Application of p-Multigrid to Discontinuous Galerkin Formulations of the Euler Equations

Brendan S. Mascarenhas* and Brian T. Helenbrook†
Clarkson University, Potsdam, New York 13699-5725
and
Harold L. Atkins ‡
NASA Langley Research Center, Hampton, Virginia 23681-2199

DOI: 10.2514/1.39765

We have investigated the p-multigrid iterative method for solving $p = 1$, 2, and 4 discontinuous Galerkin approximations to the Euler equations where p is the degree of the approximating polynomial. For comparison, we have also investigated agglomeration multigrid for $p = 0$ approximations. Block diagonal, line, and sweeping relaxation schemes were examined. The convergence rate to a uniform flow on a structured mesh was analyzed as a function of the flow angle relative to the mesh, grid resolution, underrelaxation factor used, and Mach number. The results show that, even for this simplified problem, many of these schemes do not perform well. For $p = 0$, which corresponds to a conventional finite volume discretization, only the block symmetric Gauss–Seidel line relaxation and the Gauss–Seidel alternate direction line relaxation performed well under all conditions. For $p > 0$, all schemes converged slower than the corresponding $p = 0$ case. Furthermore, all of the schemes were more sensitive to flow angle and Mach number than with $p = 0$. We also found that two-level p-multigrid performs anomalously for $p = 1$ to 0. This behavior is illustrated and explained.

I. Introduction

OVER the last two decades, considerable research effort has been directed toward finite volume discretizations of fluid flow problems and fast solution algorithms. However, computational costs can still become prohibitively large when high-accuracy simulations are required. The reason for this can be attributed to the fact that these discretizations are asymptotically second-order accurate in space. One way to improve the predictive capabilities is to use higher-order-accurate, that is, three and higher, finite element discretizations. The discontinuous Galerkin (DG) and streamwise upwind Petrov–Galerkin discretization strategies are methods for obtaining higher-order accuracy that have received considerable attention and are now fairly well understood. However, as noted by Cockburn and Shu [1], the problem of devising efficient solution algorithms for these discretizations remains open and presents an important area for research. In this work, we focus on solution algorithms for a DG formulation of the Euler equations.

A promising solution strategy is the p-multigrid method, synonymously, multi-order or spectral multigrid method, which was originally proposed by Rönquist and Patera [2]. In this method, convergence is accelerated by using a multigrid algorithm with levels coarsened by reducing the order of the approximating polynomial p. Recent works by Helenbrook et al. [3], Helenbrook and Atkins [4,5], Fidkowski et al. [6], Oliver [7], and Luo et al. [8] have again generated interest in the method.

For applications to the Euler and Navier–Stokes equations, researchers have examined and developed various solution algorithms [3,4,6–8]. However, each of these studies have concentrated on only one or two approaches and have tested them under a small range of conditions. Because these studies were performed for different flowfields, different grid resolutions, and using different discretizations, it is impossible to determine the overall relative performance of the various iterative schemes, the flow regimes within which a particular scheme can be expected to be efficient, the grid dependency of the scheme, and the sensitivity to underrelaxation parameters. In this work, we provide answers to some of these questions.

To accomplish this, we consider a 2-D uniform subsonic flow calculated on a structured rectangular mesh and analyze the convergence rate of various schemes as a function of the flow angle relative to the mesh, the grid resolution, the underrelaxation factor used, and the Mach number M. If a scheme is to perform well in practical simulations, it must be able to perform well for these simplified conditions. Furthermore, conclusions such as the best orientation for line relaxations or the optimal choice of underrelaxation parameter should be applicable to nonuniform flows and unstructured meshes as well.

We investigate block diagonal, line, and sweeping relaxation schemes. We restrict our analyses to block-relaxation schemes because previous work by Atkins and Shu [9] and also Hemker et al. [10] indicates that block-relaxation schemes give much better results than point-relaxation methods. In this work, we investigate classical iterative schemes, such as block Jacobi, Gauss–Seidel, and line solves.

In the following sections, we first present the discrete formulation of the problem, which is derived using the discontinuous Galerkin approach. We then describe the relaxation and multigrid algorithms, as well as the analysis techniques used. Finally, we compare the performance of the various schemes as predicted by the analysis. Results are obtained for approximating polynomial order p of 0, 1, 2, and 4. The $p = 0$ results are obtained using agglomeration multigrid and are presented first to establish a baseline of comparison for the performance of the higher-order methods.

II. Formulation

In this section, we introduce the compressible Euler equations and express them in a form that is suitable for discretization. We then

Presented as Paper 4331 at the 18th AIAA Computational Fluid Dynamics Conference, Miami, FL, 25–28 June 2007; received 21 July 2008; revision received 5 January 2009; accepted for publication 31 January 2009.

*Ph.D. Candidate, Mechanical and Aeronautical Engineering Department; mascarbs@clarkson.edu. Student Member AIAA.

†Associate Professor, Mechanical and Aeronautical Engineering Department; helenbrk@clarkson.edu. Member AIAA.

‡Senior Research Scientist, Computational AeroSciences Branch, MS 128; harold.l.atkins@nasa.gov. Member AIAA.

proceed to obtain a weak form of the Euler equations, based on which the discontinuous Galerkin discretization is formulated. Finally, we describe the iterative techniques analyzed and the method of analysis.

A. Governing Equations

The steady-state, compressible Euler equations can be expressed in the following form:

$$\frac{\partial}{\partial x} f(w) + \frac{\partial}{\partial y} g(w) = 0 \tag{1}$$

where

$$w = [\rho, \rho u, \rho v, \rho E]^T$$

ρ is the density of the fluid, u is the flow velocity in the x direction, v is the flow velocity in the y direction, and E is the total energy of the flow. The x- and y-direction fluxes are given by

$$f(w) = [\rho, \quad \rho u^2 + P, \quad \rho u v, \quad \rho u(E + P/\rho)]^T$$

$$g(w) = [\rho v, \quad \rho u v, \quad \rho v^2 + P, \quad \rho v(E + P/\rho)]^T$$

respectively, where the pressure P is calculated from the ideal gas equation of state as follows:

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

and γ is the ratio of specific heats of the fluid.

If we define the Jacobian matrices

$$A_x = \frac{\partial f}{\partial w}, \qquad A_y = \frac{\partial g}{\partial w}$$

Eq. (1) can be expressed as

$$A_x \frac{\partial w}{\partial x} + A_y \frac{\partial w}{\partial y} = 0 \tag{2}$$

We will analyze this problem on a unit square (Fig. 1), with characteristic boundary conditions in x and periodic boundary conditions in y. The characteristic boundary conditions will be described in the next section. For all of the problems presented here, the flow will be linearized around a freestream flow with a specified Mach number of M. We examine flows with various flow angles θ relative to horizontal, as shown in the figure. As mentioned in Sec. I, if an iterative algorithm is to perform well for more complex problems, a minimum requirement is that it must perform well for this simplified case.

Because of the periodic boundary conditions in the y direction, for a flow angle of exactly 90 deg, the velocity along any vertical line of elements is determined only up to an arbitrary constant. Physically, this corresponds to slip modes in the y direction. These modes are indeterminate and cause the problem to be singular at this angle. Thus, every iterative scheme will be nonconvergent when the flow angle is 90 deg. As the flow angle approaches 90 deg, the problem becomes increasingly difficult to solve because the length of the streamlines from inlet to outlet is $1/\cos(\theta)$.

