Accelerating Three-Dimensional Navier–Stokes Calculations

N.A. Pierce[*], M.B. Giles[†], A. Jameson[‡] AND L. Martinelli[§]

*,† *Oxford University Computing Laboratory*
Numerical Analysis Group
Oxford OX1 3QD, UK

‡ *Department of Aeronautics and Astronautics*
Stanford University
Stanford, CA 94305, USA

§ *Department of Mechanical and Aerospace Engineering*
Princeton University
Princeton, NJ 08544, USA

This paper addresses the widely observed breakdown in multigrid performance for turbulent Navier–Stokes computations on highly stretched meshes. Extending previous work in two dimensions, two alternative preconditioned multigrid methods are proposed based on an examination of the analytic expressions for the preconditioned Fourier footprints in an asymptotically stretched boundary layer cell. These methods provide for efficient multigrid performance by ensuring that *all* error modes are effectively damped inside the boundary layer. The schemes also strive to balance the trade-offs between operation count, storage overhead, and parallel scalability. The first of these methods is implemented for the present work and is shown to dramatically accelerate convergence for three-dimensional turbulent Navier–Stokes calculations.

1 Introduction

The development of efficient numerical methods for solution of the Navier–Stokes equations remains one of the enduring challenges in the field of computational fluid dynamics. The difficulty 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. This requirement proves problematic in two regards: a) the resulting high aspect ratio cells greatly reduce the efficiency of existing numerical algorithms, b) the overall increase in the number of mesh points stretches the capability of exist-

ing 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 number of mesh points is already $O(10^6\text{-}10^7)$ and 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 problem size. 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, 2, 3, 4]. Although no rigorous extension of this theory has emerged for problems involving a hyper-

bolic component, methods based on multigrid have proven highly effective for inviscid calculations with the Euler equations [5, 6, 7] and remain the most attractive approach for Navier–Stokes calculations despite the widely observed performance breakdown in the presence of boundary layer anisotropy.

Obtaining a steady state solution by time-marching the unsteady Euler or Navier–Stokes equations requires elimination of transient error modes either by damping or by expulsion from the computational domain. 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 [8]. 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 [9].

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 determination of the relative damping and propagative performances 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. Alternatively, implicit schemes theoretically offer unconditional stability but are 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. For large problems, it therefore seems advantageous to adopt an explicit approach if a scheme with suitable properties can be designed. Alternatively, if efficiency proves elusive using an explicit approach, then a scheme incorporating some implicit properties may appear more attractive.

A popular explicit multigrid smoother is the semi-discrete scheme proposed by Jameson *et al.* [10] which uses multi-stage Runge–Kutta time-stepping to integrate the o.d.e. resulting from the spatial dis-cretization. In accordance with the requirements for efficient multigrid performance, the coefficients of the Runge–Kutta scheme are chosen to promote rapid propagation and damping of error modes [6, 11]. This is accomplished by providing a large stability limit and a small amplification factor in the region of Fourier space where the residual eigenvalues of high frequency modes are concentrated. 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 [6, 7]. 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 Fourier space so that they can be neither damped nor propagated efficiently by the multi-stage relaxation scheme. For Euler computations, discrete stiffness results primarily from the use of a scalar time step, which is unable to cope with the inherent disparity in the propagative speeds of convective and acoustic modes. This problem is relatively localized 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 cir-
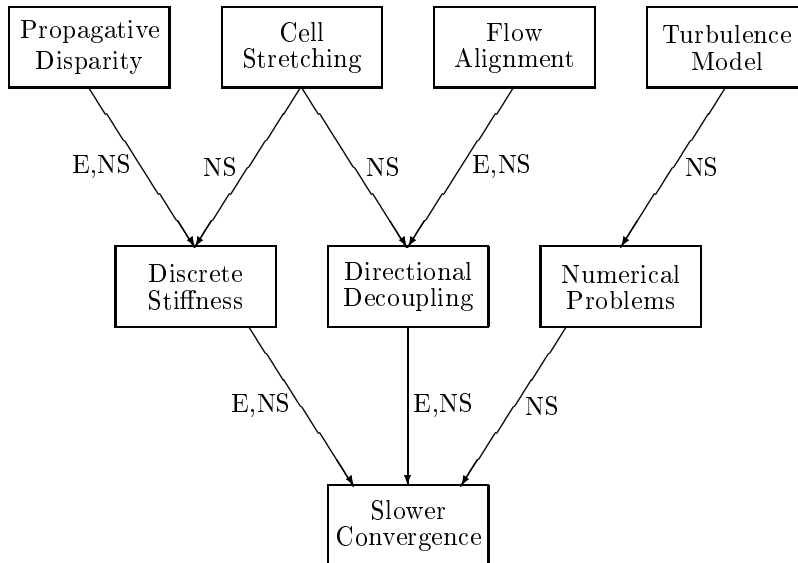
Figure 1: Diagnosis of Multigrid Breakdown for the Euler and Navier–Stokes equations.

cumstances, 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].

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 previously mentioned 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 therefore not be considered in the present work.

One means of combatting discrete stiffness in the Euler and Navier–Stokes equations is the use of a matrix time step or preconditioner [15, 16, 17] that is intended to cluster residual eigenvalues away from the origin into a region of Fourier space for which the multi-stage scheme can provide rapid damping and propagation. In certain cases, preconditioning methods can also be used to alleviate the problem of directional decoupling [17, 14, 9]. Another method for countering directional decoupling is the use of directional coarsening multigrid algorithms [18]. 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 [9, 18]. 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 damp 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 [9].

Alternatively, the semi-coarsening algorithm proposed by Mulder [18] coarsens separately in each

3

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 a point-implicit block-Jacobi preconditioner that has previously been demonstrated to be effective in clustering high frequency eigenvalues away from the origin [16]. For grids that are not coarsened in one of the mesh directions, Allmaras proposes using a semi-implicit line-Jacobi preconditioner in that direction [9].

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 that 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 $O(N)$ [18].

Seeking a less expensive approach to overcoming multigrid breakdown in the presence of boundary layer anisotropy, Pierce and Giles examined the analytic form of the two-dimensional preconditioned Fourier footprints inside an asymptotically stretched boundary layer cell [17, 14, 8]. This analysis revealed the asymptotic dependence of the residual eigenvalues on the two Fourier angles, thus exposing the clustering properties of the preconditioned algorithm. In particular, it was found 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 modes with a low frequency component in one mesh direction are effectively damped [17]. 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 turbu-

lent Navier–Stokes flows, this combination of block-Jacobi preconditioning and J-coarsened multigrid has been demonstrated to yield computational savings of roughly an order of magnitude over existing methods that rely on the standard combination of full coarsened multigrid with a scalar time step [8, 14]. The present work will extend both the theoretical validity and practical demonstration of this approach to three-dimensional turbulent Navier–Stokes calculations.

In three dimensions, the use of a J-coarsened strategy substantially sacrifices the desirable cost bounds of full coarsened multigrid. Therefore, it appears worthwhile to simultaneously 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 the predominate streamwise direction, will also provide effective damping of all error modes. Subsequent investigations may even demonstrate that this assessment is overly conservative and that full coarsening 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. Determination of the approach that best balances all the conflicting demands must await implementation and testing on problems of practical aerodynamic interest.

The present paper therefore analyzes the following two preconditioned multigrid methods:

- point-implicit block-Jacobi preconditioning with J-coarsened multigrid,

- line-implicit J-Jacobi preconditioning with IJ-coarsened multigrid.

The first of these methods has been chosen for implementation in the present work to capitalize on previous experience using this approach in two dimensions [14, 8]. The scheme is found to provide essentially the same rate of convergence for both two and three-dimensional turbulent Navier–Stokes cal-

culations, yielding rapid convergence to machine accuracy and dramatic computational savings over the standard approach employing scalar time-stepping and full coarsened multigrid.

