AIAA JOURNAL
Vol. 38, No. 1, January 2000

Finite Element Adaptive Multigrid Euler Solver for Rotary Wing Aerodynamics

Carlo L. Bottasso*
Politecnico di Milano, 20158 Milan, Italy
and
Mark S. Shephard†
Rensselaer Polytechnic Institute, Troy, New York 12180-3590

We present a multigrid strategy for the solution of the Euler equations in three-dimensional domains. The procedures include data structures and mesh modification tools for defining unstructured adaptive multilevel meshes on general curved domains and an implicit time-marching multigrid finite element solver. The code is equipped with special features for the solution of problems relevant to rotorcraft technology. The approach is applied to the simulation of hovering rotors, and the numerical results are compared with experimental data and with numerical simulations obtained using a mesh adaptive single-grid generalized minimal residual (GMRES) solver.

I. Introduction

A GENERAL purpose code for rotary wing aerodynamics should possess 1) modeling flexibility, 2) accuracy, 3) robustness, and 4) high computing performance. In this paper we present a set of procedures that, combined in a general software tool, address each one of the just-mentioned areas of concern.

Modeling flexibility is required for treating complex geometries and configurations. Modeling flexibility is crucial for being able to move from academic test cases to the solution of problems of industrial relevance. The selection of appropriate data structures and procedures for supporting the analysis on arbitrary domains is a nontrivial task, especially in three dimensions. We address the modeling issue by means of geometry-based procedures that interact with a CAD representation of the computational domain.[1] This enables mesh generation and adaption using unstructured grids in arbitrarily complex domains. The physical attribute information required to support the analysis is tied to the geometric model definition, rather than to the discrete model.[2] This offers distinct advantages in an automated analysis environment. The data structures used in this work are detailed in Sec. II.

A general automated procedure should deliver a solution to a given level of accuracy without user intervention. Ideally, the analyst would prescribe a tolerance in the solution accuracy and "push the button." Although this goal is not yet fully achieved, a high degree of automation can be obtained through the interaction of two tools: error estimates that determine which regions of the computational domain are over-refined or are not providing sufficient accuracy, and a mesh adaption procedure that optimizes the discretization based on the information provided by the error indicator. We have developed general adaptive strategies that can be used for locally adapting an existing discretization in complex geometries in a fully automated way.[3,4] Details of this procedure are given in Sec. III.

To address the issue of robustness of the numerical scheme, we use a time-discontinuous Galerkin least-squares (TDG/LS) finite element method.[5] This scheme is particularly well-suited for the incorporation in an automated adaptive environment. In fact the method has a firm mathematical foundation, and its stability and accuracy properties have been rigorously established. The finite element formulation is reviewed in Sec. IV.

Multigrid methods are at the heart of many powerful algorithms for the high-performance solution of complex problems of engineering interest. The multigrid approach takes advantage of the relations among different discretizations of the governing partial differential equations for delivering a fast numerical solution.[6] When using a grid-adaptive strategy, the multigrid hierarchy can be obtained naturally by the adaption process. This is by far more efficient and achieves a higher level of automation than the creation of multigrid levels by complete remeshing.[7] We have implemented a robust implicit multigrid solver, whose details are given in Sec. V.

Adaptivity and multigrid have been used with success for solving a number of problems in computational fluid dynamics (CFD). For example, Refs. 7 and 8 both develop unstructured adaptive multigrid methods for the Euler equations. Adaptivity has already been used in rotorcraft CFD, for example, for capturing shocks, tracking wakes, and analyzing noise problems.[3,4,9] In this paper our goal is to use adaptive multigrid for rotorcraft aerodynamic problems.

Using our procedures, the analysis process begins with the definition of the problem to be solved. This is obtained by constructing a CAD solid model of the domain, by defining all of the problem attributes (boundary conditions, initial values, etc.) and by associating each attribute with one or more entities of the solid model. An initial grid is obtained with a mesh generator,[1] and an initial solution is obtained on this grid. This initial solution can be computed using a single-grid algorithm based on a quasi-Newton scheme equipped with a preconditioned generalized minimal residual (GMRES) solver[10] or with the implicit multigrid algorithm. The multigrid hierarchy can be obtained by uniform refinement or coarsening of the initial mesh. The initial solution is then used to obtain error estimates that drive the mesh optimization process. A new solution is obtained on this grid, and the process is continued until a specified tolerance in the solution accuracy is achieved. This way, the different meshes define an adapted multigrid hierarchy that is used for the solution of the discrete problem.

This multigrid algorithm and procedures were incorporated in an existing single-grid adaptive flow solver recently developed by the authors.[4] In this paper we show how these procedures can be used for the multigrid adaptive solution of different flow problems relevant to rotorcraft technology. We demonstrate the capabilities of the code in Sec. VI, with the help of simulations of hovering rotors.

II. Data Structures

The data structures are used to store the information relevant to the problem domain and its discrete approximations, the grids. The

Received 13 October 1998; revision received 21 June 1999; accepted for publication 8 July 1999;

Dedicated to Andrea Azzoni (1971–1999).

*Ricercatore, Dipartimento di Ingegneria Aerospaziale, Campus Bovisa, Via La Masa 34.

†Director, Scientific Computation Research Center, Johnson Chair of Engineering.

data structures used in this effort are detailed in Ref. 11. Here we review the basic features of the proposed approach.

A. Basic Low-Level Functionality

The basic functionality is provided by a general geometry-based database,[12] implemented using the object-oriented design philosophy.

