AIAA JOURNAL
Vol. 30, No. 2, February 1992

Local Solution Acceleration Method for the Euler and Navier-Stokes Equations

D. Drikakis* and S. Tsangaris†
Laboratory of Aerodynamics NTUA, Athens, Greece

The solution of the compressible Euler and Navier-Stokes equations via an upwind finite volume scheme is obtained. For the inviscid fluxes, the monotone upstream-centered scheme for conservation laws (MUSCL) has been incorporated into a Riemann solver. The MUSCL scheme is used for the unfactored implicit equations that are solved by a Newton form, and relaxation is performed via Gauss-Seidel relaxation technique. The solution on the fine grid is obtained by iterating first on a sequence of coarser grids and then interpolating the solution up to the next refined grid. Since the distribution of the numerical error is nonuniform, the local solution of the equations can be obtained in regions where the numerical errors are large. The construction of the partial meshes, in which the iterations will be continued, is determined by an adaptive procedure taking into account some convergence criteria. Reduction of the computational work units for two-dimensional problems is obtained via the local adaptive mesh solution which is expected to be more effective in three-dimensional complex flow computations.

Introduction

DURING the last decade, a wide variety of upwind numerical methods for the solution of the Euler and the Navier-Stokes equations has been developed. In the present paper, the monotone upstream-centered scheme for conservation laws[1] (MUSCL) has been incorporated into a Riemann Solver[2] for the calculation of the inviscid fluxes. The calculation of the derivatives in the diffusive fluxes is obtained by central differences. For the cross derivatives a special upwind treatment[3] is used.

Many implicit methods use the unfactored Newton form of the equations and find the solution to the steady state via a Gauss-Seidel relaxation technique,[3,4] increasing the efficiency of the solver. The present method is an implicit unfactored relaxation scheme which allows high Courant-Friedrichs-Lewy (CFL) numbers.

In spite of using these techniques, the cost of solving both the Navier-Stokes and the Euler equations is high. Thus, the calculation of complicated flows may be impractical in a design environment. On the other hand, the flows are nonuniform. Nonuniformities were especially presented near the boundaries, as well as in the shock-wave regions. Although a large part of the flowfield solution had converged during the iterations, more time steps are needed for "strong" regions where the convergence criteria have not been satisfied. Thus, the solution of the equations need not be obtained in the whole flowfield during the iterations, but only in the regions where the numerical disturbances are large. An analytical study of the propagation of numerical disturbances has been presented by Panaras[5] using a thin-layer Navier-Stokes code.

In the present paper, the adaptation of a local solution method and the combination of this method with the mesh sequencing procedure are proposed as an acceleration strategy for the full Navier-Stokes and Euler equations. At first, the equations are iterated on a sequence of coarse grids and then the solution is interpolated to the next finer grid. After smoothing the solution on the finest grid and by using prescribed criteria, partial meshes are selected, as subregions of the fine mesh, and local solution is obtained. Partial meshes contain a small enough number of grid points in comparison with the whole fine mesh; therefore, the computational cost for these meshes is very small. The adaptation of the partial meshes is obtained using convergence criteria.

The local solution improves the mesh sequencing technique, increasing the efficiency of the solver. The aforementioned method is expected to be more effective for large-scale computations (e.g., three-dimensional flows).

Governing Equations and Time Integration

The governing equations are the time-dependent full Navier-Stokes equations for a compressible fluid. These equations can be written in conservation form and for a generalized coordinate system as

$$U_t + (E_{\text{inv}})_\xi + (G_{\text{inv}})_\zeta = (E_{\text{vis}})_\xi + (G_{\text{vis}})_\zeta \tag{1a}$$

$$U = J(\rho, \rho u, \rho w, e)^T \tag{1b}$$

E_{inv}, G_{inv} are the inviscid flux vectors while E_{vis}, G_{vis} are the viscous flux vectors. After nondimensionalization of the foregoing equations, Mach and Reynolds numbers appear as parameters. Body-fitted arbitrary coordinates ξ, ζ are used. The variable $J = 1/(\xi_x \zeta_z - \zeta_x \xi_z)$ is the Jacobian of the transformation $\xi = \xi(x, z)$ and $\zeta = \zeta(x, z)$ from the Cartesian to the generalized coordinates.

The implicit scheme in the present work is first-order accurate in time. The implicit form of Eq. (1) is written as

$$(U^{n+1} - U^n)/\Delta t + E_\xi^{n+1} + G_\zeta^{n+1} = 0 \tag{2a}$$

$$E = E_{\text{vis}} + E_{\text{inv}}, \qquad G = G_{\text{vis}} + G_{\text{inv}} \tag{2b}$$

A Newton method can be constructed for U^{n+1} by linearizing the fluxes in Eq. (1) about the known time level n:

$$E^{n+1} = E^n + A^n \Delta U, \qquad G^{n+1} = G^n + C^n \Delta U$$

The last relations yield the Eq. (1) to the following form:

$$\frac{\Delta U}{\Delta t} + (A^n \Delta U)_\xi + (C^n \Delta U)_\zeta = -(E_\xi^n + G_\zeta^n) = \text{RHS} \tag{3}$$

Received June 28, 1990; presented as Paper 6.10.1 at the 17th ICAS Congress, Stockholm, Sweden, Sept. 9–14, 1990; revision received April 19, 1991; accepted for publication April 22, 1991.

*Research Assistant, Department of Mechanical Engineering, P.O. Box 64070, 15710 Zografou. Member AIAA.
†Assistant Professor, Department of Mechanical Engineering, P.O. Box 64070, 15710 Zografou.

A, C are the Jacobians of the flux vectors E, G while $\Delta U = U^{n+1} - U^n$ is the time variation of the solution. The terms $(A^n\Delta U)_\xi$, $(C^n\Delta U)_\zeta$ are discretized at the computational volume up to second order of accuracy in space. For example, the term $(A^n\Delta U)_\xi$ is written as

$$(A^n\Delta U)_\xi = (A^n_{\text{inv}}\Delta U)_{i+(1/2)} - (A^n_{\text{inv}}\Delta U)_{i-(1/2)}$$
$$+ (A^n_{\text{vis}})_i(\Delta U_{i+1} - 2\Delta U_i + \Delta U_{i-1})$$

The subscript "inv" denotes the Jacobian of the inviscid fluxes. The thin-layer viscous Jacobian A_{vis} is used for all directions ξ, ζ instead of the exact form, saving computational time.[6] Numerical experiments have shown that this implementation does not change the good stability behavior of the solution method or the acceleration techniques. On the right-hand side (RHS) the full form of the Navier-Stokes equations is used.

