JOURNAL OF AIRCRAFT
Vol. 38, No. 4, July–August 2001

Euler Computations of Rotor Flows on Unstructured Rotating Meshes

Dartzi Pan*
National Cheng-Kung University, Tainan 70101, Taiwan, Republic of China
Maw-Jyi Chao†
Chung Shan Institute of Science and Technology, Taichung 40001, Taiwan, Republic of China
and
S. K. Chien‡
National Cheng-Kung University, Tainan 70101, Taiwan, Republic of China

An unstructured rotating mesh methodology is developed for the Euler flow computation over rotor blades. A two-zone unstructured mesh is set up with one rotating zone encompassing the blade and one stationary zone extending to the far stream. The rotating zone rotates rigidly with the blade and slides against the stationary zone. The sliding zonal interface is specially designed to enforce the flux conservation between the two zones. The Euler equations are solved in the inertial frame for both zones. The flowfield of a helicopter blade in hover and in forward flight are computed to validate the present methodology.

Introduction

THE accurate computation of a helicopter flowfield is important for proper and efficient airload predictions in forward flight and hover. The aerodynamic environment and loads for a rotorcraft in flight are constantly changing. Strong tip vortices in the unsteady rotor wakes dominate the flowfield, making accurate aerodynamic predictions very difficult. Recent advances on computational fluid dynamics (CFD) have made it possible to accurately obtain rotor blade aerodynamics by solving the Euler or the Navier–Stokes equations. Srinivasan et al.[1] and Srinivasan and Baeder[2] used a thin-layer Navier–Stokes method for forward-flight simulation of a nonlifting rotor blade as well as a rotor in hover. Chen and McCroskey[3] and Agarwal and Deese[4] solved the Euler equations. The governing equations in these methods are formulated in rotating frame and are recast in the absolute-flow variables, generating source terms in the formulation. These methods use single-structured grids, making it difficult to investigate the aerodynamic interference between a rotor and the fuselage. Another notable advance in CFD method is the use of moving overset grids or the chimera method.[5,6] The chimera method is capable of solving the transient flow about multiple bodies moving relative to each another. However, the chimera method requires efficient determination of the connectivity and the interpolation coefficients between the overset grids, which is not at all an easy task.

In this paper the Euler flowfield about a rotating blade is solved on a two-zone unstructured mesh. The two zones are set up such that one stationary zone contains the stationary objects, such as a helicopter fuselage and the far-field boundaries, and one rotating zone encompasses the rotating blade and rotates rigidly with the blade. The two zones have a sliding motion relative to each other along the zonal interface. The meshes in both zones are unstructured, but the zonal interface is specially designed such that certain geometric periodicity can be utilized to reduce the complexity of the sliding interface. With a proper treatment of the flux conservation through the zonal interface, the governing equations can be solved in the inertial frame of reference for both zones. This strategy of unstructured rotating mesh is simple and powerful when dealing with rotating objects in the flowfield. Several examples of helicopter blade in hover and in forward flight are computed and compared with the experiment. Finally, a propeller–wing–body combination is computed to demonstrate the applicability of the present method to complex configurations.

Received 25 January 2000; revision received 3 February 2001; accepted for publication 13 February 2001.

*Professor, Department of Aeronautics and Astronautics.
†Associate Researcher, Aeronautics Research Laboratory.
‡M.S. Student, Department of Aeronautics and Astronautics.

Unstructured Euler Solver

The semidiscrete form of the Euler equations can be written as

$$\frac{\partial Q_i V_i}{\partial t} = -\sum_{j=k(i)} F^N_{i,j}\Delta S_{i,j} = -R_i \tag{1}$$

where Q_i is the volume-averaged conserved variables of cell i, V_i the volume of cell i, $k(i)$ the list of neighboring cells of cell i, $\Delta S_{i,j}$ the area of the cell interface between cell i and cell j, and $F^N_{i,j}$ the numerical flux through $\Delta S_{i,j}$. The right-hand-side operator R_i is equivalent to the surface integral of flux functions over the control surfaces enclosing cell i. The conserved variables and the flux functions are given by

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \qquad F^N = \begin{pmatrix} \rho(\boldsymbol{v}-\boldsymbol{v}_b)\cdot\hat{n} \\ \rho u(\boldsymbol{v}-\boldsymbol{v}_b)\cdot\hat{n} + pn_x \\ \rho v(\boldsymbol{v}-\boldsymbol{v}_b)\cdot\hat{n} + pn_y \\ \rho w(\boldsymbol{v}-\boldsymbol{v}_b)\cdot\hat{n} + pn_z \\ \rho E(\boldsymbol{v}-\boldsymbol{v}_b)\cdot\hat{n} + p\boldsymbol{v}\cdot\hat{n} \end{pmatrix} \tag{2}$$

where ρ is the density, $\boldsymbol{v}$ the fluid velocity vector with Cartesian components u, v, and w, $\boldsymbol{v}_b$ the velocity vector of the cell interface, $\hat{n}$ the surface normal of the cell interface, p the pressure, and E the total energy per unit mass of the fluid. The ideal gas law is assumed for the equation of state. All variables are normalized properly by their reference values. For example, the velocity is normalized by the freestream speed of sound a_∞, the pressure by $\rho_\infty a_\infty^2$, etc.

When the grid is moving, the cell face velocity $\boldsymbol{v}_b$ must satisfy the geometric conservation law governing the rate of change of cell volume:

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} = \sum_{j=k(i)} (\boldsymbol{v}_b\cdot\hat{n})_{i,j}\Delta S_{i,j} = \sum_{j=k(i)} (v_{bn})_{i,j}\Delta S_{i,j} \tag{3}$$

where v_{bn} is the grid speed in the cell face normal direction. Note that Eq. (3) can be derived from Eq. (1) by setting Q to a constant vector.

The Euler solver used in this work has been validated[7,8] for steady-state computations. The numerical flux function $F_{i,j}^N$ through the face center is evaluated by a MUSCL-type upwind flux-differencing scheme based on Roe's approximate Riemann solver. For second-order spatial accuracy, Frink's[9] interpolation is used to obtain the conserved variables at face centers. Frink's interpolation requires cell-vertex variable values, which are obtained by a volume-weighted averaging of the surrounding cell-centered values.

