

Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes

Niles A. Pierce and Michael B. Giles

Oxford University Computing Laboratory, Numerical Analysis Group, Oxford OX1 3QD, United Kingdom
E-mail: Niles.Pierce@comlab.ox.ac.uk

Received April 29, 1996; revised May 29, 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, yielding 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. The computational savings using this new approach are roughly a factor of 10 for turbulent Navier–Stokes calculations on highly stretched meshes. © 1997 Academic Press

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 flow geometries. The present work focuses entirely on the efficiency aspects of this pursuit for the two systems of governing equations that have been of principle interest 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 producing efficient 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 over-

whelming. 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.

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 the boundary layer. The resulting high aspect ratio cells greatly reduce the efficiency of existing numerical algorithms. The design of an appropriate numerical approach must therefore be based on a careful assessment of the interaction between the discrete method, the computational mesh, and the physics of the viscous flow.

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 [1–5]. 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 [6–8] 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, 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 to ensuring insensitivity to problem size. The propagative efficiency of the relaxation method remains important to the performance of the multigrid algorithm, but it 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 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 they suffer from the limited stability imposed by the CFL condition. 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. 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. 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.* [9] which uses multistage Runge–Kutta time-stepping to integrate the o.d.e. 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 [7, 10]. 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

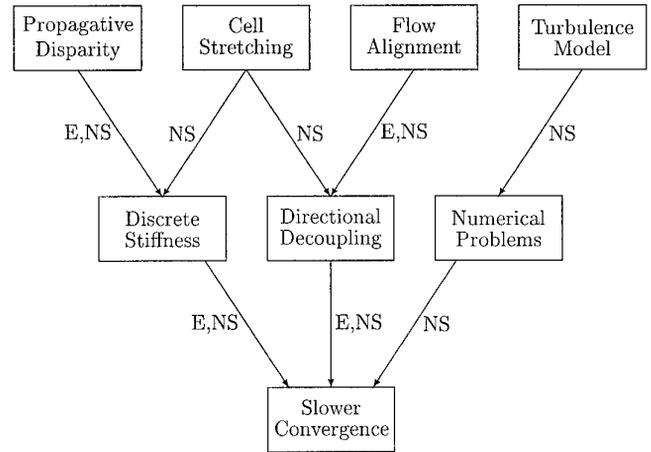


FIG. 1. Diagnosis of multigrid breakdown for the Euler and Navier–Stokes equations.

large stability limits along the imaginary and negative real axes. Explicit 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 [7, 8]. However, despite the favorable convergence rates observed for Euler computations, this 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.

The hierarchy of factors leading to multigrid inefficiency are illustrated in Fig. 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 residual eigenvalues to fall near the origin in the complex plane so that they can be neither damped nor propagated efficiently by the multistage relaxation scheme. For Euler computations, discrete stiffness results primarily from the use of a scalar preconditioner (local time step) [11], which is unable to cope with the inherent disparity in the propagative speeds of convective and acoustic modes. This problem is relatively localized for transonic flows since the 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. Although improvements are possible, these shortcomings have not prevented the attainment of sufficiently rapid convergence to meet industrial requirements for inviscid flow calculations [12] and do not represent a substantial concern to the CFD community.

For Navier–Stokes computations, 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 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 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 [13, 14].

One means of combatting discrete stiffness in the Euler and Navier–Stokes equations is the use of a matrix time step or preconditioner [15, 11, 16–20]. 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 multistage scheme can provide rapid damping and propagation [21, 22]. This is an alternative viewpoint to the one invoked by those who have developed preconditioners for low-Mach number and incompressible flows [15, 23–25], where the focus is placed on eliminating *analytic* stiffness arising from the inherent propagative disparities in the limit of vanishing Mach number. In certain cases, preconditioning methods can also be used to alleviate the problem of directional decoupling [22, 14, 26]. Another method for countering directional decoupling is the use of directional coarsening multigrid algorithms [27]. The interaction between the preconditioner and the multigrid coarsening algorithm is critical, making it 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 the traditional full coarsened multigrid and for the semi-coarsening multigrid algorithm of Mulder [26, 27]. 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 [26]. Buelow *et al.* [28] have employed a related strategy based on a different local matrix preconditioner [29] and ADI relaxation.

Alternatively, the semi-coarsening algorithm proposed by Mulder [27] 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 a single fine mesh evaluation, 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.* [19] that has previously been demonstrated to be effective in clustering high frequency eigenvalues away from the origin [21]. For grids that are not coarsened in one of the mesh directions, Allmaras proposes using a semi-implicit line-Jacobi preconditioner in that direction [26].

These strategies for preconditioning in the context of both full and semi-coarsened multigrid are well-conceived. 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 $O(N)$ [27].

The purpose of the present work is to analyze and implement two less expensive preconditioned multigrid methods that are designed to perform efficiently for Euler and turbulent Navier–Stokes calculations, respectively. In the case of the Euler equations, existing multigrid solvers employing a standard scalar preconditioner [11] and a full coarsened strategy routinely demonstrate relatively good convergence rates [12] 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 incur a substantial increase in 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. 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 block-Jacobi matrix preconditioner proposed by Morano *et al.* [19] and suggested by Allmaras for use with the more expensive semi-coarsened strategy [26]. For Euler calculations on typical inviscid meshes, the new approach of matrix preconditioning and full coarsened multigrid yields computational savings of roughly a factor of three over the standard combination of scalar preconditioning and full coarsened multigrid [22, 14].

The development of an efficient preconditioned multigrid method for turbulent Navier–Stokes calculations represents a far more demanding challenge since existing methods have proven largely inadequate for coping with the problems arising from highly stretched boundary layer cells. To identify the specific nature of these problems 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 [22, 14]. 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 [22]. 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 [14]. Therefore, it is not necessary to resort to either an implicit preconditioner or a complete semi-coarsening algorithm to produce a preconditioned multigrid method that effectively damps all modes. For the computation of two-dimensional turbulent Navier–Stokes flows, this 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 full coarsened multigrid with a scalar preconditioner.

This new preconditioned multigrid method has recently been extended to three-dimensional turbulent Navier–Stokes calculations [30] and has been shown to provide essentially the same convergence rates as for two-dimen-

sional calculations. The improved preconditioned multigrid methods have also been incorporated successfully into other research projects that rely on a steady state solver as an inner kernel. These include both optimal design by adjoint methods [31, 32] and unsteady simulations based on an inner multigrid iteration [33–35].

2. APPROACH

2.1. Scheme Description

Construction and analysis of the proposed methods proceeds from the two-dimensional linearized Navier–Stokes equations in Cartesian coordinates

$$\frac{\partial W}{\partial t} + A \frac{\partial W}{\partial x} + B \frac{\partial W}{\partial y} = C \frac{\partial^2 W}{\partial x^2} + D \frac{\partial^2 W}{\partial y^2} + E \frac{\partial^2 W}{\partial x \partial y},$$

where W is the state vector, A and B are the inviscid flux Jacobians, and C , D , and E are the viscous flux Jacobians. A preconditioned semi-discrete finite volume discretization of this system appears as

$$L_t W + PR(W) = 0, \quad (1)$$

where $R(W)$ is the residual vector of the spatial discretization, L_t represents the multistage 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 nonsingular 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 linear operator

$$R = \frac{A}{2 \Delta x} \delta_{2x} - \frac{|A|}{2 \Delta x} \delta_{xx} + \frac{B}{2 \Delta y} \delta_{2y} - \frac{|B|}{2 \Delta y} \delta_{yy} - \frac{C}{\Delta x^2} \delta_{xx} - \frac{D}{\Delta y^2} \delta_{yy} - \frac{E}{4 \Delta x \Delta y} \delta_{2x2y}, \quad (2)$$

where upwinding of the inviscid terms is accomplished by a Roe linearization [36]. Numerical dissipation of this type corresponds to a *matrix dissipation* [37], in which each characteristic field is upwinded by introducing dissipation scaled by the associated characteristic speed. A related form of *scalar dissipation* is obtained by replacing $|A|$ and $|B|$ in (2) by their spectral radii $\rho(A)$ and $\rho(B)$, so that the dissipation for each characteristic is instead scaled by the

maximum characteristic speed [9]. This approach is less expensive since it avoids matrix operations, but it 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 Pr and γ , the four independent parameters that govern the discrete Navier–Stokes residual are the cell Reynolds number, Mach number, cell aspect ratio, and flow angle:

$$\text{Re}_{\Delta x} = \frac{u \Delta x}{\nu}, \quad \text{M} = \frac{\sqrt{u^2 + v^2}}{c}, \quad \frac{\Delta y}{\Delta x}, \quad \frac{v}{u}.$$

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

2.2. 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 [38],

$$\Delta t_{\text{NS}}^{-1} = \Delta t_{\text{H}}^{-1} + \Delta t_{\text{P}}^{-1},$$

where the hyperbolic time step is given by

$$\Delta t_{\text{H}}^{-1} = \frac{1}{\text{CFL}_{\text{H}}} \left(\frac{\rho(A)}{\Delta x} + \frac{\rho(B)}{\Delta y} \right)$$

and the parabolic time step is

$$\Delta t_{\text{P}}^{-1} = \frac{1}{\text{CFL}_{\text{P}}} \left(\frac{4\rho(C)}{\Delta x^2} + \frac{4\rho(D)}{\Delta y^2} + \frac{\rho(E)}{\Delta x \Delta y} \right).$$

The factor of 4 in the parabolic time step arises from considering the worst-case scenario of a checkerboard mode, $W_{i,j} = \widehat{W}(t)e^{i(\pi i + \pi j)}$, for which the coefficients of the second-difference stencil reinforce each other in both directions. The hyperbolic and parabolic CFL numbers, CFL_{H} and CFL_{P} , reflect the extent of the stability region of the Runge–Kutta time-stepping scheme along the imaginary and negative real axes, respectively. In comparison with a uniform global time step, this local stability estimate defines a suitable scalar preconditioner for the Navier–Stokes equations, $P_{\text{NS}}^{-1} = \Delta t_{\text{NS}}^{-1}$, that reduces stiffness resulting from variation in spectral radius and cell size throughout the mesh [11].

