

Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm

Rodrick V. Chima*

NASA Lewis Research Center, Cleveland, Ohio
and

Gary M. Johnson†

Colorado State University, Fort Collins, Colorado

A multiple-grid algorithm for use in efficiently obtaining steady solutions to the Euler and Navier-Stokes equations is presented. The convergence of the explicit MacCormack algorithm on a fine grid is accelerated by propagating transients from the domain using a sequence of successively coarser grids. Both the fine- and coarse-grid schemes are readily vectorizable. The combination of multiple-gridding and vectorization results in substantially reduced computational times for the numerical solution of a wide range of flow problems. Results are presented for subsonic, transonic, and supersonic inviscid flows and for subsonic attached and separated laminar viscous flows. Work reduction factors over a scalar, single-grid algorithm range as high as 76.8.

Introduction

STEADY solutions to both the Euler and Navier-Stokes equations are commonly computed as temporal asymptotes to the unsteady equations of motion with steady boundary conditions applied. This is done because the unsteady equations are either purely hyperbolic, in the case of Euler equations, or hyperbolic-parabolic, for the Navier-Stokes equations, and are thus amenable to numerical solution by robust time-marching procedures. Furthermore, such procedures are relatively easy to implement and, to the extent that the computation is time-accurate, allow physical interpretation of the convergence history.

Relaxation procedures for solving the steady, compressible versions of the Euler or Navier-Stokes equations are a topic of current research and have not as yet resulted in production algorithms.

Both explicit and implicit time-marching procedures are presently in widespread use. The explicit methods are simple, easily vectorizable, and allow a good deal of flexibility in the treatment of boundary conditions. Their largest shortcoming lies in their conditional stability, which may place rather severe limitations on the time step size permissible on a given grid. When only a steady solution is sought and the accurate resolution of transients is of no consequence, explicit methods may consequently exhibit poor convergence rates.

Implicit methods are one possible remedy to the slow convergence of explicit schemes. These methods, at the expense of a higher operations count, are generally unconditionally linearly stable and hopefully permit time steps to be taken as large as is consistent with accuracy requirements. In practice, large time steps may excite nonlinear instabilities, and the choice of boundary condition implementation may introduce a stability limit.

The present work maintains the advantages of an explicit procedure while using a multiple-grid convergence ac-

celeration scheme to substantially improve the convergence rate at relatively low computational expense, in terms of increased operations count. This results in a net reduction in the computational work required to produce a converged solution. As the multiple-grid scheme retains the explicit nature of the underlying fine-grid procedure, it may be vectorized in a straightforward manner. This leads to a further reduction in computational time. In combination, multiple-gridding and vectorization produce a quite efficient explicit algorithm for the solution of both inviscid and viscous flow problems.

Equations of Motion

The nondimensional equations of motion may be written in conservation law form as

$$q_t = -(F_x + G_y) \quad (1)$$

where, for the full Navier-Stokes equations,

$$F \equiv f - Re^{-1}r \quad G \equiv g - Re^{-1}s$$

while, for their thin-layer version,

$$F \equiv f \quad G \equiv g - Re^{-1}\bar{s}$$

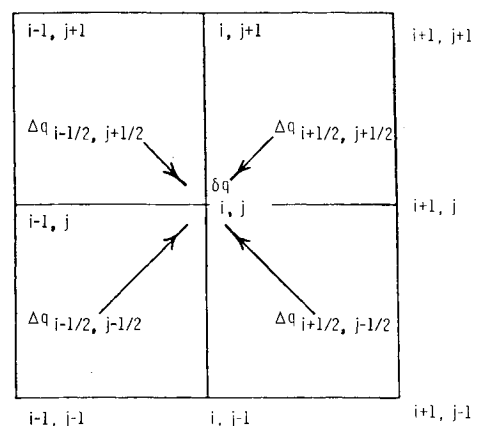


Fig. 1 One-step Lax-Wendroff scheme.

Received June 2, 1983; presented as Paper 83-1893 at the AIAA Sixth Computational Fluid Dynamics Conference, Danvers, Mass., July 13-15, 1983; revision received Feb. 28, 1984. This paper is declared a work of the U.S. Government and therefore is in the public domain.

*Research Engineer. Member AIAA.

†Director for Computational Research, Institute for Computational Studies. Associate Fellow AIAA.

and, for the Euler equations,

$$F \equiv f \quad G \equiv g$$

where

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \quad f = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E+p)u \end{bmatrix} \quad g = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}$$

$$r = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \kappa Pr^{-1}(\gamma-1)^{-1}(a^2)_x + u\tau_{xx} + v\tau_{xy} \end{bmatrix}$$

$$s = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \kappa Pr^{-1}(\gamma-1)^{-1}(a^2)_y + u\tau_{xy} + v\tau_{yy} \end{bmatrix}$$

$$\tilde{s} = \begin{bmatrix} 0 \\ \mu u_y \\ (\lambda + 2\mu)v_y \\ \kappa Pr^{-1}(\gamma-1)^{-1}(a^2)_y + \mu(uu_y + 2vv_y) + \lambda uv_y \end{bmatrix}$$

$$\tau_{xx} = (\lambda + 2\mu)u_x + \lambda v_x$$

$$\tau_{xy} = \mu(u_y + v_x)$$

$$\tau_{yy} = (\lambda + 2\mu)v_y + \lambda u_x$$

where ρ , u , v , p , a , and E are density, velocity components in the x and y directions, pressure, sound speed, and total energy per unit volume, respectively. The total energy per unit volume may be expressed as

$$E = \rho[e + \frac{1}{2}(u^2 + v^2)]$$

where the specific internal energy e is related to the pressure and density by the simple law of a calorically perfect gas, $p = (\gamma - 1)\rho e$, with γ denoting the ratio of specific heats. The coefficient of thermal conductivity κ and the viscosity coefficients λ and μ are assumed to be functions only of temperature. Furthermore, by invoking Stokes' assumption of zero bulk viscosity, λ may be expressed in terms of the dynamic viscosity μ as $\lambda = -\frac{2}{3}\mu$. Re and Pr denote the Reynolds and Prandtl numbers, respectively.

Although, for simplicity, the equations of motion are presented here in Cartesian coordinates, Viviand¹ has shown that their strong conservation law form may be maintained under an arbitrary time-dependent transformation of coordinates. Explicit detail concerning the generalized coordinate version of these equations has been provided by Steger² and need not be repeated here.

We note that the thin-layer approximation, in the words of Baldwin and Lomax,³ "... evolves directly from a realistic assessment of what is really being computed in a typical high Reynolds number Navier-Stokes simulation." A highly stretched mesh is used to resolve the large flow gradients normal to the vorticity-generating surface. Consequently,

because of limitations on computer capacity, the diffusion terms involving derivatives parallel to the surface are not resolved well enough to merit their computation.

Similar viscous terms are also neglected in the classical boundary-layer approximation. However, while the boundary-layer approximation replaces the normal momentum equation with the assumption that the normal pressure gradient is zero across the viscous layer, all momentum equations are retained in the thin-layer approximation and no assumptions are made concerning the pressure. Consequently, the separation point is not a singularity of the thin-layer model equations nor do the problems associated with matching a boundary-layer solution to an inviscid outer flow occur when they are used.

In practice, the thin-layer assumption is implemented by using a body-fitted coordinate system and neglecting the viscous terms in the coordinate direction along the body. For Cartesian coordinates, with x representing the body-conforming coordinate, the thin-layer version of the Navier-Stokes equations is as given above.