The time integration of Eq. (1) from time step n to $n+1$ with subiterations can be written as

$$\left(\frac{V_i}{\Delta t}+\theta\frac{\partial R_i^s}{\partial Q}\right)\Delta Q_i^s = -\left(\frac{(Q_i^s - Q_i^n)V_i}{\Delta t}+\theta R_i^s + (1-\theta)R_i^n\right)$$

$$= \mathrm{Res}_i^s, \qquad Q_i^{s+1} = Q_i^s + \Delta Q_i^s \tag{4}$$

where the superscript s is the subiteration index. Equation (4) is time accurate in the sense that when subiteration in s converges, Q^{s+1} is equivalent to Q^{n+1}. It is second-order time accurate with $\theta = 0.5$ and first-order with $\theta = 1$.

The implicit operator of Eq. (4) is simply the Jacobian of the right-hand-side terms. To reduce the bandwidth of the implicit operator, only first-order spatial accuracy is used on the implicit side whereas the explicit side is kept unchanged. The implicit operator is further split into a diagonal part D, a lower triangular part L, and an upper triangular part U. Neglecting the subscript i, an approximate lower–upper factorization scheme (ALU)[10] for the preceding equation can be written as

$$(D+L)D^{-1}(D+U)\Delta Q^s = \mathrm{Res}^s \tag{5}$$

The error introduced by the factorization is $LD^{-1}U\Delta Q^s$. Equation (5) has three factors on the implicit side. Normally a three-step inversion process is needed to solve it. In this paper a more efficient two-step inversion process is employed as

$$(D+L)\Delta Q^* = \mathrm{Res}^s, \qquad (D+U)\Delta Q^s = \mathrm{Res}^s - L\Delta Q^* \tag{6}$$

where ΔQ^* is an intermediate solution. To further simplifiy the inversion process, on the implicit side of Eq. (6) the absolute flux Jacobian is replaced by the spectra radius of A, or ρ_A, the positive flux Jacobian A^+ by $0.5(A+\rho_A)$ and the negative flux Jacobian A^- on by $0.5(A-\rho_A)$. This in effect has modified the operator D to a scalar diagonal matrix. The solution process of Eq. (6) is explicit and requires only matrix–vector multiplication and scalar operations. The converged solution of Eq. (6) remains the same order of accuracy in time and space as that in Eq. (4).

Sliding Zonal Interface

Two zones of unstructured mesh, one stationary and one rotating, are set up. The rotating zone encloses the blade, rotates rigidly with it, and consequently slides against the stationary zone. The sliding zonal interface is cylindrical with circular cross sections as shown in Fig. 1. There are disklike planar interfaces on both ends of the cylinder and ringlike circumferential interfaces, as shown in Fig. 2. On both kinds of interfaces, circular paths centered at the axis of rotation are drawn such that the zonal cell vertices rotate along these paths. The interfacial triangles between two circular paths are distributed with fixed geometric periodicity such that the span of each period in azimuth angle ψ is $2\pi/N$, where N is the total number of period in one revolution of rotation. Each period contains a fixed pattern of triangles. For simplicity, only a two-triangle pattern and a three-triangle pattern shown in Fig. 3 are used.

On ringlike circumferential interfaces, the gap between circular paths is designed to be constant. The two-triangle pattern and a constant N are used for all circumferential interfaces, as can be seen in Fig. 2. On disklike planar interfaces, the circular paths are actually coplanar concentric circles with constant increment in radius. The gap area between two concentric circular paths in this case becomes progressively larger as the radius becomes larger. Thus, the number of triangles between two circular paths should vary to avoid unacceptable triangle quality. In this work we use the two-triangle pattern and the three-triangle pattern alternatively between paths to obtain a smooth variation of triangle distribution. This is demonstrated on the disklike interface in Fig. 2, where the arrangement of a two-triangle pattern followed by a three-triangle pattern and then a two-triangle pattern again changes the number of triangles between two planar circular paths from M to $1.5M$ to $2M$. Note that in Fig. 2 the innermost triangles share the central axis as the common vertex. They can be viewed as a special two-triangle arrangement.

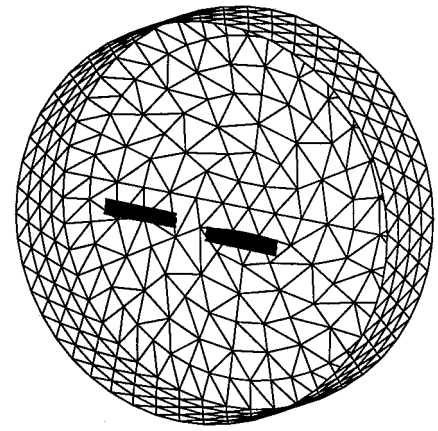

Fig. 1 Rotating zone showing sliding zonal interface.

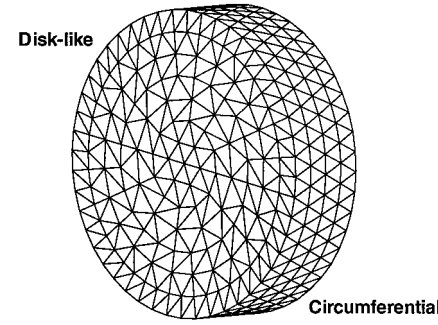

Fig. 2 Periodic patterns on disklike and circumferential interfaces.

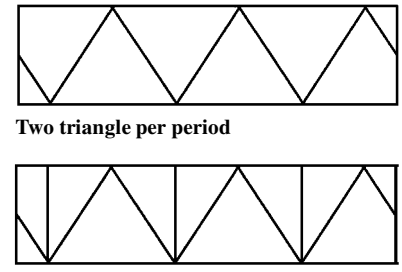

Fig. 3 Zonal interface.