The potential benefits from developing an efficient preconditioned multigrid method for three-dimensional turbulent Navier–Stokes calculations extend far beyond facilitating computation of steady state flows to many other areas of research that rely on a steady state solver as an inner kernel. Notable beneficiaries would include both viscous design using adjoint methods [19, 20] and unsteady simulations based on an inner multigrid iteration [21, 22, 23].

2 Two Dimensions

Before examining the three-dimensional case, the theoretical justifications for the proposed preconditioned multigrid methods will first be described in two dimensions. A portion of this analysis has been documented previously [17, 14, 8] and is included here to provide a basis for discussing the extension of the approach to three dimensions.

2.1 Definitions

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 analytical 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 seem to suggest that Fourier analysis does provide a useful indicator of scheme performance characteristics.

Construction and analysis of the method now proceeds from the 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 scheme appears as

$$L_t W + P R(W) = 0, \qquad (1)$$

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. For the

analysis that follows, R is taken to be a standard linear first order upwind spatial discretization using Roe-averaged characteristic variables [24]

$$
\begin{aligned}
R =\ & \frac{A}{2\Delta x}\delta_{2x} - \frac{|A|}{2\Delta x}\delta_{xx} + \frac{B}{2\Delta y}\delta_{2y} - \frac{|B|}{2\Delta y}\delta_{yy} \\
& - \frac{C}{\Delta x^2}\delta_{xx} - \frac{D}{\Delta y^2}\delta_{yy} - \frac{E}{4\Delta x \Delta y}\delta_{2x2y}.
\end{aligned}
\qquad (2)
$$

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:

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

A Cartesian mesh is assumed to simplify notation, but the theory extends naturally to a (ξ, η) mesh-aligned coordinate system for real applications using structured body-conforming meshes. The first of the proposed methods also extends naturally to unstructured mesh applications.

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 [25],

$$P_S^{-1} = \Delta t_H^{-1} + \Delta t_P^{-1},$$

where the hyperbolic time step is given by

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

and the parabolic time step is

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

The hyperbolic and parabolic CFL numbers, CFL_H and CFL_P, reflect the extent of the stability region of the multi-stage 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, that reduces stiffness resulting from variation in spectral radius and cell size throughout the mesh [26].

Point-Implicit Block-Jacobi Preconditioner

The block-Jacobi preconditioner is based on the form of the discrete residual operator and is obtained by extracting the terms corresponding to the central node in the stencil

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

It has been demonstrated in reference [14] that the preconditioner takes a fundamentally similar form for a 2nd/4th difference switched JST scheme [10] based on the same Roe linearization. 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, in an analogous manner to the practice of using a first order discretization for the Jacobian in implicit methods.

Line-Implicit J-Jacobi Preconditioner

The J-Jacobi preconditioner is obtained by including the complete inviscid and viscous operators in the normal direction but only the component corresponding to the central node for the streamwise direction

$$P_{JJ}^{-1} = \frac{1}{\text{CFL}_H} \left(\frac{|A|}{\Delta x} + \frac{B}{2\Delta y}\delta_{2y} - \frac{|B|}{2\Delta y}\delta_{yy} \right.$$
$$\left. + \frac{2C}{\Delta x^2} - \frac{D}{\Delta y^2}\delta_{yy} \right).$$

It is essential to note that the two matrix preconditioners must be used in conjunction with matrix dissipation to ensure the stability of the scheme. This realization follows from applying the stability condition for a scalar convection–diffusion equation to the individual modal families in the system [14, 8].

Fourier Footprints

In the context of a semi-discrete scheme (1), the Fourier footprint of the spatial discretization is critical in determining the effectiveness of the 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

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

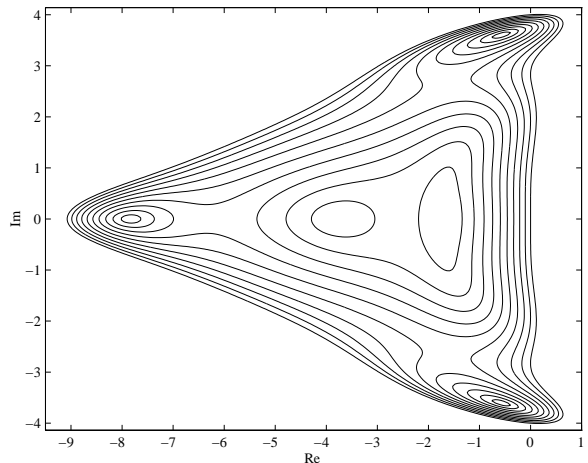
Figure 2: Stability region and contours defined by $|\psi(z)| = 0.1, 0.2, ..., 1.0$ for a 5-stage Runge–Kutta time-stepping scheme.

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 [25] 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 Fourier space where the amplification factor is significantly less than unity. The primary weakness of explicit time integration using a scalar time 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

6

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 [27]. Therefore, for rapid propagation, it is desirable for residual eigenvalues to extend far from the negative real axis. Although it is optimal if modes are both rapidly damped and propagated, when considering the demanding case of highly stretched boundary layer cells, the clustering is deemed successful as long as the eigenvalues do not cluster arbitrarily close to the origin where they can be neither damped nor propagated.

2.2   Analysis

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 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. 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 scaling for the cell aspect ratio is found by balancing 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 2 for 1st 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

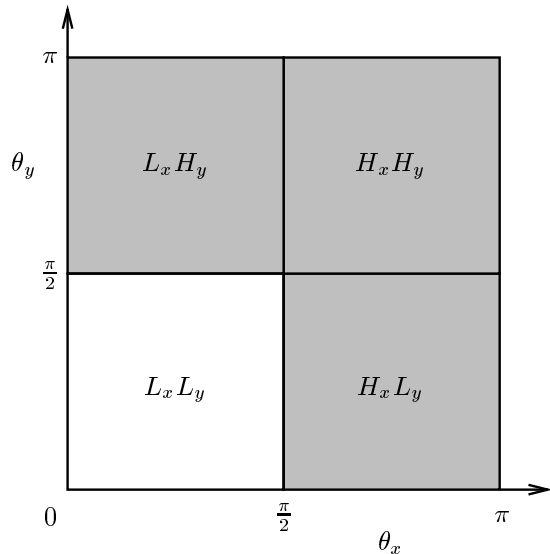The performance of the standard combination of scalar preconditioner (local time step) and full coars-

Figure 3: Fourier quadrants for which the corresponding error modes must be damped or expelled for full coarsening multigrid to function efficiently.

ened multigrid will first be assessed before examining the two proposed methods. For full coarsened multigrid to function efficiently, all modes corresponding to the three shaded Fourier quadrants in Fig. 3 must be either damped or propagated 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 next coarser mesh.

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. 4a 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 2 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 multi-

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 1: Asymptotic limits for which analytic expressions for the preconditioned Fourier footprints of first order matrix dissipation are obtained.