Specification of initial and boundary conditions complete the formulation of the problem. Initial conditions are specified as uniform flow at the isentropic Mach number implied by the ratio of exit static pressure to inlet total pressure. Boundary conditions are specified as follows: At the inlet, total temperature, total pressure, and flow angle are specified, while at the exit the static pressure is specified. Along solid surfaces the tangency condition is applied for inviscid flow, or the no-slip condition is applied and the temperature is specified for viscous flow.

Fine-Grid Solution Procedure

The fine-grid integration scheme employed in this work is the two-step Lax-Wendroff method known as the MacCormack⁴ scheme. Schemes of the Lax-Wendroff type may be arrived at intuitively by using Taylor's theorem to write the approximation

$$\delta q = \Delta t q_t + (\Delta t^2/2) q_{tt} \quad (2)$$

where we define the "correction" to q such that

$$\delta q \equiv q(t + \Delta t) - q(t)$$

Since we seek solutions to Eq. (1), time derivatives may be expressed as space derivatives:

$$q_t = -(F_x + G_y), \quad q_{tt} = [A(F_x + G_y)]_x + [B(F_x + G_y)]_y$$

where A and B are the Jacobian matrices (see Steger² for details):

$$A \equiv \frac{\partial F}{\partial q}, \quad B \equiv \frac{\partial G}{\partial q}$$

Substitution into Eq. (2) results in

$$\delta q = -\Delta t(F_x + G_y) + \frac{\Delta t^2}{2} \{ [A(F_x + G_y)]_x + [B(F_x + G_y)]_y \} \quad (3)$$

Second-order accurate spatial discretization of Eq. (3) then yields a one-step Lax-Wendroff method.

Ni's Method

Prior to discussing MacCormack's scheme, we derive the fine-grid solution procedure used by Ni.⁵ This is a necessary prerequisite to the development of the coarse-grid acceleration scheme.

If we make the following finite-volume-type approximations,

$$\begin{aligned} (F_x + G_y)_{i,j} &= \frac{I}{8\Delta x} [(F_{i+1,j+1} + 2F_{i+1,j} + F_{i+1,j-1}) \\ &\quad - (F_{i-1,j+1} + 2F_{i-1,j} + F_{i-1,j-1})] \\ &\quad + \frac{I}{8\Delta y} [(G_{i+1,j+1} + 2G_{i,j+1} + G_{i+1,j-1}) \\ &\quad - (G_{i-1,j+1} + 2G_{i,j+1} + G_{i-1,j-1})] \end{aligned}$$

and define the "change" in q at cell centers such that,

$$\begin{aligned} \Delta q_{i+\frac{1}{2},j+\frac{1}{2}} &\equiv -\frac{\Delta t}{2\Delta x} [(F_{i+1,j} + F_{i+1,j+1}) - (F_{i,j} + F_{i,j+1})] \\ &\quad - \frac{\Delta t}{2\Delta y} [(G_{i,j+1} + G_{i+1,j+1}) - (G_{i,j} + G_{i+1,j})] \end{aligned} \quad (4)$$

it then follows that a discrete approximation to the first-order term in Eq. (3) may be written as

$$\begin{aligned} -\Delta t(F_x + G_y)_{i,j} &= \frac{1}{4} [\Delta q_{i-\frac{1}{2},j-\frac{1}{2}} + \Delta q_{i-\frac{1}{2},j+\frac{1}{2}} \\ &\quad + \Delta q_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta q_{i+\frac{1}{2},j-\frac{1}{2}}] \end{aligned} \quad (5)$$

Consistent with the above approximations and definition, we may write the approximation,

$$-\Delta t(F_x + G_y)_{i+\frac{1}{2},j+\frac{1}{2}} = \Delta q_{i+\frac{1}{2},j+\frac{1}{2}}$$

This motivates the definitions

$$\Delta F_{i+\frac{1}{2},j+\frac{1}{2}} \equiv A_{i+\frac{1}{2},j+\frac{1}{2}} \Delta q_{i+\frac{1}{2},j+\frac{1}{2}}$$

$$\Delta G_{i+\frac{1}{2},j+\frac{1}{2}} \equiv B_{i+\frac{1}{2},j+\frac{1}{2}} \Delta q_{i+\frac{1}{2},j+\frac{1}{2}}$$

If we then approximate the second-order terms in Eq. (3) as

$$\begin{aligned} -\Delta t\{[A(F_x + G_y)]_x\}_{i,j} &= \frac{I}{2\Delta x} [\Delta F_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta F_{i+\frac{1}{2},j-\frac{1}{2}} \\ &\quad - \Delta F_{i-\frac{1}{2},j-\frac{1}{2}} - \Delta F_{i-\frac{1}{2},j+\frac{1}{2}}] \end{aligned} \quad (6a)$$

$$\begin{aligned} -\Delta t\{[B(F_x + G_y)]_y\}_{i,j} &= \frac{I}{2\Delta y} [\Delta G_{i+\frac{1}{2},j+\frac{1}{2}} + \Delta G_{i-\frac{1}{2},j+\frac{1}{2}} \\ &\quad - \Delta G_{i-\frac{1}{2},j-\frac{1}{2}} - \Delta G_{i+\frac{1}{2},j-\frac{1}{2}}] \end{aligned} \quad (6b)$$

we may combine Eqs. (3), (5), and (6) to yield

$$\begin{aligned} \delta q_{i,j} &= \frac{I}{4} \left[\Delta q + \frac{\Delta t}{\Delta x} \Delta F + \frac{\Delta t}{\Delta y} \Delta G \right]_{i-\frac{1}{2},j-\frac{1}{2}} \\ &\quad + \frac{I}{4} \left[\Delta q + \frac{\Delta t}{\Delta x} \Delta F - \frac{\Delta t}{\Delta y} \Delta G \right]_{i-\frac{1}{2},j+\frac{1}{2}} \\ &\quad + \frac{I}{4} \left[\Delta q - \frac{\Delta t}{\Delta x} \Delta F - \frac{\Delta t}{\Delta y} \Delta G \right]_{i+\frac{1}{2},j+\frac{1}{2}} \\ &\quad + \frac{I}{4} \left[\Delta q - \frac{\Delta t}{\Delta x} \Delta F + \frac{\Delta t}{\Delta y} \Delta G \right]_{i+\frac{1}{2},j-\frac{1}{2}} \end{aligned} \quad (7)$$

Equations (4) and (7) constitute the one-step Lax-Wendroff method used as a basic integration scheme by Ni. He gives the following heuristic interpretation to these equations: the first

calculates the change Δq occurring in a control volume during the increment Δt , while the second distributes the effects of the changes occurring in four nearest-neighbor control volumes to their common central nodal point where they are combined to form the correction δq to the vector of conservation variables, as illustrated in Fig. 1. This interpretation motivates the construction of the coarse-grid acceleration scheme to be discussed subsequently.

Notice that Ni's scheme may also be thought of as a two-step scheme with a full time increment predictor defined by Eq. (4) and a corrector defined by Eq. (7). However, such an interpretation would not be consistent with the general practice of avoiding the computation of Jacobian matrices in two-step schemes.

