

# Comparison of Three Explicit Multigrid Methods for the Euler and Navier- Stokes Equations

(NASA-TM-88878) COMPARISON OF THREE  
EXPLICIT MULTIGRID METHODS FOR THE EULER AND  
NAVIER-STOKES EQUATIONS (NASA) 15 p

N87-16784

CSCL 01B

Unclas

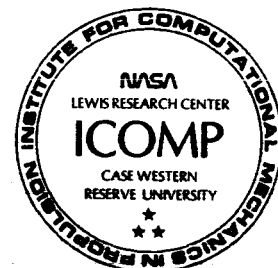
G3/01 43998

Rodrick V. Chima and Eli Turkel  
*Lewis Research Center*  
*Cleveland, Ohio*

and

Steve Schaffer  
*Institute for Computational Mechanics in Propulsion*  
*Lewis Research Center*  
*Cleveland, Ohio*

Prepared for the  
25th Aerospace Sciences Meeting  
sponsored by the American Institute of Aeronautics and Astronautics  
Reno, Nevada, January 12-15, 1987



# COMPARISON OF THREE EXPLICIT MULTIGRID METHODS FOR THE EULER AND NAVIER-STOKES EQUATIONS

Rodrick V. Chima and Eli Turkel\*  
National Aeronautics and Space Administration  
Lewis Research Center  
Cleveland, Ohio 44135

and

Steve Schaffer\*\*  
Institute for Computational Mechanics in Propulsion  
Lewis Research Center  
Cleveland, Ohio 44135

## Abstract

Three explicit multigrid methods, Ni's method, Jameson's finite-volume method, and a finite-difference method based on Brandt's work, are described and compared for two model problems. All three methods use an explicit multistage Runge-Kutta scheme on the fine grid, and this scheme is also described. Convergence histories for inviscid flow over a bump in a channel for the fine-grid scheme alone show that convergence rate is proportional to Courant number and that implicit residual smoothing can significantly accelerate the scheme. Ni's method was slightly slower than the implicitly-smoothed scheme alone. Brandt's and Jameson's methods are shown to be equivalent in form but differ in their node versus cell-centered implementations. They are about 8.5 times faster than Ni's method in terms of CPU time. Results for an oblique shock/boundary layer interaction problem verify the accuracy of the finite-difference code. All methods slowed considerably on the stretched viscous grid but Brandt's method was still 2.1 times faster than Ni's method.

## Introduction

Three explicit multigrid methods now being used for solution of the Euler equations and occasionally for the Navier-Stokes equations are discussed in this paper. The first method was developed by Ni in 1981<sup>1</sup> to accelerate the convergence of his own fine-grid Euler scheme. Subsequent work by Johnson and Chima<sup>2,5</sup> generalized the method to other finite-difference fine-grid schemes and to viscous flows for practical turbomachinery problems. A variation of Ni's method has also been used by Hall for inviscid flows over airfoils.<sup>6</sup> A second method was developed by Jameson and Baker in 1983<sup>7</sup> to accelerate the convergence of the finite-volume multi-stage Runge-Kutta schemes developed by Jameson, Schmidt, and Turkel.<sup>8</sup> This method has been used very successfully for inviscid two-dimensional flows over airfoils and three-dimensional flows over aircraft configurations,<sup>9</sup> and for two-dimensional viscous flows.<sup>10</sup>

\*Also, Institute for Computer Applications in Science and Engineering, Langley Research Center, Hampton, Virginia 23665.

\*\*Work funded under Space Act Agreement C99066-G; presently at Berea College, Berea, Kentucky 40403.

Multigrid methods were first developed for elliptic problems and have been analyzed in detail by Brandt.<sup>11</sup> For the third method considered here Chima and Schaffer used Brandt's approach to develop a finite-difference multigrid Euler and Navier-Stokes code. Many of Jameson's ideas were used on the fine grid so that a direct comparison could be made between the basic Runge-Kutta scheme, the multigrid scheme, and Ni's scheme. In parallel, Turkel has revised Jameson's finite-volume multigrid Euler codes (FLO52MG and FLO53MG) allowing comparisons with this method as well.

The intent of this paper is to develop and compare the Ni, Jameson, and Brandt types of multigrid schemes. All of these schemes use the Runge-Kutta method as the basic algorithm. Therefore we will discuss the Runge-Kutta method, its associated boundary conditions and artificial viscosity terms in both finite-difference and finite-volume forms. We will also discuss other convergence acceleration techniques including spatially-varying time steps and residual smoothing. Details of the three multigrid methods will then be presented. Finally results for several inviscid and viscous flows will be used to demonstrate the relative effectiveness of the schemes and to point out areas where further work is needed.

## Governing Equations

The two-dimensional unsteady thin-layer Navier-Stokes equations may be written in fully conservative form for an arbitrary coordinate system as follows:

$$\partial_t q = -J[\partial_{\xi} \hat{E} + \partial_{\eta} (\hat{F} - Re^{-1} \hat{S}) - D] \quad (1)$$

$\equiv R$ , the residual

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}; \quad \hat{E} = J^{-1} \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (e + p)U \end{bmatrix}; \quad \hat{F} = J^{-1} \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (e + p)V \end{bmatrix} \quad (2)$$

$e = \rho [C_v T + \frac{1}{2}(u^2 + v^2)]$  is the total energy per unit volume.

$p = (\gamma - 1) \left[ e - \frac{1}{2} \rho(u^2 + v^2) \right]$  is the static pressure and  $D$  is an artificial dissipation term to be described later.

The viscous flux term  $\hat{S}$  is given by:

$$\hat{S} = J^{-1} \mu \begin{bmatrix} 0 \\ C_1 \frac{\partial u}{\partial \eta} + C_2 \frac{\partial v}{\partial \eta} \\ C_2 \frac{\partial u}{\partial \eta} + C_3 \frac{\partial v}{\partial \eta} \\ C_4 \frac{\partial}{\partial \eta} \left[ \frac{e}{\rho} - \frac{1}{2}(u^2 + v^2) \right] \\ + (C_1 u + C_2 v) \frac{\partial u}{\partial \eta} \\ + (C_2 u + C_3 v) \frac{\partial v}{\partial \eta} \end{bmatrix} \quad (3)$$

where

$$C_1 = \frac{4}{3} \eta_x^2 + \eta_y^2; \quad C_2 = \frac{1}{3} \eta_x \eta_y; \\ C_3 = \eta_x^2 + \frac{4}{3} \eta_y^2; \quad C_4 = \frac{\gamma}{Pr} \left( \eta_x^2 + \eta_y^2 \right)$$

In Eqs. (1) to (3)  $J$  is the transformation Jacobian:

$$J = \xi_x \eta_y - \xi_y \eta_x = \frac{1}{x_\xi y_\eta - x_\eta y_\xi} = \frac{1}{(\text{cell volume})} \quad (4)$$

The contravariant velocity components  $U$  and  $V$  along the  $\xi$ - and  $\eta$ -grid lines are given by:

$$U = \xi_x u + \xi_y v; \quad V = \eta_x u + \eta_y v \quad (5)$$

#### Spatial Differencing

Two approaches are used to construct difference formulas on general curvilinear grids. In the finite-difference approach new independent variables  $\xi, \eta$  are chosen so that  $\xi = \xi(x, y)$ ,  $\eta = \eta(x, y)$  maps the original domain to a rectangle and each curvilinear coordinate to a straight line. The equations are then transformed to  $(\xi, \eta)$  coordinates and differenced based on even spacing  $\Delta \xi = \Delta \eta = 1$ . In the finite-volume approach the equations are rewritten in integral form and the divergence theorem is used to express the fluxes in terms of surface (or line) integrals. These integrals are then approximated by some integration formula.

Using central differences these two approaches give rise to identical formulas for interior cells even for the full Navier-Stokes equations, see e.g., Ref. 12. Though it is not imperative, finite difference schemes usually locate the variables at the nodes while finite volume schemes usually locate the variables at the center of the cell where they represent a cell-averaged quantity. The differences between cell-centered and node-centered schemes are mainly noticed at boundaries and in the transfer of quantities between coarse and fine grids. These differences will be discussed later.

For stability and shock capturing both schemes require an artificial viscosity term which will be discussed. Also both schemes use a local time step and implicit residual smoothing to accelerate convergence at the expense of losing time accuracy. In addition the finite-volume code uses enthalpy damping.<sup>8</sup>

#### Boundary Conditions - Finite-Difference Code

For subsonic flow, inlet values of  $P_0, T_0$ , and  $v$  are specified. The upstream-running Riemann invariant  $R^- = u - 2c/(\gamma - 1)$  is extrapolated from the interior and isentropic relations are used to compute the necessary flow quantities. For supersonic flow all inflow quantities are specified.