Case	$\mathrm{eig}(P_S Z)$	$\mathrm{eig}(P_{BJ} Z)$	$\mathrm{eig}(P_{JJ} Z)$
E1	0	$C_x + \hat{\imath} s_x$	$C_x + \hat{\imath} s_x$
	0	$C_x + \hat{\imath} M s_x$	$C_x + \hat{\imath} M s_x$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$	1
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$	1
E2	0	$\frac{1}{2}(C_x + C_y) + \frac{\hat{\imath}}{2}(s_x + s_y)$	$\frac{(C_x + C_y) + i(s_x + s_y)}{1 + C_y + \hat{\imath} s_y}$
	0	$\frac{1}{1+M}C_x + \frac{M}{1+M}[C_y + \hat{\imath}(s_x + s_y)]$	$\frac{C_x + M[C_y + \hat{\imath}(s_x + s_y)]}{1 + M(C_y + \hat{\imath} s_y)}$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$	1
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$	1
NS1	0	$\frac{2}{2+Pr}C_y + \frac{Pr}{2+Pr}(C_x + \hat{\imath} s_x)$	$\frac{2}{2C_y+Pr}C_y + \frac{Pr}{2C_y+Pr}(C_x + \hat{\imath} s_x)$
	0	$\frac{1}{1+2M}C_x + \frac{2M}{1+2M}(C_y + \frac{\hat{\imath}}{2}s_x)$	$\frac{1}{1+2MC_y}C_x + \frac{2M}{1+2MC_y}(C_y + \frac{\hat{\imath}}{2}s_x)$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$	1
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$	1
NS2	0	$\frac{1}{1+Pr}C_y + \frac{Pr}{(1+Pr)}[\frac{1}{2}(C_x + C_y) + \frac{\hat{\imath}}{2}(s_x + s_y)]$	$\frac{C_y + Pr\left[\frac{1}{2}(C_x + C_y) + \frac{\hat{\imath}}{2}(s_x + s_y)\right]}{C_y + \frac{Pr}{2}(1 + C_y + \hat{\imath} s_y)}$
	0	$\frac{1}{1+3M}C_x + \frac{3M}{1+3M}[C_y + \frac{\hat{\imath}}{3}(s_x + s_y)]$	$\frac{C_x + 3M\left[C_y + \frac{\hat{\imath}}{3}(s_x + s_y)\right]}{1 + 3M(C_y + \frac{\hat{\imath}}{3}s_y)}$
	$C_y + \hat{\imath} s_y$	$C_y + \hat{\imath} s_y$	1
	$C_y - \hat{\imath} s_y$	$C_y - \hat{\imath} s_y$	1

Table 2: Analytic expressions for the preconditioned Fourier footprints using Scalar, Block-Jacobi and J-Jacobi preconditioners applied to first order upwind matrix dissipation for the cases described in Table 1. The modes are listed in the order: entropy, vorticity, acoustic, acoustic.

Analytic Fourier Footprint
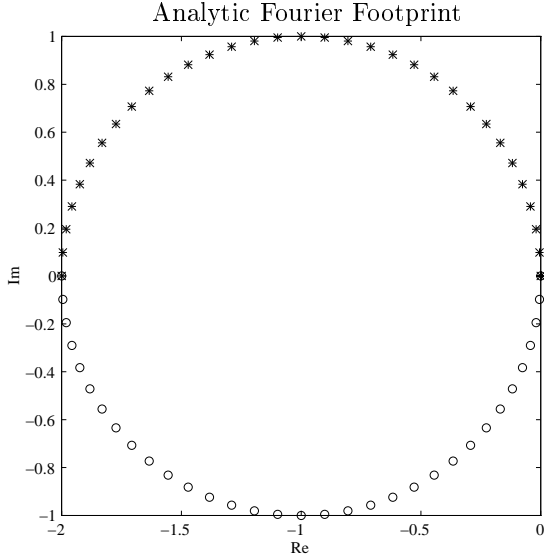
4a: Fourier Footprint for all
modes except $L_x L_y$.

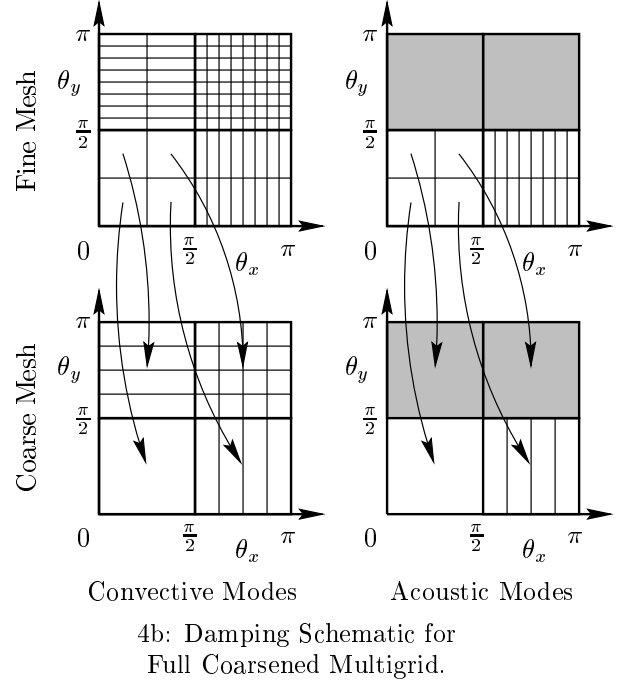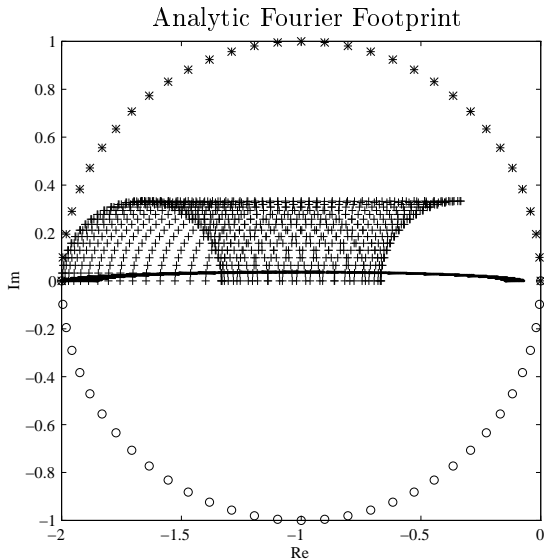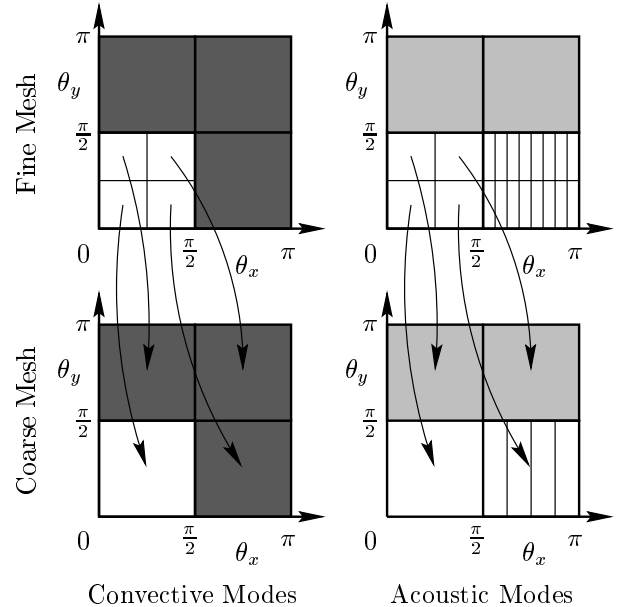4b: Damping Schematic for
Full Coarsened Multigrid.

Figure 4: Clustering performance of the Scalar Preconditioner and implications for Full Coarsened Multigrid inside highly stretched boundary layer cells with aligned flow. Footprint symbols: entropy $(+)$, vorticity $(\cdot)$, acoustic $(*, \circ)$.

Analytic Fourier Footprint

5a: Footprint for all modes except $L_x L_y$.
Case NS1 with $M = 0.04$.

5b: Damping Schematic for
Full Coarsened Multigrid.

Figure 5: Clustering performance of the Block-Jacobi Preconditioner and implications for Full Coarsened Multigrid inside highly stretched boundary layer cells with aligned flow. Footprint symbols: entropy $(+)$, vorticity $(\cdot)$, acoustic $(*, \circ)$.

