Upwind Finite-Volume Navier-Stokes Computations on Unstructured Triangular Meshes

Dartzi Pan* and Jen-Chieh Cheng†
National Cheng Kung University, Tainan, Taiwan, Republic of China

A Navier-Stokes solver using upwind finite-volume method on unstructured triangular meshes is developed and tested. A Godunov-type upwind method is used for inviscid flux computations. An accurate linear reconstruction is used to compute the two Riemann states at the cell face, and Roe's approximate Riemann solver is solved for inviscid fluxes. A finite-volume formulation for the viscous terms, which can be second-order accurate for a large class of triangular cells, is developed and tested. The algebraic Baldwin-Lomax turbulence model is implemented via the use of a turbulence reference grid, the only purpose of which is to provide necessary length scales to the model. All of the turbulence computations are done on the unstructured main mesh without interpolation to and from the reference grid. An approximate lower-upper factorization scheme is used for implicit time integration. Various low and high Reynolds number flows are computed to verify the proposed scheme, including a high Reynolds number flow over a two-element airfoil.

I. Introduction

A LOGRITHMS for solving the Euler and Navier-Stokes equations on unstructured grids have been under intense development in recent years. The use of triangular meshes in two dimensions and tetrahedral meshes in three dimensions was proven very powerful for the inviscid flow over complex geometries.[1-3] For viscous flow computations, however, the hybridization of structured quadrilateral meshes for the viscous region and unstructured meshes for the inviscid regions is generally employed.[4,5] This hybridization approach not only increases the complexity of the code, but also limits the generality of the unstructured grid methodology in dealing with arbitrary complex geometries. Only very recently, the use of highly stretched triangular and tetrahedral meshes for high Reynolds number flows has been successfully demonstrated.[6-8] These computations generally employ a Galerkin finite element-type formulation for the inviscid as well as viscous flux terms. Explicit Lax-Wendroff or Runge-Kutta-type time integrations are usually applied with a multigrid acceleration scheme. These schemes require additional artificial dissipation terms to ensure stability and to capture shocks without severe numerical oscillations.

This paper describes the development of an upwind finite-volume formulation for the Navier-Stokes equations on unstructured triangular meshes. The inviscid fluxes are computed by Roe's approximate Riemann solver at each cell face, assuming a local Riemann problem in the normal direction. A linear reconstruction using an accurate gradient estimation[9] is employed to obtain the two Riemann states at the cell face. For viscous fluxes, a finite-volume formulation for the gradient vector at each cell face is proposed and validated. The formulation can be second-order accurate for a large class of triangular cells, including some highly stretched ones. The algebraic turbulence model of Baldwin and Lomax is incorporated in the solver by using a turbulence reference grid for length scale determination. It is different from Mavriplis's approach[7] in that once the length scale has been determined, all turbulence calculations can be done on the unstructured main grid without interpolation to and from the reference grid. A similar approach for length scale determination was

described independently by Kallinderis[10] for embedded adaptive unstructured meshes. But the rest of the turbulence computations are scheme dependent.

II. Mathematical Model

The full two-dimensional Navier-Stokes equations without body force can be written in integral form as

$$\frac{\partial}{\partial t}\int\int_\Omega Q \ d\Omega + \oint_S F \cdot \hat{n} \ dS = 0 \tag{1}$$

where Ω is the domain of interest and S the boundary surrounding Ω; $\hat{n}$ is the unit normal of S in outward direction; and Q represents the vector of conservative variables. The flux function F contains the inviscid flux vector F^I and the viscous flux vector F^V, that is,

$$F = F^I - \frac{1}{Re_\infty}F^V \tag{2}$$

where Re_∞ is the freestream Reynolds number after normalization. The expressions for Q, F^I, and F^V can be found in many references and will not be repeated here.

The flowfield is discretized using triangular cells to form an unstructured mesh. The average of conservative variables is stored at the center of each triangular cell. The edges of each cell define the faces of the triangular control volume. For each triangular control volume, Eq. (1) can be rewritten as

$$\frac{\partial Q_i V_i}{\partial t} = -\oint_{\partial C_i} F \cdot \hat{n} \ dS \tag{3}$$

where Q_i now represents the averaged conservative variables of cell i, and ∂C_i and V_i denote the boundary cell face and the volume of cell i, respectively. To evaluate the right-hand side of Eq. (3), the flux vectors over each face of ∂C_i are summed, resulting in the discrete form of flux integral as

$$\oint_{\partial C_i} F \cdot \hat{n} \ dS = \sum_{j=k(i)} F_{i,j}\Delta S_j \tag{4}$$

where $k(i)$ is a list of neighboring cells of cell i, $F_{i,j}$ is the numerical flux through the interface of cell i and j, and ΔS_j is the cell-face area. For a cell-centered scheme on triangular meshes, $k(i)$ is a list of three cells.

III. Upwind Inviscid Fluxes

To evaluate the inviscid numerical flux $F_{i,j}^I$, Roe's flux difference scheme is applied locally at each cell face, assuming a local Riemann problem in the normal direction. Two discrete Riemann states separated by the cell interface Q^+ and Q^- are first interpolated using the cell-center values. The inviscid numerical flux $F_{i,j}^I$ is obtained by

$$F_{i,j}^I = \frac{1}{2}[F^I(Q_{i,j}^+) + F^I(Q_{i,j}^-) - |A_{\text{Roe}}|(Q_{i,j}^+ - Q_{i,j}^-)] \quad (5)$$

where A_{Roe} is the flux Jacobian evaluated using Roe's average fluid states. The absolute value symbols indicate that that the absolute value of the eigenvalues were used to evaluate A_{Roe}.

To construct high-order schemes, a linear or higher order distribution of flow variables over each cell is assumed to reconstruct the two Riemann states at each cell face. A piecewise linear reconstruction on cell A can be represented by

$$Q(x, y) = Q_A + \nabla Q_A \cdot r \quad (6)$$

where r is the vector from the cell center A to any point (x, y) in the cell; Q_A is the cell-center value of cell A; and ∇Q_A represents the gradient vector of cell A evaluated using the cell-center values. It is the evaluation of ∇Q that distinguishes various schemes.

