AIAA JOURNAL
Vol. 30, No. 3, March 1992

Algebraic Turbulence Modeling for Adaptive Unstructured Grids

Yannis Kallinderis*
University of Texas at Austin, Austin, Texas 78712

Algebraic turbulence models are quite common in flow numerical simulations. Their simplicity and low computation cost, as well as their applicability to specific classes of flows, have made them attractive compared to the more expensive multiple-equation models. Adaptive, unstructured grid algorithms have become quite popular over the last years, due to their flexibility in treating complex geometries, as well their high-resolution capability for local flow phenomena. However, application of algebraic turbulence models with unstructured grids proves to be nontrivial. The difficulties associated with implementation of those models with such grids have generally impeded use of unstructured grids for turbulent flow simulations. This paper presents a novel method of implementing an algebraic turbulence model with embedded, unstructured grids. The Baldwin-Lomax model was chosen in order to test the implementation. Modifications to the application of the original model on structured grids are suggested as well. Comparisons with both experiments and other numerical simulations are employed in order to evaluate accuracy of the implementation method.

I. Introduction

CONSIDERABLE progress has been made in the development of numerical methods for the solution of the Navier-Stokes equations. However, most of those methods are impractical for complicated flows in a design environment. The primary reason is poor efficiency, which makes it difficult to obtain accurate results. Very fine resolution is needed, which results in long computation times even with the use of available supercomputers.

In general, the selection of the equations to be solved, the scheme, and the grid all are determined a priori by the user before starting the solution procedure, and quite often some and even all of these factors must be modified by the user in order to improve the results. The robustness of current numerical schemes as well as present computer capabilities has recently allowed a dramatic change in this philosophy. General algorithms have been developed that are flexible enough to adaptively adjust those parameters during the solution procedure without intervention by the user. One of the most efficient adaptive methods consists of local refinement by dividing grid cells. An initially coarse grid is embedded in regions with large flow gradients (e.g., boundary layers, shocks, wakes, etc.). The algorithm senses high gradient regions and automatically divides grid cells in such regions. This approach has been used for the resolution of shocks in flowfields described by Euler equations.[1,2] The embedded scheme was used as part of an adaptive algorithm for turbulent flows.[3-6]

Algebraic turbulence models are quite common in aerodynamics numerical simulations. Their simplicity and low computation cost, as well as their accuracy for specific classes of flows, have made them attractive compared to the more expensive multiple-equation models. However, such models require nonlocal information that characterizes an entire shear-layer profile. The adaptive algorithm results in an unstructured mesh, which is a major issue in the implementation of an algebraic turbulence model. Grid interfaces frequently interrupt normal-to-wall meshlines, and, therefore, models that employ information from an entire shear-layer profile cannot be applied. Similarly, in the case of triangular meshes, such normal grid lines generally do not exist. A new method of applying algebraic turbulence models with unstructured grids was presented in Ref. 4. Another method for embedded grids was developed in Ref. 7, which is based on using the initial structured mesh for computing the eddy viscosity values. Rostand[8] and Mavriplis[9] developed implementation methods for triangular meshes.

In the following, the method of implementation is presented. Then, the specific algebraic model that was employed is described and applied with unstructured embedded grids. Modifications to the original model are also suggested. Finally, the implementation method is evaluated in terms of accuracy and computer requirements.

II. Algebraic Turbulence Model Implementation Procedure

Adaptive grid embedding consists of a local refinement by dividing grid cells. An initial coarse grid is embedded in those local regions where the algorithm detects large flow gradients. Several levels of finer grids are allowable and can be limited to those regions of the domain in which important or relevant features exist. Generally, the borders of the embedded regions have an irregular shape.

The quantities required to evaluate the eddy viscosity can be divided into two groups: those that characterize each cell (local quantities), and others that characterize an entire boundary-layer profile (global quantities). The latter quantities cannot be obtained in the normal way since the presence of embedded grids requires information to be restricted within each cell. The model is usually implemented along lines perpendicular to the surface. With an embedded mesh, continuous normal mesh lines do not necessarily exist because of interrupting interfaces (Fig. 1), and, clearly, the models cannot be directly applied along lines such as AB. A method capable of dealing with adaptive meshes must be devised. It should be general enough to treat arbitrarily complex geometries. Furthermore, the computing time and memory required for the application must be small compared to the overall computation expense.

A. Wall and Wake Turbulence Stations

To evaluate the variables that characterize the entire shear-layer profile at each streamwise location, the cells are arranged in streamwise stations, as illustrated in Fig. 2. The algorithm employs *equation adaptation*[3] during the computation process. The viscous regions are detected and the solver employs the full Navier-Stokes equations over those regions, whereas

Received July 26, 1990; revision received Oct. 27, 1990; accepted for publication Oct. 30, 1990.

*Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics.

the simpler inviscid flow (Euler) equations are applied everywhere else. The detection criterion is based on the magnitude of the viscous terms in each station cell. A cell is considered to be within the viscous region if the ratio of the magnitude of the viscous terms for that cell to the maximum magnitude over a station is $> 10^{-4}$. This number was empirically determined and was found to give a conservative estimate (overpredict) of the extent of the viscous region. It should be noted that the exact thickness of the boundary layer need not be found since the purpose of viscous region detection here is to save storage by limiting creating of turbulence mesh stations only within a certain region.

The stations consist of cells from the initial mesh plus those cells that are introduced by the first and second embedding levels. The streamwise variation of the turbulence model quantities was monitored, and it was found that creation of additional stations during a third level of embedding was not necessary for the cases considered. During the higher than second level refinements, no additional stations are created and the new cells are assigned to the previously created stations, which simplifies the necessary data structure and reduces the required storage.

An approach that has been used is employment of the initial, structured, coarse mesh in order to evaluate the model quantities, and then the eddy viscosity values are interpolated to the rest of the nodes.[7] Accuracy of turbulence model quantitites evaluation is limited to accuracy of the initial coarse mesh. Another approach has been an independently created structured mesh.[9] This mesh should follow the refinement level of the adaptive mesh in order to be accurate. An elaborate procedure is used to match the resolution level of the structured turbulence mesh to that of the unstructured grid. An interpolation procedure is required in both cases in order to transfer information between the two types of meshes. Additional interpolation transfer coefficients have to be stored, resulting in significant memory requirements. The present approach does not employ an additional structured mesh. The turbulence stations cells coincide with the actual mesh cells, which results in relatively simple data structures and avoids complicated interpolations between different meshes. The turbulence station cells are updated automatically after each adaptation, and separate grid generation algorithms for the turbulence mesh are avoided. Regions of large turbulence model quantities variations are usually regions of high flow gradients, as well. Coincidence of the turbulence mesh with the adaptive mesh results in higher resolution of those areas since the adaptive algorithm places higher resolution over regions with large flow variations.