For subsonic flow, exit static pressure is specified and  $\rho, u$ , and  $v$  are extrapolated from the interior. For supersonic flow all outflow quantities are extrapolated.

On the walls tangency is enforced for inviscid flow by extrapolating  $U$  and setting  $V = 0$ . For viscous flows  $u = v = 0$ . Density is extrapolated to the walls. Surface pressures are computed using the normal momentum equation:

$$(\eta_x \xi_x + \eta_y \xi_y) p_\xi + (\eta_x^2 + \eta_y^2) p_\eta = -\rho U (\eta_x u_\xi + \eta_y v_\xi) \quad (6)$$

where  $U = 0$  on the surface for viscous flows.

In the finite-difference code, boundary conditions are applied after all stages of the Runge-Kutta scheme have been completed.

#### Boundary Conditions - Finite Volume Code

For external flows the incoming Riemann invariant  $R^+ = u + 2c/(\gamma - 1)$ , the entropy, and the tangential velocity are specified upstream. The outgoing Riemann invariant  $R^- = u - 2c/(\gamma - 1)$  is extrapolated. At a subsonic outflow the situation is reversed with the first three quantities extrapolated and  $R^-$  specified. The downstream total energy is evaluated assuming a constant total enthalpy.

For internal flows the total enthalpy, the entropy, and the tangential velocity are specified at the inlet and  $R^-$  is extrapolated. At the exit the static pressure is specified and the entropy, the tangential velocity, and  $R^+$  are extrapolated.

Variables are not defined directly on solid surfaces where only the normal fluxes are needed. Since  $V = 0$  on solid surfaces it follows that  $\hat{F}$  depends only on  $p$  at the surface (Eq. 2). The pressure is calculated from the normal momentum Eq. (6). In practice a fictitious cell is placed inside the body. In this cell the pressure is found from Eq. (6), the density and normal velocity are set symmetric with external values, and the tangential velocity is set antisymmetric. For viscous flows simple extrapolation of the pressure is sufficient and both velocity components are set antisymmetric.

### Artificial Dissipation - Finite-Difference Code

Dissipative terms consisting of fourth and second differences are added to prevent odd-even point decoupling and to allow shock capturing respectively. The dissipative term  $D$  in Eq. (1) is given by:

$$Dq = (D\xi + D\eta)q \quad (7)$$

In the finite-difference code the dissipative operators are non-conservative. The  $\xi$ -direction operator is given by:

$$D\xi q = C (V_2 q_{\xi\xi} - V_4 q_{\xi\xi\xi\xi}) \quad (8)$$

where

$$C = \frac{\sqrt{\eta_x^2 + \eta_y^2}}{J} = \frac{\sqrt{y_\xi^2 + x_\xi^2}}{J} = S_\xi \quad (9)$$

is the arc length along the grid in the direction of the dissipation. The terms  $V_2$  and  $V_4$  are given by:

$$\begin{aligned} V_2 &= \mu_2 \text{MAX} (v_{i,j+1}, v_{i,j}, v_{i,j-1}) \\ V_4 &= \text{MAX} (0, \mu_4 - V_2) \end{aligned} \quad (10)$$

where

$$v_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{|p_{i+1,j} + 2p_{i,j} + p_{i-1,j}|} \quad (11)$$

and

$$\begin{aligned} \mu_2 &= O(1) \\ \mu_4 &= O\left(\frac{1}{16}\right) \end{aligned} \quad (12)$$

Switching functions used in Eq. (10) increase  $V_2$  slightly and switch off  $V_4$  at shocks, effectively eliminating overshoots ahead of shocks.

In smooth regions of the flow the dissipative terms are of third order and thus do not detract from the formal second-order accuracy of the scheme. Near shocks  $v_{i,j}$  is large and the second-difference dissipation becomes locally of first order.

### Artificial Dissipation - Finite-Volume Code

The artificial viscosity terms in the finite-volume code are similar to those in the finite-difference code. The dissipative term  $D$  is evaluated in conservation form using

$$D\xi q = \partial_\xi C (V_2 q_\xi - V_4 q_{\xi\xi\xi\xi}) \quad (13)$$

In recent versions of the code the fourth difference term is computed as the second difference of  $CV_4 q_{\xi\xi}$  rather than as given above.  $V_2$  and  $V_4$  are given by (10,11) as before with typical values of the constants being:

$$\begin{aligned} \mu_2 &= \frac{1}{4} \text{ to } \frac{1}{2} \\ \mu_4 &= \frac{1}{128} \text{ to } \frac{1}{64} \end{aligned} \quad (14)$$

Now  $C$  is given by the sum of the spectral radii of the Jacobians of  $\hat{E}$  and  $\hat{F}$ . Define

$$\hat{A} = \frac{\partial \hat{E}}{\partial q} \quad \text{and} \quad \hat{B} = \frac{\partial \hat{F}}{\partial q}$$

then

$$\rho(\hat{A}) = \frac{|U| + C \sqrt{\xi_x^2 + \xi_y^2}}{J} = \frac{1}{J\Delta t x}$$

$$\rho(\hat{B}) = \frac{|V| + C \sqrt{\eta_x^2 + \eta_y^2}}{J} = \frac{1}{J\Delta t y}$$

$$C = \rho(\hat{A}) + \rho(\hat{B}) = \frac{1}{J\Delta t} \quad (15)$$

An alternative is to use  $C = \rho(\hat{A})$  in the  $\xi$ -direction and  $C = \rho(\hat{B})$  in the  $\eta$ -direction. In this case the metric quantities in  $C$  represent a length scale normal to the direction of the dissipation, which is exactly opposite to that used in the finite-difference code Eq. (9).

### Multi-Stage Runge-Kutta Schemes

An explicit Runge-Kutta scheme is used to advance the Euler or thin-layer Navier-Stokes equations in time from an initial guess to a steady state. Given the residual  $R$  from a finite-difference or finite-volume representation of the flow equations, a  $k$ -stage scheme can be written as

$$\begin{aligned} q(0) &= q^n \\ q(1) &= q(0) - \alpha_1 \Delta t Rq(0) \\ &\vdots \\ q(k) &= q(0) - \alpha_k \Delta t Rq(k-1) \\ q^{n+1} &= q(k) \end{aligned} \quad (16)$$

A  $k$ -stage scheme used with central differencing of the inviscid equations can be made stable for a Courant number up to  $\lambda^* = k-1$ , depending on the choice of  $\alpha_j$ . The values of  $\alpha_j$  and  $\lambda^*$  used here are given in Table I. For consistency we must have  $\alpha_k = 1$ . For nonlinear problems the schemes are second-order accurate in time if  $\alpha_{k-1} = 1/2$ . Schemes of the form of Eq. (16) cannot be more than second-order accurate for any values  $\alpha_k$ .

For efficiency both the physical and artificial dissipation terms are calculated once based on  $q(0)$ , then held constant for the remaining stages. For a five-stage scheme these terms are usually reevaluated after the first stage. In either case the multi-stage schemes have the desirable property that if the solution converges, i.e.,  $Rq(0) = 0$ , then  $q(i) = q(0)$  and  $q^{n+1} = q^n$ , independent of the time step.

### Spatially-Variable Time Step

Both the finite-difference and finite-volume codes use a spatially-variable time step to accelerate convergence. In the finite-difference code the time step is given by:

$$\Delta t_{i,j} = \frac{\lambda}{dx|u| + dy|v| + c \sqrt{dx^2 + dy^2}} \Big|_{I.C.} \quad (17)$$

where

$$dx \equiv |\xi_x| + |\eta_x|; \quad dy \equiv |\xi_y| + |\eta_y|$$

This is calculated based on the initial conditions and is stored and not updated during the solution.

In the finite-volume code the two-dimensional time step is given by a conservative estimate as:

$$\frac{1}{\Delta t_{i,j}} = \frac{1}{\Delta t_{x,i,j}} + \frac{1}{\Delta t_{y,i,j}} \quad (18)$$

where

$$\Delta t_x = \frac{\lambda}{|u| + c \sqrt{\xi_x^2 + \xi_y^2}}$$

$$\Delta t_y = \frac{\lambda}{|v| + c \sqrt{\eta_x^2 + \eta_y^2}} \quad (19)$$

#### Multi-Stage Runge-Kutta Results

A model problem of inviscid, transonic flow over a 10 percent thick circular-arc bump in a channel was used to investigate the effects of number of stages and Courant number on convergence rate. The grid for this problem has 129 by 33 points and is shown at the top of Fig. 1. The inlet Mach number 0.675 is the one-dimensional choking Mach number for this case. Mach contours at the bottom of Fig. 1 show that although a supersonic bubble and shock do develop, the two-dimensional flow does not choke.