For the Euler equations, the corresponding scalar pre-

conditioner is defined by the purely hyperbolic time step, $P_{\text{SE}}^{-1} = \Delta t_{\text{H}}^{-1}$, assuming that the numerical dissipation introduced to prevent oscillations is sufficiently small so as not to limit the stability. The implications of this assumption for the scaling of the numerical dissipation are examined in Section 2.4.

2.3. Matrix Preconditioner

The matrix preconditioner used for the present work is a point-implicit block-Jacobi preconditioner [19, 21] that is obtained from the discrete residual operator (2) by extracting the terms corresponding to the central node in the stencil

$$P_{\text{MNS}}^{-1} = \frac{1}{\text{CFL}_{\text{H}}} \left(\frac{|A|}{\Delta x} + \frac{|B|}{\Delta y} + \frac{2C}{\Delta x^2} + \frac{2D}{\Delta y^2} \right).$$

The corresponding form of the matrix preconditioner for the Euler equations is obtained by eliminating the viscous contributions

$$P_{\text{ME}}^{-1} = \frac{1}{\text{CFL}_{\text{H}}} \left(\frac{|A|}{\Delta x} + \frac{|B|}{\Delta y} \right).$$

It has been demonstrated in Ref. [14] that the preconditioner takes a fundamentally similar form for a 2nd/4th difference switched JST scheme [9] based on the same Roe linearization [36]. This compatibility and related numerical experiments suggest that it is acceptable to base the preconditioner on a first-order discretization even when using higher order switched and limited schemes.

2.4. 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 for the scalar convection–diffusion equation,

$$u_t + au_x = \nu u_{xx},$$

discretized using central differences in space and forward differences in time,

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

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 $\nu = |a|\Delta x/2$. In this case, the stabil-

ity requirement then reduces to the standard CFL condition $\Delta t \leq \Delta x/|a|$.

The corresponding representation of the Euler equations with characteristic-based matrix dissipation is

$$W_t + AW_x = \frac{\Delta x}{2} |A| W_{xx}.$$

The one-dimensional system can be decoupled into scalar characteristic equations,

$$\Omega_t + \Lambda \Omega_x = \frac{\Delta x}{2} |\Lambda| \Omega_{xx},$$

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

The Euler equations with standard scalar dissipation take the form

$$W_t + AW_x = \frac{\Delta x}{2} \rho(A) W_{xx},$$

where $\rho(A)$ 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, $\Delta t_k \leq \Delta x/\rho(A)$. 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 and matrix preconditioner and numerical dissipation, 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.

2.5. Fourier Footprints

To assess the properties of the proposed methods, Fourier analysis is used to decompose the error 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. Numerical results for both the Euler and Navier–Stokes equations indicate that Fourier analysis does provide a useful indicator of scheme performance characteristics.

In the context of a semi-discrete scheme (1), the Fourier footprint of the spatial discretization is critical in revealing the effectiveness of the Runge–Kutta time-stepping scheme in damping and propagating error modes. The footprint is found by substituting a semi-discrete Fourier mode of the form

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

into the discrete residual operator (2). 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:

$$\begin{aligned} Z(\theta_x, \theta_y) = & \hat{i} \frac{A}{\Delta x} \sin \theta_x + \frac{|A|}{\Delta x} (1 - \cos \theta_x) \\ & + \hat{i} \frac{B}{\Delta y} \sin \theta_y + \frac{|B|}{\Delta y} (1 - \cos \theta_y) \\ & + \frac{2C}{\Delta x^2} (1 - \cos \theta_x) + \frac{2D}{\Delta y^2} (1 - \cos \theta_y) \\ & + \frac{E}{\Delta x \Delta y} \sin \theta_x \sin \theta_y. \end{aligned}$$

The Fourier footprint is defined by the eigenvalues of the matrix PZ , which are functions of the Fourier angles θ_x and θ_y . For stability, the footprint must lie within the stability region of the time-stepping scheme specified by $|\psi(z)| \leq 1$, where $|\psi(z)|$ is the amplification factor defined by

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

The stability region and contours for a 5-stage Runge–Kutta scheme due to Martinelli [38] are shown in Fig. 2 to provide a realistic context for 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 time

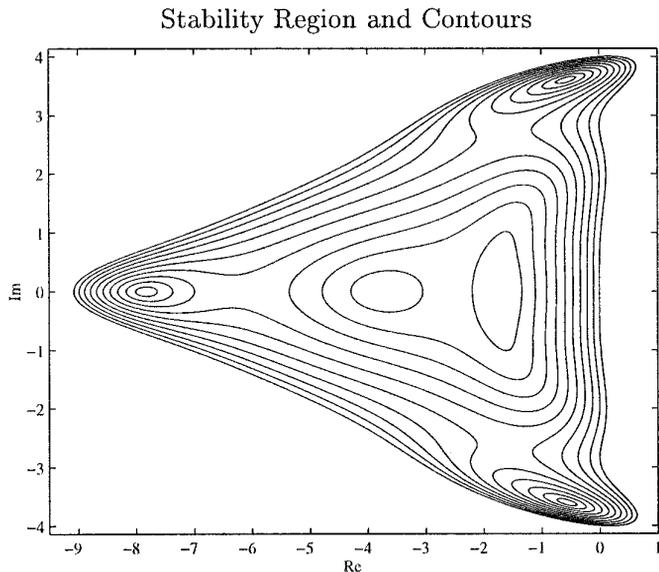


FIG. 2. Stability region and contours defined by $|\psi(z)| = 0.1, 0.2, \dots, 1.0$ for a 5-stage Runge–Kutta time-stepping scheme.

step 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. Since the expression for the group velocity depends on the form of the temporal discretization operator L_t , it is not possible to determine detailed propagative information from the Fourier footprint. However, for Runge–Kutta operators of the type used in the present work, the group velocity corresponding to a given residual eigenvalue is related to the variation in the imaginary components of all the residual eigenvalues in that modal family [39]. Therefore, for rapid propagation, it is desirable for the residual eigenvalues in each family to extend far from the negative real axis.

3. ANALYSIS

3.1. Preconditioned Multigrid for the Euler Equations

For full coarsened multigrid to function efficiently, all modes corresponding to the three shaded Fourier quadrants in Fig. 3 must be damped by the relaxation scheme since only modes which are low frequency in both mesh directions ($L_x L_y$) can be resolved without aliasing on the coarse mesh. Efficient propagation and subsequent ex-

clusion 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 ($H_x H_y$) since point-implicit methods are notoriously poor at eliminating modes with a low frequency component.

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 combinations of preconditioner and numerical dissipation in Fig. 4. The outer solid line in these plots is the stability region of the time-stepping scheme which must contain all the residual eigenvalues to ensure stability. The fact that the maximum extent along the negative real axis is roughly twice the extent in either direction along the imaginary axis suggests the definition $CFL_P = 2CFL_H$, so that only the hyperbolic CFL number need be determined and the subscript may be dropped. 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 [21]. 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.

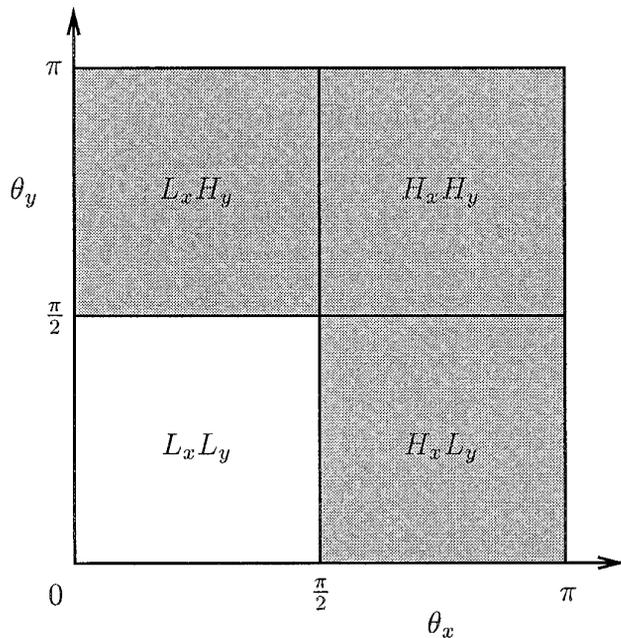


FIG. 3. Fourier quadrants for which the corresponding error modes must be damped for full coarsened multigrid to function efficiently.

