Block-Structured Solution of Euler Equations for Transonic Flows

Akin Ecer,* John T. Spyropoulos† and Vladimir Rubek†
Purdue University at Indianapolis, Indianapolis, Indiana

A block-structured solution scheme is developed for the solution of Euler equations for steady three-dimensional transonic flows. The Euler equations are expressed in terms of Clebsch variables $u = \nabla\phi + S\nabla\eta$. The resulting set of equations for conservation of mass, entropy (S) and the Clebsch variable η are solved using a block-based relaxation scheme. The flowfield is divided into a series of individual blocks representing local regions of the flowfield. The employed grid generation scheme provides local refinement of each block yet ensures one-to-one matching of grid points on the surfaces of adjoining blocks. For each of the blocks, either potential or Euler equations can be solved. For every block, the conservation of mass equation is solved to determine ϕ. For blocks with rotationality, two additional equations for determining S and η are solved. In this paper, the basic steps of the developed solution scheme are presented. The computational considerations involved in utilizing the present scheme are demonstrated for several two-dimensional sample problems.

I. Introduction

THE computation of three-dimensional transonic flows for complex configurations has attracted considerable attention during the last 15 years. In the present paper, a computational scheme is presented for solving Euler equations for three-dimensional, transonic flows. This work represents the continuation of previous efforts towards building such a computational capability. In order to explain the basic considerations that led to the development of the present scheme, it is useful to summarize some of these previous efforts.

Three-Dimensional Computational Grids

The computational work performed in the area of transonic flows in the past has established the necessity for reliable grid generation procedures for performing accurate computations. Considerable work has been done on mapping techniques for designing grids appropriate to given flowfields.[1] Since the solution schemes are based on finite element procedures, we have concentrated our efforts on generating "finite element grids," which can be classified as "unstructured grids." Although it is possible to design structured grids for simpler geometries, our assumption is that one needs unstructured grids for analyzing fully three-dimensional, complex flows.

When working with such complex three-dimensional geometries, it becomes apparent that controlling the generation of the grid at critical areas is extremely important. Although it is desirable to have grid generation schemes which can automatically adapt to local flowfields, it becomes computationally expensive to provide a general capability for adaptive grid refinement for three-dimensional transonic flows. We have chosen to develop capabilities for designing grids through an iterative process, where one can modify grids easily until the appropriate grid is generated. In generating such grids for three-dimensional problems, the experience was that local modifications took up a major portion of the total effort. Therefore, our efforts were concentrated on developing capabilities where one can rapidly make local modifications and incorporate these changes to the iterative solution schemes.

A block-structured grid generation scheme was developed for this purpose where the grids are generated locally for each block and the blocks are automatically connected together.[2] Although considerable work has been done on solving Euler equations by patching local flow zones with independent grid structures,[3] it was felt that it is important to ensure that the grid points are exactly matched on adjacent surfaces of neighboring blocks. Using the developed scheme, one can modify the grids locally while these changes are automatically propogated to the neighboring blocks in an organized manner.

Formulation of Euler Equations

In general, the unsteady Euler equations are cast in terms of the primitive variables (ρ,u,v,w,p) and numerically integrated with respect to real or pseudo-time to obtain steady solutions. The use of relaxation schemes for solving the steady Euler equations has attracted attention only during the last several years. In our efforts, we use a Clebsch transformation of the velocity field as follows:

$$u = \nabla\phi + S\nabla\eta \tag{1}$$

For irrotational flow regions, the above expression reduces to

$$u = \nabla\phi \tag{2}$$

Such a formulation is suitable for a block-structured solution scheme that unifies the treatment of rotational and irrotational flow regions as will be described later.

Another important aspect of the present formulation is the fact that all equations are cast into a second-order form and solved using a standard Galerkin procedure.[4] This is equivalent to using centered-difference approximations on second-order equations. No artificial viscosity, which would have been required for the solution of the first-order equations, is included in this case.

Numerical Solution of the Equation

Over the years, the numerical solutions of either potential or Euler equations using relaxation schemes have been mainly restricted to the use of structured grids. For example, the use of multigrid methods for analyzing unstructured grids has been attempted only recently.[7]

Received June 1986; revision received Feb. 1987.

*Professor. Member AIAA.
†Research Associate. Member AIAA.

When working with unstructured grids, one has to consider a more general problem. If the computational grid can be considered as an assembly of finite elements with irregular geometries, one has to store the nodal connectivities of each element as well as their geometries. When working with large problems, the storage of such information becomes a difficult task. If one is restricted to developing solution schemes where all of the geometry information has to be stored as a single data set, it becomes very easy to exceed the storage limitations of a given computer.