Convergence histories for the finite-difference scheme without multigrid in Fig. 2 show the log of the RMS residual in density versus iteration for two to five-stage schemes. Each scheme was run near its maximum Courant number. As expected, convergence rate improves with Courant number. The two-stage scheme required 29.6 sec of cpu time on the Cray X-MP 2/4 at NASA Lewis Research Center, and the number of iterations reached in that time is marked on the other curves. The higher-level schemes are more efficient since they obtain higher Courant numbers per number of stages while requiring fewer evaluations of the artificial dissipation.

If the convergence criterion is taken as a three decade drop in the residuals, only the five-stage results are fully converged at 2000 iterations. The four-stage results, however, are converged to plotting accuracy.

#### Implicit Residual Smoothing

Residual smoothing was introduced by Lerat (see for example Ref. 13) for use with the Lax-Wendroff scheme and was later applied to Runge-Kutta schemes by Jameson.<sup>14</sup> The technique involves replacing the residual calculated from Eq. (1) with a value smoothed by an implicit filter, e.g.,

$$(1 - \epsilon_\xi \delta_{\xi\xi})(1 - \epsilon_\eta \delta_{\eta\eta})\bar{R} = R \quad (20)$$

where  $\delta_{\xi\xi}$  and  $\delta_{\eta\eta}$  are standard second-difference operators and  $\epsilon_\xi$  and  $\epsilon_\eta$  are smoothing parameters.

Residual smoothing increases the support of the finite-difference scheme and thereby increases the stability range of the time-stepping scheme. Linear analysis has shown that the Runge-Kutta scheme with implicit smoothing may be made unconditionally stable if  $\epsilon$  is sufficiently large. In one dimension

$$\epsilon > \frac{1}{4} \left[ \left( \frac{\lambda}{\lambda^*} \right)^2 - 1 \right] \quad (21)$$

gives unconditional stability where  $\lambda^*$  is the Courant number of the unsmoothed scheme and  $\lambda$  is a larger operating Courant number. In two dimensions  $\epsilon$  may be reduced substantially and different values may be used in each direction. Turkel<sup>15</sup> has shown that since large values of  $\epsilon$  can decrease the convergence rate of the scheme, the best strategy is not to maximize the Courant number but simply to increase the Courant number of the unsmoothed scheme by a factor of two to three.

Implicit residual smoothing requires solution of a scalar tridiagonal equation for each variable in each direction, and adds 10 to 15 percent to the total CPU time for a solution on a Cray X-MP. In the finite-volume code it is applied after each Runge-Kutta stage. In the finite-difference code it is applied after every-other stage with somewhat larger values of  $\epsilon$ .

Figure 3 shows the effects of implicit smoothing on convergence rate of the five-stage scheme for the problem of Fig. 1. The top curve is for the unsmoothed scheme at a Courant number of three and is duplicated from Fig. 2. This scheme converges with a spectral radius of about 0.997. The middle curve is for a Courant number of six with smoothing after each stage. It converges about twice as fast as the unsmoothed scheme with a spectral radius of 0.994. The bottom curve for a Courant number of nine shows a spectral radius of 0.991. Again the convergence rate improves with Courant number but the amount of improvement decreases as the Courant number becomes large.

Circles on the curves can be used to compare convergence level at a given CPU time to that of the two-stage scheme shown in Fig. 2.

#### Multigrid Methods

Multigrid algorithms were originally developed for elliptic problems. Brandt<sup>11</sup> has described some of the earlier versions of these algorithms. This approach supposes that one has an iteration procedure which quickly reduces the high frequency error but then slows down in reducing the low frequency error. One applies the basic iteration procedure a few times to remove the high frequency component of the error. The remaining error is then fed to a coarser grid which can represent the error since it no longer contains high frequencies that would be aliased. On this coarser grid the basic smoothing algorithm is again used and the process is repeated.

## Ni's Method

Ni's method is basically a one-step Lax-Fredrichs scheme applied to the residual on a coarse grid. Ni used it to accelerate his own fine-grid Euler scheme.<sup>1</sup> Johnson adapted it to McCormack's fine-grid scheme<sup>2</sup> and to viscous flows by demonstrating that viscous terms may be neglected on the coarse grids. Johnson also simplified Ni's scheme considerably by replacing the flux Jacobian terms with temporal differences of the fluxes.<sup>3</sup> This "flux-based" coarse-grid scheme is outlined below.

Given a fine-grid solution  $q$  and a fine-grid change

$$\Delta q^{n+1} = q^{n+1} - q^n$$

a coarse-grid change can be estimated using a Taylor series:

$$\Delta q^{n+2} = \Delta q^{n+1} + \Delta t(\Delta q^{n+1})_t + O(\Delta t^2)$$

The Euler equations are used to replace the third term

$$(\Delta q^{n+1})_t = -\Delta t(\hat{E}_\xi + \hat{F}_\eta)_t$$

and the order of differentiation is reversed

$$(\Delta q^{n+1})_t = -\Delta t(\hat{E}_{t\xi} + \hat{F}_{t\eta}) = -[(\hat{E}^{n+1} - \hat{E}^n)_\xi + (\hat{F}^{n+1} - \hat{F}^n)_\eta]$$

giving finally

$$\Delta q^{n+2} = \Delta q^{n+1} - \Delta t[(\hat{\Delta E})_\xi + (\hat{\Delta F})_\eta] \quad (22)$$

where

$$\hat{\Delta E} = \hat{E}^{n+1} - \hat{E}^n, \text{ etc.}$$

Equation (22) is implemented on a coarse grid with spacing  $2h\Delta\xi$  and  $2h\Delta\eta$  and time step  $h\Delta t$ ,  $h = 1, 2, 4, 8 \dots$  using:

$$\begin{aligned} \Delta q_{i,j}^{n+2} = \frac{1}{4} \left\{ \left[ \Delta q - \Delta t_{i,j}(\hat{\Delta E} + \hat{\Delta F}) \right]_{i+h,j+h} \right. \\ + \left[ \Delta q - \Delta t_{i,j}(\hat{\Delta E} - \hat{\Delta F}) \right]_{i+h,j-h} \\ + \left[ \Delta q - \Delta t_{i,j}(-\hat{\Delta E} + \hat{\Delta F}) \right]_{i-h,j+h} \\ \left. + \left[ \Delta q - \Delta t_{i,j}(-\hat{\Delta E} - \hat{\Delta F}) \right]_{i-h,j-h} \right\}^{n+1}_h \end{aligned} \quad (23)$$

The fine-grid Runge-Kutta scheme is used to advance the solution one time step, giving  $q^{n+1}$  and  $\Delta q^{n+1}$  at each grid point. Choosing a value of  $h$  determines a coarser grid, and Eq. (23) is then used to determine  $\Delta q^{n+2}$  at those coarse-grid points. On the boundaries  $\Delta q^{n+1}$  is taken to be zero. Old values of  $\Delta q^{n+1}$  may be overwritten so that no additional coarse-grid storage is needed. Values of  $\Delta q^{n+2}$  at intermediate fine-grid points

are then filled in by bilinear interpolation and the fine grid is updated using  $q^{n+2} = q^{n+1} + \Delta q^{n+2}$ . This leaves us back on the fine grid with new values of  $q$  and  $\Delta q$ , and the process can be repeated for any other coarse grid. In practice, the grids are advanced from fine to coarse with one iteration on each.

Since the right-hand side of Eq. (23) depends strictly on fine-grid changes, if the fine-grid converges the coarse-grid scheme given by (23) cannot change the solution. Thus, the delta form of Ni-type schemes retains fine-grid accuracy. For the same reason, all physical viscous dissipation terms may be neglected on the coarse-grids without affecting a viscous solution on the fine grid.

Equation (23) may be interpreted as a Lax-Fredrichs scheme because of the averaging of  $\Delta q$  and of the fluxes. This averaging stabilizes the coarse-grid scheme without the use of artificial viscosity. The scheme is stable to a Courant number of one, which restricts the fine-grid scheme to a Courant number of one. Thus Ni's scheme is attractive for accelerating explicit McCormack-type schemes but is limited to a two-stage Runge-Kutta scheme. In the next section we discuss a possible multistage Ni-type scheme.

Finally, Eq. (22) shows that the coarse-grid change  $\Delta q^{n+2}$  will be of the order of the fine-grid change  $\Delta q^{n+1}$  plus a small correction of order  $\Delta t$ . Thus each coarse-grid step gives effectively one fine grid  $\Delta t$  but with fewer operations. This is demonstrated in Fig. 4 where convergence rates for the transonic model problem are compared for different numbers of grids. The fine-grid scheme (1 grid) is a two-stage Runge-Kutta scheme with a Courant number of 0.95. Convergence rates for two, three, and four grids are nearly identical to the convergence rates for Courant numbers of two, three, and four shown in Fig. 2, but require less CPU time. The four-grid curve in Fig. 4 is about the best that we have done with Ni's scheme, but it is not as fast or easy to program as the five-stage scheme with implicit residual smoothing shown in Fig. 3.

