Journal of Computational Physics **175,** 302–325 (2002)
doi:10.1006/jcph.2001.6948, available online at http://www.idealibrary.com on IDEAL®

An Assessment of Linear Versus Nonlinear Multigrid Methods for Unstructured Mesh Solvers

Dimitri J. Mavriplis

ICASE, MS 132C, NASA Langley Research Center, Hampton, Virginia 23681-0001
E-mail: dimitri@icase.edu

Received April 25, 2001; revised August 24, 2001

The relative performance of a nonlinear full approximation storage multigrid algorithm and an equivalent linear multigrid algorithm for solving two different nonlinear problems is investigated. The first case consists of a transient radiation diffusion problem for which an exact linearization is available, while the second problem involves the solution of the steady-state Navier–Stokes equations, where a first-order discrete Jacobian is employed as an approximation to the Jacobian of a second-order-accurate discretization. When an exact linearization is employed, the linear and nonlinear multigrid methods asymptotically converge at identical rates and the linear method is found to be more efficient due to its lower cost per cycle. When an approximate linearization is employed, as in the Navier–Stokes cases, the relative efficiency of the linear approach versus the nonlinear approach depends both on the degree to which the linear system approximates the full Jacobian as well as on the relative cost of linear versus nonlinear multigrid cycles. For cases where convergence is limited by a poor Jacobian approximation, substantial speedup can be obtained using either multigrid method as a preconditioner to a Newton–Krylov method.

Key Words: multigrid; unstructured; Krylov.

1. INTRODUCTION

Multigrid methods are well known as efficient solution techniques for both linear and nonlinear problems. As with many iterative solvers, multigrid methods can be used directly as nonlinear solvers [1–3] or as linear solvers operating on a linearization arising from a Newton solution strategy for the nonlinear problem at hand [4–6]. In addition, multigrid methods can also be used as a linear or nonlinear preconditioner for a Newton–Krylov method [7–9].

0021-9991/02 $35.00

Newton solution strategies for nonlinear problems incorporating linear multigrid solvers may fail when the initial guess is far removed from the domain of convergence of the nonlinear problem, and globalization methods may be required to ensure a convergent method. Nonlinear multigrid methods overcome this difficulty by using a pseudo-time-stepping analogy on the nonlinear problem directly [1, 3]. On the other hand, nonlinear multigrid methods may fail due to the nonexistence of a solution to the physical problem which is rediscretized on the coarse grid levels, particularly in the initial stages of convergence. However, for various applications, such as time-dependent problems, where the initial guess provided from the previous time step is often within the nonlinear convergence domain of the next time step, or steady-state problems with mild nonlinearities such as subsonic or transonic flows (as opposed to hypersonics), these issues are often of minor importance.

Nonlinear multigrid methods require the evaluation of the full nonlinear residual at each iteration on all grid levels, while linear multigrid methods replace these operations by matrix (Jacobian) vector products at each iteration on all grid levels, with the evaluation of nonlinear residuals only occurring on the fine grid at each outer Newton iteration. One of the great advantages of nonlinear multigrid methods is that they obviate the need to form and store the Jacobian matrix associated with the Newton strategy. For many large-scale unstructured mesh computations, where memory is the limiting factor, nonlinear multigrid methods are indeed the only viable solution strategies [10]. On the other hand, in cases where the nonlinear residual evaluation is costly, linear multigrid methods may become more attractive on a CPU-time efficiency basis, since for a fixed stencil, the cost of the Jacobian-vector products is fixed and independent of the cost of nonlinear residual evaluations, the latter of which are only performed a small number of times in the outer Newton iteration. Of course this statement is only true if the convergence of both methods is similar on a multigrid iteration basis. In the asymptotic convergence region, where solution updates become small and the effect of nonlinearities vanishes, it can be shown, and has been observed, that both methods converge at the same rates per multigrid cycle, provided equivalent iteration strategies are used in both cases (linear and nonlinear Jacobi for example).

The above discussion is only valid when an exact Newton linearization of the nonlinear problem is employed in the linear multigrid method and an exact local linearization is used in the nonlinear method. For discretizations which are not confined to nearest-neighbor stencils, such as second-order-accurate convection operators which rely on distance–two-neighbor stencils, the evaluation and storage costs of the exact Jacobian become prohibitive, and simpler Jacobians based on first-order-accurate nearest-neighbor stencils are most often employed. This practice, which can be thought of as a defect-correction scheme or a preconditioning approach [11], ensures that quadratic convergence of the outer Newton iteration will never be achieved and hence that solution of the linear system to high tolerances even in the asymptotic convergence range will be fruitless. Therefore, the overall solution efficiency of a nonlinear multigrid method versus a linear multigrid method in such cases depends not only on the relative cost of nonlinear residual evaluations versus Jacobian-matrix vector products but also on the degree to which a partial solution of the reduced Jacobian system is successful in converging the full nonlinear system.

In this paper we examine two problems which are solved with a nonlinear full approximation storage (FAS) multigrid method [1], a linear multigrid method, and multigrid preconditioned Newton–Krylov methods. The first problem is a transient two-equation radiation diffusion model which contains strong nonlinearities but for which an exact Jacobian can easily be constructed. The second problem is the solution of the steady-state

Euler and Navier–Stokes equations. In this case, the nonlinearities are less pronounced for the flow regimes considered than in the radiation problem, but a first-order-accurate Jacobian is used to solve the second-order-accurate discretization for the reasons described above. While these two test problems serve to demonstrate two different situations for the comparison of linear versus nonlinear multigrid methods, the eventual solution of coupled radiation–hydrodynamic systems is also of interest.

2. LINEAR AND NONLINEAR MULTIGRID FORMULATIONS

The goal of any multigrid method is to accelerate the solution of a fine grid problem by computing corrections on a coarser grid and then interpolating them back to the fine grid problem. Although this procedure is described in a two-grid context, it is applied recursively on a complete sequence of fine and coarser grid levels. To apply a linear multigrid method to a nonlinear problem, a linearization must first be performed. Thus, if the equations to be solved are written as

$$\mathbf{R}_h(\mathbf{w}_{\text{exact}}) = \mathbf{0}, \tag{1}$$

with the current estimate $\mathbf{w}_h$ yielding the nonlinear residual $\mathbf{r}$,

$$\mathbf{R}_h(\mathbf{w}_h) = \mathbf{r}, \tag{2}$$

the Newton linearization of this system is taken as

$$\frac{\partial \mathbf{R}_h}{\partial \mathbf{w}_h}\Delta\mathbf{w}_h = -\mathbf{r}. \tag{3}$$

This represents a linear set of equations in the solution variable $\Delta\mathbf{w}_h$ (the correction) to which a linear multigrid (i.e., multigrid correction scheme) can be applied. In this case, the coarse grid equation reads

$$\frac{\partial \mathbf{R}_H}{\partial \mathbf{w}_H}\Delta\mathbf{w}_H = -I_h^H \mathbf{r}_{\text{linear}}, \tag{4}$$

where H and h represent coarse grid and fine grid values, respectively, and I_h^H represents the restriction operator which interpolates the fine grid residuals to the coarse grid. The residual of the linear system on the fine grid is given by

$$\mathbf{r}_{\text{linear}} = \frac{\partial \mathbf{R}_h}{\partial \mathbf{w}_h}\Delta\mathbf{w}_h + \mathbf{r}. \tag{5}$$

This definition of the linear residual implies the correspondance

$$\mathbf{r}_{\text{linear}} \approx \mathbf{R}_h(\mathbf{w}_h + \Delta\mathbf{w}_h), \tag{6}$$

where $\mathbf{R}_h$ refers to the nonlinear operator, as previously. The coarse grid corrections $\Delta\mathbf{w}_H$ obtained by solving Eq. (4) are initialized on the coarse grid as zero. After the solution of Eq. (4), these corrections are prolongated or interpolated back to the fine grid.