The solution of the system of Eq. (3) is obtained by the sequence of approximations denoted by q^ν such that

$$\lim_{\nu>1} q^\nu \to U^{n+1}$$

where ν is the subiteration state. The equations are solved by a Newton method. A Newton form is obtained by the linearization of the Eq. (3), around the known subiteration state ν, as follows:

$$\frac{\Delta q^{\nu+1}}{\Delta t} + (A^\nu\Delta q^{\nu+1})_\xi + (C^\nu\Delta q^{\nu+1})_\zeta = \frac{U^n - q^\nu}{\Delta t} - (E^\nu_\xi + G^\nu_\zeta) \tag{4}$$

q^ν, $q^{\nu+1}$ are the solution vectors at the subiteration states ν, $\nu + 1$, respectively. The solution at the level $\nu + 1$ is updated as

$$q^{\nu+1} = q^\nu + \Delta q^{\nu+1}$$

Equation (4) is solved by a Gauss-Seidel type of relaxation scheme[4] using μ-subiterations in the relaxation procedure, with the right-hand side (RHS) held constant. The optimum number for the Newton subiterations is $\nu = 2$ for both the Navier-Stokes and Euler equations. For the Gauss-Seidel relaxation, typical values are $\mu = 6$ for the Navier-Stokes and $\mu = 4$ for the Euler equations, respectively. In the present acceleration strategies, fewer subiterations, both in the Newton and the Gauss-Seidel methods, can be obtained. Thus, $\mu = 3$, $\nu = 2$ subiterations for the Navier-Stokes and $\mu = 2$, $\nu = 1$ for the Euler equations are used on the coarser meshes of the mesh sequencing procedure and during the local solution of the equations on the partial meshes. Thus, a significant saving of computational time is obtained. The present implicit schemes uses CFL numbers up to 150 for transonic and supersonic flows.[4,6]

Inviscid Fluxes

For the inviscid fluxes on the right-hand side of Eq. (4), a linear, locally one-dimensional Riemann solver[2] (Godunov-type differencing) is employed at the finite volume cell faces. The state that is used for the Riemann solution depends on the sign and the size of the eigenvalues.[2] The Riemann solver construct the fluxes as a function of the conservative variables $(\rho^j, (\rho u)^j, (\rho w)^j, e^j)$ on each characteristic direction $j = 0, 1, 2$ (the characteristic directions are defined from the three eigenvalues). For the calculation of these characteristic conservative variables, a mean value is obtained at the cell faces as follows:

$$U^j_{i+1/2} = {}^1/_2[(1 + \xi_j)U_l + (1 - \xi_j)U_r], \qquad j = 0, 1, 2$$

where the term ξ_j is associated with the eigenvalues $\lambda^j_{l,r}$ ($j = 0, 1, 2$) of the left (l) and right states (r):

$$\xi_j = \frac{\lambda^j_l + \lambda^j_r}{|\lambda^j_l| + |\lambda^j_r|}$$

The left (l) and the right (r) conserved state variables are obtained by an upwind-biased interpolation[1,6] known as the MUSCL-type approach.

A general MUSCL-type scheme can be defined as

$$(U_l)_{i+(1/2)} = U_i + (S/4)[(1 - KS)\nabla_\xi + (1 + KS)\Delta_\xi]U_i$$

$$(U_r)_{i+(1/2)} = U_{i+1} - (S/4)[(1 - KS)\Delta_\xi + (1 + KS)\nabla_\xi]U_{i+1}$$

The symbols Δ and ∇ denote the forward and the backward difference operators:

$$\Delta_\xi U_i = U_{i+1} - U_i, \qquad \nabla_\xi U_i = U_i - U_{i-1}$$

S is the van Albada-type sensor for the detection of shocks and other discontinuities:

$$S = \frac{2\Delta_\xi\nabla_\xi}{(\Delta_\xi)^2 + (\nabla_\xi)^2 + \epsilon}$$

ϵ is a small number to prevent division by zero. The parameter K defines the spatial accuracy of the approximation. $K = -1$ corresponds to the fully upwind second-order scheme and $K = {}^1/_3$ to the biased scheme which is formally second-order accurate for two-dimensional problems.

The preceding MUSCL-type interpolation was incorporated into the Riemann solver for the evaluation of the conservative variables at the cell faces.

Viscous Terms

In the case of the unfactored solution of the equations, a diagonal matrix inversion is required. Thus, the discretization must augment, as much as possible, the diagonal terms.

Chakravarthy[3] has presented that an upwind-type discretization of the cross derivatives in the viscous fluxes can augment the diagonal terms of the matrix inversion. In the present method, the Chakravarthy scheme has been adopted. The other derivatives are discretized by central differences. A similar implementation has also been proposed by Schmatz.[6]

Mesh Sequencing Technique

In the mesh sequencing, an initial guess on the fine mesh is obtained by first iterating on a sequence of coarser grids and then interpolating the solution to the next finer gird.[7]

The mesh sequencing procedure is a part of the present acceleration method. For the mesh sequencing, a generated fine grid as well as the number (n) of the coarse meshes (coarse levels) must be introduced as data in the computational code, while only the fine mesh is necessary to be stored. When the fine mesh with size h is given, two different procedures can be followed for the construction of the coarse mesh with size H.

1) The coarse mesh is constructed by eliminating every second line of the fine mesh in each direction, and thus doubling the mesh spacing. A finite volume on the coarse mesh contains four cell volumes of the finer mesh. The construction of a typical coarse mesh is shown in Fig. 1. Correspondence between the points $(i, j)^{(n+1)}$ of the coarse-mesh $(n + 1)$ level and the points $(i, j)^{(n)}$ of the coarse-mesh n level must be defined as

$$i^{(n+1)} = [i^{(n)} + 1]/2, \qquad j^{(n+1)} = [j^{(n)} + 1]/2$$

in order to avoid coarse mesh points out of the computational domain. Otherwise fictitious computational volumes are needed.

The superscript n denotes the mesh levels ($n = 1$ corresponds to the fine mesh). The coordinates $x^{(n+1)}(i^{(n+1)}, j^{(n+1)})$, $z^{(n+1)}(i^{(n+1)}, j^{(n+1)})$ of a coarse level $(n + 1)$ are succesively determined from the previous levels up to the fine

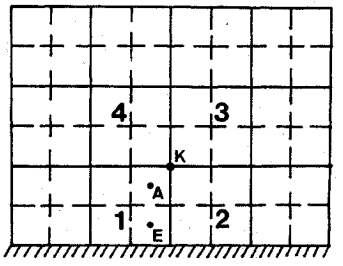

Fig. 1 Construction of a typical coarse mesh.

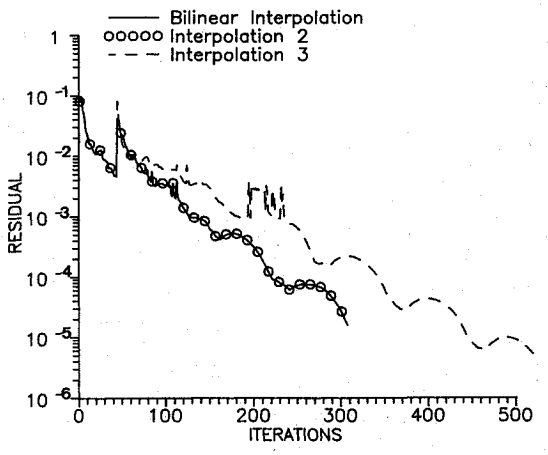

Fig. 2 Comparison between interpolation formulas.

mesh (e.g., the level 2 from the fine mesh 1, level 3 from the level 2, etc.):

$$x^{(n+1)}(i^{(n+1)}, j^{(n+1)}) = x^{n}(i^{(n)}, j^{(n)}) = \cdots = x^{(1)}(i^{(1)}, j^{(1)})$$

$$z^{(n+1)}(i^{(n+1)}, j^{(n+1)}) = z^{n}(i^{(n)}, j^{(n)}) = \cdots = z^{(1)}(i^{(1)}, j^{(1)})$$

The coarsest mesh is the initial mesh of the calculations.

