



AIAA 2001–2673

How Many Steps are Required to Solve the Euler Equations of Steady, Compressible Flow: In Search of a Fast Solution Algorithm

Antony Jameson
Stanford University
Stanford, CA 94305

D. A. Caughey
Cornell University
Ithaca, NY 14853–7501

15 Computational Fluid Dynamics Conference
June 11–14, 2001/Anaheim, CA

How Many Steps are Required to Solve the Euler Equations of Steady, Compressible Flow: In Search of a Fast Solution Algorithm

Antony Jameson

Department of Aeronautics and Astronautics
Stanford University
Stanford, California 94305-4035

David A. Caughey

Sibley School of Mechanical and Aerospace Engineering
Cornell University
Ithaca, New York 14853-7501

Abstract

New versions of implicit algorithms are proposed for the efficient solution of the Euler equations of compressible flow. The methods are based on a preconditioned, Lower-Upper (*LU*) implementation of a nonlinear, Symmetric Gauss-Seidel (*SGS*) algorithm for use as a smoothing algorithm in a multigrid method. The methods have been implemented for flows in quasi-one-dimensional ducts and for two dimensional flows past airfoils on boundary-conforming “O”-type grids for a variety of Symmetric Limited Positive (SLIP) spatial approximations, including the scalar dissipation and Convective Upwind Split Pressure (CUSP) schemes. The method is demonstrated to be significantly faster than earlier explicit or implicit methods for this class of problems, allowing solution of these problems to the level of truncation error in three to five multigrid cycles.

1 Introduction

Multigrid methods [11] using both explicit [14, 19] and implicit [9, 10, 16, 21, 27, 26] time-stepping algorithms are widely used for solving the Euler and Navier-Stokes equations for aerodynamic flows of practical interest. Although these methods have been quite intensively studied in the past several decades, the best methods developed to date still require many tens, if not hundreds, of multigrid cycles for adequate convergence. On the other hand, it is well known that elliptic problems are often solvable in ten, or fewer, cycles [5, 6]. This fact has motivated recent efforts to achieve “textbook” multigrid efficiency for non-elliptic problems [22, 23, 24]. Considering that the solution of a nonlinear equation for one unknown by Newton’s method will typically require three to five iterations, the present authors

have held the view that one should not expect to solve the large number of coupled nonlinear equations which result from discretization of the flow equations in many fewer than ten iterations. We felt that this was a target that might be achievable and have continued to search for methods capable of accomplishing it. The results presented in this paper suggest that it may, indeed, be possible to surpass this target, at least for inviscid, transonic flows. In fact, it appears that adequately converged solutions of the Euler equations for transonic flows past airfoils can be obtained in three to five multigrid “W” cycles.

The present collaboration stems from a presentation given by the first author at the conference in Half Moon Bay, California, in honor of Robert W. MacCormack’s 60th birthday, which raised the question of whether the potential of Symmetric Gauss-Seidel (*SGS*) algorithms had been fully exploited. Since both of the authors have had experience with *LU* implicit and *SGS* schemes from time to time since 1981 [8, 15, 16, 21], we agreed to work together to see if we could improve their performance.

The principal ingredients of our new scheme are as follows. First, we have adopted the *LU-SGS* scheme in a fully nonlinear form, in which the fluxes at each cell interface are recomputed for each cell using the most recently updated values of the flow solution variables while sweeping forward from left to right, and then backward from right to left. Second, the combined Jacobian matrices that would appear on the diagonal of a linearized implicit, flux-split scheme are used as a preconditioner of the nonlinear scheme. Third, we observed a slower rate of convergence in the supersonic zones, so we introduced options for additional sweeps only over cells in which the local Mach number was supersonic. (In the case of the Burgers equation we also have found that it can pay to use additional iterations of the interface fluxes in each cell before advancing to the next cell.)

⁰Copyright©2001 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

Finally, we combine this evolution algorithm with a fully nonlinear multigrid algorithm of the type formulated by Jameson [11], which uses the full-approximation scheme (*FAS*) introduced by Brandt [5]. The multigrid algorithm provides options for “V” and “W” cycles, and is set up as a full multigrid scheme in which the calculation is initiated on a coarse grid before proceeding to successively finer grids in each of which the solution process is accelerated by cycles through coarser grids. At each stage of this process a second-order accurate discretization is employed on the (currently) finest grid, while a first order accurate discretization is used on all coarser grids.

A fast rate of convergence results from the effective combination of all these features, which are analyzed in more detail in Section 2. Results for both quasi-one-dimensional nozzle flows and transonic flows past two-dimensional airfoils are presented in Section 3. In that section we also show comparisons of the results obtained with different spatial discretizations of the flow equations.

2 Analysis

We consider the construction of implicit schemes to solve the conservation law

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

If we define Jacobian matrices

$$A = \frac{\partial f}{\partial w}, \quad B = \frac{\partial g}{\partial w}, \quad (2)$$

the correction vector

$$\delta w = w^{n+1} - w^n, \quad (3)$$

and replace $f(w^{n+1})$ and $g(w^{n+1})$ by the approximations

$$f^{n+1} \sim f^n + A\delta w, \quad g^{n+1} \sim g^n + B\delta w, \quad (4)$$

we have the prototype implicit scheme

$$\{I + \mu\Delta t (D_x A + D_y B)\} \delta w + \Delta t R = 0, \quad (5)$$

where $\mu \geq \frac{1}{2}$ and

$$R = D_x f(w) + D_y g(w) + \text{artificial dissipation terms} \quad (6)$$

is the residual. If $\mu = 1$, this becomes a Newton iteration to solve the steady-state equation in the limit as $\Delta t \rightarrow \infty$.

A flux-split version of the fully implicit version ($\mu = 1$) of Eq. (5) can be written

$$\{I + \Delta t (D_x^+ A^- + D_y^+ B^-) + \Delta t (D_x^- A^+ + D_y^- B^+)\} \delta w + \Delta t R = 0, \quad (7)$$

where D_x^+, D_y^+ are forward differences, D_x^-, D_y^- are backward differences, and

$$R = D_x^+ f^- + D_y^+ g^- + D_x^- f^+ + D_y^- g^+. \quad (8)$$

The fluxes are split so that the Jacobians

$$A^+ = \frac{\partial f^+}{\partial w} \quad \text{and} \quad B^+ = \frac{\partial g^+}{\partial w}$$

have positive eigenvalues, while

$$A^- = \frac{\partial f^-}{\partial w} \quad \text{and} \quad B^- = \frac{\partial g^-}{\partial w}$$

have negative eigenvalues.

To reduce the computation required for each time step one can approximate the left hand side of Eq. (5) by a product of more easily invertible factors. Historically, this is done either by introducing an Alternating Direction Implicit (*ADI*) factorization or a Lower-Upper (*LU*) factorization. The *ADI* factorization [3, 4, 7], results in

$$(I + \mu\Delta t D_x A)(I + \mu\Delta t D_y B) \delta w + \Delta t R = 0. \quad (9)$$

which requires the solution of a sequence of block tridiagonal systems. The method can be made more efficient by diagonalizing the factors (see [20, 9]). The method is unconditionally stable for linear problems in two dimensions when $\mu \geq 1/2$, but is linearly unstable in three dimensions.