By treating Eq. (6) as the first-order Taylor expansion of $Q(x, y)$, it is evident that a first-order accurate ∇Q_A can be solved by three known point values of Q. However, the three known points must not be collinear, that is, they must form a triangle. This is equivalent to assuming that a constant Q gradient exists in the contour enclosed by the three known points, and Gauss theorem can be applied to evaluate the gradient as

$$\nabla Q_A = \frac{1}{a_\Omega} \oint_{\partial as} Q\hat{n} \, dS + \mathcal{O}(\Delta x, \Delta y) \quad (7)$$

where ∂a is the integration path connecting three known points and a_Ω the area contained in the path. It is well known that the Gauss theorem can be applied to arbitrary polygon contours, but only the result of triangular contour is equivalent to the first-order Taylor expansion.

To obtain the gradient of cell A in Fig. 1, it is reasonable to utilize the information at A itself and its neighboring cells B, C, and D. Among four possible integration paths connecting three cell centers, Barth and Jespersen[11] suggested the integration path connecting the neighboring cell centers B, C, and D. Note that in extreme cases where highly stretched and skewed triangles are formed, this integration path may degenerate into a straight line. To avoid this singularity, Barth and Jespersen

further suggested that the set of all cell centers neighboring to and sharing a common vertex with cell A be connected to form the integration path. In this paper, the former path is taken under the assumption that proper cautions have been taken in the grid generation effort to detect and eliminate such cells.

To improve accuracy, a weighted averaging process is introduced[9]

$$\nabla Q_A^{\text{mod}} = (\nabla Q_A + w_B \nabla Q_B + w_C \nabla Q_C + w_D \nabla Q_D)/2 \quad (8)$$

$$w_B = \frac{r_{AC} \times r_{CD}}{r_{BC} \times r_{CD}}, \quad w_C = \frac{r_{AD} \times r_{DB}}{r_{BC} \times r_{CD}}, \quad w_D = \frac{r_{AB} \times r_{BC}}{r_{BC} \times r_{CD}}$$

where $\nabla Q_{B,C,D}$ are Barth and Jespersen's gradient of cell B, C, and D, respectively; r_{AC} is the distance vector from cell center A to C; and other r are defined similarly. Note that the sum of the area weighting factors $w_{B,C,D}$ is always 1. This results in a gradient that has a 10-point stencil support, as shown in Fig. 1. There are two advantages of using this modified gradient. First, on a "regular" triangular mesh made of equilateral triangles, it can be proven that the modified gradient is of the order of $\mathcal{O}(\Delta x^2, \Delta y^2)$. Second, the process of area weighting usually reduces the unevenness in the computed gradients, and the stability of the overall scheme may be enhanced.

IV. Characteristic Limiter

It is known from the experience of structured grid methods that when flow discontinuities are present, the use of characteristic variables in the limiting process, in general, can produce better results than using primitive or conservative variables. In this paper, instead of using the individual cell-face normals, it is chosen to define a unified characteristic direction for the three cell faces of each cell, along which the characteristic variables are defined. The chosen characteristic direction should be able to indicate clearly the pertinent solution gradient, or discontinuity, in the flowfield. Here, the local pressure gradient is used when shock is present in the flowfield. In regions of small pressure gradient, the direction of the maximum solution gradient of primitive variables is used instead.

The limited version of the reconstruction on cell A can be written as

$$Q(x, y) = Q_A + R_p \Phi_A (L_p \nabla Q_A^{\text{mod}} \cdot r), \quad \Phi_A \in [0, 1] \quad (9)$$

where Φ_A is a chosen limiter for cell A and L_p and R_p are the left and right eigenvectors of the flux Jacobian evaluated at the cell center of cell A using the direction of local pressure gradient. If the individual cell-face normals were used instead of a unified characteristic direction, the R_p and L_p in Eq. (9) would be different for different cell faces. A popular limiter following Barth and Jespersen[11] can be written as

$$\Phi_A = \min(\Phi_{Aj}), \quad j = k(i)$$

where Φ_{Aj} is computed at each face j of cell A as

$$\Delta W_{Aj} = L_p (Q_j - Q_A) \quad (10a)$$

$$\Delta W_A^{\min} = \min\{\Delta W_{AB}, \Delta W_{AC}, \Delta W_{AD}, 0\} \quad (10b)$$

$$\Delta W_A^{\max} = \max\{\Delta W_{AB}, \Delta W_{AC}, \Delta W_{AD}, 0\} \quad (10c)$$

$$\Phi_A(\Delta W_{Aj}) = \begin{cases} \min(1, \dfrac{\Delta W_A^{\max}}{\Delta W_{Aj}}), & \text{if} \quad \Delta W_{Aj} > 0 \\[2mm] \min(1, \dfrac{\Delta W_A^{\min}}{\Delta W_{Aj}}), & \text{if} \quad \Delta W_{Aj} < 0 \quad (10d) \\[2mm] 1, & \text{if} \quad \Delta W_{Aj} = 0 \end{cases}$$
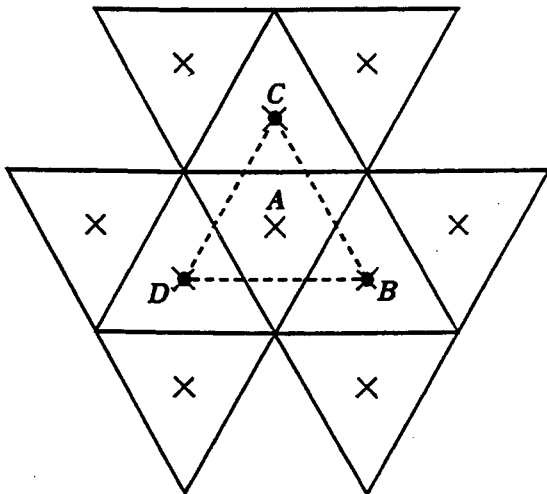
Fig. 1 Stencil support for unmodified gradient ∇Q_A: ●; stencil support of modified gradient ∇Q_A^{mod}: ✕.

where Q_j is the unlimited cell-face value of cell A obtained by $\nabla Q_A^{mod} \cdot r$. It has been our experience that whether the primitive variables or the conservative variables are being reconstructed has very minor effect on the final accuracy of the solution. Since the eigenvector in primitive variables is much simpler than that in conservative variables, it is chosen to use primitive variables in the reconstruction process.