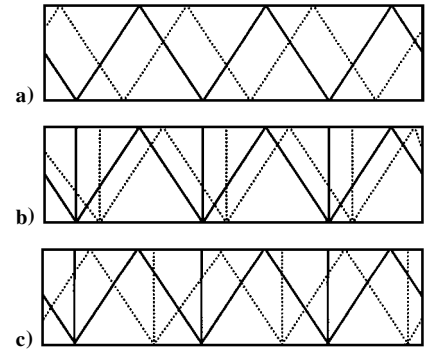

Fig. 4 Crossover patterns of a) two-triangle period, b) first-half of three-triangle period, and c) second-half of three-triangle period.

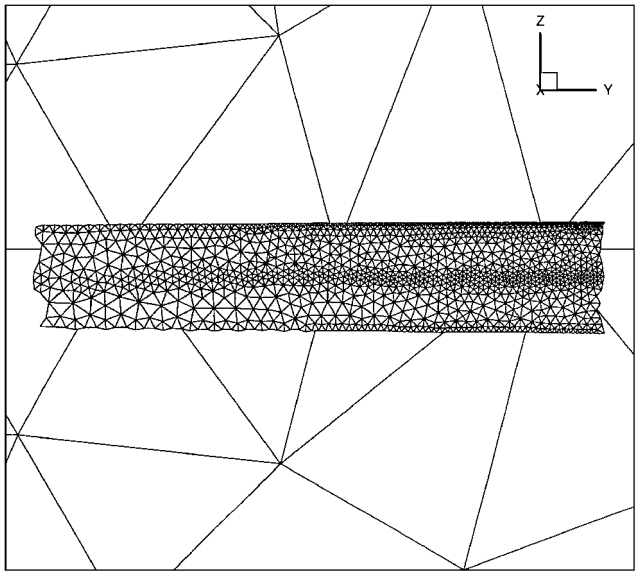

Fig. 5 Surface triangles on NACA 0012 blade.

On the stationary zone, the zonal interface is set up exactly the same way as that on the rotating zone. Before grid rotation, the two zones are connected on the zonal interface node by node without mismatch. When the rotating zone rotates, the interfacial area between two circular path lines rotate as a rigid ring, and the triangles on each ring cross over those on the corresponding stationary ring. Because of the simplicity of the geometric patterns described, the patterns of crossover can be easily determined. For example, Fig. 4a shows the crossover pattern for the two-triangle pattern with six subareas. The crossover pattern for the three-triangle structure is shown in Fig. 4b for the first-half of period and in Fig. 4c for the second-half of period, each with 10 subareas. Once the crossover patterns are determined, the next step is to calculate the fluxes through each subarea just identified and then to distribute these fluxes to the corresponding tetrahedral cell on both sides of the subarea.

To ensure flux conservation, the fluxes through the zonal interface is computed once on the stationary zone and distributed to the rotating zone. The global flux conservation between two zones can be guaranteed if the fluxes through each and every subarea have been accounted for and distributed to the rotating zone. However, the local flux conservation will depend on the accuracy of the fluxes through each subarea. There are two simplifications in the flux computation that should be mentioned here. The first one is the representation of curved surfaces and curves. Ideally, the zonal triangles just

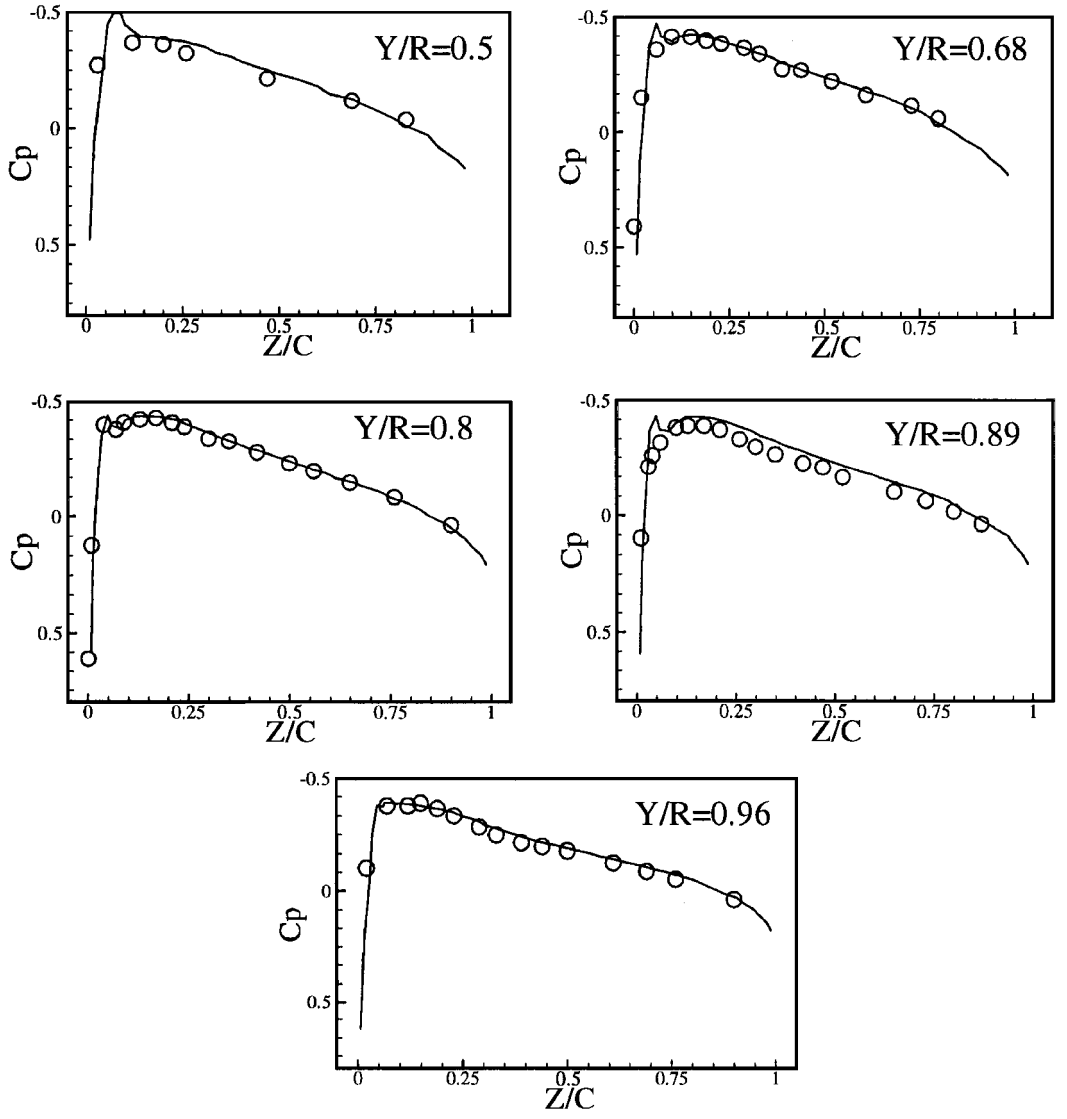

Fig. 6 Pressure distribution at different spanwise stations for two nonlifting blades in hover, NACA 0012, $AR = 6$, $M_t = 0.52$, and $\theta_c = 0$; line, computed data and symbols, experimental data.