The *LU* factorization [15, 8, 17, 26], results in

$$\{I + \Delta t (D_x^+ A^- + D_y^+ B^-)\} \{I + \Delta t (D_x^- A^+ + D_y^- B^+)\} \delta w + \Delta t R = 0. \quad (10)$$

Here, as earlier, A^+, B^+ are Jacobians having positive eigenvalues, and A^-, B^- are Jacobians having negative eigenvalues, with

$$A^+ + A^- = A, \quad B^+ + B^- = B. \quad (11)$$

The *LU* scheme requires the solution only of block bidiagonal factors for each time step.

A symmetric Gauss Seidel iteration of an implicit scheme can be constructed as follows; to simplify the notation at this point only the one-dimensional problem will be considered. Consider the flux split scheme

$$\{I + r (\delta_x^+ A^- + \delta_x^- A^+)\} \delta v + \Delta t R = 0, \quad (12)$$

where $\delta_x^+ = \Delta x D^+$, $\delta_x^- = \Delta x D^-$ are forward and backward undivided differences, $r = \frac{\Delta t}{\Delta x}$ and R is the residual. Alternatively, we can write

$$\begin{aligned} \delta v_i + r (A_{i+1}^- \delta v_{i+1} - A_i^- \delta v_i + \\ A_i^+ \delta v_i - A_{i-1}^+ \delta v_{i-1}) + \Delta t R_i = 0. \end{aligned} \quad (13)$$

A symmetric Gauss-Seidel scheme consists of the forward sweep

$$\delta \tilde{v}_i + r (A_i^+ - A_i^-) \delta \tilde{v}_i - r A_{i-1}^+ \delta \tilde{v}_{i-1} + \Delta t R_i = 0, \quad (14)$$

followed by the backward sweep

$$\begin{aligned} \delta v_i + r (A_i^+ - A_i^-) \delta v_i + r A_{i+1}^- \delta v_{i+1} - \\ r A_{i-1}^+ \delta v_{i-1} + \Delta t R_i = 0. \end{aligned} \quad (15)$$

If we subtract (14) from (15) we obtain

$$\begin{aligned} \{I + r (A_i^+ - A_i^-)\} \delta v_i + r A_{i+1}^- \delta v_{i+1} = \\ \{I + r (A_i^+ - A_i^-)\} \delta \tilde{v}_i, \end{aligned} \quad (16)$$

which can be written as

$$LD^{-1}U\delta v = -\Delta t R, \quad (17)$$

where

$$\begin{aligned} L &\equiv I - rA^- + r\delta_x^- A^+, \\ U &\equiv I + rA^+ + r\delta_x^+ A^-, \\ D &\equiv I + r(A^+ - A^-). \end{aligned} \quad (18)$$

If we take

$$A^+ = \frac{1}{2}(A + \rho I), \quad A^- = \frac{1}{2}(A - \rho I), \quad (19)$$

where $\rho = \max |\lambda(A)|$ (to ensure diagonal dominance) then D reduces to a scalar factor and this is a variation of the LU scheme.

The terms $\Delta t R_i - r A_{i-1}^+ \delta \tilde{v}_{i-1}$ of Eq. (14) are a linearization of $\Delta t R_i$ evaluated with $\tilde{v}_{i-1} = v_{i-1} + \delta \tilde{v}_{i-1}$. Following this line of reasoning, the LU - SGS scheme can be recast as

$$\{I + r (A_i^+ - A_i^-)\} \delta \tilde{w}_i + \Delta t \tilde{R}_i = 0; \quad (20)$$

$$\{I + r (A_i^+ - A_i^-)\} \delta \tilde{\tilde{w}}_i + \Delta t \tilde{\tilde{R}}_i = 0, \quad (21)$$

where

$$\tilde{w}_i = w_i + \delta \tilde{w}_i; \quad \tilde{f}_i^\pm = f^\pm(\tilde{w}_i); \quad (22)$$

$$w_i^{n+1} = \tilde{\tilde{w}}_i = \tilde{w}_i + \delta \tilde{\tilde{w}}_i; \quad \tilde{\tilde{f}}_i^\pm = f^\pm(\tilde{\tilde{w}}_i); \quad (23)$$

and

$$\tilde{R}_i = \frac{1}{\Delta x} (f_{i+1}^- - f_i^- + f_i^+ - \tilde{f}_{i-1}^+), \quad (24)$$

$$\tilde{\tilde{R}}_i = \frac{1}{\Delta x} (\tilde{\tilde{f}}_{i+1}^- - \tilde{\tilde{f}}_i^- + \tilde{\tilde{f}}_i^+ - \tilde{\tilde{f}}_{i-1}^+). \quad (25)$$

With the definitions of Eq. (19), Eqs. (20) and (21) can be written

$$\delta \tilde{w}_i = -\frac{\Delta t}{1 + \mathcal{C}} \tilde{R}_i \quad (26)$$

$$\delta \tilde{\tilde{w}}_i = -\frac{\Delta t}{1 + \tilde{\mathcal{C}}} \tilde{\tilde{R}}_i \quad (27)$$

where $\mathcal{C} = \rho \Delta t / \Delta x$ is the Courant number. Alternatively, with the Jacobian splitting defined as

$$A^+ = \frac{1}{2}(A + |A|), \quad A^- = \frac{1}{2}(A - |A|), \quad (28)$$

where $|A| = M|A|M^{-1}$, with $|A|$ the diagonal matrix whose entries are the absolute values of the eigenvalues of the Jacobian matrix A and M , M^{-1} are the modal matrix of A and its inverse, Eqs. (20) and (21) can be written

$$\{I + r|A|\} \delta \tilde{w}_i = -\Delta t \tilde{R}_i \quad (29)$$

$$\{I + r|A|\} \delta \tilde{\tilde{w}}_i = -\Delta t \tilde{\tilde{R}}_i \quad (30)$$

and, in the limit as the time step Δt goes to infinity, these equations represent the SGS Newton iteration

$$|A| \delta \tilde{w}_i = -\Delta x \tilde{R}_i \quad (31)$$

$$|A| \delta \tilde{\tilde{w}}_i = -\Delta x \tilde{\tilde{R}}_i \quad (32)$$

The introduction of the splitting defined by Eqs. (28) is motivated, in part, by the success of the similar preconditioner introduced by Allmaras [1] and used by Pierce and Giles [19] to accelerate the convergence of codes based on explicit Runge-Kutta time stepping. This preconditioner seems to have its roots in the Diagonally-Dominant ADI scheme [2, 18].