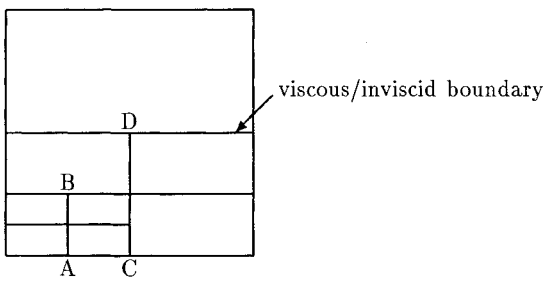


Fig. 1 Interruption of normal lines by grid interfaces.

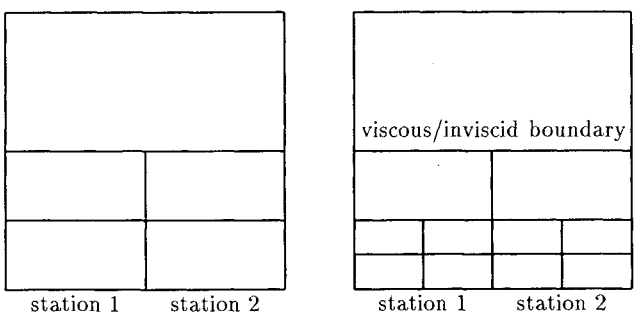


Fig. 2 Cells grouped by stations.

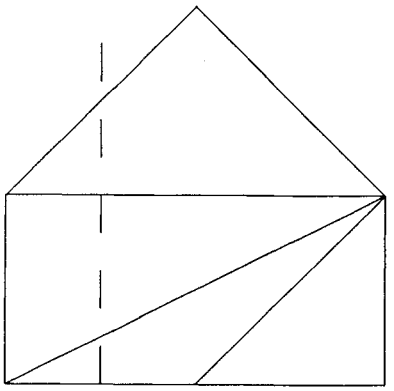


Fig. 3 Triangles arranged in streamwise stations.

This procedure can be applied to triangular meshes as well, in a way similar to the application for embedded quadrilateral meshes. The triangles are arranged in streamwise stations, as shown in Fig. 3. The normal to the wall at each wall face is considered, and the triangles that are intersected by this normal are assigned to the same streamwise station. Such an approach has been independently introduced in Ref. 8. Cells that are not intersected by a normal are assigned to the station with a normal that is closer to the centroid of that cell. Cases for which two neighboring normals intersect (e.g., for a concave surface) usually are not encountered since creation of stations is limited to within a small region close to the surfaces.

The treatment for wakes is similar to this procedure with few modifications. The algorithm detects the wake line position by finding the minimum velocity cell among the cells consisting of a pair of turbulence wake stations with the same streamwise location. An example of such a wake line is shown in Fig. 4a. The wake stations are arranged in pairs formed from upper and lower parts (Figs. 4b and 4c). The wake line location is updated every certain number of iterations (a typical value was 500). Then cells migrate from one station to its counterpart so that those that are above the minimum velocity cell are assigned to the upper station, and the remaining are assigned to the lower station of the pair. The normal distances of the cells from the center of the wake then are modified by simply adding or subtracting the normal distance over which the wake moved from the stored normal distances for each cell.

B. Data Structure

Use of a suitable pointer system for storing and accessing the relevant information is required. The implementation method was designed so that a minimum number of pointers is employed for its applicaiton. The main pointer is an array that lists the cells that belong to each turbulence station. Its form is ITURBCEL (ist, j) pointing to the cell that belongs to turbulence station (ist), and that is the jth within that station. An auxiliary pointer is NNORMCEL (ist), which gives the number of cells that belong to station (ist). The ITURBCEL array is constructed so that the cells of each station are listed in spatial order of increasing distance from the wall or wake. This is useful when station operations such as smoothing of the Baldwin-Lomax function F (Sec. III.B.) are performed. Also, the stations (ist) are arranged in streamwise order in order to facilitate any turbulence model's application that requires upstream or downstream information.

Finally, the normal distance of each cell (ic) from either the wall or the wake line is stored in the array YN (ic), which is updated after each grid adaptation or wake line position change. Following cell subdivision, new normal distances are computed and stored for the "children" cells. The new distances are evaluated from the old ones by simply adding or subtracting the distances along the direction of the normal between the centroid of the original "parent" cell and the centroids of the children cells. In cases of wake movement, the previously stored distances are modified by an amount equal to the motion of the wake line along the direction of the normal.

III. Application of an Algebraic Turbulence Model

The implementation approach is now employed for the Baldwin-Lomax[10] algebraic model, which is one of the most commonly used models. Modifications to application of the original model are also suggested.

A. Model

The algebraic turbulence model developed by Baldwin and Lomax[10] is a two-layer mixing length-type model. In the inner layer, the normalized eddy viscosity is given by

$$\mu_{t_i} = Re\rho l^2 |\omega| \tag{1}$$

where ρ is the density, ω the vorticity, and l the mixing length, and it is expressed as

$$l = \kappa y\left(1 - \exp^{\frac{-y^+}{A^+}}\right) \tag{2}$$

with $1 - \exp^{\frac{-y^+}{A^+}}$ being Van Driest's damping factor. The normal to the wall length y^+ is given by

$$y^+ = Re\,\frac{u_\tau y}{\nu_w}$$

with

$$u_\tau = \frac{1}{\sqrt{Re}}\sqrt{\frac{|\tau_w|}{\rho_w}}$$

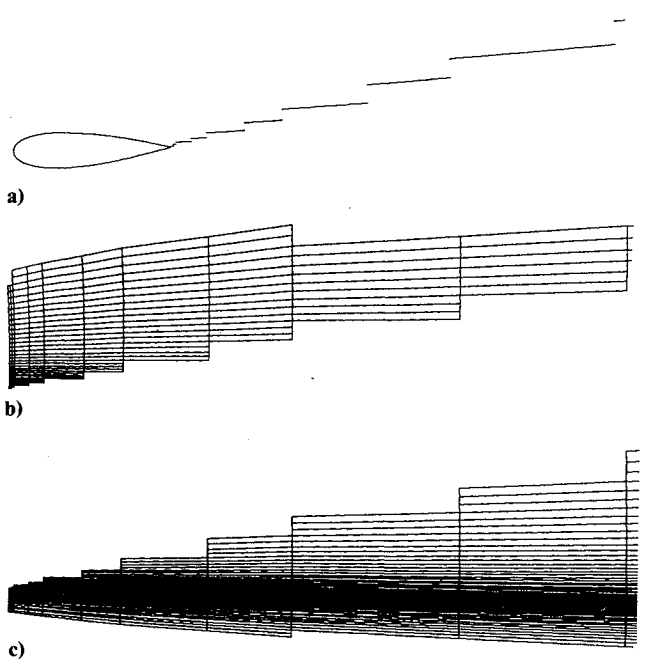


Fig. 4 Wake cells arranged in upper and lower stations (vertical scales enlarged): a) wake line; b) cells forming upper wake stations; c) cells forming lower wake stations.