MacCormack's Method

Following Richtmyer⁶ many two-step Lax-Wendroff schemes have been developed. They have superseded the one-step schemes by virtue of their lower operations counts. MacCormack's method is a particularly popular and efficient member of this class. The forward predictor-backward corrector version of this method may be written as

$$\Delta q_{i,j} = -\frac{\Delta t}{\Delta x} (F_{i+1,j}^n - F_{i,j}^n) - \frac{\Delta t}{\Delta y} (G_{i,j+1}^n - G_{i,j}^n)$$

$$\tilde{q}_{i,j} = q_{i,j}^n + \Delta q_{i,j}$$

$$\delta q_{i,j} = \frac{\Delta q_{i,j}}{2} - \frac{\Delta t}{2\Delta x} (\tilde{F}_{i,j} - \tilde{F}_{i-1,j}) - \frac{\Delta t}{2\Delta y} (\tilde{G}_{i,j} - \tilde{G}_{i,j-1})$$

$$q_{i,j}^{n+1} = q_{i,j}^n + \delta q_{i,j}$$

where

$$\tilde{F}_{i,j} = F(\tilde{q}_{i,j}), \quad \tilde{G}_{i,j} = G(\tilde{q}_{i,j})$$

First derivatives in the viscous terms are backward differenced in the predictor and forward differenced in the corrector.

A second-order dissipative term is used to enforce the entropy condition across shocks in the supercritical results presented later. Second derivatives in the dissipation term are multiplied by the absolute value of the density gradient. This improves shock resolution but makes the dissipation first-order in the vicinity of shocks. Dissipation was not used for any of the subcritical results.

This approach to solving fluid flow problems is quite robust and has been in widespread and successful use for some time, both for the time-accurate computation of unsteady flow and for the time-asymptotic solution of steady flow problems. In the latter case, where accurate resolution of physical transients is not required, the numerical stability limitation inherent in this explicit method may severely restrict the speed of its convergence to the steady state. Providing a method to accelerate convergence in this case is one objective of this work.

One method used throughout this work to accelerate the convergence rate of the MacCormack algorithm is the use of a spatially variable time step. Here the time step Δt , usually taken as the global minimum for stability, is replaced by Δt_{ij} , such that the Courant number is constant over the grid, typically 0.9. We calculate Δt_{ij} based on the inviscid stability limit and constant initial conditions, so it varies as a function of grid stretching only. The time step is stored and is not updated during the computations.

Coarse-Grid Acceleration Scheme

Given the fine-grid corrections, which may be computed by any one- or two-step Lax-Wendroff scheme, as shown in Johnson,⁷ we wish to use successively coarser grids to

propagate these corrections throughout the computational domain, thus accelerating convergence to the steady state while maintaining the accuracy determined by the fine-grid discretization. Define a fine grid such that the number of points in each direction is expressible as $n(2^p) + 1$ for p and n integers such that $p \geq 0$ is the number of grid coarsenings and $n \geq 2$ the number of intervals on the coarsest grid. Then successively coarser grids can be defined by successive deletion of every other point in each coordinate direction.

Full Coarse-Grid Scheme

The full coarse-grid acceleration scheme, as illustrated in Fig. 2, replaces the computation of coarse-grid changes using Eq. (4) with a restriction of the latest fine-grid correction. This restricted fine-grid correction is then distributed according to a coarse-grid version of Eq. (7) to obtain a coarse-grid correction. This in turn is prolonged to the fine grid to become the new fine-grid correction. One time-cycle of the multiple-grid scheme is composed of an application of some Lax-Wendroff scheme on the fine grid followed by an application of the coarse-grid solution procedure to each successively coarser grid. The flow of information in this process is depicted in Fig. 3. Boundary conditions are updated only during the fine-grid computation. This decouples the coarse-grid acceleration scheme from the details of boundary condition implementation.

In the basic integration scheme, a change at one grid point affects only its nearest neighbors while, in a k -level multiple-grid scheme, the same change affects all points up to 2^{k-1} mesh spacings distant. Furthermore, since the change is always determined by information from the fine grid and simply propagated by the distribution formulas for coarser grids, fine-grid accuracy is maintained. This concept for convergence acceleration was introduced by Ni for use in conjunction with his one-step Lax-Wendroff schemes, as described previously. He illustrated its utility by solving the homoeuthalpic two-dimensional Euler equations.

Convective Coarse-Grid Scheme

In Ref. 8, consideration of the physical processes being modeled in a viscous flow computation led to the formulation of an alternative coarse-grid scheme. Dissipative effects have a local character and their influence need not be taken into account in the construction of coarse-grid distribution formulae. Rather, it is the convective terms, with their global character, that are the key element in coarse-grid propagation. Hence, a coarse-grid scheme for viscous flow computations may be formulated on the basis of the inviscid equations of motion. Such a convective coarse-grid scheme is inherently more efficient than the full coarse-grid scheme because of the diminished computational effort associated

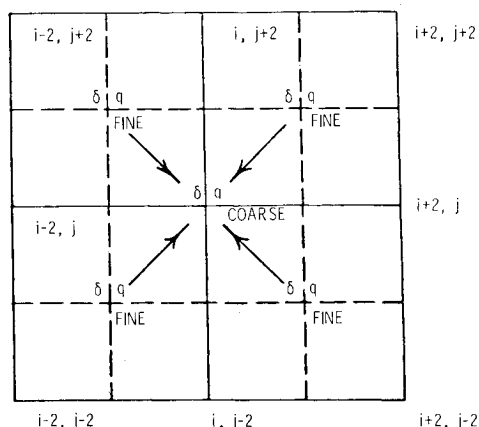


Fig. 2 Coarse-grid scheme.

with forming the Jacobian matrices of the Euler flux vectors rather than those of the viscous flux vectors. An additional benefit is that the convective coarse-grid scheme leads to a multiple-grid convergence acceleration procedure that is independent of the nature of the dissipative terms retained in the viscous model equations. That is to say that the coarse-grid scheme based on the Euler equations may be employed, without modification, to accelerate the convergence of viscous flow computations based on the Navier-Stokes equations, the thin-layer equations, or any other viscous model equations that contain the full inviscid Euler equations. This claim is supported by the computational results presented subsequently.

Vectorization

The original multiple-grid MacCormack code was developed on the IBM 370/3033 computer at the NASA Lewis Research Center. This is a scalar machine that runs at about one-half the speed of a CDC 7600. All scalar timings presented in this paper were obtained on the Lewis 370 with full compiler optimization.

It is well known that explicit algorithms in general and MacCormack's algorithm in particular are highly amenable to vectorization (e.g., Shang, et al.⁹). As the multiple-grid scheme presented here is explicit, it may also be vectorized in a straightforward manner. Indeed, we have vectorized the original research code for use on the Cray 1-S computer recently installed at NASA Lewis and have decreased its execution time by factors ranging from 11.2 to 13.3 over the scalar code. The changes made to the code can be grouped into the following five categories:

- 1) Unrolling short inner DO loops over the four conservation variables. This change allows vectorization over longer outer loops. It also eliminates considerable loop overhead, thereby improving scalar performance.
- 2) Revising DO loop ordering to make innermost loops the longest. The code is thus vectorized over grid rows. We remark that both the base MacCormack algorithm and the multiple-grid scheme can be vectorized over the entire domain to obtain much longer vectors. While this would be desirable for a Cyber 205, it does not seem to be worth the added programming complexity for the Cray. Note also that making the innermost loop the longest lowers the paging rate on the virtual storage IBM machine.
- 3) Removing IF statements and subroutine calls from vectorizable loops.
- 4) Storing metric invariants used in a second-order damping term for the supercritical cases. This would be impractical in a storage-limited three-dimensional code.
- 5) Addition of Cray compiler directives to ignore possible recursion due to ambiguous subscripts in the multiple-grid subroutine. To allow for several grid levels this subroutine is programmed to sweep the grid with a variable stride equal to $2^{**}(\text{IGD}-1)$, where IGD is the grid level. The Cray compiler does not know the value of this stride a priori, and, to avoid possible recursion, will not vectorize loops with an ambiguous stride. We know that the multiple-grid scheme is not recursive

