also interesting to slice the contour plots in the other direction as displayed in Fig. A.5b for $\sigma = 2.5$. Here it is evident that disturbances with wave number $\theta \approx \frac{\pi}{4}$ will actually propagate with a group velocity significantly larger than the Courant number, while errors with $\theta \approx \frac{\pi}{2}$ will propagate very slowly in the direction opposite to the physical convection velocity. Fortunately, as seen from Fig. A.3c, this phenomenon occurs only for modes which have wave numbers significantly larger than zero, and can therefore be damped effectively.



A.3a Amplification factor contours.



A.3b Phase velocity contours.



A.3c Group velocity contours.

Figure A.3 Contours describing the damping and propagation properties of the combined space-time discretization.

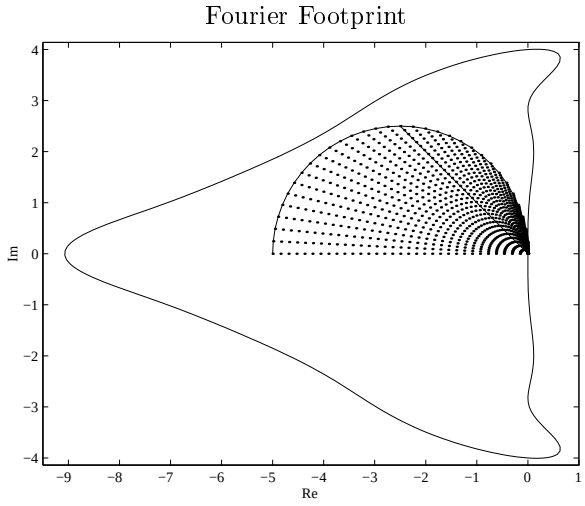
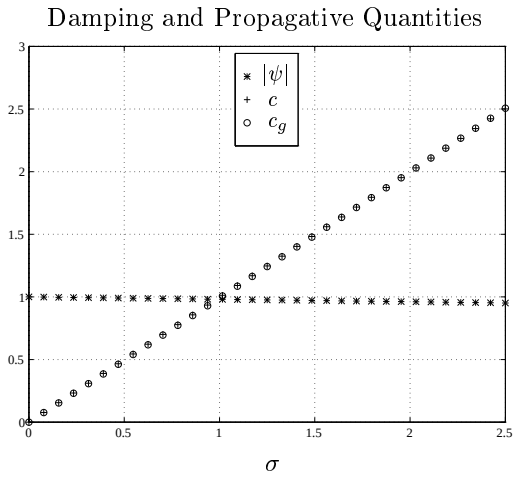
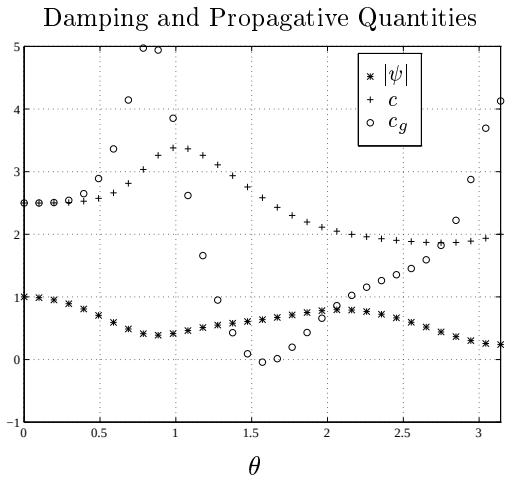


Figure A.4 Fourier footprint for which contours of damping and propagative quantities are shown in Fig. A.3.



A.5a Properties for $\theta = \frac{\pi}{16}$.



A.5b Properties for $\sigma = 2.5$.

Figure A.5 Variation in amplification factor, phase velocity and group velocity.

Appendix B

Discretization

This appendix provides a brief but complete description of the two-dimensional discretization implemented for the present work. Three-dimensional calculations were performed using a modified version of the flow solver FLO107 written by Jameson and Martinelli [34]. In most cases, the three-dimensional discretization is a straightforward extension of the two-dimensional case and details are not provided here. The principal exception is the evaluation of the viscous fluxes, for which separate expositions are provided in Section B.2.3.

B.1 Governing Equations

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

$$\begin{aligned} x^* &= \tfrac{x}{L}, & y^* &= \tfrac{y}{L}, & p^* &= \tfrac{p}{p_\infty}, & \rho^* &= \tfrac{\rho}{\rho_\infty}, \\ t^* &= \tfrac{t\sqrt{p_\infty/\rho_\infty}}{L}, & u^* &= \tfrac{u}{\sqrt{p_\infty/\rho_\infty}}, & v^* &= \tfrac{v}{\sqrt{p_\infty/\rho_\infty}}, & E^* &= \tfrac{E}{p_\infty/\rho_\infty}, \\ \mu^* &= \tfrac{\sqrt{\gamma}M_\infty}{Re}\tfrac{\mu}{\mu_\infty}, & k^* &= \tfrac{\gamma\mu^*}{Pr}, & T^* &= \tfrac{p^*}{(\gamma-1)\rho^*}. \end{aligned} \tag{B.1}$$

Omitting the asterisks for clarity, the two-dimensional Navier-Stokes equations then take the form

$$\frac{\partial W}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} \quad \text{in } \Omega, \tag{B.2}$$

where the state vector W and inviscid flux vectors F_x and F_y are given by

$$W = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix}, \quad F_x = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{Bmatrix}, \quad F_y = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{Bmatrix}, \tag{B.3}$$

and the viscous flux vectors G_x and G_y are described by

$$G_x = \begin{Bmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ u\sigma_{xx} + v\sigma_{xy} - q_x \end{Bmatrix}, \quad G_y = \begin{Bmatrix} 0 \\ \sigma_{yx} \\ \sigma_{yy} \\ u\sigma_{yx} + v\sigma_{yy} - q_y \end{Bmatrix}. \tag{B.4}$$

The Euler equations are obtained by neglecting the viscous fluxes on the right hand side of (B.2).

In these definitions, ρ is the density, (u,v) are the Cartesian velocity components and E is the total energy. The pressure is determined by the equation of state,

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

where γ is the ratio of the specific heats ($\gamma = 1.4$ for air), and the stagnation enthalpy is given by

$$H = E + \frac{p}{\rho}.$$

The viscous stresses may be written as

$$\begin{aligned} \sigma_{xx} &= 2\mu\frac{\partial u}{\partial x} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \\ \sigma_{yy} &= 2\mu\frac{\partial v}{\partial y} + \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \\ \sigma_{xy} &= \sigma_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \end{aligned}$$

where the molecular viscosity μ is modeled by Sutherland's law (s.t.p. conditions),

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

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

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

The heat fluxes are given by Fourier's law

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

with the coefficient of thermal conductivity and the temperature defined by

$$k = \frac{\gamma\mu}{Pr}, \qquad T = \frac{p}{(\gamma-1)\rho}, \tag{B.5}$$

where Pr is the Prandtl number ($Pr = 0.72$ for air).

For turbulent calculations, the Reynolds averaged Navier–Stokes equations [65] are solved using a turbulence model for closure. Following from the Boussinesq hypothesis [9], 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

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

and the thermal conductivity is given by

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

where μ_t is the turbulent eddy viscosity and Pr_t is the turbulent Prandtl number ($Pr_t = 0.9$ for air). The calculation of the turbulent eddy viscosity using a turbulence model is described in Appendix E.

To obtain a well-posed problem, appropriate boundary conditions must be imposed on the domain boundary $\partial\Omega$. 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 zero 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.

B.2 Semi-Discrete Finite Volume Scheme

The finite volume method is based on an integral formulation of the governing conservation laws

$$\frac{d}{dt}\int_\Omega W\,dxdy + \int_{\partial\Omega}(F_x - G_x)dy - (F_y - G_y)dx.$$

The domain Ω is subdivided into a number of discrete mesh cells which are treated as individual control volumes over which the integrals are evaluated. Using a structured mesh, it is convenient to adopt a mesh-aligned (ξ,η) coordinate system defined by the metrics

$$K = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}, \qquad J = \det(K), \qquad K^{-1} = \frac{1}{J}\begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix}, \tag{B.6}$$

where the Jacobian of the mapping J represents the local cell volume. The projections of the cell faces in the Cartesian coordinate directions can then be expressed as

$$\begin{bmatrix} S_{\xi x} & S_{\xi y} \\ S_{\eta x} & S_{\eta y} \end{bmatrix} = JK^{-1}, \tag{B.7}$$

where, for example, $S_{\xi x}$ represents the projection of the ξ face in the x direction. In computational coordinates, the integral form of the Navier–Stokes equations then becomes

$$\frac{d}{dt}\int_\Omega J\,W\,d\xi d\eta + \int_{\partial\Omega}(F_\xi - G_\xi)d\eta - (F_\eta - G_\eta)d\xi,$$

with inviscid and viscous fluxes defined with respect to the computational cell faces by

$$F_\xi = S_{\xi x}F_x + S_{\xi y}F_y, \qquad G_\xi = S_{\xi x}G_x + S_{\xi y}G_y,$$
$$F_\eta = S_{\eta x}F_x + S_{\eta y}F_y, \qquad G_\eta = S_{\eta x}G_x + S_{\eta y}G_y.$$

Following the semi-discrete approach of Jameson *et al.* [36], the fluxes are evaluated first to produce a coupled system of o.d.e.'s,

$$\frac{d}{dt}(W_{i,j}) + R_{i,j} = 0,$$

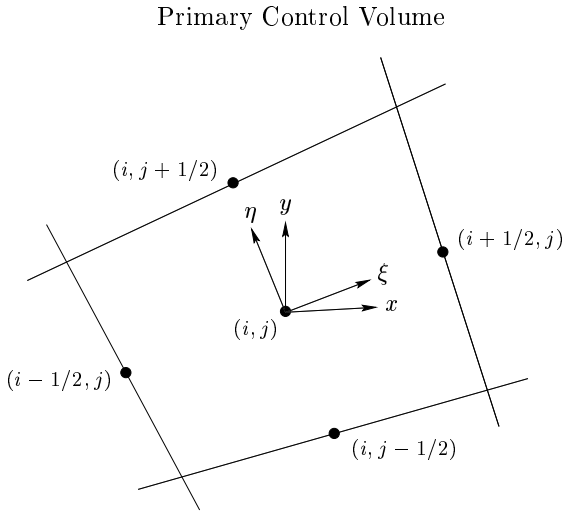


Figure B.1 Finite volume mesh cell for evaluation of the inviscid, viscous and numerical dissipation flux balances.

which can then be integrated in time using a multi-stage Runge–Kutta scheme. Here, (i,j) are the cell-centered mesh indices in the (ξ,η) directions as shown in Fig. B.1. The residual vector

$$R_{i,j} = \frac{1}{J_{i,j}} \left(E_{i,j} - V_{i,j} - N_{i,j} \right)$$

represents the contributions E and V of the physical inviscid and viscous fluxes as well as the numerical dissipation fluxes N. Numerical dissipation is introduced to suppress spurious oscillations and to ensure numerical stability. The scheme is constructed so that the numerical dissipation preserves second order accuracy in smooth regions of the flow while allowing sharp non-oscillatory resolution of discontinuities.