9

grid in combination with the scalar preconditioner, which is the strategy in widespread use throughout the CFD community, is illustrated schematically in Fig. 4b. 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.

Block-Jacobi Preconditioner and Full Coarsened Multigrid

Developing an understanding for the behavior of the block-Jacobi preconditioner requires a careful examination of the expressions in Table 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 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. 5. 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 [8]. Fig. 5a reveals that the entropy footprint is clustered well away from the origin for all modes. The vorticity mode 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 full coarsening multigrid described in Fig. 5b. 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.
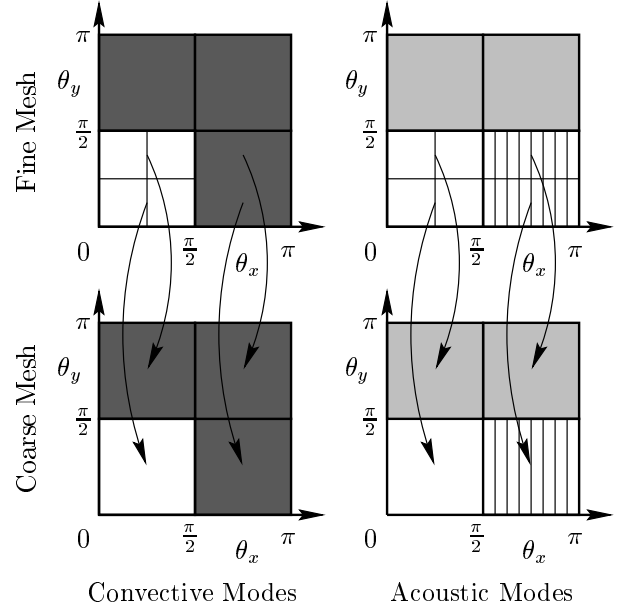
Block-Jacobi Preconditioner and J-Coarsened Multigrid

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 the 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. The implications for multigrid performance with this approach are summarized in Fig. 6. 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. 5a. Only those modes corresponding to the $L_x H_y$ and $H_x H_y$ Fourier quadrants are displayed in Fig. 6a since modes from the other two quadrants can now be resolved on the coarse mesh. The corresponding residual eigenvalues are effectively clustered away from the origin for all families.

The schematic of Fig. 6b 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. For typical turbulent 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. 5a. For most boundary layer cells, the Mach number is large enough that even the vortic-
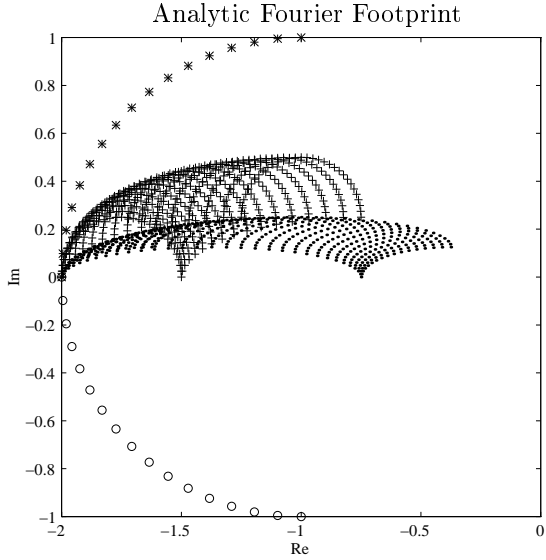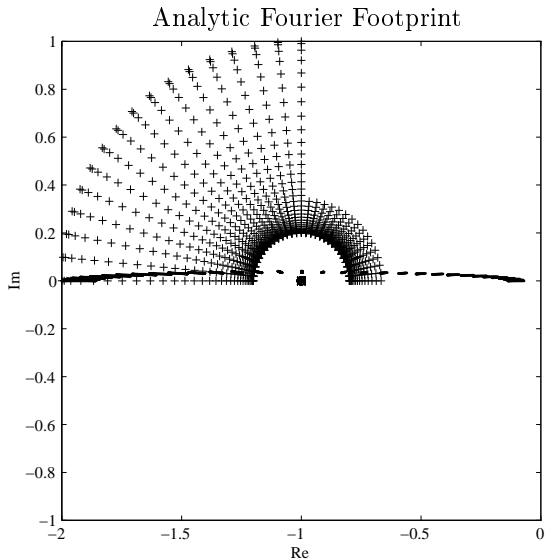
6a: Footprint for L_xH_y and H_xH_y quadrants.
Case NS2 with $M = 0.2$.

6b: Damping Schematic for
J-Coarsened Multigrid.

Figure 6: Clustering performance of the Block-Jacobi Preconditioner and implications for J-Coarsened Multigrid inside highly stretched boundary layer cells with aligned flow. Footprint symbols: entropy ($+$), vorticity ($\cdot$), acoustic ($*$, $\circ$).

7a: Footprint for all modes except L_xL_y.
Case NS1 with $M = 0.04$.
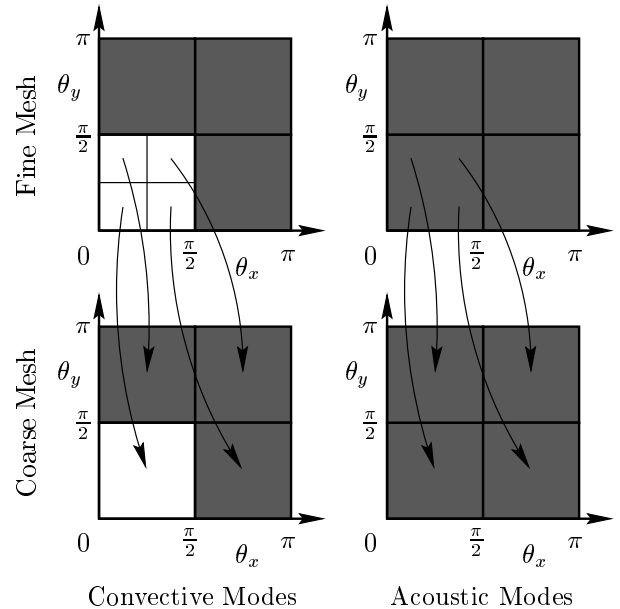
7b: Damping Schematic for
Full Coarsened Multigrid.

Figure 7: Clustering performance of the J-Jacobi Preconditioner and implications for Full Coarsened Multigrid inside highly stretched boundary layer cells with aligned flow. Footprint symbols: entropy ($+$), vorticity ($\cdot$), acoustic ($*$, $\circ$).

ity footprint is clustered well away from the origin and all modes are rapidly damped as in Fig. 6a. The interaction of 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 3, 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 [28]. 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-Jacobi preconditioner for improved eigenvalue clustering [17, 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 search direction since edge collapsing [29, 30] or agglomeration [31, 32] procedures can be employed to provide normal coarsening near the walls and full coarsening in the inviscid regions.

J-Jacobi Preconditioner and Full Coarsened Multigrid

Another attractive alternative that avoids the increased cost bounds of J-coarsened multigrid is a combination of line-implicit J-Jacobi preconditioning with full coarsened multigrid. The general intention of this approach is that the preconditioner should continue to couple the convective families to both Fourier angles while simultaneously providing for effective damping of all acoustic modes including those that are low frequency in θ_y. From the expressions in Table 2, it is evident that the basic structure of the convective footprints is very similar to the case of block-Jacobi preconditioning. The only notable difference is the introduction of terms dependent on the Fourier angles in the denominator. The impact of the J-Jacobi preconditioner on the acoustic footprints is far more dramatic. In highly stretched cells aligned with the flow, the acoustic footprints are now independent of both Fourier angles and are clustered well away from the origin.