Because of these considerations, we have chosen a block-structured solution scheme where one divides the problem into a series of substructures. The main assumption is that the blocks are small enough to be processed in the main storage, where all the geometry and solution related information can be stored at one time. Also, starting with this assumption, we chose to employ a block-relaxation scheme rather than a point or line-relaxation scheme for treating unstructured grids in a general manner. In fact, the point will be made that the computational efficiency of the present scheme depends mostly on transferring the data, rather than processing it for irregular grids. In this paper, we provide the basic formulation for the solution of Euler equations which follows the procedure previously developed for potential flows.[8]

II. Mathematical Formulation of Euler Equations

The derivation of the Euler equations in terms of the Clebsch variables can be systematically developed using a variational formulation.[4] For iso-energetic flows, one can write a variational functional.

$$\Pi = \int_V \left[\tfrac{1}{2}\rho \boldsymbol{u}\cdot\boldsymbol{u} - \rho E(\rho,S) + \rho H + \phi\{\nabla\cdot(\rho\boldsymbol{u})\} + \eta\{\nabla\cdot(\rho S\boldsymbol{u})\}\right] dV \tag{3}$$

where ϕ is the Lagrange multiplier for the conservation of mass equation and η the Lagrange multiplier for conservation of entropy equation. On the boundaries, one has to define the normal mass flux and normal entropy flux. If the entropy is a constant, then only the conservation of mass equation is needed in the above functional. Following the above variational problem, one can derive the Clebsch transformation of the velocity vector in Eq. (1) as well as the following set of governing equations:

$$\nabla\cdot(\rho\boldsymbol{u}) = 0 \tag{4}$$

$$\rho\boldsymbol{u}\cdot\nabla S = 0 \tag{5}$$

$$\rho\boldsymbol{u}\cdot\nabla\eta = -p/R \tag{6}$$

The conservation of mass equation is in second-order form in terms of ϕ and η, while the other two equations are first-order. These equations can also be cast in a second-order form,[6] as follows:

$$\rho\boldsymbol{u}\cdot\nabla(\rho\boldsymbol{u}\cdot\nabla S) = 0 \tag{7}$$

$$\rho\boldsymbol{u}\cdot\nabla(\rho\boldsymbol{u}\cdot\nabla\eta) = -\rho\boldsymbol{u}\cdot\nabla(p/r) \tag{8}$$

Equations (5) and (6) become the boundary for Eqs. (7) and (8).

The solution of the above set of equations is equivalent to solving the mass and momentum equations in the form of primitive variables.[7] In the above formulation, the primitive variables become ϕ, S, and η. These are updated at each step of the relaxation scheme, where the velocity field is defined from Eq. (1) at each step as a derived variable. Pressure is also computed from the local flow speed by assuming iso-energetic flow conditions. The details of the finite element formulation of these equations can be found in Refs.[4-6] In this paper, only the details of the formulation relating to a block-structured solution scheme will be presented.

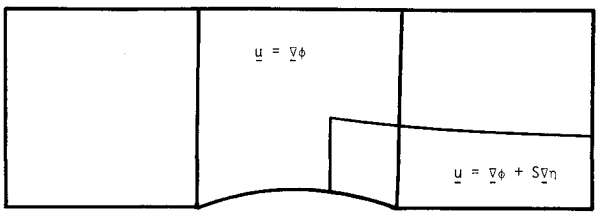

Fig. 1 Block-structure for analyzing the flow over a 10% circular bump in a channel ($\mu_\infty = 0.675$).

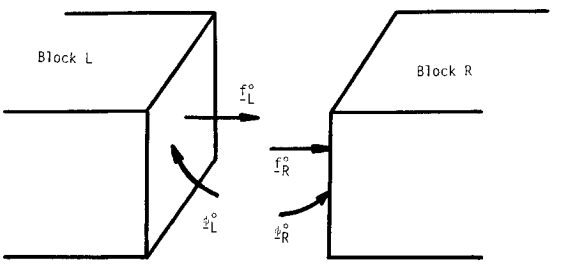

Fig. 2 Description of boundary mass fluxes and ϕ vectors on the surfaces between neighboring blocks.

III. Computational Scheme

For the block-structured solution of Euler equations, the flowfield is divided into a series of blocks as shown in Fig. 1. As can be seen from this figure, certain flow regions can be irrotational, while the rotationality generated by the shock affects only certain other blocks. In this case, the entropy is equal to a constant for all the blocks that remain irrotational, and only the conservation of mass equation has to be solved for such blocks. The overall relaxation scheme can be summarized in the following steps: 1) Assume a velocity field, 2) solve Eqs. (7) and (8) for each block with rotationality, 3) solve conservation of mass equation for each block (Eq. 4), and 4) calculate a new velocity field from Eq. (1) or Eq. (2) and repeat the procedure.

Solution of Conservation of Mass Equation

This part of the procedure was developed and tested originally for potential flows.[8] Starting with the conservation of mass equation

$$\nabla\cdot(\rho\boldsymbol{u}) = 0 \tag{9}$$

and substituting Eq. (1), one obtains

$$\nabla\cdot(\rho\nabla\phi) = -\nabla\cdot(\rho\eta\nabla S) \tag{10}$$

In the case of blocks with irrotational flowfields, the right-hand side of Eq. (10) is identically zero. For rotational blocks, the right-hand side is determined from the calculated values of S and η of each iteration. For elements located in the supersonic pocket, local densities are upwinded by using a local artificial viscosity parameter μ.[9]

For each block, the equations are solved in two steps. First, Eq. (10) is solved for each block individually by specifying the normal mass flux on all block boundaries, and the Clebsch variable ϕ^0 is calculated on all block surfaces. On each interface between the two neighboring blocks, shown in Fig. 2,