The scheme corresponding to Eqs. (31) and (32) is implemented for the finite-volume form [14] of the equations, which can be represented (in two dimensions) as

$$\{|A| + |B|\} \delta \tilde{w}_{i,j} = -\sigma \tilde{R}_{i,j}, \quad (33)$$

$$\{|A| + |B|\} \delta \tilde{\tilde{w}}_{i,j} = -\sigma \tilde{\tilde{R}}_{i,j}, \quad (34)$$

where

$$\begin{aligned} \tilde{R}_{i,j} &= F_{i+1,j}^- - F_{i,j}^- + F_{i,j}^+ - \tilde{F}_{i-1,j}^+ + \\ &G_{i,j+1}^- - G_{i,j}^- + G_{i,j}^+ - \tilde{G}_{i,j-1}^+ \end{aligned} \quad (35)$$

and

$$\begin{aligned} \tilde{\tilde{R}}_{i,j} &= \tilde{\tilde{F}}_{i+1,j}^- - \tilde{\tilde{F}}_{i,j}^- + \tilde{\tilde{F}}_{i,j}^+ - \tilde{\tilde{F}}_{i-1,j}^+ + \\ &\tilde{\tilde{G}}_{i,j+1}^- - \tilde{\tilde{G}}_{i,j}^- + \tilde{\tilde{G}}_{i,j}^+ - \tilde{\tilde{G}}_{i,j-1}^+ \end{aligned} \quad (36)$$

and σ is a relaxation factor that can be used to optimize convergence rates. In these equations F^+ , F^- , G^+ , and G^- represent the split approximations to the cell area h times the contravariant components of the flux vectors in the corresponding mesh coordinate directions. The residual fluxes are approximated using either the scalar or Convective Upwind Split Pressure (CUSP) versions of the Symmetric Limited Positive (SLIP) approximations developed by Jameson [12, 13].

The implementation of this procedure is made computationally very efficient by locally transforming the residuals to those corresponding to the equations written in primitive variables (see, e.g., [25]), then transforming the corrections back to the conserved variables. Numerical experiments indicate that it can be beneficial to perform additional corrections in supersonic zones, when they are present in the solution. The CPU time required for these multiple sweeps is reduced by “freezing” the matrix coefficients $|A|$ and $|B|$ that appear in Eqs. (31) and (32). The additional memory required to store these coefficient matrices is minimized by storing only the symmetrized form of the Jacobians (which requires only 7 additional quantities to be stored for each mesh cell).

Boundary conditions for two-dimensional problems are enforced in the standard way. At solid boundaries, the no-flux condition is enforced directly, and the pressure is determined from the normal momentum equation. Values of density and total energy are required in dummy cells, and are set by extrapolating the entropy and total enthalpy with zero gradients normal to the boundary. The flows considered here are subsonic at infinity, so at far field boundaries the outgoing Riemann invariant is extrapolated from the interior and the incoming Riemann invariant is set by free stream quantities.

The inflow and outflow boundary conditions for the quasi-one-dimensional nozzle flows are even more important than those for the two-dimensional problem, since it is the interplay between the outflow boundary condition on static pressure and the inflow conditions on stagnation pressure and temperature that determine the rate of flow through the nozzle. Well-posed boundary conditions are especially important for schemes that converge very rapidly to the steady state.

The following treatment has been found effective. At the inflow boundary, the value of the outgoing Riemann invariant, determined in terms of the flow speed u and speed of sound c as

$$R = u - \frac{2c}{\gamma - 1}, \quad (37)$$

is extrapolated from the interior of the domain. Since the total enthalpy

$$h_0 = \frac{c^2}{\gamma - 1} + \frac{u^2}{2} \quad (38)$$

is fixed, these two equations can be combined to give a single (quadratic) equation that can be solved for the speed of sound

$$c = \frac{\gamma - 1}{\gamma + 1} \left[\sqrt{(\gamma + 1)h_0 - \frac{\gamma - 1}{2}R^2 - 1} \right] \quad (39)$$

Equation (37) then determines the velocity u , and all other quantities can then be determined as functions of the known Mach number $\mathbf{M} = u/c$ at the inflow boundary. At the outflow boundary the pressure is specified. The entropy is then extrapolated from the interior to determine the density, and the total enthalpy is extrapolated to determine the total energy. The mass flux density is then specified to be consistent with the other variables already determined

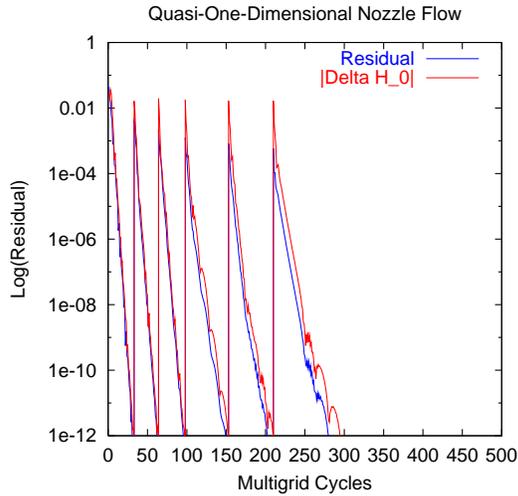
$$w_2 = \rho u = \sqrt{2\rho \left[e - \frac{p}{\gamma - 1} \right]} \quad (40)$$

The algorithm is implemented within the framework of the multigrid method, following the (now standard) procedure of Jameson [11].

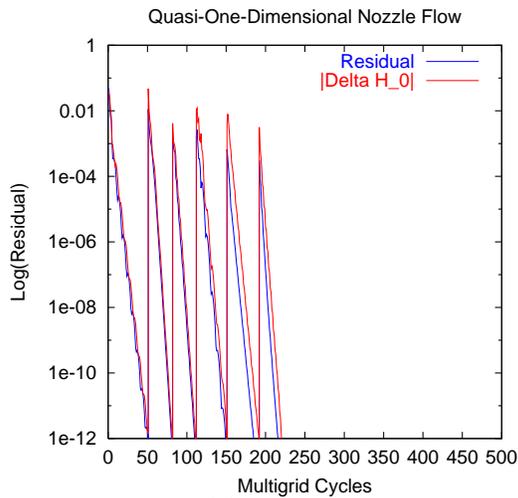
3 Results

The results of several sample computations are presented here to illustrate both, the asymptotic rate of convergence of the method, and the speed with which global measures of the solutions, such as force and moment coefficients and surface pressure distributions, converge. First, results of a quasi-one-dimensional nozzle flow will be presented, then the results of several computations for two-dimensional, transonic airfoils.