The Fourier footprint produced by J-Jacobi preconditioning for the challenging flow conditions of Case NS1 with $M = 0.04$ is shown in Fig. 7a. The clustering of the vorticity eigenvalues is nearly identical to that obtained when using the block-Jacobi preconditioner, as recalled from Fig. 5a. The shape of the entropy envelope is substantially altered, but still corresponds to both efficient damping and propagation. The most notable difference is the collapse of the acoustic envelopes to a point on the negative real axis, where they are still efficiently damped, but can no longer be propagated.

The performance of full coarsened multigrid using J-Jacobi preconditioning is illustrated in Fig. 5b. As demonstrated by the dark shading, the matrix preconditioner is now responsible for all damping during the multigrid cycle. Use of the J-Jacobi preconditioner increases the cost of relaxation and the amount of required storage, but provides the significant benefit of returning to a full coarsened strategy, with a corresponding reduction in the cost bound for each multigrid cycle. Furthermore, the use of a line-implicit preconditioner normal to the wall need not impair parallel efficiency. The J-Jacobi approach is only appropriate inside the highly stretched bound-

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

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

IIIa: 2D Multigrid Cost Bounds. IIIb: 3D Multigrid Cost Bounds.

Table 3: Cost comparison for V and W-cycles using Full and J-Coarsened Multigrid.

ary layer cells so there is no need to extend the preconditioner beyond those blocks that are adjacent to a solid boundary. In the regions of the flow where cell stretching is not severe and the flow is primarily inviscid, the block-Jacobi preconditioner performs very efficiently in combination with full coarsened multigrid [17, 14].

Preliminary efforts to implement this approach in two dimensions appear to reveal an instability resulting from the use of higher order dissipation on the fine mesh. A more thorough investigation will be required to either verify or eliminate this behavior.

3   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 eliminated. 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. 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 [33]. Side-stepping 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 a reasonable assumption for the cases previously examined if $\Delta z \geq O(\Delta x)$ and the flow is aligned with the x direction.

Block-Jacobi Preconditioning and J-Coarsened Multigrid

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 both appear identical to the two-dimensional schematic of Fig. 6. 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. 8 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 eigenvalues corresponding to convective modes in the $L_x L_y H_z$ octant are assumed not to be dependent on θ_z. However, using a J-coarsened strategy, these modes can be transferred to a coarser mesh where they become $L_x H_y H_z$ modes which can be effectively damped. The coarsening strategy therefore assumes the extra responsibility of eliminating some of the convective modes in addition to all of the acoustic modes with a low frequency component in θ_y.

In three dimensions, the point-implicit block-Jacobi preconditioner requires solution of a 5×5 system for each mesh cell at each stage of the Runge–Kutta scheme. The operation count is minimized by computing the LU decomposition of the matrix before the first stage and then storing the resulting 25 three-dimensional arrays for back substitution during each stage. By avoiding a globally implicit approach, the algorithm leads to efficient parallel implementations so that the increase in memory overhead is mitigated by the suitability for distributed memory architectures. 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 using full coarsening in three dimensions. Therefore, it is worth considering an alternative approach that preserves the damping properties of the scheme while recovering the low multigrid cost bounds of a full coarsened approach.

J-Jacobi Preconditioning and IJ-Coarsened Multigrid

Assuming that there is no asymptotic dependence of the residual eigenvalues on θ_z, the three-dimensional damping schematic for this approach can again be constructed by taking slices for low and high values of θ_z which will appear identical to the corresponding two-dimensional schematic of Fig. 7b. As with the previous case, the convective modes in the $L_x L_y H_z$ octant will not be damped by the J-Jacobi preconditioner. Instead, these modes must be passed to the coarse mesh, which effectively prohibits coarsening in the K mesh direction. In three dimensions, the J-Jacobi preconditioner therefore requires an IJ-coarsened strategy rather than the full coarsened strategy of two dimensions. This situation still represents a substantial improvement over the multigrid cost bounds of the J-coarsened approach since IJ-coarsening costs the same as full coarsening in two

8a: Damping Schematic for Low θ_x.
$(0 \le \theta_x \le \frac{\pi}{2})$.

8b: Damping Schematic for High θ_x.
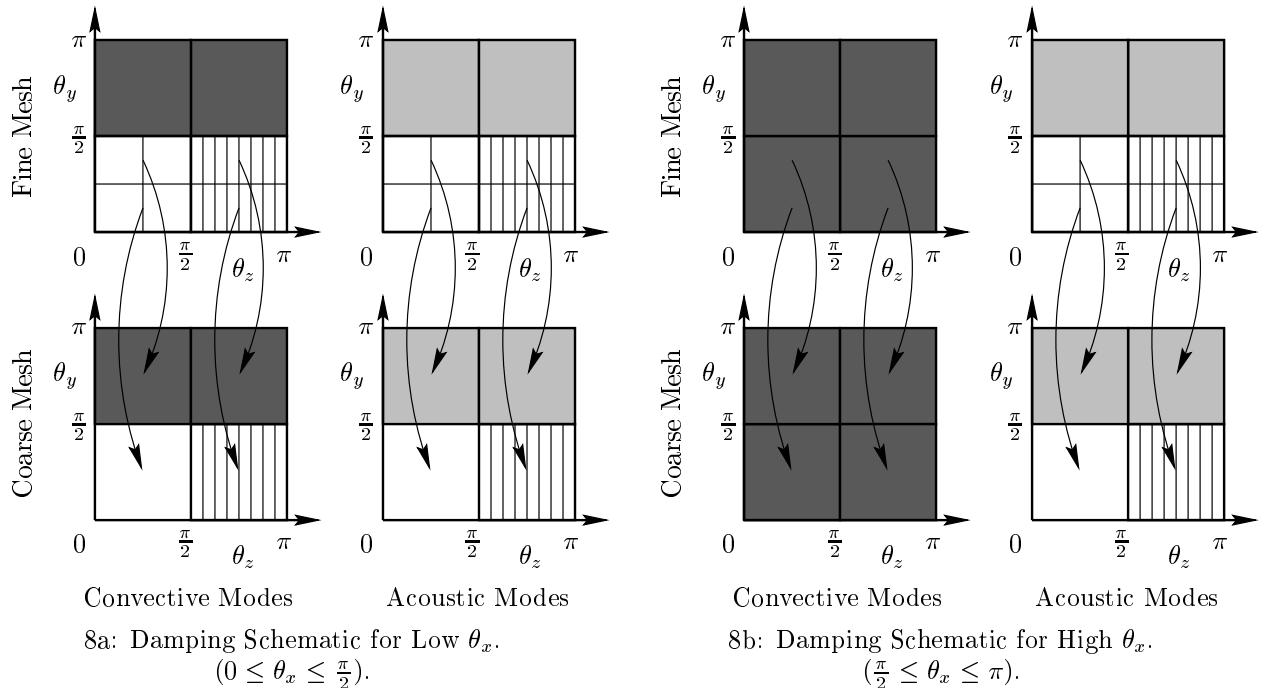$(\frac{\pi}{2} \le \theta_x \le \pi)$.

Figure 8: Implications for 3D performance of Block-Jacobi Preconditioning with J-Coarsened Multigrid inside highly stretched boundary layer cells with aligned flow.

dimensions.