B.2.1 Inviscid Fluxes

For a cell-centered scheme, the inviscid flux through a cell face is evaluated by averaging the fluxes computed at the cell centers on either side (e.g. $F^{\xi}_{i+1/2,j} = \frac{1}{2}[F^{\xi}_{i,j} + F^{\xi}_{i+1,j}]$), where it is understood that all the metric terms in this expression are evaluated at the face. The inviscid contribution to the residual vector then takes the form

$$E_{i,j} = F^{\xi}_{i+1/2,j} - F^{\xi}_{i-1/2,j} + F^{\eta}_{i,j+1/2} - F^{\eta}_{i,j-1/2}. \tag{B.8}$$

On a regular mesh, this spatial discretization corresponds to central differencing so that there is no mechanism for eliminating spurious odd/even modes that oscillate between positive and negative values in alternate cells. It is therefore imperative to introduce some form of numerical dissipation to ensure that these spurious oscillations are suppressed.

B.2.2 Numerical Dissipation Fluxes

The numerical dissipation fluxes are evaluated in terms of characteristic variables and are introduced independently in each mesh direction. For this purpose, the Roe matrix A that identically satisfies $\Delta F = A\Delta W$ [64] is constructed at the cell face and decomposed into right and left eigenvector matrices T and T^{-1} and a diagonal eigenvalue matrix Λ

$$A = T\Lambda T^{-1}.$$

These matrices are identical to the standard flux Jacobians described in Appendix C if they are evaluated in terms of Roe-averaged variables. The Roe-averaged velocity components and stagnation enthalpy are determined at the cell faces using expressions of the form

$$\bar{u}_{i+1/2,j} = \frac{\sqrt{\rho_{i+1,j}}u_{i+1,j} + \sqrt{\rho_{i,j}}u_{i,j}}{\sqrt{\rho_{i+1,j}} + \sqrt{\rho_{i,j}}},$$

and the Roe-averaged speed of sound is evaluated in terms of these quantities using

$$\bar{c}^2 = (\gamma - 1)\left(\bar{H} - \frac{\bar{u}^2 + \bar{v}^2}{2}\right).$$

Defining the characteristic difference $\Delta\Omega$ across the cell face,

$$\Delta\Omega = T^{-1}\Delta W,$$

the dissipative flux corresponding to first order upwinding of each characteristic field takes the form

$$F^{\xi}_{N_{i+1/2,j}} = \frac{1}{2}(T|\Lambda|\Delta\Omega)_{i+1/2,j} = \frac{1}{2}\sum_{k=1,4}(|\lambda_k|T_k\Delta\Omega_k)_{i+1/2,j},$$

where the second expression reveals that the dissipation for the k-th characteristic field is scaled by the modulus of the corresponding eigenvalue $|\lambda_k|$. This type

of dissipation is termed matrix dissipation because it requires matrix operations (multiplication by T and T^{-1}) to introduce the dissipation independently to each characteristic field.

As a brief aside, note that a less expensive scalar dissipation may be constructed by scaling the dissipation for all characteristic fields by the maximum eigenvalue

$$F^{\xi}_{N_{i+1/2,j}} = \frac{1}{2}[\rho(A)\Delta W]_{i+1/2,j}.$$

This scheme is not employed for the present work because it reduces the solution quality by introducing excess numerical dissipation into all but one of the characteristic fields.

The matrix scheme described above does not naturally incorporate an entropy condition, so to ensure that non-physical expansion shocks are not allowed to persist at the sonic line, an entropy fix must be applied to the two acoustic eigenvalues to prevent the numerical dissipation from vanishing in the corresponding characteristic fields. The fix is also applied to the two convective eigenvalues to prevent the dissipation in the convective characteristic fields from vanishing near stagnation points and near the wall in viscous calculations.

In both roles, the fix smoothly bounds the absolute value of the characteristic speed away from zero whenever it falls below a specified threshold ϵ using the construction

$$|\lambda|_{\text{fix}} = \begin{cases} |\lambda|, & |\lambda| \geq \epsilon, \\ \frac{1}{2}\left(\frac{\lambda^2}{\epsilon} + \epsilon\right), & |\lambda| < \epsilon. \end{cases} \tag{B.9}$$

Following van Leer *et al.* [76], the threshold parameter is based on the gradient across the cell face so that the fix is active whenever the average value at the face becomes small compared to the gradient across the face. For the acoustic families, using an expanded definition for the gradient

$$\epsilon_{i+1/2,j} \equiv 2(\lambda_{i+1,j} - \lambda_{i,j}), \tag{B.10}$$

ensures that the fix modifies the eigenvalues at two cell faces on either side of an expansion and will remain inactive in a shock. For the convective families, this

expression should be modified by taking the absolute value of the gradient across the face so that the fix will activate in both accelerating and decelerating stagnation points. This trigger based on gradient vs. average has the desirable property that the fix remains inactive in regions of the flow where the speeds are uniformly small but do not pass through zero, thus avoiding contaminating the boundary layer with additional numerical dissipation.

This first order scheme admits monotone one-point shocks but fails to provide sufficient accuracy in smooth regions of the flow. As a suitable higher order alternative, the 2nd/4th difference switch of Jameson *et al.* [36] can be applied to the characteristic variables formed using the Roe linearization. The numerical dissipation fluxes then take the form

$$\begin{aligned} F^{\xi}_{N_{i+1/2,j}} &= \varepsilon^{(2)} T_{i+1/2,j} |\Lambda|_{i+1/2,j} \Delta\Omega_{i+1/2,j} \\ &- \varepsilon^{(4)} T_{i+1/2,j} |\Lambda|_{i+1/2,j} (\Delta\Omega_{i+3/2,j} - 2\Delta\Omega_{i+1/2,j} + \Delta\Omega_{i-1/2,j}) \end{aligned}$$

where the dissipative coefficients are determined using a sensor based on pressure curvature [36, 55]. Near a discontinuity, $\varepsilon^{(2)} = \frac{1}{2}$ and $\varepsilon^{(4)} = 0$ to yield first order upwinding and a monotone one-point shock structure as before. In smooth regions, $\varepsilon^{(2)} = O(\Delta x^2)$ and $\varepsilon^{(4)} = O(1)$ so that the accuracy of the dissipative flux balance is $O(\Delta x^3)$, and the discretization will retain the second order accuracy of the central differences arising from the inviscid flux calculations. The numerical dissipation flux balance has the same appearance as the inviscid flux balance (B.8) with F^{ξ}_N and F^{η}_N substituted for F^{ξ} and F^{η}.

B.2.3 Viscous Fluxes

Two Dimensions

To evaluate the viscous fluxes, both the flow variables and their gradients are required at the faces. The gradients are found at the midpoint of a cell face by applying Gauss' theorem to an auxiliary control volume of the type shown in Fig. B.2, formed by joining the centers of two adjacent cells with the end points of their dividing side

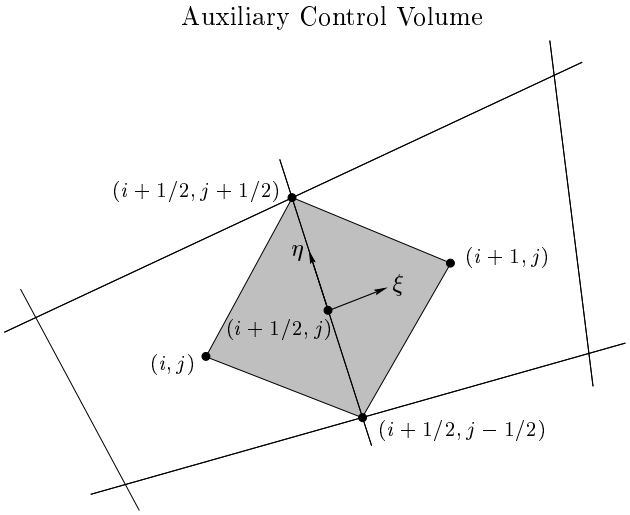


Figure B.2 Auxiliary control volume for calculating velocity and temperature gradients at the cell face.

[45]. This leads to expressions for the gradients of the form

$$\begin{aligned}\frac{\partial u}{\partial x} &= \frac{1}{2J_{\text{aux}}}\left(\frac{\partial u}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial u}{\partial \eta}\frac{\partial y}{\partial \xi}\right), \\ \frac{\partial u}{\partial y} &= -\frac{1}{2J_{\text{aux}}}\left(\frac{\partial u}{\partial \xi}\frac{\partial x}{\partial \eta} - \frac{\partial u}{\partial \eta}\frac{\partial x}{\partial \xi}\right),\end{aligned}$$

where J_{aux} is the volume of the auxiliary cell and differencing with respect to ξ and η corresponds to differencing across the diagonals of the auxiliary control volume. Averaging all other necessary flow variables to the cell face, the viscous fluxes are then assembled directly at the face midpoint and the flux balance is performed in an analogous manner to (B.8). This discretization of the viscous terms has the desirable property that it does not admit odd/even modes that oscillate between positive and negative values at alternate cells. Unfortunately, the auxiliary control volumes centered at the cell faces are difficult to generalize to three dimensions, so some alternative formulation is generally adopted.