Alternatively, a nonlinear FAS multigrid scheme can be used to solve Eq. (1) directly without resorting to a linearization. In this case, the FAS coarse grid equation reads

$$\mathbf{R}_H(w_H) = \mathbf{R}_H\left(\tilde{I}_h^H w_h\right) - I_h^H \mathbf{r}, \tag{7}$$

where the term on the right-hand side is often referred to as the defect correction [1, 11]. $\mathbf{R}_H$ represents the coarse grid discretization and I_h^H and $\tilde{I}_h^H$ denote the restriction operators, which are now used to interpolate residuals as well as flow variables from the fine grid to the coarse grids. In principle, different restriction operators for residuals and variables may be employed. If Eq. (7) is rewritten as

$$\mathbf{R}_H(w_H) - \mathbf{R}_H\left(\tilde{I}_h^H w_h\right) = -I_h^H \mathbf{r}, \tag{8}$$

the right-hand sides of Eqs. (4) and (8) represent similar approximations of the restricted nonlinear residual in view of Eq. (6) and the fact that these restricted residuals in the FAS scheme are always evaluated at the most recently available fine grid updates. Therefore, by equating the left-hand sides of Eqs. (4) and (8), the equivalence between the linear multigrid scheme and the nonlinear FAS scheme is given by

$$\mathbf{R}_H(w_H) - \mathbf{R}_H\left(\tilde{I}_h^H w_h\right) \approx \frac{\partial \mathbf{R}_H}{\partial \mathbf{w}_H} \boldsymbol{\Delta} \mathbf{w}_H, \tag{9}$$

which means that the FAS multigrid scheme corresponds to an approximation to a linear multigrid scheme where the coarse grid Jacobians are approximated by finite differencing the operator. Therefore, in the limit of asymptotic convergence, i.e., when $\boldsymbol{\Delta}\mathbf{w}_H \ll 1$, the two methods should yield similar convergence rates.

Note that the above discussion involves no specifications for the coarse grid operator or the Jacobian construction. Therefore, a fair comparison of linear versus nonlinear multigrid methods should utilize similar constructions for both of these quantities in the respective algorithms.

3. MULTIGRID ALGORITHMS

The two multigrid variants implemented in this work are based on the agglomeration multigrid strategy. Agglomeration multigrid was originally developed for finite-volume schemes [12–14], and is based on agglomerating or fusing together neighboring fine grid control volumes to form larger coarse grid control volumes as depicted in Fig. 1. This approach has since been generalized for arbitrary discretizations following algebraic multigrid principles [15]. In fact, agglomeration multigrid can be viewed as a simplification and extension of algebraic multigrid to nonlinear systems of equations. The control-volume agglomeration algorithm can be recast as a graph algorithm, which is similar to algebraic multigrid methods where the “seed” vertex initiating an agglomerated cell corresponds to a coarse grid point and the neighboring agglomerated points correspond to fine grid points in the algebraic multigrid terminology [16]. While weighted graph algorithms can be employed for agglomeration, these weights cannot depend on solution values, as in the algebraic multigrid case, but can only depend on grid metrics. In this manner, the coarse grid levels are static and need only be constructed at the beginning of the simulation.

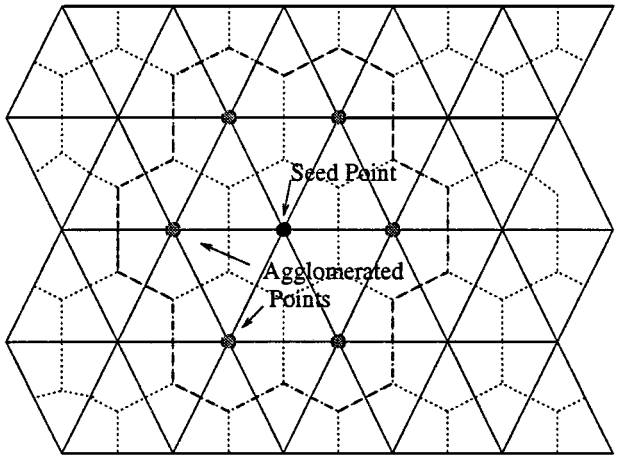

FIG. 1. Agglomeration multigrid coarse level construction.

As in the algebraic multigrid case, the agglomeration multigrid employs a Galerkin projection for the construction of the coarse grid equations. Thus, the coarse grid operator is given by

$$\mathbf{R}_H = I_h^H \mathbf{R}_h I_H^h, \tag{10}$$

where I_h^H is the restriction operator, I_H^h is the prolongation operator, and both operators are taken as piecewise constants. This simple construction applies equally to linear and nonlinear operators and reduces to forming the coarse grid equation at an agglomerated cell as the sum of the fine grid equations at each fine grid cell contained in the coarse grid cell. The nonlinearities in the operator are evaluated using solution variables on the coarse grid interpolated up from the fine grid.

Given this multigrid infrastructure, two particular algorithms which differ mainly in the manner in which nonlinearities are handled are developed for comparison. The first involves a standard nonlinear FAS multigrid algorithm, and the second involves a linear multigrid algorithm applied to the linearization of the governing equations.

3.1. *FAS Scheme*

In the nonlinear FAS multigrid algorithm, Eq. (1) is solved directly. The coarse grid equations are formed by Galerkin projection (c.f. Eq. (10)) and the nonlinearities in the coarse grid operator are evaluated using coarse level solution variables interpolated up from the fine grid using the $\tilde{I}_h^H$ restriction operator (as per Eq. (8)). On each grid level, the discrete equations are solved using a Jacobi preconditioned multistage time-stepping scheme (for the Navier–Stokes equations) [17–20] or a nonlinear block Jacobi iteration which can be written as

$$\mathbf{w}^{\text{new}} = \mathbf{w}^{\text{old}} + [D]^{-1}\mathbf{R}(\mathbf{w}^{\text{old}}), \tag{11}$$

where $[D]$ represents the block diagonal of the Jacobian matrix. This smoother constitutes a nonlinear solver since the nonlinear residual is updated at each stage. The smoother

incurs minimum memory overheads since only the storage of the block matrix $[D]$ representing the coupling between the solution variables at each grid point is required. This scheme is equivalent to a single-stage Jacobi preconditioned multistage time-stepping scheme.

3.2. *Linear Multigrid Scheme*

The linear multigrid scheme solves Eq. (3) on the fine grid and Eq. (4) on the coarse levels. On the fine grid, the Jacobian $\partial \mathbf{R}_h / \partial \mathbf{w}_h$ is formed by explicitly differentiating (hand coding) the discrete operator $\mathbf{R}_h$. On the coarse levels, for consistency with the FAS multigrid algorithm, the Jacobian is taken as the explicit differentiation of the coarse nonlinear operator obtained by the Galerkin approximation (c.f. Eq. (10)). Thus flow variables as well as residuals are restricted to the coarser grids, but the nonlinear residuals on these coarser levels are not evaluated; only the Jacobians corresponding to the linearization of the nonlinear coarse level residuals are evaluated. These coarse level Jacobians are evaluated at the beginning of the solution phase for the nonlinear time-step problem and are then held fixed throughout the linear multigrid iterations. The multilevel linear system constructed in this manner more closely approximates the equivalent FAS scheme as opposed to the more traditional approach of agglomerating the fine grid Jacobian terms directly. Memory requirements for the linear multigrid scheme are increased over those of the FAS scheme due to the required storage of the fine and coarse level Jacobians.

An outer Newton iteration is employed to solve the complete nonlinear problem $\mathbf{R}(w) = 0$. Within each Newton iteration, the linear system defined by Eq. (3) is solved by the linear multigrid algorithm. This provides a new fine grid nonlinear correction Δw, which is then used to update the nonlinear residual. These nonlinear iterations converge quadratically provided the linear system is solved within a sufficient tolerance and a consistent linearization is employed. When approximate Jacobian representations are employed, such as in the Navier–Stokes equations, slower convergence of this outer iterative procedure is obtained.

On each grid level, the linear multigrid scheme solves the linear system using a block Jacobi smoother. If the Jacobian is divided into diagonal and off-diagonal block components, labeled as $[D]$ and $[O]$, respectively, the Jacobi iteration can be written as

$$[D]\boldsymbol{\Delta}\mathbf{w}_h^{n+1} = -\mathbf{r} - [O]\boldsymbol{\Delta}\mathbf{w}_h^{n}, \tag{12}$$

where $\boldsymbol{\Delta}\mathbf{w}_h^n$ represents corrections from the previous linear iteration, and $\boldsymbol{\Delta}\mathbf{w}_h^{n+1}$ represents the new linear corrections produced by the current linear iteration. At each linear iteration, the solution of Eq. (12) requires the inversion of the block matrix $[D]$ at each grid point. The linear corrections $\boldsymbol{\Delta}\mathbf{w}_h$ are initialized to zero at the first iteration on each grid level. Therefore, this linear iteration strategy reduces to the nonlinear Jacobi scheme described above in the event only a single linear iteration is employed.

In contrast to the nonlinear FAS multigrid algorithm, this strategy only evaluates the residuals, Jacobians (i.e., $[D]$ and $[O]$ terms), and the variables interpolated up to the coarse grids at the start of the nonlinear iteration; they are held fixed during all inner linear multigrid cycles within a nonlinear iteration.

