Adaptive CFD Analysis for Rotorcraft Aerodynamics

Mustafa Dindar, Mark S. Shephard,
Joseph E. Flaherty and Kenneth Jansen

SCOREC Report #2-1999
Scientific Computation Research Center

Computer Methods in Applied Mechanics and Engineering, 189:1055-1076, 2000

Scientific Computation Research Center
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
voice 5182766795
fax 5182764886

ELSEVIER

Comput. Methods Appl. Mech. Engrg. 189 (2000) 1055–1076

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Adaptive CFD analysis for rotorcraft aerodynamics

Mustafa Dindar *, Mark S. Shephard, Joseph E. Flaherty, Kenneth Jansen

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

Received 19 May 1999

Abstract

An adaptive, parallel computational fluid dynamics (CFD) technique is developed to compute rotor-blade aerodynamics. For this purpose an error indicator is formulated and coded into an adaptive finite element framework. It is shown that the error indicator is effective in resolving the global features of the flow-field. Furthermore, for efficiency and problem size considerations, once the interpolation errors are reduced to acceptable levels, the adaptive refinement is done only in regions affected by the vortical flows. To accomplish this, a novel vortex core detection technique is used to capture vortex tubes. The combination of interpolation error estimate and vortex core detection technique proved to be very effective in computing vortical flow-field of rotor blades. Example adaptive, parallel calculations of hovering rotor blades, requiring 1–3 million tetrahedral elements, are presented.

Keywords: Rotorcraft; Aerodynamics; Tip vortex; Adaptive refinement; Vortical flows; Hover

1. Introduction

The goal of computational fluid dynamics (CFD) is to approximate the physical flows that we encounter in nature, as accurately as possible using numerical techniques. However, the cost of accuracy of any CFD procedure has to be matched by enough computer power and storage. Practical problems usually are the most complex. The flow-field around a complete aircraft, the internal flow structure of a turbojet engine with several stages and hundreds of blades and passages, the detailed effects of a wake and vortex flow generated by a helicopter rotor system on rotorcraft aerodynamics, aeroacoustic and aeroelasticity problems are some important examples that aeronautical engineers have to solve today and in the future. These large-scale problems require huge amounts of computer power and storage. Despite the steady growth in computer speed and power, the complexity of modern problems still requires the application of the most effective solution procedures.

Adaptive solution procedures running on parallel computers represent the most promising method to solve large-scale CFD problems. The main advantage of the parallel-adaptive procedure is its ability to distribute a large problem onto several CPU units, then locally adjust either the resolution of the computational mesh or increase/decrease the degree of the interpolating polynomials used in the variational formulation. The adaptive procedures can be classified into the following four categories: *h*-, *p*-, *hp*- and *r*-adaptive. Both *h*- and *r*-adaptive procedures are based on modifying the computational mesh based on an error estimate to increase/decrease resolution accordingly. Whereas *p*-adaptive scheme modifies the degree of the interpolating polynomials in the finite element formulation again guided by a measure of the

* Corresponding author. Present address: General Electric Corporate Research & Development, One Research Circle, Room 3C2, Building K1, Niskayuna, NY 12309, USA.

0045-7825/00/$ - see front matter
PII: S0045-7825(99)00368-0

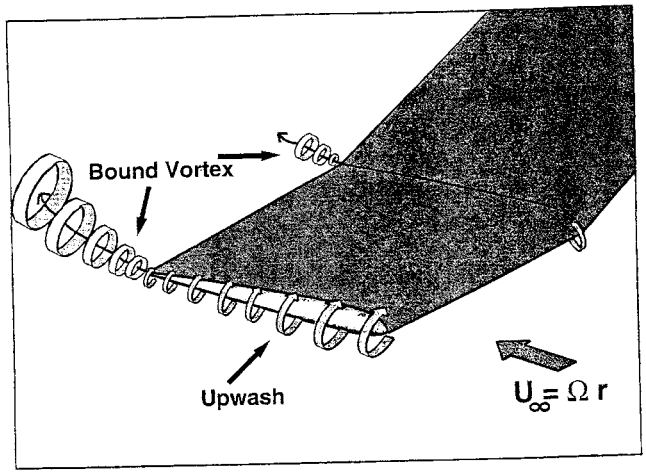

Fig. 1. A schematic of blade tip vortex formation.

computational error [9,10]. Alternatively, an *hp*-adaptive method is a hybrid of *h*- and *p*-adaptive procedures [2–4].