2) Another way is to construct coarser meshes in terms of the cell volumes by eliminating every second volume of the finer mesh in each direction. This is known as the cell-centered approximation.[8] The coarser mesh volumes are a subset of the finer mesh volumes; therefore the interpolation of the conservative variables is unnecessary. The coarser grid points are not a subset of the corresponding points of the finer grid and thus interpolation and smoothing of the mesh lines (after the interpolation) are needed. Numerical experiments have shown that the cell-centered approximation may deteriorate the numerical convergence.

For the aforementioned reasons, the first procedure (four cells of the fine grid constitute one cell of the coarse grid) is implemented in the present study.

Coarsening Ratio

In the present study, coarse meshes are defined with a coarsening ratio $H/h = 2$, which is the smallest possible number and is already big enough to make computational work on the coarse mesh quite small relative to the fine mesh work. This coarsening ratio has also been used by other authors[8–10] in multigrid methods. Numerical experiments have shown that a coarsening ratio $H/h = 4$ may deteriorate the numerical convergence because of the large interpolation errors.

Refining Grid Criterion—Convergence Criterion

The mesh sequencing idea is the generation of a better initial guess for the solution of the equations, instead of the classical uniform freestream initial condition in the whole flowfield. Numerical experiments have shown that this initial guess is not necessary to constitute the final converged solution on the coarse mesh. Thus, the coarse meshes are used only for a number of iterations until the characteristic regions of the flowfield (shock wave, boundary layers) have started to appear, but without the absolute correct values of the steady-state solution.

The user cannot know the required number of iterations for the appearance of the flowfield regions, because this number depends on the flow case. On the other hand, we have found that the variation $\tilde{Q} = \max(|\Delta\rho|, |\Delta(\rho u)|, |\Delta(\rho w)|, |\Delta e|)$ of the dimensionless conservative variables is a proper criterion to check the appearance of flowfield regions during the convergence. For the present Riemann solver and for inviscid flowfield, the appearance is satisfactory when $\tilde{Q}$ is about 1×10^{-3}. In viscous flowfields the order of $\tilde{Q}$ must be about 5×10^{-4}. In our computational code, $\tilde{Q}$ is the refining grid criterion for each coarse level. For the present Riemann solver, the variations of the conservative variables $|\Delta\rho|$, $|\Delta(\rho u)|$, $|\Delta e|$ are of the same order of magnitude during the convergence. The variation of the conservative variable $|\Delta(\rho w)|$ is almost one order of magnitude lower than the other variations.

The measure of the convergence is also the variation $\tilde{Q}$. It is noted that we use the absolute number $\tilde{Q}$ instead of another criterion, e.g., the maximum value of the RHS in Eq. (3), because in the present solver these quantities converge with the same speed after about 30 iterations from the beginning of the calculations. It is also noted that we do not use the variation of the conservative solution vector U because this vector is divided by the Jacobian $J^{-1} = \xi_x\zeta_z - \zeta_x\xi_z$. Panaras[5] has shown that the large variations of the Jacobian within a grid may have significant masking effects on the numerical mean-value convergence criteria if the convergence criteria are scaled by J^{-1}. For this reason, the maximum variation $\tilde{Q}$ of the unscaled dimensionless conservative variables is used instead of the variation ΔU.

Interpolation Procedure

In the present mesh sequencing procedure, the center of the volumes of the fine mesh are not a subset of the volumes of the coarse mesh and consequently the values of the conservative variables on the fine mesh must be calculated by a weighted average of the values on the coarse mesh. A proper way to find the values on the volume A of the fine mesh (Fig. 1) from the values 1, 2, 3, 4 (volume centers) of the coarse mesh is the bilinear interpolation procedure (interpolation 1):

$$U_A = (1/16)(9U_1 + 3U_2 + 3U_4 + U_3) \tag{5}$$

This procedure has also been used by other authors[8–10] in the field of multigrid algorithms. A second formula (interpolation 2) which has been used during the development of the mesh sequencing takes into account the cell area of the volume defining an interpolating value on the point K (Fig. 1). The value on the volume A is determined by the average of the values of the points K and 1 as

$$U_A = \tfrac{1}{2}(U_1 + U_K) \qquad \text{with} \qquad U_K = \left(\sum_{i=1}^{4} S_i U_i\right) \Big/ \left(\sum_{i=1}^{4} S_i\right)$$

where $S_i (i = 1, \ldots, 4)$ are the cell areas of the coarse mesh. A similar formula (interpolation 3) can be obtained defining the point K as

$$U_K = \tfrac{1}{4}(U_1 + U_2 + U_3 + U_4)$$

and the point A as in the interpolation 2 formula. A typical comparison between the bilinear interpolation [Eq. (5)], interpolation 2, and interpolation 3 is shown in Fig. 2 for the inviscid transonic flow ($M = 0.95$, $a = 0$ deg) over a NACA 0012 airfoil. No difference exists in the convergence using the bilinear interpolation and interpolation 2, while deterioration of the convergence is presented using the interpolation 3. It has been found that the same behavior of the interpolations is presented independently of the mesh size and the flow test case. The jump of the residuals in the convergence histories is ap-

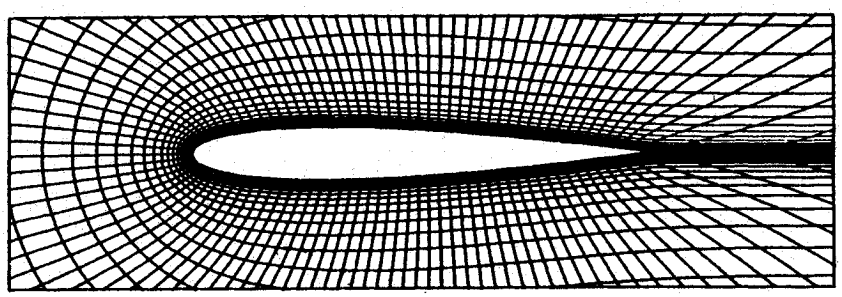

Fig. 3a Partial view of the fine mesh.