V. Viscous Fluxes

To evaluate the viscous numerical flux $F_{i,j}^V$, it is essential to compute the gradient of velocity and temperature at each cell face. Once the gradient vector at each cell face has been obtained, the computation of viscous fluxes is straightforward. Since the cell gradient ∇Q_A^{mod} has been evaluated for each cell, an obvious approach is to assert that the cell-face gradient is just the average of the cell gradients at the two cells that contain the cell face. Tests on flat-plate boundary-layer flow done by the authors have shown that the accuracy of this approach is only marginal. A more accurate gradient estimation at cell face PQ in Fig. 2a, for example, is to utilize the region enclosed by the two neighboring cell centers A and B and the two end vertexes P and Q. The flow variables at each vertex are obtained by the weighted average of the cell-center values surrounding that particular vertex. For example, the flow variables at vertex P are the distance-weighted average of cell-center values of A, D, E, F, G, and B. A gradient vector ∇Q_{APQ} can be obtained by the triangle APQ, and similarly, ∇Q_{BQP} by BQP, using the Gauss theorem as in Eq. (7). Then, the gradient at the midpoint of cell face PQ can be obtained by

$$\nabla Q_{PQ} = w_B \nabla Q_{APQ} + w_A \nabla Q_{BQP} \qquad (11)$$

where w_A is the area ratio of triangle APQ to quadrilateral $APBQ$, and w_B is defined similarly. It can be proven that, assuming accurate variable values at A, B, P, and Q, Eq. (11) is second-order accurate for a parallelogram $APBQ$ in which line segment AB passes through the midpoint of PQ. For a general quadrilateral $APBQ$ in which AB does not pass through the midpoint of PQ, however, Eq. (11) is only first-order accurate.

For a triangle on the boundary as shown in Fig. 2b, the gradient at the boundary cell face PQ is evaluated by the triangular contour APQ only. The use of vertex values makes it easy to implement solid wall boundary conditions. The velocity is zero at P and Q by no-slip condition. The temperature is either given at P and Q for an isothermal condition, or the temperature gradient normal to the wall PQ is known with known heat flux. For highly stretched boundary triangles, however, it is important to construct triangles with equidistant AP and AQ such that the estimated gradient can better represent the flow gradient at the midpoint of the boundary face.
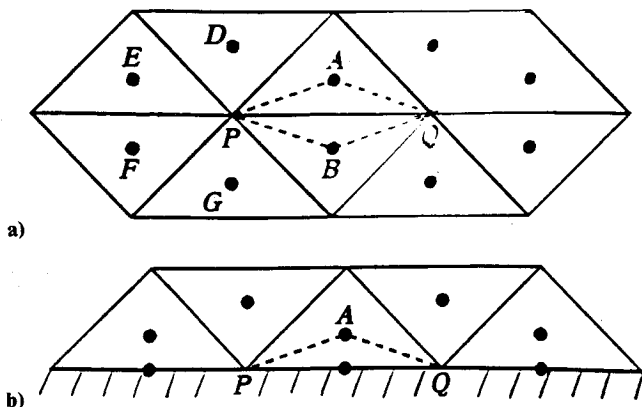
VI. Turbulence Model

The Baldwin and Lomax algebraic turbulence model has been implemented in the solver. A turbulence reference grid is used to supply the necessary length scale for each cell. The current approach differs from the approach of Mavriplis.[7] In the latter approach, the flow variables are first interpolated onto a structured background grid on which all of the turbulence computations are done. Then the computed turbulence eddy viscosity is interpolated back to the unstructured main mesh for the flow solver. In the current approach, all of the turbulence computations are done on the unstructured main mesh, and, hence, no interpolation is needed between the main mesh and the reference grid. The only purpose of the reference grid is to determine necessary length scales needed by the turbulence model. It should be mentioned here that a similar approach for length scale determination was described independently by Kallinderis[10] for embedded adaptive unstructured meshes. But the rest of the turbulence computations are scheme dependent.

The turbulence reference grid is constructed by dividing the computational domain into strips, or narrow bands, which emanate from the body walls or some wake centerlines. The width of each strip should be on the order of one to two cell sizes. The cells whose cell center reside in one particular strip are treated as one group of cells. In each group, the distance of each cell center to the end wall or the wake centerline can be measured, which, in turn, will be used as the reference length in the turbulence model. Once the reference length for each cell is established, the computation of turbulence model can proceed on the unstructured mesh without difficulty. For example, the vorticity at each cell center can be obtained directly from the velocity gradient computed by the flow solver. The peak of a functional distribution can be located by searching the extremum points along each strip or within each group. The turbulence eddy viscosity at each cell center can then be determined easily by the two-layer algebraic model.

Ideally, the divided strips should cover the entire computational domain without overlapping each other. However, for the flowfield with multiple walls or wake lines, regions of overlapping strips will occur. This is equivalent to saying that the turbulence in these regions is influenced by multiple turbulence sources. In this case, the turbulence computation within each group still proceeds as before, but a weighting procedure to determine the influences from each wall or wake line must be included to compute the final turbulence eddy viscosity. In this paper, the simplest weighting is taken such that the turbulence is influenced only by the closest wall or wake line. A more sophisticated approach may take the distance-weighted average as the final turbulence eddy viscosity. Also, instead of using a preset wake centerline, it is not difficult to adopt Mavriplis's approach[7] in which the wake centerline is determined by the minimum velocity or zero vorticity in a strip traversing the entire wake flow.