4. RADIATION DIFFUSION PROBLEM

The nonequilibrium radiation diffusion equations can be written as

$$\begin{aligned} \frac{\partial E}{\partial t} - \nabla \cdot (D_r \nabla E) &= \sigma_a (T^4 - E), \\ \frac{\partial T}{\partial t} - \nabla \cdot (D_t \nabla T) &= -\sigma_a (T^4 - E), \end{aligned} \tag{13}$$

with

$$\sigma_a = \frac{z^3}{T^3}, \qquad D_r(T, E) = \frac{1}{3\sigma_a + \frac{1}{E}\left|\frac{\partial E}{\partial n}\right|}, \qquad D_t(T) = \kappa T^{5/2}.$$

Here, E represents the photon energy, T is the material temperature, and κ is the material conductivity. In the nonequilibrium case, the nonlinear source terms on the right-hand-side are nonzero and govern the transfer of energy between the radiation field and material temperature. Additional nonlinearities are generated by the particular form of the diffusion coefficients, which are functions of E and T. In particular, the energy diffusion coefficient $D_r(T, E)$ contains the term $|\partial E/\partial n|$, which refers to the gradient of E in the direction normal to the cell interface (in the direction of the flux). This limiter term is an artificial means of ensuring physically meaningful energy propagation speeds (i.e., no greater than the speed of light) [7, 8, 21]. The atomic number z is a material coefficient, and while it may be highly variable, it is only a function of position (i.e., $z = f(x, y)$ in two dimensions).

Equations (13) represent a system of coupled nonlinear partial differential equations which must be discretized in space and time. Spatial discretization on two-dimensional triangular meshes is achieved by a Galerkin finite-element procedure, assuming linear variations of E and T over a triangular element. The nonlinear diffusion coefficients are evaluated by first computing an average T and E value along a triangle edge and then computing the nonlinear diffusion coefficient at the edge midpoint using these averaged values. The gradient of E in the D_r diffusion coefficient is also taken as a one-dimensional gradient along the direction of the stencil edge. The source terms are evaluated using the local vertex values of E and T exclusively rather than linear variations of these variables.

The time derivatives are discretized as first-order backward differences with lumping of the mass matrix, which leads to an implicit scheme requiring the solution of a nonlinear problem at each time step. This approach is first-order accurate in time and is chosen merely for convenience since the principal objective is the study of the solution of the nonlinear system.

The Jacobian of the required linearizations is obtained by differentiation (hand coding) of the discrete nonlinear residual. Because the spatial discretization involves a nearest-neighbor stencil, the Jacobian can be expressed on the same graph as the residual discretization, which corresponds to the edges of the triangular grid. The initial guess for the solution of the nonlinear problem at each time step is taken as the solution obtained at the previous time step.

The test case chosen for this work is taken from [7] and is depicted in Fig. 2. We consider a unit square domain of two dissimilar materials, where the outer region contains an atomic number of $z = 1$ and the inner region $(1/3 < x < 2/3)$, $(1/3 < y < 2/3)$ contains an atomic number of $z = 10$. The top and bottom walls are insulated, and the inlet and outlet

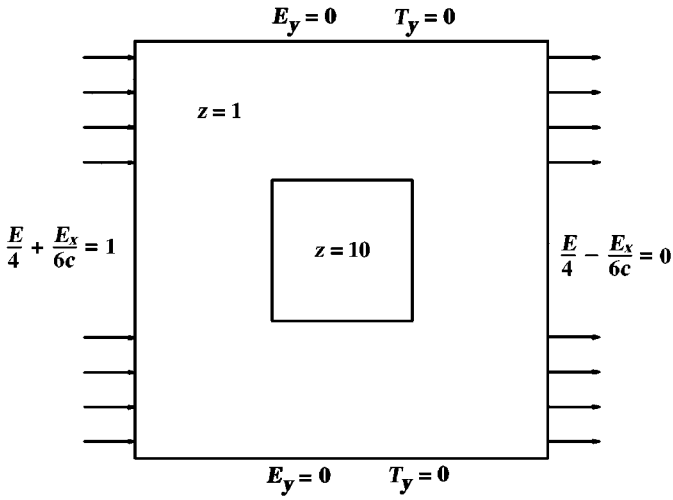

FIG. 2. Sample test problem for nonlinear radiation diffusion equations.

boundaries are specified using mixed (Robin) boundary conditions, as shown in the figure. This domain is discretized using a triangular grid containing 7502 vertices, shown in Fig. 3. The grid conforms to the material interface boundaries in such a way that no triangle edges cross this boundary.

Figure 4 illustrates a typical simulation for this case. Incoming radiation sets up a traveling thermal front in the material, the progress of which is impeded by the region of higher atomic number z. At critical times in the simulation, the diffusion coefficients can vary by up to six orders of magnitude near the material interfaces, thus providing a challenging nonlinear behavior for the multigrid algorithms. At each physical time step, a nonlinear problem must be solved. It is the solution of this transient nonlinear problem at a given time step which forms the test problem for the two agglomeration multigrid algorithms. Clearly, the size of the physical time step affects the stiffness of the nonlinear problem to be solved, with smaller physical time steps leading to more rapidly converging systems. The

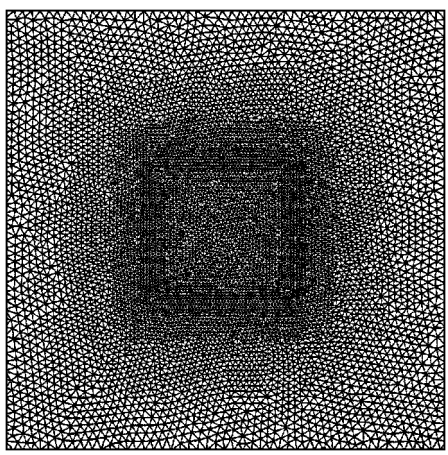

FIG. 3. Unstructured grid for nonlinear radiation diffusion problem: 7502 vertices.

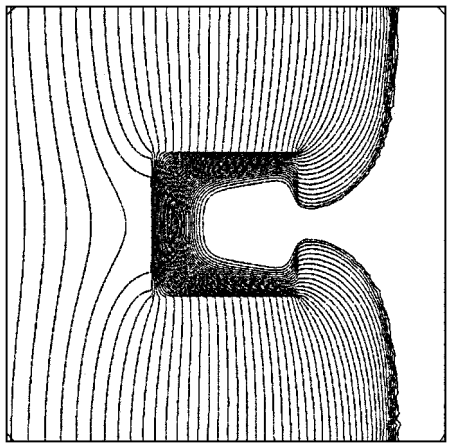

FIG. 4. Solution for nonlinear radiation diffusion problem: contours of T.

nondimensional time step chosen in this simulation was taken as 0.01. This constitutes a rather large value compared to those employed in Ref. [7] (usually of the order of 10^{-3}) and may have an adverse effect on overall temporal accuracy, but it provides a more stringent test case for the multigrid solvers. Of the order of 1000 time steps are required to propagate the thermal front from the inlet to the outlet boundary in the current simulation.

Table I depicts the relative CPU times required for a nonlinear residual evaluation on the fine grid, assembly of the various Jacobian matrix entries on the fine grid, and various components of the linear and nonlinear multigrid (MG) algorithms. The residual and Jacobian terms are assembled within the same loop for cache efficiency reasons, and minimal incremental work is incurred for computing the additional off-diagonal Jacobian terms required for the linear multigrid scheme because many of the block diagonal (point) Jacobian terms consist of the sum of the corresponding off-diagonal Jacobian terms and thus require the same computations. For both linear and nonlinear schemes, the block diagonal Jacobians must be inverted, as shown in Eqs. (11) and (12). For multiple Jacobi sweeps, the lower upper (LU) decomposition of these block matrices is formed on the first pass and then is frozen for

TABLE I

Relative CPU Time Required for Various Components of Linear and Nonlinear Multigrid Methods for the Radiation Diffusion Problem

Component	Normalized timing
Nonlinear residual	1.0
Residual + point jacobians	2.52
Residual + entire jacobian	2.82
First stage nonlinear sweep	2.82
Additional stages nonlinear sweeps	1.07
First linear Jacobi sweep	0.364
Additional linear Jacobi sweeps	0.173
FAS MG cycle	13.04
Linear MG cycle	3.31

subsequent passes. Thus the first linear or nonlinear Jacobi iteration incurs additional cost over subsequent passes, as depicted in the table. The timings illustrate the lower cost of the linear iterations, which are up to five times faster than the corresponding nonlinear iterations. In the nonlinear case, the initial iteration involves the computation of a nonlinear residual, the diagonal Jacobian terms, and the LU decomposition of these Jacobians, while subsequent iterations only require the evaluation of the nonlinear residuals. In the linear case, the first iteration includes the LU decomposition of the point Jacobians but does not include residual and Jacobian construction timings (which are relegated to the outer Newton iteration). From the table, the nonlinear FAS multigrid cycle is seen to require four times more CPU time than the equivalent linear multigrid cycle. In this case, a four-level W(3,0) saw-tooth cycle was used, with three (linear or nonlinear) Jacobi iterations performed on each level when going from fine to coarse levels. These timings do not include the outer Newton iteration in the linear case, which incurs a nonlinear residual evaluation and Jacobian construction, because this expense may be amortized over a variable number of linear multigrid cycles.