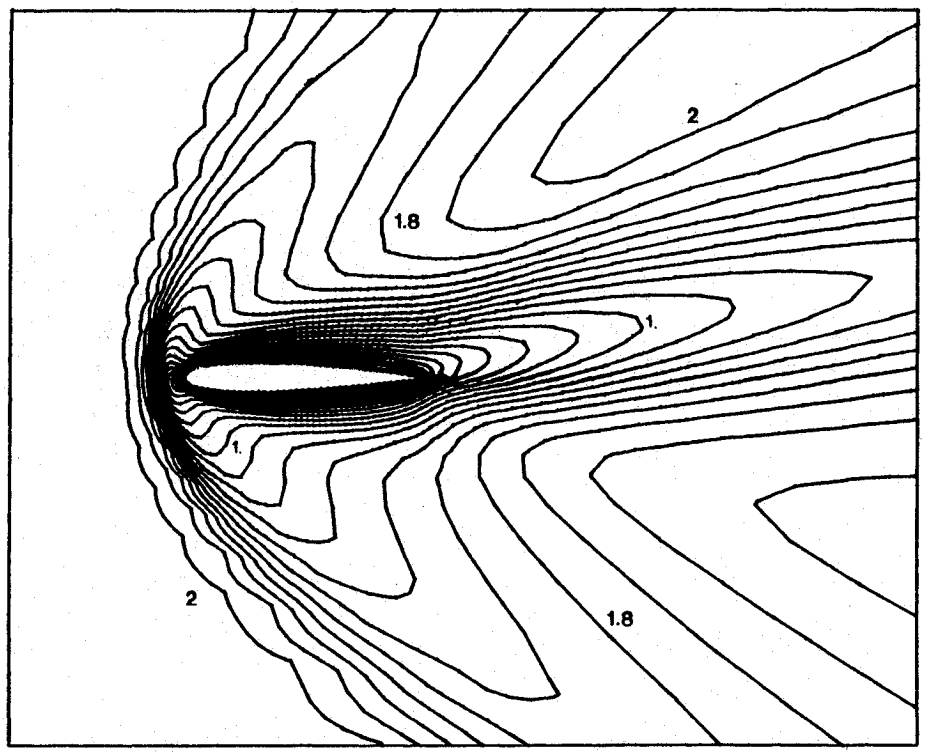

Fig. 3b Iso-Mach lines for the supersonic laminar viscous flow ($M_\infty = 2$, $\alpha = 10$ deg, $Re = 106$).

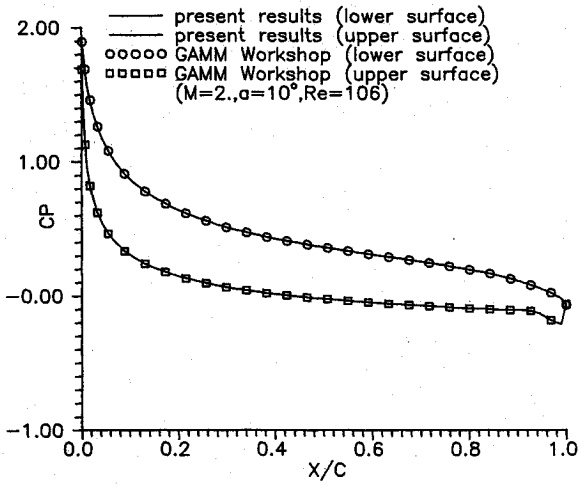

Fig. 4 Pressure-coefficient distribution for the supersonic laminar flow ($M_\infty = 2$, $\alpha = 10$ deg, $Re = 106$).

peared after the interpolation on the fine mesh. All of the results that follow have been obtained using the bilinear interpolation.

Bilinear extrapolation is also used for the calculation of the variables on the first volume of the fine mesh over the boundary (Fig. 1):

$$U_E = (^{15}/_{16})U_1 + (^{5}/_{16})U_2 - (^{1}/_{16})U_3 - (^{3}/_{16})U_4$$

Local Adaptive Mesh Solution

Generation of the Numerical Disturbances

The solution of the equations is not necessary to be obtained in the whole flowfield during the iterations. This fact originates from the nonuniformities of the flow variation towards a steady or unsteady solution. An analytical study of the numerical disturbances has been presented by Panaras[5] using an implicit approximately factored scheme for the thin-layer Navier-Stokes equations. A very simple form of partial mesh solution has also been used in the past by Cline[11] in the VNAP code.

The values of the numerical disturbances are decreased rapidly away from the solid boundary and the regions of the discontinuities (e.g., shock waves). The propagation of the numerical disturbances can be studied using the variations $|\Delta(\rho u)|$ and $|\Delta(\rho w)|$ of the conservative variables. In the present work, the behavior of the numerical disturbances has been examined

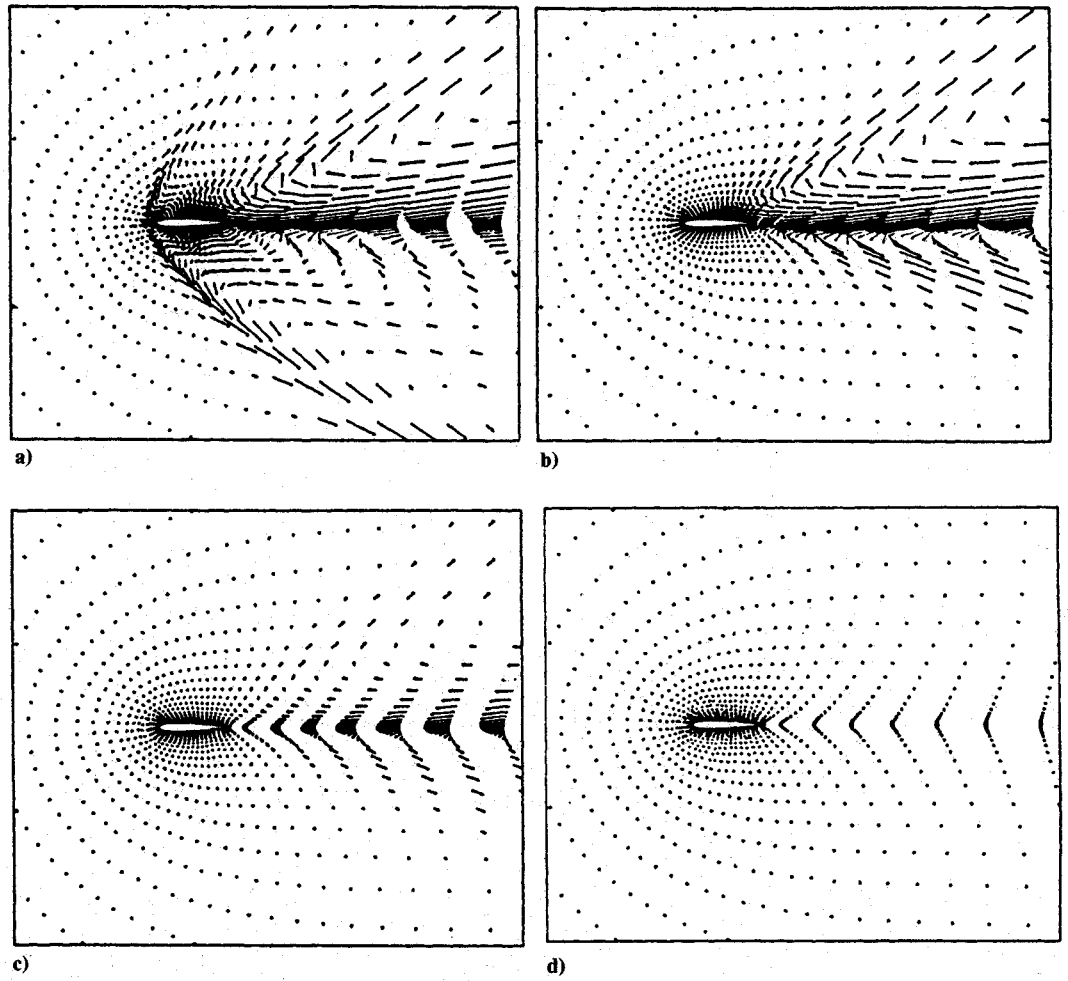

Fig. 5 Propagation of the numerical disturbances for the supersonic laminar viscous flow.