B. Discontinuous Galerkin Formulation

To formulate the discontinuous Galerkin finite element discretization, we need to introduce a finite-dimensional space of functions to represent the solution. The domain is subdivided into rectangular elements, and on each element we use a polynomial basis to describe the four components of the solution vector w. Following the notation in Arnold et al. [11], we define the following space

$$V_h = \{v \in L^2(\Omega) : v|_K \in \mathcal{P}_p \forall K \in \mathcal{Q}_h\} \tag{3}$$

where $L^2(\Omega)$ is the space of square-integrable functions on the domain Ω, the unit square. $\mathcal{Q}_h$ is the set of quadrilateral elements K that triangulate the domain, and the subscript h refers to the element length associated with a particular mesh. $\mathcal{P}_p$ is a two-dimensional polynomial basis. This basis is formed using the tensor product of one-dimensional sets of polynomial functions, $P_p(x) \cdot P_p(y)$, where P_p is the set of polynomials having degree at most p.

All of the DG formulations we analyze are based on the weak form, obtained by multiplying Eq. (2) by a vector test function v, where $v \in [V_h]^4$, and then integrating by parts. This weak form can be expressed as

$$\sum_{K=1}^{N}\left(\int_{\Gamma_K} v^T G(w_{\Gamma^-}, w_{\Gamma^+})\,\mathrm{d}\Gamma - \iint_{\Omega_K} \nabla v^T \cdot (A_x w, A_y w)\,\mathrm{d}\Omega\right) = 0 \quad \forall\ v \in V_h \tag{4}$$

where Γ_K is the boundary of element K, Ω_K is the domain of element K, w_{Γ^-} is the solution evaluated from the inside of the element boundary, and w_{Γ^+} is the solution evaluated from the outside of the element. These are different because the solution is discontinuous across element boundaries. The function $G(w_{\Gamma^-}, w_{\Gamma^+})$ is the flux through the element boundary. Because the solution is discontinuous at the boundaries, the flux is calculated based on characteristic upwind variables using

$$G(w_{\Gamma^-}, w_{\Gamma^+}) = (A_x, A_y) \cdot \hat{n}(w_{\Gamma^-} + w_{\Gamma^+})/2 - |(A_x, A_y) \cdot \hat{n}|(w_{\Gamma^+} - w_{\Gamma^-})/2 \tag{5}$$

where $\hat{n}$ is the outward normal to the element boundary. For a matrix A, which can be diagonalized as $A = S\Lambda S^{-1}$, with Λ as a diagonal matrix of the eigenvalues of A, $|A|$ is defined as $|A| = S|\Lambda|S^{-1}$, where $|\Lambda|$ is a diagonal matrix of the absolute values of the eigenvalues of A. The boundary flux function provides the interelement coupling by using information from both sides of the element boundary. The characteristic boundary conditions are implemented by setting the exterior values of the flow variables in the flux equal to the freestream values.

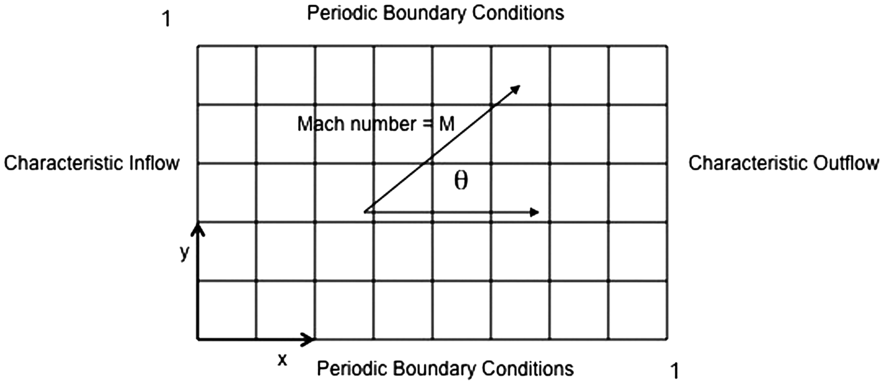

Fig. 1 Domain, boundary conditions, and a typical mesh.

C. Relaxation Schemes

Before explaining the relaxation schemes, we introduce some matrix notation for the linear systems of equations generated by the DG formulations. Because DG formulations are dominated by operations on elements, we label the solution coefficients by element using two indices, that is, $w_{j,k}$, where the indices j and k correspond, respectively, to the horizontal and vertical position of the element. Hence, $w_{j,k}$ is a vector of length equal to the number of physical unknowns per element (four), multiplied by the number of terms in the tensor product forming the polynomial basis set $(p+1)^2$. The solution on each element can be written as $\phi^T w_{j,k}$, where ϕ is the vector of basis functions from $\mathcal{P}_p^4$. Over the entire domain of $N \times N$ elements, the vector of coefficients and basis functions is of length $N^2 \cdot 4 \cdot (p+1)^2$

All of the relaxation schemes are written in the form

$$R\Delta w - Aw = 0 \tag{6}$$

where R is a relaxation matrix, w is the vector of unknown coefficients for all elements, and A is the stiffness matrix. The schemes are all block-relaxation schemes. A block is defined as the matrix of coefficients of the discrete equations for an individual element. In two dimensions, each block is thus of size $[4(p+1)^2] \times [4(p+1)^2]$. The first scheme we use is block Jacobi, where R is the block on the diagonal of A. For each relaxation scheme investigated, we introduce an underrelaxation parameter ω. Here, ω scales the magnitude of the diagonal blocks of the relaxation matrix by a factor of $1/\omega$. Hence, a smaller value for ω corresponds to reducing the magnitude of the correction applied at each iteration.

The second scheme we investigate is a block Gauss–Seidel (GS) scheme. This is the same as the block Jacobi scheme except that the residual Aw and the solution are updated element by element. If the elements are ordered left to right then bottom to top, this corresponds to adding to R the block matrices of A that couple the element being updated to the elements to the left and down (essentially the lower block triangle of A). To keep R circulant, these block matrices are added even when the position of the corresponding element has wrapped around the domain due to periodicity. When an underrelaxation factor is used, only the diagonal blocks are scaled. This corresponds to reducing the magnitude of the correction, but still updating the residual with the most current estimate of the solution.

The block symmetric Gauss–Seidel (SGS) scheme is a two-sweep scheme. The first sweep is the same as block GS, whereas the second is a reverse sweep, that is, the element solutions are updated from right to left then top to bottom. In the results, we concentrate on SGS rather than GS, because these schemes are less dependent on the direction of sweeping relative to the flow direction.

The line relaxation is a block tridiagonal preconditioner that must be inverted along lines of elements. The lines are oriented in the x direction. The line relaxation scheme can also be given a sweeping component by performing the updates sequentially from bottom to top, and each line uses the most recent information in its update, that is, it is a line solve in x and GS in y. The SGS line solve is again a two-sweep scheme, with the first sweep identical to the GS line solve. The second is a reverse sweep, that is, line updates are performed sequentially from top to bottom where each line uses the most recent information in its update. When the underrelaxation parameter ω is used, the entire line of elements is scaled by ω, but the GS components are not. This corresponds to reducing the correction obtained from the line solve but using the most recent data to evaluate the residual.