Figure 5 compares the convergence rates of the linear and nonlinear multigrid schemes in terms of the number of multigrid cycles. The four-level W(3,0) saw-tooth cycle described above is employed in both cases. For the linear scheme, three linear multigrid cycles are employed for each Newton update, and the convergence history of the linear residual is plotted alongside that of the nonlinear residual. As expected, quadratic convergence of the nonlinear residual is initially observed in the linear multigrid Newton scheme. However, this quadratic behavior is lost after four Newton iterations due to the inexact solution of the linear problem (using three multigrid cycles). In this region, the convergence rate of the outer nonlinear Newton scheme becomes governed or limited by the convergence of the inner linear problem. In the quadratic convergence region, each time the nonlinear residual is updated, the linear residual increases slightly before resuming its downward trend. In the asymptotic region, the linear and nonlinear residuals become approximately equal, as expected from Eq. (6). In this region, the convergence rates of the nonlinear FAS multigrid scheme and the linear multigrid scheme become equivalent in terms of the number of

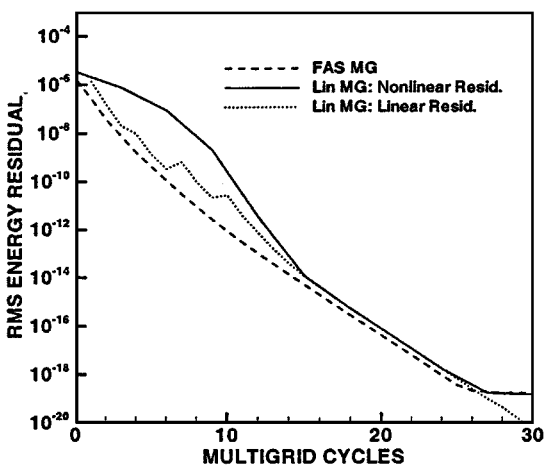

FIG. 5. Convergence rates for the transient radiation problem in terms of multigrid cycles (three linear MG cycles per Newton update).

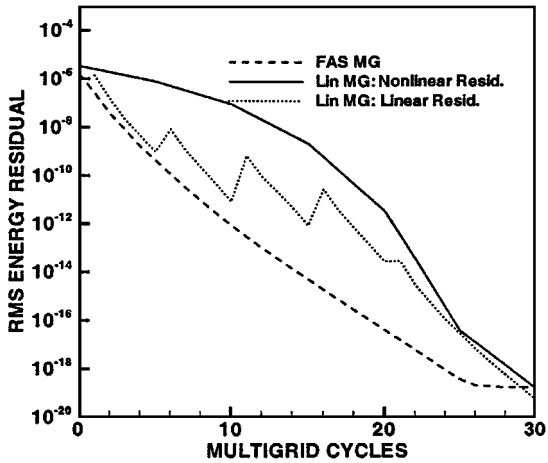

FIG. 6. Convergence rates for the transient radiation problem in terms of multigrid cycles (five linear MG cycles per Newton update).

multigrid cycles, as expected from Eq. (9). The fact that the linear multigrid convergence plot lies slightly to the right of the FAS convergence curve is due to the additional effort spent solving the linear system in the initial phases of slow nonlinear convergence. Figure 6 further illustrates this point by comparing the convergence history for the same approach using five linear multigrid cycles per Newton iteration. In this case, the final asymptotic convergence rate is similar but reached at a later stage and with additional numbers of multigrid sweeps due to increased oversolving of the linear system in the initial stages of nonlinear convergence. Adaptive convergence criteria for the linear system can clearly aid in reducing oversolution of the linear system, although this has not been considered in this work.

Figure 7 compares the convergence efficiencies of the nonlinear FAS multigrid approach with the linear multigrid approach using three multigrid cycles per Newton iteration, in terms of CPU time. The linear multigrid method is over three times more efficient; this efficiency advantage can be entirely attributed to the lower cost per multigrid cycle of the linear multigrid scheme.

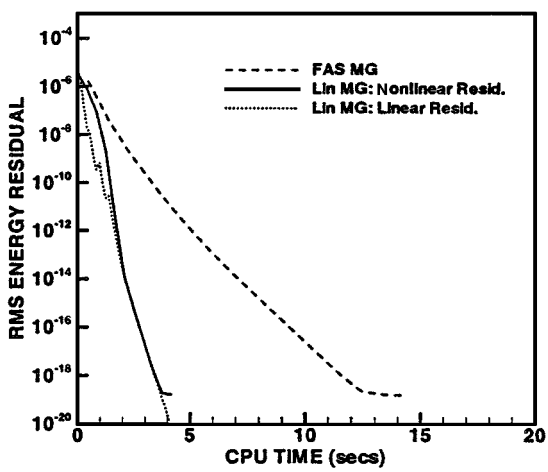

FIG. 7. Convergence rates for the transient radiation problem in terms of normalized CPU time.

5. SOLUTION OF STEADY-STATE NAVIER–STOKES EQUATIONS

The Navier–Stokes equations are discretized on mixed triangular-quadrilateral meshes using a vertex-based approach where the flow variables are stored at the grid vertices. Median-dual control volumes are constructed around each vertex, and fluxes at control-volume interfaces are evaluated using a Roe approximate Rieman solver [22]. Second-order accuracy is obtained through a simplified gradient reconstruction technique which results in a distance–two-neighbor stencil. Viscous terms are constructed as diffusion operators involving nearest-neighbor stencils. For inviscid flow simulations, the viscous fluxes are neglected, while for viscous turbulent flows these terms are retained and the influence of turbulence is simulated using the Spalart–Allmaras one equation turbulence model [23]. The turbulence equation is discretized in the same manner as the flow equations, with the exception that the convective terms are only first-order accurate. The turbulence equation is solved simultaneously but uncoupled from the flow equations using the same multigrid algorithm.

The nonlinear FAS multigrid solver employs a multistage time-stepping scheme as a smoother on all grid levels, which requires the evaluation of the nonlinear residual at each stage. While the fine grid equations are discretized to second-order accuracy, the coarse level equations are only discretized to first-order accuracy. This simplifies their implementation on the coarse level agglomerated graphs and is consistent with practices used on structured geometric multigrid solvers for similar problems [3, 15]. Local preconditioning is applied to the multistage scheme by premultiplying the nonlinear residual by the inverted block diagonal Jacobian matrix at each stage [17–20]. This is equivalent to a (scaled) nonlinear Jacobi iteration at each stage. For viscous flows, line preconditioning is employed, which involves inverting the block tridiagonal Jacobian entries along lines constructed in boundary layer regions (c.f. Fig. 12) [20, 24]. This corresponds to a nonlinear iterative line solution technique which can be described by Eq. (11), where $[D]$ now represents the line Jacobians instead of the diagonal elements. In isotropic grid regions, the lines reduce to a single point and the line preconditioning becomes equivalent to Jacobi preconditioning. In all cases, the local Jacobian entries correspond to those derived from a first-order discretization.

The LU decomposition of these local Jacobians is performed on the first stage of the multistage scheme and then frozen for the subsequent stages of the scheme, thus amortizing the LU decomposition cost over multiple stages.