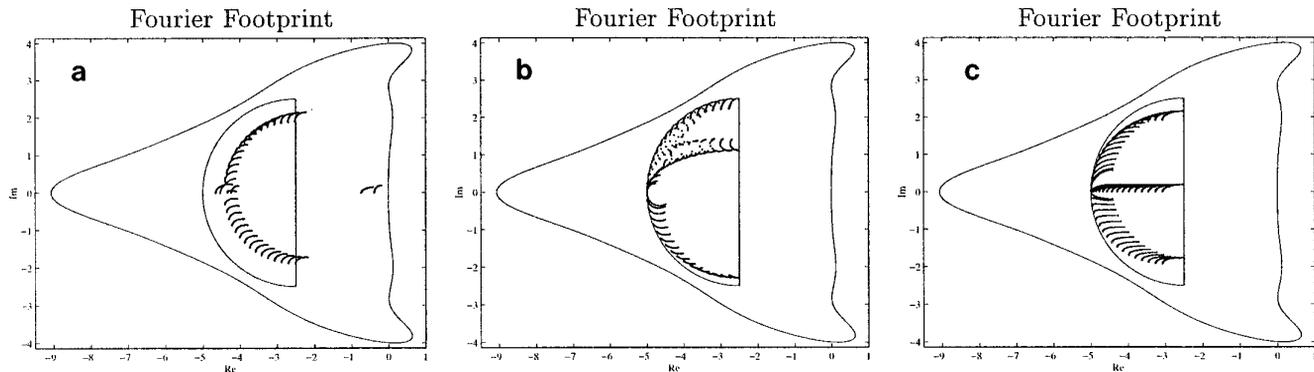


FIG. 4. Preconditioned Fourier footprints for high frequency modes ($H_x H_y$), with $P_S R_M$, $P_M R_M$, and $P_S R_S$. $\text{Re}_{\Delta x} = \infty$, $M = 0.5$, $\Delta y/\Delta x = 1/5$, $v/u = 0$, $\text{CFL} = 2.5$. (a) Scalar preconditioner with matrix dissipation. (b) Matrix preconditioner with matrix dissipation. (c) Scalar preconditioner with scalar dissipation.

Examining Fig. 4a, 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 nearly clustered inside the optimal envelope and will be both rapidly damped and propagated. This behavior reflects the better balance that exists between the magnitude of the scalar time step and the dissipative and propagative coefficients of the acoustic modes.

The matrix preconditioner applied to matrix dissipation ($P_M R_M$) provides optimal damping clustering for all four modes. 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 and are nearly optimally propagated. Only the vorticity footprint, which forms a tongue between the two acoustic footprints does not approach optimal propagative clustering, although 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 time step 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 time step.

The properties of the three combinations of preconditioner and numerical dissipation are summarized in Table I. Only the combination of matrix preconditioning and matrix dissipation is satisfactory on all three counts. Although the $P_S R_S$ combination provides a significant clustering 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.

High and Low Frequency Modes. We have seen that the matrix preconditioner provides excellent damping and propagative clustering for all modes in the $H_x H_y$ quadrant. This result has been shown to hold for a wide range of flow and mesh conditions [21, 22]. 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.

TABLE I

Comparison of Attributes for Different Combinations of Preconditioner and Numerical Dissipation

	$P_S R_M$	$P_M R_M$	$P_S R_S$
Damping		●	●
Propagation		●	
Accuracy	●	●	

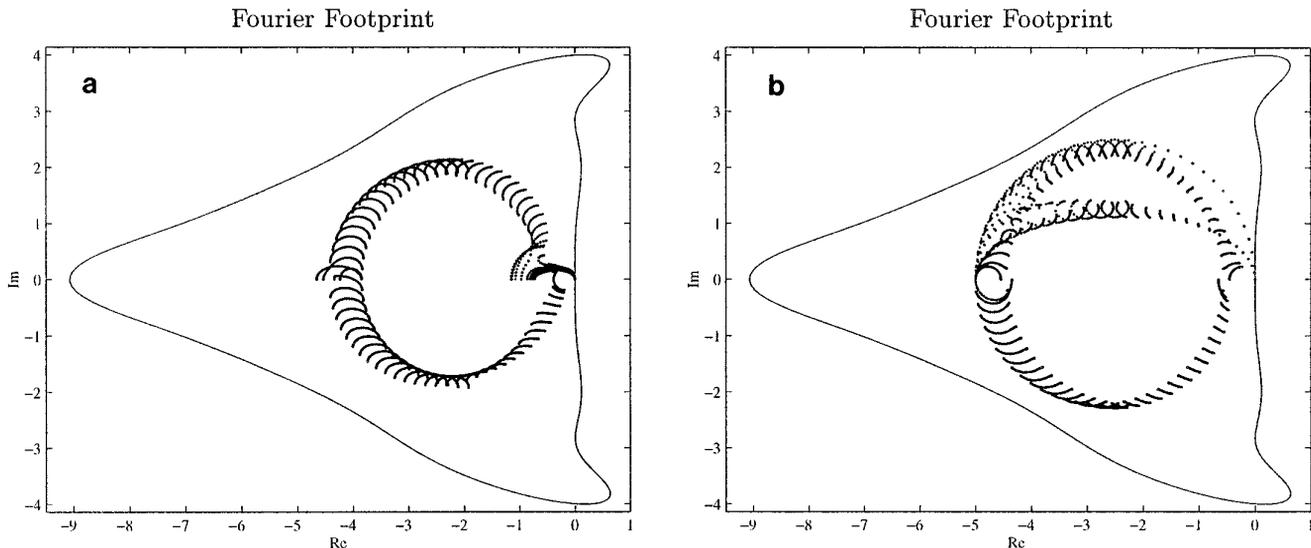


FIG. 5 Preconditioned Fourier footprints for all modes except $L_x L_y$ using first-order upwind matrix dissipation. $Re_{\Delta x} = \infty$, $M = 0.5$, $\Delta y/\Delta x = 1/5$, $\nu/u = 0$, $CFL = 2.5$. (a) Scalar preconditioner. (b) Matrix preconditioner.

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. Propagative disparities are most pronounced near the stagnation point, across the sonic line, and at shocks, while flow alignment results 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 eigenvalues corresponding to the $L_x H_y$ and $H_x L_y$ quadrants away from the origin. However, there is a qualitative difference in the magnitude of this shortcoming, as illustrated by the footprints in Fig. 5 which contain the residual eigenvalues corresponding to all modes except those in the $L_x L_y$ quadrant for the same aligned subsonic flow conditions as were previously considered. Using the scalar preconditioner, the entire footprints of both convective families are densely clustered near the origin so that both damping and propagation are impeded. 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 axis and the origin. As a result, those modes clustered near the origin which cannot be effectively damped will still propagate relatively efficiently in the streamwise direction since the associated group velocity is proportional to the maximum imaginary component achieved by any of the eigenvalues in that family. For typical inviscid computations, the impact on convergence of the few troublesome sawtooth modes that are not well damped using the matrix preconditioner is almost certainly insufficient to warrant

switching from full coarsened multigrid to a more expensive algorithm.

3.2. Preconditioned Multigrid for the Navier–Stokes Equations

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 eigenvalues do not cluster arbitrarily close to the origin where the amplification factor is unity.

The most effective means of understanding the phenomenon of multigrid breakdown is an examination of the form of the preconditioned residual eigenvalues in a highly stretched boundary layer cell. For this purpose, the analytic expressions for the preconditioned Fourier footprints are obtained for the important set of asymptotic limits summarized in Table II. 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. Case 1 represents a stretched cell with perfect flow alignment while Case 2 corresponds 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

TABLE II

Asymptotic Limits for Which Analytic Expressions for the Preconditioned Fourier Footprints of First-Order Upwind Matrix Dissipation Are Obtained

Case E1	$\text{Re}_{\Delta x} = \infty$	$\frac{\Delta y}{\Delta x} \rightarrow 0$	$\frac{v}{u} = 0$
Case E2	$\text{Re}_{\Delta x} = \infty$	$\frac{\Delta y}{\Delta x} \rightarrow 0$	$\frac{v}{u} = \frac{\Delta y}{\Delta x}$
Case NS1	$\text{Re}_{\Delta x} \rightarrow \infty$	$\frac{\Delta y}{\Delta x} = \text{Re}_{\Delta x}^{-1/2}$	$\frac{v}{u} = 0$
Case NS2	$\text{Re}_{\Delta x} \rightarrow \infty$	$\frac{\Delta y}{\Delta x} = \text{Re}_{\Delta x}^{-1/2}$	$\frac{v}{u} = \frac{\Delta y}{\Delta x}$

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

which leads to the relation

$$\frac{\Delta y}{\Delta x} = \text{Re}_{\Delta x}^{-1/2}.$$

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 III 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.