In an *h*-adaptive process, the main idea is to subdivide (refine) the computational mesh into smaller elements to increase the resolution or collapse (de-refine) a group of small elements into one or more larger elements when less resolution is needed. The decision to refine or de-refine is based on the local measure of the computational error. The areas that have errors higher than some allowable tolerance are refined and the areas that have lower error measures are de-refined. Examples of *h*-adaptive calculations for CFD problems can be found in [18,21,22].

Accurate prediction of rotorcraft aerodynamics using existing CFD tools is still a challenge. This is because a rotorcraft often operates in more severe flow conditions than a fixed-wing aircraft. For example, an isolated rotor blade system in hover conditions is under the influence of its own wake. Rotor blade performance calculations become a complex task for CFD due to strong blade and tip vortex interactions. The blade tip is responsible for producing the tip vortex because of the up-wash flow due to pressure difference between upper and lower surfaces, as shown in Fig. 1. An accurate representation of the tip vortex is needed to be able to predict the thrust loading at the outboard portions of a rotor blade.

One of the early applications of adaptive solution techniques for rotor-blade flows is due to Strawn et al. [22] where they analyzed a two-bladed rotor system in hover using an unstructured grid that was obtained by tessellating the hexahedral cells of a finite difference grid to linear tetrahedral elements. During the solution process they used vorticity magnitude as the error indicator in the flow-field to resolve the wake and the tip-vortex structures. Although vorticity magnitude can be used as an indicator of wake and vortical flow regions, it does not reflect the error in the solution. One possible avenue in designing an error indicator for flow problems is to use the interpolation error estimates [20]. In this paper, we will describe an *h*-adaptive procedure for computing rotor-blade wakes in the framework of a parallel Galerkin/least-squares finite element formulation [5,15].

2. Governing equations

Governing equations of an inviscid flow are given by the Euler equations, which can be written in conservative quasi-linear form as follows:

$$U_{,t} + A_i U_{,i} = \mathscr{R}. \tag{1}$$

In Eq. (1), $U = [\rho, \rho u, \rho v, \rho w, \rho e]$ is the vector of conservative variables, where ρ is mass density, u, v and w are fluid velocity components and e is the total energy per unit mass. A_i is the ith inviscid flux vector and $\mathscr{R} = [0, \rho v\Omega, -\rho u\Omega, 0, 0]$ is the source term to account for the rotating frame analysis about the z-axis.

The inviscid flux vector in Eq. (1), A_i, is non-symmetric when written in terms of conservative variables. A symmetrization of inviscid flux vectors can be achieved by performing a transformation using a change of variables $U \rightarrow V$ [15]:

$$V = \frac{\partial H}{\partial U}, \tag{2}$$

where $H(U) = -\rho(s - s_0)$ is the generalized entropy function. Details of this transformation procedure can be found in Ref. [7]. After this transformation, Eq. (1) can be re-written as

$$\tilde{A}_0 V_{,t} + \tilde{A}_i V_{,i} = \mathscr{R}, \tag{3}$$

where $\tilde{A}_0 = \partial U/\partial V$ and $\tilde{A}_i = A_i \tilde{A}_0$. Note that $\tilde{A}_0$ is a symmetric positive-definite matrix, and $\tilde{A}_i$ is a symmetric matrix.

A stabilized finite element formulation of Eq. (3) can be obtained using time-discontinuous Galerkin least-squares method [15]. All the calculations in this paper are done using a Galerkin least-squares method, and the GMRES [23] solution technique is used as a linear equation solver.

3. Error estimation/indication

Understanding the flow fields with vortical structures and adaptive mesh refinement requires an understanding of finite element approximation. Interpolation errors exist due to finite dimensional space approximation and depend on the order of the finite element basis.