The mesh is described using a boundary representation. Because the data structure must be able to support operations that are highly demanding from the topology understanding point of view, it was designed to provide efficiently all of the possible adjacencies, i.e., the relationships among topological entities that bound each other. All of the first-order adjacencies, i.e., all of the adjacencies between entities that are one dimension apart, are explicitly stored in the database. This choice effectively compromises the contrasting requirements of quick determination of adjacencies of each order, and control of the amount of data storage required. All other adjacencies can be obtained through mesh-size independent local traversals.

The database makes use of the concept of classification of a derived model from its parent model. Classification is defined as the unique association of a topological mesh entity of dimension d_j (region, face, edge or vertex), $M_i^{d_j}$, to a topological geometric domain entity of dimension d_k, $G_j^{d_k}$, being $d_j \leq d_k$. This is written as $M_i^{d_j} \sqsubset G_j^{d_k}$. In practice, classification is the mechanism through which the relationship between the mesh and the geometric model is represented. The understanding of these relationships allows one 1) to guarantee the validity of the generated discretizations during any mesh modification operation, 2) to determine when local modifications of the mesh would create invalid elements, and 3) to guarantee that refinement improves the geometric approximation of the model through the positioning of newly generated vertices on the model boundaries when the mesh entities are classified on the model boundary.

The representation of the geometric domain is obtained by functionally interfacing the mesh generator with the CAD system through a set of topological and geometric interrogation operators.[1]

B. Data Structures for Multilevel Meshes

The low level database just described was modified to handle multigrids. We found that only minor modifications to the basic functionalities of the database were required to support unstructured multiple grids in a compact and efficient way.

Our approach stores all first-order adjacencies for all of the grids in one single global pool. The only required change to the single-grid database is the provision for the fact that a mesh face can now be shared by more than two mesh regions (clearly, of different levels, because at any given level only a maximum of two regions can share a face). This is a situation that is not possible for a single grid.[11]

During the multigrid procedure, it is necessary to extract the various multigrid levels from the global pool. To provide this functionality, we associate two unsigned integers with each entity. The two integers correspond to the minimum and maximum level, respectively, where the entity appears. In fact, in general a mesh entity can be shared by more than one level for multilevel meshes generated by adaptive modifications. With our approach duplicated entities appear just once in the data set, thus realizing savings in storage, yet the process of obtaining all of the adjacency information at a given mesh level is fast and requires only local searches.

During adaption and the solution phase, it is often necessary to traverse each mesh level by entity type. To provide this functionality in an efficient manner, we maintain lists of entities of the same dimension for each mesh level. The small additional storage required by these lists is compensated by the faster traversals, which otherwise would require a scan of the complete pool of entities in order to select only the ones belonging to the level considered.

III. Mesh Adaption

A multigrid algorithm requires procedures to generate a hierarchy of coarser or finer meshes starting from a given base mesh. As the solution advances, error estimates can be invoked to adapt locally the discretization to the features of the solution, for example, in the presence of boundary layers, shocks, singular points, etc., or to exploit local smoothness and coarsen the mesh. These adapted grids can be incorporated in the hierarchy of meshes used by the multigrid algorithm, leading to multilevel adaptive solution schemes.

In this work the initial multiple levels and the adapted grids are obtained by means of a set of mesh adaption tools that include refinement, coarsening, and local retriangulation as a way to improve and control the mesh quality locally. Details of the algorithms are given in Ref. 13. The adaptive procedures make use of the data structures described in Sec. II.

A. Coarsening

Coarsening is performed by means of edge collapsing. This allows the coarsening even of the initial base discretization. The algorithm loops on all of the edges marked for coarsening at a certain mesh level, and an attempt is made at collapsing it to one of its end vertices. Before actually performing collapsing, checks are made to ensure that collapsing is topologically and geometrically possible and that the quality of the created mesh regions is superior to a predetermined threshold. Finally, the best target vertex for collapsing is chosen with respect to a given mesh quality measure. A limit can be placed on the longest edge to collapse to avoid the creation of excessively large elements.

The multigrid solution process requires the intergrid transfer of field variable corrections as well as residuals. We employ standard linear shape functions for these tasks. In the coarsening case the edge-collapsing operation deletes a vertex from a mesh level, and consequently the mapping from this vertex to the corresponding mesh entity at the coarser level must be computed. The value of the shape functions at the collapsed point are obtained through parametric inversion. These transfer operators are realized on the fly, at the end of collapsing of each mesh edge. The operation involves only local information and consequently is inexpensive.

B. Refinement

The refinement scheme adopted in this work is based on edge split. The algorithm implements all possible subdivision patterns corresponding to all possible configurations of marked edges to allow the maximum flexibility in how mesh refinement is accomplished. A limit can be placed on the shortest edge to split to avoid excessive refinement.

In the presence of curved geometries, newly generated vertices classified on the model boundary must be properly placed on the true geometric boundary. This snapping procedure is critical in the sense that it represents the mechanism for improving the geometric approximation that a mesh gives of the geometry. This operation relies on the interaction of the refinement procedure with the geometric modeler storing the geometric information (CAD) and the classification information of the mesh entities. The snapping of a refined vertex can produce invalid regions of negative volume or grids of poor quality. In this case the retriangulation procedure explained in the following section is applied and repeated until the vertex can be successfully snapped.

For supporting the computation of the restriction and prolongation operators, a double link is stored from the child vertex to the parent edge and back, together with the value of the split location along the edge. This is also realized on the fly during refinement of each marked edge.