Three Dimensions

The baseline version of the viscous flux discretization in FLO107 [34] evaluates the velocity gradients and forms the stress components at the vertices before averaging to the centers of the cell faces to compute the viscous flux balance [42]. This

discretization is very efficient because it avoids having to compute the velocity gradients separately for each cell face (each cell being associated with three faces but only one vertex). However, as a result of this averaging process, the discretization does admit odd/even spurious modes that oscillate between positive and negative values at alternate cells. This was previously thought not to be a concern since the numerical dissipation is designed to eliminate this type of oscillation in the solution.

One of the surprises during the course of the present work was the discovery that the admission of these odd/even modes actually can present a significant impediment to convergence. Using the standard scalar preconditioner, this discretization was found to necessitate a smaller Courant number in comparison to the modified scheme described below, and was actually unstable when used in conjunction with the block-Jacobi preconditioner. This problem also arose for the viscous fluxes in the Spalart–Allmaras turbulence model, which failed to converge when using this discretization.

To overcome this difficulty, one option is to accept the additional cost of calculating the velocity gradients at the cell faces in order to return to a compact viscous discretization stencil that does not admit oscillatory modes. An alternative approach, which was proposed by Jameson and Caughey in developing a discretization for the transonic potential equation [33] and later suggested for use with the Navier–Stokes equations by Liu and Zheng [43], is to add a correction stencil to the velocity gradients calculated at the vertices in order to approximately convert the large stencil of the present scheme to the smaller stencil of a face-centered approach.

Consider a regular Cartesian grid with unit mesh spacing and mesh indices (i, j, k) located at the cell centers in the (x, y, z) coordinate directions. Using a compact face-centered scheme to compute a velocity gradient at the face $(i+1/2, j, k)$ produces

$$\left.\frac{\partial u}{\partial x}\right|_{\text{compact}} = \Delta_{i+1/2} u_{j,k},$$

where $\Delta_{i+1/2}$ denotes differencing across the cell face in the i direction. By contrast, computing the gradients at the vertices and averaging to the cell face gives

$$\left.\frac{\partial u}{\partial x}\right|_{\text{averaged}} = \Delta_{i+1/2} M,$$

where M is the stencil

$$M = \begin{bmatrix} \frac{1}{16}u_{j+1,k-1} & \frac{1}{8}u_{j+1,k} & \frac{1}{16}u_{j+1,k+1} \\ \frac{1}{8}u_{j,k-1} & \frac{1}{4}u_{j,k} & \frac{1}{8}u_{j,k+1} \\ \frac{1}{16}u_{j-1,k-1} & \frac{1}{8}u_{j-1,k} & \frac{1}{16}u_{j-1,k+1} \end{bmatrix}.$$

Comparing these expressions reveals that in the case of a regular grid, the compact stencil can be exactly recovered by subtracting a correction stencil from the averaged formula

$$\left.\frac{\partial u}{\partial x}\right|_{\text{compact}} = \left.\frac{\partial u}{\partial x}\right|_{\text{averaged}} - \Delta_{i+1/2}N,$$

where N is given by

$$N = \begin{bmatrix} \frac{1}{16}u_{j+1,k-1} & \frac{1}{8}u_{j+1,k} & \frac{1}{16}u_{j+1,k+1} \\ \frac{1}{8}u_{j,k-1} & -\frac{3}{4}u_{j,k} & \frac{1}{8}u_{j,k+1} \\ \frac{1}{16}u_{j-1,k-1} & \frac{1}{8}u_{j-1,k} & \frac{1}{16}u_{j-1,k+1} \end{bmatrix}.$$

Generalizing this formula to mesh-aligned computational coordinates (ξ, η, ζ) then produces the approximate relationship for the gradient at a ξ face

$$\left.\frac{\partial u}{\partial x}\right|_{\text{compact}} \approx \left.\frac{\partial u}{\partial x}\right|_{\text{averaged}} - \frac{S_{\xi x}}{J_{i+1/2}}\Delta_{i+1/2}N,$$

where $S_{\xi x}$ represents the projection of the ξ face in the x coordinate direction and $J_{i+1/2}$ is the average volume of the cells on either side of the face. Similar expressions can be obtained for the gradients of the other relevant flow variables for each of the (x, y, z) coordinate directions at each of the (ξ, η, ζ) cell faces. Note that for a given face, the stencil N need only be calculated once per flow variable and then the metric term is varied for each Cartesian direction. The same correction stencil is applied to the calculation of the viscous fluxes for the Spalart–Allmaras turbulence model.

Using this approach, the velocity gradients are computed at the vertices as in the original version of FLO107 and then averaged to the cell faces, where the correction stencil is applied before assembling the stress components. This method is relatively inexpensive compared to computing the velocity gradients directly at the cell faces

and provides a good approximation to the compact stencil that eliminates odd/even grid point decoupling. In fact, running side by side with the two-dimensional flow solver on a straight wing with an aspect ratio of 10^6 (so as to eliminate the influence of the third coordinate direction), the two codes produced identical convergence histories to machine accuracy.

B.3 Boundary Conditions

For airfoil calculations, both the wall and far field boundary conditions are enforced by assigning values to dummy cells outside the computational domain. For Euler flows, zero mass flux through a solid wall boundary is ensured by reflecting the normal velocity component in the dummy cell and the pressure is extrapolated using an approximation based on the wall curvature.

For Navier–Stokes flows, zero mass flux and no slip conditions are enforced by reflecting both velocity components, and all other flow quantities are set equal to those in the cell adjacent to the wall to produce an adiabatic boundary with zero normal pressure gradient.

At the far field, non-reflecting boundary conditions are employed for both Euler and Navier–Stokes calculations. These are introduced using Riemann invariants based on the approximation of one-dimensional flow normal to the boundary. Assuming the far field is subsonic, the outgoing and incoming Riemann invariants at the boundary are defined by

$$\begin{aligned} R_{\text{out}} &= q_{n_e} + \frac{2c_e}{\gamma - 1}, \\ R_{\text{in}} &= q_{n_{ff}} - \frac{2c_{ff}}{\gamma - 1}, \end{aligned}$$

where q_n is the normal velocity and c is the speed of sound. The outgoing invariant is formed by extrapolating quantities from the computational cell adjacent to the boundary. For the incoming invariant, the normal velocity is based on the free stream velocity and a far field vortex correction derived from linear potential thin airfoil theory [4]. The vortex model includes the Prandtl-Glauert compressibility

correction for validity in the transonic regime [26]. The speed of sound for the incoming invariant is found from the corrected far field velocities and an isenthalpic condition.

These invariants are then added and subtracted to produce the values to be specified in the far field dummy cells

$$\begin{aligned} q_n &= \frac{1}{2}\left(R_{\text{out}} + R_{\text{in}}\right), \\ c &= \frac{\gamma - 1}{4}\left(R_{\text{out}} - R_{\text{in}}\right). \end{aligned}$$

At an outflow boundary ($q_n > 0$), the entropy and tangential velocity are extrapolated from the last computational cell to fully determine the state vector in the far field. At an inflow boundary, the tangential velocity is based on the far field model and the entropy is set to the free stream value. The only deviation from this approach is in a viscous wake region, where the static pressure is set to the free stream value and all other quantities are extrapolated from the last computational cell.

B.4 Multi-Stage Runge–Kutta Time-Stepping Scheme

After evaluating the spatial residual vector, the system is integrated in time using a multi-stage Runge–Kutta scheme [36],

$$L_t W_{i,j} + P_{i,j} R_{i,j} = 0,$$

where L_t is the Runge–Kutta operator and P is a local preconditioner that acts as a time step. To improve the damping properties of the Runge–Kutta operator and to provide large stability limits along both the real and imaginary axes, the residual is divided into convective and diffusive contributions

$$\begin{aligned} R_{i,j} &= C_{i,j} - D_{i,j}, \\ C_{i,j} &= \frac{1}{J_{i,j}} E_{i,j}, \\ D_{i,j} &= \frac{1}{J_{i,j}} (V_{i,j} + N_{i,j}), \end{aligned}$$

that are treated separately during the integration procedure. For an M-stage scheme, the Runge–Kutta operator L_t is then defined by

$$\begin{aligned} W^{(0)} &= W^n \\ &\vdots \\ W^{(k)} &= W^n - \alpha_k P R^{(k-1)} \\ &\vdots \\ W^{n+1} &= W^{(M)} \end{aligned} \tag{B.11}$$

with

$$\begin{aligned} R^{(k)} &= C(W^{(k)}) - D^{(k)}, \\ D^{(k)} &= \beta_{k+1} D(W^{(k)}) + (1 - \beta_{k+1}) D^{(k-1)}, \end{aligned}$$

and the consistency requirements $\alpha_M = 1$ and $\beta_1 = 1$ [28]. The beneficial properties of this construction are that only one previous solution level need be stored and that the diffusive terms, which are the most expensive to compute, need not be calculated at every stage. The present work employs a 5-stage scheme due to Martinelli [45] that requires three evaluations of the diffusive terms and has the coefficients

$$\alpha_1 = \tfrac{1}{4}, \quad \alpha_2 = \tfrac{1}{6}, \quad \alpha_3 = \tfrac{3}{8}, \quad \alpha_4 = \tfrac{1}{2}, \quad \alpha_5 = 1,$$

$$\beta_1 = 1, \quad \beta_2 = 0, \quad \beta_3 = \tfrac{14}{25}, \quad \beta_4 = 0, \quad \beta_5 = \tfrac{11}{25},$$

which produce the stability region and amplification factor contours previously displayed in Fig. 2.1. With this scheme, all calculations are performed using a Courant number of 2.5 using either scalar or matrix preconditioning in either two or three dimensions.

B.5 Parallelization

To take advantage of modern computer architectures, the three-dimensional code was parallelized using the MPI (Message Passing Interface) library to implement a domain decomposition approach using a SPMD (Single Program Multiple Data) paradigm [3]. Halo data is exchanged between neighboring processors after every Runge–Kutta stage on all meshes so that the convergence of the parallel code is identical to that of the serial version.