VII. Time Integration

The semidiscrete form of Eq. (3) can be written as

$$\frac{\partial Q_i V_i}{\partial t} = - \sum_{j=k(i)} F_{i,j} \Delta S_j = -R(Q) \qquad (12)$$

An implicit time discretization can be written as

$$V \frac{\Delta Q^n}{\Delta t} = -[\theta R(Q^{n+1}) + (1 - \theta)R(Q^n)] \qquad (13)$$

where $\Delta Q^n = Q^{n+1} - Q^n$. The scheme is the first-order Euler implicit method for $\theta = 1$ and the second-order trapezoidal method for $\theta = 0.5$. It can be linearized about a known state Q^n to obtain

$$\left[\frac{VI}{\Delta t} + \theta \frac{\partial}{\partial Q} R(Q) \right] \Delta Q^s =$$

$$- \left[\frac{(Q^s - Q^n)}{\Delta t} V + \theta R(Q^s) + (1 - \theta)R(Q^n) \right] = RHS \qquad (14)$$

a)

b)

Fig. 2 Stencil support for gradient vector at: a) cell-face PQ and b) boundary surface PQ.
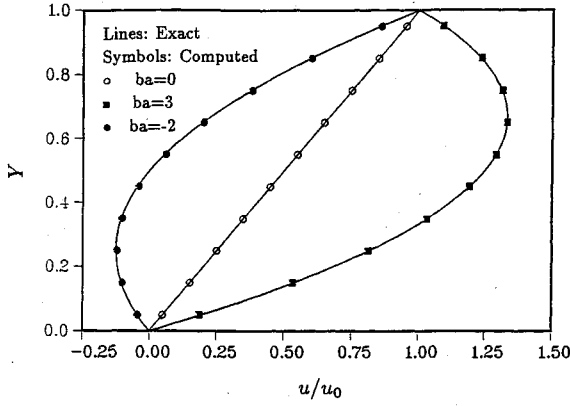
Fig. 3 Couette flow velocity profiles with different pressure gradient; $ba = -(h^2/2\mu u_0)[(dp/dx)]$, $h = 1$, $M_{upper} = 0.1$, and $Re = 5$.
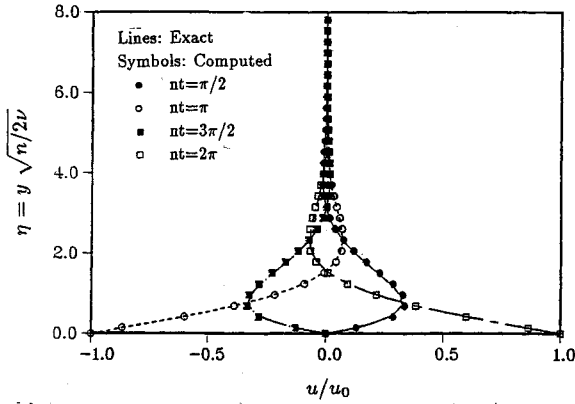

Fig. 4 Unsteady Stoke's second problem, velocity profiles at different times; $u_p = u_0\cos(nt)$, $n = 1$, $M = 0.1$, $Re = 1.5$, and 120 cells.

where $\Delta Q^s = Q^{s+1} - Q^s$ and I is the identity matrix. Here s is a subiteration index which can be subiterated to obtain a better approximation from Q^{s+1} to Q^{n+1}.

To reduce the computational work of the inversion process, two simplifications are made to the left-hand side whereas the full high-order accuracy is kept on the explicit side of Eq. (14). First, only a first-order upwind scheme is used for the inviscid Jacobian on the implicit side. Second, instead of the true viscous Jacobian, a Laplacian-like operator that simulates the incompressible viscous Jacobian is used on the implicit side. This is derived by taking the incompressible approximation of the viscous fluxes as

$$-\frac{1}{Re_\infty}\sum_{j=k(i)} F_{i,j}^V \Delta S_j \simeq -\frac{\mu}{Re_\infty} I_V \sum_{j=k(i)} \nabla Q_{i,j}^V \cdot \hat{n} \Delta S_j$$

$$\simeq -\frac{\mu}{Re_\infty} I_V \sum_{j=k(i)} \frac{r_{j,i} \cdot \hat{n} \Delta S_j}{|r_{j,i}|^2} (Q_j^V - Q_i^V)$$

(15)

where

$$I_V = \text{diag}\left[0, 1, 1, \frac{1}{(\gamma - 1)Pr}\right]$$

is a diagonal matrix with Prandtl number Pr; $Q^V = [0, u, v, a^2]^T$ is a vector of velocity components u and v and the square of the sound speed a^2; and $r_{j,i}$ is the distance vector from cell center i to cell center j. Note that Eq. (15) is Laplacian-like and has a stencil support the same as the first-order upwind inviscid scheme. The Jacobian of this simplified viscous flux is then used on the implicit side of Eq. (14).

The above two simplifications will not alter the time and spatial accuracy of the solution when the subiteration in s

converges. The fully discretized equation can be written in matrix form as

$$(D + L + U)\Delta Q^s = \text{RHS}$$

(16)

where D is a block diagonal matrix, L a block lower-triangular matrix with zero diagonal elements, and U a block upper-triangular matrix with zero diagonal elements. An approximate lower-upper factorization (ALU) scheme originally developed for the structured grid method is adopted to factorize the left-hand side of Eq. (16) as

$$(D + L)D^{-1}(D + U)\Delta Q^s = \text{RHS}$$

(17)

The factorization error of Eq. (17) is $LD^{-1}U\Delta Q^s$. Since a first-order upwind scheme and a Laplacian-like operator are used on the left-hand side, the factors $(D + L)$ and $(D + U)$ are diagonally dominant. The inversion processes are simple because they require only matrix multiplications and additions and 4×4 matrix inversions for two dimensions. To make the ALU scheme fully vectorizable, a black-gray-white (BGW) coloring technique that divides the computational cells into three sets of different colors is implemented. The rule of coloring is that no two adjacent cells have the same color. Since the stencil support for the implicit side of Eq. (17) is simplified, this BGW coloring has completely eliminated the data dependency and allows a full vectorization of the inversion process. The cost of this high degree of vectorization may be some extra memory for the color management and, more seriously, a change in the convergence characteristics because of the change in the sequence of information propagation of boundary conditions.
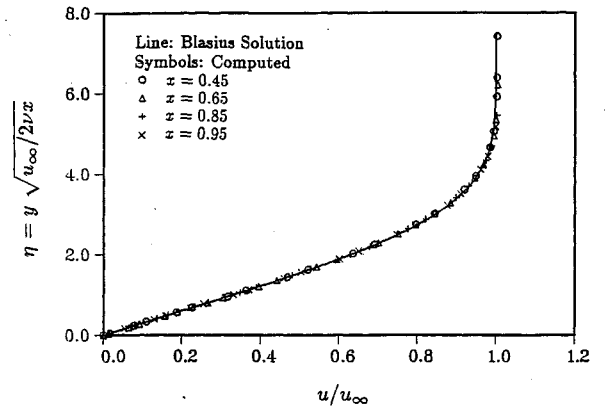