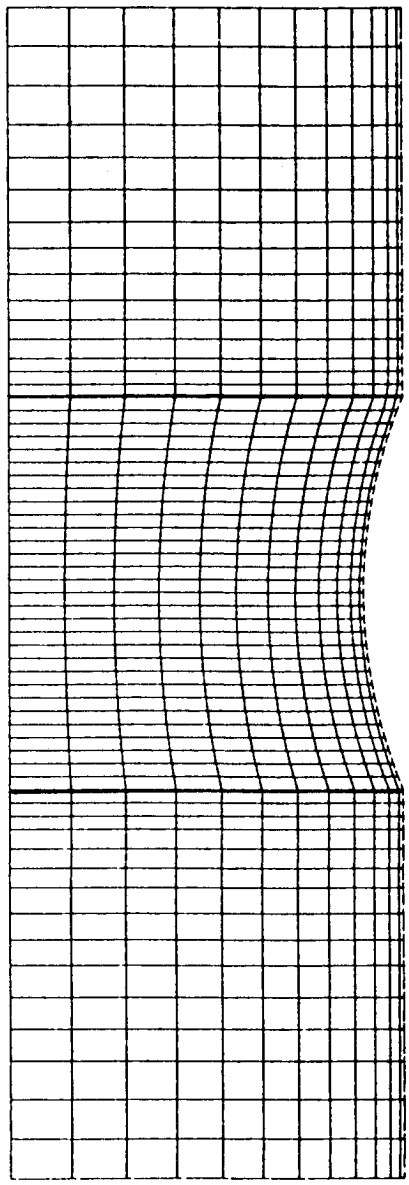

Fig. 3 Finite-element grid for the channel with a 10% bump.

the values of ϕ^0 are balanced in the following manner:

$$\phi_R^n = (1-\omega_\phi)\,\phi_R^0 + 0.5\omega_\phi(\phi_R^0 + \phi_L^0) \tag{11}$$

$$\phi_L^n = (1-\omega_\phi)\,\phi_L^0 + 0.5\omega_\phi(\phi_R^0 + \phi_L^0) \tag{12}$$

when 0 and n indicate the calculated and balanced values of velocity potentials on the surfaces of neighboring blocks R and L, and ω_ϕ is a relaxation parameter.

During the second step, the conservation of mass equation is solved for a single block by specifying calculated values of ϕ^n on each of the block boundaries. On each interface between the neighboring blocks, the values of normal mass flux $f = \rho\phi_n$ are calculated for all block surfaces. On each interface between the neighboring blocks, as shown in Fig. 2, the flux values are balanced in the following manner:

$$f_R^n = (1-\omega_f)\,f_R^0 + 0.5\omega_f(f_R^0 + f_L^0) \tag{13}$$

$$f_L^n = (1-\omega_f)\,f_L^0 + 0.5\omega_f(f_R^0 + f_L^0) \tag{14}$$

In the above equation, R and L indicate the blocks on each side of an interface, 0 and n describe the calculated and balanced values of the boundary mass fluxes. ω_f is again a relaxation parameter.

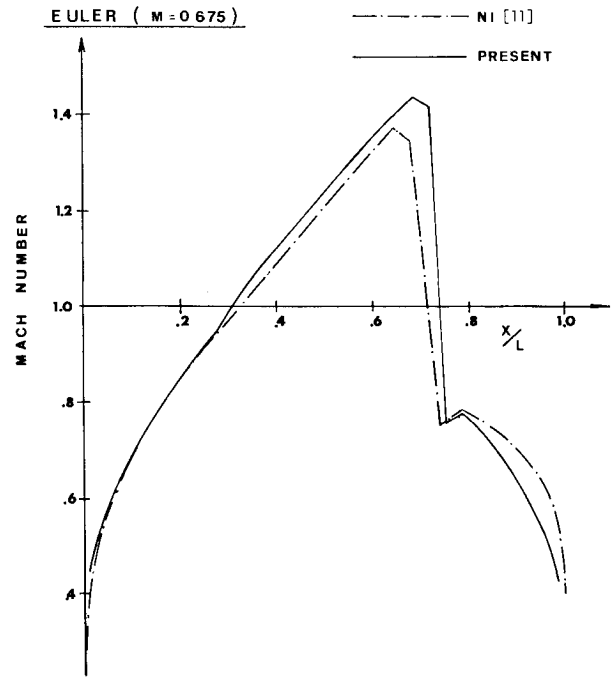

Fig. 4 Pressure distribution over the 10% circular bump.

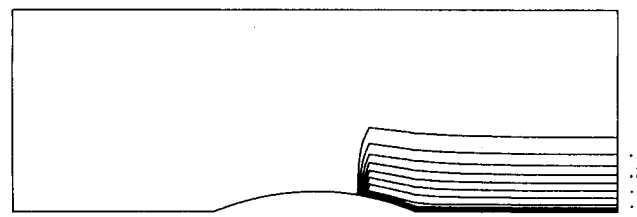

Fig. 5 Vorticity/pressure contours for bump case.