C. Local Retriangulation

We have included local retriangulation algorithms in the procedures in order to control and improve the grid quality. We have considered edge removal and multiface removal, which do not change the number of vertices, and edge collapsing and splitting of one or more edges, faces, or a region, which remove or add vertices.[13]

The procedures are region-based, in the sense that the algorithm tries to improve all regions that violate a given mesh quality criterion. We typically use dihedral angles as optimization targets for their negative impact on the conditioning of the discrete problem.

Although local in nature, these tools have proven valuable in controlling the degradation of the mesh quality during its adaptive modification, in particular during coarsening[11] and for the snapping of newly created vertices on curved boundaries.

IV. Finite Element Formulation

The governing equations for the problem are the unsteady Euler equations. These equations with body forces are written in vector form:

$$\boldsymbol{U}_{,t} + \boldsymbol{A}_i \boldsymbol{U}_{,i} = \boldsymbol{E} \qquad (i = 1, \ldots, n_{\mathrm{sd}}) \tag{1}$$

The problem statement is completed by well-posed initial and boundary conditions. In Eq. (1) n_{sd} is the number of space dimensions. $\boldsymbol{U} = \rho(1, u_1, u_2, u_3, e)$ are the conservative variables. $\boldsymbol{F}_i = \rho u_i(1, u_1, u_2, u_3, e) + p(0, \delta_{1i}, \delta_{2i}, \delta_{3i}, u_i)$ are the inviscid fluxes, whereas their spatial derivatives take the form $\boldsymbol{F}_{i,i} = \boldsymbol{A}_i \boldsymbol{U}_{,i}$. The source vector is $\boldsymbol{E} = \rho(0, b_1, b_2, b_3, b_i u_i + r)$. In the preceding expressions ρ is the density, $\boldsymbol{u} = (u_1, u_2, u_3)$ is the velocity vector, e is the total energy, p is the pressure, δ_{ij} is the Kronecker delta, $\boldsymbol{b} = (b_1, b_2, b_3)$ is the body force vector per unit mass, and r is the heat supply per unit mass.

We use the TDG/LS finite element method[5] that is developed starting from the symmetric form of the equations expressed in terms of the entropy variables $\boldsymbol{V}$. The finite element procedure is based upon the simultaneous discretization of the space-time computational domain. A least-squares operator and a discontinuity capturing term are added to the formulation for improving stability without sacrificing accuracy. The TDG/LS finite element method takes the form

$$\begin{aligned}
&\int_{Q_n} \left[-\boldsymbol{W}^h_{,t} \cdot \boldsymbol{U}(\boldsymbol{V}^h) - \boldsymbol{W}^h_{,i} \cdot \boldsymbol{F}_i(\boldsymbol{V}^h) + \boldsymbol{W}^h \cdot \boldsymbol{E}(\boldsymbol{V}^h) \right] \mathrm{d}Q \\
&\quad + \int_{\mathcal{D}(t_{n+1})} \boldsymbol{W}^{h^-} \cdot \boldsymbol{U}(\boldsymbol{V}^{h^-}) \, \mathrm{d}\mathcal{D} - \int_{\mathcal{D}(t_n)} \boldsymbol{W}^{h^+} \cdot \boldsymbol{U}(\boldsymbol{V}^{h^-}) \, \mathrm{d}\mathcal{D} \\
&\quad + \int_{P_n} \boldsymbol{W}^h \boldsymbol{F}_i(\boldsymbol{V}^h) \cdot \mathrm{d}\boldsymbol{P} + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} (\mathcal{L}\boldsymbol{W}^h) \cdot \tau(\mathcal{L}\boldsymbol{V}^h) \, \mathrm{d}Q \\
&\quad + \sum_{e=1}^{(n_{\mathrm{el}})_n} \int_{Q_n^e} \nu^h \hat{\nabla}_\xi \boldsymbol{W}^h \cdot \mathrm{diag}[\tilde{\boldsymbol{A}}_0] \hat{\nabla}_\xi \boldsymbol{V}^h \, \mathrm{d}Q = 0
\end{aligned} \tag{2}$$

Integration is performed over the space-time slab Q_n, the evolving spatial domain $\mathcal{D}(t)$ of boundary $\Gamma(t)$, and the surface P_n described by $\Gamma(t)$ as it traverses the time interval $I_n =]t_n, t_{n+1}[$. $\boldsymbol{W}^h$ and $\boldsymbol{V}^h$ are suitable spaces for test and trial functions, whereas τ and ν^h are appropriate stabilization parameters. $\tilde{\boldsymbol{A}}_0 = \partial \boldsymbol{U} / \partial \boldsymbol{V}$ is the metric tensor of the transformation from conservation to entropy variables and $\mathcal{L} \bullet = \tilde{\boldsymbol{A}}_0 \partial(\bullet)/\partial t + \tilde{\boldsymbol{A}}_i \partial(\bullet)/\partial x_i$ is the compressible Euler differential operator with $\tilde{\boldsymbol{A}}_i = \boldsymbol{A}_i \tilde{\boldsymbol{A}}_0$.

The shape functions are linear in space and constant in time. The resulting marching scheme is first-order accurate in time; it has excellent stability properties and is well-suited for solving steady problems using a local time-stepping strategy.

The adaption process is driven by the following simple error indicator[14]:

$$e_i = \frac{h^2 |\text{second derivative of } v|}{h |\text{first derivative of } v| + \varepsilon |\text{mean value of } v|} \tag{3}$$

where e_i is the error indicated at node i, h is a mesh size parameter, v is the solution variable being monitored, and ε is a tuning parameter. The second derivative of v is computed using a variational recovery technique. This particular indicator was chosen for its simplicity and acceptable performance, but clearly other more sophisticated solution error indicators or error estimators could be used.