Scalar Preconditioner and Full Coarsened Multigrid. The performance of the standard combination of scalar preconditioning and full coarsened multigrid will first be assessed before examining some alternative strategies. Asymptotic dependence on a Fourier angle amounts to effective damping of modes in that direction, since the corresponding eigenvalues will not be clustered at the origin. Using the scalar preconditioner, the Fourier footprints are identical for all four cases and are displayed in Fig. 6a for all modes except those in the $L_x L_y$ 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. From Table III it is evident that the real and imaginary parts of the acoustic footprints are both dependent on θ_y so that modes with a high frequency component in the y direction will be both effectively damped and propagated. However, acoustic modes that are low frequency in the y direction will be poorly damped, and in the worst case, the eigenvalue for a sawtooth acoustic mode that is constant in the y

direction and high frequency in the x direction will fall exactly on the origin.

The resulting scenario for full coarsened multigrid in combination with scalar preconditioning, which is the strategy in widespread use throughout the CFD community, is illustrated schematically in Fig. 6b. 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 $H_x L_y$ quadrant. It is not surprising that poor convergence is observed when using this algorithm for viscous computations with highly stretched boundary layer cells.

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 III. For the aligned inviscid

TABLE III

Analytic Expressions for the Fourier Footprints of Scalar and Matrix Preconditioners Applied to First-Order Upwind Matrix Dissipation for the Cases Described in Table II

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

Note. The modal families are listed in the order: entropy, vorticity, acoustic, acoustic.

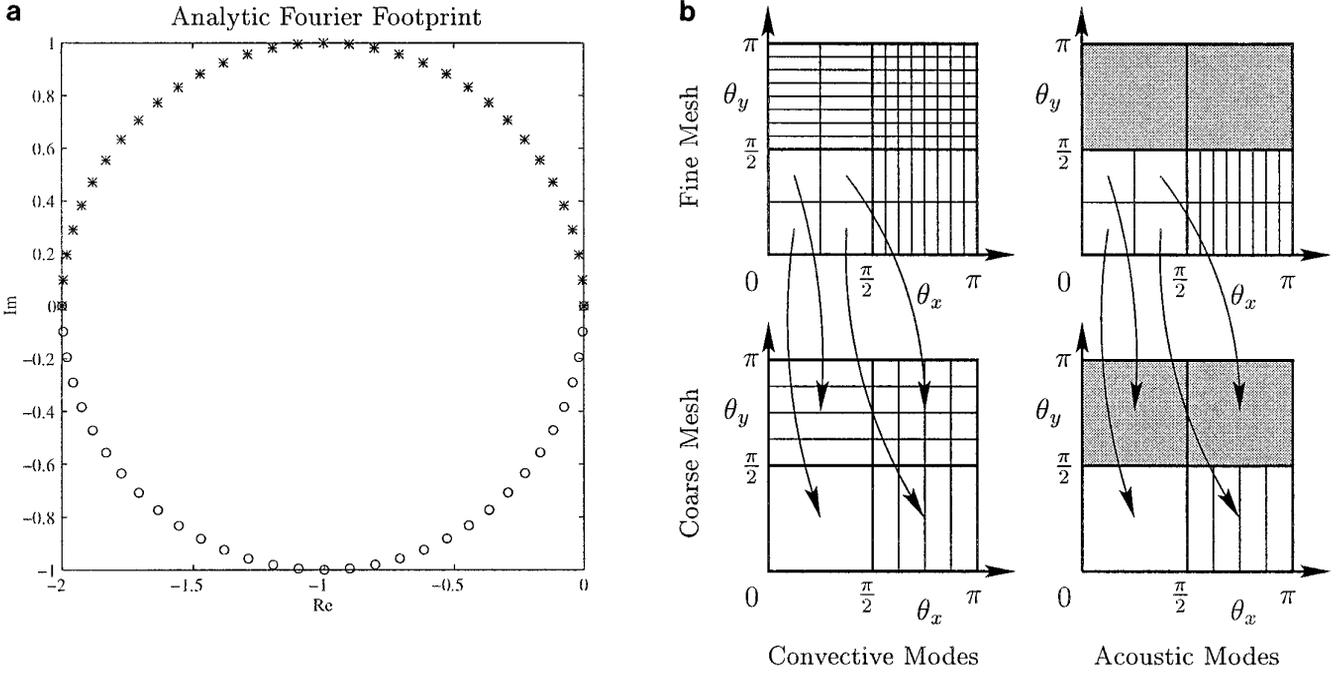


FIG. 6. Clustering performance of the scalar preconditioner and implications for full coarsened multigrid inside a highly stretched boundary layer cell with aligned flow. Footprint symbols: entropy (+), vorticity (\cdot), acoustic (*, \circ). (a) Fourier footprint for all modes except $L_x L_y$. (b) Damping schematic for full coarsened multigrid.

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 effectively damped 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 $L_x L_y$ will be effectively damped. For the entropy family, this property is independent of 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 Case 2 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. 7. The footprint depicts all modes except $L_x L_y$ for the perfectly aligned viscous flow of Case NS1 with $M = 0.04$. This Mach number represents a realistic value for a highly stretched boundary layer cell at the wall, the specific value being observed at the mid-chord for a cell with $y^+ < 1$ in an RAE2822

AGARD Case 6 calculation [14]. Figure 7a 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. 7b. 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 footprints 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 $H_x L_y$ quadrant.

Matrix Preconditioner and J-Coarsened Multigrid. The fact that the block-Jacobi preconditioner provides effective clustering of convective eigenvalues in all but the $L_x L_y$ quadrant provides the freedom to modify the multigrid coarsening strategy with only the damping of $H_x L_y$ 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

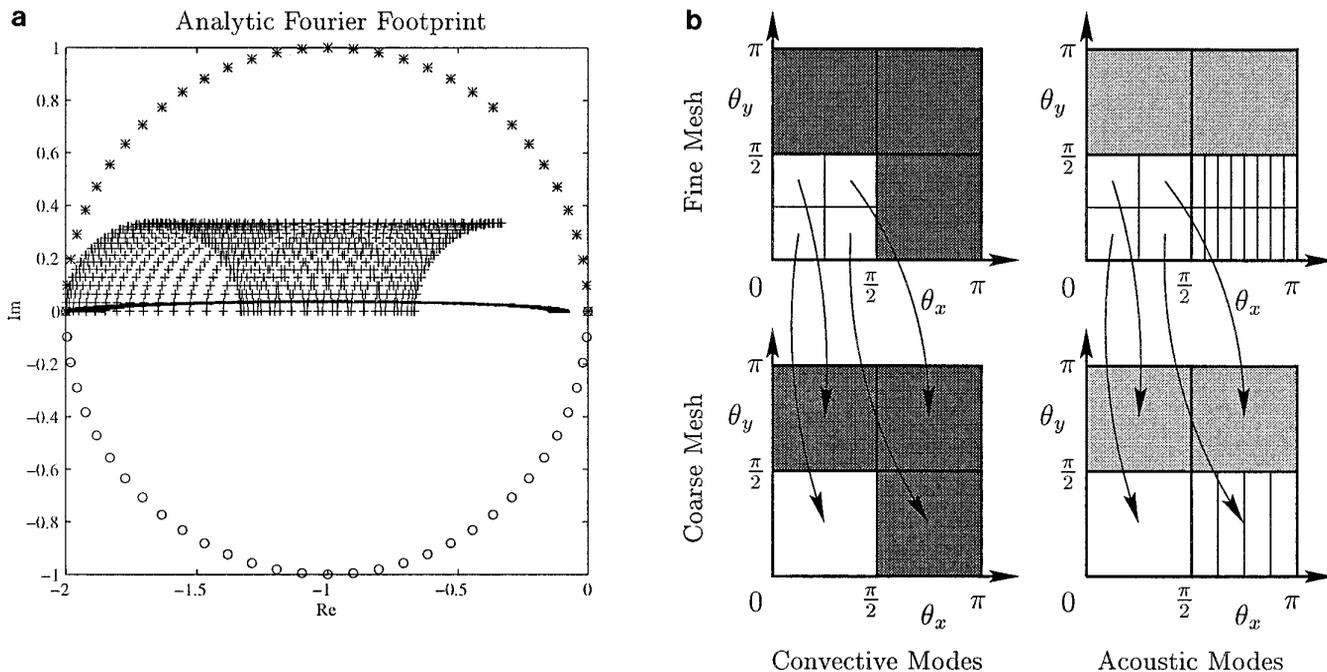


FIG. 7. Clustering performance of the block-Jacobi matrix preconditioner and implications for full coarsened multigrid inside a highly stretched boundary layer cell with aligned flow. Footprint symbols: entropy (+), vorticity (\cdot), acoustic (*, \circ). (a) Footprint for all modes except $L_x L_y$. Case NS1 with $M = 0.04$. (b) Damping schematic for full coarsened multigrid.

J-coarsened strategy in which coarsening is performed only in the direction normal to the wall. The implications for multigrid performance with this approach are summarized in Fig. 8. The Fourier footprint is plotted for the diagonal cross flow 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. 7a. Only residual eigenvalues corresponding to modes in the $L_x H_y$ and $H_x H_y$ Fourier quadrants are displayed in Fig. 8a 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. 8b 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. 7a. 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. 8a. The interaction between

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

Cost bounds for full and J-coarsened cycles are presented in Table IV, where N is the cost of a single flow evaluation on the fine mesh. 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 three dimensions. Use of a J-coarsened W-cycle is inadvisable since the cost depends on the number of multigrid levels (K). 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.