The following argument suggests that this viewpoint may be overly pessimistic and that full coarsened multigrid may be acceptable in combination with J-Jacobi preconditioning. Consider a mesh cell on the surface of a wing in which x is parallel to the chord, y is normal to the surface and z points along the span. The solid wall boundary condition at $y = 0$ effectively enforces flow tangency with the y faces of the cell so that it is critical to account for the possibility of perfect flow alignment with these faces, as reflected in the asymptotic cases previously examined. However, there is no similar boundary condition (except at the intersection with the body) which enforces tangency to the z faces of the cell, so that flow alignment with these faces is unlikely rather than expected. In general, there will be some side slip in the spanwise direction which will introduce a dependence on θ_z into the convective eigenvalues. As a result, convective modes from the $L_x L_y H_z$ octant may then be efficiently damped by the relaxation scheme so that full coarsened multigrid again becomes viable. Alternatively, by increasing the spanwise resolution of the mesh so that $\Delta z = O(\Delta y)$, the viscous diffusion will cause the convective modes to couple in both the y and z directions, once again

leading to efficient damping in the $L_x L_y H_z$ octant.

In three dimensions, the J-Jacobi preconditioner requires solution of a block-tridiagonal system at each stage of the Runge–Kutta scheme. The LU factors may be computed before the first stage and the 75 three-dimensional arrays stored for back substitution in later stages. Alternatively, the preconditioner may be formed and factored at every stage to avoid the storage overhead. However, since the J-Jacobi preconditioner will only be used in blocks adjacent to solid boundaries, and the multigrid storage requirement is reduced in comparison to a J-coarsened approach, it will likely prove most advantageous to store the factored preconditioner to reduce the operation count.

4   Implementation

This section briefly documents the numerical methods implemented in the present work. Only those techniques which are non-standard will be treated in detail.

Basic Discretization

The baseline code used as a starting point for the present work is the well-validated 3D viscous flow solver FLO107 written by Jameson and Martinelli [34]. The discretization is based on a conservative cell-centered semi-discrete finite volume scheme. For the present work, characteristic-based matrix dissipation based on a Roe linearization [24] provides a basis for the construction of a matrix switched JST scheme [10, 12]. Updates are performed using a 5-stage Runge–Kutta time-stepping scheme to drive a V-cycle with time steps computed moving both up and down the cycle [10, 6, 25]. 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.

Preconditioner

The 5×5 block-Jacobi preconditioner is computed for each cell before the first stage of each time step, and then stored as LU factors for inexpensive back substitution during each stage of the Runge–Kutta scheme. 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. When using the block-Jacobi preconditioner on high aspect ratio cells, the van Leer entropy fix [35] used in the numerical dissipation does not sufficiently limit the time step to provide robustness, so a more severe Harten entropy fix [36] is used in the preconditioner, with the minimum of the bounding parabola equal to one eighth the speed of sound. This choice of bounding constant has proven suitable for both 2D and 3D calculations [8].

Turbulence Models

Both the algebraic Baldwin–Lomax turbulence model [37] and the one-equation Spalart-Allmaras turbulence model [38] 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 that 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, either by blinking in the case of the BL model, or by hanging in the case of the SA model, the turbulent viscosity field is frozen after the r.m.s. density residuals have decreased by four orders of magnitude. In a few of the cases, the turbulence field had to be frozen after only three orders.

Viscous Discretization

The baseline version of the viscous flux discretization in FLO107 calculates 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. 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 alternate 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 CFL 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 surfaced with the viscous fluxes in the SA turbulence model, which failed to converge when using this discretization.

One option is to accept the additional cost of calculating the velocity gradients at the cell faces, which produces a compact discretization stencil that does not admit odd/even modes. An alternative approach, which was proposed by Jameson and Caughey in developing a discretization for the transonic potential equation [39] and later suggested for use with the Navier–Stokes equations by Liu and Zheng [13], is to add a correction stencil to the velocity gradients calculated at the vertices 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 cell-faced 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 body-conforming computational coordinates (ξ, η, ζ) then produces the approximate relationship for the gradient at a ξ face

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

where $S_{\xi x}$ represents the projection of the ξ face in the x coordinate direction and Vol is the average volume at the cell 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
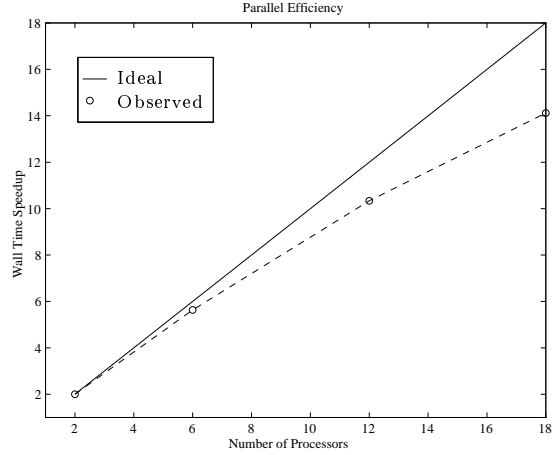

Figure 9: Parallel efficiency for a 288×64×16 mesh.

the correction stencil is applied before assembling the stress components. This method is relatively inexpensive compared to computing the velocities at the cell faces and provides a good approximation to the compact stencil which does not allow odd/even grid point decoupling. In fact, running side by side with the first author's two-dimensional flow solver, which directly computes the viscous terms using the compact formulation as in reference [25], 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 exactly the same convergence history.

Parallelization

The code is parallelized using the MPI (Message Passing Interface) library to implement a domain decomposition approach using a SPMD (Single Program Multiple Data) paradigm [40]. Halo data is exchanged between neighboring processors after every stage on all meshes so that the convergence of the parallel code is identical to that of the serial version.

The parallel scalability of the approach is demonstrated in Fig. 9 for 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.

Wing	Section	AR	Sweep	Twist	Dimensions	$y^+_{\mathrm{ave/max}}$	M_∞	α	Re_L	Spanwise b.c.
Straight	RAE2822	4.0	$0.0°$	$0.0°$	$288\times64\times16$	1.02/2.12	0.725	$2.4°$	6.5×10^6	Periodicity
Swept	RAE2822	4.0	$30.0°$	$0.0°$	$288\times64\times16$	1.02/2.12	0.800	$2.8°$	6.5×10^6	Periodicity
Twisted	RAE2822	4.0	$0.0°$	$2.0°$	$288\times64\times16$	1.02/2.12	0.725	$3.4°$	6.5×10^6	Periodicity
Fully-3D	RAE2822	8.0	$30.0°$	$3.5°$	$288\times64\times16$	1.02/2.12	0.800	$4.0°$	6.5×10^6	Symmetry

Table 4: Test case definitions: wing name, airfoil section, aspect ratio, sweep, twist, mesh dimensions, average and maximum y^+ at the first cell height, free stream Mach number, angle of attack, Reynolds number, and type of boundary condition in the spanwise direction.

5   Results

This section will demonstrate the acceleration provided by the proposed preconditioned multigrid method of block-Jacobi preconditioning and J-coarsened multigrid (New) compared to the standard combination of scalar preconditioning and full coarsened multigrid (Standard). The following series of test cases are chosen so as 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,

- 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 4. The first case is essentially a two-dimensional flow corresponding to AGARD Case 6 [41] and provides an opportunity for direct comparison with the first author's two-dimensional flow solver [14, 8]. The second case introduces a constant spanwise velocity into the flow and the third cases 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.

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. Convergence information for all the test cases is provided in various useful forms in Table 5 for the initial convergence rate between residual levels of 10^0 and 10^{-4} and in Table 6 for the asymptotic rate between residual levels of 10^{-4} and 10^{-8}. 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 would continue to occur beyond this point in the convergence history.