The alternate direction line relaxation (ADL) is a variant of the line relaxation. In this case, the line relaxation is performed with the lines first oriented horizontally and then again with the lines oriented vertically. We incorporate GS sweeping for the ADL scheme with a first sweep identical to the GS line relaxation. For the second sweep, that is, when the lines are oriented vertically, the GS updates are performed sequentially from left to right. The ADL scheme is also underrelaxed in a manner similar to the line relaxation.

The reader is advised to note that, because all the schemes discussed in this work are block relaxation schemes, henceforth, we shall omit the term block when referring to any scheme.

D. Multigrid

Multigrid is used to accelerate the convergence provided by the relaxation schemes. For the $p = 0$ approximation, we use agglomeration multigrid [12], whereas higher-order DG discretizations are solved using p-multigrid [2,3].

When $p = 0$, a DG discretization is identical to a first-order cell-centered finite volume discretization. In recent work, Mavriplis et al. [12–14] have demonstrated that agglomeration multigrid is efficient for solving two- and three-dimensional finite volume approximations of the compressible Euler equations. In this algorithm, coarse grids are generated by aggregating control volumes. When using unstructured meshes, the main difficulty with this technique is in generating this coarse grid. Because our mesh is structured, it can be easily coarsened by deleting alternate grid lines in each coordinate direction, that is, we agglomerate four cells of the fine mesh to form one cell on the coarse mesh. To transfer residuals to the coarse mesh, we add together the residuals from the cells that are combined to form one cell on the coarse mesh. We use a two-level multigrid cycle; hence, on the coarse mesh, the coarse mesh stiffness matrix is directly inverted. A correction from the coarse element is then transferred back to the four component elements on the fine grid.

For high-order formulations, p-multigrid is used. As in a standard multigrid algorithm, in p-multigrid, restriction and prolongation operators are needed in addition to the relaxation scheme. The restriction operation consists of moving solution residuals from a space of high polynomial order to a lower order. We choose the order of the polynomial spaces such that the coarse space has a degree, $p_c = p/2$. Prolongation is the reverse operation in which the solution correction from the low-order space is transferred to the higher-order space. For a basis ϕ_{l+1} of degree p_c that is contained in the space spanned by a higher-order basis, ϕ_l of degree p, the prolongation operator on an element is given by

$$I_{l,l+1} = \left(\int_K \phi_l \phi_l^T \, \mathrm{d}x\right)^{-1} \int_K \phi_l \phi_{l+1}^T \, \mathrm{d}x \tag{7}$$

This is a matrix of dimension $p \times p_c$ that takes a correction represented using the basis ϕ_{l+1} and gives an equivalent representation using the basis ϕ_l. For this work, we will use a monomial basis, which is hierarchical. In this case, prolongation just consists of correcting the low-order terms while keeping the high-order terms unchanged. The restriction operator is the transpose of prolongation. For a hierarchical basis, this corresponds to transferring only the residuals derived from integration with respect to the low-order polynomials to the coarser space. Both the agglomeration and the p-multigrid algorithms can be written as a recursive subroutine as follows:

```
cycle($l$) {
   if ($l ==$ coarsest level) {
   Direct inversion on coarse mesh:
      $w_{[l]} = A_{[l]}^{-1}(F_{[l]})$
      return
   }
   Relaxation:
      $w_{[l]} = w_{[l]} + R_{[l]}^{-1}(F_{[l]} - A_{[l]}w_{[l]})$
   Restriction:
      $F_{[l+1]} = I_{l,l+1}^T(F_{[l]} - A_{[l]}w_{[l]})$
      $w_{[l+1]} = 0$
   Recursion:
      cycle($l + 1$)
   Prolongation:
      $w_{[l]} = w_{[l]} + I_{l,l+1}w_{[l+1]}$
   return
}
```

The subscript $[l]$ indicates the multigrid level: $l = 0$ corresponds to the finest level. The solution to the discrete equation is $w_{[0]}$. For $l > 1$,

$w_{[l]}$ is a solution correction that will be prolongated to a higher-order space. $F_{[l]}$ is the restriction of the residual from the previous level. At the finest level, $F_{[0]} = 0$. The results presented here are for a cycle with only two levels. The two-level cycle gives an upper bound for the convergence rate because the second-level equations are solved directly rather than iteratively.

E. Analysis Techniques

To determine the convergence rates, we examine the eigenvalues of the multigrid iteration. For a two-level iteration, one multigrid cycle can be simplified to the following form:

$$w^{[1]} = (I - I_{0,1}A_{[1]}^{-1}I_{0,1}^{T}A_{[0]})(I - R_{[0]}^{-1}A_{[0]})w^{[0]} \tag{8}$$

where I is the identity matrix. The spectral radius of the preceding matrix determines the damping factor, which is the factor by which the error is decreased in each multigrid cycle. A damping factor less than one indicates that the scheme is convergent (stable), whereas a damping factor greater than or equal to one indicates that a scheme is divergent (unstable). A stable scheme converges rapidly if its damping factor is closer to zero, whereas a damping factor near one indicates slower convergence.

Because of the periodic boundary conditions in the y direction, we can use the discrete Fourier transform in y to reduce the size of the eigenvalue problem. We assume the solution on each element has the form

$$w_{j,k} = \tilde{w}_j e^{ik\theta_y} \tag{9}$$

where j is the horizontal element index, k is the vertical element index, $\tilde{w}_j$ is a vector of dimension $4(p+1)^2$, and θ_y can take values of $-\pi$ to π by increments of $2\pi/N$, where N^2 is the number of elements in the mesh. Substitution of the form given by Eq. (9) reduces the eigenvalue problem from a size of $4N^2(p+1)^2$ to a set of N eigenvalue problems of size $4N(p+1)^2$. These are then solved numerically for each value of θ_y. The maximum eigenvalue over the range $-\pi < \theta_y < \pi$ is the damping factor of the iterative scheme.

III. Results for $p = 0$ Approximations

We begin by establishing some baseline trends for the relaxation schemes when combined with agglomeration multigrid for $p = 0$. In each case, we study the variation of the damping factors with the flow angle on an 8×8 mesh. The effects of grid resolution on the convergence properties of the schemes are studied by examining variation of the damping factors at flow angles of 0 and 36 deg on meshes with 8×8, 16×16, and 32×32 elements. The baseline Mach number is 0.1. Mach number effects are investigated by comparing results for $M = 0.02$, 0.1, and 0.5.

A. Jacobi Relaxation

We investigate the convergence of Jacobi relaxation by itself and coupled with a two-level agglomeration multigrid algorithm. Figure 2 shows the damping factor with and without multigrid as a function of the angle of flow. The dashed line at $y = 1$ indicates graphically the level below which the damping factors must lie for a scheme to be stable. The two dot-dash curves represent the damping factors for the relaxation scheme using $\omega = 2/3$ and $\omega = 1$. The solid curves are the damping factors with multigrid. It is difficult to see the $\omega = 1$ relaxation results because, for most flow angles, the results with multigrid are exactly the same. All of the schemes are stable at all flow angles other than at 90 deg. This nonconvergence at 90 deg is for the reasons explained in Sec. II.A. As will be seen in the following results, the degradation of the relaxation and multigrid damping factors at flow angles near ±90 deg occurs for most relaxation schemes.

Examining Fig. 2, we see that, for most flow angles, the multigrid scheme converges faster with $\omega = 2/3$ than with $\omega = 1$, whereas, without multigrid, the relaxation scheme converges faster with $\omega = 1$ than with $\omega = 2/3$. The reason for this is that, with $\omega = 1$, the damping factor of the relaxation scheme is better over the entire spectrum of eigenfunctions, but with $\omega = 2/3$, there is more damping of the high wave number modes which improves multigrid performance.