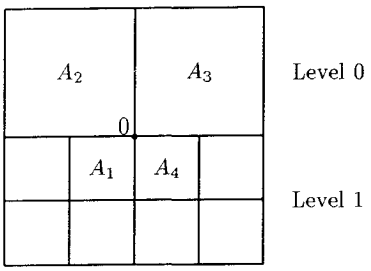


Fig. 5 Interpolation of eddy viscosity values from cell centers to node 0.

being the friction velocity. The subscript w implies quantities evaluated at the wall.

The eddy viscosity of the outer layer is given by

$$\mu_{t_o} = Re\ 0.0168\ C_{CP}\,\rho\,F_{\text{wake}}\,F_{\text{Kleb}} \tag{3}$$

where

$$F_{\text{wake}} = \min\left\{y_{\max}\,F_{\max},\ C_{wk}\,y_{\max}\,\frac{u_{\text{dif}}^2}{F_{\max}}\right\} \tag{4}$$

$F_{\max}$ is the maximum of the function

$$F = y\,|\omega|\left(1 - \exp^{\frac{-y^+}{A^+}}\right) \tag{5}$$

which occurs at $y = y_{\max}$. This function is very important because it yields the length and velocity scales of the model's outer layer. Also u_{dif} is the difference between the maximum and the minimum total velocity within a shear layer profile. Finally, the Klebanoff intermittency function F_{Kleb} is given by

$$F_{\text{Kleb}} = \frac{1}{1 + 5.5(C_{\text{Kleb}}y/y_{max})^6} \tag{6}$$

B. Modifications to the Original Model

The main problem that was encountered as far as application of the model is concerned relates to the model function F, which is employed in order to evaluate the length scale for the outer layer formulation. This length is used in order to scale Klebanoff's intermittency function. A small value of the length results in premature switching on of the outer formulation, which yeilds low values of eddy viscosity coefficient. At separation, the function F exhibits two distinct peaks, with the outermost one being considered by the algorithm.

The model function F is proportional to vorticity, which is a derivative quantity, and, therefore, tends to be noisy. That is, a small kink in the velocity shear-layer profile corresponds to a peak in function F. As a consequence, the model may use this false peak to determine the length scale. In the present work, an algorithm that ignores fictitious peaks has been used. The differences in F values $dF = |Fj - Fj-1|$ between neighboring cells are monitored. In cases in which $dF > dF_{\max}$, where $dF_{\max} = \alpha \bar{F}_{\max}$, the peak is ignored in the search for the maximum in F. In this equation, α is an empirical constant with value equal to 0.25. Its value was determined after monitoring different F function distributions. $\bar{F}_{\max}$ is the maximum of the initial distribution of F. Turbulence mesh stations are created only within the viscous region, and, therefore, the search for the maximum in F is restricted to within the shear layer ignoring any maxima that the function may exhibit in the inviscid region.

The wake model employs the outer layer formulation [Eq. (3)] over the entire wake profile since there is no wall to induce damping to the turbulent stresses. However, switching abruptly between the inner and the outer layer formulation at the trailing edge is both numerically unstable and nonphysical. It may cause numerical instabilities since the eddy viscosity

coefficient changes abruptly at the trailing edge. At the trailing edge node, it is μ_{t_i}, whereas at the immediate node downstream, it is μ_{t_o}. This switching is nonphysical because turbulent stresses cannot change instantaneously along a streamline. The approach that is employed in the present work is to switch off the inner layer formulation at the wake gradually over a distance $X^* = 5\delta_{te}$ away from the trailing edge, with δ_{te} being the maximum of the thicknesses of the upper and lower surface shear layers at the trailing edge.

A sinusoidal switch is used as follows:

$$\mu_t = \left\{ -\sin\left[\frac{\pi}{2X^*}\left(X - X_{te} - X^*\right)\right]\right\}\mu_{t_i} + \left\{1 + \sin\left[\frac{\pi}{2X^*}\left(X - X_{te} - X^*\right)\right]\right\}\mu_{t_o} \tag{7}$$

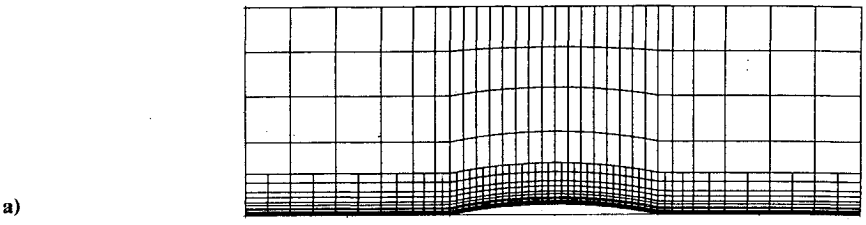
In Eq. (7), X is the x location of the specific wake profile and X_{te} is the trailing-edge node location. The same smoothing formula is applied to the outer layer length and velocity scales in order to avoid abrupt change in these scales when going from the airfoil surface to the wake.