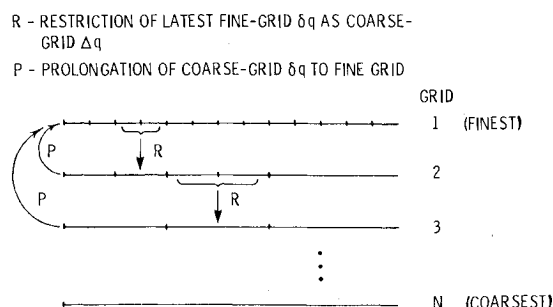


Fig. 3 Multiple-grid information flow.

and direct the Cray compiler to ignore the possible vector dependencies. Cray compiler directives start with a "C" in column one and look like comments to the IBM compiler.

The multiple-grid scheme, as presently constructed, requires the use of grids of length $n(2^p)+1$ for n and p integers as described earlier. Since boundary conditions are computed separately from interior points, fine-grid vectors are of length $n(2^p)-1$ and coarse-grid vectors are of length $n(2^{p-k+1})-1$, where $k \geq 2$ is the grid number. For the cases presented later, $n=4$ and $p=4$ (in the x direction) giving a fine-grid vector length of 63, which is a near-optimal vector length for the Cray. Coarse-grid vector lengths can be much shorter, and, consequently, coarse-grid calculations do not vectorize as efficiently as fine-grid calculations. Nevertheless, the coarse-grid calculations benefit from vectorization as long as the vector length remains greater than about four.

Note that a Cray vector is defined by a starting location, a length, and a stride through memory, making the multiple-grid scheme readily vectorizable on the Cray. However, on the Cyber 205, vectors are defined only by a starting location and length, with elements assumed to be in contiguous memory locations. This would make the multiple-grid scheme more difficult to vectorize on a 205.

The vectorized multiple-grid code is completely machine independent and has been run on both the IBM 370/3033 and the Cray 1-S, in 64-bit precision on each, to produce the results in this paper. To illustrate the performance increase due to vectorization, we present the following timing results for a typical inviscid supersonic flow case. First, the vectorized code runs 1.77 times faster on the IBM than the original code did, due strictly to scalar efficiency improvements. Thus many of the changes made in the name of "vectorization" may also be thought of as "scalar optimization," and in this sense vectorization may reduce the amount of work involved in executing a given algorithm. Second, in a scalar mode on the Cray the code runs 5.82 times faster on one grid and 5.31 times faster on three grids than on the IBM, reflecting mostly the clock times of the two computers (12.5 ns for the Cray, 60 ns for the IBM). Third, in vector mode on the Cray the code runs 2.36 times faster on one grid and 2.15 times faster on three grids than in scalar mode. Averaged over several cases we find that the vectorized Cray code runs 13.3 times faster on one grid and 11.2 times faster on three grids than the scalar IBM code. Again, the relative difference in speedup due to vectorization between the single- and multiple-grid codes is due primarily to the short vector lengths on the coarse grids. It may be possible to improve the vectorization of the multiple-grid algorithm by restructuring the storage of the coarse-grid data.

Computational Results

We report on a sampling of the computations performed thus far with the vectorized, multiple-grid MacCormack algorithm. The full Euler equations have been solved across the entire spectrum of subsonic, transonic, and supersonic flow. The thin-layer version of the Navier-Stokes equations has been solved for both attached and separated laminar subsonic flows. All computations have been performed in two dimensions. Extension of these results to the full Navier-Stokes equations, turbulent flow or three dimensions presents no essential difficulties. Results for attached and separated turbulent subsonic flows are reported in Ref. 10.

For each case presented here we define a multiple-grid work reduction factor to be the ratio of the work required to produce a converged solution on a single fine grid to the work required to produce the same result on a sequence of grids.

Inviscid Flow

Ultimately we intend to use the multiple-grid MacCormack algorithm to compute flows in turbomachinery cascades. To investigate the robustness of the algorithm we have computed the flow about the cascade of bicircular arc airfoils shown in Fig. 4 over a wide range of flow conditions, from low speed to choked. We will reference the different flow conditions by the nominal Mach number implied by the imposed isentropic static-to-total pressure ratio.

All inviscid computations were made using the 65×17 node fine grid, as shown in Fig. 5. All computations have been run to convergence on a single grid and on a three-grid sequence on both the IBM 370/3033 and the Cray 1-S. We emphasize that the converged single- and multiple-grid solutions are identical. The computations are considered to be converged

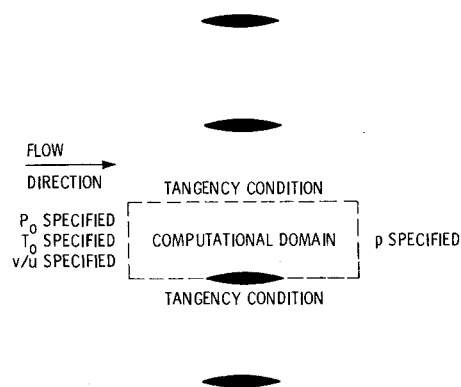


Fig. 4 Inviscid bicircular arc cascade problem.

Table 1 Timing and convergence data for inviscid flow cases

Case	Fig.	Machine	Time for 5K cycle, s		Cycle to converge		Time to converge, s	
			T1 grid	T3 grids	N1 grid	N3 grids	t1 grid	t3 grids
$M=0.2$ Low speed	7	IBM	368.9	551.5	5740	780	420.2	86.0
		Cray	28.0	51.2	5740	780	32.2	8.0
$M=0.5$ Subcritical	—	IBM	370.7	572.4	4300	710	318.8	81.3
		Cray	28.0	51.2	4300	710	24.1	7.3
$M=0.675$ Supercritical	8	IBM	481.6	665.2	2310	830	222.5	110.4
		Cray	35.1	58.3	2310	830	16.2	9.7
$M=0.73$ Choked	9	IBM	478.7	658.3	1350	710	129.2	93.5
		Cray	35.1	58.3	1350	710	9.5	8.3
$M=1.6$ Inlet diffuser	10	IBM	467.0	654.7	1840	950	171.9	124.3
		Cray	36.1	58.1	1840	950	12.8	11.0
$M=0.1-0.5$ Inlet shear	11	IBM	366.9	555.1	3030	1410	222.3	156.5
		Cray	28.0	51.2	3030	1410	17.0	14.5

when the average unscaled residual in pu drops below 10^{-4} , a decrease of approximately three decades. Detailed timing and convergence data are presented in Tables 1 and 2 and will be discussed subsequently.

Figure 6 shows computed points on a mass flow vs pressure ratio operating curve for the cascade. The largest error in integrated mass flow along any grid line in any of the cases was 0.38%. The computed points are compared to the one-dimensional isentropic theory. The one-dimensional choking pressure ratio is 0.73503 while the computed two-dimensional choking pressure ratio has been bracketed between 0.70155 and 0.72093. Details of the flows at three points on the curve are shown in Figs. 7-9.