In the solution of the conservation of mass equation with both types of boundary conditions, an approximate solution procedure is employed. The iterative scheme employed can be written as

$$\boldsymbol{K}_{\phi\phi}^n \Delta\phi^{n+1} = \omega_\phi(f_\phi - \boldsymbol{K}_{\phi\phi}^n \phi^n - \boldsymbol{K}_{\phi\eta}^n \eta^n) \tag{15}$$

where $\boldsymbol{K}_{\phi\phi}$ is the symmetric Laplace operator, ω a relaxation parameter, and

$$\Delta\phi^{n+1} = \phi^{n+1} - \phi^n \tag{16}$$

However, instead of using the exact operator $\boldsymbol{K}_{\phi\phi}^n$ on the left-hand side of the equations at each iteration step, the following approximate solution scheme is employed.

$$\boldsymbol{K}_{\phi\phi}^0 \Delta\phi^{n+1} = \omega_\phi(f_\phi - \boldsymbol{K}_{\phi\phi}^n \phi^n - \boldsymbol{K}_{\phi\eta}^n \eta^n) \tag{17}$$

Here, $\boldsymbol{K}_{\phi\phi}^0$ is the initially computed Laplace operator which is usually calculated by setting the local Mach number to zero during the computations.[10] It provides a symmetric banded coefficient matrix for each block which is decomposed into a triangular form and stored only once. During every iteration step, the residual vector is assembled for each block and a forward-backward substitution is performed on this decomposed coefficient matrix.

It is important to note that all of the block and surface operations can be performed in any order and processed using

Table 1 Convergence history of the potential solution

Iteration steps	ω_ϕ	μ
1–150	0.1	8.0
151–300	0.1	6.0
301–450	0.2	5.0
451–620	0.2	4.5
621–760	0.2	4.25
761–900	0.2	4.15

Table 2 Convergence history of the Euler solution

Iteration steps	ω_ϕ	ω_s, ω_η	μ	Solution type
1–150	0.1	—	8.0	Potential
150–300	0.1	—	6.0	Potential
301–380	0.2	—	5.0	Potential
381–400	0.2	1.0	5.0	Nonisentropic potential
401–600	0.2	0.2	5.0	Euler
601–870	0.2	0.2	4.5	Euler

only the information for describing individual blocks and their neighbors. This provides considerable flexibility in utilizing the computer resources as will be described later.

Solution of the Conservation of Entropy Equation

As described above, the first step of the iterations involve the determination of the entropy and Laplace multiplier η for each of the blocks with rotational flowfields. Starting with an assumed velocity field, Eqs. (7) and (8) are solved together since the differential operator is the same for both equations. The solution of these equations in second-order form is an important consideration. It ensures that the conservation of entropy S and η is always satisfied. Also, since for the Euler equations, the vorticity can be written as

$$\omega = \nabla S \times \nabla \eta \tag{18}$$

for second-order accurate solutions of S and η, one also obtains accurate solutions of rotationality itself.[5]

For the solution of Eqs. (7) and (8), one of the three types of flow conditions may exist in each block. In the example in Fig. 1, for the first block on the left, the flowfield is irrotational. In this case, both S and η can be set equal to an arbitrary constant for all the nodes of this block, and Eqs. (7) and (8) need not be solved for such a block.

For the second block in the flow direction, a shock occurs in the flowfield. The flow is rotational only in the flow region past the shock. For this block, S and η are set to a constant for all the nodes located in the irrotational flow region. Shock is detected as a jump in the local Mach numbers across the neighboring elements in the flow direction. The entropy jump is calculated from the Rankine-Hugoniot conditions[4] and specified as boundary condition along the shock. Equations (7) and (8) are then solved for all grid points downstream of the shock using a relaxation scheme similar to the one described in Eq. (16).

$$K^0_{ss}\Delta S^{n+1} = \omega_s\left(f_s - K^n_{ss}S^n\right) \tag{19}$$

$$K^0_{\eta\eta}\Delta \eta^{n+1} = \omega_\eta\left(f_\eta - K^n_{\eta\eta}\eta^n\right) \tag{20}$$

The coefficient matrices, K^0_{ss} and $K^0_{\eta\eta}$ represent the same second-order form of the convection operator in Eqs. (7) and (8). These matrices may be updated periodically after several iteration steps as the flowfield develops.

For the third block in the flow direction, which is located down-stream of the shock, the complete set of equations has to be solved in this case. S and η are specified at the upstream boundary of the block as determined from the downstream conditions of the neighboring block. As will be discussed later, during the application of the computational scheme, the blocks can be processed in any order for the solution of the conservation of mass equation. However, the block-by-block solution of Eqs. (7) and (8) are performed by sweeping the blocks in the flow direction.

IV. Numerical Results

Two transonic flow cases were tested using the developed procedure for two-dimensional flows.

10% Circular Bump in a Channel

The first test case involves the flowfield in a channel with a 10% circular bump and an upstream Mach number of 0.675. The block-structure for this case consisting of three blocks is shown in Fig. 1. In both cases, the block interfaces were placed at critical sections, such as the stagnation points, to test the robustness of the developed scheme. The finite-element grid employed in this analysis is shown in Fig. 3. The pressure distribution on the lower surface of the channel is shown in Fig. 4 and compared with results reported by N_i.[11] Calculated nondimensionalized vorticity/pressure contours are shown in Fig. 5. Calculated vorticity/pressure profiles at the shock and channel exit as well as the Mach number