A. Special Features for Rotorcraft Problems

Problems related to rotorcraft technology are of special interest for this research effort and can be more effectively solved by incorporating some simple features in a general code. For efficiently solving hovering rotor problems we have developed a TDG/LS formulation starting from the Euler equations written in a rotating frame. This allows one to treat a hovering rotor as a steady-state problem, thus lowering the computational cost. Details of this formulation are discussed in Refs. 4 and 15.

For a hovering rotor with n_b blades, the flowfield is periodic with angle of periodicity $\psi = 2\pi / n_b$. The periodicity conditions are treated as two-point constraints applied via transformation as part of the assembly process. This avoids the need to include Lagrange multiplier in the formulation. Details of this technique are given in Ref. 4. The algorithm requires the mesh discretizations on the two symmetric faces of the computational domain to match on a vertex-by-vertex basis. This is obtained with an ad hoc procedure that operates on the base mesh. When adaptive coarsening and refinement are applied to construct the multigrid levels, we explicitly ensure the matching of the discretizations of the two faces.

B. Boundary Conditions

The inflow/outflow far-field conditions are imposed following the methodology suggested in Ref. 15, where the one-dimensional helicopter momentum theory is used for determining the outflow velocity caused by the rotor wake system. The inflow velocities at the remaining portion of the far field are determined considering the rotor as a point sink of mass for achieving conservation of mass and momentum within the finite computational domain.

Nonreflecting boundary conditions based on the Riemann invariants for a one-dimensional flow normal to the boundary are also implemented and have been used in some simulations at the far field.

V. Multigrid Algorithm

We use a time-marching strategy for the unsteady Euler equations to reach steady-state. To overcome the time-step limitations of explicit schemes, we use implicit solution techniques. The underlying temporal discretization scheme is based on the first-order time discontinuous Galerkin method. Local time stepping based on the Courant–Friedrichs–Lewy (CFL) condition is used for enhancing convergence.

The code implements both a single-grid as well as a multigrid strategy. The single-grid algorithm is detailed in Ref. 4, and it is based on a quasi-Newton method equipped with a block-diagonally preconditioned GMRES solver. A matrix-free option is available for very large problems.

For the multigrid algorithm residuals are restricted using standard linear shape function weighting. Corrections are prolongated from a coarse grid to the next fine grid using linear shape functions.

For the approximation of the system Jacobian on coarse grids, we use the discretization coarse grid approximation method,[16] i.e., the Jacobian is calculated by discretization of the partial differential equations on each level. In our experience, this results in faster solution times than in the case where Jacobian restriction is used (Galerkin coarse grid approximation). To compute the coarse level Jacobians, the entropy field variables are injected from the fine level to the next coarse level. Boundary conditions are applied at the left- and right-hand-side operators at each coarse level, and they are treated exactly as in the finest grid case.

The algorithm implements V, W, and F multigrid schedules. Our numerical experience has shown that a simple V cycle with no postsmoothing generally yields the best compromise between performance and robustness. The number of smoothing sweeps on coarse grids is proportionally increased based on their distance from the fine level. A sketch of a typical time step is given in Fig. 1.

For smoothing we use the successive overrelaxation (SOR) scheme. The discrete system is preconditioned before each multigrid cycle by means of a nodal block-diagonal scaling transformation.

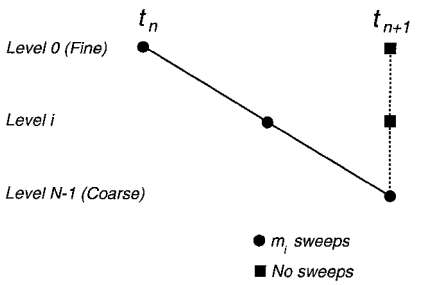

Fig. 1 Multigrid time-stepping scheme.

VI. Numerical Examples

A. Hovering Rotor in Transonic Regime

We consider the two-bladed model helicopter rotor studied experimentally by Caradonna and Tung[17] a classical test case for validating rotor simulation procedures. The same cases were considered in Ref. 4, using a parallel version of the adaptive single-grid code. The blades have rectangular planform, square tips, no twist or taper, NACA0012 airfoil sections, and an aspect ratio equal to 6.

We analyze the transonic flow case for collective pitch angle $\theta_c = 8$ deg and tip Mach number $M_t = 0.877$. We generated a first grid of 142,193 tetrahedra. Because of symmetry, only one blade was modeled, and periodic boundary conditions were applied to the symmetric faces. The grid was uniformly coarsened two times for obtaining a three-level multigrid that was used for computing a first solution. The code was run on this grid using the implicit multigrid algorithm with the V cycle and without postsmoothing. We used two, three, and four presmoothing sweeps on levels 0, 1, and 2, respectively, where level 0 denotes the finest discretization. The CFL number was set equal to 15. The time marching was arrested when a reduction of five orders of magnitude in the energy norm of the residual was achieved. Reduced (one-point) integration was used for the tetrahedral elements for lowering the computational cost, whereas full (three-point) integration was used at the boundary triangular elements for better resolution of the airloads, especially at the trailing edge of the blade.

The solution obtained this way was then used to compute the error indicator of Eq. (3) using density and Mach number. The error indicator drove the mesh adaption process, clustering more elements at the leading edge, at the region where shocks were present, and in the blade-tip region. A new multigrid hierarchy was created with the adapted mesh as fine grid and the previous base mesh as coarse level. The two grids obtained by uniform coarsening at the beginning of the process were then discarded.