Low-speed results at $M=0.2$ are shown in Fig. 7. Figure 7a shows isomachs while Fig. 7b shows Mach number distributions on the body and symmetry lines. Figures 7c and 7d show convergence histories on one and three grids, respectively. The low-speed solution converges extremely slowly on a single grid, taking 5740 time cycles. On three grids the solution converges in 780 time cycles, with a net multiple-grid work reduction factor of 4.92 on the IBM and 4.02 on the Cray.

Supercritical results at $M=0.675$ are shown in Fig. 8. This case has been well documented in Ref. 7. The single-grid solution converges in 2310 time cycles while on three grids the solution converges in 830 time cycles. The multiple-grid work reduction factor is 2.01 on the IBM and 1.67 on the Cray.

Choked results at $M=0.73$ are shown in Fig. 9. Of particular interest is the greatly increased single-grid convergence rate over that for the supercritical but unchoked case shown in Fig. 8. The choked single-grid solution converges in 1350 time cycles, which is almost twice as fast as the unchoked solution. On three grids the choked solution converges in 710 time cycles but, because the single-grid convergence rate is high, the multiple-grid work reduction factor is only 1.38 on the IBM and 1.14 on the Cray.

By fixing all inlet conditions at $M=1.6$ (below the choking mass flow) and by extrapolating all exit conditions, the same cascade was run as a supersonic inlet diffuser. These results are shown in Fig. 10. Isomachs, shown in Fig. 10a, show a strong oblique shock leaving the leading edge. The leading-edge shock intersects the upper symmetry plane as a normal shock that reflects downward to intersect with another oblique shock leaving the trailing edge. Surface Mach numbers, shown in Fig. 10b show some overshoot before the normal shock on the symmetry plane, indicating too little damping. The single-grid convergence history in Fig. 10c is not as steep as the choked case in Fig. 9c, converging here in 1840 time cycles. The three-grid history in Fig. 10d shows

convergence in 950 time cycles. The multiple-grid work reduction factor is 1.38 on the IBM and 1.21 on the Cray.

Subcritical flow through a similar cascade but with a 40% blade thickness and subject to a linear inlet shear profile is shown in Fig. 11. The inlet profile has a nominal Mach number of 0.5 at the top and 0.1 at the bottom. Isomachs in Fig. 11a show near-perfect left-to-right symmetry. Surface Mach numbers are shown in Fig. 11b. The single-grid convergence rate in Fig. 11c is slow, as are the other subcritical cases, converging in 3030 time cycles. On three grids the solution converges in 1410 time cycles. The multiple-grid work reduction factor is 1.42 on the IBM and 1.18 on the Cray.

Detailed timing and convergence data for the six inviscid cases are presented in Table 1. Note that times to convergence on the Cray ranged from 7.3 to 14.5 s. Multiple-grid work reduction factors are summarized in Table 2. The most in-

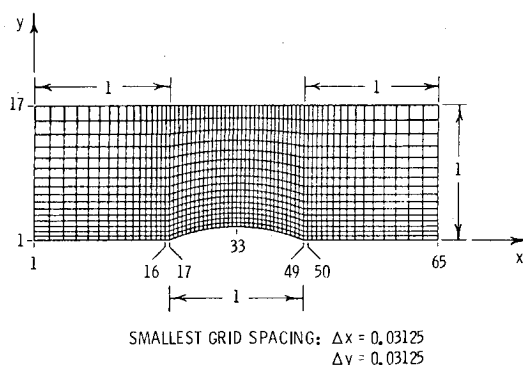


Fig. 5 Inviscid flow fine grid.

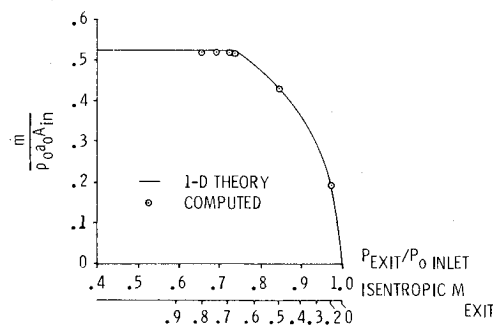
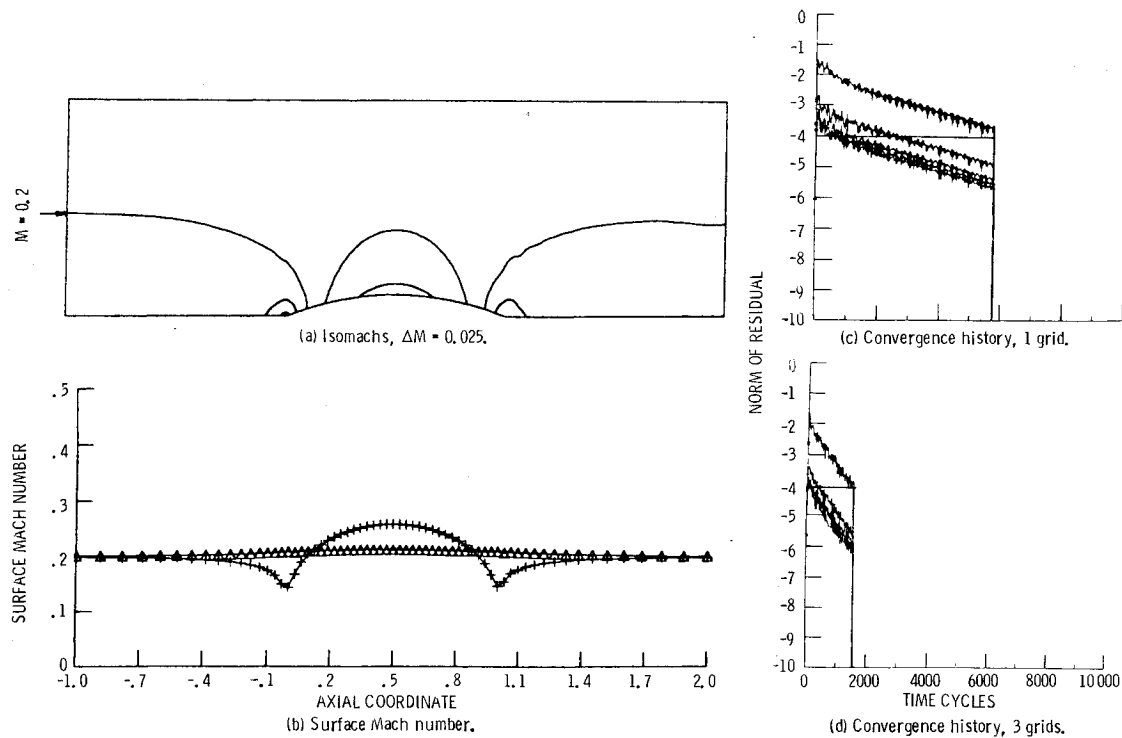
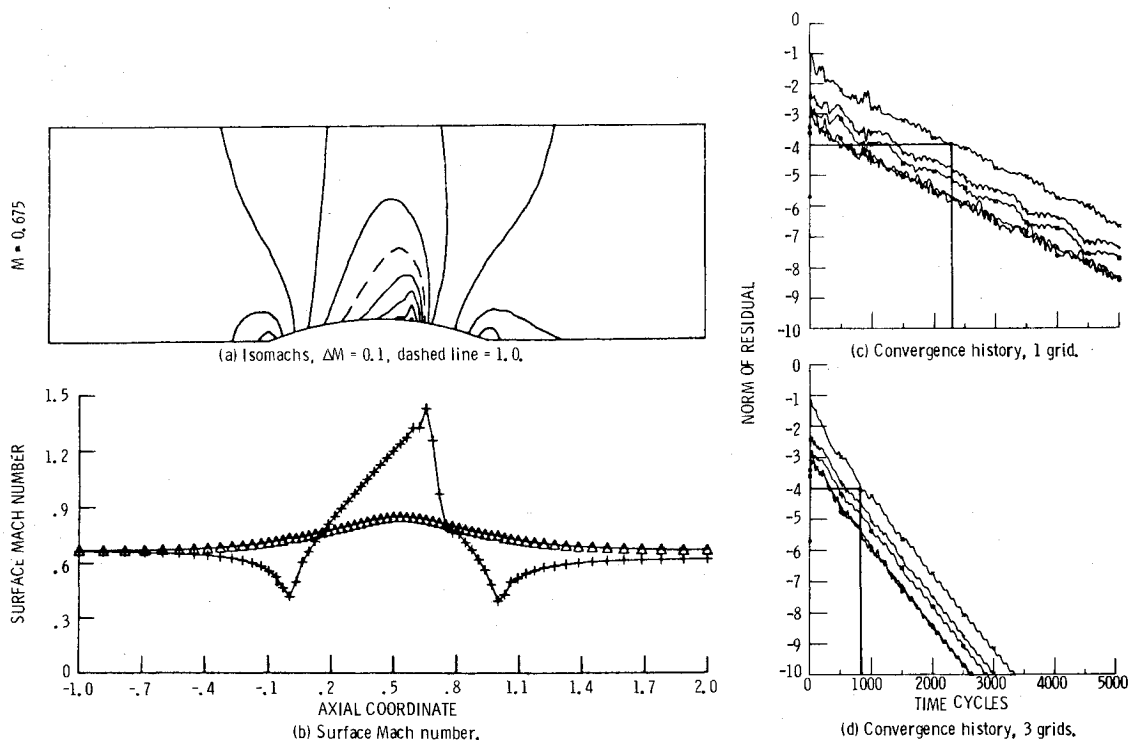