## Multistage Ni Scheme

The major drawback of Ni-type schemes is that the Lax-Fredrichs coarse-grid scheme has a Courant limit of one. Hence, there is no major gain in using a multistage Runge-Kutta method on the fine grid with a large Courant number when this cannot be maintained on coarser grids. An alternative is to use a multistage Runge-Kutta algorithm on all grids. This necessitates rewriting the Runge-Kutta scheme so that only  $\Delta q$  appears. Consider the equation

$$q_t = F_x$$

A typical stage of a Runge-Kutta scheme is

$$\Delta q_j^{(k+1)} = q_j^{(k+1)} - q_j^n = \alpha_{k+1} \frac{\Delta t}{2\Delta x} [F(q_{j+1}^{(k)}) - F(q_{j-1}^{(k)})] \quad (24)$$

We wish to rewrite the right-hand side of (24) so that only  $\Delta q$  appears. Note that

$$q_j^{(k)} = q_j^{(n)} + (q_j^{(k)} - q_j^n) = q_j^n + \Delta q_j^{(k)}.$$

Using a Taylor series expansion for  $F(q_j^n + \Delta q_j^{(k)})$  we find that an alternative to Eq. (24) is

$$\Delta q_j^{(k+1)} = \alpha_{k+1} \frac{\Delta t}{2\Delta x} [F(q_{j+1}^n) - F(q_{j-1}^n) + F'(q_{j+1}^n) \Delta q_{j+1}^{(k)} - F'(q_{j-1}^n) \Delta q_{j-1}^{(k)}]$$

or

$$\Delta q_j^{(k+1)} = \frac{\alpha_{k+1}}{\alpha_1} \Delta q_j^{(1)} + \frac{\alpha_{k+1}}{2\Delta x} [F'(q_{j+1}^n) \Delta q_{j+1}^{(k)} - F'(q_{j-1}^n) \Delta q_{j-1}^{(k)}]. \quad (25)$$

Here only  $\Delta q$ 's appear, as in Ni's original scheme. Furthermore, for a linear problem we have the standard Runge-Kutta scheme and so the stability conditions are unchanged.

The multistage coarse-grid scheme given by Eq. (25) is thus proposed as an alternative to Ni-type schemes that would permit larger Courant numbers on the coarse grids.

#### Full Approximation Storage (FAS) Multigrid Method

Brandt's FAS multigrid has been developed as a general strategy for accelerating iterative schemes and may be applied directly to the multi-stage schemes. It results in a coarse-grid equation that has the same form as the fine-grid equation with the addition of a forcing function. The multistage scheme may thus be used directly on the coarser grids, with successively larger time steps on successively coarser grids. This is the primary advantage of Brandt's or Jameson's methods over Ni's method. Here we develop FAS multigrid using Brandt's notation, then show the relationship to Jameson's method.

Brandt's notation.<sup>11</sup> Consider a general steady nonlinear equation on a grid with spacing parameter  $h$ .

$$R_h q_h = f_h \quad (26)$$

where the forcing function  $f_h$  may be zero. Consider an approximate solution  $\tilde{q}_h$ , evaluate  $R_h \tilde{q}_h$  and subtract it from each side of Eq. (26) to get the residual equation.

$$R_h q_h - R_h \tilde{q}_h = f_h - R_h \tilde{q}_h$$

This may be approximated on a coarser 2h grid using

$$R_{2h} q_{2h} = R_{2h} I_h^{2h} q_h + I_h^{2h} (f_h - R_h q_h) \quad (27)$$

where  $I_h^{2h}$  means interpolation from the  $h$ -grid to the  $2h$ -grid. The unsteady terms may be added to Eq. (27) to give:

$$(q_{2h})_t + R_{2h} q_{2h} = f_{2h} \quad (28)$$

where

$$f_{2h} = R_{2h} I_h^{2h} q_h + I_h^{2h} (f_h - R_h q_h)$$

Equation (28) may now be solved on the  $2h$ -grid using the Runge-Kutta scheme:

$$q_{2h}^{(h)} = q_{2h}^{(0)} - \alpha_1 \Delta t (R_{2h} q_{2h}^{(i-1)} - f_{2h}^{(0)}) \quad (29)$$

and the corrections may be interpolated back to the  $h$ -grid using:

$$q_h \leftarrow q_h + I_{2h}^h (q_{2h} - I_h^{2h} q_h) \quad (30)$$

On more than two grids, Eqs. (29) and (30) may be applied recursively.

It can be shown that Eqs. (29) and (30) are the  $2h$ -grid approximation to linearizing Eq. (26) around the current solution  $\tilde{q}_{2h}$ , solving Eq. (26) for the error  $e_{2h} = q_{2h} - \tilde{q}_{2h}$ , and then updating

the current solution by  $\tilde{q}_h \leftarrow \tilde{q}_h + I_{2h}^h e_{2h}$ . This approximation is reasonable only if the error  $e_{2h}$  is smooth and therefore visible on the  $2h$ -grid. Thus, it is desirable to choose the parameters of the Runge-Kutta scheme to insure smoothing of high-wavenumber errors.

If the fine grid converges,  $f_h - R_h q_h = 0$  and Eq. (28) becomes

$$(q_{2h})_t + R_{2h} q_{2h} = R_{2h} I_h^{2h} q_h + I_h^{2h} (f_h - R_h q_h) \quad \text{with } f_h - R_h q_h = 0$$

or

$$(q_{2h})_t = 0 \text{ since } q_{2h} = I_h^{2h} q_h$$

so that the coarse-grid scheme maintains fine-grid accuracy. Like Ni's scheme, this is true even if the coarse-grid residual  $R_{2h}$  is different from the fine-grid residual  $R_h$ , so that on the coarse grids a simple first-order artificial viscosity may be used and the physical viscosity may be neglected altogether.

Jameson's notation.<sup>9</sup> Jameson starts with a general unsteady equation written in finite-volume form:

$$\frac{d}{dt} (Vq) + Rq = 0; \quad V = \text{volume} \quad (31)$$

and defines a coarse-grid forcing function as the difference between the coarse-grid residual and the sum of the fine-grid residuals:

$$f_{2h}^{(0)} = - \sum R_h q_h^{(0)} + R_{2h} q_{2h}^{(0)} \quad (32)$$

On the coarse grids the multistage scheme is implemented using

$$q_{2h}^{(1)} = q_{2h}^{(0)} - \alpha_1 \Delta t (R_{2h} q_{2h}^{(i-1)} - f_{2h}^{(0)})$$

or

$$q_{2h}^{(i)} = q_{2h}^{(o)} - \alpha_i \Delta t \left[ R_{2h} q_{2h}^{(i-1)} + \sum R_h q_h^{(o)} - R_{2h} q_{2h}^{(o)} \right] \quad (33)$$

Jameson's coarse-grid scheme Eq. (33) may be compared directly with the one based on Brandt Eq. (28), which is repeated below for reference.

$$q_{2h}^{(h)} = q_{2h}^{(o)} - \alpha_i \Delta t \left[ R_{2h} q_{2h}^{(i-1)} + I_h^{2h} \left( R_h q_h^{(o)} - f_h^{(o)} \right) - R_{2h} I_h^{2h} q_h^{(o)} \right] \quad (28)$$

It is evident that the two schemes are equivalent in form but differ in the interpolation or grid-transfer operators.

#### Grid-Transfer Operators

Before discussing these operators we note some differences in coarse-grid data locations between node-centered (finite-difference or F.D.) and cell-centered (finite-volume or F.V.) schemes. In node-centered schemes, coarse-grid nodes coincide with fine-grid nodes (Fig. 5(a)) so that variables can be transferred between grids directly without interpolation. This process is termed "injection." In cell-centered schemes, however, coarse-grid cells are made up of several fine-grid cells, and the centers of the coarse and fine-grid cells are not coincident (Fig. 5(b)). Thus, it is always necessary to use some form of interpolation between grids with cell-centered schemes. The grid transfer operators used in the FD and FV codes are listed below:

Restriction operators. For the term  $I_h^{2h} q_h$  the FD code uses injection and the FV code uses a volume-weighted average.

$$\text{FD: } q_{2h} = q_h$$

$$\text{FV: } q_{2h} = \sum \frac{V_h q_h}{V_{2h}} \quad (34)$$

For the restriction of the fine-grid residual,  $I_h^{2h} (R_h q_h - f_h)$ , Brandt recommends a weighted-average over several grid points. Although the FD code could use injection here we have obtained much better convergence rates using an unweighted average over nine neighboring nodes. The FV code uses a sum over the four fine-grid cells that make up the coarse-grid cell.

$$\begin{aligned} \text{FD: } I_h^{2h} (R_h q_h - f_h) &= \frac{4}{9} \sum_{9 \text{ nodes}} (R_h q_h - f_h) \\ \text{FV: } I_h^{2h} (R_h q_h - f_h) &= \sum_{4 \text{ cells}} (R_h q_h - f_h) \end{aligned} \quad (35)$$

Prolongation operators. For the expression  $q_h + q_h + I_{2h}^h (q_{2h} - I_h^{2h} q_h)$ , both codes calculate

the term  $I_h^{2h} q_h$  as described above, subtract it from  $q_{2h}$  to get the coarse-grid correction, then transfer the correction to the fine grid using bilinear interpolation. If more than one Runge-Kutta sweep is performed on a coarse grid then  $I_h^{2h} q_h$  must be stored or reconstructed.