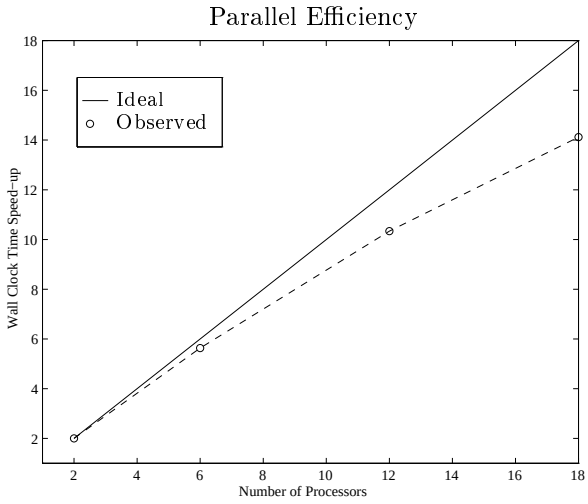


Figure B.3 Parallel efficiency for a 288×64×16 mesh.

The parallel scalability of the approach is demonstrated in Fig. B.3 for a turbulent Navier–Stokes calculation on a 288×64×16 mesh with 294,912 cells. The calculation scales to 18 processors with an efficiency of 78% relative to a calculation with 2 processors. The calculation could not be performed on one processor due to memory constraints. For viscous meshes with an order of magnitude more cells, the method will continue to scale efficiently to far larger numbers of processors.

Appendix C

Preconditioners

This appendix provides a thorough documentation of the form of the preconditioners in both two and three dimensions to serve as a reference for future code development.

C.1 Two Dimensions

The expressions for the scalar and matrix preconditioners follow from the linearized Navier–Stokes equations in (ξ,η) mesh-aligned coordinates. Starting from conservation form in Cartesian coordinates, the system appears as

$$\frac{dW}{dt} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}, \tag{C.1}$$

where F_x and F_y are the inviscid flux vectors and G_x and G_y are the viscous flux vectors. The viscous fluxes contain terms of the form

$$\frac{\partial}{\partial x}\left(\mu u \frac{\partial u}{\partial x}\right),$$

which can be linearized about a constant state $(\bar{u}, \bar{\mu})$ to produce

$$\bar{\mu}\bar{u}\frac{\partial^2 u}{\partial x^2},$$

where fluctuations in viscosity have been neglected and u now represents the local velocity perturbation. Regrouping terms with like derivatives gives

$$\frac{dW}{dt} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = \frac{\partial^2 G_{xx}}{\partial x^2} + \frac{\partial^2 G_{yy}}{\partial y^2} + \frac{\partial^2 G_{xy}}{\partial x \partial y},$$

so that upon transforming to mesh-aligned computational coordinates, the system becomes

$$\frac{dW}{dt} + \frac{1}{J}\left(\frac{\partial F_\xi}{\partial \xi} + \frac{\partial F_\eta}{\partial \eta}\right) = \frac{1}{J^2}\left(\frac{\partial^2 G_{\xi\xi}}{\partial \xi^2} + \frac{\partial^2 G_{\eta\eta}}{\partial \eta^2} + \frac{\partial^2 G_{\xi\eta}}{\partial \xi \partial \eta}\right).$$

Here, the inviscid fluxes are of the form

$$F_\xi = S_{\xi x} F_x + S_{\xi y} F_y,$$

and the viscous fluxes appear as

$$\begin{aligned} G_{\xi\xi} &= S_{\xi x}^2 G_{xx} + S_{\xi y}^2 G_{yy}, \\ G_{\xi\eta} &= 2 S_{\xi x} S_{\eta x} G_{xx} + 2 S_{\xi y} S_{\eta y} G_{yy} + (S_{\xi x} S_{\eta y} + S_{\xi y} S_{\eta x}) G_{xy}. \end{aligned}$$

The metric terms are defined by (B.7), where, as before, J represents the cell volume and $S_{\xi x}$ represents the projection of the ξ face in the x direction.

Introduction of the flux Jacobians then leads to the linearized form of the two-dimensional Navier–Stokes equations in computational coordinates

$$\frac{dW}{dt} + \frac{1}{J}\left(A_\xi \frac{\partial W}{\partial \xi} + A_\eta \frac{\partial W}{\partial \eta}\right) = \frac{1}{J^2}\left(B_{\xi\xi} \frac{\partial^2 W}{\partial \xi^2} + B_{\eta\eta} \frac{\partial^2 W}{\partial \eta^2} + B_{\xi\eta} \frac{\partial^2 W}{\partial \xi \partial \eta}\right).$$

The derivation of the Jacobians is simplified by first differentiating the fluxes with respect to a non-conservative vector of flow variables $\widetilde{W} = \{\rho, u, v, w, p\}^T$ and then transforming to conservative variables using the transformation matrix $M = \frac{\partial W}{\partial \widetilde{W}}$

$$\begin{aligned} A_\xi &= \frac{\partial F_\xi}{\partial \widetilde{W}} M^{-1}, \\ B_{\xi\xi} &= \frac{\partial G_{\xi\xi}}{\partial \widetilde{W}} M^{-1}. \end{aligned}$$

The spatial residual operator based on this linear mesh-aligned representation then has the form

$$\begin{aligned} R &= \frac{1}{J}\left(\frac{A_\xi}{2\Delta\xi}\delta_{2\xi} - \frac{|A_\xi|}{2\Delta\xi}\delta_{\xi\xi} + \frac{A_\eta}{2\Delta\eta}\delta_{2\eta} - \frac{|A_\eta|}{2\Delta\eta}\delta_{\eta\eta}\right) \\ &- \frac{1}{J^2}\left(\frac{B_{\xi\xi}}{\Delta\xi^2}\delta_{\xi\xi} + \frac{B_{\eta\eta}}{\Delta\eta^2}\delta_{\eta\eta} + \frac{B_{\xi\eta}}{4\Delta\xi\Delta\eta}\delta_{2\xi 2\eta}\right), \end{aligned} \tag{C.2}$$

where first order upwinding of the inviscid terms is accomplished by a Roe linearization [64].

C.1.1 Scalar Preconditioner

The scalar preconditioner employed for Navier–Stokes calculations is a simplified version of the standard local time step described by equation (3.2) on p. 34

$$P_S^{-1} = \frac{1}{\sigma}\left[\frac{1}{J}\left(\frac{\rho(A_\xi)}{\Delta\xi} + \frac{\rho(A_\eta)}{\Delta\eta}\right) + \frac{2}{J^2}\left(\frac{\rho(B_{\xi\xi})}{\Delta\xi^2} + \frac{\rho(B_{\eta\eta})}{\Delta\eta^2}\right)\right]. \tag{C.3}$$

This simplified form is obtained by neglecting the viscous cross-diffusion term and by noting that $\sigma_p \approx 2\sigma_h$ for the 5-stage Runge–Kutta scheme employed in the present work (see Fig. 2.1). Alternatively, the present simplified form may be obtained directly from the block-Jacobi matrix preconditioner (3.3) by replacing its constituent matrices by their spectral radii. For Euler calculations, only the inviscid terms are included in the preconditioner. Note that by definition, all of the metric differences are equal to unity.

Defining the flow speed q,

$$q^2 = u^2 + v^2,$$

the speed of sound c,

$$c^2 = (\gamma - 1)\left(H - \frac{q^2}{2}\right),$$

the flux rate through the ξ face Q_ξ,

$$Q_\xi = S_{\xi x}u + S_{\xi y}v, \tag{C.4}$$

and the area of the ξ face S_ξ,

$$S_\xi^2 = S_{\xi x}^2 + S_{\xi y}^2, \tag{C.5}$$

the spectral radii of the inviscid flux Jacobians may then be expressed as

$$\rho(A_\xi) = |Q_\xi| + cS_\xi, \qquad \rho(A_\eta) = |Q_\eta| + cS_\eta,$$

and spectral radii of the viscous flux Jacobians take the form

$$\rho(B_{\xi\xi}) = \frac{\gamma\mu}{Pr\rho}S_\xi^2, \qquad \rho(B_{\eta\eta}) = \frac{\gamma\mu}{Pr\rho}S_\eta^2.$$

The spectral radii are evaluated at the cell faces and then averaged to the cell centers to form the preconditioner before the first stage of each time step. Note that one of the factors of J^{-1} will cancel out of both the fluxes and the preconditioner and may therefore be left out of the discretization.

C.1.2 Matrix Preconditioner

The block-Jacobi matrix preconditioner is formed by extracting the coefficients of the residual operator (C.2) that correspond to the central node

$$P^{-1} = \frac{1}{\sigma}\left[\frac{1}{J}\left(\frac{|A_\xi|}{\Delta\xi} + \frac{|A_\eta|}{\Delta\eta}\right) + \frac{2}{J^2}\left(\frac{B_{\xi\xi}}{\Delta\xi^2} + \frac{B_{\eta\eta}}{\Delta\eta^2}\right)\right].$$

The structure of the inviscid and viscous Jacobian matrices is identical for the ξ and η mesh directions, the only difference being that the metric terms must be switched to correspond to the cell face in question. Therefore, it is convenient to present a generic form of the inviscid and viscous matrices with the mesh direction left unspecified in the metric terms. It is then understood that S_x represents $S_{\xi x}$ when computing $|A_\xi|$ and $S_{\eta x}$ when computing $|A_\eta|$. For Euler calculations, the matrix preconditioner incorporates only the inviscid Jacobians.

Inviscid Contributions

The absolute value of the inviscid Jacobian is based on the eigenvector decomposition

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

Defining the face normals,

$$n_x = \frac{S_x}{S}, \qquad n_y = \frac{S_y}{S},$$

and the normal and tangential velocities,

$$q_n = u\,n_x + v\,n_y, \qquad q_t = v\,n_x - u\,n_y,$$

the absolute value of the eigenvalue matrix takes the form

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

The right eigenvectors are the columns of

$$T = \begin{bmatrix} 1 & 0 & 1 & 1 \\ u & -c\,n_y & u + c\,n_x & u - c\,n_x \\ v & c\,n_x & v + c\,n_y & v - c\,n_y \\ \frac{q^2}{2} & c\,q_t & H + c\,q_n & H - c\,q_n \end{bmatrix},$$

and the left eigenvectors are the rows of T^{-1}, which has the form

$$\frac{1}{c^2} \begin{bmatrix} c^2 - (\gamma-1)\frac{q^2}{2} & (\gamma-1)u & (\gamma-1)v & -(\gamma-1) \\ -c\,q_t & -c\,n_y & c\,n_x & 0 \\ \frac{1}{2}\left[(\gamma-1)\frac{q^2}{2} - c\,q_n\right] & -\frac{1}{2}[(\gamma-1)u - c\,n_x] & -\frac{1}{2}[(\gamma-1)v - c\,n_y] & \frac{1}{2}(\gamma-1) \\ \frac{1}{2}\left[(\gamma-1)\frac{q^2}{2} + c\,q_n\right] & -\frac{1}{2}[(\gamma-1)u + c\,n_x] & -\frac{1}{2}[(\gamma-1)v + c\,n_y] & \frac{1}{2}(\gamma-1) \end{bmatrix}.$$

In evaluating the inviscid Jacobians for the preconditioner, the use of Roe-averaged variables was found to provide no benefit over simple arithmetic averages.