For the new vertices created by the adaptation process, the solution was projected from the coarser mesh using simple edge interpolation. This was used as the initial condition for a new analysis. Using two and three presmoothing sweeps at levels 0 and 1, respectively, the time-marching procedure was restarted again with a CFL of 15 until the drop in the energy norm of the initial residual reached a value larger than four orders of magnitude. The adaption process was now repeated, obtaining a new refined mesh as level 0 of a three-level multigrid hierarchy. New elements were added to the shock region, raising the number of tetrahedra from 191,432 to 262,556. The intermediate and final meshes are virtually identical to those of Ref. 4. The time-marching process was again restarted using the same parameters until convergence. Figure 2 shows the final numerical values of the pressure coefficients compared with the experimental data. Good correlation with the experimental measurements is observed at all spanwise locations, including those close to the blade tip.

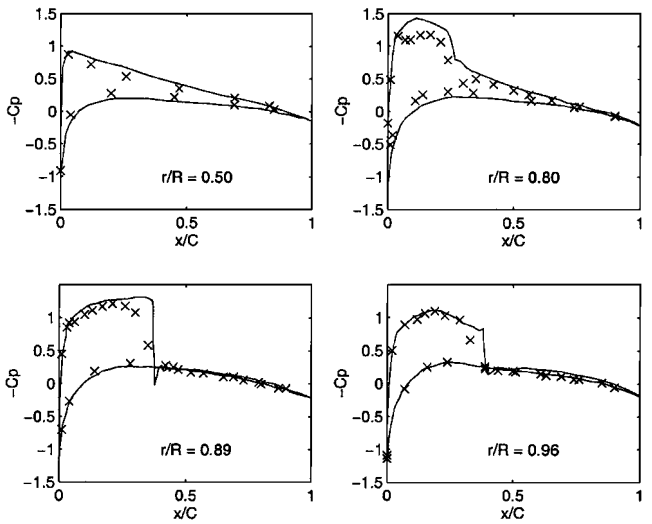

Fig. 2 Pressure distributions at different spanwise locations. Hovering rotor in transonic regime, $\theta_c = 8$ deg, and $M_t = 0.877$: ——, numerical simulation, and ×, experimental values.

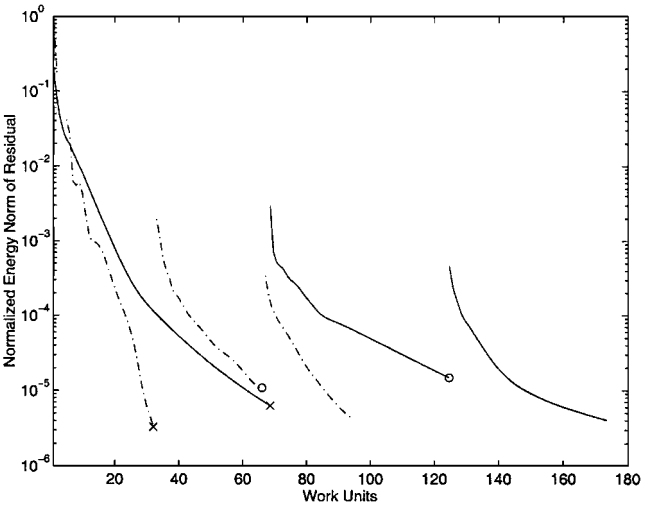

Fig. 3 Comparison of the convergence history in the energy norm of the residual for the single and multigrid algorithms. Hovering rotor in transonic regime, $\theta_c = 8$ deg, and $M_t = 0.877$: ——, single-grid algorithm; – ·–, multigrid algorithm; ×, first grid adaption; and ○, second grid adaption.

The problem was then solved using the adaptive single-grid algorithm. The same initial discretization and the same error indicators were used in order to obtain a similar sequence of meshes and hence essentially the same final results in terms of blade-pressure distributions. The convergence histories of the single and multigrid algorithms are compared in Fig. 3, where the energy norm of the residual is plotted against the number of work units (WU). In this work a WU is defined as the cost of a right- and of a left-hand side evaluation on the fine grid. The convergence histories of the single and multigrid algorithms are plotted using solid and dash-dotted lines, respectively. The first and second refinements are indicated with the symbols × and ○, respectively. The plot shows that the multigrid algorithm allows a reduction in computing time of approximately a factor of two. As expected, the two schemes show similar rapid convergence at the beginning of the solution process, but then the single-grid algorithm slows down as the smoother components of the error become predominant. This is exactly when the multigrid algorithm becomes more effective.

B. Wake Modeling by *h* Adaptivity

Among the test cases of Ref. 17, we selected the one denoted by $\theta_c = 8$ deg, $M_t = 0.439$, and traction coefficient $C_t = 0.00459$ for testing the ability of the code in performing wake modeling by *h* adaptivity. The goal of this exercise is to capture the vortical structures shed by the blades and their mutual interactions, without any ad hoc wake model but simply by performing *h* adaption on the mesh. The same problem was successfully solved using the single-grid algorithm, as presented in Ref. 3.

The initial mesh of 284,342 tetrahedra was used for obtaining an initial solution using the single-grid algorithm. A 3.5 order-of-magnitude reduction in the energy norm of the residual was obtained in 740 time steps. The solution was then used to compute an error indicator based on vorticity, with limiters on the minimum edge length in order to control the size of the mesh. The initial and adapted meshes gave a two-level multigrid. The final mesh is composed by 527,456 tetrahedra, mainly clustered at the tip of the blade and in the wake region.

The problem was also solved using the single-grid scheme, with the strategy already outlined. The convergence histories for the single- and two-level multigrid schemes are compared in Fig. 4, again in terms of WUs. The grid adaption is indicated with the symbol ×. The single and multigrid plots use solid and dash-dotted lines, respectively. For the single-grid algorithm about 530 implicit iterations at a CFL of 20 were performed on the adaptively refined mesh.