Fig. 5 Laminar flat-plate flow, velocity profile compared with Blasius solution; $M_\infty = 0.5$, $Re = 10^4$, and 3000 cells.
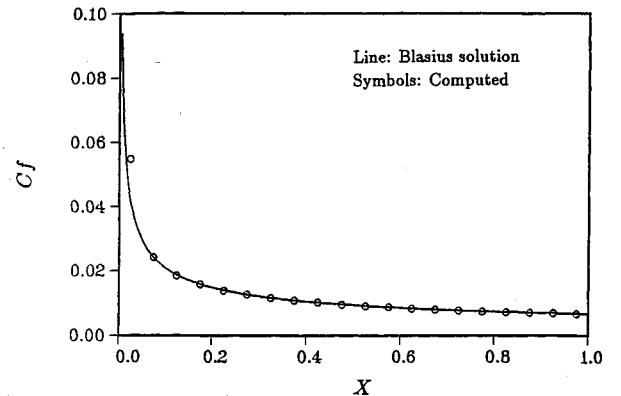

Fig. 6 Laminar flat-plate flow, skin friction C_f compared with Blasius solution; $M_\infty = 0.5$, $Re = 10^4$, and 3000 cells.

VIII.   Validations

A.   Low Reynolds Number Flows

Two low Reynolds number flows with well-known exact solutions (incompressible) are computed to validate the viscous term calculations of the solver. The two-dimensional grid used is obtained by cutting the quadrilateral cells of the structured mesh along the diagonals such that each quadrilateral cell produces four triangles. The first problem is the Couette flow with the upper wall moving at Mach number 0.1. The Reynolds number is 5 based on the channel width and upper wall velocity. No velocity and density gradient is assumed in x boundaries. The velocity profiles with positive, negative, and zero pressure gradient are compared with exact solutions in Fig. 3. The comparison is excellent.

The second problem is the unsteady Stoke's second problem. The bottom wall is oscillating horizontally with a speed $u_p = u_0 \cos(nt)$. It is chosen that $n = 1$; u_0 corresponds to $M = 0.1$; and ν corresponds to Reynolds number 1.5 based on the unit length and u_0. Since there is no gradient in x, the x boundary values are extrapolated from interior cells. A $\theta = 0.5$ time integration is used with the ALU factorization on the implicit side. The time step is chosen to be 0.001, whereas the period of the oscillation is 2π. For such a small time step, two or three subiterations are sufficient to drop the residual to 10^{-6}. The computed velocities at different times for one complete cycle are compared with exact solutions in Fig. 4. In general, good agreement is obtained.

B.   High Reynolds Number Flows

A laminar flat-plate boundary-layer flow is computed using $M_\infty = 0.5$ and $Re_\infty = 10^4$ based on the unit length. A grid of 3000 cells is obtained by cutting a structured mesh along the diagonals such that each quadrilateral produces four triangles. Although this is a quite "structured" triangular mesh, it still
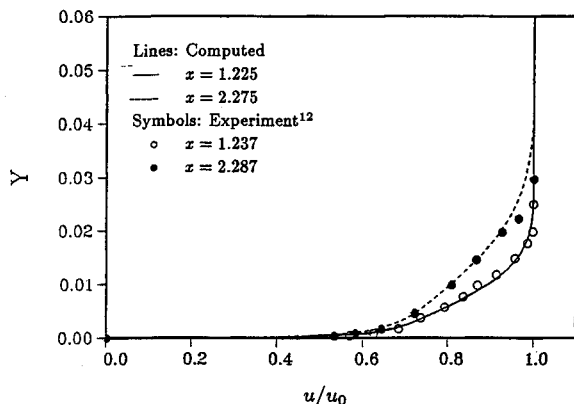
Fig. 7   Turbulent flat-plate flow, velocity profile compared with experimental data; $M_\infty = 0.3$, $Re = 2.2 \times 10^6$, and 10,500 cells.
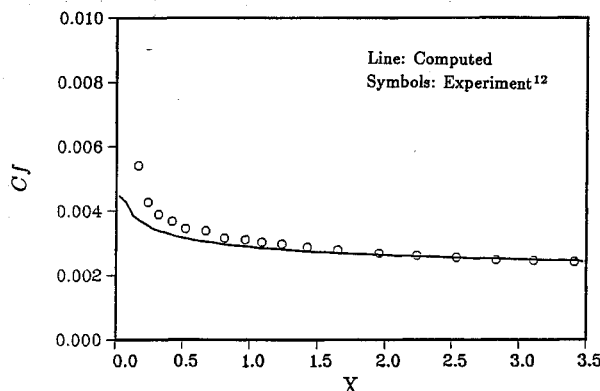
Fig. 8   Turbulent flat-plate flow, skin friction C_f compared with experimental data; $M_\infty = 0.5$, $Re = 2.2 \times 10^6$, and 10,500 cells.
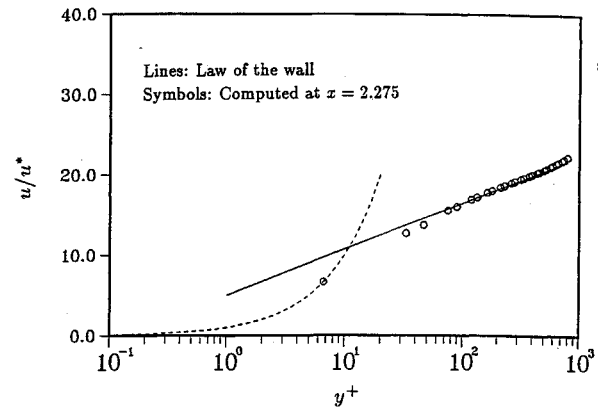
Fig. 9   Turbulent flat-plate flow, comparison of the law of the wall; $M_\infty = 0.3$, $Re = 2.2 \times 10^6$, and 10,500 cells.
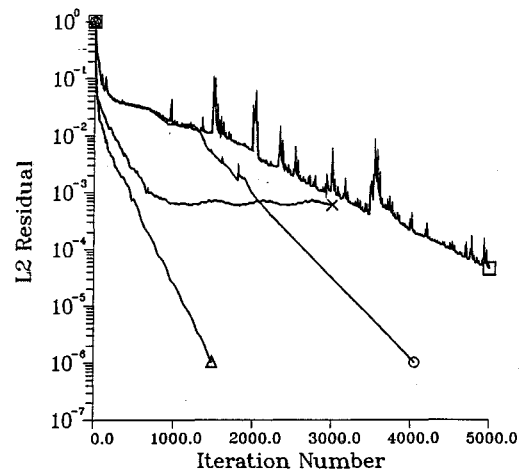
Fig. 10 L_2 residual vs iteration count for flat-plate flow; $\triangle$: laminar ALU, $\times$: laminar ALU + BGW, $\bigcirc$: turbulent ALU, $\square$: turbulent ALU, no viscous Jacobian.