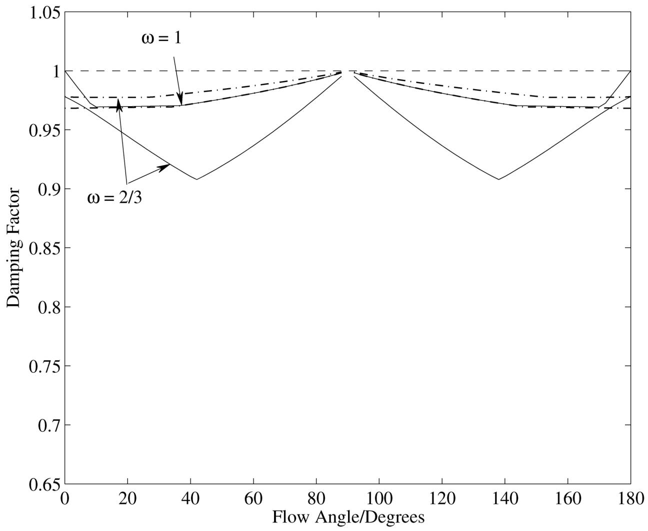

Fig. 2 Damping factor versus flow angle for Jacobi relaxation on $p = 0$ approximation on an 8 × 8 mesh. The dash-dot lines are the relaxation scheme by itself, and the solid lines are for Jacobi relaxation with two-level agglomeration multigrid.

Figure 2 also indicates that the multigrid performance degrades at angles near 0 and 180 deg. This is not because the problem is singular; the characteristic boundary-conditions guarantee that the solution should be uniform. To further investigate this, Fig. 3 shows the effects of grid resolution on the damping factors. At 36 deg and with $\omega = 2/3$, on grid refinement, the multigrid damping factor asymptotes to a value less than one, which indicates that the multigrid scheme is grid independent. This is true at all flow angles away from 0, 90, and 180 deg. However, when the flow is aligned with the mesh ($\theta = 0$), on grid refinement, the multigrid damping factor approaches one. The curve with the filled circle markers is discussed later.

To better understand the aforementioned behaviors, Fig. 4 shows a plot of the ρu component of the slowest converging eigenfunction of the multigrid scheme at 0-deg flow angle. The magnitude of ρv is identically zero for this eigenfunction. The coordinate axes indicate the element number of the grid along the x and y directions, respectively. The figure shows that the slowest-damped mode is oscillatory in y and smooth in x. Because the mode is high wave number in y, multigrid does nothing to eliminate it. Furthermore, because the scheme allows slip, there is no coupling between

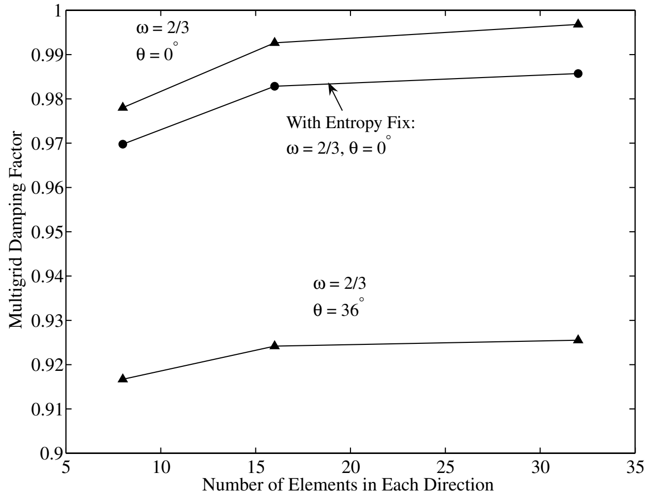

Fig. 3 Variation of the multigrid damping factor with grid resolution at flow angles of 0 and 36 deg for Jacobi relaxation and $\omega = 2/3$, with two-level agglomeration multigrid on a $p = 0$ approximation.

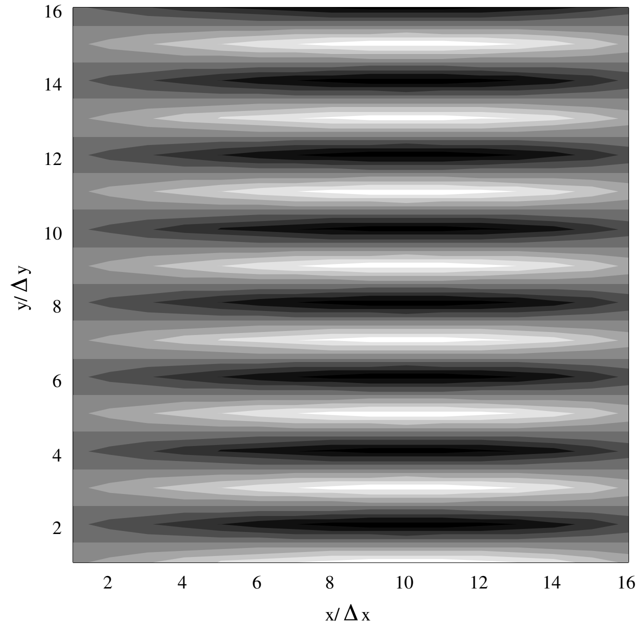

Fig. 4 Eigenfunction plot of ρu component of the critical mode for the Jacobi relaxation scheme with $\omega = 2/3$, on a $p = 0$ approximation at 0-deg flow angle. Damping factor = 0.98.

vertically adjacent values of u when the flow is aligned with the mesh. Thus, the relaxation scheme does not effectively eliminate this mode either. Hence, the scheme has a high damping factor. This slip mode also explains the high sensitivity of the scheme to grid resolution; to eliminate this error mode, information from the inlet boundary must be propagated across the domain. The Jacobi scheme propagates information by, at most, one grid cell per iteration. Hence, a refined grid will require many iterations to converge.

The introduction of an "entropy fix" could prevent the slip modes from decoupling completely when the flow is aligned with the mesh. In our test case, this is done by replacing $|\Lambda|$ in Eq. (5) with $\max(|\Lambda|, 0.1|u_\infty|I)$, where I and u_∞ are the identity matrix and the freestream velocity, respectively. The results are shown in Fig. 3 by the curve with the circle markers. The convergence of the scheme improves with the addition of an entropy fix, but the grid sensitivity persists. It should be noted that $0.1u_\infty$ is probably the largest value that might be practically used, because the entropy fix reduces the accuracy of the solution. Figure 3 thus shows the maximum improvement one might expect from an entropy fix. The benefit of an entropy fix is not significant, hence we have chosen not to incorporate it into any of the following results.

B. Symmetric Gauss–Seidel Relaxation

Figure 5 shows the multigrid damping factor of the SGS scheme as a function of flow angle for the $p = 0$ approximation. This scheme gives improved convergence rates over the diagonal scheme, which can be attributed to the symmetric sweeps helping to propagate information. The curves for the damping factors are not symmetric about 90 deg. This is because the sweep operations are not commutative; a forward sweep followed by a reverse sweep is not the same as a reverse sweep followed by a forward sweep. As the flow angle changes from less than 90 deg to greater than 90 deg, the order of sweeping is effectively permuted, causing the results to be asymmetric about 90 deg.