for a supersonic laminar viscous flow test case ($M_\infty = 2$, $\alpha = 10$ deg, $Re = 106$). The flow has been calculated using a 137 × 37 mesh (108 points around the airfoil). The computational mesh is C-type, generated by the Sorenson's method.[12] The outer boundary is located 9 chords from the airfoil, and the first mesh line has a distance of 5×10^{-4} chords from the surface and the wake line. An enlarged view of the fine mesh around the airfoil is shown in Fig. 3a. The iso-Mach lines are presented in Fig. 3b and the pressure coefficient distribution in Fig. 4. The results are compared with the corresponding results of a GAMM Workshop.[13]

The propagation of the disturbances is shown in Figs. 5a–5d plotting the quantity $\max|\Delta(\rho u)|$ multiplied by a factor 10^4–10^6 for presentation in the plots. Initially, the disturbances are larger in the region of the bow shock, in the boundary layer, and the wake of the flowfield (Fig. 5a). After a number of iterations, the disturbances disappear from the region of the bow shock and are propagated downstream (Figs. 5b, 5c). On the upper side, larger disturbances on the surface near the tailing edge are presented. Finally, the disturbances are limited to the wake region and die out when the convergence is achieved (Fig. 5d).

From the above plots, it is implied that the local solution can be applied only in regions where the disturbances are large because in the rest field the solution has been achieved. It is noted that the plots constitute a qualitative representation of the flowfield convergence because the quantity $\max|\Delta(\rho u)|$ has been multiplied by a factor 10^4–10^6. Convergence of the algorithm is determined by the convergence criterion, mentioned in previous section.

Construction Criteria of the Partial Meshes

The local solution of the equation is obtained in subregions of the fine mesh characterized by large values of the numerical disturbances. The computational cost in the subregions (partial meshes) of the fine mesh is very small in comparison with the cost in the whole flowfield. To construct partial meshes, regions of the fine mesh with large numerical disturbances must be detected. The detection parameter is the $\max\Delta Q = \max(|\Delta(\rho u)|, |\Delta(\rho w)|)$. The same criterion has also been proposed by Panaras[5] for the check of the convergence on the flowfield solution. The steady-state solution has been achieved when the quantity $\tilde{Q} = \max(|\Delta\rho|, |\Delta(\rho u)|, |\Delta(\rho w)|, |\Delta e|)$ tends to zero.

As input in the computational code a value for the $\max\Delta Q$ is given. This value (partial convergence criterion) is the limit at which the local solution starts to be obtained (for the present results it is about 10^{-3}). When the iterations meet this criterion, the procedure searches the whole flowfield and detects the regions where the numerical disturbances are larger than a second input criterion. The second input criterion, called total convergence criterion, is a value for the quantity $\bar{Q} = \max(|\Delta\rho|, |\Delta(\rho u)|, |\Delta(\rho w)|, |\Delta e|)$ defining the converged steady-state solution.

Construction of the Partial Meshes

For the construction of the partial meshes, two different processes may be selected. In the first, the detection process checks if a cell does not satisfy the total convergence criterion, storing the detection information into an array. A cell may satisfy the total convergence criterion while its neighboring volume may not satisfy it, and thus the equations must be solved, without

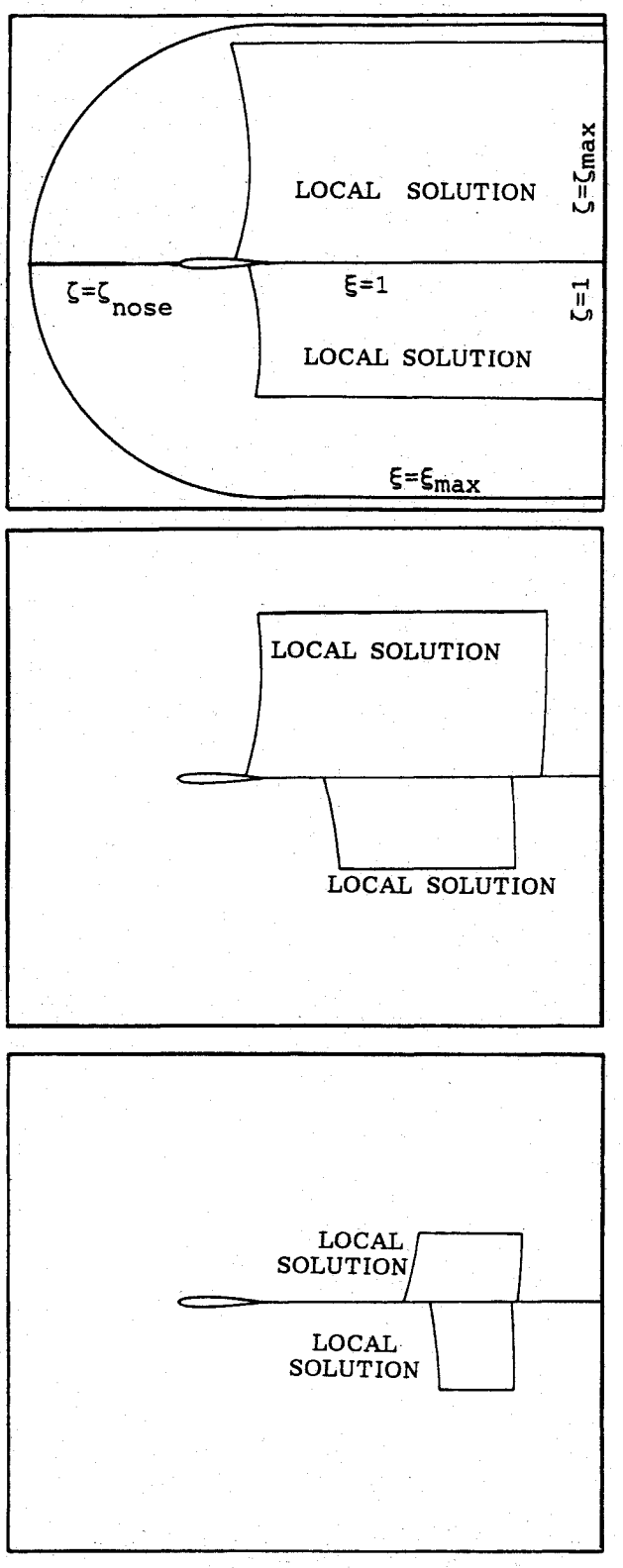

Fig. 6 Partial meshes for the local adaptive solution.

sense, only for this cell. The above process also needs an array for the storage of the detection information, increasing the computer storage. For the aforementioned reasons, this implementation is not used in the present study.