provides a good validation test for the viscous computation on stretched triangles. The cell aspect ratio, defined as the maximum ratio of the height to the base, of the cell adjacent to the plate is 1:50; but for a cell-centered scheme, a more meaningful aspect ratio may be the height of the cell center to the base, which amounts to 1:150 for this case. The boundary conditions used are no slip and zero pressure gradient on the plate, characteristic-like boundary conditions for inflow x and far stream y boundaries and extrapolations for the outflow boundary. The computed velocity profiles at several x stations are plotted in Fig. 5. They all collapse onto a single curve, and the comparison with the Blasius solution is excellent. Figure 6 is the computed friction coefficient along the plate. In general, except near the leading edge, excellent agreement with Blasius solution is obtained.

A turbulent flat-plate boundary-layer flow is also computed to validate the turbulence model. There are 10,500 cells in the grid. The y^+ of the first cell center above the plate is about 6. The cell aspect ratio is 1:350, which amounts to 1:1050 if measured from the cell center. The freestream is at $M_\infty = 0.3$ and $Re_\infty = 2.2 \times 10^6$. The computed velocity profiles at two x stations and the skin friction coefficient along the plate are compared with the experimental data[12] in Figs. 7 and 8. In general, good agreement is obtained except near the leading edge. Figure 9 shows the comparison between the law of the wall and the computed result at $x = 2.275$. The computed inner region velocity profile is reasonably accurate.

The preceding computations (and those which follow) are done on a scalar HP-720 workstation with 32-MB memory, whose speed is rated at 17 Mflops double precision. The
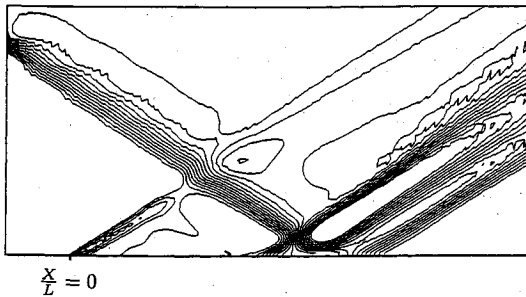
Fig. 11 Pressure contours for laminar shock/boundary-layer interaction; $M_\infty = 2.0$, $\theta_i = -32.6$ deg, $Re_L = 2.96 \times 10^5$, and 9984 cells.
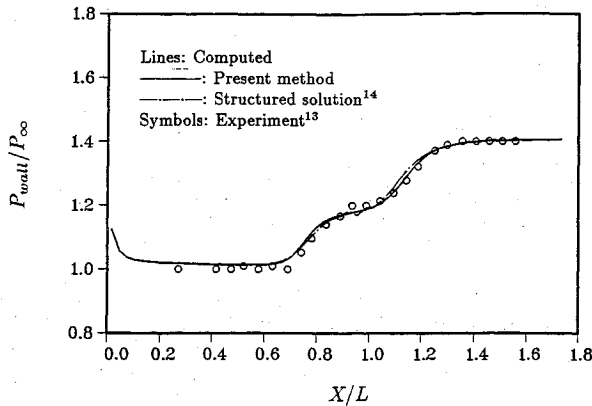

Fig. 12 Wall pressure distribution for laminar shock/boundary-layer interaction; $M_\infty = 2.0$, $\theta_i = -32.6$ deg, $Re_L = 2.96 \times 10^5$, and 9984 cells.

convergence histories are shown in Fig. 10. For the laminar case, the CPU second per iteration per cell is 2.31×10^{-4} for single precision and 2.85×10^{-4} for double precision. It is seen that when BGW coloring is used, the convergence ceases after dropping three orders of magnitude in L_2 residual. It should be mentioned here that the BGW technique was tested by authors[9] on a single Alliant FX-80 vector processor for some inviscid flow problems, and the speedup can be as high as 45% of the CPU time per iteration per cell. But judging from the result in Fig. 10, this speedup in computational efficiency is not enough to balance the deterioration in the convergence rate. For the turbulent case, the CPU second per iteration per cell is 2.56×10^{-4} for single precision and 3.16×10^{-4} for double precision. The memory requirement is 39 words per cell. It is clear that the inclusion of the simplified viscous Jacobian stabilizes and accelerates the convergence. The maximum Courant-Fredricks-Lewy (CFL) number based on the square root of the cell area for this case is about 7.5. If the CFL number is based on the shortest distance from cell center to cell face, it will be much higher for highly stretched triangles. The sudden spikes in the convergence without implicit viscous Jacobian is probably because of too large a CFL number.

C. Shock/Boundary-Layer Interaction

The problem computed is a shock impinging on the laminar boundary layer of a flat plate inducing a separation bubble with downstream reattachment. The conditions calculated are $M_\infty = 2.0$ and $Re_L = 2.96 \times 10^5$ based on the distance from the leading edge of the plate to the point of shock impingement on the plate. The shock impingement angle θ_i is -32.6 deg to the horizontal. Experimental data by Hakkinen et al.[13] are used for comparison. Figure 11 shows the computed pressure contours. The shock produced by the presence of the leading edge, the reflected shocks, and the expansion fan due

to the separation bubble are clear. Figure 12 is the wall pressure distribution plot. For comparison, the results computed by Chakravarthy's total variation diminishing (TVD) scheme[14] are also plotted. The calculated pressure plateau around the impingement point is evident. Figure 13 is the skin friction plot along the plate. The wall pressure prediction is generally in good agreement with the experiment, although, in general, the skin friction is underpredicted. A larger separation bubble is predicted than that revealed by the experiment.