Although SGS converges faster than Jacobi, the performance again degrades when the flow becomes aligned with the mesh. We have confirmed that this is again caused by the decoupling of slip modes. Figure 6 shows the variation of the multigrid damping factor with grid resolution at flow angles of 0 and 36 deg. Similar to the Jacobi relaxation, this scheme is sensitive to grid resolution when the flow is aligned with the mesh. Thus, although SGS helps to propagate information across the domain, this is apparently not enough to eliminate slip modes.

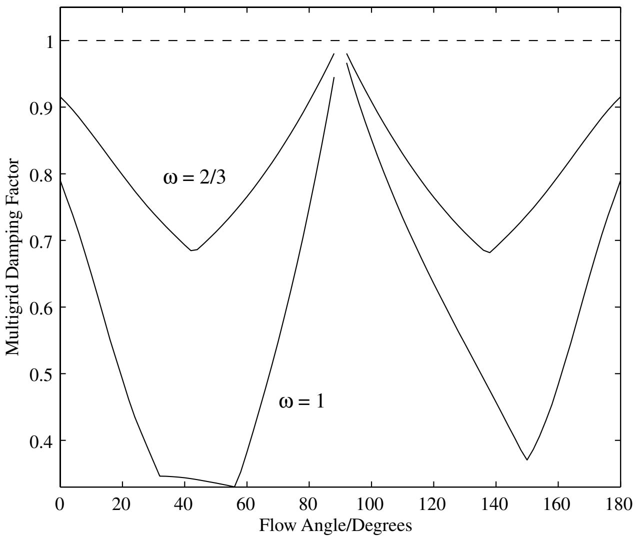

Fig. 5 Multigrid damping factor versus flow angle for SGS relaxation with two-level agglomeration multigrid on a $p = 0$ approximation on an 8 × 8 mesh.

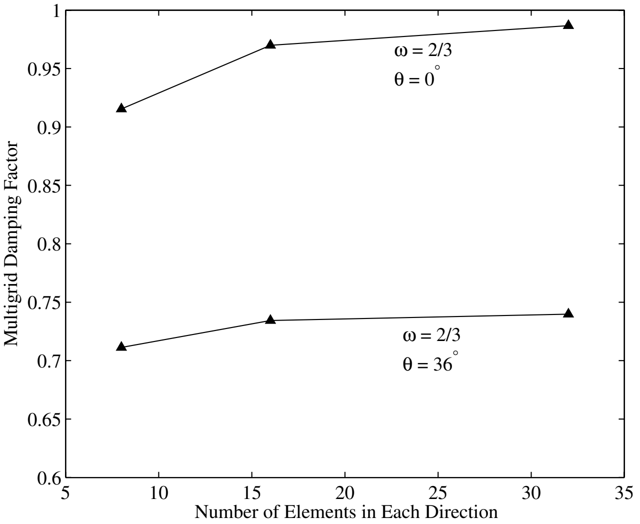

Fig. 6 Variation of the multigrid damping factor with grid resolution at flow angles of 0 and 36 deg for SGS relaxation and $\omega = 2/3$, with two-level agglomeration multigrid on a $p = 0$ approximation.

C. Line Relaxation and Symmetric Gauss–Seidel Line Relaxation

Figure 7 shows the results for line relaxation and SGS line relaxation. The lines are oriented along the x direction. On comparing the multigrid damping factors of the line relaxation scheme with that of the Jacobi scheme (Fig. 2), we see that, for flow angles between 42 and 138 deg, the schemes perform similarly. This is because, for the line relaxation with lines oriented along the x axis, information is still propagated by a single row per iteration in the y direction. This limitation makes the scheme similar to the Jacobi scheme and hence gives comparable convergence rates. The line relaxation performs better than Jacobi when the flow is aligned with the x axis. This indicates that the line relaxation is effective in eliminating slip modes. In fact, every variant of the line relaxation that we studied gives damping factors that are independent of grid resolution at all flow angles including 0 deg. Adding the SGS sweeps to the line relaxation improves performance at all flow angles.

D. Gauss–Seidel Alternate Direction Line Relaxation

The last scheme we show for $p = 0$ is the GS ADL scheme. ADL by itself does not perform as well as SGS line relaxation, whereas GS

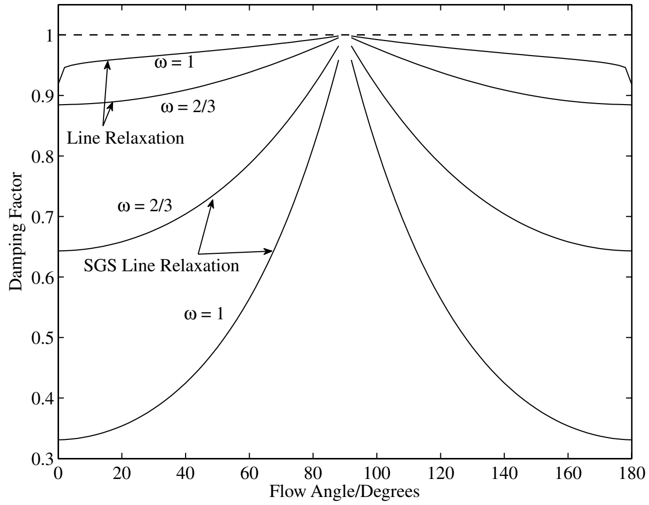

Fig. 7 Multigrid damping factor versus flow angle for line relaxation and SGS line relaxation with two-level agglomeration multigrid on a $p = 0$ approximation on an 8 × 8 mesh.

ADL is the only scheme we have found that gives damping factors that do not degrade approaching 90 deg. Figure 8 shows the damping factor as a function of flow angle. Recall that the GS ADL relaxation is a GS line relaxation performed twice: first with the lines oriented horizontally (sweeping from bottom to top) and then with lines oriented vertically (sweeping from left to right). Because of the alternating direction of the lines and sweeps, this scheme gives different results for flow angles in each of the four quadrants. For this reason, we show results over a 360-deg range of flow angles. When the second sweep (using the vertical lines) is in the direction of the flow (i.e., flow angles between −90 and 90) and $\omega = 1$, the damping factors are better than any of the previous schemes and also do not degrade as the flow angle approaches 90 deg. Instead, at 90 deg, there is a jump discontinuity in the convergence rate from around 0.44 when approaching from $|\theta| < 90$ to 1.0 when approaching from $|\theta| > 90$. Exactly at 90, the damping factor is 1.0 because of the singularity at this point. All of these results are grid independent.

E. Dependence on Mach Number

Figure 9 shows the dependence of the damping factor on Mach number for the best performing schemes. All results are obtained on an 8×8 mesh and at a flow angle of 36 deg. Only the GS ADL scheme with $\omega = 1$ gives Mach number independent results. For all other schemes, the damping factors increase significantly as the Mach number changes from 0.1 to 0.02. Thus, for $p = 0$, the GS ADL scheme with $\omega = 1$ is a reasonably efficient approach for solving the Euler equations.