Fig. 6 Operating curve for inviscid 20% thick bicircular arc cascade.

Table 2 Multiple-grid work reduction factors for inviscid flow cases

Case	Fig.	Machine	(Added work) ⁻¹ × T 1 grid/T 3 grids	Convergence rate incl. N 1 grid/N 3 grids	Work reduction factor T 1 grid/T 3 grids
$M=0.2$	7	IBM	0.669	7.36	4.92
Low speed		Cray	0.547	7.36	4.02
$M=0.5$	—	IBM	0.648	6.06	3.92
Subcritical		Cray	0.547	6.06	3.31
$M=0.675$	8	IBM	0.724	2.78	2.01
Supercritical		Cray	0.602	2.78	1.67
$M=0.73$	9	IBM	0.727	1.90	1.38
Choked		Cray	0.602	1.90	1.14
$M=1.6$	10	IBM	0.713	1.94	1.38
Inlet diffuser		Cray	0.621	1.94	1.21
$M=0.1-0.5$	11	IBM	0.661	2.15	1.42
Inlet shear		Cray	0.647	2.15	1.18

Fig. 7 Inviscid bicircular arc cascade, low speed $M = 0.2$.Fig. 8 Inviscid bicircular arc cascade, supercritical, $M = 0.675$.

interesting result in this table is that the multiple-grid work reduction factor is greatest for the lowest-speed flows. However, in all cases the increased computational work of the multiple-grid scheme was more than offset by the improved convergence rate.

Viscous Flow

We first consider the subsonic flow through a cascade of unstaggered flat plates at zero angle of attack. The ratio of exit static pressure to upstream total pressure is 0.8430191,

yielding flow Mach numbers in the vicinity of 0.5 for the test case exhibited here. Computations have been made for three Reynolds numbers in the range 8.4×10^3 to 2.0×10^5 , based on the critical speed and cascade gap. Symmetry is invoked to limit the size of the computational domain and the flow is assumed to be laminar. These assumptions are made for convenience in specifying the number and location of the fine-grid nodal points and do not imply limitations on the generality of the method.

Three different fine grids were employed in this study. All have the same number of nodal points (65×33) and have their

transverse grid lines located at the same positions. They differ in the positioning of their lateral grid lines. These are smoothly stretched away from the solid boundary in a geometric progression, starting from initial spacings ranging from $\Delta y_i = 0.0125$ to 0.0025 . These fine grids each allow the construction of four successively coarser grids that may be used to form multiple-grid sequences of length one through five.

Results for a Reynolds number of 8.4×10^3 and an initial grid spacing of 0.0125 are shown in Fig. 12. Isomach contours for the converged solution are shown. The contour levels displayed are not equally spaced and are the same for all five

cases shown. Nevertheless, they provide a good qualitative indication of the nature of the computed flowfields. More quantitative information is provided by the normalized u -velocity profiles, which are also illustrated. The u velocity, normalized with its value at the top boundary and same streamwise station, is plotted as a function of relative distance from the bottom boundary. Curves for every second streamwise station, starting with the plate leading edge and ending at the outflow boundary, are displayed. They are staggered in proportion to the spacing of their respective streamwise stations. The data displayed in Fig. 12 are for conditions of optimal work reduction, which, in this case, was

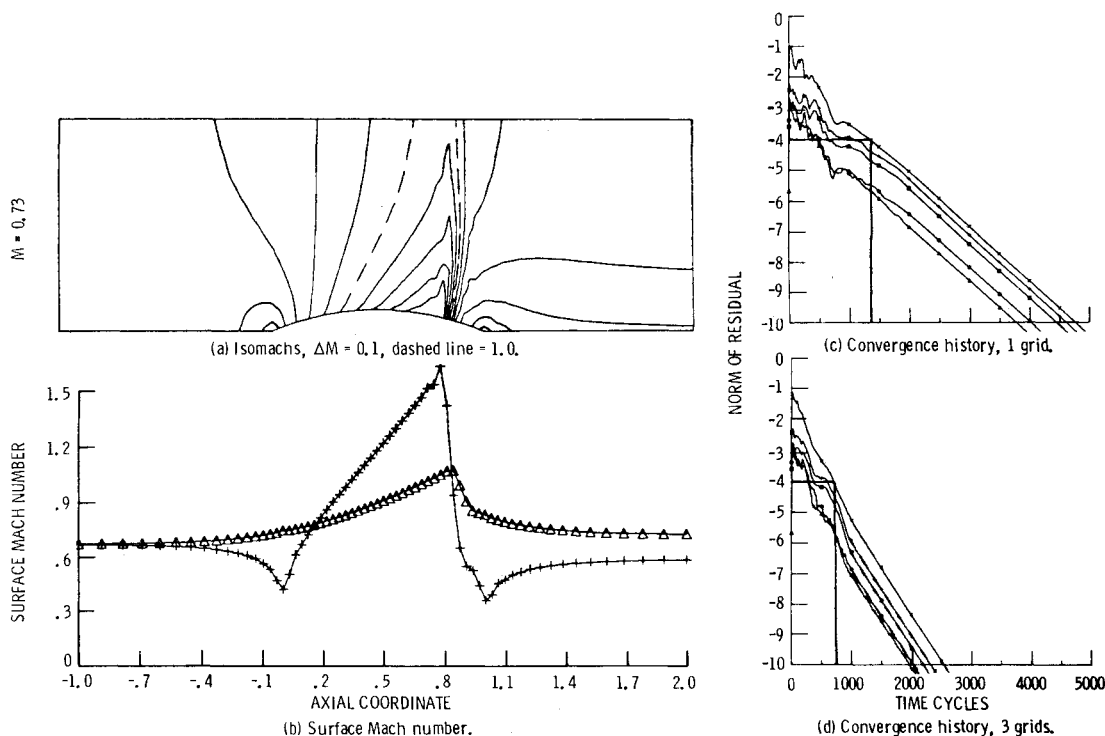


Fig. 9 Inviscid bicircular arc cascade, choked, $M = 0.73$.