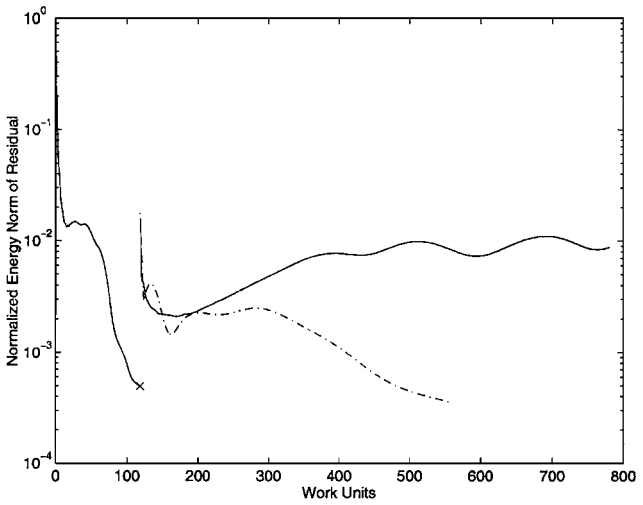

Fig. 4 Comparison of the convergence history in the energy norm of the residual for the single and multigrid algorithms. Hovering rotor in transonic regime, $\theta_c = 8$ deg, and $M_t = 0.439$: ——, single-grid algorithm; – ·–, multigrid algorithm; and ×, grid adaption.

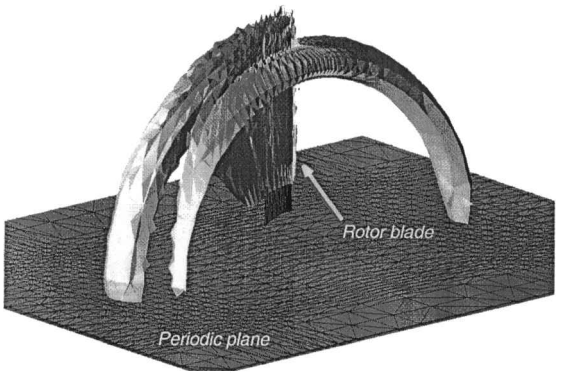

Fig. 5 Vorticity isosurface for the wake confinement problem. Hovering rotor, $\theta_c = 8$ deg, and $M_t = 0.439$.

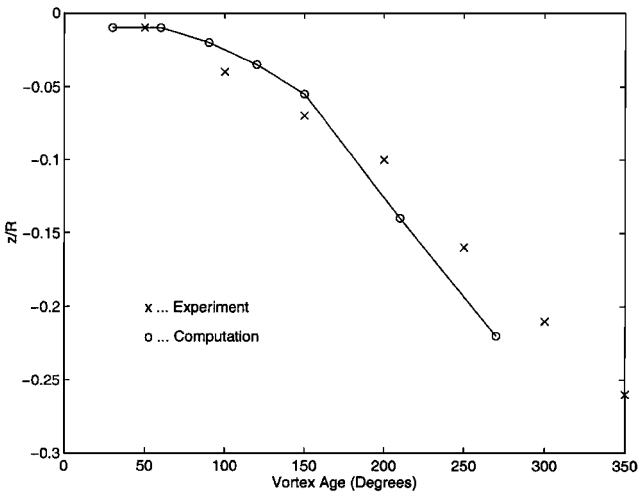

Fig. 6 Wake geometry for the wake confinement problem. Hovering rotor, $\theta_c = 8$ deg, and $M_t = 0.439$: ○, numerical simulation, and ×, experimental values.

The time stepping on the refined mesh for the single-grid algorithm is problematic, as clearly shown by the same figure, where even after a large number of implicit time steps the residual does not seem to be converging. As soon as the mesh starts to capture the vortical structures, the time-marching process is unable to escape a situation where the wake undergoes an undamped oscillatory motion of small amplitude. Clearly, even thought the results in terms of wake geometry are acceptable as shown later, the performance of the algorithm

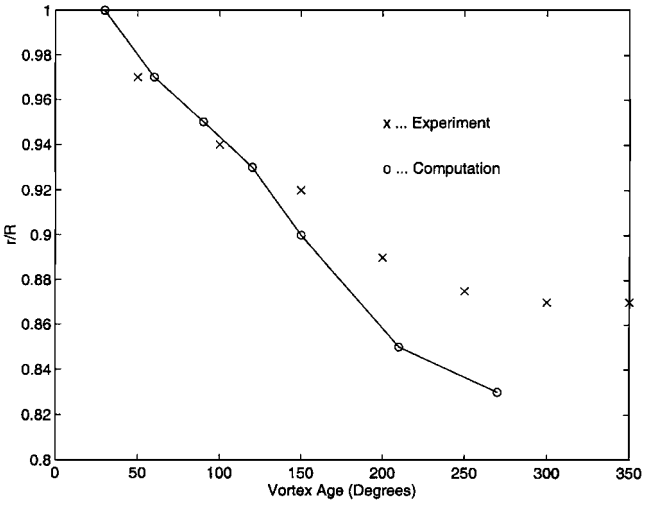

Fig. 7 Wake geometry for the wake confinement problem. Hovering rotor, $\theta_c = 8$ deg, and $M_t = 0.439$: ○, numerical simulation, and ×, experimental values.

is highly unsatisfactory in this case. The multigrid scheme does not seem to suffer from the same limitations, thus achieving a better, less problematic convergence.