Thus the main difference between the finite-difference and finite-volume multigrid schemes is the restriction operators which have been chosen appropriately for each scheme. We suspect that the differences have minor effects on convergence rates, but have not quantified those effects.

#### Coarse-Grid Boundary Conditions

Most of the boundary conditions described earlier involve extrapolations or one-sided differences and do not maintain fine-grid accuracy when applied directly on coarse grids. It is possible to define a boundary condition forcing function similar to Jameson's interior forcing function Eq. (32), but this increases programming complexity. Instead, coarse-grid boundary conditions are computed with coarse-grid accuracy, but only the change in boundary values during the Runge-Kutta cycle is transferred back to the fine grid, i.e.,

$$q_{hBC} \leftarrow q_{hBC} + I_{2h}^h (q_{2hBC}^{(k)} - q_{2hBC}^{(o)}) \quad (36)$$

This formulation maintains consistency in that if the interior scheme converges the coarse-grid scheme cannot change the fine-grid boundary values.

In the finite-difference code, boundary conditions are updated after every complete Runge-Kutta cycle. In the original version of FL052MG, boundary conditions were frozen on coarse grids. In the present version of FL052MG, boundary conditions are updated after every stage of the Runge-Kutta scheme and after every grid transfer.

#### Programming Considerations

FAS multigrid is usually programmed by storing all the grids end-to-end in one long, singly-dimensioned array. This increases the required storage by about 4/3 in two-dimension or 8/7 in three-dimension. Brandt typically accesses these long arrays by indirect addressing which can be difficult to program and vectorize. Jameson overcomes these difficulties in FL052MG by defining singly-dimensioned arrays in the main program but working with multiply-dimensioned arrays in the subroutines which have the array size and starting location passed through their argument list. This technique allows existing nonmultigrid subroutines to be converted to multigrid with minimal changes. Nevertheless, FAS multigrid is considerably more difficult to program than Ni's method.

#### Full Multigrid

Full multigrid (FMG) combines successive grid refinement with FAS multigrid. The solution is started on the coarsest grid and is iterated a few times using the fine-grid scheme, possibly with FAS multigrid as well. The solution is next interpolated to the next-finer grid where it provides a good initial guess, then the process is repeated until the finest grid is reached and the



solution converges. Experience has shown that it is wasteful to drive the residual to zero on coarse FMG grids. We have found that 25 Runge-Kutta cycles are sufficient, with more iterations used only on the finest grid.

#### Multigrid Cycle

In the finite-difference code we use a V or sawtooth-cycle with one Runge-Kutta iteration on each grid. A typical cycle is diagrammed in Fig. 6. Turkel's experience has shown some increase in efficiency by doing one iteration on the  $h$ -grid, two iterations on the  $2h$ -grid, and three iterations on all coarser grids. More iterations on the coarsest grid may actually decrease the convergence rate.

#### Multigrid Results

Figure 7 shows FMG convergence rates for the problem of Fig. 1 using the finite-difference code. Here 300 iterations were run on each of four successive grids to show the convergence rate on each. Normally only 25 iterations would be run on the first three grids. The coarsest grid (17 by 5) was run with the four-stage scheme alone and has a spectral radius of 0.962. The second and third grids have spectral radii of 0.937, and the finest grid (129 by 33) converges with a spectral radius of 0.95.

Figure 8 shows the FMG convergence rate of Turkel's version of Jameson's finite-volume code FLO52MG for external flow about an RAE2822 supercritical airfoil with  $M_\infty = 0.75$  and  $\alpha = 3^\circ$ . A C-type grid was used with 160 by 32 points on the finest grid. Thirty iterations were run on both the coarse and medium grids, with two and three grid levels used respectively. Ninety iterations were run on the finest grid with four grid levels, giving a spectral radius of about 0.763. This is a typical convergence rate for this code for inviscid external flows on uniform grids.

Figure 9 compares convergence rates for Ni's method, the finite-difference Runge-Kutta scheme alone, and the finite-difference and finite-volume Runge-Kutta schemes with FAS multigrid. The Ni multigrid results were run with four grids at a Courant number of 0.95, and show a spectral radius of 0.995. The four-stage Runge-Kutta results were run with implicit residual smoothing at a Courant number of 5.2 and show a spectral radius of 0.994. The finite-difference FAS multigrid results were run with the same Runge-Kutta parameters as above but with 25 FMG iterations on each coarse grid and 300 FAS iterations on the finest grid. The spectral radius of the FAS multigrid scheme is 0.943. The finite-volume FAS multigrid results were run with a four-stage scheme at a Courant number of 6.0 and also with a five-stage scheme at a Courant number of 7.5. These results show spectral radii of 0.855 and 0.822 respectively. The differences in convergence rate between the finite-difference and finite-volume codes are probably due to the additional enthalpy damping step used in the finite-volume code.

Circles on Fig. 9 show equal CPU times of 16.7 sec on the Cray X-MP. In terms of CPU time required to reach a certain convergence level, the implicitly-smoothed Runge-Kutta scheme alone is marginally faster than Ni's scheme. The two FAS

multigrid schemes are similar to each other in performance and are approximately 8.5 times faster than Ni's scheme or the implicitly-smoothed scheme alone.

#### Error Smoothing Versus Time-Marching

In the original work on multigrid for elliptic equations, point Jacobi or Gauss-Seidel iteration schemes were used because of their ability to smooth high-frequency errors. The fact that these schemes could be interpreted as time-marching schemes was considered irrelevant. In contrast Ni's scheme has been referred to as "hyperbolic multigrid" with its main purpose being to advance rapidly in time using coarse-grid information.

It is not clear which interpretation is more appropriate for analysis of FAS multigrid schemes applied to hyperbolic problems. Jameson<sup>9</sup> bases his analysis on the smoothing properties of the Runge-Kutta scheme. On the other hand, Jespersen<sup>16</sup> has shown that multigrid solutions possess time-accurate properties. Thus FAS may work because it allows larger time-steps on the coarser grids.

Using a stability analysis of the linear one-dimensional convection equation with fourth-difference artificial dissipation it is possible to choose the multistage scheme parameters  $\alpha_k$ ,  $\mu$ ,  $\epsilon$ , and  $\lambda$  to maximize the smoothing and the time step. Jameson has published several sets of these parameters in Ref. 9. During the current work Schaffer developed an optimization code that chooses these parameters to minimize the area under the amplification factor curve.

Experience with these "optimal parameter schemes" has been inconclusive. It is clear that both large Courant numbers and high smoothing lead to fast convergence. However, schemes with equal Courant numbers but very different one-dimensional smoothing properties often have similar convergence rates. We suspect that the effects of stretched grids on the artificial viscosity and the effects of applying implicit residual smoothing as a sequence of one-dimensional operators are such that the amplification factor for the two-dimensional codes do not look much like the one-dimensional model results. Also, Jameson has shown that amplification factors can vary considerably from grid to grid<sup>17</sup> making it difficult to predict the behavior of the overall multigrid scheme.

#### Viscous Results

Experimental data for the interaction of an oblique shock wave with a laminar boundary layer have been published by Hakkinen et al.<sup>18</sup> This case has been computed by many researchers, notably by McCormack and Baldwin<sup>19</sup> using a 32 by 32 mesh.

Here we have computed this flow with a 113 by 41 mesh. The free-stream Mach number is 2. The upper boundary was treated as an inviscid wall which was bent  $3.091^\circ$  to generate an oblique shock that intersects the lower wall at 0.16 ft at an angle of  $32.585^\circ$ . The grid has a constant x-spacing of 0.003 ft and a y-spacing at the wall of 0.0001 ft, stretching geometrically through the boundary layer to a constant spacing above. At the wall the grid aspect ratio is 30:1.

Figure 10 shows an Euler solution computed on the viscous grid. Static pressure contours are shown at the bottom and the lower-wall pressure distribution is compared to experimental data at the top. The inviscid lower wall has no effect on the flow except to reflect the incident shock. The computed pressure distribution shows some overshoot after the reflected shock but a good comparison with the overall shock strength.