described have curved surfaces or circular edges. Although it is possible to compute the actual area of a curved surface and to integrate the fluxes through it, in this work the circular ring structures on both types of interface are relaxed to two-dimensional straight coplanar bands with periodic conditions on both ends. Under this simplification, the triangles and subareas on the zonal interface have straight edges, which greatly simplifies the area computation. In our case, it is easy to determine that these subareas are either linear or quadratic functions in time or in azimuth angle ψ when the angular speed is constant. The normal vector of each subarea is taken the same as that of the stationary triangular interface containing the subarea.

The second simplification is for the variable distribution on the zonal interface. On each subarea, a local Riemann problem is solved to compute the fluxes through it. The two Riemann states on the subarea need to be evaluated from the cell-center values on both sides of the subarea. For first-order spatial accuracy, a constant variable distribution is taken for each tetrahedral cell containing the subarea. For second-order spatial accuracy, a linear variable distribution should be used to compute the Riemann states at the center of each subarea. Remembering that the Frink's interpolation has been used to compute the cell-face center values, it is further assumed that the Riemann states of a subarea are equal to the cell-face center values of the triangle containing the particular subarea. This is equivalent to assuming a zero gradient on the triangular interface, even though a linear gradient (Frink's method) is used for the tetrahedron. The spatial accuracy of the interface calculation under the preceding assumption is greater than first order, but lower than second order.

The simplifications employed inevitably introduce some errors in the computation of fluxes. Fortunately, these errors decrease as the number of zonal triangles increases. The actual influences of these errors on the solutions are problem and mesh dependent. Careful validations should be performed before a zonal interface can be accepted in the routine computations. In our work, a freestream correction[11] term is added to the right hand side of Eqs. (1) and (4) to reduce the impact of these errors.

Grid Velocity

The grid velocity $\boldsymbol{v}_b$ for each cell face of the rotating zone follows the rigid-body rotation $\boldsymbol{v}_b = \boldsymbol{\Omega} \times \boldsymbol{r}$ where $\boldsymbol{\Omega}$ is the vector of angular velocity and $\boldsymbol{r}$ is the distance vector to the center of zonal triangle. Here it is assumed that the axis of rotation passes through the origin of the distance vector. For all rotating cells, the grid velocity computed in this way satisfies Eq. (3) automatically. The grid speed in the normal direction of a cell face can be written as $v_{bn} = (\boldsymbol{r} \times \hat{n}) \cdot \boldsymbol{\Omega}$. The geometric term in the parenthesis can be computed once during the grid-generation phase and saved for later use. Note that theoretically on the sliding interface $\boldsymbol{v}_b$ is parallel to the interface, and, hence, v_{bn} should be zero. On the disklike planar interfaces at both ends of the zonal cylinder, this is automatically satisfied because $\hat{n}$ is parallel to $\boldsymbol{\Omega}$. On the circumferential interfaces, however, it is important to realize that to achieve zero v_{bn}, we must require the zonal triangles be equilateral and symmetric in ψ, making $\hat{n}$, $\boldsymbol{r}$, and $\boldsymbol{\Omega}$ coplanar.