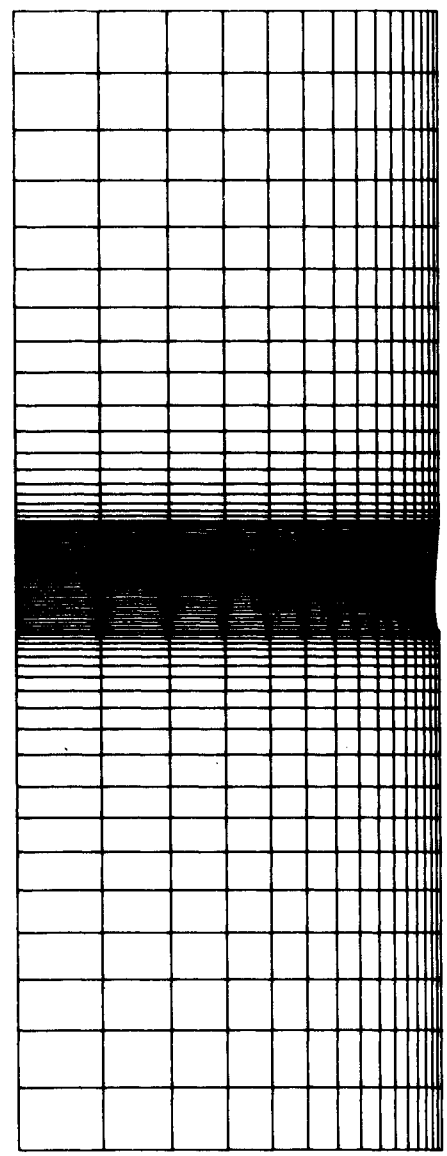

Fig. 9 Finite-element grid for the NACA0012 airfoil case.

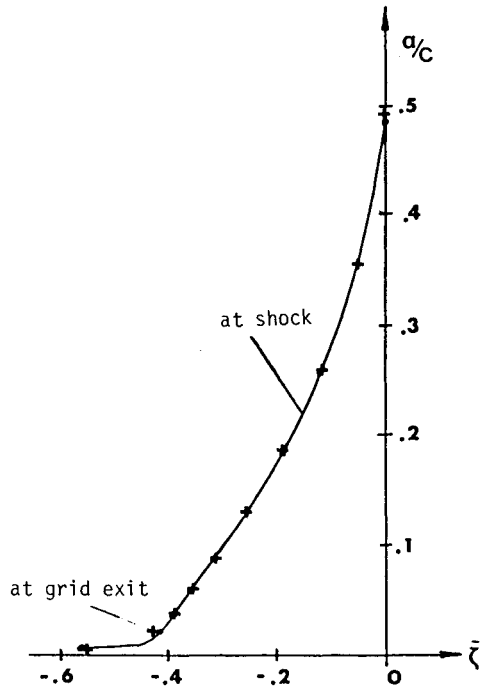

Fig. 6a Nondimensional vorticity/pressure profiles for bump case.

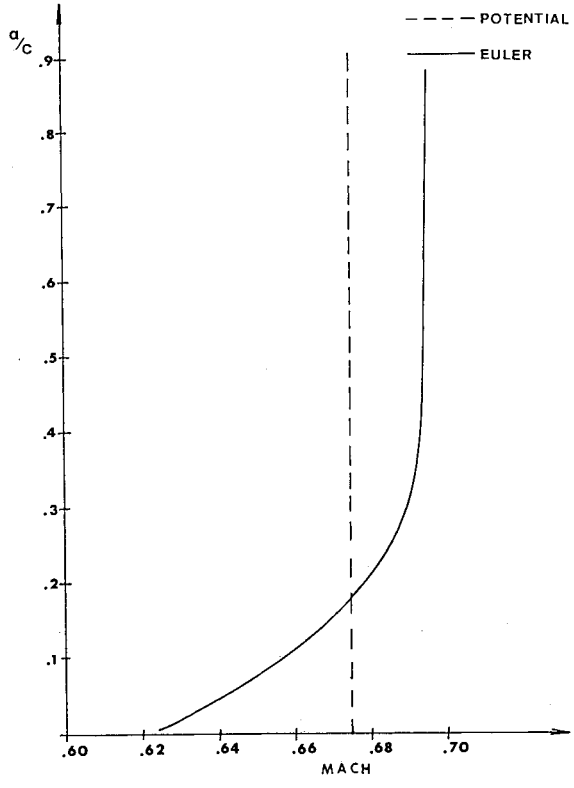

Fig. 6b Mach number profile at the channel exit.

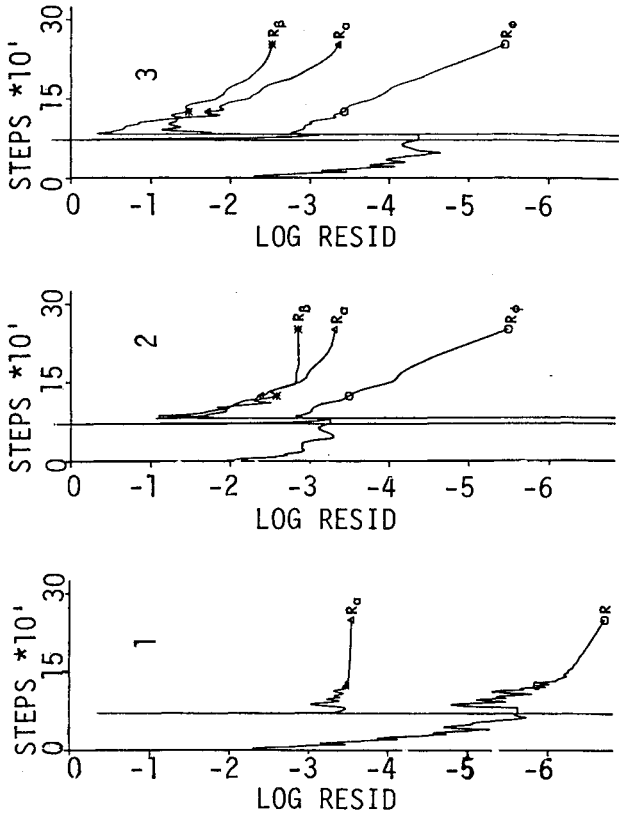

Fig. 7 Residual history for the different blocks in bump case.