C. Model with Embedded Grids

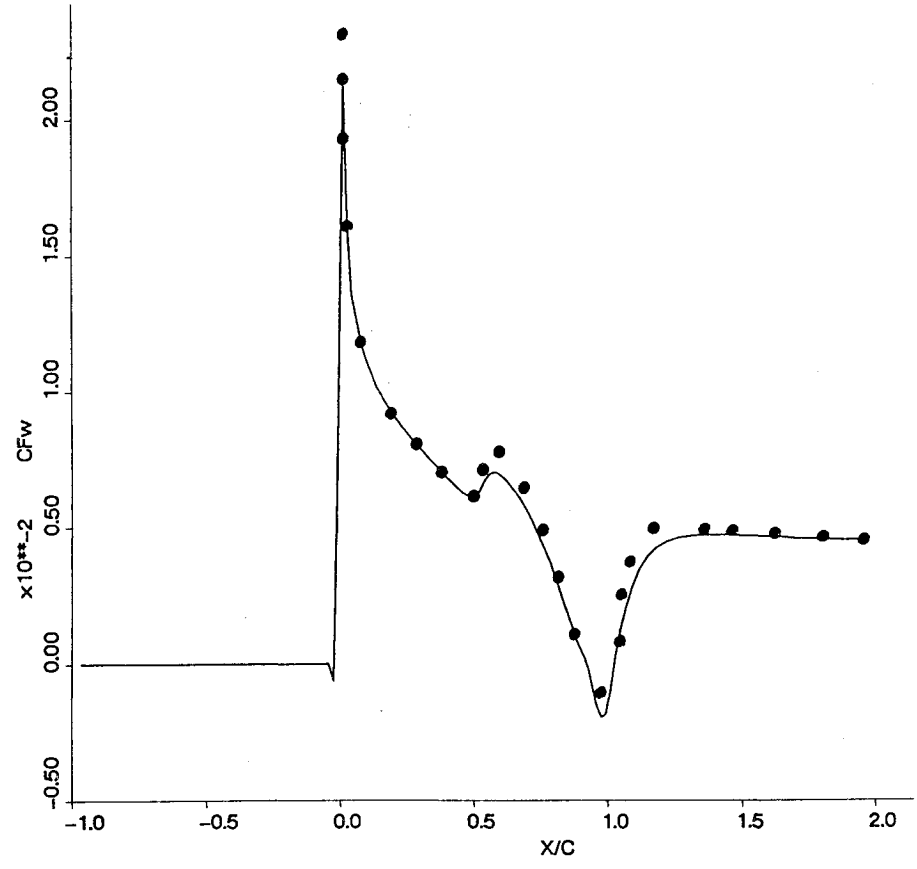
Vorticity ω and the distance y from the center of the cell to the wall are both local cell-based quantities. The friction velocity u_τ, as well as the model parameters $y_{\max}$, $F_{\max}$, u_{dif}, characterize an entire profile at a certain streamwise location. The present approach is to apply the model in a cell-wise manner based on parameters that are known at the center of each cell. This is consistent with the overall approach when dealing with unstructured meshes; i.e., not using information from outside of each cell. For example, vorticity is calculated using Green's theorem over each cell:

$$\omega = -\frac{1}{S_{\text{cell}}}\oint_{\text{cell}} (u\,dx + v\,dy)$$

where S_{cell} is the cell area. The station quantities $y_{\max}$, $F_{\max}$ are calculated by scanning through all cells that belong to each station. The model function F is formed for each cell, and its maximum value $F_{\max}$, which occurs at the station cell with distance from the wall equal to $y_{\max}$, is found. The wall quantity u_τ is evaluated as average from the station cells that are adjacent to the wall. Finally, the eddy viscosity values that are



a)



b)

Fig. 6 Turbulent flow model case—5% bump in a channel ($M_\infty = 0.5$, $Re = 10^5$): a) two-level embedded mesh; b) comparison of wall-shear distributions; —— = two-level embedded mesh and globally-fine mesh; • = Ref. 12.

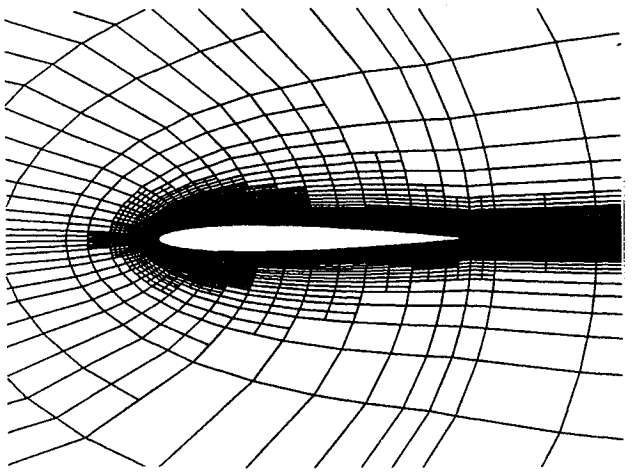


Fig. 7 Two-level embedded grid—subsonic NACA 0012, horizontal scale enlarged ($M_\infty = 0.5$, $Re = 2.91 \times 10^6$, $\alpha = 1.77$ deg).

evaluated at the cell centers are interpolated to the nodes. This is accomplished by using a weighted interpolation. At each node (e.g., 0 in Fig. 5), the eddy viscosity value is evaluated employing the corresponding values at the centers of the surrounding cells according to the formula:

$$\mu_{t0} = \left(\frac{A-A_1}{3A}\right)\mu_{t1} + \left(\frac{A-A_2}{3A}\right)\mu_{t2} + \left(\frac{A-A_3}{3A}\right)\mu_{t3} + \left(\frac{A-A_4}{3A}\right)\mu_{t4} \tag{8}$$

where A_1, A_2, A_3, A_4 are surrounding cell areas and A is their sum. This interpolation formula reduces to linear interpolation in one dimension.

IV. Evaluation of Method of Implementation

The previously described algebraic turbulence model implementation on embedded grids is now evaluated in terms of accuracy, computing time, and memory requirements. An explicit, finite volume, Lax-Wendroff-type numerical scheme was used for discretization of the convective terms of the Navier-Stokes equations.[11] The scheme was extended to include viscous terms as well.[3] A first step evaluates the stress and heat-conduction terms (first-order derivatives), whereas a second step evaluates the viscous terms (second-order derivatives) in a manner similar to treatment of the inviscid terms.

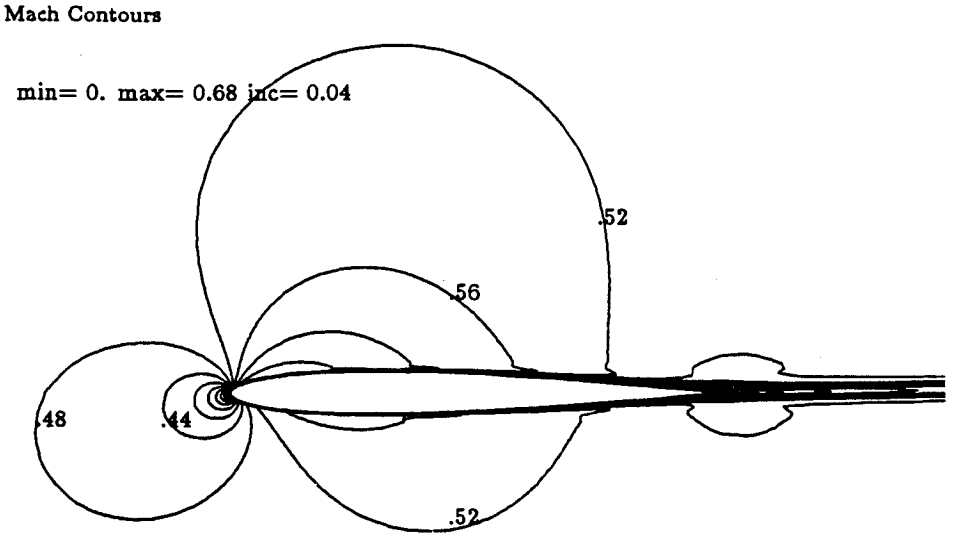


Fig. 8 Flowfield around a subsonic NACA 0012, horizontal scale enlarged ($M_\infty = 0.5$, $Re = 2.91 \times 10^6$, $\alpha = 1.77$ deg).