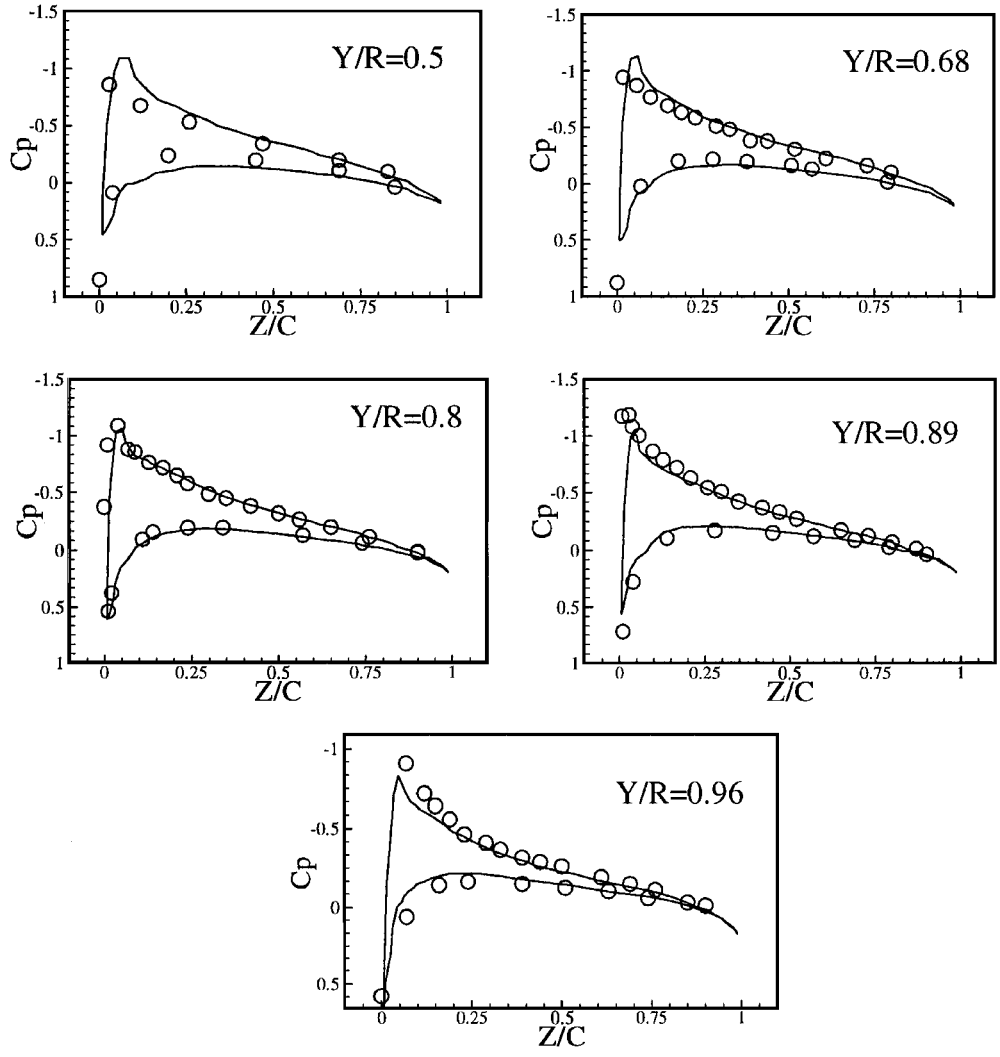

Fig. 7 Pressure distribution at different spanwise stations for two lifting blades in hover, NACA 0012, $AR = 6$, $M_t = 0.44$, $\theta_c = 8$ deg, and $\alpha_e = 4.2$ deg; line, computed data and symbols, experimental data.

Test Cases

In Ref. 12, experiments are performed on a NACA 0012 rotor for a range of blade tip Mach numbers M_t and the collective pitch angles θ_c. The rotor has two untwisted and untapered blades of aspect ratio 6. In this work, Euler computations of this rotor are performed and results compared with the results in Refs. 12 and 13. All computations are done on a Pentium II-350 personal computer with 256-MB memory. Only single precision is used to save the computational resources. A self-developed advancing-front grid generator is used to generate the tetrahedral meshes shown in this work. For all computations in this work the Euler implicit time integration ($\theta = 1$) is used for a better convergence characteristic. The far-field boundary is treated by a one-dimensional nonreflecting characteristic like boundary condition.[14] On the blade surface, the normal component of velocity is set to zero, and the tangent component is extrapolated from the cell center, ensuring the flow tangency condition on the blade surface. Both pressure and density on the blade surface are extrapolated from the cell center.

A freestream test is performed first on a two-zone mesh with 704,442 stationary cells, 112,114 rotating cells, and a sliding zonal interface of 1,560 triangles. The cylindrical rotating interface is 21 units in radius and 18 units in height. Initially a freestream condition is assumed everywhere, and no boundary condition is enforced except the sliding interface boundary condition. At each time step, the rotating zone rotates a fixed preset increment $\Delta\psi$ in azimuth angle, and the value of the time step is chosen to achieve the desired angular speed or tip Mach number. In this case, $\Delta\psi = 1$ deg, and a time step corresponding to $M_t = 0.52$ is chosen. When integrating Eq. (4) without the freestream correction term, the L_2 norm of the right-hand residual fluctuates around 5×10^{-4} (single precision), indicating the errors introduced by the sliding interface. With the freestream correction term activated, the residual can be dropped to and maintained around 5×10^{-7} regardless of the angular speed. For all of the following computations, the rotating zone always rotates with $\Delta\psi = 1$ deg per time step, and the freestream correction term is always activated.

The first validation case is for a nonlifting rotor in hover with $M_\infty = 0$, $M_t = 0.52$, and $\theta_c = 0$. The blade chord length is normalized to unity. The same rotating interface used in the freestream test is employed here. The cylindrical rotating zone has a radius of 21 chords length and a height of 18 chords length with 1560 zonal triangles on the surface. The stationary zone is outside the rotating zone and extends to 34 chords length to the far field. There are 215,342 tetrahedral cells in the rotating zone and 31,254 cells in the stationary zone. There are 6090 surface triangles on one blade with some clustering around the midchord area and the blade tip, as shown in Fig. 5. This grid clustering is not solution adaptive, and it is chosen just for convenience. Note that the interface is very far from the blade surface, which reduces the error of the interface treatment.

The computation starts with the blade rotating at $\Delta\psi = 1$ deg per time step, which gives $M_t = 0.52$. Three subiterations are performed at each time step. Because the flow field relative to the blade is