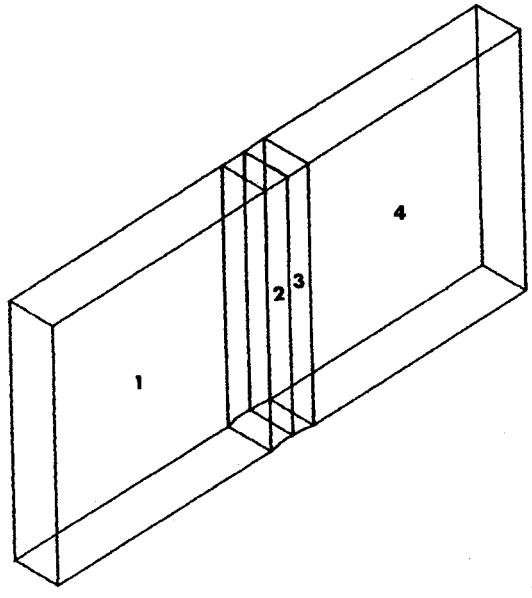

Fig. 8 Block-structure for analyzing the flow over a NACA0012 airfoil ($\mu_\infty = 0.85$).

profiles at the exit are shown in Fig. 6. The residual history for each block is shown in Fig. 7. As can be seen from this figure, the grid points located at the furthest down-stream block are slowest to converge. The iterations were performed using the following procedure:

1) 40 iteration steps potential solution for all blocks (artificial viscosity $\mu = 5.0$, $\omega_\phi = 0.4$).
2) 30 iteration steps potential solution for all blocks (artificial viscosity $\mu = 3.0$, $\omega_\phi = 0.4$).
3) 10 iteration steps potential solution for the first block; a nonisentropic potential solution for the other two blocks.[12] In this case, Eqs. (7) and (8) were solved for these two blocks with rotational flowfields, but Eq. (2) was employed for determining the velocity field ($\mu = 3.0$, $\omega_\phi = 0.4$).
4) 170 iteration steps potential solution for the first block and an Euler solution for the other two blocks ($\mu = 2.0$, $\omega_\phi = \omega_S = \omega_\eta = 0.4$).

As can be seen from these results, the iterations start by obtaining an approximate potential solution to locate shock.

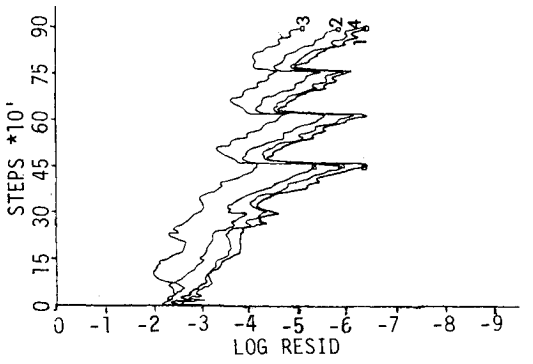

Fig. 10 Residual history for potential flow solution for the different blocks in NACA0012 airfoil case.

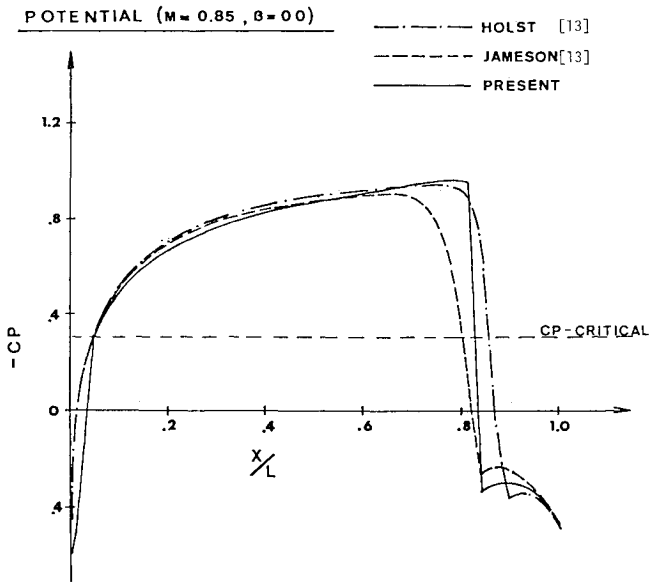

Fig. 11 Pressure distribution over the NACA0012 airfoil from the potential flow solution.