Straight Wing

The initial convergence rates for the density and turbulent viscosity using the Spalart–Allmaras turbulence model are displayed in Fig. 10a. 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 5, 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 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 time. A wall time cost comparison for the present calculations is shown in Fig. 10b. 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 us-

Wing	Turb Model	Cycles		Rate		Wall Time (s)		Cost Ratio
		Standard	New	Standard	New	Standard	New	
Straight	SA	2120	109	.9957	.9187	44,188	5000	8.84
Straight	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

Table 5: Comparison of initial convergence ($10^0 \rightarrow 10^{-4}$) for scalar preconditioning with full coarsened multigrid (standard) vs. block-Jacobi preconditioning with J-coarsened multigrid (new): multigrid cycles, convergence rate per cycle, wall time, wall time speed-up.

Wing	Turb Model	Cycles		Rate		Wall Time (s)		Cost Ratio
		Standard*	New	Standard*	New	Standard*	New	
Straight	SA	9013	199	.9990	.9548	186,780	9185	20.34
Straight	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 6: Comparison of asymptotic convergence ($10^{-4} \rightarrow 10^{-8}$) for scalar preconditioning with full coarsened multigrid (standard) vs. block-Jacobi preconditioning with J-coarsened multigrid (new): multigrid cycles, convergence rate per cycle, wall time, wall time speed-up. *Extrapolated based on first 500 cycles after reaching 10^{-4}.

ing the 3D code is compared to the convergence history on the same mesh using a 2D code in Fig. 11a. 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. The corresponding solutions are displayed in Fig. 11b where it is evident that the two implementations yield almost identical results for the SA model, producing a shock somewhat forward of the experimental location as has been previously observed [38]. The two implementations of the BL model yield slightly different shock locations which are both in better agreement with the experiment. 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. 12. 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 rate 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. 13 for a calculation employing the Spalart–Allmaras turbulence model. This test case produces pressure distributions at angles of attack ranging between ±1° 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 cycle to reach four orders compared to 2113 using the standard approach, which represents a computational savings of 8.17. The computational savings in terms of asymptotic convergence rate is 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 by using a constant planform with a linear twist distribution along the span. The pressure distributions at six stations along the wing are shown in Fig. 14. 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 [41].

The convergence results for the new and standard methods are displayed in terms of multigrid cycles and wall time in Figs 15a and 15b. 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 order require only an additional 213 cycles at a rate of .9575. As for the periodically twisted case, 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 of convergence at a rate of .9984 with 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 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 lead to mesh singularities that do not fall within the assumptions of the theoretical analysis used to motivate this approach.

6   Conclusions

Two preconditioned multigrid methods are proposed that are designed to perform efficiently for three-dimensional turbulent Navier–Stokes computations on highly stretched meshes

- point-implicit block-Jacobi preconditioning with J-coarsened multigrid,

- line-implicit J-Jacobi preconditioning with IJ-coarsened multigrid.

The properties of these schemes are illustrated by examining analytic expressions for the preconditioned Fourier footprints in the asymptotic limit of high cell Reynolds number and aspect ratio to reveal that all convective and acoustic modes are effectively eliminated inside the boundary layer.

The first of these methods is implemented in the present work, and is shown to provide rapid convergence to machine accuracy for three-dimensional turbulent Navier–Stokes calculations. The computational savings over the standard combination of scalar preconditioning and full coarsened multigrid are roughly an order of magnitude to achieve engineering accuracy and much larger in terms of asymptotic performance.

Acknowledgements

References

[1] R.P. Fedorenko. The speed of convergence of one iterative process. *Zh. vych. mat.*, 4(3):559–564, 1964. *(USSR Comp. Math. and Math. Phys.*, 4:227-235, 1964).

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

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

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

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

[6] A. Jameson. Solution of the Euler equations by a multigrid method. *Applied Math. and Computation*, 13:327–356, 1983.

[7] A. Jameson. Multigrid algorithms for compressible flow calculations. In *Second European Conference on Multigrid Methods*, 1985.

[8] N.A. Pierce and M.B. Giles. Preconditioned multigrid methods for compressible flow calculations on stretched meshes. Submitted to *J. Comp. Phys.*, April, 1996.

[9] 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.

[10] 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.

[11] 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.

[12] 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. of Comp. Fluid Dyn.*, 4:171–218, 1995.

[13] 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. Comp. Phys.*, 128:289–300, 1996.

[14] 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.

[15] 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.

[16] 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.

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

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

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

[20] A. Jameson, N.A. Pierce, and L. Martinelli. Optimum aerodynamic design using the Navier–Stokes equations. AIAA paper 97-0101, AIAA 35th Aerospace Sciences Meeting, Reno, NV, 1997.

[21] 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.

[22] 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.

[23] N.A. Pierce and J.J. Alonso. Efficient computation of unsteady viscous flows by an implicit preconditioned multigrid method. Submitted to *AIAA Journal*, January, 1997.

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

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

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

[27] N.A. Pierce. *Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes*. PhD thesis, Oxford University, (expected completion June, 1997).

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

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

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

[31] 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.

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

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

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

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

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

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

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

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

[40] 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.

[41] 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.
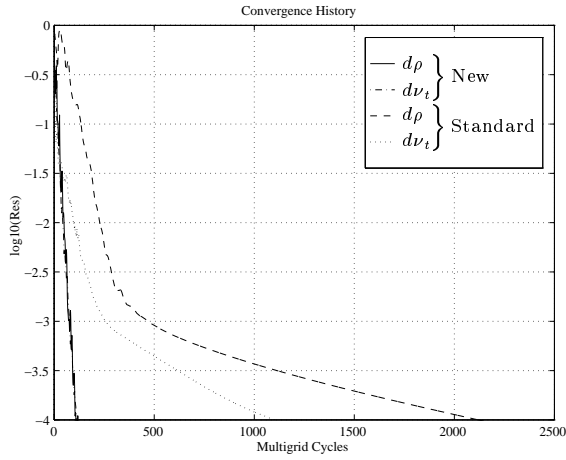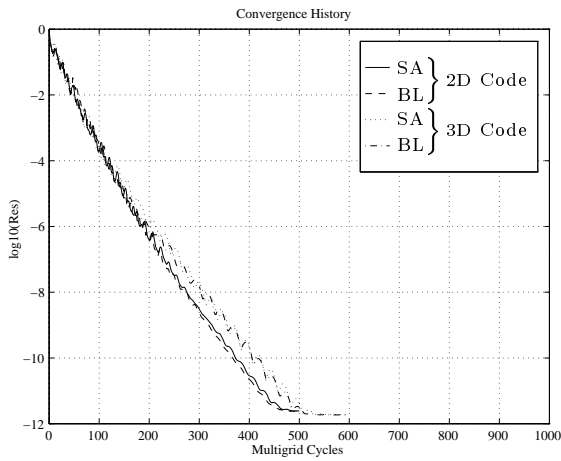
10a: Initial convergence of density and
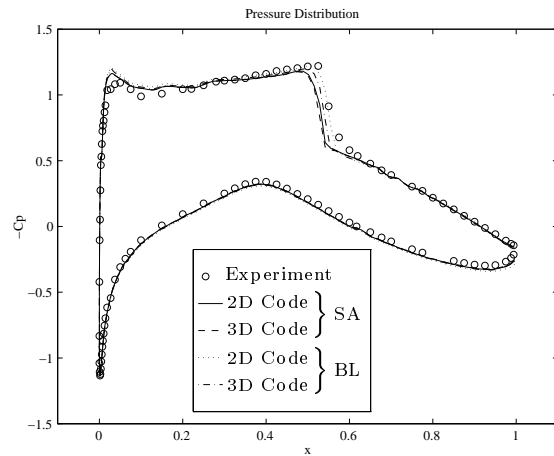turbulent viscosity residuals.
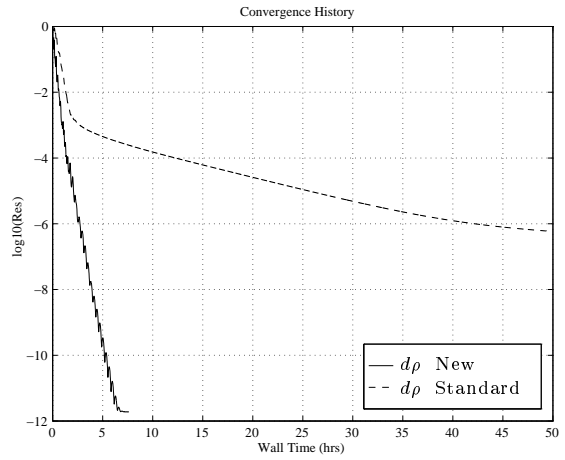
10b: Cost based on wall time.