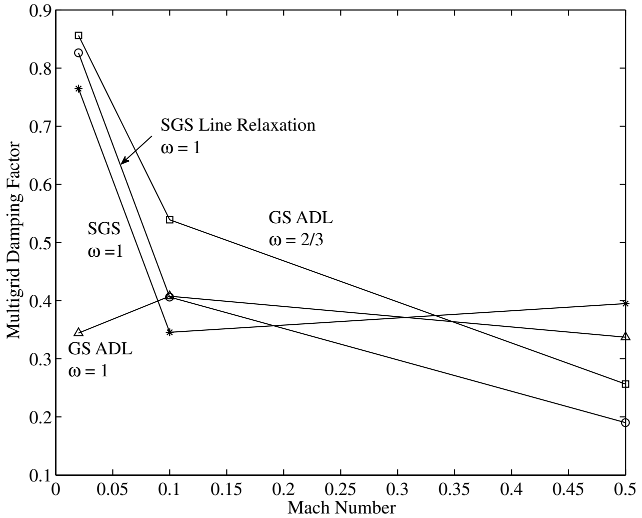

Fig. 9 Multigrid damping factors as a function of Mach number for $p = 0$ and at 36-deg flow angle on an 8 × 8 mesh.

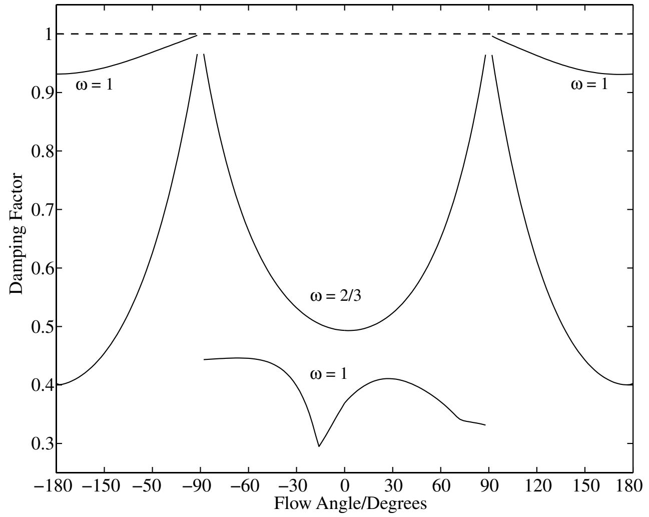

Fig. 8 Multigrid damping factor versus flow angle for the GS ADL relaxation with two-level agglomeration multigrid on a $p = 0$ approximation on an 8 × 8 mesh.

IV. Results for $p > 0$ Approximations

Using the results for the $p = 0$ case as a reference, we now examine the performance of p-multigrid in solving higher-order DG discretizations. Because the Jacobi scheme did not perform well even with $p = 0$, we only examine the symmetric sweep schemes and line relaxation variants. All results for the higher-order approximations are generated on an 8×8 grid. For the $p = 1$ approximation, we have two 1-D polynomial basis functions in each direction. Hence, this is equivalent in total unknowns to a 16×16 mesh at $p = 0$. For the $p = 2$ approximation, we have three 1-D polynomial basis functions in each direction, which is equivalent in total unknowns to a 24×24 mesh at $p = 0$. Similarly, the $p = 4$ approximation is equivalent to a 40×40 mesh at $p = 0$.

A. Symmetric Gauss–Seidel Relaxation

Figure 10 shows the variation of the damping factor with respect to flow angle for the SGS relaxation for $p = 1$–0, $p = 2$–1, and

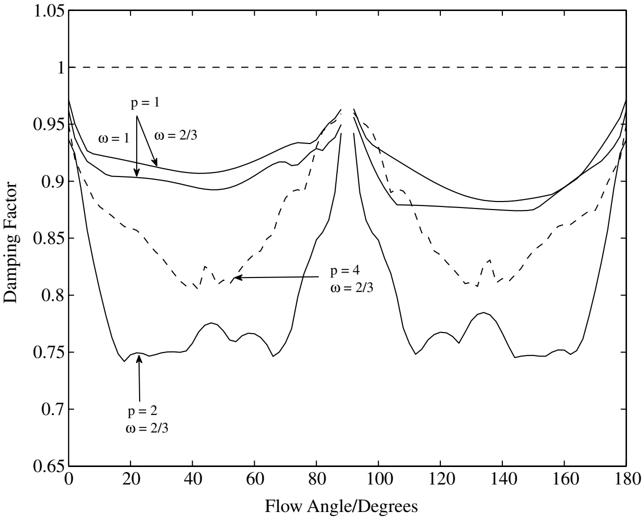

Fig. 10 Multigrid damping factor versus flow angle for SGS relaxation with two-level p-multigrid with $p = 1$–0, $p = 2$–1, and $p = 4$–2 on an 8 × 8 mesh.

$p = 4$–2, two-level, p-multigrid cycles. The results for $p = 2$ and $p = 4$ without underrelaxation are not shown because the relaxation scheme is unstable. With p-multigrid the iteration becomes stable, but such an approach would most likely lack robustness. In the following, we exclude any results where the relaxation scheme is unstable, even if the multigrid iteration is not. Unexpectedly, for most flow angles, the $p = 2$–1 and $p = 4$–2 results are better than the $p = 1$–0 case. This anomalous behavior for the $p = 1$–0 case of p-multigrid was found to be common to all the relaxation schemes that we studied and has been observed by Helenbrook and Atkins [5] for diffusion problems as well.

Helenbrook and Atkins [5] showed that the reason the $p = 1$–0 iteration gives slower convergence is that the slowly damped modes of the $p = 1$ relaxation are essentially continuous functions. These functions cannot be represented well in the $p = 0$ space and thus the damping by multigrid is poor. To verify whether this is true in this case, Fig. 11 shows the ρu component of the slowest converging mode of the relaxation scheme without multigrid at a 36-deg flow angle. This mode occurs at $\theta_y = 2\pi/8$, thus there is one wavelength in the y direction of the 8×8 mesh. Figure 12 is a plot of the 1-D profile taken at $y = 0$ of the same function and shows that the mode is nearly continuous in the x direction. The ρv component of the eigenfunction leads to similar conclusions. The ρ and ρE components appear more discontinuous, but are an order of magnitude smaller in (nondimensional) magnitude. Thus, this seems to indicate that restricting to a continuous space would lead to improved performance for $p = 1$, although further investigation is required to be sure. Because of this anomalous behavior of the $p = 1$–0 case, in the following, we only present results for $p = 2$–1 and $p = 4$–2.

Even for the $p = 2$–1 and $p = 4$–2 cases, the damping factors are not as good as those achieved for the $p = 0$ case. For example, for $p = 0$, the relaxation was stable with $\omega = 1$ and damping factors as low as 0.35 could be obtained (Fig. 5), whereas the minimum damping factor here is about 0.74 for $p = 2$–1 and about 0.81 for $p = 4$–2. Hence, the p-multigrid algorithm provides slower convergence. Also, as we saw for the $p = 0$ case, the damping factor of this scheme rapidly increases as the flow becomes aligned with the mesh. Similar to $p = 0$, for 0-deg flow angle, the results are grid dependent.