Implementation for structured grid applications is straightforward for single block codes but 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. One means of circumventing this difficulty is to adopt an overset grid approach with interpolation between the overlapping blocks [40]. Since the J-coarsened approach is only beneficial inside the boundary layer, those blocks which are in the inviscid region of the flow should employ a full coarsened strategy, while continuing to use the block-

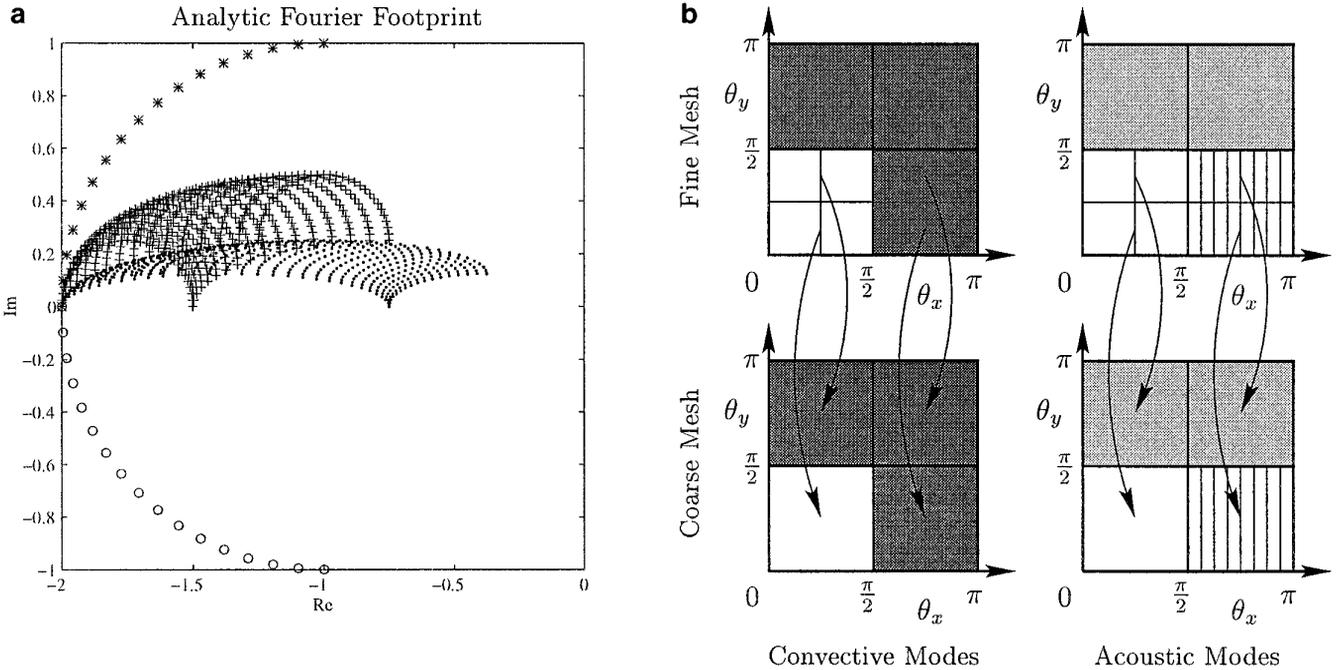


FIG. 8. Clustering performance of the block-Jacobi matrix preconditioner and implications for J-coarsened multigrid inside a highly stretched boundary layer cell with aligned flow. Footprint symbols: entropy (+), vorticity (-), acoustic (*, O). (a) Footprint for $L_x H_y$ and $H_x H_y$ quadrants. Case NS2 with $M = 0.2$. (b) Damping schematic for J-coarsened multigrid.

Jacobi preconditioner for improved eigenvalue clustering [22, 14]. 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 [41, 42] or agglomeration [43, 44] procedures can be employed to provide normal coarsening near the walls and full coarsening in the inviscid regions.

4. IMPLEMENTATION

Basic Discretization. The two-dimensional flow solver developed for the present work is based on a conservative

cell-centered semi-discrete finite volume scheme [9]. Characteristic-based matrix dissipation formed using a Roe linearization [36] provides a basis for the construction of a matrix switched scheme [9, 12]. To achieve the convergence properties demonstrated in the present work, it is critical to employ a viscous flux discretization that does not admit odd/even modes that oscillate between positive and negative values at alternate cells. For this purpose, the compact formulation of Ref. [38] is employed in which the gradients are computed at the midpoint of each face by applying Gauss’ theorem to an auxiliary control volume formed by joining the centers of the two adjacent cells with the end points of their dividing side. Updates are performed using a 5-stage Runge–Kutta time-stepping scheme to drive the multigrid algorithm [9, 7, 38]. For Euler calculations, full coarsened W-cycles are employed with a single time step performed at each level when moving down the multigrid cycle. For turbulent Navier–Stokes calculations, full and J-coarsened V-cycles are employed with a single time step computed at each level when moving both up and down the cycle. The CFL number is 2.5 on all meshes and the switched scheme is used only on the fine mesh with a first-order upwind version of the numerical dissipation used on all coarser meshes.

TABLE IV

Cost Comparison for V and W-cycles Using Full and J-Coarsened Multigrid

2D	Full	J	3D	Full	J
V	$\frac{5}{3}N$	$3N$	V	$\frac{9}{7}N$	$3N$
W	$2N$	KN	W	$\frac{4}{3}N$	KN
IVa: 2D multigrid cost bounds.			IVb: 3D multigrid cost bounds.		

Preconditioner. The 4×4 block-Jacobi preconditioner is computed for each cell before the first stage of each time step. The matrix is then inverted using Gaussian elimina-

tion 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, the 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 inviscid 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.

In the context of preconditioning, an entropy fix serves to prevent the time step from becoming too large near the stagnation point, across the sonic line, at shocks and in the boundary layer. For inviscid calculations, the block-Jacobi preconditioner incorporates the same van Leer entropy fix [45] that is used in the numerical dissipation. When using the block-Jacobi preconditioner on high aspect ratio cells, this approach does not sufficiently limit the time step to provide robustness, so a more severe Harten entropy fix [46] is used in the preconditioner, with the minimum of the bounding parabola equal to one eighth the speed of sound.

Turbulence Models. Both the algebraic Baldwin–Lomax (BL) turbulence model [47] and the one-equation Spalart–Allmaras (SA) turbulence model [48] 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 to drive the same multigrid algorithm as is 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. This solution procedure is very convenient because the turbulent viscosity can be treated in nearly all subroutines as an extra variable in the state vector. The transition point is set using the trip term built into the Spalart–Allmaras model. To prevent the turbulence models from adversely affecting the convergence of the flow equations, it is sometimes beneficial to freeze the turbulent viscosity after a certain initial level of convergence has been achieved. For the results presented in this paper, the only calculation for which this way necessary was for AGARD Case 9 using the standard scheme with the SA turbulence model, when the turbulence field was frozen after the density had converged by four orders of magnitude. All other calculations with either the BL or SA turbulence models converged smoothly to machine accuracy without freezing the turbulent viscosity.

5. RESULTS

This section demonstrates the acceleration provided by the proposed preconditioned multigrid methods (new) in

TABLE V

Euler Test Case Definitions: Airfoil, Free Stream Mach Number, Angle of Attack, Mesh Dimensions, Maximum Cell Aspect Ratio at the Wall

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	0.800	1.25°	320×64	2

comparison to the standard approach (standard) for both Euler and turbulent Navier–Stokes calculations. For the convergence comparisons that follow, the plotted residuals represent the r.m.s. change in density during one application of the time-stepping scheme on the finest mesh in the multigrid cycle.

5.1. Euler Calculations

The Euler test cases are defined in Table V and convergence information is provided in various useful forms in Table VI 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 case is a standard transonic NACA0012 test case with a strong shock on the upper surface and weak shock on the lower surface, for which the computed pressure distribution and convergence histories are shown in Fig. 11. The computation is performed on the 160×32 O-mesh shown in Fig. 9, which provides good near-field resolution for inviscid calculations but does not introduce significant cell stretching, having a maximum cell aspect ratio of only two. Both the new and standard methods converge to machine accuracy with very little degradation in asymptotic convergence relative to the initial rate, requiring approximately 150 and 700 cycles, respectively. As detailed in Table VI, 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 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. 12 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

TABLE VI

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): Multigrid Cycles, Convergence Rate per Cycle, CPU Time, CPU Speedup

	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

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. 13 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 speedups 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. Similar improvements have also been demonstrated using this technique for laminar Navier–Stokes calculations [22, 14]. Ollivier-Gooch has obtained comparable accelerations using the same matrix preconditioner for Euler and laminar Navier–Stokes calculations on unstructured grids [49].