Convergence histories for the Ni, Runge-Kutta, and FAS multigrid schemes are shown in Fig. 11. All three schemes show fast initial convergence for this purely supersonic flow, but all slow down abruptly after a three-to-four decade drop in the residuals, presumably because of the very fine grid at the wall. It appears that the multigrid schemes parallel the convergence behavior of the fine-grid scheme but at a faster rate. The Ni scheme shows a surprising dip and jump in convergence near the end. Circles showing convergence levels at equal CPU times of 22.3 sec indicate that the three schemes are comparable in speed for this case.

Figure 12 shows the thin-layer Navier-Stokes solution for this problem. Flow conditions were chosen to give a Reynolds number of  $2.96 \times 10^5$  at the shock ( $x = 0.16$  ft). Static pressure contours at the bottom of Fig. 12 show not only the incident and reflected shocks but a leading edge shock that reflects from the upper wall and compression waves generated by the separation bubble. The lower-wall static pressure distribution is compared to the experimental data at the center of Fig. 12 and shows good agreement after the leading-edge and reflected shocks, but slightly under predicts the plateau pressure in the separated region. Computed skin friction shown at the top of Fig. 12 shows very good agreement with the data, including separation and reattachment points. Note that skin friction was not measured in the reverse flow region.

Viscous flow convergence rates shown in Fig. 13 are slower than for the Euler case and show the same tendency to slow down, but not so abruptly. Circles at equal CPU times of 28 sec show that the Runge-Kutta scheme with implicit smoothing is slightly faster than Ni's scheme, but that the FAS multigrid scheme is about 2.1 times faster.

### Conclusion

In this work we have developed and compared the Ni, Jameson, and Brandt types of multigrid schemes. Each scheme uses the Runge-Kutta method as the basic algorithm and that method has been studied without multigrid as well.

We have demonstrated that the efficiency of the Runge-Kutta scheme increases with number of stages and that convergence rate increases with Courant number. We have shown that Ni's scheme gives one fine-grid time step on each coarse grid but that its Courant limit of one limits the effectiveness of the scheme. We propose a multistage coarse-grid scheme to improve the Courant limit. Using implicit residual smoothing to increase the Courant limit of the Runge-Kutta scheme without multigrid gives a scheme slightly more efficient than Ni's.

We have shown that Jameson's finite-volume multigrid scheme and a finite-difference scheme based on Brandt's work are similar in form, but

with differences between interpolation methods for the cell-centered and node-centered schemes. For an inviscid transonic model problem the finite-difference multigrid scheme is about 8.5 times faster than the best Ni or Runge-Kutta scheme without multigrid. Jameson's finite-volume code is about three times faster than that in terms of iterations, probably due to the additional enthalpy damping step. The two multigrid codes are about equal in terms of CPU time.

Convergence rates of all methods decrease on highly-stretched grids. For a shock-boundary layer interaction problem the finite-difference multigrid code was still 2.1 times faster than the best Ni or Runge-Kutta scheme without multigrid. We believe that significant improvements can still be made in multigrid convergence rates for viscous flows.

### References

1. Ni, R.H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, No. 11, Nov. 1982, pp. 1565-1571.
2. Johnson, G.M., "Convergence Acceleration of Viscous Flow Computations," NASA TM-83039, 1982.
3. Johnson, G.M., "Flux-Based Acceleration of the Euler Equations," NASA TM-83453, 1983.
4. Chima, R.V., and Johnson, G.M., "Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm," *AIAA Journal*, Vol. 23, No. 1, Jan. 1985, pp. 23-32.
5. Chima, R.V., "Analysis of Inviscid and Viscous Flows in Cascades with an Explicit Multiple-Grid Algorithm," *AIAA Journal*, Vol. 23, No. 10, Oct. 1985, pp. 1556-1563.
6. Hall, M.G., and Salas, M.D., "Comparison of Two Multigrid Methods for the Two-Dimensional Euler Equations," AIAA Paper 85-1515-CP, July 1985.
7. Jameson, A., and Baker, T.J., "Multigrid Solution of the Euler Equations for Aircraft Configurations," AIAA Paper 84-0093, Jan. 1984.
8. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.
9. Jameson, A., and Baker, T.J., "Multigrid Solution of the Euler Equations for Aircraft Configurations," AIAA Paper 84-0093, Jan. 1984.
10. Martinelli, L., Jameson, A., and Grasso, F., "A Multigrid Method for the Navier-Stokes Equations," AIAA Paper 86-0208, Jan. 1986.
11. Brandt, A., "Multi-level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, Vol. 31, No. 138, Apr. 1977, pp. 333-390.
12. Swanson, R.C., and Turkel, E., "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations," AIAA Paper 85-0035, Jan. 1985.
13. Hollanders, H., Lerat, A., and Peyret, R., "Three-Dimensional Calculation of Transonic Viscous Flows by an Implicit Method," *AIAA Journal*, Vol. 23, No. 11, Nov. 1985, pp. 1670-1678.
14. Jameson, A., and Baker, T.J., "Solution of the Euler Equations for Complex Configurations," *AIAA 6th Computational Fluid Dynamics Conference*, AIAA, New York, 1983, pp. 293-302.

15. Turkel, E., "Acceleration to a Steady-State for the Euler Equations," Numerical Methods for the Euler Equations of Fluid Dynamics, edited by F. Angrand, et al., SIAM, Philadelphia, 1985, pp. 281-311.
16. Jespersen, D.C., "A Time-Accurate Multiple-Grid Algorithm," AIAA 7th Computational Fluid Dynamics Conference, AIAA, New York, 1985, pp. 58-66.
17. Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," MAE Report 1743, Princeton University, Oct. 1985.
18. Hakkinen, R.J., Greber, I., Trilling, L., and Abarbanel, S.S., "The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer," NASA Memo 2-18-59W, Mar. 1959.
19. McCormack, R.W., and Baldwin, B.S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interactions," AIAA Paper 75-1, Jan. 1975.

TABLE I. - COEFFICIENTS  $\alpha_i$  AND  
APPROXIMATE MAXIMUM COURANT  
NUMBER  $\lambda^*$  FOR k-STAGE  
RUNGE-KUTTA SCHEMES  
USED HERE

k	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\lambda^*$
2	1.2	1	---	---	---	.95
3	.6	.6	1	---	---	1.5
4	1/4	1/3	1/2	1	---	2.8
5	1/4	1/6	3/8	1/2	1	3.6

ORIGINAL PAGE IS  
OF POOR QUALITY

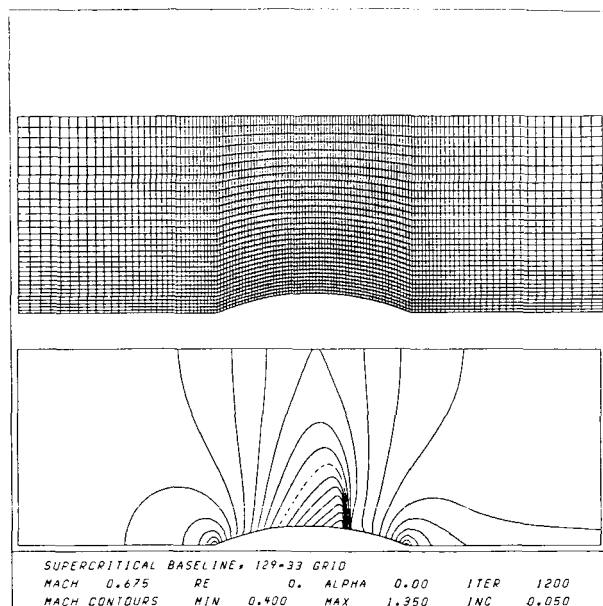


FIGURE 1.- TOP: 129x33 GRID OVER A 10% THICK CIRCULAR ARC BUMP. BOTTOM: MACH CONTOURS,  $M_{\infty} = 0.675$ , SONIC LINE IS DASHED.

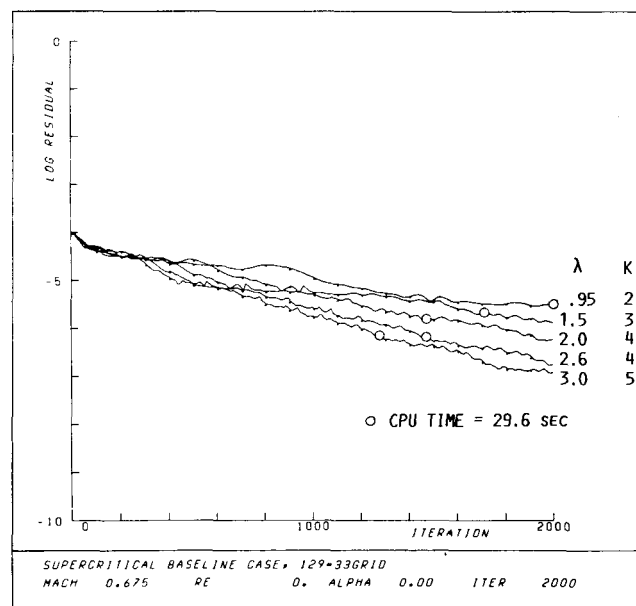


FIGURE 2.- COMPARISON OF CONVERGENCE RATES FOR k-STAGE SCHEMES AT VARIOUS COURANT NUMBERS  $\lambda$ .