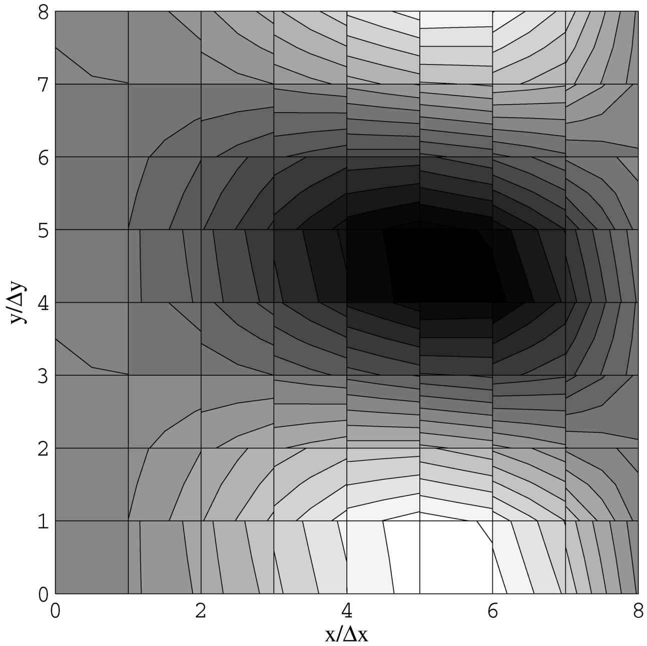

Fig. 11 Contour plot of the slowest converging ρu mode for the SGS relaxation scheme, $p = 1$ approximation, and at 36-deg flow angle on an 8 × 8 mesh.

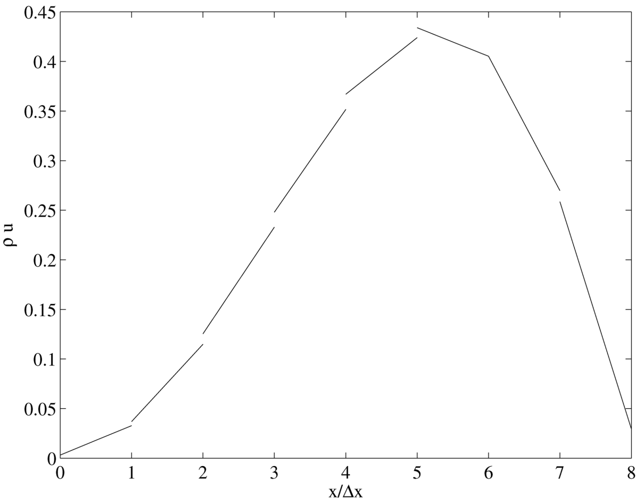

Fig. 12 One-dimensional profile along a row of elements in the x direction for the slowest converging ρu mode for the SGS relaxation scheme, $p = 1$ approximation, and at 36-deg flow angle on an 8 × 8 mesh.

B. Line Relaxation and Symmetric Gauss–Seidel Line Relaxation

We next study the convergence characteristics of line relaxation and SGS line relaxation. Figure 13 shows results for both schemes when using $\omega = 2/3$. For the $p = 2$–1 case, the SGS line relaxation is moderately better than the line relaxation for most flow angles, but is unstable for flow angles between 60 and 120 deg, whereas the line relaxation is stable. This instability is caused by multigrid; the SGS line relaxation itself is stable at these flow angles. Results are not shown for either scheme with $\omega = 1$. The line relaxation scheme with $\omega = 1$ is stable without multigrid, but with multigrid it becomes unstable for flow angles between 69 and 111 deg. Elsewhere, the multigrid performance is similar to that with $\omega = 2/3$. The SGS line relaxation with $\omega = 1$ is unstable.

As with the $p = 2$ schemes, the $p = 4$ SGS line relaxation performs better than the $p = 4$ line relaxation. A notable difference of the $p = 4$ schemes is the degradation in the performance as the flow angle approaches 0 and 180 deg. In general, the damping for the $p = 4$ schemes are not as good as the corresponding $p = 2$ cases.

C. Alternate Direction Line Relaxation

The last scheme we study is the ADL scheme with and without one way GS sweeping. Figure 14 shows the variation of the multigrid

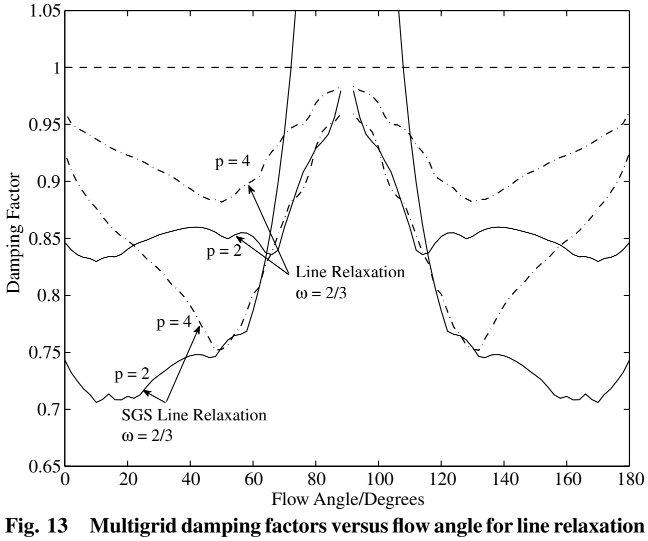

Fig. 13 Multigrid damping factors versus flow angle for line relaxation and SGS line relaxation with two-level p-multigrid, $p = 2$–1 and $p = 4$–2, on an 8 × 8 mesh.

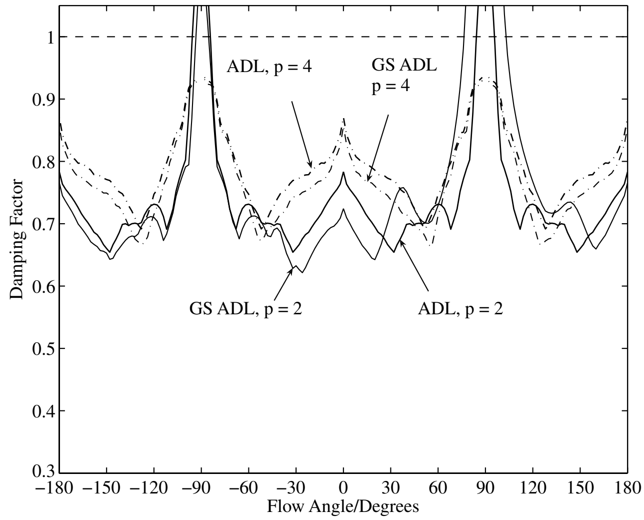

Fig. 14 Multigrid damping factors versus flow angle for the ADL and GS ADL $p = 2$–1 and $p = 4$–2 multigrid schemes with $\omega = 2/3$ on an 8 × 8 mesh.

damping factor with flow angle for the ADL and GS ADL schemes, all with $\omega = 2/3$. The $\omega = 1$ relaxations are unstable for most flow angles. For the reasons outlined previously, the damping factors are shown over a 360-deg range of flow angles. In all cases, the damping factors are much higher than the corresponding $p = 0$ results. However, similar to the other line relaxations, the schemes are grid independent. Of the p-multigrid schemes we have considered, the ADL scheme converges the most rapidly over the widest range of flow angles.

D. Dependence on Mach Number

Lastly, Fig. 15 shows the dependence of the damping factor on Mach number for the best performing schemes at $p = 2$. At $p = 4$, the behavior is similar. All results are obtained on an 8 × 8 mesh and at a flow angle of 36 deg. The figure shows that, for all schemes, the damping factor degrades as Mach number approaches zero. Thus, none of these basic schemes will give adequate results for low Mach number flows. This could be problematic in that almost all practical flow simulations involve regions of stagnated flow. Even if the freestream flow had a fairly large Mach number, these regions could degrade the convergence rate of the entire iteration.