Figure 5 shows an isosurface of vorticity obtained on the final adapted mesh. Figures 6 and 7 show a comparison of the wake geometry with the experimental data of Ref. 17. The pictures show the vertical position of the wake core, as well as the radial position of the core vs the azimuth angle. Both quantities show acceptable correlation with the experimental data, although better results could be achieved with finer meshes in the wake region, as shown in Ref. 3. Similar results are reported in Ref. 18, although a finer mesh of about three times as many elements was used in that case. The position of the wake core was determined by computing the centroid of the isosurface of Fig. 5. Even though the convergence histories for the single and multigrid schemes are somewhat different as already discussed, the results in terms of wake core geometry for the two solutions are very similar, and only the multigrid solution is reported.

C. Helicopter Noise Predictions

Coupled CFD–Kirchhoff procedures have been studied and applied to the prediction of helicopter noise.[19] CFD methods are used for simulating the acoustic field close to the rotor, whereas Kirchhoff formulations are responsible for efficiently propagating the acoustic signal to the far field. These procedures have yielded good results for rotors both in hover and forward flight in reasonable simulation times.

Although we have not interfaced our code with a Kirchhoff solver, we have tested its ability to model situations relevant to the simulation of high-speed impulsive noise of hovering rotors. To this purpose, we have considered a test case experimentally investigated in Ref. 20. The same problem was numerically simulated by a number of researchers, e.g., Ref. 9. The problem is that of a rectangular blade hovering rotor with NACA0012 airfoil sections and aspect ratio of 13.71, operating at a tip Mach number $M_t = 0.95$. The test case is characterized by a marked tip delocalization. The simulation is conducted for the nonlifting case; therefore, the problem is symmetric about the plane of the rotor, and the computational domain extends only on one side of the plane itself. The far-field boundaries are located at 1.5 radii above the plane and at 3 radii from the hub. Periodic boundary conditions are applied at the symmetric faces, while slip conditions are applied at the rotor disk plane to account for symmetry.

The adaptive analysis was conducted with four refinement levels, each followed by subsequent analysis to convergence. The multigrid solution strategy discussed in the preceding pages was used also in this case, resulting in one-, two-, three-, and four-level multigrids in the first, second, third, and fourth analysis, respectively. The

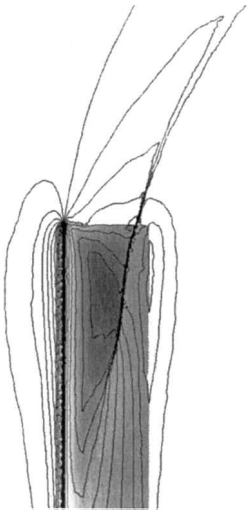

Fig. 8 Pressure distribution on the blade and on the plane of the rotor. Hovering rotor, $\theta_c = 0$ deg, and $M_t = 0.95$.

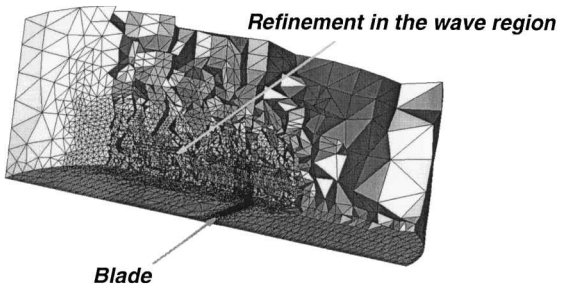

Fig. 9 View of the refined mesh, cut in the region of the acoustic wave. Hovering rotor, $\theta_c = 0$ deg, and $M_t = 0.95$.

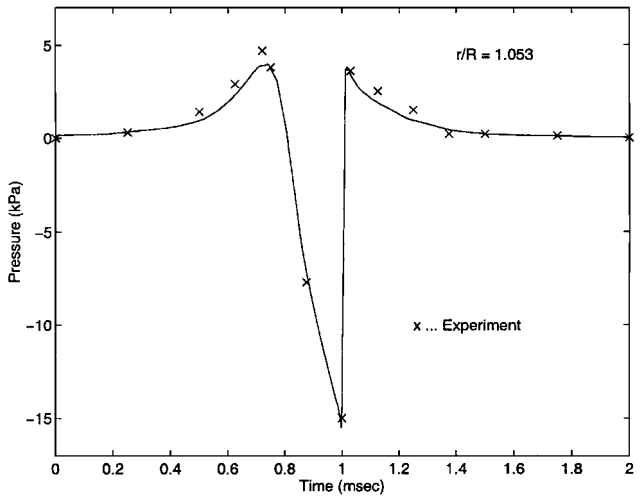

Fig. 10 Acoustic pressures. Hovering rotor, $\theta_c = 0$ deg, and $M_t = 0.95$: ——, numerical simulation, and ×, experimental values.

error indicator was based on the norm of the gradient of pressure with limiters on the minimum allowable edge lengths. The pressure distribution on the blade surface and on the plane of the rotor is presented in Fig. 8. Note that the shock wave is very well captured, yielding a very sharp pressure jump. Figure 9 shows a view of the fine grid. The mesh is cut in the region of the acoustic wave to show the three-dimensional refinement in this zone of high gradients. Figure 10 shows a comparison between the computed and experimental acoustic pressures at $r/R = 1.053$. The plots show good agreement for the wave shape. At higher values of r/R the correlation of the numerical and experimental data is less satisfactory, and finer meshes would be needed. The numerical results are similar to analogous Euler simulations conducted in Ref. 9, although in this case the correlation with experimental data remains good up to $r/R = 1.78$. This difference is probably because of the high degree of anisotropy of the mesh used in that work, which helps to reduce the number of necessary elements.