In the present study, an alternative process is proposed. Regions of the input fine mesh are defined as given prescribed detection blocks. The number of the prescribed blocks may be arbitrary (the summation of the blocks constitute the fine mesh). The detection process is obtained on each prescribed block independently from the others. The partial meshes are constructed into the prescribed blocks. Each block contains only one partial mesh or may be itself a partial mesh. For instance, the upper and lower sides of an airfoil topology may be the two prescribed detection blocks. The first block is contained between $\xi = 1 \div \xi_{max}$, $\zeta = 1 \div \zeta_{nose}$-lines, while the second one is contained between $\xi = 1 \div \xi_{max}$ and $\zeta = \zeta_{nose} \div \zeta_{max}$-lines, where ζ_{nose} intersects the leading edge of the airfoil. For sake of simplicity, partial meshes are constructed (into the detection blocks) as boxes (Fig. 6, for the supersonic viscous test case) between two ξ-lines and two ζ-lines of the fine mesh, and thus cells that satisfy the total convergence criterion may be included in the local solution. Searching each block, the first and the last volume, in which the total convergence criterion is not satisfied, correspond to the first and the last line of the partial mesh, respectively. Sweeps are obtained in each direction ξ, ζ for the determination of the two ξ-lines and two ζ-lines of the partial mesh box. The simplicity of the preceding procedure increases the versatility of the technique, without using complex algorithmic structures into the computer code.

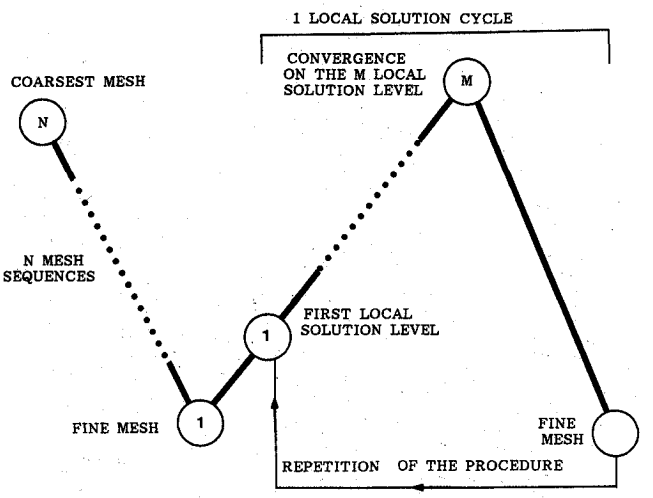

Fig. 7 Presentation of the total solution algorithm.

The solution of the equations is obtained on each partial mesh. The values of the conservative variables on the boundary cells of the partial meshes during the local solution can be calculated either as far-field boundary conditions (based upon the Riemann invariants[9,10]) or freezing previous updated values. Numerical test cases have shown that the boundary conditions on the partial meshes do not influence the stability of the solution and the convergence behavior. In the present test cases the boundary values are frozen.

The present local solution technique avoids the difficulties in the interfaces between coarse and fine meshes, which are presented in other adaptation methods.[14] This occurs because in the local solution there are not, at the same time, coarse and fine meshes, but only subsets (partial meshes) of the fine mesh. Thus, there is no need for any special treatment, as an interpolation procedure, in the boundaries of the partial meshes.

Adaptation Solution Procedure

The stage in which partial meshes are constructed for the first time is called the first local solution level (Fig. 7). Several solution levels can be constructed during the convergence to the steady-state solution. Input to the solver must be the criteria in which the solution levels start to be constructed, as well as the number of the local solution levels. Partial convergence criteria may be an input number of iterations, the aforementioned convergence number $\max\Delta Q$, or a criterion for the convergence rate. It has been found that an input convergence number for the $\max\Delta Q$ is a proper criterion. The most inappropriate criterion is an input number of iterations while the convergence rates depend on the flow test case. The required data for the adaptation of the local solution are sum-

marized as follows: 1) total convergence criterion; 2) partial convergence criterion for each local solution level; 3) number of detection blocks on the fine mesh; and 4) number of local solution levels.

All of the present calculations have been obtained by three local solution levels. The adaptation of the partial meshes is obtained into the prescribed blocks, and thus on each level partial meshes are constructed independently from their positioning on the previous level and so may appear in different positions on each level.

The solution algorithm consists of the following steps (Fig. 7)

1) A fine mesh is given as input, as well as all of the aforementioned criteria for the mesh sequencing technique, local solution, and adaptation.

2) The coarsest mesh (initial grid of the calculation) is constructed according to the description in the section of the mesh sequencing procedure. The iterations begin.

3) Computations are made on the sequence of the coarse meshes, and then interpolation of the initial guess to the fine grid is conducted.

4) Computation on the fine mesh continues until the residual falls below the value of the input partial convergence criterion for the first local solution level. The large values of the numerical disturbances are detected, and the partial meshes are constructed.

5) Local solution on the partial meshes of the present level is executed until the residual falls below the value of the input partial convergence criterion of the next local solution level. The partial mesh is constructed as on the next level.

6) Step 5 is repeated until the final local solution level, where the solution is marched to steady state (satisfaction of the total convergence criterion).

In this stage two choices can be selected: 1) termination of the computations or 2) return to the whole fine mesh. In the second case, the flow will be disturbed but rapid decay of the residual is obtained. Numerical experiments have shown that there is no difference in the results of flowfield if the calculations are terminated on the final local solution level. Of course, there are many alternative strategies as 3) selection of the case 2 before the solution converges on final local level. In this case, the solution is repeated from step 4; 4) Selection of case 2 and repetition of step 4 before the steady-state solution on the fine mesh. In this case, a "local solution cycle" is completed (Fig. 7). It has been found that satisfactory results are obtained terminating the calculations before step 6.

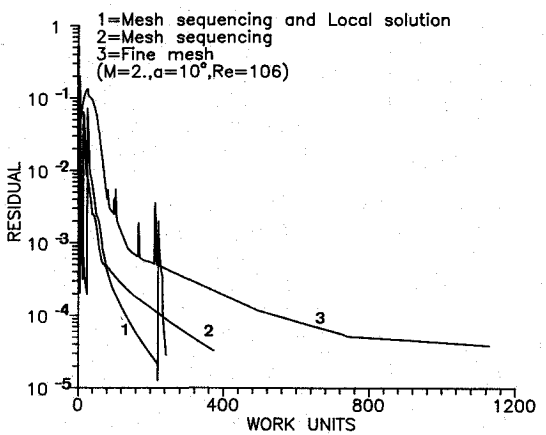

Fig. 8 Convergence histories for the supersonic laminar flow.

Acceleration of the Convergence

Supersonic Laminar Flow (M_∞ = 2, α = 10 deg, Re = 106) over NACA 0012 Airfoil

The convergence histories are presented in Fig. 8. Comparisons are shown between the single grid (fine mesh), mesh sequencing without local solution, and total procedure (mesh sequencing and local adaptive mesh solution), plotting the quantity $\bar{Q} = \max(|\Delta\rho|, |\Delta(\rho u)|, |\Delta(\rho w)|, |\Delta e|)$ (in the present viscous test cases the variation of the density is slightly higher than the other variations).