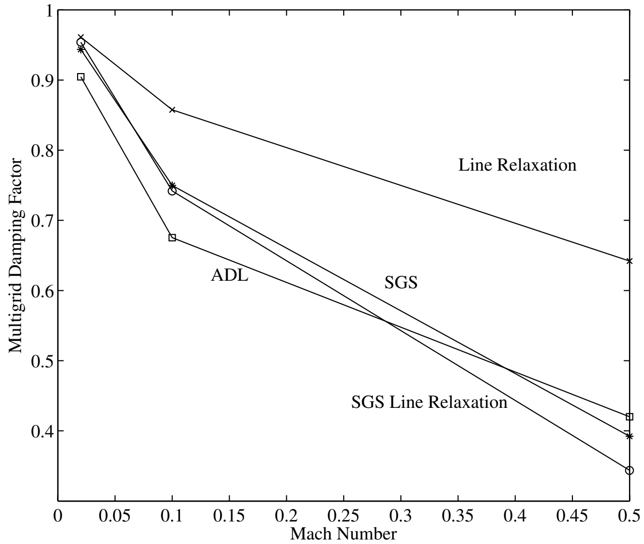

Fig. 15 Multigrid damping factors for $p = 2$–1 on an 8 × 8 mesh at 36-deg flow angle and with $\omega = 2/3$, as a function of Mach number.

V. Conclusions

For $p = 0$ with agglomeration multigrid, the best performing scheme is the GS ADL scheme. This scheme gives flow angle, grid resolution, and Mach number independent results as long as the sweeps are oriented properly. In practical flow simulations, it may be difficult to ensure that this condition is met, but, in that case, symmetric sweeps in both directions would probably eliminate the difficulty.

In general, the p-multigrid schemes on the $p = 2$ and $p = 4$ approximations converge slower than at $p = 0$. Without under-relaxation, most p-multigrid schemes are not robust enough for practical applications because the relaxation schemes themselves are unstable. Furthermore, many of the schemes become unstable when the flow angle approaches 90 deg. Of the schemes analyzed in this work, the ADL scheme with $\omega = 2/3$ was the best performing, as it converged well over the widest range of flow angles. Lastly, all schemes were Mach number dependent and gave damping factors near one for low Mach number flows. On the positive side, all schemes involving line relaxations were grid independent at $p = 2$ and $p = 4$, as they were for $p = 0$.

As was found for diffusive systems [5], there is an inherent problem switching from $p = 1$ to $p = 0$. This was again due to the inability of the $p = 0$ space to represent continuous functions. As we have yet to find a rapidly converging scheme for $p = 2$, no solution to the $p = 1$ problem was sought. Looking ahead, we plan to analyze some of the time-implicit schemes that researchers have implemented [15–17]. If a rapidly converging scheme can be found, then some method of solving the $p = 1$ problem must still be investigated. We have some ideas for this, which have been tested for the diffusion equation [5] and the convective-diffusive equation [18], but not for the Euler equations.

References

[1] Cockburn, B., and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261.
doi:10.1023/A:1012873910884

[2] Rönquist, E. M., and Patera, A. T., "Spectral Element Multigrid I: Formulation and Numerical Results," *Journal of Scientific Computing*, Vol. 2, No. 4, 1987, pp. 389–406.
doi:10.1007/BF01061297

[3] Helenbrook, B. T., Mavriplis, D., and Atkins, H., "Analysis of p-Multigrid for Continuous and Discontinuous Finite Element Discretizations," AIAA Paper 2003-3989, June 2003.

[4] Helenbrook, B. T., and Atkins, H., "Application of p-Multigrid to Discontinuous Galerkin Formulations of the Poisson Equation," *AIAA Journal*, Vol. 44, No. 3, 2006, pp. 566–575.
doi:10.2514/1.15497

[5] Helenbrook, B. T., and Atkins, H. L., "Coupling p-Multigrid to Geometric Multigrid for Discontinuous Galerkin Formulations of the Poisson Equation," *AIAA Journal*, Vol. XXX, No. XXX, XXXX, pp. XX–XX.

[6] Fidkowski, K. J., Oliver, T. A., Lu, J., and Darmofal, D. L., "p-Multigrid Solution of High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 207, No. 1, July 2005, pp. 92–113.
doi:10.1016/j.jcp.2005.01.005

[7] Oliver, T. A., "Multigrid Solution for High-Order Discontinuous Galerkin Discretizations of the Compressible Navier–Stokes Equations," M.S. Thesis, Massachusetts Inst. of Technology, Cambridge, MA, 2004.

[8] Luo, H., Baum, J. D., and Lohner, R., "A p-Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids," *Journal of Computational Physics*, Vol. 211, No. 2, 2006, pp. 767–783.
doi:10.1016/j.jcp.2005.06.019

[9] Atkins, H. L., and Shu, C.-W., "Analysis of Preconditioning and Relaxation Operators for the Discontinuous Galerkin Method Applied to Diffusion," AIAA Paper 2001-2554, 2001.

[10] Hemker, P., Hoffmann, W., and Van Raalte, M., "Two-Level Fourier Analysis of a Multigrid Approach for Discontinuous Galerkin Discretization," *Journal of Scientific Computing*, Vol. 25, No. 3, 2003, pp. 1018–1041.
doi:10.1137/S1064827502405100

[11] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems," *SIAM Journal on Numerical Analysis*, Vol. 39, No. 5, 2002, pp. 1749–1779.
doi:10.1137/S0036142901384162

[12] Mavriplis, D. J., and Venkatakrishnan, V., "A 3D Agglomeration Multigrid Solver for the Reynolds-Averaged-Navier–Stokes Equations on Unstructured Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 23, No. 6, Sept. 1996, pp. 527–544.
doi:10.1002/(SICI)1097-0363(19960930)23:6<527::AID-FLD429>3.0.CO;2-Z

[13] Mavriplis, D. J., "Unstructured Grid Techniques," *Annual Review of Fluid Mechanics*, Vol. 29, Jan. 1997, pp. 473–514.
doi:10.1146/annurev.fluid.29.1.473

[14] Mavriplis, D. J., "Directional Agglomeration Multigrid Techniques for High-Reynolds-Number Viscous Flows," *AIAA Journal*, Vol. 37, No. 10, Oct. 1999, pp. 1222–1230.
doi:10.2514/2.590

[15] Jameson, A., and Caughey, D., "How Many Steps are Required to Solve the Euler Equations of Steady Compressible Flow: In Search of a Fast Solution Algorithm," AIAA Paper AIAA 2001-2673, June 2001.

[16] Caughey, D. A., and Jameson, A., "Fast Preconditioned Multigrid Solution of the Euler and Navier-Stokes Equations for Steady, Compressible Flows," *International Journal for Numerical Methods in Fluids*, Vol. 43, No. 5, 2003, pp. 537–553.
doi:10.1002/fld.521

[17] Liang, C., Kannan, R., and Wang, Z. J., "A *p*-Multigrid Spectral Difference Method with Explicit and Implicit Smoothers on Unstructured Grids," *18th AIAA Computational Fluid Dynamics Conference*, Vol. 2, AIAA, Reston, VA, 2007, pp. 1467–1481; also AIAA Paper 2007-4326.

[18] Mascarenhas, B. S., Helenbrook, B. T., and Atkins, H. L., "Coupling *p*-Multigrid to Geometric Multigrid for Discontinuous Galerkin Formulations of the Convection-Diffusion Equation," *Journal of Computational Physics*, Dec. 2008 (submitted for publication).

Z. Wang
Associate Editor