Results of a one-dimensional transonic nozzle flow are presented to illustrate the effectiveness with which information is propagated through the solution domain by the algorithm in both the upstream and downstream directions. Propagation of information in both directions is critical for this one-dimensional problem, as the flow rate is established by communication between the downstream boundary, where the static pressure is fixed, and the upstream boundary, where the stagnation pressure and temperature are fixed. The nozzle has a converging-diverging geometry, with an inlet to throat area ratio



(a) Scalar SLIP



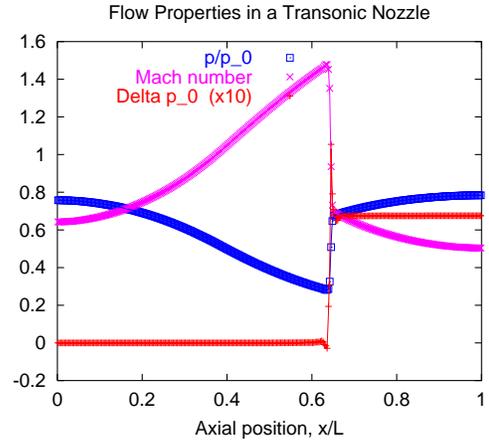
(b) CUSP

Figure 1: Convergence rates for transonic flow in a quasi-one-dimensional nozzle. Computations proceed on a sequence of six grids, with the finest grids containing 320 cells.

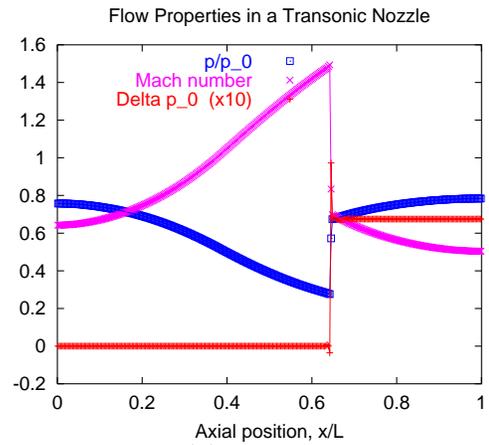
of $A_{in}/A_{th} = 1.111$, and an exit to throat area ratio of $A_{ex}/A_{th} = 1.3889$. The boundary conditions are chosen so that the inlet Mach number is approximately $M_{in} = 0.6424$, and a shock with upstream Mach number $M_1 = 1.48$ stands at approximately 65 per cent station. The convergence history for a sequence of computations on a succession of grids containing 10, 20, 40, 80, 160, and 320 uniformly-spaced grid cells is shown in Fig. 1.

The average rates of convergence for these computations are summarized in Table 1.

To illustrate the speed with which the pressure and entropy distributions are established, solutions at the end of only 3 multigrid cycles are compared with iteratively converged solutions in Fig. 2. It is clear that these solutions (on 320 cell grids) have



(a) Scalar SLIP

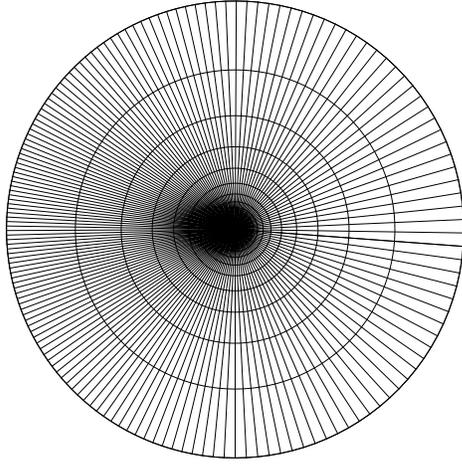


(b) CUSP

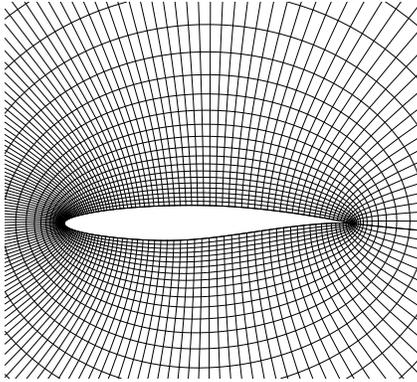
Figure 2: Comparison of fast, preconditioned multigrid solutions after only 3 multigrid cycles on fine grid with fully converged solutions on identical grid. Quasi-one-dimensional nozzle solutions are computed on 320 cell grids.

converged to within the plottable accuracy of these figures in just 3 multigrid cycles.

The results of airfoil computations presented here are computed on “O”-type grids extending from the airfoil to a far-field boundary located approximately 26 chord lengths from the body surface. Results are again presented for computations using both, the scalar SLIP and CUSP dissipation models. It is found from numerical experiments that best convergence rates are achieved when extra correction sweeps are performed locally in supersonic zones. In the present computations, three extra (bi-directional) sweeps are performed on the fine grid, and one extra (bi-directional) sweep is performed on each of the coarser grids. Corrections on the fine grid are under-relaxed slightly (typically $\sigma = 0.95$), and are over-relaxed on all coarser grids (typically



(a) Entire grid



(b) Region near body

Figure 3: Fine mesh, containing 160×32 cells in wrap-around and body normal directions, respectively, for computing the transonic flow past the RAE 2822 airfoil. (a) Complete mesh; (b) mesh in vicinity of airfoil surface.

$\sigma = 1.40$),

All results are computed using a sequence of four grids; having 20×4 , 40×8 , 80×16 , and 160×32 mesh cells in the wrap-around and body-normal directions, respectively. An uniform free stream is used as the initial condition for the solution on the coarsest grid, and interpolated coarser grid solutions are used as initial conditions for the subsequent, finer, grids. Two views of the final fine mesh used for computing the flow past the RAE 2822 airfoil are shown in Fig. 3.

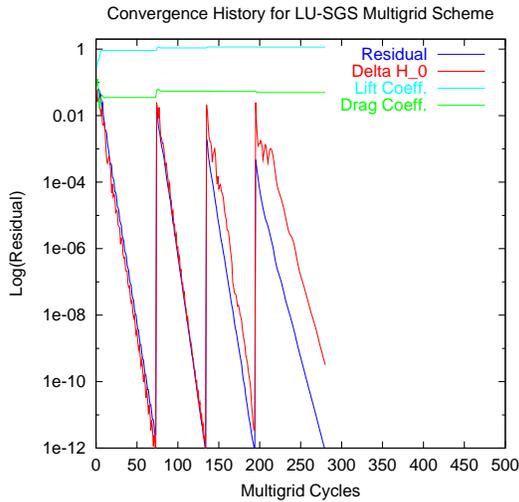
Figures 4, 5 and 6 show the convergence histories for the solution of three representative transonic flow problems; the flow past the RAE 2822 airfoil at Mach 0.75 and 3.00 degrees incidence, and the

flow past the NACA 0012 airfoil at Mach 0.80 and 1.25 degrees incidence and at Mach 0.85 and 1.00 degrees incidence. In each case, a sequence of up to 100 multigrid cycles on each of the four successive grids is shown; the iterations are halted if the residual or the fractional deviation in total enthalpy reaches 10^{-12} . Results are displayed for computations using the scalar SLIP scheme and the CUSP scheme. Both these schemes are designed to drive the total enthalpy of iteratively-converged solutions to a constant, and the fractional deviation in total enthalpy is plotted in the figures along with the residual. (Surface pressure distributions for these cases will be displayed later.) These figures demonstrate that the asymptotic rate of convergence of the method is nearly independent of problem size (i.e., number of grid cells), and that residuals approaching 64-bit round-off levels can usually be reached in fewer than 100 multigrid cycles.