The application of an entropy fix (B.9) to the eigenvalues in the numerical dissipation has the effect of reducing the size of the local stability limit in each characteristic field in which the fix is active. Applying an entropy fix to the eigenvalues in the preconditioner produces a corresponding reduction in the time step so that stability is ensured. For Euler calculations, the same van Leer entropy fix (B.10) that is used in the numerical dissipation is also applied to the preconditioner, although this precaution has proven unnecessary in practice. However, for turbulent Navier–Stokes calculations on highly stretched meshes, this approach does not sufficiently limit the time step to provide robustness, so a more severe Harten entropy fix [24] is adopted, in which the eigenvalues are bounded away from zero based on a fraction of the local speed of sound. For the present calculations, the threshold parameter is chosen to be $\epsilon = \frac{c}{4}$, so that the minimum of the bounding parabola is one eighth the speed of sound and the fix in the preconditioner is active throughout much of the boundary layer.

Viscous Contributions

The viscous Jacobians in conservative variables are defined by

$$B = \frac{\partial G}{\partial \widetilde{W}} M^{-1},$$

where the columns of the matrix

$$\frac{\partial G}{\partial \widetilde{W}} = [C_1|C_2|C_3|C_4]$$

are given by

$$C_1 = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ -\frac{\gamma\mu}{Pr(\gamma-1)}\frac{p}{\rho^2}S^2 \end{array} \right\}, \quad C_2 = \left\{ \begin{array}{c} 0 \\ S_x^2(2\mu+\lambda)+S_y^2\mu \\ S_xS_y(\mu+\lambda) \\ \bar{u}\left[S_x^2(2\mu+\lambda)+S_y^2\mu\right]+\bar{v}S_xS_y(\mu+\lambda) \end{array} \right\},$$

$$C_3 = \left\{ \begin{array}{c} 0 \\ S_xS_y(\mu+\lambda) \\ S_y^2(2\mu+\lambda)+S_x^2\mu \\ \bar{v}\left[S_y^2(2\mu+\lambda)+S_x^2\mu\right]+\bar{u}S_xS_y(\mu+\lambda) \end{array} \right\}, \quad C_4 = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ \frac{\gamma\mu}{Pr(\gamma-1)}\frac{1}{\rho}S^2 \end{array} \right\},$$

and the transformation matrices have the form

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

Implementation

The 4×4 block-Jacobi preconditioner is computed for each cell before the first stage of each time step. The contributing Jacobian matrices are evaluated at the cell boundaries and averaged to the cell centers to form the preconditioner. The resulting matrix is then inverted using Gaussian elimination and stored for rapid multiplication by the residual vector during each stage of the Runge–Kutta scheme. To avoid the need for pivoting during the inversion process, elimination is begun

from the (4,4) element since the heat flux contribution ensures that in contrast to the (1,1) element, this term does not tend to zero at the wall. The Roe matrices are computed separately for the preconditioner and the numerical dissipation since, for reasons of economy, it is undesirable to explicitly form the matrices in evaluating the numerical dissipation. Using this implementation, the additional computational expense of matrix preconditioning relative to scalar preconditioning ranges between 12%-15% for both inviscid and viscous calculations.

C.2 Three Dimensions

The expressions for the three-dimensional scalar and matrix preconditioners are obtained by following the same procedure as in the two-dimensional case. Starting from conservative form in Cartesian coordinates

$$\frac{dW}{dt} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} + \frac{\partial G_z}{\partial z},$$

and linearizing the viscous terms about a constant state, the system can be re-grouped in terms of like derivatives

$$\begin{aligned} \frac{dW}{dt} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} &= \frac{\partial^2 G_{xx}}{\partial x^2} + \frac{\partial^2 G_{yy}}{\partial y^2} + \frac{\partial^2 G_{zz}}{\partial z^2} \\ &+ \frac{\partial^2 G_{xy}}{\partial x \partial y} + \frac{\partial^2 G_{yz}}{\partial y \partial z} + \frac{\partial^2 G_{zx}}{\partial z \partial x}. \end{aligned}$$

This representation can then be transformed to (ξ, η, ζ) mesh-aligned coordinates

$$\begin{aligned} \frac{dW}{dt} + \frac{1}{J}\left(\frac{\partial F_\xi}{\partial \xi} + \frac{\partial F_\eta}{\partial \eta} + \frac{\partial F_\zeta}{\partial \zeta}\right) = \\ \frac{1}{J^2}\left(\frac{\partial^2 G_{\xi\xi}}{\partial \xi^2} + \frac{\partial^2 G_{\eta\eta}}{\partial \eta^2} + \frac{\partial^2 G_{\zeta\zeta}}{\partial \zeta^2} + \frac{\partial^2 G_{\xi\eta}}{\partial \xi \partial \eta} + \frac{\partial^2 G_{\eta\zeta}}{\partial \eta \partial \zeta} + \frac{\partial^2 G_{\zeta\xi}}{\partial \zeta \partial \xi}\right), \end{aligned}$$

where, for example, the inviscid fluxes have the form

$$F_\xi = S_{\xi x} F_x + S_{\xi y} F_y + S_{\xi z} F_z,$$

and the viscous fluxes have the form

$$\begin{aligned} G_{\xi\xi} &= S_{\xi x}^2 G_{xx} + S_{\xi y}^2 G_{yy} + S_{\xi z}^2 G_{zz} \\ &+ S_{\xi x} S_{\xi y} G_{xz} + S_{\xi y} S_{\xi z} G_{yz} + S_{\xi z} S_{\xi x} G_{zx}, \end{aligned}$$

$$
\begin{aligned}
G_{\xi\eta} &= 2S_{\xi x}S_{\eta x}G_{xx} + 2S_{\xi y}S_{\eta y}G_{yy} + 2S_{\xi z}S_{\eta z}G_{zz} \\
&+ (S_{\xi x}S_{\eta y} + S_{\xi y}S_{\eta x})G_{xy} + (S_{\xi y}S_{\eta z} + S_{\xi z}S_{\eta y})G_{yz} + (S_{\xi z}S_{\eta x} + S_{\xi x}S_{\eta z})G_{zx}.
\end{aligned}
$$

Introduction of the flux Jacobians then leads to the linearized form of the three-dimensional Navier–Stokes equations in computational coordinates

$$
\begin{aligned}
\tfrac{dW}{dt} &+ \tfrac{1}{J}\left(A_\xi \tfrac{\partial W}{\partial \xi} + A_\eta \tfrac{\partial W}{\partial \eta} + A_\zeta \tfrac{\partial W}{\partial \zeta}\right) = \\
&\tfrac{1}{J^2}\left(B_{\xi\xi}\tfrac{\partial^2 W}{\partial \xi^2} + B_{\eta\eta}\tfrac{\partial^2 W}{\partial \eta^2} + B_{\zeta\zeta}\tfrac{\partial^2 W}{\partial \zeta^2} + B_{\xi\eta}\tfrac{\partial^2 W}{\partial \xi \partial \eta} + B_{\eta\zeta}\tfrac{\partial^2 W}{\partial \eta \partial \zeta} + B_{\zeta\xi}\tfrac{\partial^2 W}{\partial \zeta \partial \xi}\right).
\end{aligned}
$$

The Jacobians are once again found by first differentiating with respect to a non-conservative vector of flow variables $\widetilde{W} = \{\rho, u, v, w, p\}^T$ and then transforming to conservative variables using the matrix $M = \frac{\partial W}{\partial \widetilde{W}}$. The spatial residual operator based on the linear system then takes the form

$$
\begin{aligned}
R &= \tfrac{1}{J}\left(\tfrac{A_\xi}{2\Delta\xi}\delta_{2\xi} - \tfrac{|A_\xi|}{2\Delta\xi}\delta_{\xi\xi} + \tfrac{A_\eta}{2\Delta\eta}\delta_{2\eta} - \tfrac{|A_\eta|}{2\Delta\eta}\delta_{\eta\eta} + \tfrac{A_\zeta}{2\Delta\zeta}\delta_{2\zeta} - \tfrac{|A_\zeta|}{2\Delta\zeta}\delta_{\zeta\zeta}\right) \\
&- \tfrac{1}{J^2}\left(\tfrac{B_{\xi\xi}}{\Delta\xi^2}\delta_{\xi\xi} + \tfrac{B_{\eta\eta}}{\Delta\eta^2}\delta_{\eta\eta} + \tfrac{B_{\zeta\zeta}}{\Delta\zeta^2}\delta_{\zeta\zeta} + \tfrac{B_{\xi\eta}}{4\Delta\xi\Delta\eta}\delta_{2\xi 2\eta} + \tfrac{B_{\eta\zeta}}{4\Delta\eta\Delta\zeta}\delta_{2\eta 2\zeta} + \tfrac{B_{\zeta\xi}}{4\Delta\zeta\Delta\xi}\delta_{2\zeta 2\xi}\right),
\end{aligned}
$$

where the inviscid terms have once again been upwinded using a Roe linearization [64].