This potential solution provides a stronger shock than what is predicted by an Euler solution. One can use high values of artificial viscosity in the supersonic pocket to obtain a quick yet approximate-potential solution. At this point, by switching to a nonisentropic potential flow solution, the shock strength is recalculated, and the entropy generated at the shock is convected downstream in terms of density corrections only. This procedure provides a shock strength which is much closer to an Euler solution. Finally, the full Euler solution is obtained at a lower level of artificial viscosity in the isentropic, supersonic pocket. Obviously, no artificial viscosity is employed in the rotational but subsonic flow regions.

NACA0012 Airfoil with $\mu_\infty = 0.85$

The second example is the flow around a NACA0012 airfoil with an upstream Mach number of 0.85. In this case, the block-structure consisted of 6 blocks, as shown in Fig. 8. The finite element grid employed in the analysis is shown in Fig. 9. For this problem, two experiments were conducted: to discover the potential flow solution and the Euler solution.

Potential Flow Solution

First a potential solution was obtained for this problem by solving the potential flow equations for all blocks. The history of relaxation factors and the artificial viscosity multiplier for the supersonic flow region are shown in Table 1. For high

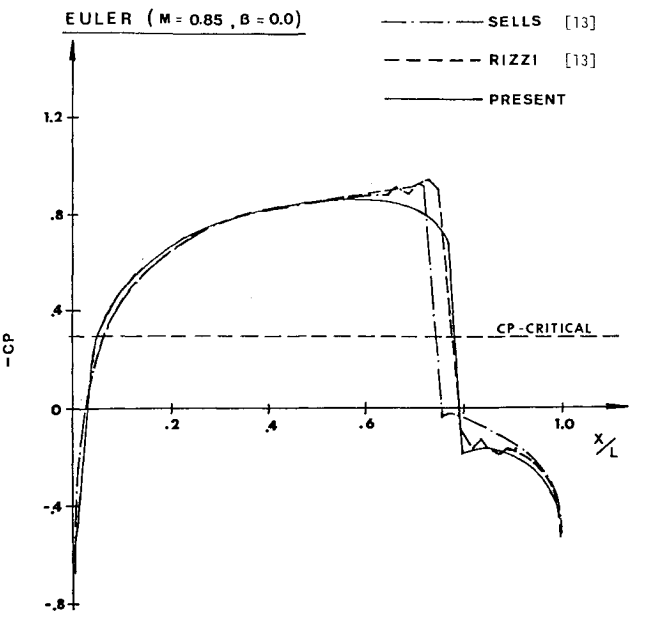

Fig. 12 Pressure distribution over the NACA0012 airfoil case from the potential flow solution.

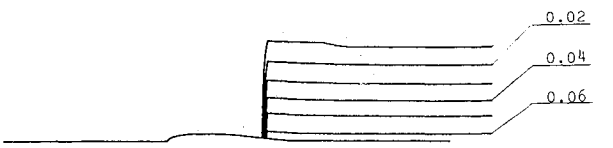

Fig. 13 Contours for nondimensional vorticity/pressure values ($\Delta\xi = 0.01$) for NACA0012 case.

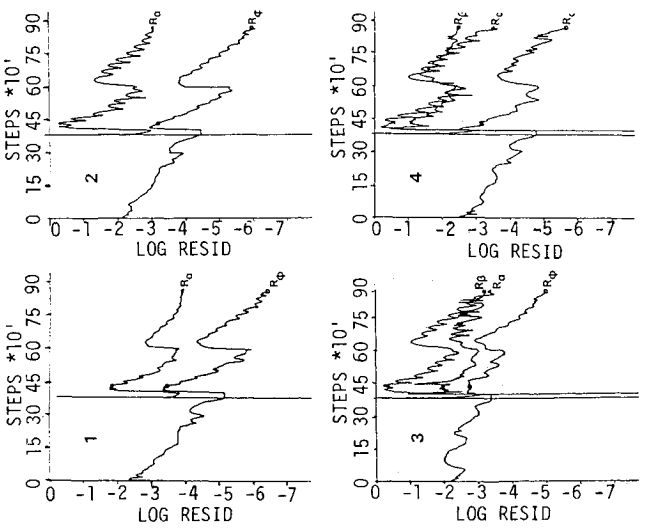

Fig. 14 Residual history for Euler solution for the different blocks in NACA0012 airfoil case.

values of artificial viscosity, and by using low relaxation parameters, the shock was initially captured. The artificial viscosity was reduced for better accuracy and the relaxation parameter was increased. The residual history for this solution is shown in Fig. 10. A comparison of computed pressure coefficients over the surface of the airfoil with others is shown in Fig. 11.

Euler Solution

For obtaining an Euler solution to the same problem, the same procedure was repeated. First, 380 steps of iteration

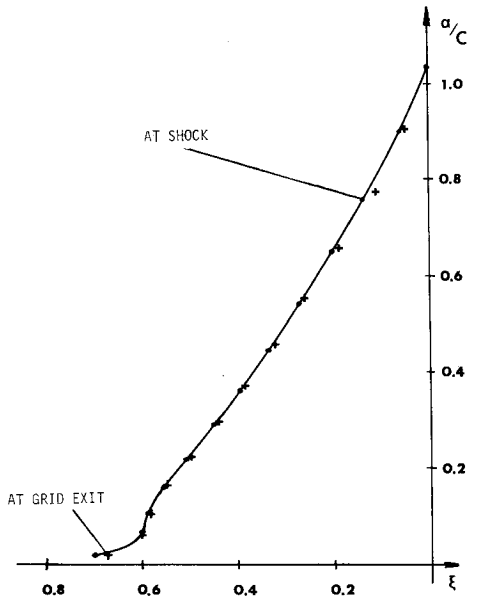

Fig. 15a Nondimensional vorticity/pressure profile for NACA0012 case.