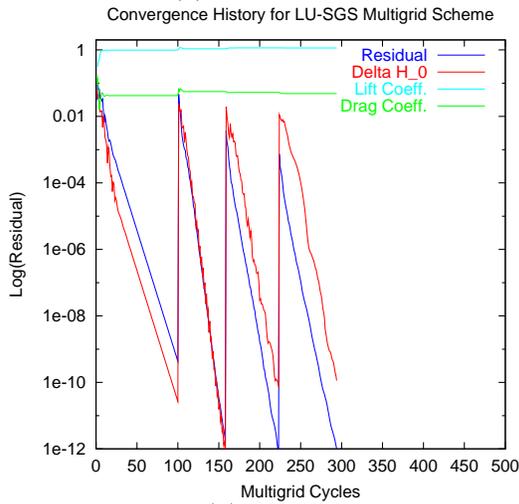
The average rates of convergence for these computations are summarized in Table 2. It should be noted that the same relaxation factors were used for all grids in these computations; better rates of convergence can be achieved if the relaxation factors are optimized for each grid.

In addition to having an impressive asymptotic rate of convergence, the method also converges the global measures of error quickly. Errors in lift and drag coefficients typically are reduced to the level of truncation error (on the order of 1 - 2% of their iteratively converged values) in 3 to 5 multigrid cycles. although more cycles may be required on the coarsest grid since it is starting from a poorer initial guess. (Of course, additional computations on the coarsest grid cost very little in CPU time.) To illustrate this rapid convergence, surface pressure distributions computed using just 3 multigrid cycles are compared with fully converged distributions in Figs. 7, 8, and 9 for the RAE 2822 airfoil and NACA 0012 airfoil test cases presented above. In each case just 3 multigrid cycles were performed on a full multigrid sequence of six successively finer grids, starting with a 5×1 cell grid. The figures demonstrate that the surface pressure distributions are well converged in just 3 cycles for these cases, although the distribution of entropy (plotted as fractional total pressure loss) are not fully converged.

In order to converge the more sensitive measures of error, such as the entropy, it appears that several more cycles are required. Surface pressure and entropy (total pressure loss) distributions computed using just 5 multigrid cycles are compared with fully converged distributions in Figs. 10, 11, and 12 for the same RAE 2822 and NACA 0012 airfoil test cases



(a) Scalar SLIP



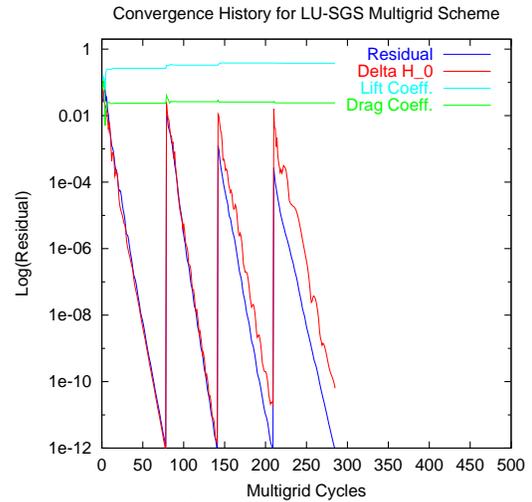
(b) CUSP

Figure 4: Convergence rates for transonic flow past the RAE 2822 airfoil at Mach 0.75 and 3.0 degrees incidence. Computations proceed on a sequence of three grids, with the finest grids containing 160×32 cells.

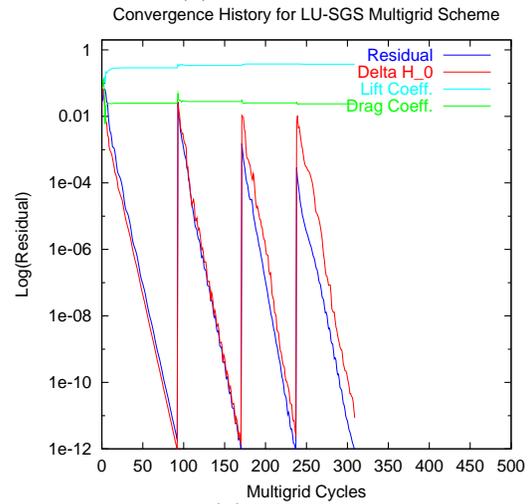
presented above. In each case just 5 multigrid cycles were performed on each of the sequence of six successively finer grids. The figures demonstrate that the surface entropy distributions are well converged in just 5 cycles for these cases.

The lift and drag coefficients for these computations are compared with their (iteratively) converged values in Tables 3 and 4. The data in the table verified that the lift and drag coefficients are within about 1% of their iteratively-converged values after only 3 multigrid cycles, and to within a fraction of 1% after only 5 cycles.

The results displayed in this section indicate that the asymptotic convergence rates of the present method are significantly faster than previous re-



(a) Scalar SLIP



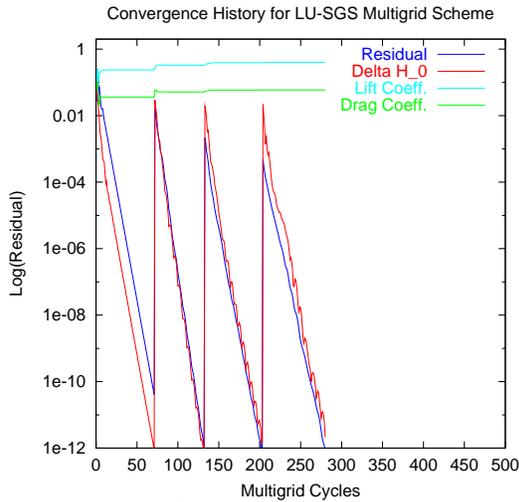
(b) CUSP

Figure 5: Convergence rates for transonic flow past the NACA 0012 airfoil at Mach 0.80 and 1.25 degrees incidence. Computations proceed on a sequence of three grids, with the finest grids containing 160×32 cells.

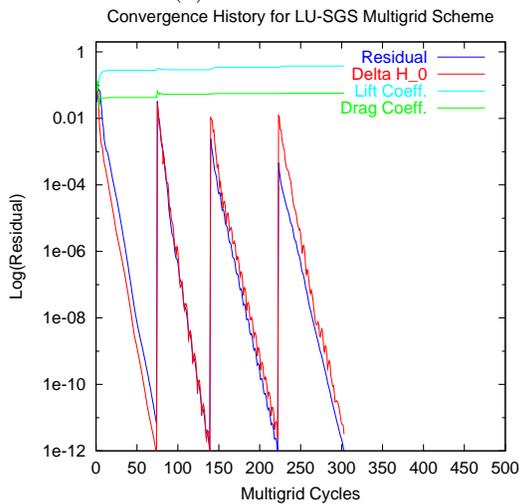
sults using the well-tuned Runge-Kutta multigrid method of Jameson [11] and the diagonalized implicit method of Caughey [9, 10]. In addition, the CPU time per multigrid cycle is as much as 50 per cent less than that of the previous methods (including those based on explicit Runge-Kutta integration), indicating that converged solutions can be obtained with the current method with almost an order of magnitude less computer time than those methods.