C.2.1 Scalar Preconditioner

The three-dimensional scalar preconditioner for the Navier–Stokes equations takes the form

$$
P_s^{-1} = \frac{1}{\sigma}\left[\frac{1}{J}\left(\frac{\rho(A_\xi)}{\Delta\xi} + \frac{\rho(A_\eta)}{\Delta\eta} + \frac{\rho(A_\zeta)}{\Delta\zeta}\right) + \frac{2}{J^2}\left(\frac{\rho(B_{\xi\xi})}{\Delta\xi^2} + \frac{\rho(B_{\eta\eta})}{\Delta\eta^2} + \frac{\rho(B_{\zeta\zeta})}{\Delta\zeta^2}\right)\right].
$$

Defining the flow speed q,

$$
q^2 = u^2 + v^2 + w^2,
$$

the speed of sound c,

$$
c^2 = (\gamma - 1)\left(H - \frac{q^2}{2}\right),
$$

the flux rate through the ξ face Q_ξ,

$$
Q_\xi = S_{\xi x}u + S_{\xi y}v + S_{\xi z}w,
$$

and the area of the ξ face S_ξ,

$$
S_\xi^2 = S_{\xi x}^2 + S_{\xi y}^2 + S_{\xi z}^2,
$$

the spectral radii of the inviscid flux Jacobians may then be expressed as

$$\rho(A_\xi) = |Q_\xi| + cS_\xi, \qquad \rho(A_\eta) = |Q_\eta| + cS_\eta, \qquad \rho(A_\zeta) = |Q_\zeta| + cS_\zeta,$$

and spectral radii of the viscous flux Jacobians take the form

$$\rho(B_{\xi\xi}) = \frac{\gamma\mu}{Pr\rho}S_\xi^2, \qquad \rho(B_{\eta\eta}) = \frac{\gamma\mu}{Pr\rho}S_\eta^2, \qquad \rho(B_{\zeta\zeta}) = \frac{\gamma\mu}{Pr\rho}S_\zeta^2.$$

C.2.2 Matrix Preconditioner

The three-dimensional block-Jacobi preconditioner takes the form

$$P^{-1} = \frac{1}{\sigma}\left[\frac{1}{J}\left(\frac{|A_\xi|}{\Delta\xi} + \frac{|A_\eta|}{\Delta\eta} + \frac{|A_\zeta|}{\Delta\zeta}\right) + \frac{2}{J^2}\left(\frac{B_{\xi\xi}}{\Delta\xi^2} + \frac{B_{\eta\eta}}{\Delta\eta^2} + \frac{B_{\zeta\zeta}}{\Delta\zeta^2}\right)\right].$$

Inviscid Contributions

The absolute value of the conservative Jacobian is

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

Defining the face normals,

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

and the normal velocity,

$$q_n = u\,n_x + v\,n_y + w\,n_z,$$

the absolute value of the eigenvalue matrix appears as

$$\Lambda = J^{-1}\begin{bmatrix} |Q| & 0 & 0 & 0 & 0 \\ 0 & |Q| & 0 & 0 & 0 \\ 0 & 0 & |Q| & 0 & 0 \\ 0 & 0 & 0 & |Q + c\,S| & 0 \\ 0 & 0 & 0 & 0 & |Q - c\,S| \end{bmatrix}.$$

The right eigenvectors are the columns of

$$T = [R_1|R_2|R_3|R_4|R_5],$$

given by

$$R_1 = \left\{ \begin{array}{c} n_x \\ u\,n_x \\ v\,n_x + c\,n_z \\ w\,n_x - c\,n_y \\ \frac{q^2}{2}\,n_x + c\,(v\,n_z - w\,n_y) \end{array} \right\}, \quad R_2 = \left\{ \begin{array}{c} n_y \\ u\,n_y - c\,n_z \\ v\,n_y \\ w\,n_y + c\,n_x \\ \frac{q^2}{2}\,n_y + c\,(w\,n_x - u\,n_z) \end{array} \right\}$$

$$R_3 = \left\{ \begin{array}{c} n_z \\ u\,n_z + c\,n_y \\ v\,n_z - c\,n_x \\ w\,n_z \\ \frac{q^2}{2}\,n_z + c\,(u\,n_y - v\,n_x) \end{array} \right\}, \quad R_4 = \left\{ \begin{array}{c} 1 \\ u + c\,n_x \\ v + c\,n_y \\ w + c\,n_z \\ H + c\,q_n \end{array} \right\}, \quad R_5 = \left\{ \begin{array}{c} 1 \\ u - c\,n_x \\ v - c\,n_y \\ w - c\,n_z \\ H - c\,q_n \end{array} \right\},$$

and the left eigenvectors are the rows of

$$T^{-1} = \frac{1}{c^2}[L_1|L_2|L_3|L_4|L_5]^T,$$

given by

$$L_1 = \left\{ \begin{array}{c} \left[c^2 - (\gamma-1)\frac{q^2}{2}\right] n_x + c\,(w\,n_y - v\,n_z) \\ (\gamma-1)u\,n_x \\ (\gamma-1)v\,n_x + c\,n_z \\ (\gamma-1)w\,n_x - c\,n_y \\ -(\gamma-1)n_x \end{array} \right\},$$

$$L_2 = \left\{ \begin{array}{c} \left[c^2 - (\gamma-1)\frac{q^2}{2}\right] n_y + c\,(u\,n_z - w\,n_x) \\ (\gamma-1)u\,n_y - c\,n_z \\ (\gamma-1)v\,n_y \\ (\gamma-1)w\,n_y + c\,n_x \\ -(\gamma-1)n_y \end{array} \right\},$$

$$L_3 = \left\{ \begin{array}{c} \left[c^2 - (\gamma-1)\frac{q^2}{2}\right] n_z + c(v\,n_x - u\,n_y) \\ (\gamma-1)u\,n_z + c\,n_y \\ (\gamma-1)v\,n_z - c\,n_x \\ (\gamma-1)w\,n_z \\ -(\gamma-1)\,n_z \end{array} \right\},$$

$$L_4 = \left\{ \begin{array}{c} \frac{1}{2}\left[(\gamma-1)\frac{q^2}{2} - c\,q_n\right] \\ -\frac{1}{2}\left[(\gamma-1)u - c\,n_x\right] \\ -\frac{1}{2}\left[(\gamma-1)v - c\,n_y\right] \\ -\frac{1}{2}\left[(\gamma-1)w - c\,n_z\right] \\ \frac{1}{2}(\gamma-1) \end{array} \right\}, \quad L_5 = \left\{ \begin{array}{c} \frac{1}{2}\left[(\gamma-1)\frac{q^2}{2} + c\,q_n\right] \\ -\frac{1}{2}\left[(\gamma-1)u + c\,n_x\right] \\ -\frac{1}{2}\left[(\gamma-1)v + c\,n_y\right] \\ -\frac{1}{2}\left[(\gamma-1)w + c\,n_z\right] \\ \frac{1}{2}(\gamma-1) \end{array} \right\}.$$

The same entropy fix is applied to the eigenvalues in the three-dimensional preconditioner as in the two-dimensional case.

Viscous Contributions

The viscous Jacobian in conservative variables is defined by $B = \frac{\partial G}{\partial \widetilde{W}} M^{-1}$, where

$$\frac{\partial G}{\partial \widetilde{W}} = [C_1|C_2|C_3|C_4|C_5],$$

with columns defined by

$$C_1 = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{\gamma\mu}{(\gamma-1)Pr}\frac{p}{\rho^2}S^2 \end{array} \right\}, \qquad C_5 = \left\{ \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ \frac{\gamma\mu}{(\gamma-1)Pr}\frac{1}{\rho}S^2 \end{array} \right\},$$

$$C_2 = \left\{ \begin{array}{c} 0 \\ S_x^2(2\mu+\lambda) + (S_y^2+S_z^2)\mu \\ S_xS_y(\mu+\lambda) \\ S_zS_x(\mu+\lambda) \\ \bar{u}\left[S_x^2(2\mu+\lambda) + (S_y^2+S_z^2)\mu\right] + (\mu+\lambda)\left[S_xS_y\bar{v} + S_zS_x\bar{w}\right] \end{array} \right\},$$

$$C_3 = \left\{ \begin{array}{c} 0 \\ S_xS_y(\mu+\lambda) \\ S_y^2(2\mu+\lambda) + (S_z^2+S_x^2)\mu \\ S_yS_z(\mu+\lambda) \\ \bar{v}\left[S_y^2(2\mu+\lambda) + (S_z^2+S_x^2)\mu\right] + (\mu+\lambda)\left[S_yS_z\bar{w} + S_xS_y\bar{u}\right] \end{array} \right\},$$

$$C_4 = \left\{ \begin{array}{c} 0 \\ S_z S_x(\mu+\lambda) \\ S_y S_z(\mu+\lambda) \\ S_z^2(2\mu+\lambda) + (S_x^2+S_y^2)\mu \\ \bar{w}\left[S_z^2(2\mu+\lambda) + (S_x^2+S_y^2)\mu\right] + (\mu+\lambda)\left[S_z S_x \bar{u} + S_y S_z \bar{v}\right] \end{array} \right\}.$$

The transformation matrices have the form

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

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

Implementation

For the three-dimensional implementation, the 5×5 block-Jacobi preconditioner is once again computed for each cell before the first stage of each time step. As an alternative to inverting the matrix, the preconditioner is instead stored as UL factors for inexpensive back substitution during each stage of the Runge–Kutta scheme. A UL decomposition is used instead of the more standard LU approach to avoid the need for pivoting by starting with the (5,5) element which does not tend to zero at the wall due to the heat flux contribution. For the three-dimensional solver, the preconditioner was formed without using Roe-averaged variables in the inviscid contributions to avoid the expense of computing multiple square roots. No adverse effects on either efficiency or robustness were observed. Using this implementation, the additional computational expense of matrix preconditioning relative to scalar preconditioning is approximately 15% for turbulent Navier–Stokes calculations.

Appendix D

Multigrid Algorithms

This appendix describes both the full and J-coarsened multigrid algorithms in two dimensions. The three-dimensional implementations are obtained by straightforward extension of these descriptions.

D.1 Full Coarsened Multigrid

Full coarsened multigrid is implemented based on the full approximation scheme developed for the Euler equations by Jameson [27] and extended to the Navier–Stokes equations by Martinelli and Jameson [46]. Coarser meshes are generated by removing alternate mesh points in both directions. In the descriptions that follow, quantities on the fine mesh are denoted by subscripts (a,b) and quantities on the next coarser mesh have subscripts ($2a$,$2b$) to indicate that the size of the mesh interval has doubled in both the ξ and η directions.