D. Single- and Multielement Airfoils

The turbulent flow over a NACA 0012 airfoil at $M = 0.5$, $Re = 2.39 \times 10^6$, and $\alpha = -0.05$ deg is computed and compared with the experiment.[15] The main mesh near the airfoil is shown in Fig. 14a. The turbulence reference grid is shown in Fig. 14b. There are 12,078 cells in total and 128 cells on the airfoil surface. To generate and cluster highly stretched triangles around the desired places in a truly unstructured manner is a current and difficult research topic. But this is basically a problem for the grid generation, not for the numerical scheme. In this paper, a compromise is made by using a layer of highly stretched viscous grid around the airfoil and the present wake line, which is obtained by cutting each quadrilateral into four triangles. Outside the viscous grid, the unstructured triangular mesh is generated. The maximum cell aspect ratio is about 200, which amounts to about 600 if measured from the cell center. The mesh stretching ratio, defined as the
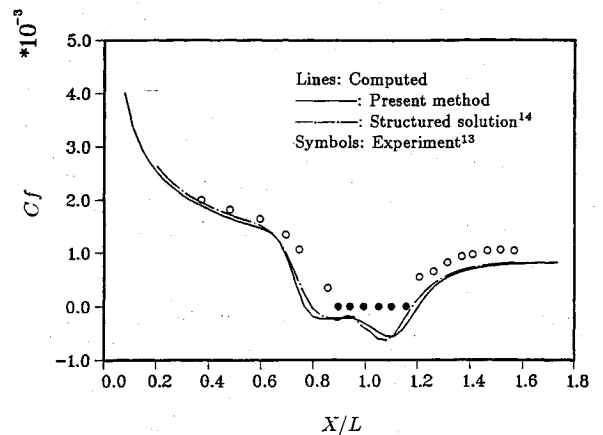

Fig. 13 Skin friction distribution for laminar shock/boundary-layer interaction; $M_\infty = 2.0$, $\theta_i = -32.6$ deg, $Re_L = 2.96 \times 10^5$, and 9984 cells.
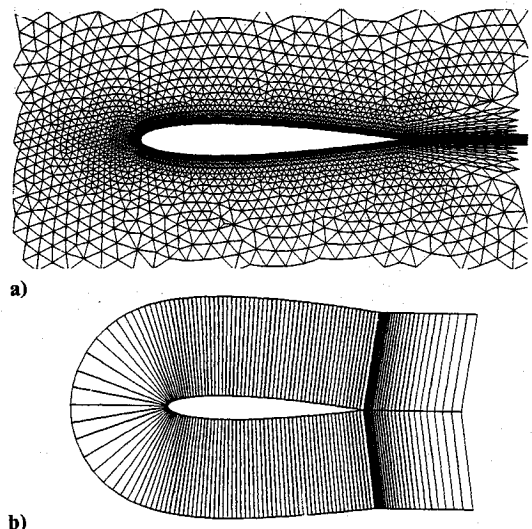

Fig. 14 Grid for turbulent flow over NACA-0012 airfoil: a) mesh near airfoil and b) turbulence reference grid; $M_\infty = 0.5$, $Re = 2.39 \times 10^6$, and 12,078 cells.
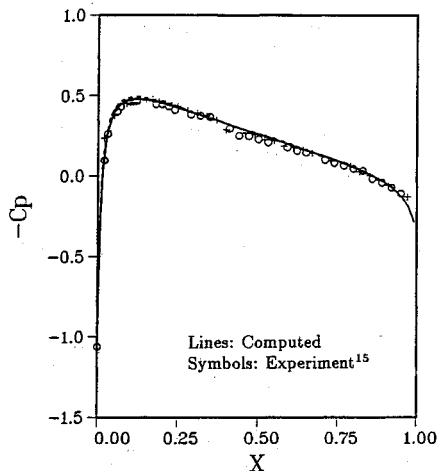
Fig. 15 Pressure coefficient for turbulent flow over NACA-0012 airfoil; $M_\infty = 0.5$, $Re = 2.39 \times 10^6$, and 12,078 cells.
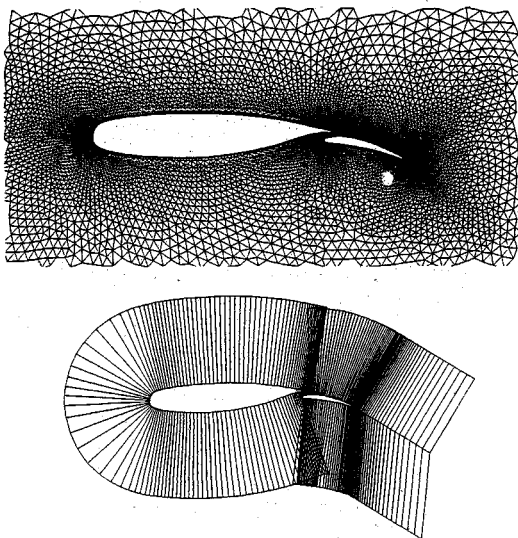

Fig. 16 Grid for turbulent flow over two-element airfoil: a) mesh near airfoil and b) turbulence reference grid; $M_\infty = 0.185$, $Re = 2.62 \times 10^6$, and 21,160 cells.
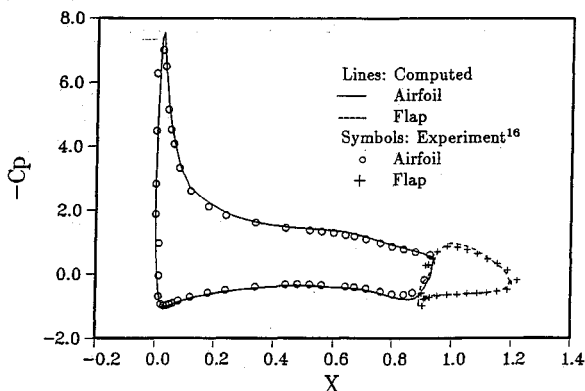

Fig. 17 Pressure coefficient for turbulent flow over two-element airfoil; $M_\infty = 0.185$, $Re = 2.62 \times 10^6$, and 21,160 cells.

square root of volume ratio of the largest cell to the smallest cell, is about 2236. The computed pressure coefficients are shown in Fig. 15. Since α is small, the computed results are almost symmetric on the upper and lower surfaces. The comparison with experiment is good.