4 Concluding Remarks

Results of computations verifying the efficiency of a new preconditioned, *LU-SGS* implicit multigrid



(a) Scalar SLIP

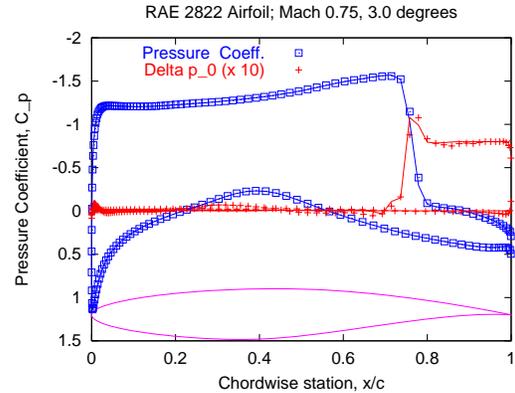


(b) CUSP

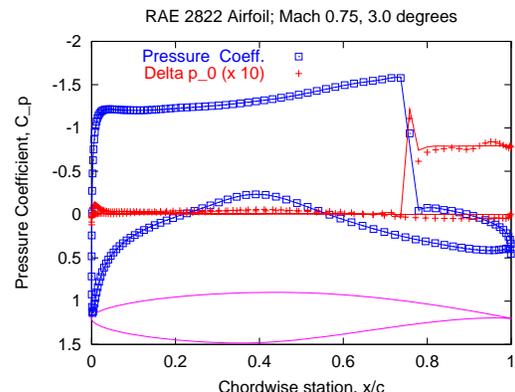
Figure 6: Convergence rates for transonic flow past the NACA 0012 airfoil at Mach 0.85 and 1.00 degrees incidence. Computations proceed on a sequence of three grids, with the finest grids containing 160×32 cells.

method for solving the Euler equations have been presented. Results presented in this paper have focused on implementations of “O”-type grids, and are computed using the scalar SLIP and CUSP dissipation spatial discretizations. The method currently is being extended to computations on “C”-type grids, which will be more suitable for solving the Navier-Stokes equations. The present results indicate that solutions, converged to approximately the level of truncation error, can be obtained in 3 - 5 multigrid cycles on fine grids for the cases tested, which are widely used benchmarks for transonic flow computations.

The results presented here approach the goal of “textbook” multigrid efficiency, at least for inviscid



(a) Scalar SLIP



(b) CUSP

Figure 7: Comparison of fast, preconditioned multigrid solutions after only 3 multigrid cycles on fine grid with fully converged solutions on identical grid. RAE 2822 airfoil at Mach 0.75 and 3.0 degrees incidence on 160×32 cell grids.

cid flows. We are working on extensions to three-dimensional flows, and to the solution of viscous flows modeled by the Reynolds-Averaged Navier-Stokes (*RANS*) equations. While we are very encouraged by the results to date for inviscid flow, we recognize that solution of the *RANS* equations present a much more severe challenge.

The progress of our work has been greatly facilitated by the extraordinary advances in computer technology over the past decade. The algorithm development and numerical tests have been carried out entirely on the laptop computers that we each possess, running the Linux OS. In fact, the Sony Vaio 505 owned by the first author, which has a 750 MHz Pentium III processor, is quite sufficient for three-dimensional flow calculations. We have benefited equally from the ease with which we can exchange information and code updates over the internet. All

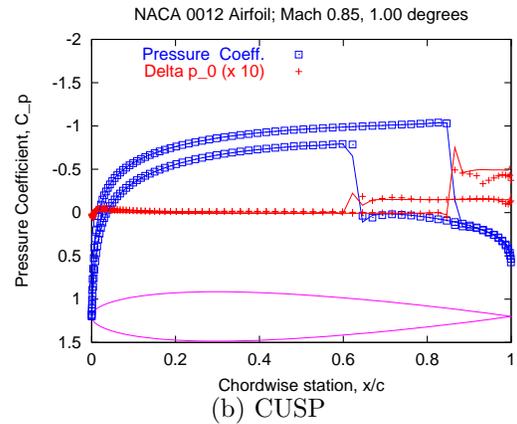
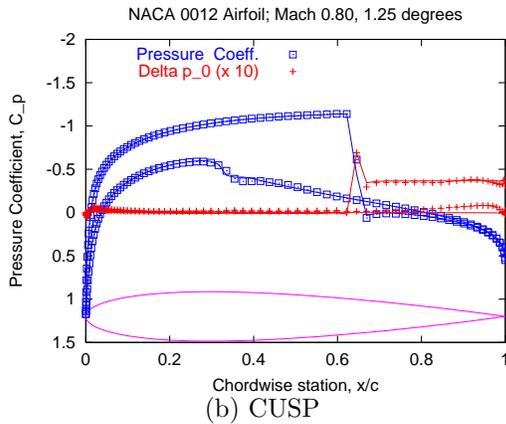
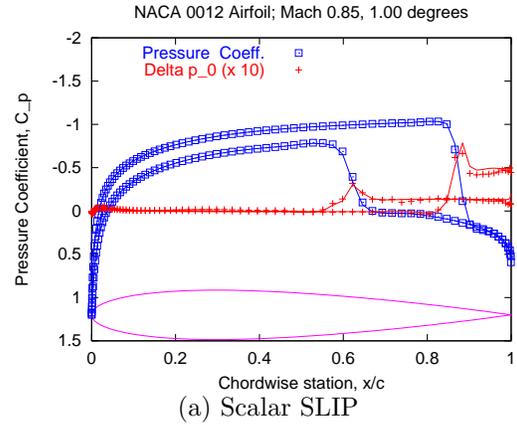
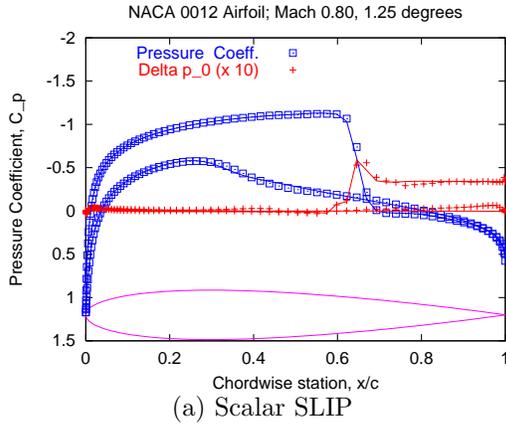


Figure 8: Comparison of fast, preconditioned multi-grid solutions after only 3 multigrid cycles on fine grid with fully converged solutions on identical grid. NACA 0012 airfoil at Mach 0.80 and 1.25 degrees incidence on 160×32 cell grids.

Figure 9: Comparison of fast, preconditioned multi-grid solutions after only 3 multigrid cycles on fine grid with fully converged solutions on identical grid. NACA 0012 airfoil at Mach 0.85 and 1.00 degrees incidence on 160×32 cell grids.

this provides an environment for algorithm development entirely different from that which prevailed during most of our careers, and which has allowed us to collaborate easily in spite of our being at opposite ends of the continent.