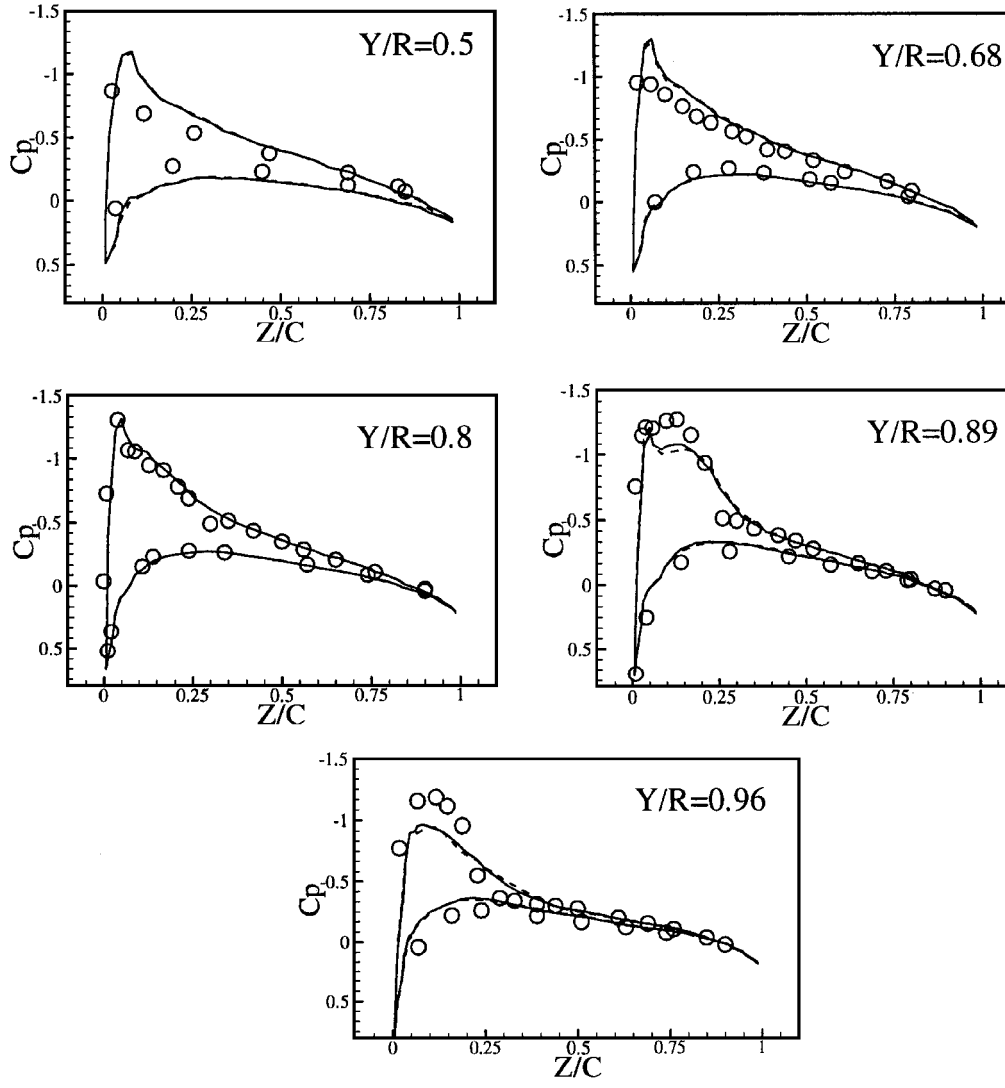

Fig. 8 Pressure distribution at different spanwise stations for two lifting blades in hover, NACA 0012, $AR = 6$, $M_t = 0.794$, $\theta_c = 8$ deg, and $\alpha_e = 4.2$ deg: solid line, computed data with large interface; dashed line, computed with small interface; and symbols, experimental data.

independent of the azimuth angle ψ, this is equivalent to a steady-state computation. It is found that about one-half blade revolution is enough to drop three orders of magnitude in the L_2 norm of the residual, reaching the steady state. Figure 6 shows the pressure distribution at various spanwise stations Y/R, where R is the blade span. The agreement with the experimental data of Ref. 12 is quite good.

The next case is a lifting rotor in hover with a pitch angle $\theta_c = 8$ deg. The mesh is very similar to the one used earlier, with some differences introduced by the geometric angle of attack. Because the blade is experiencing lift, there is a wake flow in this case. The correct modeling of the wake effect is important to the blade airloads. Although in theory an Euler solver is capable of capturing the wake flow without any wake model, its accuracy depends on the grid resolution. Unfortunately, the current mesh is too coarse to capture wake effects. Following the treatment in Ref. 4, the wake effects are modeled in the form of a correction applied to the geometric angle of attack. According to Ref. 4, the wake induced angle of attack is approximately $\alpha_i = 3.8$ deg throughout the blade, and hence the effective angle of attack used in the computation is $\alpha_e = \theta_c - \alpha_i = 4.2$ deg. This correction is employed in the computation. Figure 7 show the computed surface pressure coefficients on the blade at various spanwise stations for a tip Mach number $M_t = 0.44$. In this case, the flowfield is smooth without shock. The comparison with experimental data is quite satisfactory.

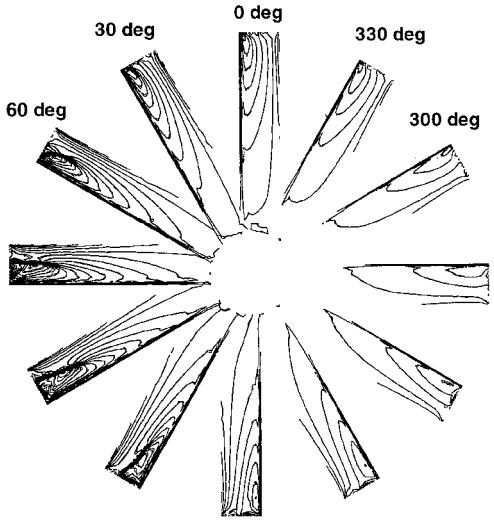

Fig. 9 Pressure contours on advancing blade every 30 deg of azimuth angle, wind direction from 180 to 0 deg (bottom up), nonlifting, NACA 0012, $AR = 6$, $M_t = 0.8$, $\theta_c = 0$ deg, and $\mu = 0.2$.

Figure 8 shows the computed surface pressure for $M_t = 0.794$ in which shocks are formed near the blade tip region. The results generally follow the experimental data, but the leading-edge pressure suction and the shock strength are generally underpredicted. This is basically due to the lack of grid resolution. It is speculated that an adaptive grid clustering could improve the solution significantly. These results are similar to those obtained in Ref. 4.

In general, the accuracy of the interface treatment depends on the interface position as well as the number of interfacial triangles. To investigate the influences of the interface, a new sliding zonal interface is constructed with a radius of 7 chords length and a height of 2.2 chords length. This puts the interface only one chord length away from the blade surface, which increases its influence on the blade surface. The new mesh contains 177,506 rotating tetrahedral cells and 48,360 stationary cells. To reduce the interface inaccuracy, the number of triangles on the interface is increased from 1560 to