5.2. Turbulent Navier–Stokes Calculations

The turbulent Navier–Stokes test cases used for the present work are defined in Table VII and correspond to

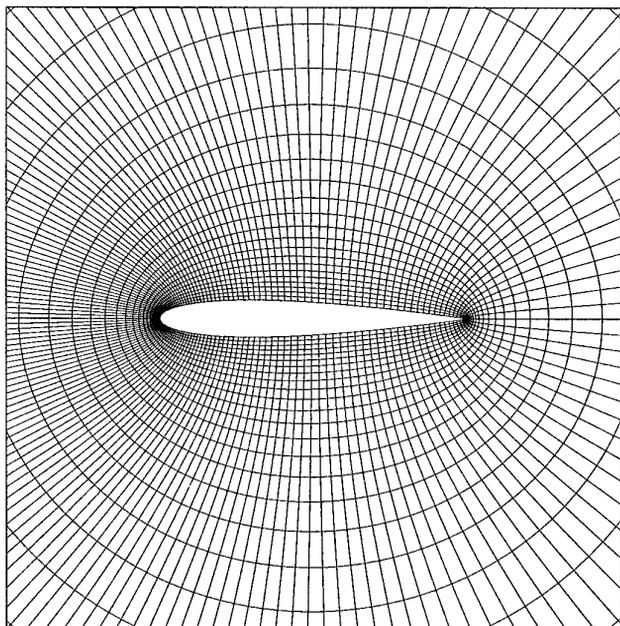


FIG. 9. 160×32 O-mesh for the NACA0012 Airfoil.

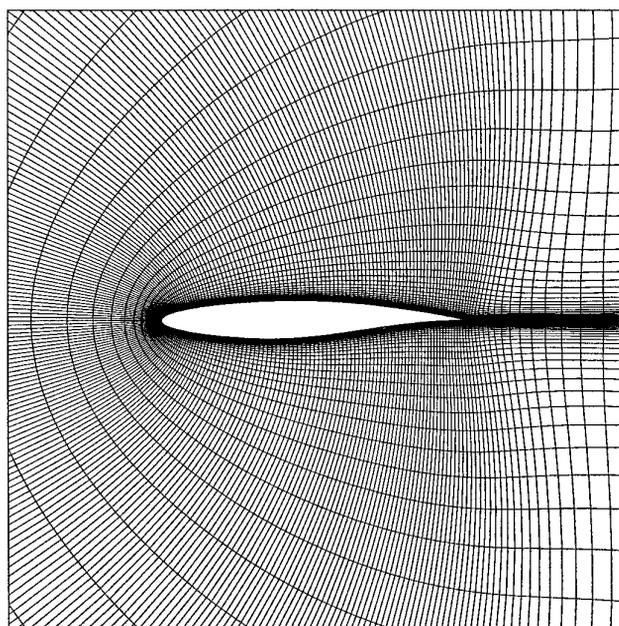


FIG. 10. 288×64 C-mesh for the RAE2822 Airfoil.

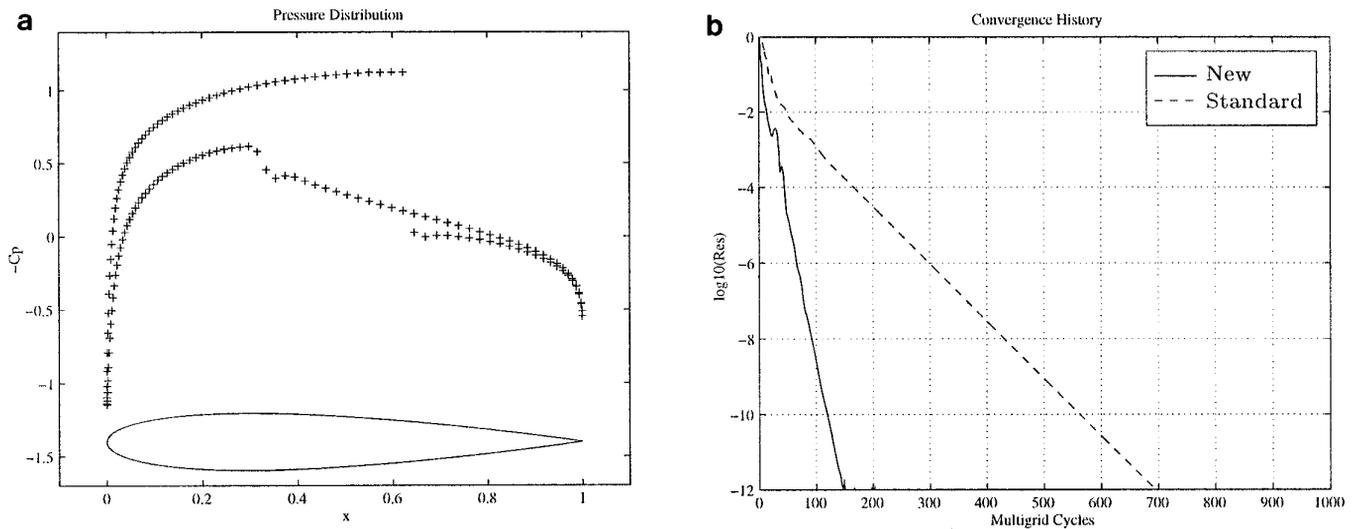


FIG. 11. NACA0012 Airfoil. $M_\infty = 0.8$, $\alpha = 1.25$, 160×32 O-mesh. (a) Coefficient of pressure. $C_l = 0.3527$, $C_d = 0.0227$. (b) Convergence comparison.

RAE2822 AGARD Cases 6 and 9 [50]. Initial and asymptotic convergence information for these calculations is provided in Table VIII. The calculations were performed on a 288×64 C-mesh with 224 cells on the surface of the airfoil as shown in Fig. 10. 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 [48] and Baldwin–Lomax [47] turbulence models are shown in Fig. 14. The computed pressure

distributions compare well with the experimental results [50] as shown in Fig. 14a. The Spalart–Allmaras turbulence model produces a shock somewhat forward of the experimental location as has been previously observed [48].

Convergence of the density and SA turbulent viscosity residuals is shown in Fig. 14b 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

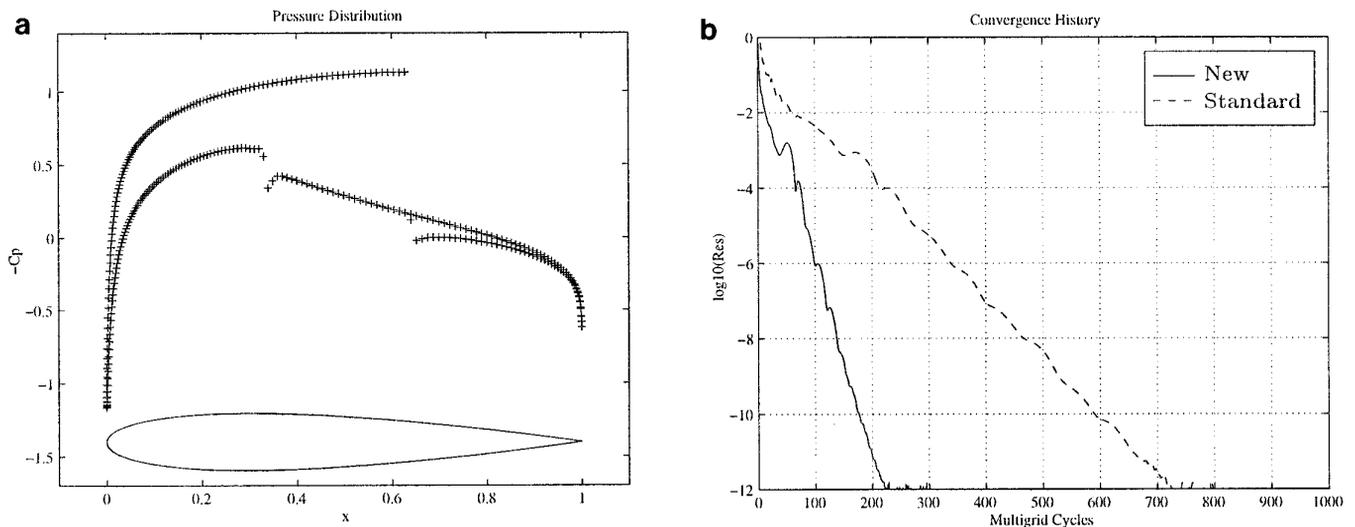


FIG. 12. NACA0012 Airfoil. $M_\infty = 0.8$, $\alpha = 1.25$, 320×64 O-mesh. (a) Coefficient of pressure. $C_l = 0.3536$, $C_d = 0.0225$. (b) Convergence comparison.