Figure 10: Straight Wing: Comparison of the new and standard methods using the SA turbulence model.

11a: Convergence of the density residual.

11b: Solution comparison.

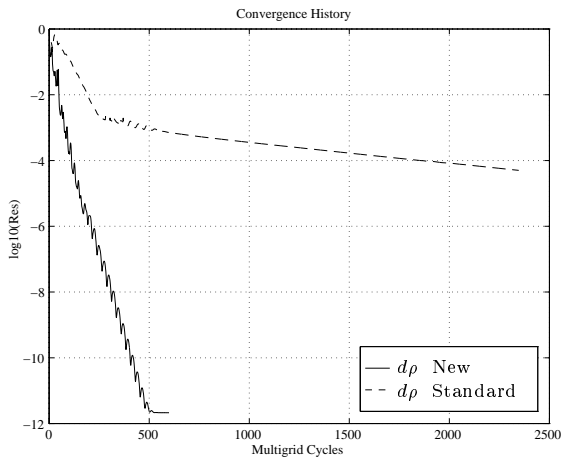Figure 11: Straight Wing: Comparison of the 2D and 3D codes using SA and BL turbulence models.

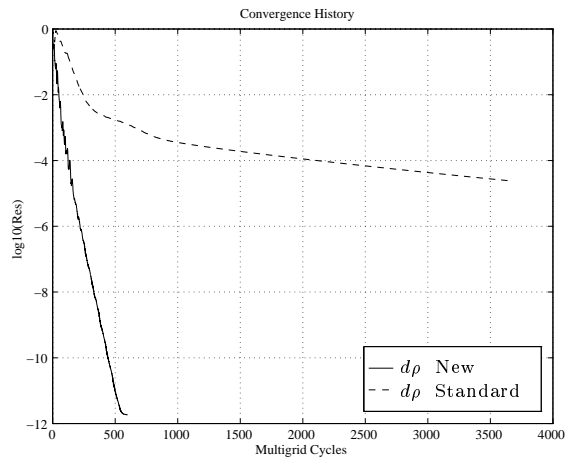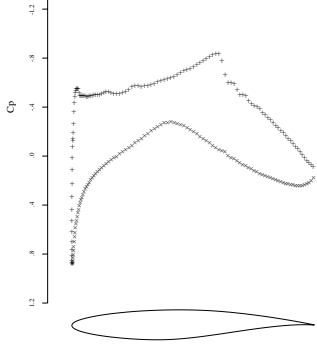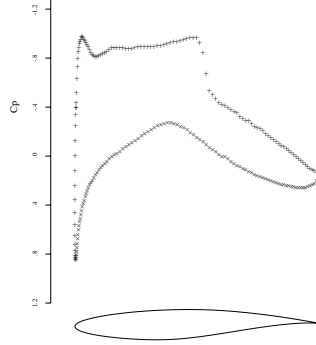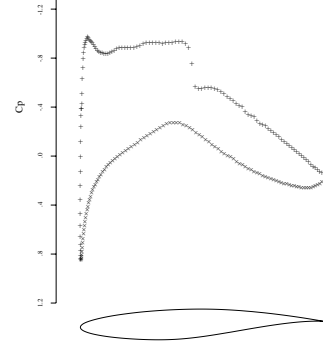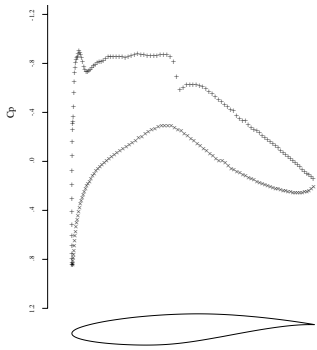Figure 12: Swept Wing: Convergence comparison
using the BL turbulence model.

Figure 13: Twisted Wing: Convergence comparison
using the SA turbulence model.

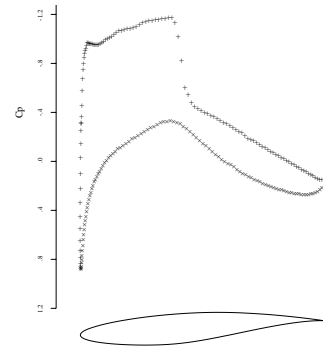14a: C_p at span station $z = 0.25$.   14b: C_p at span station $z = 1.75$.   14c: C_p at span station $z = 3.25$.
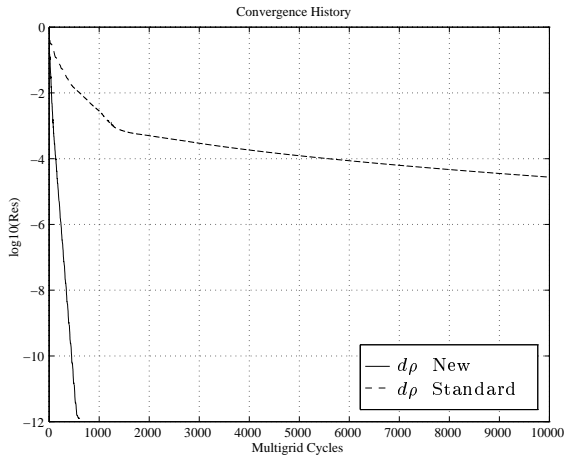

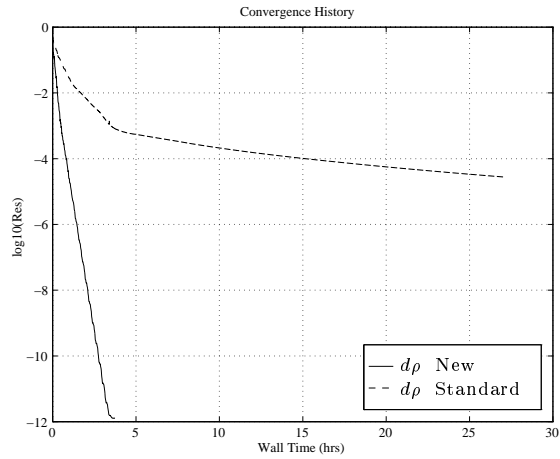14d: C_p at span station $z = 4.75$.   14e: C_p at span station $z = 6.25$.   14f: C_p at span station $z = 7.75$.
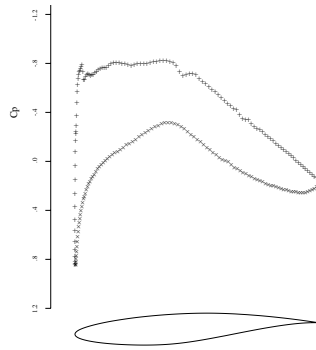
Figure 14: Spanwise pressure distributions for a wing with endplates.
Geometry: RAE2822 Airfoil, Aspect Ratio = 8, Sweep = 30°, Twist = 3.5°.
Flow conditions: $M = 0.8, \alpha = 4.0°, Re = 6.5 \times 10^{-6}$.

15a: Convergence comparison.   15b: Cost comparison.

Figure 15: Fully-3D Wing: Comparison of the new and standard methods using the BL turbulence model.