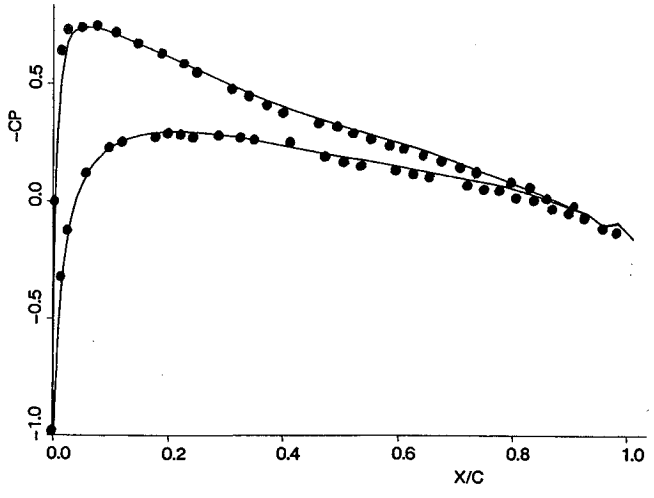


Fig. 9 Comparison of pressure coefficient distributions—subsonic NACA 0012 ($M_\infty = 0.5$, $Re = 2.91 \times 10^6$, $\alpha = 1.77$ deg): —— = embedded mesh and globally fine mesh; • = experiment.[13]

A. Accuracy

The cell quantities (including the model function F) are calculated with the locally finest cell accuracy that is achieved by the solver. However, the station quantities like F_{max}, y_{max}, are calculated over the cells of a station that may include cells with different streamwise locations. The approximation here is that there is no appreciable streamwise variation of these quantities over a station. Monitoring of the streamwise variation of F_{max} was employed in order to decide that creation of turbulence mesh stations during the first two embedding levels was sufficient.

A verification of the approach has been carried out for the test case of subsonic flow ($M_\infty = 0.5$, $Re = 10^5$) for a 5%

bump in a channel. Transition from laminar to turbulent flow was fixed at the middle of the bump. Two levels of embedding were used, as illustrated in Fig. 6a, and comparisons of skin-friction distributions with Ref. 12 are shown in Fig. 6b. The sudden increase in C_f at $X = 0.5$ corresponds to the transition point from laminar to turbulent flow. Another kind of comparison that is valuable is comparison with the results obtained using the same algorithm with the equivalent globally fine, structured mesh. Such a mesh is created by subdividing all of the cells in the domain a number of times equal to the number of embedding levels of the corresponding embedded mesh. It was found that there is no difference in the distributions between the embedded and the globally fine cases.

The turbulence model implementation together with the overall adaptive algorithm are also tested for airfoil flows at high Reynolds numbers. An Alliant FX/8 with three processors (1.5 Mflops speed) was used for the runs. The cases include subsonic and transonic flows, and the geometries consist of both single and two-element airfoils. Specifically, the NACA 0012 airfoil was considered for both subsonic and transonic flow; a two-element airfoil consisting of a main airfoil with a flap was considered in subsonic flow. In all cases, comparisons are made with experimental data. Both angle of attack and Mach number values are those suggested in the experiments to take into account for wind-tunnel wall effects.

1. Single Airfoil Fields

The subsonic flow conditions were $M_\infty = 0.50$, $Re = 2.91 \times 10^6$, $\alpha = 1.77$ deg. An initial C mesh of 33×17 points was employed, with two levels of embedding resulting in a final number of 5225 cells within the domain. The minimum grid normal spacing at the airfoil leading edge was 9×10^{-5} chord lengths, whereas that for the trailing-edge region was 9×10^{-4}. The spacing in the streamwise direction in the leading- and trailing-edge regions was 0.002 and 0.026, respectively. Figure 7 illustrates the embedded grid. The case took 4000 iterations to converge and required 1.8 h. The resulting flowfield is depicted in Fig. 8 in terms of Mach number contour plots. The boundary layers are attached and the adverse pressure gradient is mild. The Baldwin-Lomax model is generally accurate for such a case. Therefore, the case is a valuable test for the implementation method since inaccuracies that may appear will be due to adaptation rather than the model itself. The experiment[13] provided pressure distribution data (Fig. 9); the comparison shows very good agreement between numerics and measurement. The computed C_L of 0.192 compares very well with the experimental value of 0.195. Figure 9 also shows that there is no difference in the distributions between the embedded and the globally fine cases.

The transonic flow conditions were $M_\infty = 0.754$, $Re = 3.76 \times 10^6$, $\alpha = 3.02$ deg. An inital C mesh of 65×41 points was applied with the far-field boundary placed at 15 chord lengths away from the airfoil. Three levels of embedding were introduced by the algorithm, resulting in a total number of 40,440 cells; the final grid is illustrated in Fig. 10. The minimum grid normal spacing at the airfoil leading edge is 2×10^{-5} chord lengths, whereas the spacing at the trailing-edge region is 2×10^{-4}. The spacing in the streamwise direction at the leading-edge region is 3×10^{-4}, whereas the corresponding spacing at the trailing edge is 0.004. The case took 5000 iterations to converge and consumed 8.5 h of computing time. Figure 11 illustrates the flowfield in terms of Mach number contours. A shock forms on the suction side at 40% of the chord. The wiggles that appear are the result of odd-even modes. Presence of grid interfaces was found to excite such modes. Higher values of artificial viscosity would have eliminated those modes. However, the solution within the viscous region would