Let us seek L_2-norms of errors of the vector field $\vec{u}$ in a finite element procedure which utilizes piecewise linear polynomials in R^3. The semi-norm of the interpolation error is given by [20]

$$|\varepsilon^h|_{0,2} \leqslant Ch^2 |\vec{u}^h|_{2,2}, \tag{4}$$

where

$$|\mathscr{F}|_{m,p} \equiv |\mathscr{F}|_{L_p^m(\Omega)} = \left(\int_\Omega \sum_{|\alpha|=m} |D^\alpha \mathscr{F}|^p \, \mathrm{d}\Omega \right)^{1/p} \tag{5}$$

and

$$D^\alpha \equiv \frac{\partial^{\alpha_1+\alpha_2+\alpha_3}}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \partial x_3^{\alpha_3}}, \quad \alpha_1, \alpha_2, \alpha_3 \geqslant 0. \tag{6}$$

In Eq. (4), evaluation of the error requires second partial derivatives of $\vec{u}^h$. Let us denote the finite element error indicator by the symbol ϵ_i, then a practical error indicator using the L_2-norm of the second derivatives of the velocity vector can be written as

$$\epsilon_i = h^2 \left(\int_{\Omega^h} \sum_{i+j+k=2} \left| \frac{\partial^2 \vec{u}_i^h}{\partial x_1^i \partial x_2^j \partial x_3^k} \right|^2 \mathrm{d}\Omega^h \right)^{1/2}. \tag{7}$$

Note that Eq. (7) requires the evaluation of second order derivatives which can be achieved by a gradient recovery (smoothing) procedure given as follows:

$$\nabla_r \vec{u}^h = \sum_{\Omega^h \ni P} \frac{\mathscr{V}_{\Omega^h}}{\mathscr{V}} \nabla \vec{u}^h. \tag{8}$$

Here, Eq. (8) constructs a piecewise linear distribution of $\nabla_r \vec{u}$ that is exact for a *quadratic* solution and is $O(h^2)$ in general [14,26]. Note that Eq. (8) is simply a weighted average of $\nabla \vec{u}$ on a polytope of elements P, where $\mathscr{V}$ and $\mathscr{V}_{\Omega^h}$ are the volume of the polytope and element Ω^h, respectively. After recovering the first

derivatives, the second derivatives can be approximated using the derivatives of interpolation functions into $\nabla_r \vec{u}$ on Ω^h.

Another approach to evaluate second derivatives of flow variables on linear elements is to re-construct the solution data using a quadratic polynomial as a basis function [14]:

$$\vec{u}(x,y,z)_0 = \vec{u}_0 + \Delta r^{\mathrm{T}} \nabla \vec{u}_0 + \frac{1}{2} \Delta r^{\mathrm{T}} H_0 \Delta r + \mathrm{O}(\Delta r^3), \tag{9}$$

where $\Delta r = x_i - x_0$ and H_0 is the Hessian matrix of second derivatives:

$$H_0 = \begin{bmatrix} \vec{u}_{xx} & \vec{u}_{xy} & \vec{u}_{xz} \\ \vec{u}_{yx} & \vec{u}_{yy} & \vec{u}_{yz} \\ \vec{u}_{zx} & \vec{u}_{zy} & \vec{u}_{zz} \end{bmatrix}. \tag{10}$$

Given the solution data at the vertices of the mesh, we seek to minimize the L_2 norm of the distance between $\vec{u}$ sampled at the vertex v_0 and the element quadrature points. For a polytope of elements surrounding vertex v_0, this operation yields a non-square matrix problem of the form

$$\begin{bmatrix} L_1^1 & L_1^2 & \cdots & L_1^8 & L_1^9 \\ L_2^1 & L_2^2 & \cdots & L_2^8 & L_2^9 \\ L_3^1 & L_3^2 & \cdots & L_3^8 & L_3^9 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ L_i^1 & L_i^2 & \cdots & L_i^8 & L_i^9 \\ \vdots & \vdots & \cdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vec{u}_x \\ \vec{u}_y \\ \vec{u}_z \\ \vec{u}_{xy} \\ \vdots \\ \vec{u}_{zz} \end{bmatrix} = \begin{bmatrix} \Delta\vec{u}_1 \\ \Delta\vec{u}_2 \\ \Delta\vec{u}_3 \\ \vdots \\ \Delta\vec{u}_i \\ \vdots \end{bmatrix}, \tag{11}$$

where L_i is a row vector defined as

$$L_i = \begin{bmatrix} \Delta x_i & \Delta y_i & \Delta z_i & \frac{1}{2}\Delta x_i^2 & \cdots & \Delta y_i \Delta z_i & \frac{1}{2}\Delta z_i^2 \end{bmatrix} \tag{12}$$

and $\Delta\vec{u} = \vec{u}_{\Omega^h} - \vec{u}_0$.