5 Acknowledgement

The first author has benefited greatly from the continuing support of the AFOSR under Grant No. AF F49620-98-1-022. While the principal goal of this grant is improved methods for aerodynamic shape optimization, these depend critically on the availability of fast solution algorithms, and this has been one of the prime motivations of the work described in this paper.

References

- [1] Allmaras, S. (1993). Analysis of a Local Matrix Preconditioner for the 2-D Navier-Stokes Equations, **AIAA Paper 93-3330**.
- [2] Bardina, J. & Lombard, C. K. (1987). Three-dimensional Hypersonic Flow Simulations with the CSCM Implicit Upwind Navier-Stokes Method, **AIAA Paper 87-1114**.
- [3] Beam, R. M. & Warming, R. F. (1976) An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form. **J. Comp. Phys.** Vol. 22, 87-110
- [4] Beam, R. M. & Warming, R. F. 1978 An Implicit Factored Scheme for the Compressible

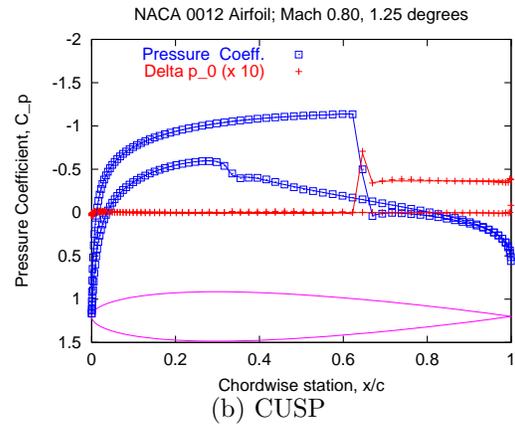
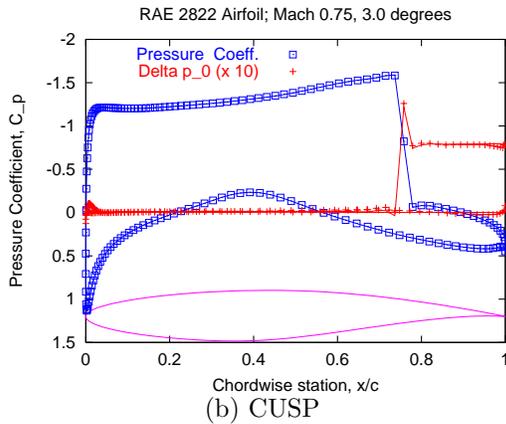
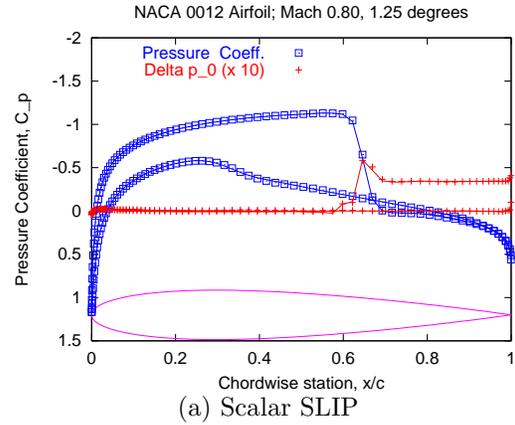
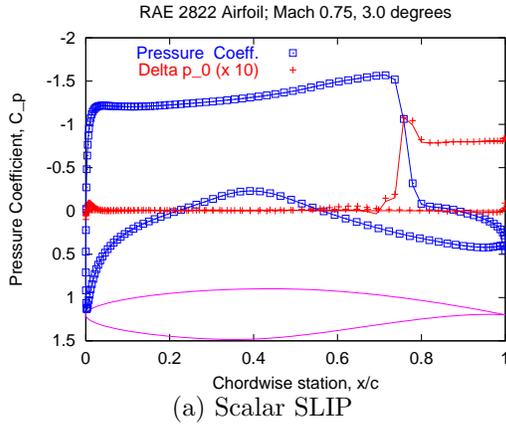


Figure 10: Comparison of fast, preconditioned multigrid solutions after only 5 multigrid cycles on fine grid with fully converged solutions on identical grid. RAE 2822 airfoil at Mach 0.75 and 3.0 degrees incidence on 160×32 cell grids.

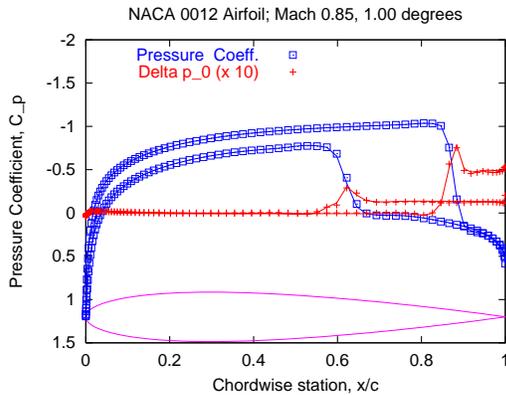
Figure 11: Comparison of fast, preconditioned multigrid solutions after only 5 multigrid cycles on fine grid with fully converged solutions on identical grid. NACA 0012 airfoil at Mach 0.80 and 1.25 degrees incidence on 160×32 cell grids.

Navier-Stokes Equations. **AIAA Journal** Vol. 16, 393–402.

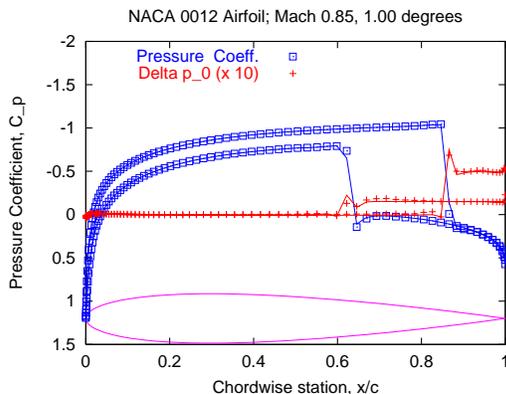
- [5] Brandt, A. (1977). Multi-Level Adaptive Solution to Boundary Value Problems, **Math. Comp.**, Vol. 31, No. 138, pp. 333-390.
- [6] Brandt, A. (1985). Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, **GMD-Studie 85**, GMD-FIT.
- [7] Briley, W. R. & McDonald, H. (1974) Solution of the Three-Dimensional Compressible Navier Stokes Equations by an Implicit Technique. **Lecture Notes in Physics** Vol. 35, Springer-Verlag, New York, 105–110
- [8] Buratynski, E. W. & Caughey, D. A. (1986) An Implicit LU Scheme for the Euler Equations

Applied to Arbitrary Cascades. **AIAA Journal** Vol. 24, 39–46.