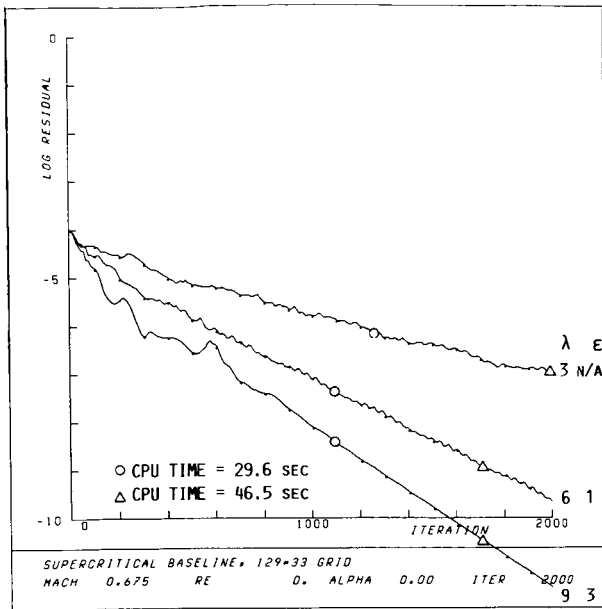


FIGURE 3.- COMPARISON OF CONVERGENCE RATES AT COURANT NO.  $\lambda$ . 5-STAGE SCHEME, 2 EVALUATIONS OF DISSIPATION. IMPLICIT SMOOTHING WITH PARAMETER  $\epsilon$  AFTER EACH STAGE.

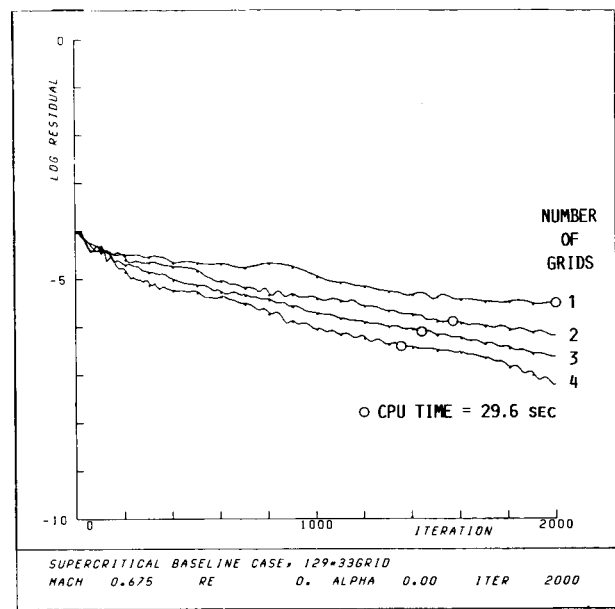


FIGURE 4.- COMPARISON OF CONVERGENCE RATES FOR N-GRID N1-SCHMES.

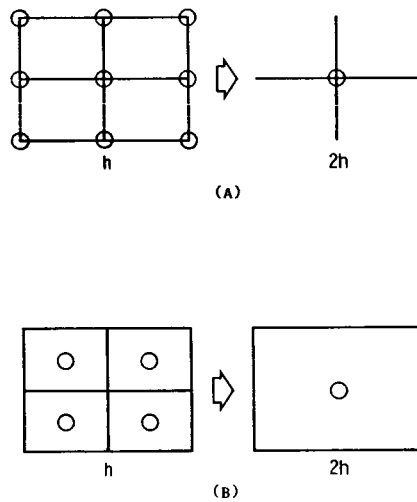


FIGURE 5.- RELATIONSHIP BETWEEN FINE- AND COARSE-GRID POINTS. (A) NODE-CENTERED (FINITE DIFFERENCE) SCHEMES. (B) CELL-CENTERED (FINITE VOLUME) SCHEMES.

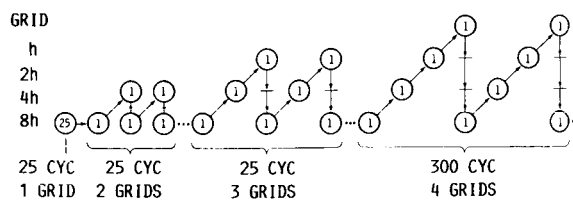


FIGURE 6.- TYPICAL FMG CYCLE FOR THE FINITE-DIFFERENCE CODE.

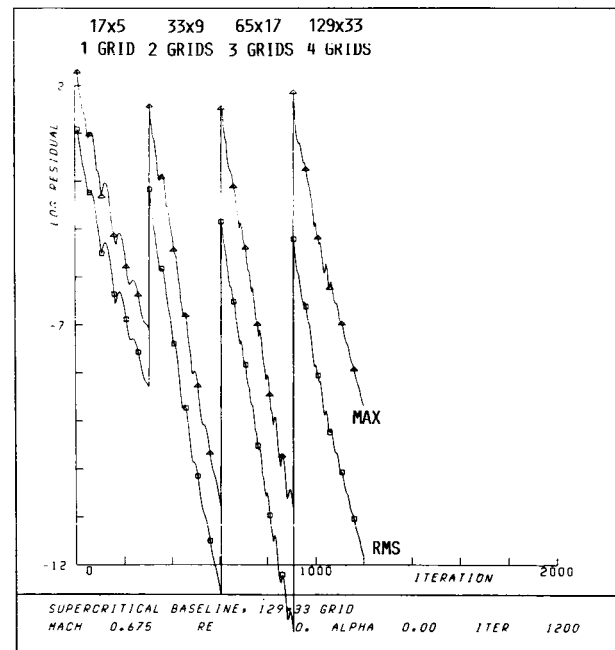


FIGURE 7.- FULL MULTIGRID CONVERGENCE RATES (MAX AND RMS DENSITY RESIDUAL) FOUR STAGES,  $\lambda = 5.2$ .

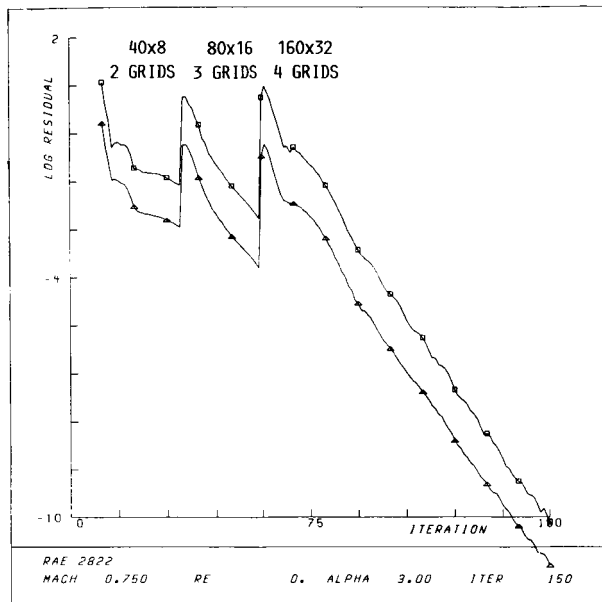


FIGURE 8.- CONVERGENCE HISTORY FOR INVISCID FLOW ABOUT A RAE 2822 AIRFOIL USING THE FINITE-VOLUME MULTIGRID METHOD.

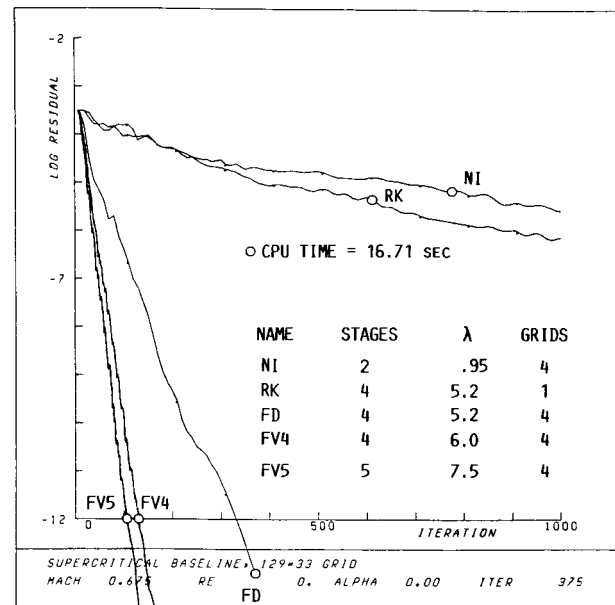


FIGURE 9.- COMPARISON OF CONVERGENCE RATES FOR FIVE DIFFERENT SCHEMES.