Solution of this non-square matrix equation can be obtained by using a least-squares solver such as the singular value decomposition (SVD). Note that, regardless of which method we use to calculate the second derivatives, they are going to be *constant* for linear element. Therefore, Eq. (7) can be further simplified and written in the following form:

$$\epsilon = h^2 \sqrt{V} \left(\left| \frac{\partial^2 u}{\partial x^2} \right|^2 + \left| \frac{\partial^2 u}{\partial x \partial y} \right|^2 + \left| \frac{\partial^2 u}{\partial x \partial z} \right|^2 + \left| \frac{\partial^2 u}{\partial y^2} \right|^2 + \left| \frac{\partial^2 u}{\partial y \partial z} \right|^2 + \left| \frac{\partial^2 u}{\partial z^2} \right|^2 \right)^{1/2}. \tag{13}$$

Note that the true interpolation error is $|\varepsilon|_{0,2}$, whereas Eq. (13) measures $|\varepsilon|_{0,2}/C$. Therefore, unless C is unbounded, the error indicator should measure qualitatively the right convergence rate in the L_2 semi-norm of u and it should be of $\mathrm{O}(h^2)$, in general.

4. Two-level adaptation procedure for vortical flows

So far, the sources of interpolation error have been identified by an error indicator. It is clear that for a finite amount of mesh resolution, there will always be some error in the computational domain. Although, in principle, one could use the error indicator to drive the adaptation to resolve all the features of a flow problem, because of computational efficiency and storage limitations, this would not be practical. Therefore, it is desirable to monitor the global accuracy during the adaptive solution procedure and, if permissible, resort to more localized adaptation of the mesh for small-scale features of specific interest after sufficient global resolution is obtained.

4.1. A definition for a vortex core

From the topological point of view, a vortex core contains features that can be used to distinguish it from other regions of the flow-field. First, a vortex core is a stationary point where flow trajectories spiral in a plane and the vorticity is maximal at the core center. It is possible to define and distinguish a vortex core by looking at the velocity vector field. Let $\vec{u}$ represent the velocity field in and around a vortex core. When $\vec{u}$ is expanded into its Taylor series around the vortex core point P_0, we have

$$\vec{u}(x,y,z) = \vec{u}_0 + \Delta r^{\mathrm{T}} \nabla \vec{u} + \mathrm{O}(\Delta r^3) \tag{14}$$

For a real fluid, viscous effects cause the core of a vortex to rotate approximately as if it were a rigid body, hence, on the plane of rotation $\vec{u}_o \to 0$ at the vortex core [25]. For inviscid flows with the existence of numerical diffusion, a vortex behaves much like it is in a viscous flow. Therefore, we identify a vortex core to be a stationary point. The second term in Eq. (14), the velocity-gradient tensor, $\nabla \vec{u}$, has complex eigenvalues if the stationary point is a vortex core. The velocity-gradient tensor can be disassembled into a rotation tensor and a strain-rate tensor. At a vortex core, the rotation tensor dominates over the strain-rate tensor [8]. This approach has been used recently to define and extract flow topology information for scientific visualization [16].

Based on the velocity-gradient tensor,

$$\nabla \vec{u} = \begin{bmatrix} \partial u/\partial x & \partial u/\partial y & \partial u/\partial z \\ \partial v/\partial x & \partial v/\partial y & \partial v/\partial z \\ \partial w/\partial x & \partial w/\partial y & \partial w/\partial z \end{bmatrix}, \tag{15}$$

we have three eigenvalues. For these eigenvalues, there are only two possibilities: (1) all real eigenvalues, and (2) real and complex-conjugate eigenvalues combined. Abraham and Shaw [1] classify the behavior of a dynamic system based on the characteristic eigenvalues. Namely there are three types: (1) nodes, where all the eigenvalues are real and positive, (2) saddle, where all the eigenvalues are real and negative and (3) spiral saddle, where both real and imaginary eigenvalues exist. Clearly a vortex core is said to exist at a stationary point with eigenvalues forming a spiral saddle. It is known [8] that the magnitude of the imaginary part of a complex-conjugate pair represents the strength of the vortex. We should also try to represent the core diameter of a vortex. Since at the core vorticity reaches a maximum, then a possible definition of a core diameter can be made by limiting the magnitude of the vorticity at the outer surface of the vortex tube. In practical applications it is found that an acceptable definition of the core diameter is to extend the core up to a point where the magnitude of the vorticity drops below 10% of the vorticity magnitude at the center of the core of the vortex. Clearly this loose definition can only detect isolated vortex tubes, better definitions should be found for more complex situations.