VII. Conclusions

We have developed a multigrid finite element algorithm for the time-marching implicit solution of the compressible Euler equations. The scheme is implemented on unstructured adaptive grids that allow the modeling of problems defined in complex domains and the capturing of the relevant features of the solution in an automatic way and to a desired level of accuracy.

The procedures were applied to the solution of problems related to hovering helicopter rotors. In particular, we have studied the pressure fields on the blades of rotors in transonic regime, the modeling of wakes, and the pressure distributions needed for hybrid CFD-Kirchhoff acoustic analyses. In all of these cases, the adaptive grids allowed the accurate automatic resolution of the important features of the solution.

We have compared the performance of the multigrid algorithm with a time-marching scheme based on a quasi-Newton method and a preconditioned GMRES iterative solver. We found that in all of these cases the multigrid scheme allows faster convergence to the steady solution, as expected. In particular, the multigrid algorithm seems to be able to overcome the convergence problems of wake modeling simulations. The multigrid solution is also quite robust with respect to the algorithm parameters, such as the type of cycle and the number of sweeps, which are virtually not problem dependent.

Acknowledgments

This research was funded by the U.S. Army European Research Office through contract N68171-97-C-9033 with Politecnico di Milano. This support is gratefully acknowledged. The authors would like to thank Ottmar Klaas, Mark W. Beall, Hugues L. De Cougny, and Pascal J. Frey for their many contributions to this work.

References

[1] Shephard, M. S., and Georges, M. K., "Automatic Three-Dimensional Mesh Generation by the Finite Octree Technique," *International Journal for Numerical Methods in Engineering*, Vol. 32, 1991, pp. 709–749.

[2] Shephard, M. S., "The Specification of Physical Attribute Information for Engineering Analysis," *Engineering with Computers*, Vol. 4, 1988, pp. 145–155.

[3] Bottasso, C. L., and Shephard, M. S., "Parallel Adaptive Finite Element Solution of Helicopter Rotors in Hover," 22nd European Rotorcraft Forum, Sept. 1996.

[4] Bottasso, C. L., and Shephard, M. S., "A Parallel Adaptive Finite Element Euler Flow Solver for Rotary Wing Aerodynamics," *AIAA Journal*, Vol. 35, No. 6, 1997, pp. 937–944.

[5] Shakib, F., Hughes, T. J. R., and Johan, Z., "A New Finite Element Formulation for Computational Fluid Dynamics: X. The Compressible Euler and Navier Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 89, 1991, pp. 141–219.

[6] Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," *Mathematics of Computation*, Vol. 31, 1977, pp. 333–390.

[7] Connell, S. D., and Holmes, D. G., "Three-Dimensional Unstructured Adaptive Multigrid Scheme for the Euler Equations," *AIAA Journal*, Vol. 32, No. 8, 1994, pp. 1626–1632.

[8] Parthasarathy, V., and Kallinderis, Y., "New Multigrid Approach for Three-Dimensional Unstructured, Adaptive Grids," *AIAA Journal*, Vol. 32, No. 5, 1994, pp. 956–963.

[9] Strawn, R., Biswas, R., and Garceau, M., "Unstructured Adaptive Mesh Computations of Rotorcraft High-Speed Impulsive Noise," *Journal of Aircraft*, Vol. 32, No. 4, 1995, pp. 754–760.

[10] Saad, Y., and Schultz, H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[11] Bottasso, C. L., Klaas, O., and Shephard, M. S., "Data Structures and Mesh Modification Tools for Unstructured Multigrid Adaptive Techniques," *Engineering with Computers*, Vol. 14, 1998, pp. 235–247.

[12] Beall, M. W., and Shephard, M. S., "A General Topology-Based Mesh Data Structure," *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 1573–1596.

[13] De Cougny, H. L., and Shephard, M. S., "Parallel Mesh Adaptation by Local Mesh Modification," Scientific Computation Research Center, Rensselaer Polytechnic Inst., Scientific Rept. 21-95, Troy, NY, 1995.

[14]Loehner, R., "An Adaptive Finite Element Scheme for Transient Problems in CFD," *Computer Methods in Applied Mechanics and Engineering*, Vol. 61, 1987, pp. 323–338.

[15]Srinivasan, G. R., Raghavan, V., and Duque, E. P. N., "Flowfield Analysis of Modern Helicopter Rotors in Hover by Navier–Stokes Method," International Technical Specialist Meeting on Rotorcraft Acoustics and Rotor Fluid Dynamics, Oct. 1991.

[16]Wesseling, P., "Introduction to Multigrid Methods," Inst. for Computer Applications in Science and Engineering, Rept. 95-11, NASA, Hampton, VA, Feb. 1995.

[17]Caradonna, F. X., and Tung, C., "Experimental and Analytical Studies of a Model Helicopter Rotor in Hover," USAAVRADCOM TR-81-A-23, Moffett Field, CA, Sept. 1981.

[18]Strawn, R., and Barth, T. J., "A Finite-Volume Euler Solver for Computing Rotary-Wing Aerodynamics on Unstructured Meshes," 48th Annual American Helicopter Society Forum, June 1992.

[19]Lyrintzis, A. S., "Review: the Use of Kirchhoff's Method in Computational Aeroacoustics," *Journal of Fluids Engineering*, Vol. 166, 1994, pp. 665–675.

[20]Purcell, T. W., "CFD and Transonic Helicopter Sound," 14th European Rotorcraft Forum, Sept. 1988.

J. Kallinderis
Associate Editor