The local solution improves the mesh sequencing technique while a significant reduction of the computational time in comparison with the single grid is obtained. A work unit is defined as one time step on the finest grid. This figure shows that after the local solution and the return on the fine mesh, jump of the residual is presented but the decay is very fast. All of the calculations have been obtained on a workstation computer sys-

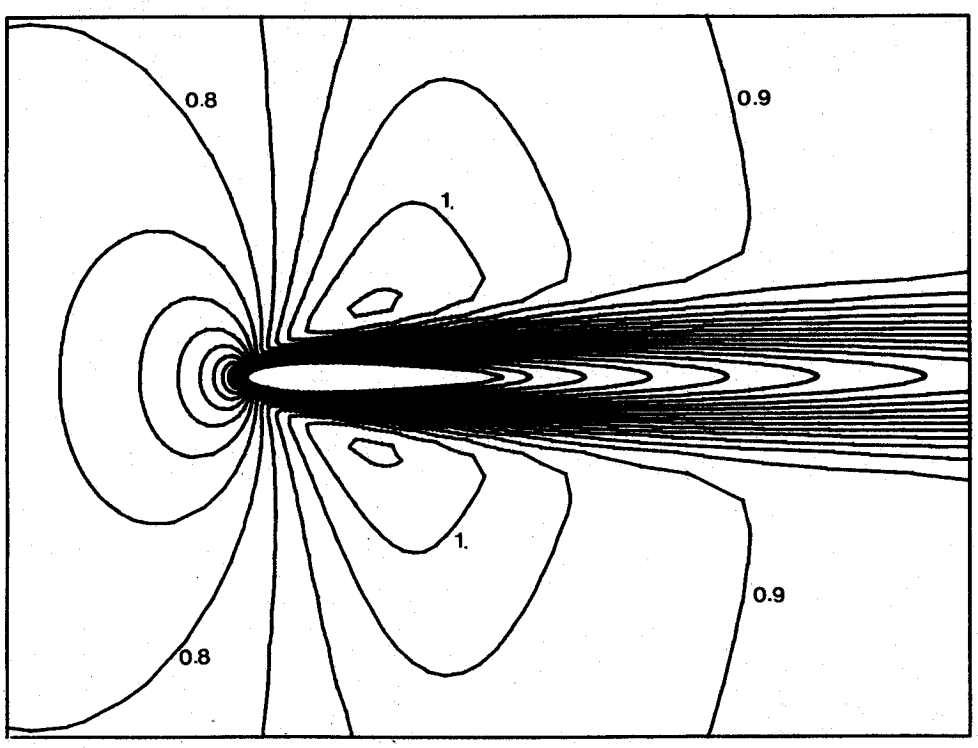

Fig. 9 Iso-Mach lines for the transonic laminar flow (M_∞ = 0.85, α = 0 deg, Re = 500).

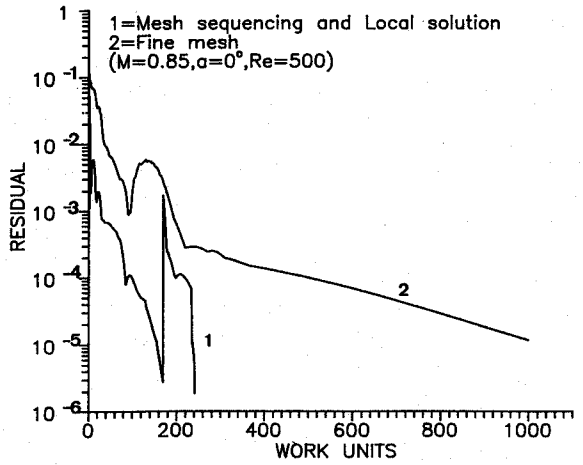

Fig. 10 Convergence histories for the transonic laminar flow.

tem (main processor consists of the R3000 and R3010 floating point coprocessor with 20-MHz clock frequency). The total CPU time (for $\tilde{Q} = 5 \times 10^{-5}$) is 17.4 h for the fine mesh, 5.5 h for the mesh sequencing, and 2.6 h using the combination of the local solution and the mesh sequencing technique. The jump of the residuals in the convergence history appeared after the interpolation on the mesh sequencing procedure or after the local solution and the return on the whole fine mesh.

Laminar Transonic Flow ($M_\infty = 0.85$, $\alpha = 0$ deg, $Re = 500$) over NACA 0012 Airfoil

The iso-Mach lines are plotted in Fig. 9. Comparison of the convergence history (Fig. 10), between the total procedure (mesh sequencing and local solution) and the fine mesh, indicates the effectiveness of the present technique. The decay of the residual after the local solution and the repetition of the solution on the fine mesh is rapid.

Transonic Inviscid Flow ($M_\infty = 0.95$, $\alpha = 0$ deg) over NACA 0012 Airfoil

The transonic inviscid flow over a nonlifting NACA 0012 airfoil has been studied using the above technique. In Fig. 11, the Mach number contours are plotted for the present test case. An oblique shock wave is formed at the trailing edge. The results have been compared with corresponding results from the AGARD test cases.[15] Figure 12 indicates excellent agreement for the pressure coefficient. For this test case the convergence histories are presented in Fig. 13. Comparisons are shown between the single grid (fine mesh), mesh sequencing without local solution, and total procedure (mesh sequencing and local adaptive mesh solution). Spikes occurring in the convergence-history case of the single fine mesh (curve 3), because the large skewness of mesh lines (in the oblique shock wave region at the trailing edge), cause very slow decay of the numerical disturbances in this region. This effect is confined using mesh sequences (curve 2). For the total procedure (curve 1), three local solution cycles (indicated by the letters A, B, and C in Fig. 13) have been used because returning on the fine mesh, after the local solution levels, the convergence is not rapid as in the previous test cases. The peaks in the curve 1 are not spikes but "jumps" of the residuals marching from the local solution levels to the fine mesh on each local solution cycle. It is noted that the calculations may be terminated before the cycle A. Figure 13 indicates that the convergence of the mesh sequencing procedure is improved by the local solution. In some regions of the curve 1 (Fig. 13), the convergence is almost vertical because the ratio of the number of partial mesh volumes over the number of volumes on the whole grid is very small and thus the computational work units are few.