The topological features of a vortex core can be used to predict the vortex tube in numerical calculations. Numerical implementation of the vortex core detection technique used here follows from [16]. All tetrahedral elements are passed through an eigenvalue test on the velocity-gradient tensor. If the velocity-gradient tensor for the element happens to have both real and imaginary eigenvalues, then that element is flagged as a possible vortex core region. This information is passed to the adaptive refinement procedure so that the tetrahedral element which contains the vortex core can be refined appropriately. This simple implementation of the eigenvalue extraction technique works quite nicely as an indicator for resolving vortex tubes. We should note here that the eigenvectors that correspond to complex-conjugate eigenvalues span a plane perpendicular to the vortex core, whereas the eigenvector of the real eigenvalue is directed along the trajectory of the vortex tube.

In our numerical examples we use the eigenvalue extraction technique to refine the mesh locally. To do this, the error indicator based on an interpolation error estimate is modified such that

$$\epsilon^* = \begin{cases} \epsilon & \text{if a vortex exists on } \Omega^h, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

The mesh in the vortex tube is refined until the error tolerance drops to an acceptable level so that the vortex core is resolved accurately.

5. Numerical examples

We present parallel, *h*-adaptive numerical examples using the error indicators presented in previous sections. The application problems are helicopter rotor blades at steady-state hover conditions. Two different rotor blade configurations are presented with varying levels of numerical difficulties.

5.1. Carradona–Tung blade

As a first example, we will compare finite element analysis results with test data on a rotor blade tested by Caradonna and Tung [6]. This two bladed rotor uses the NACA0012 airfoil section, it is untwisted and untapered. The aspect ratio of the blade is 6. The flow case analyzed for the Caradonna–Tung blade is at a tip Mach number $M_t = 0.439$, thrust coefficient $C_T = 0.00459$, collective setting $\theta_c = 8°$ and rotor speed of 1250 rpm. For computational efficiency, only one blade of the rotor system is modeled. The surface definition of the blade is generated using a solid modeling system. The computational outer boundaries are approximated by enclosing the blade in a finite radius cylinder. To avoid reflection of boundary conditions, the radius of the cylinder is chosen to be three times the blade radius and the blade is placed 1.5 and 2 radii away from top and bottom surfaces of the cylindrical domain, respectively. An initial mesh for this geometry is generated by using an automatic mesh generation program [13]. The initial mesh contains 214,000 linear tetrahedral elements. The geometric model and the initial mesh of the Caradonna-Tung blade are shown in Fig. 2. The inflow and outflow boundary conditions of a hovering blade are applied according to momentum theory and 3D potential sink [24]. Starting from an initial mesh, a series of results have been obtained using an automatic adaptive refinement procedure. The initial mesh is distributed over 16 processors of an IBM SP2 parallel machine. At each adaptive step, the solution is interpolated from the previous adaptive level and the program run until the adapted solution reaches steady state. The finite element procedure typically required 200–300 GMRES cycles to converge, where each cycle uses a Krylov space of size 10.

One potential problem with running a parallel, *h*-adaptive, large CFD problem on a distrubuted memory parallel computer is the available physical memory of each CPU unit and the ability of the finite element framework to use this memory efficiently and effectively. When an adaptive refinement is performed on a given mesh, it is quite likely that some of the processors refine much more than others. For a given physical memory of a CPU unit, which is 128 MB for the SP2 machine on which this problem is run, we are limited to create not more than 80,000 tetrahedral elements. In our adaptive refinement experiences with this and all other example cases reported here, it was extremely difficult to perform an autonomous mesh refinement step without manually limiting the number of newly created elements for a given processor. Although some of the processors created no new elements, other processors created large numbers of elements which eventually halted the entire adaptive procedure because of insufficient physical memory. An example of this