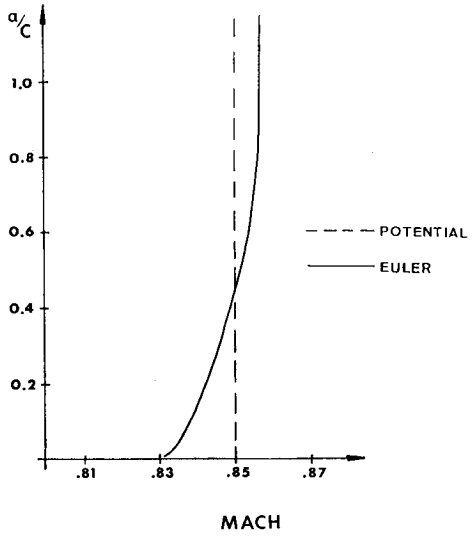

Fig. 15b Mach number profile at grid exit for NACA0012 airfoil case.

were conducted by gradually lowering the artificial viscosities to obtain an approximate potential solution only. After a shock was developed, 20 iteration steps were performed using the nonisentropic potential formulation to correct the strength of the shock. Finally, 470 steps of iteration were performed to obtain an Euler solution. The history of iterations are summarized in Table 2. The comparison of obtained surface pressure distributions with results obtained by others are shown in Fig. 12. The vorticity/pressure contours are shown in Fig. 13. The history of iterations for each block is shown in Fig. 14. Finally, the vorticity/pressure and Mach profiles at the shock location and at the downstream of the computational grid are shown in Fig. 15. As can be seen from these results, the convergence rates depend strongly on the artificial viscosity employed in the relaxation scheme for both potential and Euler solutions.

Concluding Remarks

The procedure provides a general approach to the solution of potential and Euler equations in a block-structured form. The formulation is being extended to the solution of Navier-Stokes equations where one can include the viscous effects in the solution of entropy equation for selected blocks as discussed.

As demonstrated through the simple examples, both the accuracy and convergence of transonic flow problems require proper treatment of local flow characteristics. This block-structured solution scheme provides the necessary tools for generating and modifying the grids locally, and iterating the equations in a most efficient manner and possibly performing local changes during the iterations. For the solutions of large problems with complex geometries, this method provides the necessary capabilities for utilizing given computer resources in a most efficient manner. Using this procedure, one can solve large, three-dimensional problems using parallel processing capabilities.

References

[1]Thompson, J.F., (ed.), *Numerical Grid Generation*, Elsevier Science Publishing Co., New York, 1982.

[2]Ecer, A., Spyropoulos, J.T. and Maul, J., "A Block-Structured Finite Element Grid Generation Scheme for the Analysis of Three-Dimensional Transonic Flows," *AIAA Journal*, Vol. 23, Oct., 1985, pp. 1483–1490.

[3]Hessenius, K.A. and Rai, M.M., "Applications of a Conservative Zonal Scheme to Transient and Geometrically Complex Problems," AIAA Paper 84-1532, June 1984.

[4]Ecer, A., and Akay, H.U., "Finite Element Formulation of Euler Equations for the Solution of Steady Transonic Flows," *AIAA Journal*, Vol. 21, March 1983, pp. 410–416.

[5]Akay, H.U. and Ecer, A., "Application of a Finite Element Algorithm for the Solution of Transonic Euler Equations," *AIAA Journal*, Vol. 21, Nov. 1983, pp. 1518–1524.

[6]Ecer, A., Akay, H.U., and Sener, B., "Solution of Three-Dimensional Inviscid Rotational Flows in a Curved Duct," AIAA Paper 84-0032, Jan. 9–12, 1984.

[7]Lohner, R. and Morgan, K., "Improved Adaptive Refinement Strategies for Finite Element Aerodynamic Computations," AIAA Paper 86-0499, Jan. 1986.

[8]Ecer, A. and Spyropoulos, J.T., "Block-Structured Solution Scheme for Analyzing Three-Dimensional Transonic Potential Flows," AIAA Paper 86-0510, Jan. 1986.

[9]Ecer, A. and Akay, H.U., "Investigation of Transonic Flow in a Cascade Using the Finite Element Method," *AIAA Journal*, Vol. 19, Sept. 1981, pp. 1174–1182.

[10]Akay, H.U. and Ecer, A., "Finite Element Analysis of Transonic Flows in Highly Staggered Cascades," *AIAA Journal*, Vol. 20, Feb. 1982, pp.

[11]Ni, R.H., "A Multiple Grid Scheme for Solving Euler Equations," AIAA Paper 81-1025, *AIAA Computational Fluid Dynamics Conference*, June 1981.

[12]Akay, H.U., Ecer, A., and Willhite, P.A., "Finite Element Solutions of Euler Equations for Lifting Airfoils," *AIAA Journal*, Vol. 24, April 1986, pp. 562–569.

[13]Rizzi, A. and Viviand, H. (eds.), *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves*, Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, FRG, 1981.