Finally, a turbulent flow over the NLR 7301 two-element airfoil at $M = 0.185$, $Re = 2.62 \times 10^6$, and $\alpha = 6$ deg is computed and compared with the experiment.[16] The main mesh
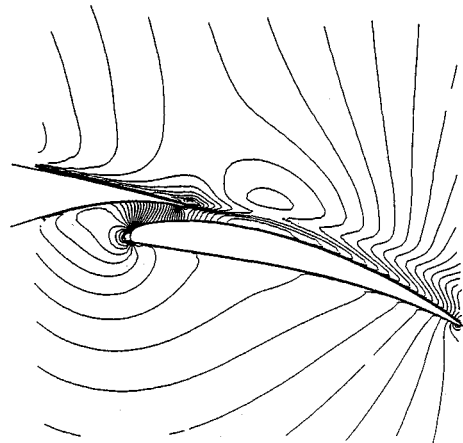
Fig. 18 Mach contours near the flap airfoil, turbulent flow over two-element airfoil, $M_\infty = 0.185$, $Re = 2.62 \times 10^6$, and 21,160 cells.

near the airfoil is shown in Fig. 16a. Again, a layer of viscous grid is used around the airfoil surfaces. The turbulence reference grid with regions of overlapping strips is shown in Fig. 16b. There are 21,078 cells in total, 139 cells on the main airfoil surface, and 99 cells on the flap airfoil surface. The maximum cell aspect ratio is about 100, which amounts to about 300 if measured from the cell center. The mesh stretching ratio is about 8917. The L_2 residual drops three orders of magnitude in about 10,000 steps. The computed pressure coefficients are compared with data in Fig. 17. In general, the comparison is good, with some differences around the flap leading and trailing edge. Figure 18 shows the computed Mach contours near the flap airfoil. The wake flow from the trailing edge of the main airfoil is clear. It will be interesting to further investigate the interactions between the wake flow and the flap flowfield.

IX. Conclusions

An upwind finite-volume scheme for the Navier-Stokes solutions on unstructured triangular meshes has been developed and tested. A Godunov-type upwind method is used for inviscid flux computations. A finite-volume formulation for the viscous terms has been validated by various low and high Reynolds number flows. Accurate laminar and turbulent solutions can be obtained using highly stretched triangles. In general, the convergence rate decreases with decreasing cell aspect ratio. Consequently, the turbulent computation usually requires thousands of ALU integration steps. The simplified viscous Jacobian is helpful to stabilize and sometimes to accelerate the convergence. Finally, the turbulent flow over a two-element airfoil was successfully calculated by the proposed numerical scheme.

X. Acknowledgment

References

[1]Jameson, A., Baker, T. J., and Weatherill, N. P., "Calculation of Inviscid Transonic Flow over a Complete Aircraft," AIAA Paper 86-0103, Jan. 1986.

[2]Mavriplis, D. J., "Three Dimensional Unstructured Multigrid for the Euler Equations," AIAA Paper 91-1549, June 1991.

[3]Frink, N. T., Parikh, P., and Pirzadeh, S., "Aerodynamic Analysis of Complex Configurations Using Unstructured Grids," AIAA Paper 91-3292, Sept. 1991.

[4]Holmes, D. G., and Connel, S. D., "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA Paper 89-1932, June 1989.

[5]Nakahashi, N., "FDM-FEM Zonal Approach for Viscous Flow Computations over Multiple Bodies," AIAA Paper 87-0604, Jan. 1987.

[6]Mavriplis, D. J., Jameson, A., and Martinelli, L., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," AIAA Paper 89-0120, Jan. 1989.

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

[8]Marcum, D. L., and Agarwal, R. K., "A Three-Dimensional Finite Element Navier-Stokes Solver with k-ϵ Turbulence Model for Unstructured Grids," AIAA Paper 90-1652, June 1990.

[9]Pan, D., and Cheng, J.-C., "A Second-Order Upwind Finite-Volume Method for the Euler Solution on Unstructured Triangular Meshes," *International Journal for Numerical Methods in Fluids* (to be published).

[10]Kallinderis, Y., "Algebraic Turbulence Modeling for Adaptive Unstructured Grids," *AIAA Journal*, Vol. 30, No. 3, 1992, pp. 631–639.

[11]Barth, T. J., and Jespersen, D. C., "The Design and Application of Upwind Schemes on Unstructured Meshes," AIAA Paper 89-0336, Jan. 1989.

[12]Weighardt, K., "Flat Plate Flow," *Proceedings of the AFOSR-IFP Stanford Conference on Computation of Turbulent Boundary Layers,* edited by D. E. Coles and E. A. Hirst, Vol. 2, 1968, pp. 98–123.

[13]Hakkinen, R. J., Greber, I., Trilling, L., and Abarbanel, S. S., "The Interaction of an Oblique Shock Wave with a Laminar Boundary Layer," NASA Memo. 2-18-59w, March 1959.

[14]Chakravarthy, S. R., "The Versatility and Reliability of Euler Solvers Based on High-Accuracy TVD Formulations," AIAA Paper 86-0243, Jan. 1986.

[15]Thibert, J. J., Grandjacques, M., and Ohmman, L. H., "NACA 0012 Airfoil," Experimental Data Base for Computer Program Assessment, AGARD Advisory Rept. 138, May 1979.

[16]Van der Berg, B., and Oskam, B., "Boundary Layer Measurements on a Two-Dimensional Wing with Flap and a Comparison with Calculation," Paper 18, AGARD-CP-271, 1979.