The multigrid cycle begins with an application of an M-stage Runge–Kutta time-stepping scheme on the fine mesh. Flow variables are then defined on the next coarser mesh using the cell volumes as weights to conserve mass, momentum and energy

$$W^{(0)}_{2a,2b} = \frac{\sum_1^4 J_{a,b} W^{(M)}_{a,b}}{J_{2a,2b}} .$$

A forcing function for the coarse mesh is defined based on a set of residuals calculated

after the M-th stage on the fine mesh and before the first stage on the coarse mesh

$$Q_{2a,2b} = \left[\sum_{1}^{4} R_{a,b}\left(W_{a,b}^{(M+1)}\right)\right] - R_{2a,2b}\left(W_{2a,2b}^{(0)}\right) . \tag{D.1}$$

This forcing function drives all stages of the M-stage Runge–Kutta scheme on the coarse mesh, so that the k-th stage appears as

$$W^{(k)} = W^{n} - \alpha_k P(R_{2a,2b}^{(k-1)} + Q_{2a,2b}) . \tag{D.2}$$

The unforced form of the Runge–Kutta scheme (B.11) is used only on the finest mesh. High resolution numerical dissipation is used only on the finest mesh and first order dissipation is used on all coarser meshes. Wall and far field boundary conditions are evaluated on the coarse mesh exactly as on the fine mesh, with the exception that for Euler calculations, the pressure extrapolation based on wall curvature is abandoned on coarse meshes due to the loss of smoothness in the geometric definition. After completing the M-stage scheme on grid $(2a, 2b)$, the process is then repeated for the next coarser grid denoted by $(4a, 4b)$.

After applying the M-stage scheme on the coarsest grid, say $(8a, 8b)$ for a 4 level cycle, the coarse grid correction to the next finer grid is calculated by subtracting out the initial coarse grid values

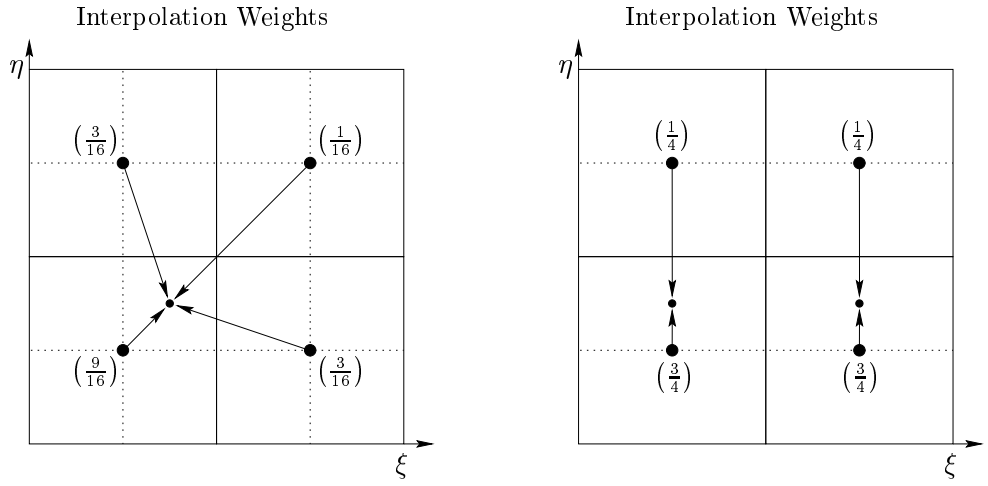
$$\Delta W_{8a,8b} = W_{8a,8b}^{(M)} - W_{8a,8b}^{(0)} .$$

This correction is then bilinearly interpolated to the next finer mesh and added to the stored solution on that mesh

$$W_{4a,4b}^{\text{new}} = W_{4a,4b} + I\left(\Delta W_{8a,8b}\right) , \tag{D.3}$$

where I is the interpolation operator based on the weighting coefficients $\left(\frac{9}{16}, \frac{3}{16}, \frac{3}{16}, \frac{1}{16}\right)$ as depicted in Fig. D.1a.

The multigrid cycle is traversed in either a V or a W pattern as defined in Fig. D.2. Using a V-cycle, the process continues by applying the M-stage scheme (driven by the previously computed forcing function $Q_{(4a,4b)}$) to the new solution $W_{4a,4b}^{\text{new}}$, before calculating and interpolating the correction to the next finer mesh.



D.1a Full Coarsened Multigrid. **D.1b** J-Coarsened Multigrid.

Figure D.1 Interpolation weights for transferring coarse grid corrections to the next finer grid.

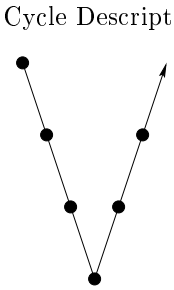
This process continues until the cycle is completed by correcting the solution on the finest mesh. Using a W-cycle, the M-stage scheme is applied only when moving down the cycle and the corrections are computed and interpolated without performing a flow evaluation when moving up the cycle. The forcing function Q is re-computed at each point in the cycle at which the solution is transferred from a finer to a coarser mesh.

D.2 J-Coarsened Multigrid

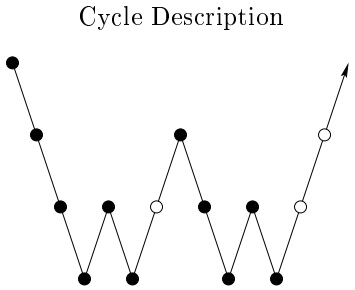
For a J-coarsened multigrid approach, the mesh is coarsened only in the direction normal to the wall, which is assumed to be located at $\eta = 0$. Quantities on the fine mesh are denoted by subscripts (a, b) as before, but quantities on the next coarser mesh now have subscripts $(a, 2b)$ to indicate that the mesh interval remains unchanged in the ξ direction and has doubled in the η direction. With the following new definitions for the multigrid operators, the basic procedure for traversing the multigrid cycle remains unchanged from the previously defined full coarsened approach.

The volume-weighted transfer to a coarse mesh becomes

$$W_{a,2b}^{(0)} = \frac{\sum_1^2 J_{a,b} W_{a,b}^{(M)}}{J_{a,2b}},$$



D.2a V-cycle (4 levels).



D.2b W-Cycle (4 levels).

Figure D.2 Multigrid cycle descriptions: (•) Apply M-stage Runge–Kutta scheme and transfer, (∘) Transfer without relaxation.

while the forcing function for the coarse mesh now takes the form

$$Q_{a,2b} = \left[\sum_{1}^{2} R_{a,b}\left(W_{a,b}^{(M+1)}\right)\right] - R_{a,2b}\left(W_{a,2b}^{(0)}\right),$$

and the k-th stage of the M-stage Runge–Kutta scheme on the coarse mesh appears as

$$W^{(k)} = W^{n} - \alpha_k P(R_{a,2b}^{(k-1)} + Q_{a,2b}). \tag{D.4}$$

For a four level scheme, the correction from the coarsest mesh is then

$$\Delta W_{a,8b} = W_{a,8b}^{(M)} - W_{a,8b}^{(0)},$$

and the correction is linearly interpolated to the next finer mesh using

$$W_{a,4b}^{\text{new}} = W_{a,4b} + I\left(\Delta W_{a,8b}\right),$$

where I is the interpolation operator based on the weighting coefficients $\left(\frac{3}{4}, \frac{1}{4}\right)$ as depicted in Fig. D.1b.

Appendix E

Turbulence Models

This appendix thoroughly documents the two-dimensional implementations of the Baldwin–Lomax and Spalart–Allmaras turbulence models used in the present work. The extension of these implementations to three dimensions follows naturally from the approaches described here.

E.1 Baldwin–Lomax Model

The Baldwin–Lomax turbulence model [6] is a zero-equation or algebraic model which defines the eddy viscosity using a two-layer formulation

$$\mu_t = \begin{cases} \mu_{t_{\text{inner}}}, & y_n \leq y_{\text{crossover}}, \\ \mu_{t_{\text{outer}}}, & y_n > y_{\text{crossover}}, \end{cases}$$

where y_n is the normal distance to the wall and $y_{\text{crossover}}$ is the minimum value of y_n at which $\mu_{t_{\text{inner}}} = \mu_{t_{\text{outer}}}$. In the present implementation, y_n is approximated by the distance to the wall moving along mesh lines in the J direction.

In the inner layer, the turbulent eddy viscosity is defined by

$$\mu_{t_{\text{inner}}} = \rho l^2 S,$$

where ρ is the density, l is the reference length,

$$l = k y_n \left[1 - \exp(-y^+/A^+)\right],$$

and S is the magnitude of the vorticity

$$S = \left| \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right|.$$

The wall factor is given by

$$y^+ = \frac{\sqrt{\rho_w \tau_w}}{\mu_w} y_n,$$

where ρ_w, μ_w and τ_w are the density, molecular viscosity, and laminar shear stress at the wall.

In the outer layer, the turbulent eddy viscosity is defined by

$$\mu_{t_{\text{outer}}} = K C_{cp} \rho F_{\text{wake}} F_{\text{kleb}},$$

with

$$F_{\text{wake}} = \min(y_{\max} F_{\max}, C_{wk} y_{\max} U_{\text{diff}}^2 / F_{\max}),$$

and

$$F_{\text{kleb}} = \left[1 + 5.5 \left(C_{\text{kleb}} \frac{y_n}{y_{\max}} \right)^6 \right]^{-1}.$$

Here, $F_{\max}$ represents the maximum within a given profile of the function

$$F(y_n) = y_n S \left[1 - \exp(-y^+ / A^+) \right],$$

and $y_{\max}$ is the value of y_n at which this maximum occurs. Also, U_{diff} is given by

$$U_{\text{diff}} = q_{\max} - q_{\min},$$

where $q_{\max}$ is the velocity magnitude at $y_{\max}$ and $q_{\min}$ is the minimum velocity magnitude in the profile (zero for a profile containing a wall). To promote smoothness in the eddy viscosity field, the values of $F_{\max}$, $y_{\max}$ and $U_{\max}$ are evaluated using quadratic polynomial fits to the cell centered values that bracket the maximum of $F(y_n)$. For wake profiles, only the outer formulation is used and the exponential is dropped in the evaluation of $F(y_n)$. The model constants are given by

$$k = 0.4, \qquad A^+ = 26, \qquad K = 0.0168,$$
$$C_{cp} = 1.6, \qquad C_{wk} = 1.0, \qquad C_{\text{kleb}} = 0.3.$$

Transition is fixed based on the experimental trip location by setting the turbulent viscosity to zero up to a specified percentage of the chord. For use with multigrid, the turbulent viscosity is evaluated prior to relaxation on the fine mesh and frozen on all coarser meshes.