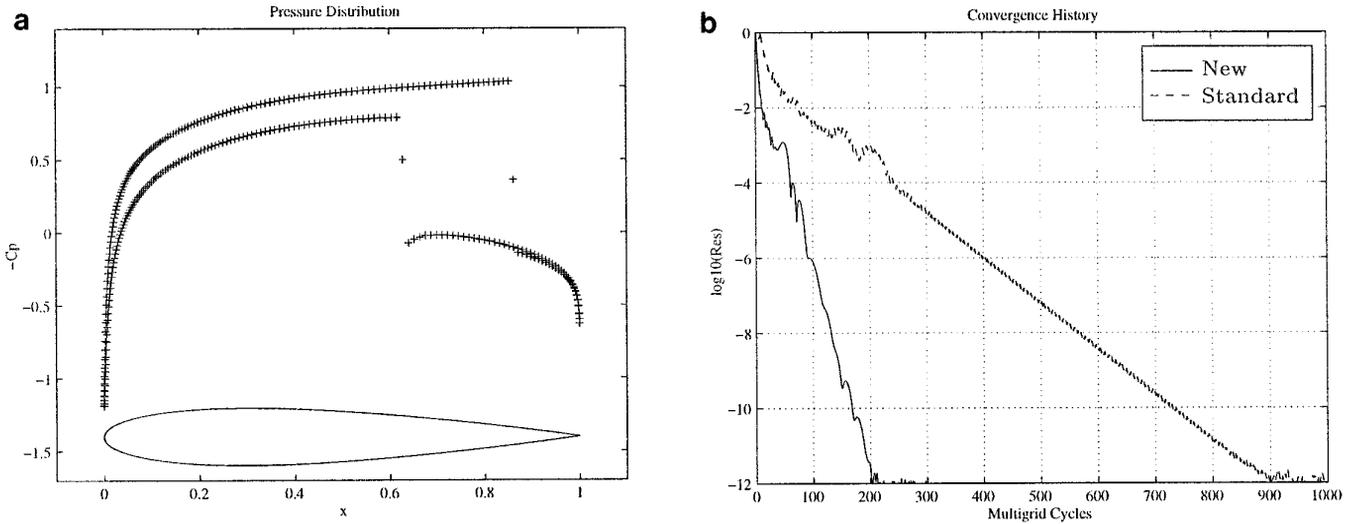


FIG. 13. NACA0012 Airfoil. $M_\infty = 0.85$, $\alpha = 1.0$, 320×64 O-mesh. (a) Coefficient of pressure. $C_l = 0.3721$, $C_d = 0.0572$. (b) Convergence comparison.

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 about 35,000 cycles. From Table VIII it is evident that the new approach converges four orders of magnitude in 113 cycles at a rate of 0.9205 while the standard approach requires 2212 cycles at a rate 0.9958, 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 speedup 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. 14c. These schemes are designated $P_M MG_J$, $P_S MG_J$, $P_M MG_{Full}$, and

$P_S MG_{Full}$, where the first and last combinations correspond to the schemes otherwise referred to as “new” and “standard.” First, it is worth mentioning that overplotting 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.

Returning to the discussion of the four combinations of preconditioners and coarsening strategies, it is evident from Fig. 14c 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 results 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 dominated by the convective modes, while the asymptotic convergence is dominated by the acoustic modes. Reexamining Fig. 14c, 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

TABLE VII

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

Test	Geometry	M_∞	α	Re_L	Mesh	AR_{max}	$y^+_{ave/max}$
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 VIII

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, CPU Time, CPU Speedup

	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

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.

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. 14d. To reach a residual level of 10^{-4} , the new approach (P_MMG_J) requires 114 cycles and 1234.0 s, while the standard approach (P_SMG_{Full}) requires 2262 cycles and 13,310.3 s. The intermediate scheme using scalar preconditioning and J-coarsened multigrid (P_SMG_J) requires 589 cycles and 5664.2 s while the other intermediate scheme using matrix preconditioning and full coarsened multigrid (P_MMG_{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 speedup 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. 15 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 [50]. 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. 15b 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 speedup 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 speedup to 58.45.

6. CONCLUSIONS

Efficient preconditioned multigrid methods were 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 due to the discrete stiffness and directional decoupling introduced by the highly stretched cells in the boundary layer.

For Euler calculations on moderately stretched meshes,

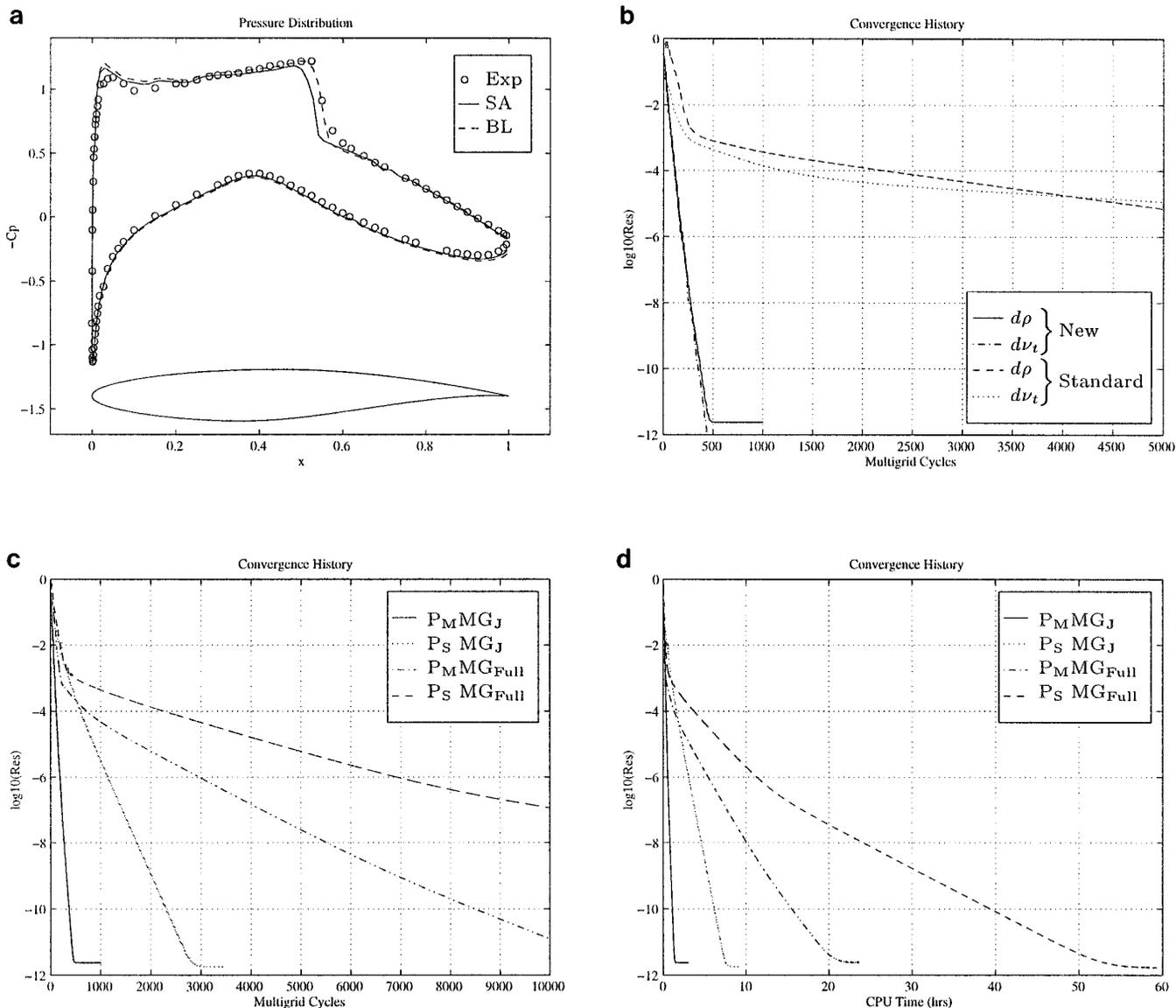


FIG. 14. RAE2822 AGARD Case 6. $M_\infty = 0.725$, $\alpha = 2.4$, $Re = 6.5 \times 10^6$. 288×64 C-mesh. (a) Coefficient of pressure. (b) Convergence comparison using SA model. (c) Convergence comparison using BL model. (d) CPU cost comparison using BL model.

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 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 on highly stretched meshes. The computational savings relative to the standard approach are roughly a factor of 10 for engineering accuracy and a factor of forty in asymptotic performance.

tioned 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 on highly stretched meshes. The computational savings relative to the standard approach are roughly a factor of 10 for engineering accuracy and a factor of forty in asymptotic performance.