The linear multigrid method operates on the discrete Jacobian of the first-order discretization of the nonlinear flow equations, although the fine grid flow equations are discretized to second-order accuracy. Nonlinear residuals are only evaluated on the fine grid at the beginning of each linear solution phase, which may involve multiple linear multigrid sweeps. Coarse grid Jacobians are obtained by linearizing the nonlinear coarse grid agglomerated operator to provide a more consistent comparison between equivalent linear and nonlinear methods. On each grid level, multiple passes of a linear Jacobi or Gauss–Seidel smoother are employed for inviscid flows. For viscous flows, multiple passes of a linear line solver are employed, following Eq. (12), where $[D]$ corresponds to the block tridiagonal Jacobians taken along the set of lines constructed in the grid and $[O]$ corresponds to the remaining Jacobian entries. The Jacobi implementations of point and line algorithms correspond to the linear counterparts of the nonlinear smoothers used in the FAS multigrid algorithm. In the Gauss–Seidel implementation, lines and points are presorted in increasing x direction,

TABLE II
Relative CPU Time Required for Various Components of Linear and Nonlinear Multigrid Methods for the Inviscid Fluid Flow Problem

Component (Euler)	Normalized timing
Nonlinear residual	1.0
Residual + line Jacobians	1.62
Residual + entire Jacobians	1.86
First stage nonlinear sweep	1.96
Additional stages Nonlinear sweeps	1.26
First Linear GS Sweep	0.43
Additional Linear GS Sweeps	0.38
Three-stage FAS MG cycle	8.92
Five-stage FAS MG cycle	9.86
Linear MG cycle (1 W-cycle)	5.70
Linear MG cycle (2 W-cycles)	8.98

and sweeps using latest available updates are performed on the grid in increasing and decreasing x direction at odd and even smoothing passes, respectively. When multiple linear smoothing passes are employed, the LU decomposition of the local point or line Jacobians is performed on the first pass and then is frozen for subsequent passes.

For the inviscid case, an isotropic coarsening strategy which results in a coarsening ratio of $4:1$ is employed for generating coarse agglomerated levels, while a directional coarsening strategy which proceeds in the direction of the implicit lines is employed in the viscous flow cases, also yielding a $4:1$ reduction in complexity between fine and coarse levels [20, 24].

Tables II and III depict the relative CPU times required for the various components of the linear and nonlinear algorithms on the grid of Fig. 11, for both inviscid Euler computations and viscous Navier–Stokes computations. Both a three-stage [20, 25] and a five-stage [26] nonlinear smoother are examined. The time required for the five-stage smoother increases only moderately over that required for the three-stage scheme because the local Jacobian

TABLE III
Relative CPU Time Required for Various Components of Linear and Nonlinear Multigrid Methods for the Viscous Fluid Flow Problem

Component (Navier–Stokes)	Normalized timing
Nonlinear residual	1.17
Residual + line Jacobians	2.13
Residual + entire Jacobian	2.39
First stage nonlinear sweep	2.44
Additional stages nonlinear sweeps	1.44
First linear GS sweep	0.43
Additional linear GS sweeps	0.38
Three-stage FAS MG cycle	10.4
Five-stage FAS MG cycle	11.3
Linear MG cycle (1 W-cycle)	6.3
Linear MG cycle (2 W-cycles)	9.6

LU decomposition is only performed once for each scheme and the dissipative terms are only evaluated three times for both schemes (at odd stages only for the five-stage scheme) [26]. Assembly of the complete Jacobian required for the linear scheme incurs relatively little extra overhead beyond that required for assembling the point or line Jacobians since many of the same terms required for the point Jacobians can be used in these additional off-diagonal Jacobian elements. Both linear and nonlinear smoothers involve additional startup costs on the first smoothing pass due to the need to perform the LU decomposition of the local Jacobians, which are frozen on subsequent passes. The Jacobi (not shown) and Gauss–Seidel variants of each linear solver are approximately equivalent in overall cost and are about four times less costly than the equivalent nonlinear solver, mainly because these smoothers avoid evaluation of the nonlinear residual. Overall, a nonlinear update using a single linear multigrid W saw-tooth cycle, with four Gauss–Seidel smoothing sweeps on each grid level, requires approximately 60% of the effort of a three-stage FAS scheme. Using two linear multigrid cycles per nonlinear update results in a nonlinear update cost approximately equal to that observed with the three-stage FAS scheme for the inviscid flow case. The linear method's efficiency advantage is slightly higher in the viscous flow case, since the nonlinear residual is now augmented by the additional viscous terms which must be computed, while the linear smoother remains identical in cost since the stencil is unchanged from the inviscid case.

The inviscid test case consists of flow over a NACA 0012 airfoil at a Mach number of 0.8 and an incidence of 1.25°. This well-known test case produces a strong upper surface shock and a weaker lower surface shock. The unstructured triangular mesh for this case contains a total of 7884 vertices. Figure 8 depicts the observed convergence rates for this case with the various schemes discussed above, the rates are compared in terms of CPU time required for a given level of reduction in the rms average of density residuals. A four-level W-cycle was used for both multigrid schemes. The figure shows that the linear multigrid approach, using a single W-cycle with four Gauss–Seidel (GS) smoothing passes, is approximately twice as efficient as the three-stage FAS scheme. The increase in efficiency between the GS and Jacobi linear multigrid algorithms is due to the superior convergence properties of GS over Jacobi, since both sweeps require approximately the same amount of CPU time. The

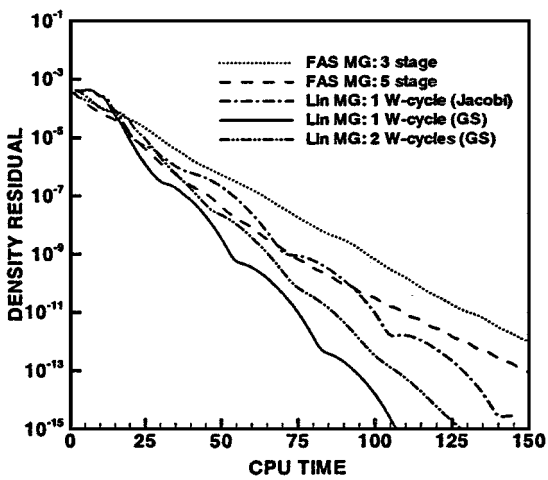

FIG. 8. Convergence efficiencies for various multigrid algorithms for the inviscid flow problem.

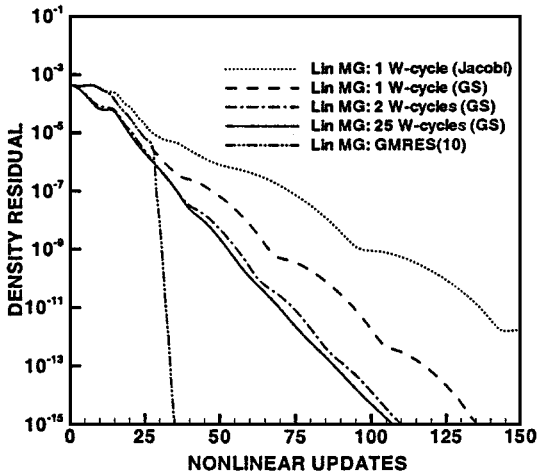

FIG. 9. Nonlinear convergence rates for various levels of the linear system solution and for the linear MG Newton–Krylov method.

five-stage FAS scheme is slightly more efficient than the three-stage scheme as much of the local Jacobian LU decomposition and multigrid overhead is amortized over more grid sweeps.

Figure 9 illustrates the nonlinear convergence rates achieved for the linear multigrid scheme per nonlinear update, as a function of the number of linear multigrid cycles. The nonlinear convergence rate has a lower bound which is approached as the number of linear multigrid cycles is increased and the linear system is solved more exactly. This asymptotic rate, which is in the neighborhood of 0.78, can be viewed as a measure to which the first-order Jacobian approximates the second-order discretization. (Quadratic convergence would be observed for an exact match.) Note that only two linear W-cycles are effective at achieving most of this nonlinear convergence, although the scheme using a single linear multigrid cycle is the most efficient overall, as shown in Fig. 8. Improving the convergence past this threshold cannot be achieved with better linear solvers; it can only be achieved through a more accurate Jacobian representation.

One way to achieve this is to use a matrix-free Newton–Krylov method [9, 27, 28] to approximate the exact Jacobian of the full second-order-accurate residual. The linear multigrid solver provides a natural candidate for a preconditioner of the Newton–Krylov method. The nonlinear GMRES routine developed by Wigton and Yu [29] is employed for this purpose. This approach, which corresponds to a left-preconditioning strategy [27], also allows the use of a nonlinear solver as a preconditioner, and hence the FAS multigrid solver is also implemented as a preconditioner for GMRES. Very rapid convergence in terms of nonlinear updates is observed in Fig. 9 when the Newton–Krylov method (using 10 search directions) is applied with the linear multigrid method as a preconditioner. Figure 10 illustrates the convergence obtained with both multigrid schemes employed as solvers and as preconditioners for GMRES, using 10 search directions, in terms of CPU time. The improvement is less dramatic when measured in this manner since each nonlinear update involves 10 multigrid cycles. However, the Newton–Krylov method provides similar overall gains in efficiency for both the linear and nonlinear schemes, particularly in the asymptotic convergence region.