E.2 Spalart–Allmaras Model

E.2.1 Description

The Spalart–Allmaras turbulence model [68] is a one-equation model that takes the form of a scalar convection–diffusion equation with source terms

$$\frac{\partial \tilde{\nu}}{\partial t} + u\frac{\partial \tilde{\nu}}{\partial x} + v\frac{\partial \tilde{\nu}}{\partial y} = \frac{1}{c_{b3}}\left\{\nabla \cdot [(\nu + \tilde{\nu})\nabla\tilde{\nu}] + c_{b2}\nabla\tilde{\nu}\cdot\nabla\tilde{\nu}\right\} + Q,$$

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

$$Q = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} - (c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2})\left(\frac{\tilde{\nu}}{d}\right)^2 + f_{t1}(\Delta U)^2, \tag{E.1}$$

which may be divided into production, destruction and trip contributions

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

using the definitions

$$\begin{aligned} \tilde{\nu}P(\tilde{\nu}) &= c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu}, \\ \tilde{\nu}D(\tilde{\nu}) &= (c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2})\left(\frac{\tilde{\nu}}{d}\right)^2, \\ T &= f_{t1}(\Delta U)^2. \end{aligned} \tag{E.2}$$

Here, the production and destruction terms have been represented as products for future notational convenience. The trip term provides a mechanism for triggering transition at a specified location on the geometry.

The turbulent transport equation retains its original form after non-dimensionalization using the variables

$$\tilde{\nu}^* = \frac{\tilde{\nu}}{\sqrt{p_\infty/\rho_\infty}L}, \qquad \mu_t^* = \rho^*\tilde{\nu}^* f_{v1},$$

and the other non-dimensional variables B.1 previously defined for the flow equations. The turbulent eddy viscosity is defined by

$$\mu_t = \rho\tilde{\nu}f_{v1},$$

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

$$f_{v1} = \frac{\chi^3}{\chi^3+c_{v1}^3}, \qquad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \qquad \chi = \frac{\tilde{\nu}}{\nu_L},$$

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

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

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

$$c_{b1} = 0.1355, \qquad c_{b2} = 0.622, \qquad c_{b3} = \frac{2}{3}, \qquad c_{v1} = 7.1,$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{c_{b3}}, \qquad c_{w2} = 0.3, \qquad c_{w3} = 2, \qquad \kappa = 0.41.$$

The auxiliary relations for the trip terms are

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

with the additional closure constants given by

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

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

E.2.2 Implementation

In discretizing the scalar turbulence equation, the same basic solution strategy is employed as for the flow equations. However, to improve the numerical properties of the discrete turbulence model, several modifications have been introduced both to the model description and to the discretization procedure.

Modifications to Model Description

To improve robustness, the non-conservative convective terms in the model are replaced by the conservative form

$$\frac{\partial \tilde{\nu} u}{\partial x} + \frac{\partial \tilde{\nu} v}{\partial y},$$

which amounts to adding the term

$$\tilde{\nu} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right),$$

to the original model. This term could become significant across a shock, but numerical tests have revealed that the influence on the solution is negligible compared to the error bars implied by the deviation from experimental data and results computed using other turbulence models.

In implementing the turbulence model, one of the primary concerns is ensuring that the turbulent working variable $\tilde{\nu}$ always remains positive. The non-conservative diffusion term

$$\nabla \tilde{\nu} \cdot \nabla \tilde{\nu}$$

is difficult to discretize in a manner that ensures positivity, so the combined diffusion operators are reformulated in the conservative form [67]

$$\frac{1 + c_{b2}}{c_{b3}} \nabla \cdot [(\nu + \tilde{\nu}) \nabla \tilde{\nu}] - \frac{c_{b2}}{c_{b3}} (\nu + \tilde{\nu}) \nabla^2 \tilde{\nu}.$$

This corresponds to supplementing the original model with the term

$$\frac{c_{b2}}{c_{b3}} \nabla \nu \cdot \nabla \tilde{\nu},$$

which is negligible in regions where the turbulent viscosity is significant.

Modifications to Discretization Procedure

Using the conservative reformulation of the convective and diffusive operators described above, the flux balances can be constructed using the scalar analogs of the flux formulae described for the Navier–Stokes equations in Sections B.2.1, B.2.2 and

B.2.3. For the turbulence equation, only the first order form of the numerical dissipation is used since the dominant flux balance occurs between the physical diffusion operator and the turbulent source terms. The turbulent source terms are sensitive to the method used to compute the distance to the wall, so it is important to determine the actual distance from the center of each field cell to the surface of the computational model rather than use a less expensive approximation such as the distance along mesh lines.

To preserve the positivity of $\tilde{\nu}$, several modifications are made to the Runge–Kutta time integration procedure described in Section B.4. The negative contributions of the turbulent source operator Q are treated implicitly to limit the rate of exponential decay in the solution. This is equivalent to employing the standard explicit integration procedure (B.11) with a reduced local time step (scalar preconditioner)

$$\Delta t_{\text{imp}} = \frac{\Delta t}{1 + Q_{\tilde{\nu}} \Delta t}, \tag{E.3}$$

where Δt is the explicit time step given by

$$\Delta t^{-1} = \frac{1}{\sigma} \left\{ \frac{1}{J} \left(\frac{|Q_\xi|}{\Delta \xi} + \frac{|Q_\eta|}{\Delta \eta} \right) + \frac{2(1 + c_{b2})}{J^2 c_{b3}} \left[\frac{(\nu + \tilde{\nu}) S_\xi^2}{\Delta \xi^2} + \frac{(\nu + \tilde{\nu}) S_\eta^2}{\Delta \eta^2} \right] \right\}.$$

Here, σ is the Courant number and the flux rate Q_ξ and face area S_ξ are defined by (C.4) and (C.5), respectively. For robustness, the hyperbolic contribution to the time step is prevented from becoming too small using the fix construction (B.9) with $\epsilon = \frac{c}{8}$, where c is the local speed of sound.

Following the approach of Spalart and Allmaras [67], the Jacobian $Q_{\tilde{\nu}}$ is defined so as to preserve positivity while minimizing the adverse effects on convergence. Defining the operator

$$\text{pos}(q) = \begin{cases} q, & q \geq 0, \\ 0, & q < 0, \end{cases}$$

the Jacobian is then formulated as

$$Q_{\tilde{\nu}} \equiv -\text{pos}(D - P) - \text{pos}(D' - P')\tilde{\nu},$$

where P and D are production and destruction terms defined by (E.2). These terms are analytically differentiated with respect to $\tilde{\nu}$ to determined P' and D'.

This definition for the Jacobian ensures that the time step will not be unnecessarily reduced in regions of the flow where cancellation occurs between individually large contributions from the production and destruction operators. No implicit treatment of the trip term is required since it is positive independently from the value of $\tilde{\nu}$.

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 amount to a guarantee of 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. A single stage of the Runge–Kutta update process (B.11) takes the form

$$\tilde{\nu}^{(k)} = \tilde{\nu}^n - \Delta\tilde{\nu},$$

where the standard update is defined by

$$\Delta\tilde{\nu} = \alpha_k \Delta t_{\text{imp}} R^{(k-1)}.$$

To guarantee positivity, this update is then replaced by the limited update

$$\Delta\tilde{\nu}_{\text{lim}} = \begin{cases} \Delta\tilde{\nu}, & \Delta\tilde{\nu} \leq 0, \\ \frac{(\tilde{\nu}^n - \tilde{\nu}_{\text{min}})\Delta\tilde{\nu}}{(\tilde{\nu}^n - \tilde{\nu}_{\text{min}}) + \Delta\tilde{\nu}}, & \Delta\tilde{\nu} > 0, \end{cases} \tag{E.4}$$

which smoothly reduces the change in the solution for positive values of $\Delta\tilde{\nu}$, so that a minimum value $\tilde{\nu}_{\text{min}}$ is maintained throughout the domain. In practice, this minimum is chosen to be the free stream value $\tilde{\nu}_{\text{min}} = \tilde{\nu}_\infty = 1{\times}10^{-12}$.

The turbulence equation is solved using the same multigrid algorithm as for the flow equations, with the following modifications to account for the numerical properties of the turbulence model. Multigrid is ineffective in removing errors in the source terms because there is no coupling between the source operators in neighboring cells. Furthermore, the source terms are poorly resolved on the coarse meshes so that the corrections arising from these meshes can actually interfere with the convergence process. Therefore, in regions of the flow where the source terms are strongly active, it is desirable to deflate the fine mesh contribution to the forcing function (D.1) that drives the coarse mesh corrections. This is accomplished using

the denominator from the implicit time step (E.3), which is equal to unity in inviscid regions of the flow and is dominated by the Jacobian $Q_{\tilde{\nu}}$ inside the boundary layer. For a full coarsened algorithm, the forcing term on mesh $(2a, 2b)$ then takes the form

$$Q_{2a,2b} = \frac{1}{1 + Q_{\tilde{\nu}}\Delta t}\left[\sum_{1}^{4} R_{a,b}\left(\tilde{\nu}_{a,b}^{(M+1)}\right)\right] - R_{2a,2b}\left(\tilde{\nu}_{2a,2b}^{(0)}\right) .$$

The positivity of $\tilde{\nu}$ is maintained by the volume weighted transfer operator (D.1) when moving to coarser meshes in the cycle and by the limited update (E.4) when applying the Runge–Kutta scheme on each of these meshes. However, the possibility exists that negative values could be produced by the interpolation of coarse grid corrections to the finer meshes. Therefore, the correction procedure (D.3) is modified to become

$$\tilde{\nu}_{4a,4b}^{\text{new}} = \max(\tilde{\nu}_{4a,4b} + I\left(\Delta\tilde{\nu}_{8a,8b}\right), \tilde{\nu}_{\min}) ,$$

so that $\tilde{\nu}$ is prevented from falling below a specified minimum value $\tilde{\nu}_{\min} = \tilde{\nu}_{\infty}$.