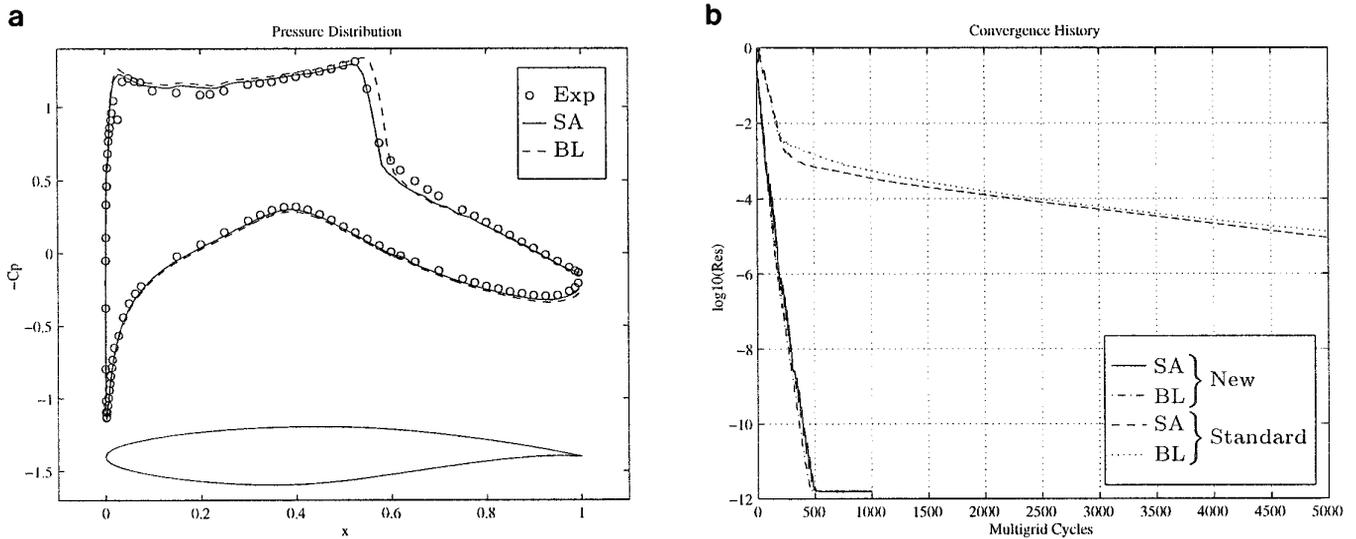


FIG. 15. RAE2822 AGARD Case 9. $M_\infty = 0.73$, $\alpha = 2.79$, $\text{Re} = 6.5 \times 10^6$. 288×64 C-mesh. (a) Coefficient of pressure. (b) Convergence using SA and BL models.

ACKNOWLEDGMENTS

The first author gratefully acknowledges the funding of the Rhodes Trust. The computational facilities provided during the final phase of this research by the CFD Laboratory for Engineering Analysis and Design at Princeton University were also much appreciated.

REFERENCES

- R. P. Fedorenko, The speed of convergence of one iterative process, *Zh. Vychisl. Mat. Mat. Fiz.* **4**(3), 559 (1964). [*USSR Comput. Math. Math. Phys.* **4**, 227 (1964)].
- 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 (1966). [*USSR Comput. Math. Math. Phys.* **6**, 101 (1966)].
- R. A. Nicolaides, On the l^2 convergence of an algorithm for solving finite element equations, *Math. Comp.* **31**, 892 (1977).
- A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comp.* **31**(138), 333 (1977).
- W. Hackbusch, On the multi-grid method applied to difference equations, *Computing* **20**, 291 (1978).
- R.-H. Ni, A multiple-grid scheme for solving the Euler equations, *AIAA J.* **20**(11), 1565 (1982).
- A. Jameson, Solution of the Euler equations by a multigrid method, *Appl. Math. Comput.* **13**, 327 (1983).
- A. Jameson, Multigrid algorithms for compressible flow calculations, in *Second European Conference on Multigrid Methods, Cologne, 1985*, edited by W. Hackbusch and U. Trottenberg, [Lecture Notes in Mathematics, Vol. 1228] (Springer-Verlag, Berlin, 1986), p. 166.
- 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.
- 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. Numer. Math.* (1991).
- C. P. Li, Numerical solution of the viscous reacting blunt body flows of a multicomponent mixture, AIAA Paper 73-202, 1973.
- 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 (1995).
- F. Liu and X. Zheng, A strongly coupled time-marching method for solving the Navier-Stokes and $k-\omega$ turbulence model equations with multigrid, *J. Comput. Phys.* **128**, 289 (1996).
- N. A. Pierce and M. B. Giles, Preconditioning compressible flow calculations on stretched meshes, in *34th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1996*. [AIAA Paper 96-0889]
- A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).
- E. Turkel, Fast solutions to the steady state compressible and incompressible fluid dynamics equations, in *Ninth Int. Conf. Num. Meth. Fluid Dyn.* (Springer-Verlag, New York/Berlin, 1984), p. 571. [Lect. Notes in Physics, Vol. 218].
- H. Viviand, Pseudo-unsteady systems for steady inviscid flow calculations, in *Numerical Methods for the Euler Equations of Fluid Dynamics*, edited by F. Angrand *et al.* (SIAM, Philadelphia, 1985), p. 334.
- 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.
- E. Morano, M.-H. Lallemand, M.-P. Leclercq, H. Steve, B. Stoufflet, and A. Dervieux, Local iterative upwind methods for steady compressible flows, in *Third European Conference on Multigrid Methods, Bonn, 1990* (Springer-Verlag, Berlin, 1991), p. 227. [GMD-Studien Nr. 189, Multigrid Methods: Special Topics and Applications II]
- E. Turkel, Review of preconditioning methods for fluid dynamics, *Appl. Num. Math.* **12**, 257 (1993).
- S. Allmaras, Analysis of a local matrix preconditioner for the 2-D Navier-Stokes equations, in *11th Computational Fluid Dynamics Conference, Orlando, FL, 1993*. [AIAA Paper 93-3330-CP]
- N. A. Pierce and M. B. Giles, Preconditioning on stretched meshes,

- in *12th AIAA CFD Conference, San Diego, CA, 1995*. [Oxford University Computing Laboratory Technical Report 95/10]
23. E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations. *J. Comput. Phys.* **72**, 277 (1987).
 24. Y.-H. Choi and C. L. Merkle, The application of preconditioning in viscous flows. *J. Comput. Phys.* **105**, 207 (1993).
 25. 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 *Frontiers of Computational Fluid Dynamics*, edited by D. A. Caughey and M. M. Hafez, p. 215. 1994.
 26. S. Allmaras, Analysis of semi-implicit preconditioners for multigrid solution of the 2-D compressible Navier–Stokes equations, in *12th Computational Fluid Dynamics Conference, San Diego, CA, 1995*. [AIAA Paper 95-1651-CP]
 27. W. A. Mulder, A new multigrid approach to convection problems, *J. Comput. Phys.* **83**, 303 (1989).
 28. P. E. O. Buelow, S. Venkateswaran, and C. L. Merkle, Effect of grid aspect ratio on convergence, *AIAA J.* **32**(12), 2401 (1994).
 29. S. Venkateswaran, J. M. Weiss, C. L. Merkle, and Y.-H. Choi, Propulsion-related flowfields using the preconditioned Navier–Stokes equation, AIAA Paper 92-3437, 1992.
 30. N. A. Pierce, M. B. Giles, A. Jameson, and L. Martinelli, Accelerating three-dimensional Navier–Stokes calculations, AIAA Paper 97-1953, 1997.
 31. A. Jameson, Aerodynamic design via control theory, *J. Sci. Comput.* **3**, 233 (1988).
 32. A. Jameson, N. A. Pierce, and L. Martinelli, Optimum aerodynamic design using the Navier–Stokes equations, in *AIAA 35th Aerospace Sciences Meeting, Reno, NV, 1997*. [AIAA paper 97-0101] [*J. Theor. Comput. Fluid Mech.*, to appear]
 33. A. Jameson, Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, in *10th Computational Fluid Dynamics Conference, Honolulu, HI, 1991*. [AIAA Paper 91-1596]
 34. N. A. Pierce and J. J. Alonso, A preconditioned implicit multigrid algorithm for parallel computation of unsteady aeroelastic compressible flows, in *35th Aerospace Sciences Meeting and Exhibit, Reno, NV, 1997*. [AIAA paper 97-0444]
 35. N. A. Pierce and J. J. Alonso, Efficient computation of unsteady viscous flows by an implicit preconditioned multigrid method, *AIAA J.*, [in revision].
 36. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
 37. R. C. Swanson and E. Turkel, On central-difference and upwind schemes, *J. Comput. Phys.* **101**, 292 (1991).
 38. L. Martinelli, *Calculations of Viscous Flows with a Multigrid Method*, Ph.D. thesis, Princeton University, 1987.
 39. N. A. Pierce, *Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes*, Ph.D. thesis, Oxford University. [Expected completion September, 1997]
 40. J. A. Benek, P. G. Buning, and J. L. Steger, A 3-D Chimera grid imbedding technique, *7th AIAA CFD Conference, Cincinnati, OH, 1985*. [AIAA Paper 85-1523]
 41. P. I. Crumpton and M. B. Giles, Implicit time accurate solutions on unstructured grids, *12th AIAA CFD Conference, San Diego, CA, 1995*. [AIAA Paper 95-1671]
 42. 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.
 43. W. A. Smith, Multigrid solution of transonic flow on unstructured grids. In O. Baysal, editor, *Recent Advances and Applications Computational Fluid Dynamics*, Nov. 1990. Proceedings of the ASME Winter Annual Meeting.
 44. D. J. Mavriplis and V. Venkatakrishnan, Multigrid techniques for unstructured meshes, ICASE Report No. 95-30, 1995.
 45. B. van Leer, W.-T. Lee, and K. G. Powell, Sonic-point capturing, *AIAA 9th Computational Fluid Dynamics Conference, 1989*. [AIAA Paper 89-1945-CP]
 46. A. Harten, High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **49**, 357 (1983).
 47. B. Baldwin and H. Lomax, Thin layer approximation and algebraic model for separated turbulent flows. AIAA Paper 78-257, 1978.
 48. P. R. Spalart and S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, *Rech. Aerosp.* **1**, 5 (1994).
 49. 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*, Lake Tahoe, NY, 1995, p. 913.
 50. 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.