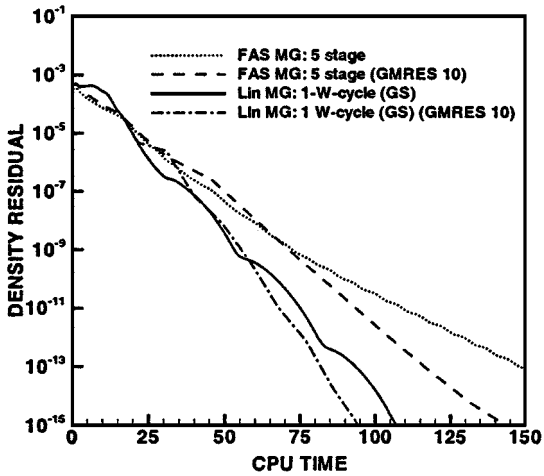

FIG. 10. Acceleration provided by the Newton–Krylov method for the inviscid flow problem.

The next test case involves the computation of viscous turbulent transonic flow over an RAE 2822 airfoil at a Mach number of 0.73, incidence of 2.31°, and a Reynolds number of 6.5×10^6 on the grid depicted in Fig. 11. This grid contains a total of 16,167 vertices and uses quadrilaterals in the highly stretched boundary layer and wake regions and triangles in isotropic regions. The linear and nonlinear line algorithms are used in this case on the set of lines depicted in Fig. 12, which were constructed using a previously developed graph algorithm [20]. The flowfield was initialized with freestream conditions and the turbulence model is converged simultaneously with the flow equations. Figure 13 illustrates the overall convergence of the various algorithms versus the number of nonlinear iterations. In this case, a single linear W-cycle using four Jacobi smoothing passes on each level provides an asymptotically faster convergence rate per cycle than either FAS scheme, while the GS

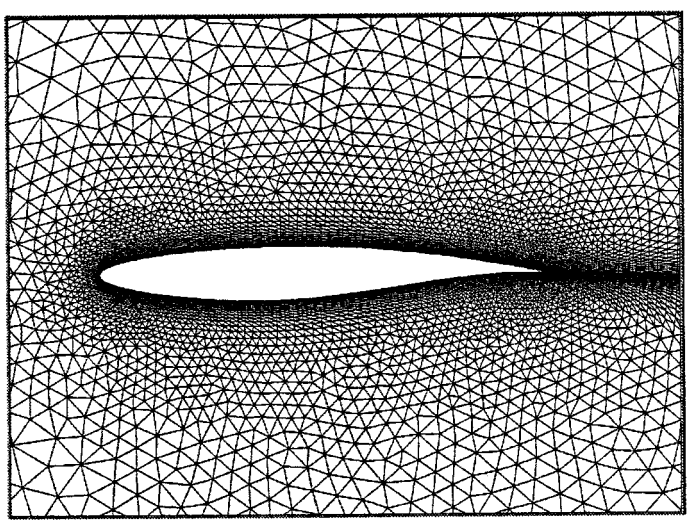

FIG. 11. Unstructured grid for viscous flow over an airfoil (16,167 points).

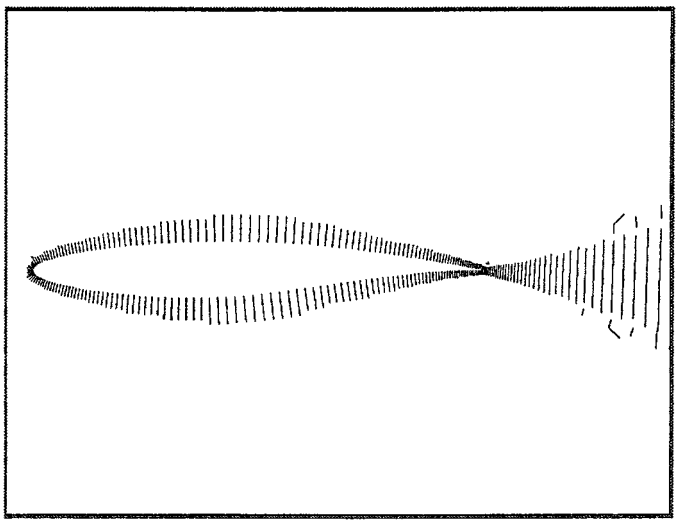

FIG. 12. Line structure for the line solver for viscous flow over an airfoil.

version of this scheme is even faster. When these schemes are compared in terms of CPU time in Fig. 14, the linear GS scheme is three times more efficient than either nonlinear FAS scheme due to the superior convergence rate as well as the lower cost per cycle of the linear multigrid scheme.

Figure 15 illustrates the increased convergence efficiency achieved when using the linear or nonlinear multigrid scheme as a preconditioner for GMRES, employing 10 search directions. Similar increases in convergence efficiency are obtained in both cases, with the nonlinear scheme benefiting slightly more than the linear scheme. However, the linear multigrid preconditioned GMRES approach remains the overall most efficient solution technique.

The final test case involves subsonic viscous flow over a multielement airfoil. The grid and associated line system are depicted in Figs. 16 and 17. This mesh contains a total

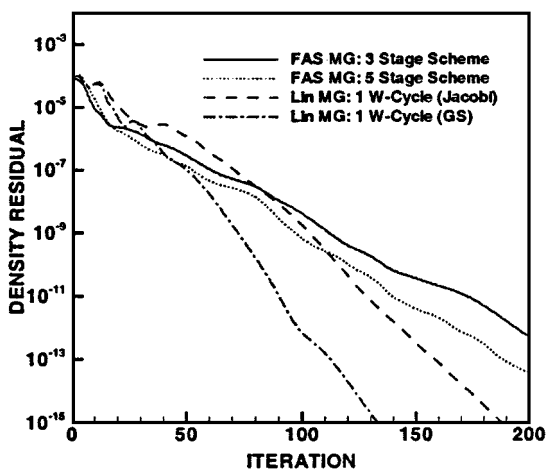

FIG. 13. Convergence efficiencies of various algorithms in terms of outer iteration cycles for the viscous airfoil flow problem.

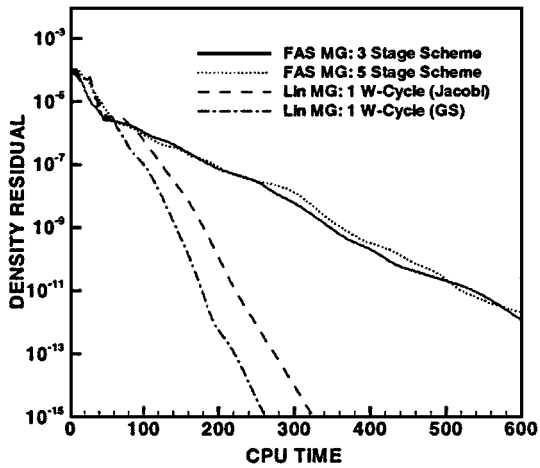

FIG. 14. Convergence efficiencies for various algorithms in terms of CPU time for the viscous airfoil flow problem.

of 61,104 vertices, with quadrilateral elements in the boundary layer and wake regions and triangular elements elsewhere. The Mach number is 0.2, the incidence is 16°, and the Reynolds number is 9×10^6. For this case, the Rieman solver is modified according to the low-Mach-number preconditioning techniques developed previously [19, 20, 30]. The final computed solution in terms of Mach number contours is depicted in Fig. 18. Complex cases of this nature have proved to be the most difficult to converge efficiently in past studies [20]. A five-level W-cycle is used in all cases for the multigrid algorithms. The flowfield is initialized with a preconverged solution obtained after 150 cycles of the FAS multigrid scheme (itself initialized from freestream conditions), and the turbulence model is frozen at its final converged values throughout these computations. This is done to focus on the asymptotic convergence behavior of the linear versus nonlinear methods and to avoid the complications of nonlinear continuation, which is required in this case for the linear solver operating on a freestream initialization.

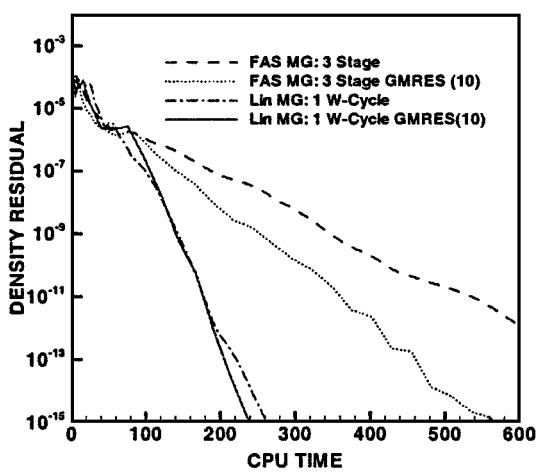

FIG. 15. Efficiency gains of the Newton–Krylov method for linear and nonlinear multigrid algorithms for the viscous airfoil flow problem.