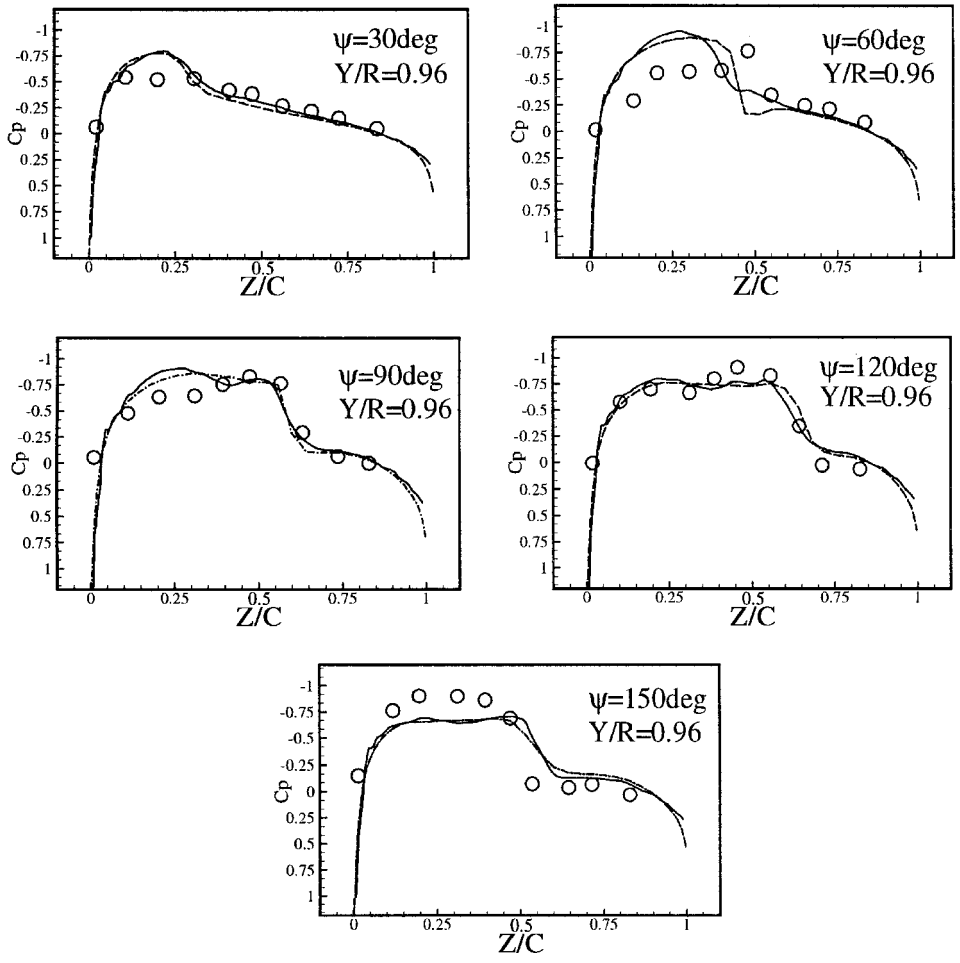

Fig. 10 Pressure distribution on spanwise location $Y/R = 0.96$ for an advancing lifting blade, wind direction from azimuth angle 180–0 deg (bottom up), NACA 0012, $AR = 6$, $M_t = 0.8$, $\theta_c = 0$ deg, and $\mu = 0.2$; solid line, current solution; dashed line, structured Euler solution; and symbols, experimental data.

2808. The blade surface triangles remain unchanged. With this new interface, the computed surface pressure coefficients are shown as the dashed lines in Fig. 8. When compared with those computed with the larger interface (solid lines), the difference is very small, indicating that in this case the inaccuracy due to the interface treatment is small.

The fourth case is a nonlifting rotor blade in forward flight with $M_t = 0.8$, $\theta_c = 0$ deg, and an advancing ratio of $\mu = 0.2$. This is equivalent to an incoming freestream with Mach number $M_\infty = 0.16$. There is no steady-state solution in this case, and ultimately the flow will become periodic in time. The computation starts with $\Delta\psi = 1$ deg for each time step with 50 subiterations. Because the blade advancement $\Delta\psi = 1$ deg is quite small, a first-order time accuracy ($\theta = 1$) is considered adequate. The solutions at each time step are compared with those obtained in the last blade revolution for time periodicity. It is found that at least three-quarter blade revolution is necessary to set up the time periodicity. That is, if the computation starts with the right (or left) blade at ψ_0 azimuth angle, then time periodicity can be achieved after the right (or left) blade past $\psi_0 + 3/2\pi$ azimuth angle. The solutions from $\psi_0 + 3/2\pi$ to $\psi_0 + 5/2\pi$ on both the right and left blades constitute a complete cycle of the solution. The number of subiterations required for a stable solution needs to be increased to about 100 when one blade is at the location around $\psi = 30$ deg (see Fig. 9) where the tip shock is about to form. After the shock has formed, for example, after $\psi > 40$ deg, the number of subiterations can be reduced to 50 again. This control over subiteration is done manually.

Figure 9 shows the pressure contours at every other $\Delta\psi = 30$ deg for one complete cycle of rotation. The freestream is flowing from $\psi = 180$ to 0 deg. The transonic shock starts to form around $\psi = 30$ deg and disappears before $\psi = 180$ deg. The shock extends about one chord length inward from the tip. There is no shock for the $\pi \le \psi \le 2\pi$ region, or when the blade is moving downwind. Figure 10 shows the comparisons of the periodic solutions with experimental data at spanwise location $Y/R = 0.96$ for several azimuth angles. For comparison sake, an Euler solution in a rotating frame obtained by a structured code[15] using a $169 \times 40 \times 73$ grid is also shown in Fig. 10 (dashed lines). The two computations agree with each other fairly well except for $\psi = 60$ deg. Both computations roughly follow the experimental data, but some obvious discrepancies in shock positions and strengths can be identified.

The last example is for an idealized propeller–wing–body combination shown in Fig. 11. The mesh contains 249,875 rotating tetrahedral cells and 75,632 stationary tetrahedral cells. This is a very crude computation for such a complex configuration. Because we are limited by the computing power of one personal computer, this serves only as a demonstration of the current method to handle complex geometry. The same NACA 0012 rotor as in the preceding examples

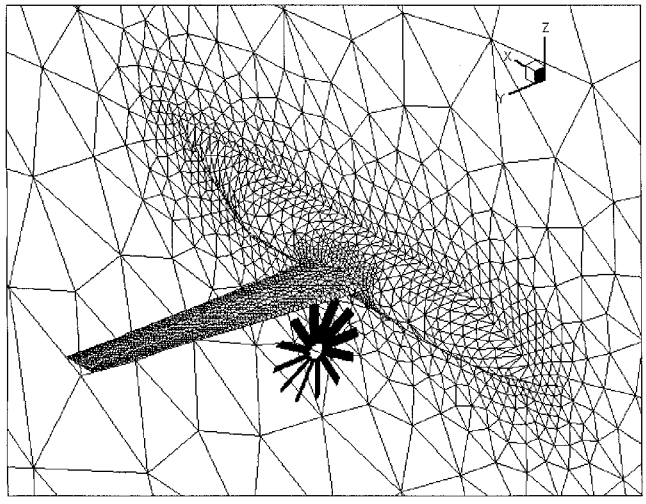

Fig. 11 Grids for propeller–wing–body configuration.