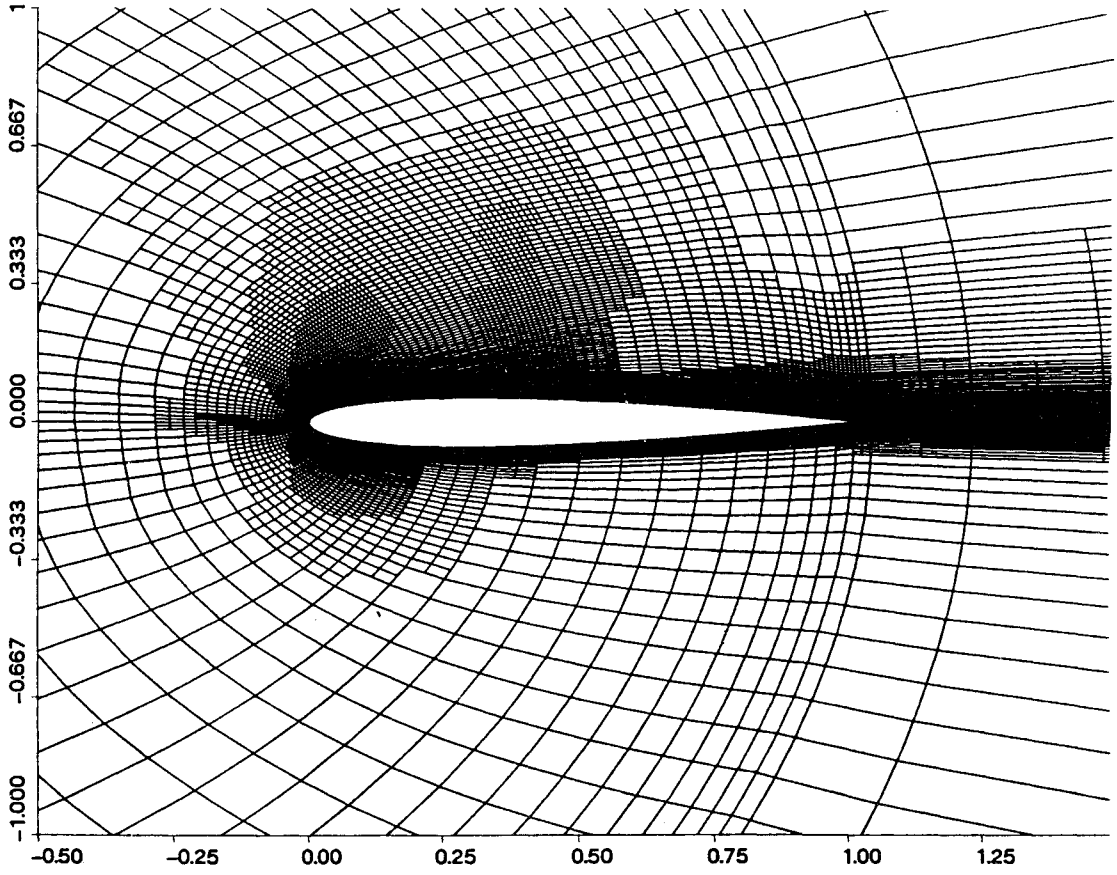


Fig. 10 Final three-level embedded grid—transonic NACA 0012 ($M_\infty = 0.754$, $Re = 3.76 \times 10^6$, $\alpha = 3.02$ deg).

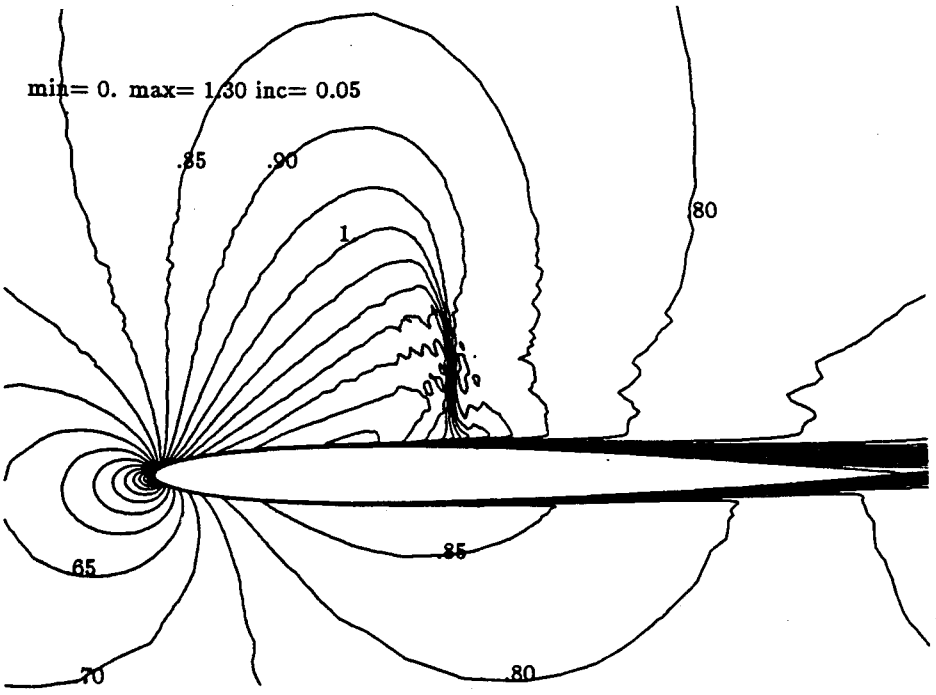


Fig. 11 Flowfield around a transonic NACA 0012 ($M_\infty = 0.754$, $Re = 3.76 \times 10^6$, $\alpha = 3.02$ deg).

have been seriously contaminated.[5] Such oscillations did not affect shock location.

Procedure accuracy may be examined by comparing the experimental surface pressure coefficient distribution with the corresponding numerical result (Fig. 12). The shock location is predicted reasonably accurately but is somewhat smeared. A fourth level of embedding that would provide a more crisp shock was disallowed due to computer limitations. The agreement remains good downstream of the shock. However, as the trailing edge is approached, the boundary layer does not resist the adverse pressure gradient and separates, causing the pressure distribution to tend to level out. Such trailing-edge separation is not observed in the experiment. The algebraic turbulence model that is employed is believed to be largely responsible for this behavior as has been concluded by comparative studies of different turbulence models for transonic airfoils.[14] The pressures on the pressure and suction sides match at the trailing edge, and the somewhat lower pressure level at the suction side influences the pressure side distribution causing it to deviate slightly from the experimental results. The deviation is approximately the same over most of the pressure surface. Unfortunately, corresponding measurements for skin friction were not performed.

2. *Two-Element Airfoil Field*

Generality and flexibility of the implementation method was tested considering the relatively complex geometry of a two-element airfoil. Virtually all numerical results for multielement airfoils originate from panel methods, Euler computations,[15,16] and the viscous-inviscid interaction schemes.[17] Very few numerical simulations of multielement airfoils have used the full Navier-Stokes equations.[5,6,9] The basic airfoil section is a NLR 7301 airfoil[18] with a 32% flap chord, an overlap region of extent $0.053c$ between the main airfoil and the flap, and a gap width of $0.026c$. The experiment considered a flap deflection angle δ set 20 deg down.

The flow conditions were $M_\infty = 0.185$, $Re = 2.51 \times 10^6$, $\alpha = 6.0$ deg. Both laminar and turbulent flow regions were observed during the experiment, and the experimentally measured transition locations on the upper and lower surfaces were assumed by the algorithm since a transition model has not been incorporated into the solver. Specifically, the boundary layer on the main airfoil element suction and pressure sides was assumed to be turbulent downstream of $X = 0.03c$ and $0.65c$, respectively. The flap pressure side flow remained laminar, and the suction side became turbulent at a distance of $0.20c$ downstream of the flap leading edge.