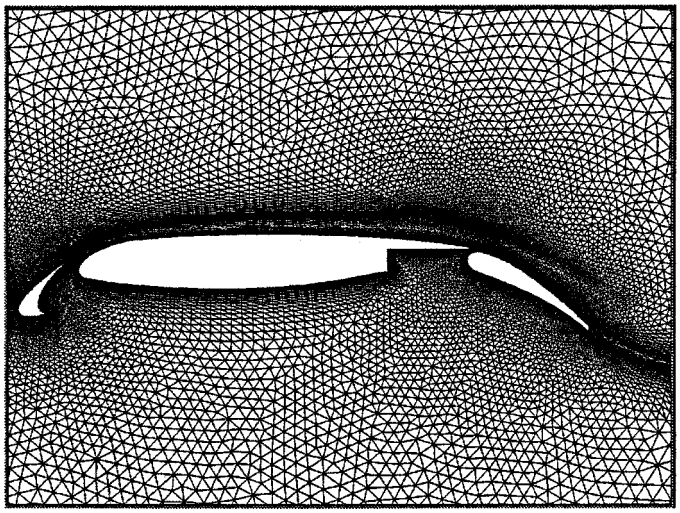

FIG. 16. Unstructured grid for viscous flow over a three-element airfoil (61,104 points).

In Fig. 19 the relative convergence efficiencies of the linear and nonlinear methods are displayed as functions of the number of nonlinear iterations. Although the linear multigrid scheme using a single W-cycle (with four GS smoothing passes) initially converges faster than the three-stage FAS scheme; the latter achieves a slightly faster asymptotic rate of convergence. However, both methods are almost equivalent asymptotically in terms of CPU time since the linear multigrid method provides lower cost multigrid sweeps, as shown in Fig. 20. In both cases, the overall convergence rate is over six times slower than that observed in the previous two cases. Solving the linear system to completion at each nonlinear update (using 20 linear W-cycles) produces no observable benefit in nonlinear convergence rate, as depicted in Fig. 19. This indicates that the slower convergence in this case is attributable

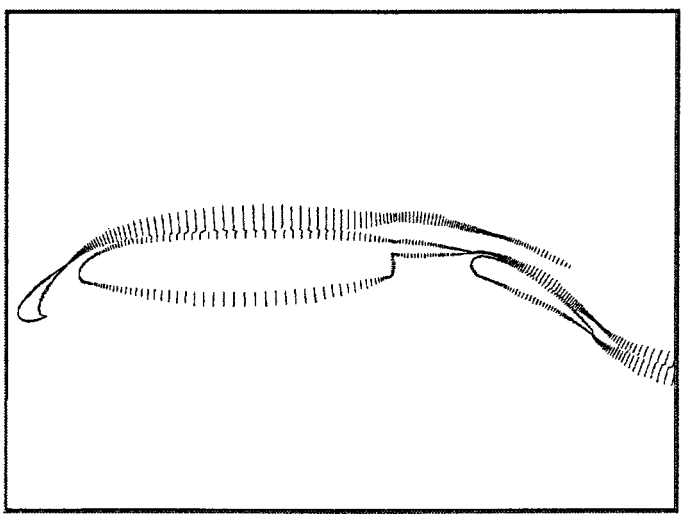

FIG. 17. Implicit lines for viscous flow over a three-element airfoil.

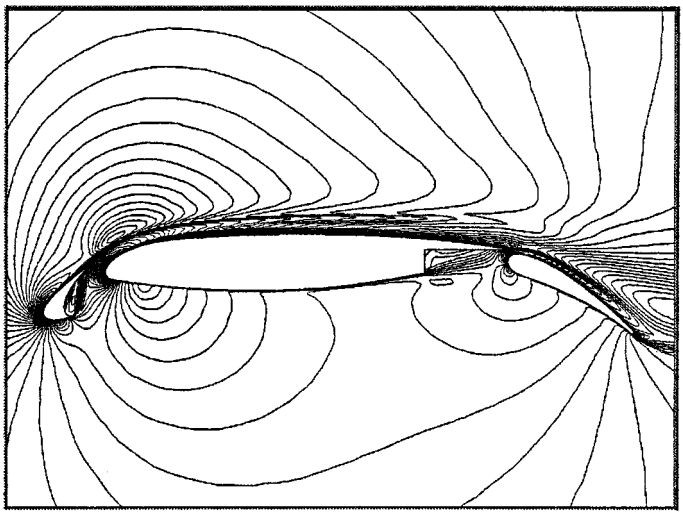

FIG. 18. Computed Mach contours for viscous flow over a three-element airfoil.

to a poor approximation of the full Jacobian by the reduced first-order Jacobian used in the linear multigrid scheme. Using either the nonlinear or the linear multigrid solver as a preconditioner for GMRES, with 20 search directions, produces a sizable increase in speed of convergence, as shown in Fig. 21. However, the speedup is more pronounced in the case of the linear multigrid solver, where the Krylov method produces a factor of 2.5 increase in the overall convergence per CPU time.

Upon initiating the Krylov method, a large jump in the residuals is observed because the current Newton–Krylov method operates on the preconditioned residual (i.e., left preconditioning). When the convergence history is plotted in terms of the preconditioned residual, which corresponds to the nonlinear corrections produced by the multigrid scheme, a monotonic behavior is observed. However, as can be seen from these two plots, any

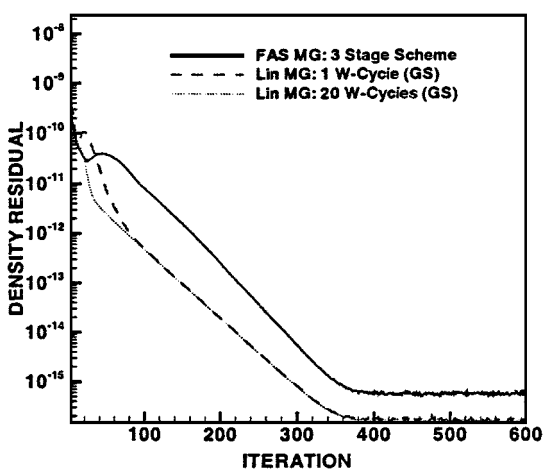

FIG. 19. Convergence rates for linear and nonlinear multigrid algorithms in terms of outer nonlinear iterations for the three-element airfoil flow problem.

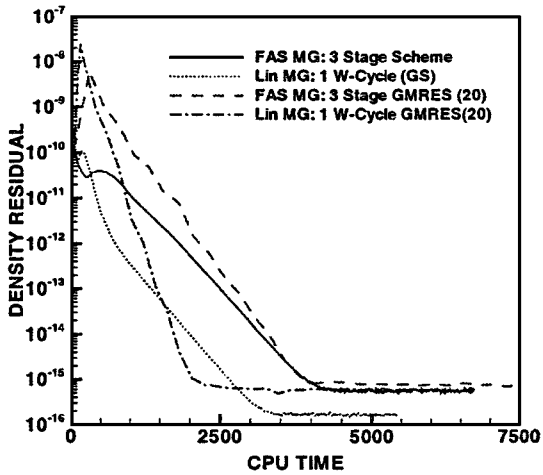

FIG. 20. Convergence efficiencies for various algorithms for the three-element airfoil flow problem.

conclusion concerning the relative solution efficiencies of the various schemes may be a function of the particular measure of convergence.

Table IV lists the asymptotic convergence rates achieved by the linear multigrid scheme for all three cases when the linear system is solved to completion at each nonlinear update. This represents a lower limit achievable with the multigrid schemes as solvers and is due to the difference between the true Jacobian of the second-order discretization and the approximate first-order Jacobian employed in the linearization for the linear multigrid scheme, which is also used in the local Jacobians and coarse levels for the FAS multigrid scheme.