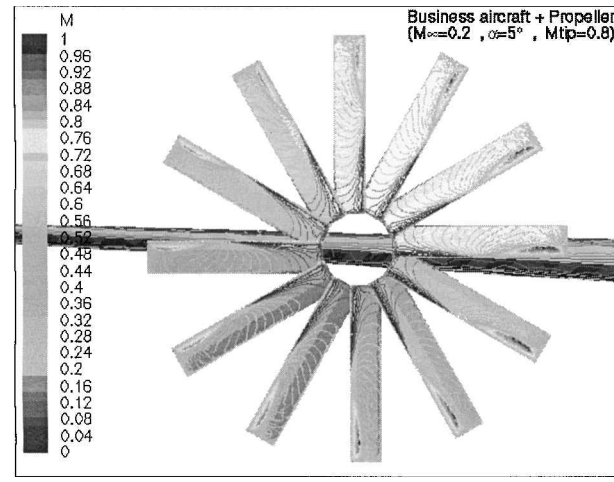

Fig. 12 Mach contours on blade surfaces at various instances of time.

is used as a propeller, which is not a realistic one. To produce positive thrust, the pitch angle is set to 27 deg. The tip Mach number is $M_t = 0.8$. The incoming freestream is at $M_\infty = 0.2$, and the angle of attack is 5 deg. The propeller rotates clockwise. Figure 12 shows the front view of the computed surface Mach number on the propeller for various instances of time. There are shocks close to the tip of the propeller. The blade surface Mach contours are almost independent of the azimuth angle, indicating that the influence from the wing and body is small. The flow differences among rotors are mainly due to the freestream angle of attack.

Conclusions

An unstructured rotating mesh methodology for Euler blade flows has been demonstrated. The method is unique in that it combines the power of unstructured mesh generation and a specially designed sliding zonal interface to allow easy treatment of the Euler equations in an inertial frame. The global fluxes through the zonal interface are strictly conserved. However, the local flux computation is simplified based on geometric considerations, generating interface errors that depend on the interface position and the number of interfacial triangles. It is shown that the errors are small in the test cases. The flowfield of a helicopter blade in hover and in forward flight are computed and generally compare well with experiments. The applicability of the present method to complex geometry is demonstrated by an idealized propeller–wing–body configuration. The present method is also applicable to viscous flows, provided that the viscous fluxes through each zonal subarea can be accurately computed.

Acknowledgment

This work is partially supported by Grant NSC88-2623-D006-008.

References

[1] Srinivasan, G. R., Baeder, J. D., Obayashi, S., and McCroskey, W. J., "Flowfield of a Lifting Rotor in Hover: A Navier–Stokes Simulation," *AIAA Journal*, Vol. 30, No. 10, 1992, pp. 2371–2378.

[2] Srinivasan, G. R., and Baeder, J. D., "TURNS: A Free-Wake Euler/Navier–Stokes Numerical Method for Helicopter Rotors," *AIAA Journal*, Vol. 31, No. 5, 1993, pp. 959–962.

[3] Chen, C. L., and McCroskey, W. J., "Numerical Solutions of Forward Flight Rotor Flow Using an Upwind Method," AIAA Paper 89-1846, June 1989.

[4] Agarwal, R. K., and Deese, J. E., "Euler Calculations for Flowfield of a Helicopter Rotor in Hover," *Journal of Aircraft*, Vol. 24, No. 4, 1987, pp. 231–238.

[5] Meakin, R., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper 93-3350, July 1993.

[6]Ahmad, J., and Duque, E. P. N., "Helicopter Rotor Blade Computation in Unsteady Flows Using Moving Overset Grids," *Journal of Aircraft*, Vol. 33, No. 1, 1996, pp. 54–60.

[7]Pan, D., and Chao, M. J., "Hybrid Octree/Advancing-Front Mesh Generation and Implicit Agglomeration Multigrid Euler Solver," AIAA Paper 99-0918, Jan. 1999.

[8]Pan, D., and Chao, M. J., "An Octree-Based Implicit Agglomeration Multigrid Method," *AIAA Journal*, Vol. 37, No. 8, 1999, pp. 1002–1005.

[9]Frink, N. T., "Upwind Scheme for Solving the Euler Equations on Unstructured Tetrahedral Meshes," *AIAA Journal*, Vol. 30, No. 1, 1992, p. 70.

[10]Pan, D., and Cheng, J. C., "Upwind Finite Volume Navier–Stokes Computations on Unstructured Triangular Meshes," *AIAA Journal*, Vol. 31, No. 9, 1993, pp. 1618–1625.

[11]Pulliam, T. H., and Steger, J. L., "Implicit Finite-Difference Simulations of Three-Dimensional Compressible Flow," *AIAA Journal*, Vol. 18, No. 1, 1980, p. 159.

[12]Caradonna, F. X., and Tung, C., "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," NASA TM 81232, Sept. 1981.

[13]Caradonna, F. X., Desopper, A., and Tung, C., "Finite-Difference Modeling of Rotor Flow Including Wake Effects," Paper 2.7, 8th European Rotorcraft Forum, Aix-en-Provence, France, Aug. 1982.

[14]Pan, D., and Chang, C.-H., "The Capturing of Free Surfaces in Incompressible Multi-fluid Flows," *International Journal For Numerical Methods In Fluids*, Vol. 33, 2000, pp. 203–222.

[15]You, H. D., "Acoustic Forcing of Blade Rows in Three-Dimensional Transonic Flow," Ph.D. Dissertation, Inst. of Aeronautics and Astronautics, National Cheng-Kung Univ., Taiwan, ROC, June 2000.