An initial H grid of 77×103 points was employed with two levels of embedding and a final 50,185 cells over the entire domain. Figure 13 shows detail of the embedded grid at the region between the main airfoil and the flap. The minimum grid normal spacing at the leading edges of both elements is 10^{-4} chord lengths. The average y^+ distribution on the sur-

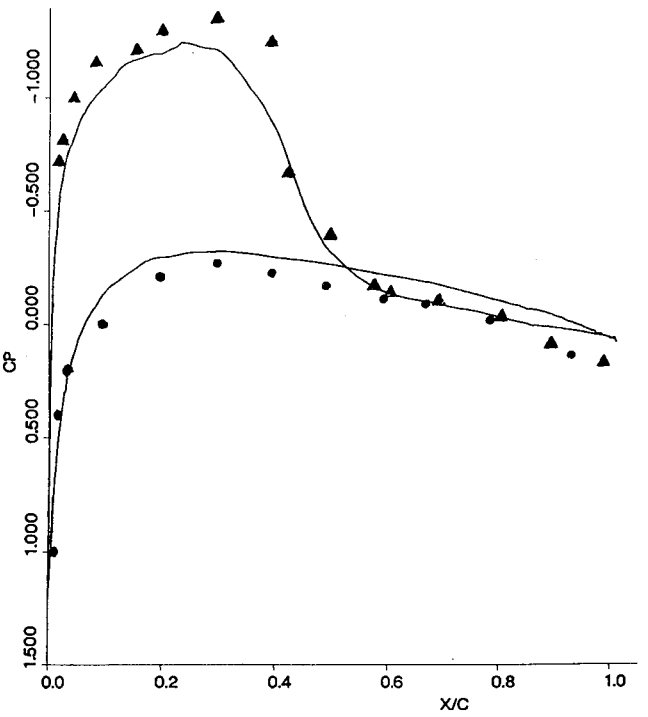


Fig. 12 Comparison of pressure coefficient distribution with experiment—transonic NACA 0012 ($M_\infty = 0.754$, $Re = 3.76 \times 10^6$, $\alpha = 3.02$ deg): —— = embedded mesh; • = pressure side[13]; ▲ = suction side.[13]

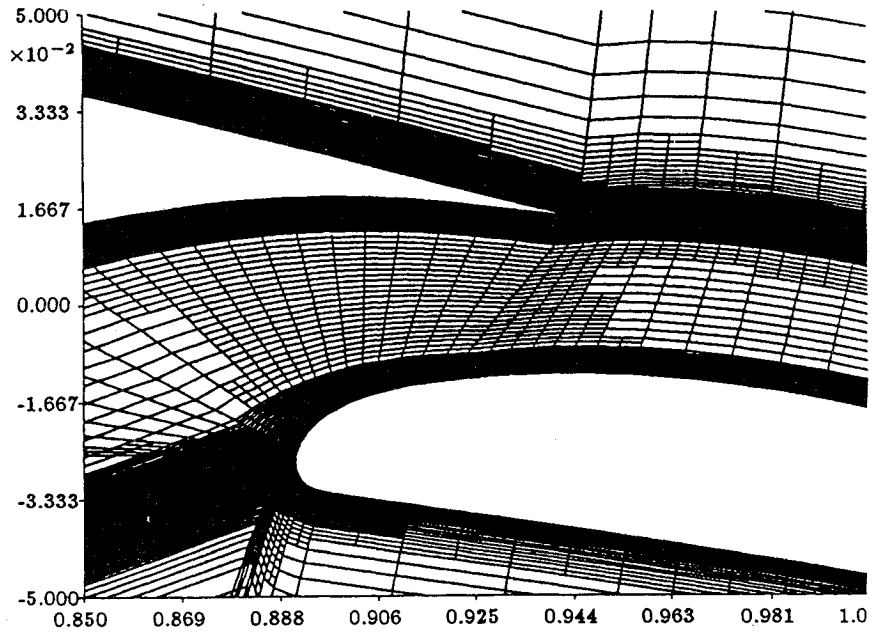


Fig. 13. Detail of the embedded grid between the main airfoil and flap—two-element airfoil ($M_\infty = 0.185$, $Re = 2.51 \times 10^6$, $\alpha = 6.0$ deg, $\delta = 20$ deg).

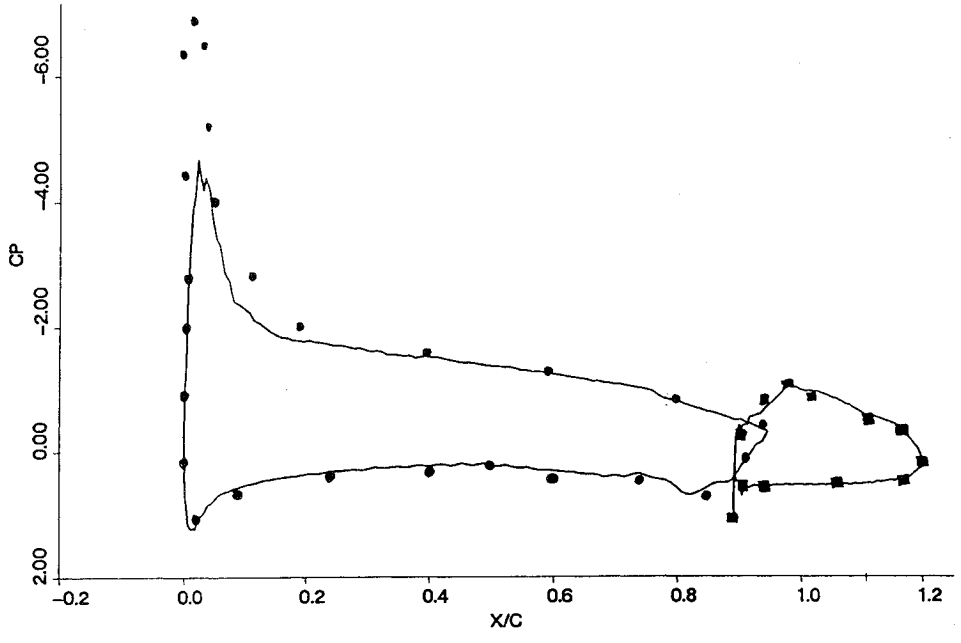


Fig. 14 Pressure coefficient comparison with experimental results—two-element airfoil ($M_\infty = 0.185$, $Re = 2.51 \times 10^6$, $\alpha = 6.0$ deg, $\delta = 20$ deg): —— = present work; •, ■ = experiment.[18]

faces was about 15. Computer limitations did not allow a third level of embedding that would yield smaller values of y^+. The algorithm took 5000 iterations to reduce the residual by three orders of magnitude and required 24 CPU h. The low subsonic freestream Mach number makes computations with an explicit scheme more expensive due to the smaller time-steps constraint.