This rate is seen to be substantially slower for the last case, indicating that convergence difficulties in this case cannot be addressed through improved linear multigrid methods or, for that matter, any linear solver based on the first-order Jacobian. Hence, improved restriction, prolongation, coarse grid operators, or agglomeration techniques will have little

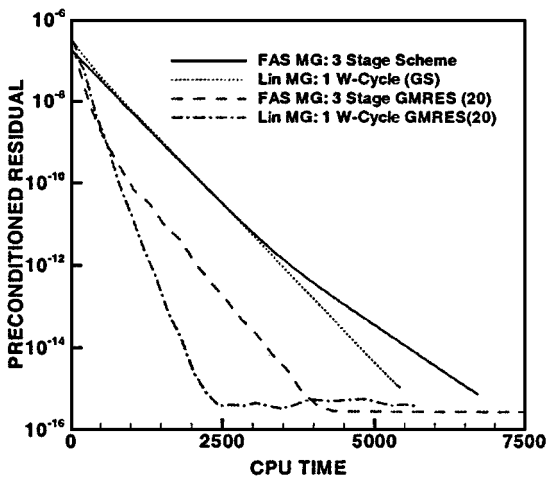

FIG. 21. Convergence efficiencies for various algorithms for the three-element airfoil flow problem as measured by preconditioned residual history.

TABLE IV
Asymptotic Convergence Rates Observed for Various Cases When the Linear System Is Solved to Completion at Each Nonlinear Iteration

Case	Asymptotic rate
NACA 0012 Euler transonic	0.78
RAE 2822 NS transonic	0.76
Multielement NS subsonic	0.965

effect in this case, and future research should concentrate on either better full Jacobian approximations or improved Krylov methods.

6. CONCLUSIONS

The preceding examples demonstrate how linear multigrid methods can deliver superior asymptotic convergence efficiency over nonlinear multigrid methods for fluid flow or radiation diffusion problems. When exact Jacobians are available, similar asymptotic convergence rates per multigrid cycle are observed for equivalent linear and nonlinear multigrid methods. The efficiency gains of the linear methods are largely attributed to the reduced number of costly nonlinear residual evaluations required and to the ability to employ a linear Gauss–Seidel smoother in the place of a Jacobi smoother. Therefore, in cases where costly or complicated nonlinear discretizations are employed, the use of linear methods can be advantageous.

Additional convergence acceleration can be achieved by using both linear and nonlinear methods as preconditioners to a Newton–Krylov method. This approach is particularly beneficial in cases where an inaccurate linearization is employed by the multigrid solvers.

These conclusions only apply to the solution efficiency in regions of monotonic asymptotic nonlinear convergence and when globalization methods are not required. While many practical cases exist (particularly for time-dependent problems) where this behavior is observed, the issues of nonlinear convergence and robustness have not been addressed herein and may affect the performance of a nonlinear method over a linear method. Furthermore, the required Jacobian storage for the linear multigrid approach can be prohibitive for many applications, particularly in three dimensions.

This study is to be extended into three dimensions in the near future and to parallel computer environments. While the overall comparisons can be expected to be similar in three dimensions, evidence shows that linear methods may suffer more efficiency degradation on parallel machines due to the larger number of cheaper grid sweeps employed, which has the effect of raising the communication-to-computation ratio [31].

REFERENCES

1. A. Brandt, Multigrid techniques with applications to fluid dynamics: 1984 guide, in *VKI Lecture Series*, (von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 1984), p. 1.
2. W. Hackbush, *Multigrid Methods and Applications* (Springer-Verlag, Berlin, 1985).
3. A. Jameson, Solution of the Euler equations by a multigrid method, *Appl. Math. Comput.* **13**, 327 (1983).

4. J. L. Thomas, D. L. Bonhaus, and W. K. Anderson, *An (O) (nm^2) Plane Solver for the Compressible Navier–Stokes Equations*, Meeting Paper 99-0785 (AIAA Press, Washington, DC, 1999).
5. G. Carre, An implicit multigrid method by agglomeration applied to turbulent flows, *Comput. Fluids* **26**, 299 (1997).
6. M. Raw, *Robustness of Coupled Algebraic Multigrid for the Navier–Stokes Equations*, Technical Paper 96-0297 (AIAA Press, Washington, DC, 1996).
7. V. A. Mousseau, D. A. Knoll, and W. J. Rider, Physics-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion, *J. Comput. Phys.* **160**, 743 (2000).
8. P. N. Brown and C. S. Woodward, *Preconditioning Strategies for Fully Implicit Radiation Diffusion With Material Energy Transfer*, Technical Report UCRL-JC-139087 (Lawrence Livermore National Laboratory, 2000).
9. E. J. Nielsen, W. K. Anderson, R. W. Walters, and D. E. Keyes, Application of Newton–Krylov methodology to a three-dimensional unstructured Euler code, in *Proceedings of the 12th AIAA CFD Conference, San Diego, CA, June 1995*, AIAA Paper 95-1733-CP.
10. D. J. Mavriplis and S. Pirzadeh, Large-scale parallel unstructured mesh computations for 3D high-lift analysis, *AIAA J. Aircraft* **36**, 987 (1999).
11. D. J. Mavriplis, *On Convergence Acceleration Techniques for Unstructured Meshes*, Meetings Paper 98-2966. (AIAA Press, Washington, DC, 1998).
12. M. Lallemand, H. Steve, and A. Dervieux, Unstructured multigridding by volume agglomeration: Current status, *Comput. Fluids* **21**, 397 (1992).
13. W. A. Smith, Multigrid solution of transonic flow on unstructured grids, in *Recent Advances and Applications in Computational Fluid Dynamics. Proceedings of the ASME Winter Annual Meeting November 1990*, edited by O. Baysal (ASME, New York, 1990).
14. V. Venkatakrishnan and D. J. Mavriplis, Agglomeration multigrid for the three-dimensional Euler equations, *AIAA J.* **33**, 633 (1995).
15. D. J. Mavriplis, Multigrid techniques for unstructured meshes, in *VKI Lecture Series VKI-LS 1995-02* (von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 1995).
16. J. W. Ruge and K. Stüben, Algebraic multigrid, in *Multigrid Methods*, edited by S. F. McCormick (Soc. for Industr. & Appl. Math., Philadelphia, 1987), pp. 73–131.
17. N. Pierce, M. Giles, A. Jameson, and L. Martinelli, Accelerating three-dimensional Navier–Stokes calculations, in *Proceedings of the 13th AIAA CFD Conference, Snowmass Village, CO, June 1997*, AIAA Paper 97-1953.
18. B. van Leer, W. T. Lee, and P. Roe, Characteristic time-stepping or local preconditioning of the Euler equations, in *Proceedings of the 10th AIAA CFD Conference, Honolulu, HI, June 1991*, AIAA Paper 91-1552-CP.
19. E. Turkel, Preconditioning-squared methods for multidimensional aerodynamics, in *Proceedings of the 13th AIAA CFD Conference, Snowmass, CO, June 1997*, AIAA Paper 97-2025-CP.
20. D. J. Mavriplis, Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes, *J. Comput. Phys.* **145**, 141 (1998).
21. D. A. Knoll, W. J. Rider, and G. L. Olson, An efficient nonlinear solution method for nonequilibrium radiation diffusion, *J. Quant. Spectrosc. Radiat. Transfer* **63**, 15 (1999).
22. P. L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, *J. Comput. Phys.* **43**, 357 (1981).
23. P. R. Spalart and S. R. Allmaras, A. one-equation turbulence model for aerodynamic flows, *Rech. Aerosp.* **1**, 5 (1994).
24. D. J. Mavriplis, Directional agglomeration multigrid techniques for high-Reynolds number viscous flows, *AIAA J.* **37**, 1222 (1999).
25. B. van Leer, C. H. Tai, and K. G. Powell, *Design of Optimally-Smoothing Multi-stage Schemes for the Euler Equations*, Technical Paper 89-1933 (AIAA Press, Washington, DC, 1989).
26. L. Martinelli and A. Jameson, *Validation of a Multigrid Method for the Reynolds*-Averaged Navier–Stokes Equations, Technical Paper 88-0414 (AIAA Press, Washington, DC, 1988).

27. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Series in Computer Science (PWS Publishing Co. Boston, 1996).
28. V. Venkatakrishnan and D. J. Mavriplis, Implicit solvers for unstructured meshes, *J. Comput. Phys.* **105**, 83 (1993).
29. L. B. Wigton, N. J. Yu, and D. P. Young, GMRES acceleration of computational fluid dynamic codes, in *Proceedings of the 7th AIAA CFD Conference, July 1985*, AIAA Paper 85-1494-CP.
30. D. Jespersen, T. Pulliam, and P. Buning, *Recent Enhancements to OVERFLOW*, Technical Paper 97-0644 (AIAA Press, Washington, DC, 1997).
31. L. Stals, The parallel solution of radiation transport equations, in *Proceedings of the Tenth SIAM Conference on Parallel Processing for Scientific Computing, Portsmouth, VA, March 2001* (SIAM, Philadelphia, 2001).