- [9] Caughey, D. A. (1988) Diagonal Implicit Multigrid Algorithm for the Euler Equations. **AIAA Journal** Vol. 26, 841–851.
- [10] Caughey, D. A. (1993) Implicit Multigrid Euler Solutions with Symmetric Total Variation Diminishing Dissipation. **Proc. AIAA 11th CFD Conf.**, pp. 676–684.
- [11] Jameson, A. (1983) Solution of the Euler Equations for Two-dimensional, Transonic Flow by a Multigrid Method. **Appl. Math. Comp.** Vol. 13, pp. 327–356.
- [12] Jameson, A. (1995) Analysis and Design of Numerical Schemes for Gas Dynamics 1: Artificial



(a) Scalar SLIP



(b) CUSP

Figure 12: Comparison of fast, preconditioned multigrid solutions after only 5 multigrid cycles on fine grid with fully converged solutions on identical grid. NACA 0012 airfoil at Mach 0.85 and 1.00 degrees incidence on 160×32 cell grids.

Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence. **Int'l. Journal of Computational Fluid Dynamics** Vol. 4, pp. 171–218.

- [13] Jameson, A. (1995) Analysis and Design of Numerical Schemes for Gas Dynamics 2: Artificial Diffusion and Discrete Shock Structure. **Int'l. Journal of Computational Fluid Dynamics** Vol. 5, pp. 1–38.
- [14] Jameson, A., Schmidt, W. & Turkel, E. (1981) Numerical Solutions of the Euler Equations by Finite-Volume Methods using Runge-Kutta Time-stepping Schemes. **AIAA Paper 81-1259**.
- [15] Jameson, A. & Turkel, E. (1981) Implicit Schemes and LU Decompositions, **Math. Comp.** Vol. 37, pp. 385–397.
- [16] Jameson, A. & Yoon, S (1986) Multigrid Solutions of the Euler Equations using Implicit Schemes. **AIAA Journal** Vol. 24, pp. 1737–1743.
- [17] MacCormack, R. W., (1981) A Numerical Method for Solving the Equations of Compressible, Viscous Flow. **AIAA Paper 81-0110**.
- [18] MacCormack, R. W., (1997) A New Implicit Algorithm for Fluid Flow. **Proc. AIAA 13th CFD Conference**, Snowmass, Colorado, pp. 112-119.
- [19] Pierce, N. A. & Giles, M. B. (1997) Preconditioned Multigrid Methods for Compressible Flow Codes on Stretched Meshes. **J. Comp. Phys.** Vol. 136, pp. 425–445.
- [20] Pulliam, T. H. & Chaussee, D. S. (1981) A Diagonal Form of an Implicit Approximate Factorization Algorithm. **J. Comp. Phys.** Vol. 39, 347–363
- [21] Rieger, H. & Jameson, A. (1988) Solution of Steady Three-dimensional Compressible Euler and Navier-Stokes Equations by an Implicit LU Scheme. **AIAA Paper 88-0619**.
- [22] Roberts, T. W., Sidilkover, D. & Swanson, R. C. (1997). Textbook Multigrid Efficiency for the Steady Euler Equations, **AIAA Paper 97-1949**.
- [23] Roberts, T. W., Sidilkover, D. & Thomas, J. L. (2000). Multigrid Relaxation of a Factorizable, Conservative Discretization of the Compressible Flow Equations, **AIAA Paper 2000-2252**.
- [24] Ta'asan, S. (1994). Canonical-Variables Multigrid Method for Steady-State Euler Equations, **ICASE Report 94-14**.
- [25] Warming, R. F., Beam, R. M. & Hyett, B. J. (1975) Diagonalization and Simultaneous Symmetrization of the Gas Dynamic Equations. **Math. Comp.** Vol. 29, 1037–1045
- [26] Yokota, J. W. & Caughey, D. A. (1988). *LU* Implicit Multigrid Algorithm for the Three-dimensional Euler Equations, **AIAA Journal**, Vol. 26, pp. 1061-1069.
- [27] Yoon, S. & Jameson, A. (1988) Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations. **AIAA Journal** Vol. 26, pp. 1025–1026

Number of Grid Cells:	10	20	40	80	160	320	
Quasi-1-D Nozzle	Scalar	0.442	0.467	0.514	0.665	0.675	0.771
	CUSP	0.611	0.456	0.471	0.550	0.553	0.445

Table 1: Average rates of convergence per multigrid cycle for 100 cycles, or until residual or fractional error in total enthalpy is reduced to less than 10^{-12} .

Number of Grid Cells:		20×4	40×8	80×16	160×32
RAE 2822; $\mathbf{M} = .75$, $\alpha = 3.0^\circ$	Scalar	0.726	0.673	0.692	0.789
	CUSP	0.833	0.656	0.705	0.747
NACA 0012; $\mathbf{M} = .80$, $\alpha = 1.25^\circ$	Scalar	0.734	0.684	0.730	0.771
	CUSP	0.773	0.731	0.722	0.759
NACA 0012; $\mathbf{M} = .85$, $\alpha = 1.0^\circ$	Scalar	0.748	0.678	0.732	0.768
	CUSP	0.737	0.684	0.767	0.779

Table 2: Average rates of convergence per multigrid cycle for 100 cycles, or until residual or fractional error in total enthalpy is reduced to less than 10^{-12} .

Case	Fig.	MG Cycles	C_L	C_D
RAE 2822; $\mathbf{M} = 0.75$; $\alpha = 3.00$	–	100	1.1509	0.04960
	10(a)	5	1.1529	0.04977
	7(a)	3	1.1554	0.05028
NACA 0012; $\mathbf{M} = 0.80$; $\alpha = 1.25$	–	100	0.3788	0.02419
	11(a)	5	0.3813	0.02439
	8(a)	3	0.3851	0.02461
NACA 0012; $\mathbf{M} = 0.85$; $\alpha = 1.00$	–	100	0.4039	0.05945
	12(a)	5	0.4010	0.05949
	9(a)	3	0.3967	0.05932

Table 3: Force coefficients for the fast, preconditioned multigrid solutions using scalar SLIP spatial discretization; see Figs. 7(a), 8(a), 9(a), 10(a), 11(a), and 12(a).

Case	Fig.	MG Cycles	C_L	C_D
RAE 2822; $\mathbf{M} = 0.75$; $\alpha = 3.00$	–	100	1.1417	0.04851
	10(b)	5	1.1429	0.04851
	7(b)	3	1.1451	0.04886
NACA 0012; $\mathbf{M} = 0.80$; $\alpha = 1.25$	–	100	0.3725	0.02377
	11(b)	5	0.3746	0.02391
	8(b)	3	0.3770	0.02387
NACA 0012; $\mathbf{M} = 0.85$; $\alpha = 1.00$	–	100	0.3753	0.05748
	12(b)	5	0.3721	0.05722
	9(b)	3	0.3644	0.05670

Table 4: Force coefficients for the fast, preconditioned multigrid solutions using CUSP spatial discretization; see Figs. 7(b), 8(b), 9(b), 10(b), 11(b), and 12(b).