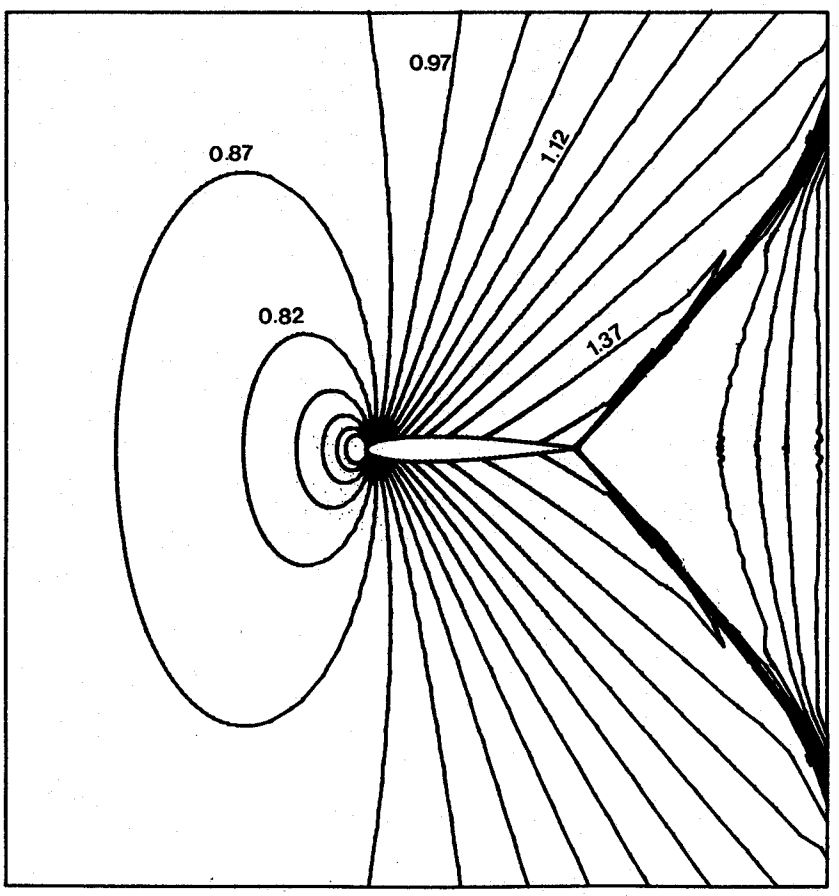

Fig. 11 Iso-Mach lines for the transonic inviscid flow ($M_\infty = 0.95$, $\alpha = 0$ deg).

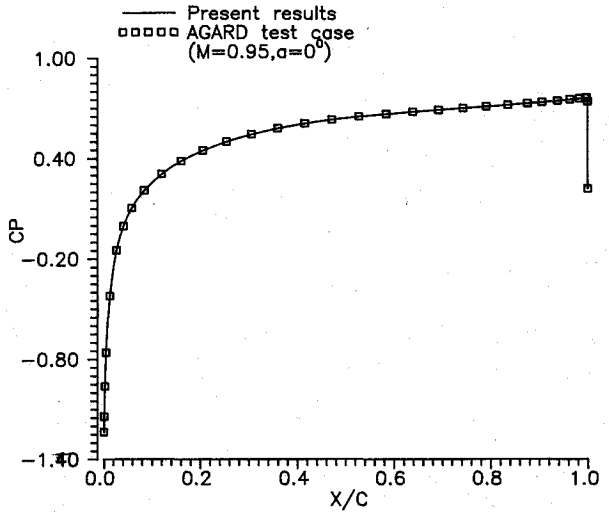

Fig. 12 Pressure-coefficient distribution for the transonic inviscid flow.

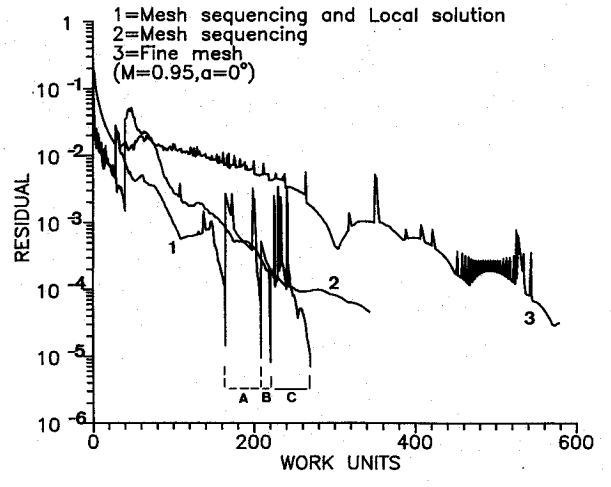

Fig. 13 Convergence histories for the transonic inviscid flow.

Conclusions

The basic conclusions of the present work can be summarized as follows:

1) The solution of the Euler and Navier-Stokes equations can be obtained with less computational work using the combination of mesh sequencing and the local adaptive mesh solution.

2) Reduction of the number of subiterations in the Newton method and the Gauss-Seidel relaxation can be used in the mesh sequencing and the local solution. This is an important advantage in the convergence acceleration of the Gauss-Seidel type methods.

3) The local adaptive mesh solution improves the known mesh sequencing technique.

4) Convergence criteria can be used for the adaptation of the partial meshes, as well as for the mesh sequencing procedure.

5) The present technique does not require the use of interfaces between the coarse and fine meshes because partial meshes are subregions of the fine mesh. Thus, an interpolation procedure is not needed in the boundaries of the partial meshes. The partial mesh boundary values are frozen during the local solution.

The preceding method is proposed as an effective tool for the reduction of the computational time in the numerical simulation of compressible flows.

Acknowledgment

We would like to thank A. G. Panaras for his helpful conversations on the subject of this work.

References

[1]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V: A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136.

[2]Eberle, A., "Characteristic Flux Averaging Approach to the Solution of Euler's Equations," *VKI Lecture Series*, Computational Fluid Dynamics, 1987-04, 1987.

[3]Chakravarthy, S. R., "High Resolution Upwind Formulations for the Navier-Stokes Equations," *VKI Lecture Series*, Computational Fluid Dynamics, 1988-05, 1988.

[4]Schmatz, M. A., "Simulation of Viscous Flows by Zonal Solutions of the Euler, Boundary Layer, and Navier-Stokes Equations," *Zeitscrift für Flugwissenschaften und Weltraumsforschung*, Vol. 11, No. 415, 1987, pp. 282–290.

[5]Panaras, A. G., "The Spatially Non-uniform Convergence of the Numerical Solution of Flows," *Journal of Computational Physics*, Vol. 82, No. 2, 1989, pp. 429–453.

[6]Schmatz, M. A., "Hypersonic Three-Dimensional Navier-Stokes Calculations for Equilibrium Gas, 7th Applied Aerodynamics Conf., AIAA Paper 89-2183, Seattle, WA, July 1989.

[7]Pulliam, T. H., and Steger, J. L., "Recent Improvements in Efficiency Accuracy and Convergence for Implicit Approximate Factorisation Algorithm," AIAA Paper 85-0360, Reno, NV, Jan. 1985.

[8]Dendy, J. E., Jr., "Black Box Multigrid," *Journal of Computational Physics*, Vol. 48, No. 3, 1982, pp. 366–382.

[9]Caughey, D. A., "Diagonal Implicit Multigird Algorithm for the Euler Equations," AIAA Journal, Vol. 26, No. 7, 1988, pp. 841–851.

[10]Jameson, A., and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal*, Vol. 25, No. 7, 1987, pp. 929–935.

[11]Cline, M. C., Los Alamos National Lab. Rep., LA-8872, Los Alamos, NM, 1981.

[12]Sorenson, R. L., "A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation," NASA TM-81198, 1980.

[13]Bristeau, M. D., Glowinski, R., Periaux, J., and Viviand, H., "Numerical Simulation of Compressible Navier-Stokes Flows," *Notes on Numerical Fluid Mechanics, Vol. 18, GAMM Workshop*, Vieweg & Verlag, Braunschweig, Germany, 1987, pp. 183–200.

[14]Kallinderis, J. G., and Baron, J. R., "Adaptation Methods for a new Navier-Stokes Algorithm," *Proceedings of the 8th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 87-1167, AIAA, New York, 1987.

[15]"Test Cases for Inviscid Flowfield Methods," AGARD Working Group 07, AR-211, 1985, pp. 6-29–6-35.