The wake lines emanating from the trailing edges of the main airfoil and the flap, as well as the wall and wake turbulence stations, are found by the adaptive algorithm in the same way they are found in cases of single airfoils. The difference is that the domain is divided into the main airfoil part and the flap part. Similarly, the turbulence stations are divided into the ones belonging to the main airfoil surface and wake, as well as into the stations belonging to the flap surface and wake. The dividing line is a grid line, which passes in the middle between the gap of the main airfoil and the flap.

Figure 14 shows good agreement between numerical and measured pressure coefficient distributions over both the main airfoil and the flap surfaces, with the exception of the airfoil leading-edge suction peak region. Higher resolution in the streamwise direction is required in order to predict the magnitude of the suction peak accurately. However, a third level would result in more than 150,000 cells in the domain, which was beyond our computer memory capabilities.

B. Computer Requirements

The application of the model does not require additional CPU time compared to the time that is consumed when applying the model on a regular structured mesh. The pointers that

are employed are calculated once following each embedding. Overall, the model, which is applied every five iterations, takes approximately 2% of the total CPU time required by the solver.

Additional storage is required for the pointers that are employed by the implementation procedure. The required memory (Mem1, Mem2, Mem3) for the ITURBCEL, NNORMCEL, YN arrays is given by

$$\text{Mem1} = \text{stations} \times \text{cells/station} \times 1I \tag{9a}$$

$$\text{Mem2} = \text{stations} \times 1I \tag{9b}$$

$$\text{Mem3} = \text{cells} \times 1R \tag{9c}$$

where I and R denote integer and real numbers, respectively. The additional memory that is required is about 3%.

V. Summary

A method of implementing algebraic turbulence models with unstructured, embedded grids was developed. The grid cells within the turbulent flow regions are arranged in stations in order to facilitate evaluation of model quantities that require entire shear-layer profile information. The wake is adapted during the solution process by tracking its position and rearranging the turbulence stations accordingly. The method is general enough to treat relatively complex geometries, such as two-element airfoils.

The Baldwin-Lomax model was chosen in order to test the implementation. Modifications to the application of the original model on structured grids are suggested. The model smoothly transitions from the wall to the wake formulation at the trailing-edge region. Fictitious peaks of the model function F are ignored.

Comparisons with experiments, with another numerical simulation, and with equivalent globally fine mesh results evaluate accuracy of the turbulent flow computations. Accuracy of the presented solutions depends not only on the implementation method but on the adaptive solver and on the Baldwin-Lomax model as well. Both the solver and the model were found to be accurate for the bump, as well as the subsonic NACA 0012 cases. Therefore, those two cases are good indicators of accuracy of the implementation method. The computer requirements of the implementation method in terms of computing time and memory are minimal.

Acknowledgment

This work was supported by the Air Force Office of Scientific Research, where L. Sakell was the technical monitor. The author would like to thank J. Brown of the Department of Aeronautics and Astronautics at the Massachusetts Institute of Technology for his helpful discussions and comments.

References

[1]Berger, M., and Jameson, A., "Automatic Adaptive Grid Refinement for the Euler Equations," *AIAA Journal,* Vol. 23, No. 4, pp. 561-568.

[2]Dannenhoffer, J. F., III, and Baron, J. R., "Grid Adaptation for the 2-D Euler Equations," AIAA Paper 85-0484, Jan. 1985.

[3]Kallinderis, Y., and Baron, J. R., "Adaptation Methods for a New Navier-Stokes Algorithm," *AIAA Journal,* Vol. 27, No. 1, 1989, pp. 37-43.

[4]Kallinderis, Y., and Baron, J. R., "Unsteady and Turbulent Flow Using Adaptation Methods," *Proceedings of the 11th Conference on Numerical Methods in Fluid Dynamics,* edited by D. Dwoyer, M. Y. Hussaini, and R. G. Voigt, Springer-Verlag, Berlin, 1989, p. 326.

[5]Kallinderis, Y., "Adaptation Methods for Viscous Flows," Ph.D. Dissertation, Dept. of Aeronautics and Astronautics, Massachusetts Inst. of Technology, Rept. CFDL-TR-89-5, Cambridge, MA, May 1989.

[6]Kallinderis, Y., and Baron, J. R., "Application of an Adaptive Algorithm to Single and Two-element Airfoils in Turbulent Flow," AIAA Paper 90-0698, Jan. 1990.

[7]Davis, R. L., and Dannenhoffer, J. F., "Adaptive Grid Embedding Navier-Stokes Technique for Cascade Flows," AIAA Paper 89-0204, Jan. 1989.

[8]Rostand, P., "Algebraic Turbulence Models for the Computation of 2-D High Speed Flows Using Unstructured Grids," Inst. for Computer Applications in Science and Engineering, Rept. 88-63, Nov. 1988.

[9]Mavriplis, D. J., "Algebraic Turbulence Modeling for Unstructured and Adaptive Meshes," AIAA Paper 90-1653, June 1990.

[10]Baldwin, B. S., and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Flows," AIAA Paper 78-257, 1978.

[11]Ni, R. H., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal,* Vol. 20, No. 11, 1982, pp. 1565-1571.

[12]Davis, R. L., Ni, R. H., and Carter, J. E., "Cascade Viscous Flow Analysis Using the Navier-Stokes Equations, AIAA Paper 86-0033, Jan. 1986.

[13]Thibert, J. J., Granjacques, M., and Ohman, L. H., "NACA 0012 Airfoil," AGARD AR 138, 1979.

[14]Holst, T. L., "Viscous Transonic Airfoil Workshop—Compendium of Results," AIAA Paper 87-1460, 1987.

[15]Caughey, D., "An Inviscid Analysis of Transonic Slatted Airfoil," *Journal of Aircraft,* Vol. 13, No. 2, 1976, pp. 47-64.

[16]Wigton, L. B., "Application of MACSYMA and Sparse Matrix Technology to Multielement Airfoil Calculations," AIAA Paper 87-1142, June 1987.

[17]Grossman, B., and Volpe, G., "The Viscous Transonic Flow Over Two-Element Airfoil Systems," AIAA Paper 77-688, June 1977.

[18]Van der Berg, B., "Boundary Layer Measurements on a Two-Dimentional Wing with Flap," National Aerospace Lab., The Netherlands, TR-79009 U, Jan. 1979.