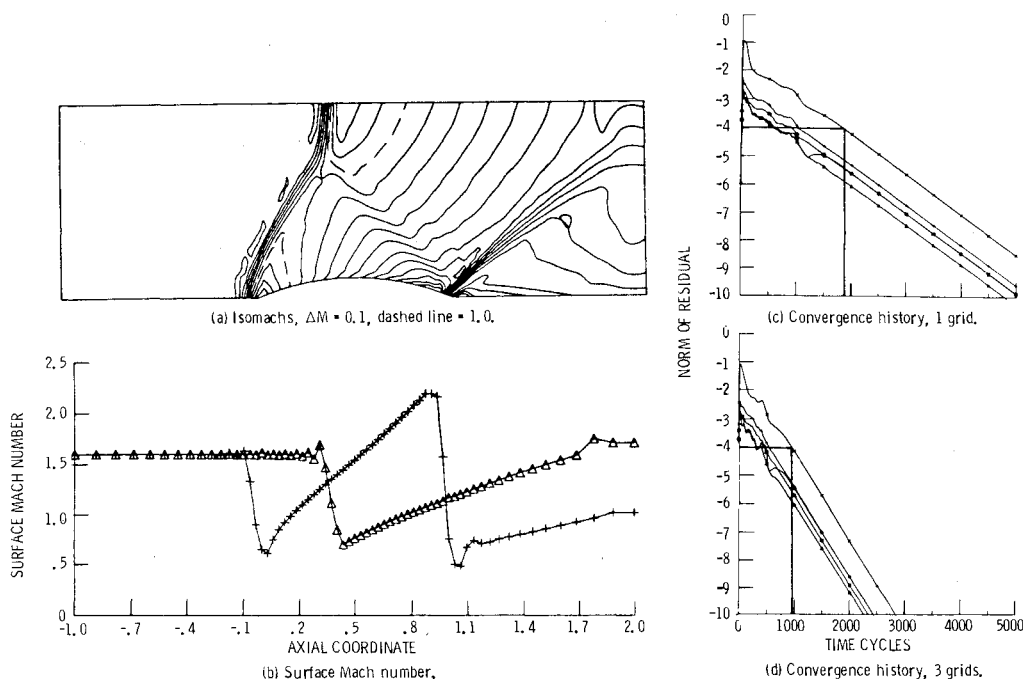


Fig. 10 Inviscid supersonic diffuser, $M_{in} = 1.6$.

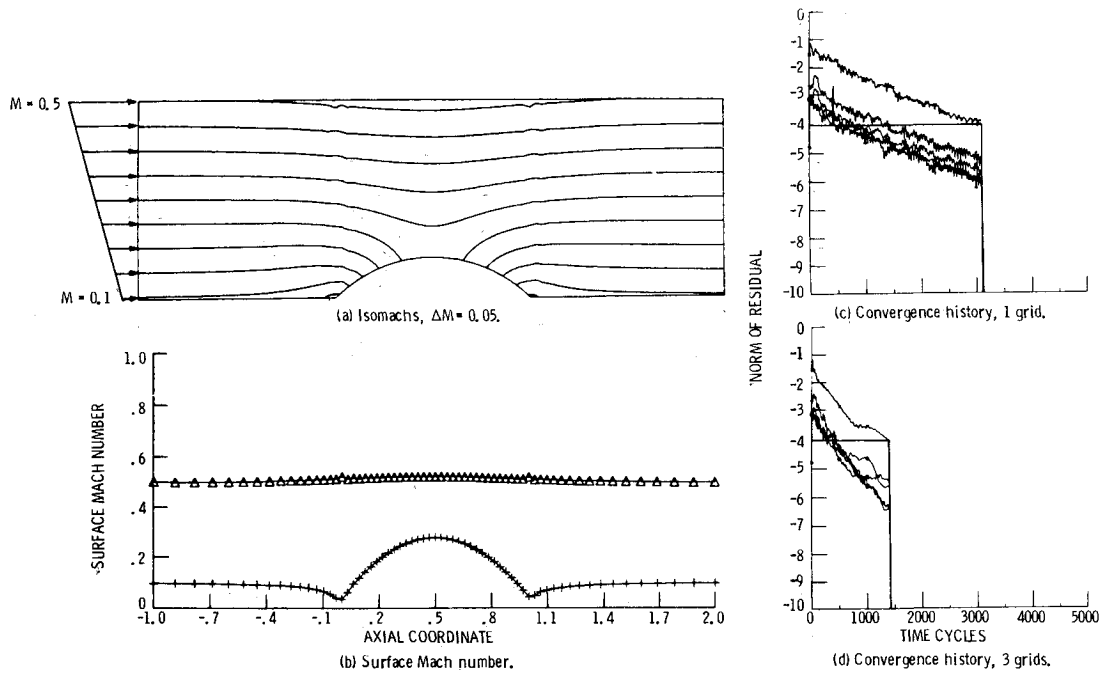


Fig. 11 Inviscid bicircular arc cascade with linear inlet shear.

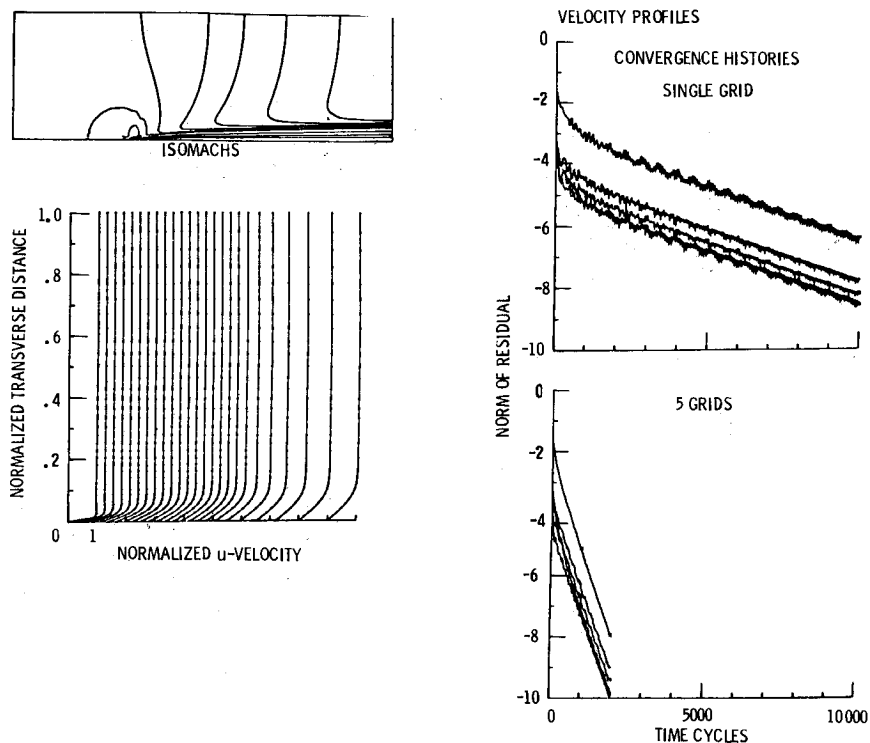


Fig. 12 Viscous flat plate cascade results, $Re = 8.4 \times 10^3$.

a five-grid sequence. However, the solution obtained is not a function of grid sequence length. This was verified by extensive computational experimentation.