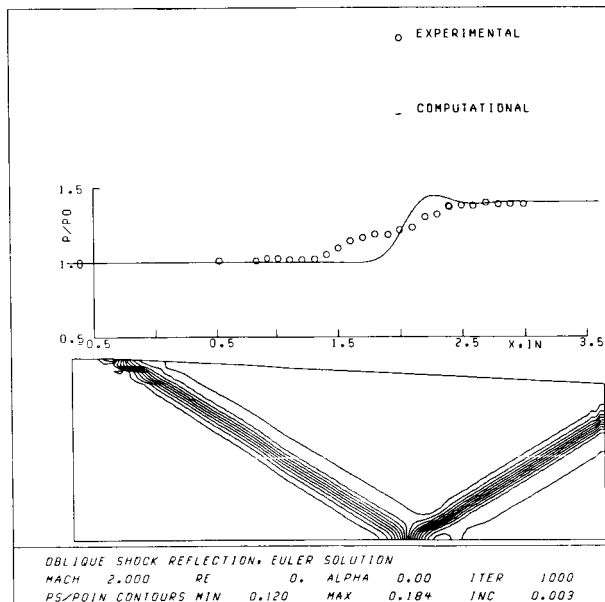


FIGURE 10.- OBLIQUE SHOCK REFLECTION, EULER SOLUTION.  
TOP: COMPARISON OF COMPUTED AND MEASURED STATIC PRESSURES.  
BOTTOM: PRESSURE CONTOURS.

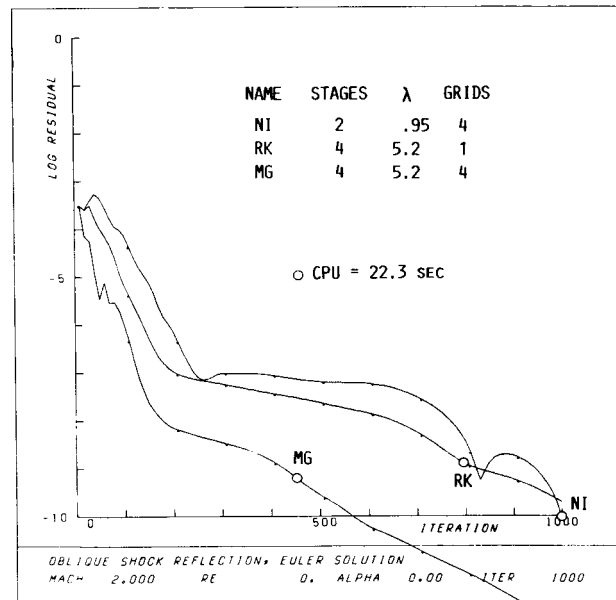


FIGURE 11.- COMPARISON OF CONVERGENCE RATES FOR INVISCID FLOW.

ORIGINAL PAGE IS  
OF POOR QUALITY

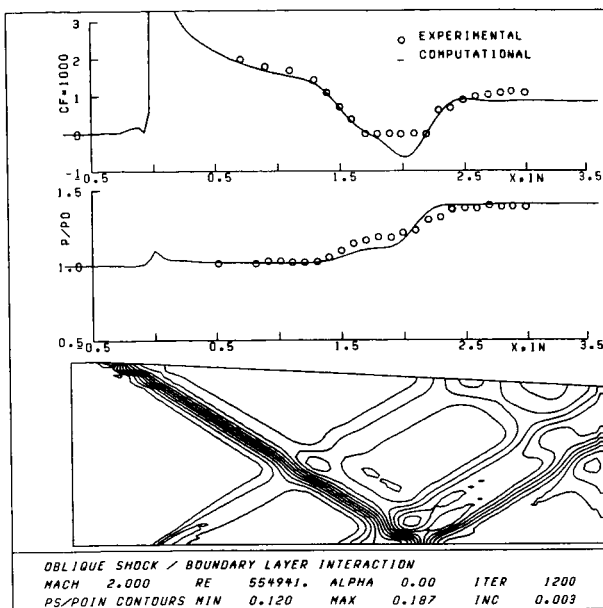


FIGURE 12.- OBLIQUE SHOCK/BOUNDARY LAYER INTERACTION.  
 TOP: COMPARISON OF COMPUTED AND MEASURED SKIN FRICTION  
 MIDDLE: COMPARISON OF COMPUTED AND MEASURED STATIC PRESSURES.  
 BOTTOM: PRESSURE CONTOURS.

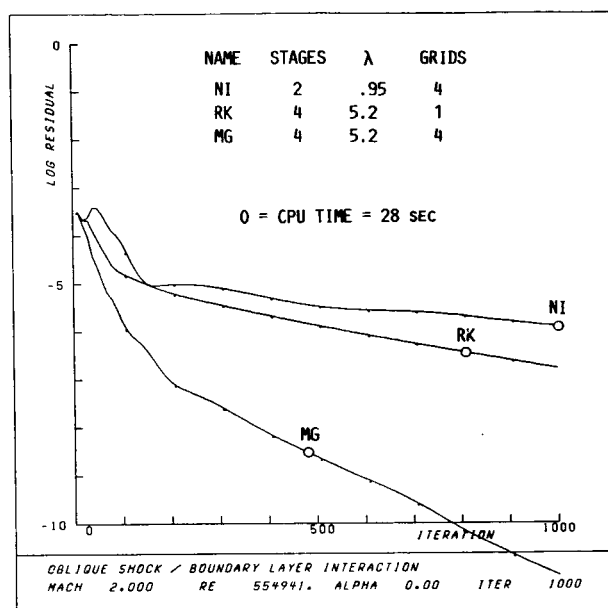


FIGURE 13.- COMPARISON OF CONVERGENCE RATES FOR LAMINAR FLOW.

ORIGINAL PAGE IS  
 OF POOR QUALITY

1. Report No. <b>NASA TM-88878 ICOMP-86-3</b>		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle  <b>Comparison of Three Explicit Multigrid Methods for the Euler and Navier-Stokes Equations</b>				5. Report Date	
				6. Performing Organization Code <b>505-31-04</b>	
7. Author(s)  <b>Rodrick V. Chima, Eli Turkel, and Steve Schaffer</b>				8. Performing Organization Report No. <b>E-3126</b>	
				10. Work Unit No.	
9. Performing Organization Name and Address  <b>National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135</b>				11. Contract or Grant No.	
				13. Type of Report and Period Covered <b>Technical Memorandum</b>	
12. Sponsoring Agency Name and Address  <b>National Aeronautics and Space Administration Washington, D.C. 20546</b>				14. Sponsoring Agency Code	
15. Supplementary Notes <b>Prepared for the 25th Aerospace Sciences Meeting, sponsored by the American Institute of Aeronautics and Astronautics, Reno, Nevada, January 12-15, 1987. Rodrick V. Chima, NASA Lewis Research Center. Eli Turkel, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Virginia 23665 and NASA Lewis Research Center; Steve Schaffer, Institute for Computational Mechanics in Propulsion, NASA Lewis Research Center (work funded under Space Act Agreement C99066-G), presently at Berea College, Berea, Kentucky 40403.</b>					
16. Abstract  Three explicit multigrid methods, Ni's method, Jameson's finite-volume method, and a finite-difference method based on Brandt's work, are described and compared for two model problems. All three methods use an explicit multistage Runge-Kutta scheme on the fine grid, and this scheme is also described. Convergence histories for inviscid flow over a bump in a channel for the fine-grid scheme alone show that convergence rate is proportional to Courant number and that implicit residual smoothing can significantly accelerate the scheme. Ni's method was slightly slower than the implicitly-smoothed scheme alone. Brandt's and Jameson's methods are shown to be equivalent in form but differ in their node versus cell-centered implementations. They are about 8.5 times faster than Ni's method in terms of CPU time. Results for an oblique shock/boundary layer interaction problem verify the accuracy of the finite-difference code. All methods slowed considerably on the stretched viscous grid but Brandt's method was still 2.1 times faster than Ni's method.					
17. Key Words (Suggested by Author(s))  <b>Computational fluid dynamics; Multigrid; Euler equations; Navier-Stokes equations</b>				18. Distribution Statement  <b>Unclassified - unlimited STAR Category 01</b>	
19. Security Classif. (of this report) <b>Unclassified</b>		20. Security Classif. (of this page) <b>Unclassified</b>		21. No. of pages	
				22. Price*	