Convergence histories are also shown in Fig. 12 where we display the fine-grid convergence history and the corresponding convergence history for that grid sequence length which produced the best work reduction factor.

For each of three Reynolds numbers, three initial grid spacings, and the five possible grid sequence lengths, the computational work required to reduce a standard error measure to a specified tolerance has been computed. In each

case, based on these data, we have estimated a multiple-grid work reduction factor for the corresponding optimal grid-sequence length. Multiple-grid work reduction factors ranging from 1.8 to 4.8 have been realized. We note that Johnson⁸ observed that although the work reduction factor and, possibly, the optimal grid sequence length decrease with increasing grid stretching, they do not appear to decrease with increasing Reynolds number.

Further viscous flow computations have been performed in a cascade where the flat plates were replaced by sting-mounted bicircular-arc airfoils. Here the grids used were

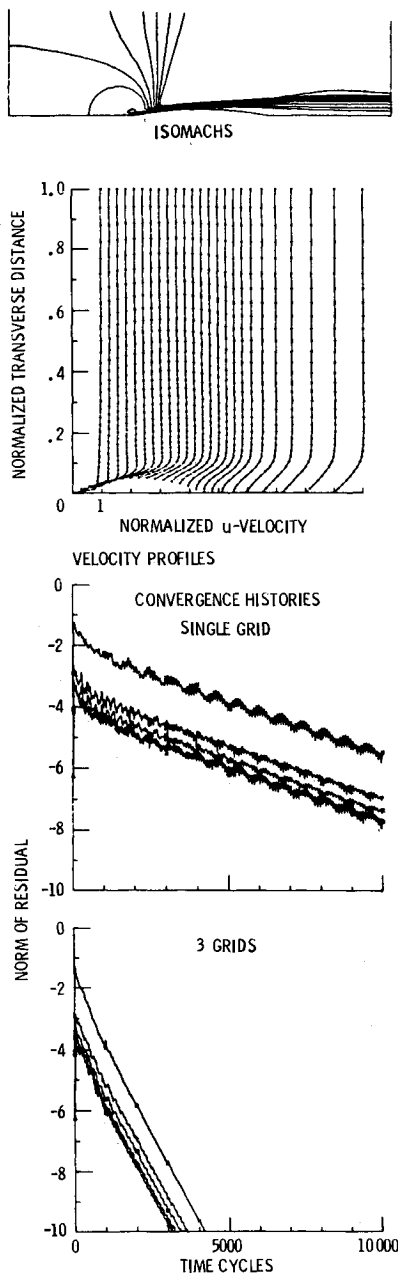


Fig. 13 Viscous bicircular arc cascade results, $Re = 8.4 \times 10^3$, $t/c = 0.1$.

simply sheared versions of those used previously in the flat plate test cases. Four combinations of Reynolds number, thickness-to-chord ratio, and grid configuration were tested. Results are shown in Fig. 13 for a 10% thick airfoil with an initial grid spacing of 0.0125 and a Reynolds number of 8.4×10^3 . The inlet Mach number is approximately 0.5. Isomach contours, normalized u -velocity profiles, and convergence histories are shown. An optimal work reduction factor of 3.8 was obtained for this case with a three-grid sequence. Work reduction factors ranged from this value to 1.5 on a finer grid.

The viscous results presented in this report were run as scalar computations on the IBM 370/3033. A vectorized version of the viscous code has been benchmarked on the Cray 1-S to determine the work reduction attainable through vectorization for viscous cases. By combining the multiple-grid work reduction factor with the work reduction attained

by vectorizing the algorithm, we have determined that the overall work reduction for the viscous flow cases ranged as high as 76.8.

Several issues bearing on the multiple-grid work reduction factor should be mentioned at this juncture. When one considers convergence acceleration of the turbulent full Navier-Stokes equations, greater work reduction than obtained here will result by virtue of the inclusion of the full viscous terms and turbulence modeling on the fine grid. The treatment of the Jacobian matrices used in the coarse-grid acceleration scheme has a large influence on the efficiency of the coarse-grid computations and, hence, on the work reduction factor. Substantial improvement appears to be possible over the current treatment of these Jacobians. In the present computations, injection is used as the restriction operator and linear interpolation is chosen as the prolongation operator. These choices may not be optimal for use on highly stretched grids. Better choices could increase both the optimal grid sequence length and the work reduction factor. Similar consequences might result from an alternative coarse-grid formation strategy.

Given the encouraging results obtained to date, more comprehensive testing and more sophisticated applications of the inviscid and viscous flow convergence acceleration and vectorization ideas presented here are planned.

Conclusions

A vectorized, machine-independent, explicit multiple-grid algorithm for the efficient solution of the steady Euler and Navier-Stokes equations has been presented. The coarse-grid scheme used to accelerate convergence is compatible with all one- and two-step Lax-Wendroff algorithms. Here, it has been used in conjunction with MacCormack's method.

The convective version of the coarse-grid scheme may be used with any set of flow equations in the hierarchy ranging from the Euler equations to the full Navier-Stokes equations. We have used it herein with the thin-layer version of the Navier-Stokes equations.

Computational results have been presented for subsonic, transonic, and supersonic inviscid flows and for laminar, subsonic viscous flows. Vectorization of the algorithm proved relatively straightforward and, in conjunction with multiple-gridding, yielded overall work reduction factors ranging from 15.3 to 76.8, over a fairly broad range of flow conditions.

References

- ¹Viviani, H., "Formes Conservatives des Equations de la Dynamique des Gaz" ("Conservative Forms of Gas Dynamics Equations"), *La Recherche Aeronautique*, Jan.-Feb. 1974, pp. 65-66.
- ²Steger, J. L., "Implicit Finite Difference Simulation of Flow About Arbitrary Geometries with Application to Airfoils," AIAA Paper 77-665, June 1977.
- ³Baldwin, B. S. and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257, Jan. 1978.
- ⁴MacCormack, R. W., "The Effect of Viscosity in Hypervelocity Impact Cratering," AIAA Paper 69-354, April 1969.
- ⁵Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," AIAA Paper 81-1025, June 1981.
- ⁶Richtmyer, R. D., "A Survey of Difference Methods for Non-Steady Fluid Dynamics," National Center for Atmospheric Research, Boulder, Colorado, NCAR-TN-63-2, Aug. 1963.
- ⁷Johnson, G. M., "Multiple-Grid Acceleration of Lax-Wendroff Algorithms," NASA TM-82843, 1982.
- ⁸Johnson, G. M., "Convergence Acceleration of Viscous Flow Computations," NASA TM-83039, 1982.
- ⁹Shang, J. S., Buning, P. G., Hankey, W. L., and Wirth, M. C., "The Performance of a Vectorized 3-D Navier-Stokes Code on the Cray-1 Computer," AIAA Paper 79-1448, July 1979.
- ¹⁰Johnson, G. M., "Multiple-Grid Convergence Acceleration of Viscous and Inviscid Flow Computations," NASA TM-83361, 1983.