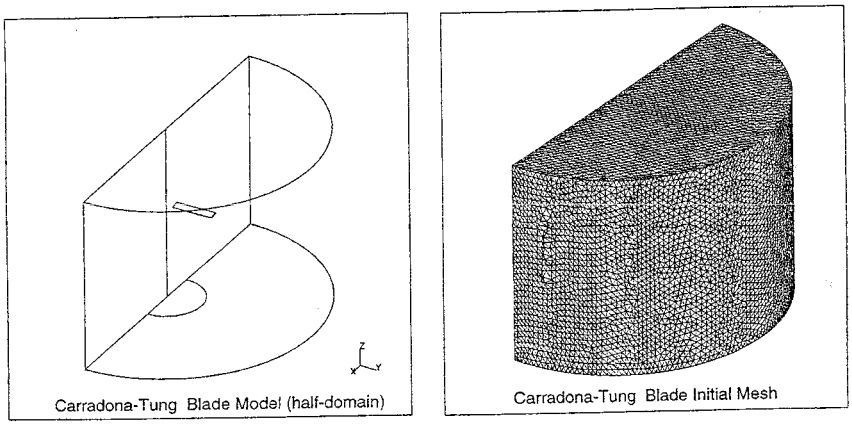

Fig. 2. Geometric model and the initial mesh for the Caradonna–Tung blade.

unsuccessful case is shown in Fig. 3(a), where processors 3, 4 and 5 create excessive number of elements than the allowable. To alleviate the unbalanced usage of memory per processor, a predictive load balancing procedure [11,12] has been incorporated into the adaptive refinement procedure. This predictive load balancing scheme works with the adaptive refinement procedure and distributes tetrahedral elements, before they are split, across processors based on refinement load level, thus reducing the possibility of unbalanced physical memory usage per processor. A successful adaptive run with predictive load balancing is shown in Fig. 3(b). In Fig. 3(b), using the estimated load prior to refinement, the final adapted mesh is distributed equally over the processors.

As stated before, the adaptive procedure is performed using an interpolation error estimate along with a vortex core detection technique. The order of refinement strategies has been selected by looking at the global and local quality of the solution in comparison to experimental data. For example, after two levels of refinement with error indicator based on interpolation estimate of Eq. (13), the pressure distribution on the blade surface showed significant improvement with respect to the initial mesh. Then we examined the state of the tip vortex and used the vortex core detection technique to locally refine and enhance the resolution near the tip of the blade. Given a rotor blade flow case, we started the adaptation procedure first using the Eq. (13) to refine the mesh globally. In the consecutive adaptive levels, we used Eq. (16) to refine the mesh only in regions effected by the blade tip vortex. In general, the rotor blade flow cases presented in this paper required 1 and 2 levels of global refinement using Eq. (13) and 2 and 3 levels of local refinement using Eq. (16).

Fig. 4 shows the changes in mesh resolution on the blade surface with the adaptive refinement levels. The initial mesh, which contains 214,000 elements is shown as level 0. In adaptive level 1, to reduce the global errors and enhance the quality of solution near the blade, the indicator given in Eq. (13) is used to compute the error and refine the mesh. Notice that, in level 1 the mesh resolution on the blade surface is almost doubled. In adaptive level 2, Eq. (16) is used to bracket the adaptive refinement to the vicinity of the tip vortex. As shown in Fig. 4, the mesh resolution on the blade surface increased only near the leading and trailing edges of the blade tip. This step also shows that the upwash from the lower surface of the blade to the upper surface creates a rotating flow which triggers the vortex core detection process to indicate that this region needs refinement. This adaptive process is repeated two more levels to resolve both the tip vortex and flow quality near the leading edge of the blade. At the end of adaptive level 4 the mesh size reached to 2,215,000 tetrahedral elements. Also note that the mesh resolution increased at the leading edge of the blade where the flow stagnates. To be able to show the mesh refinement process in the interiors of the computational domain, a faceted slice, which is 15° behind the trailing edge, is taken from mesh and the progress of the adaptation is shown in Fig. 5. It is clear from Fig. 5 that for refinement levels 3 and 4, most of the mesh adaptation takes place near the tip vortex.

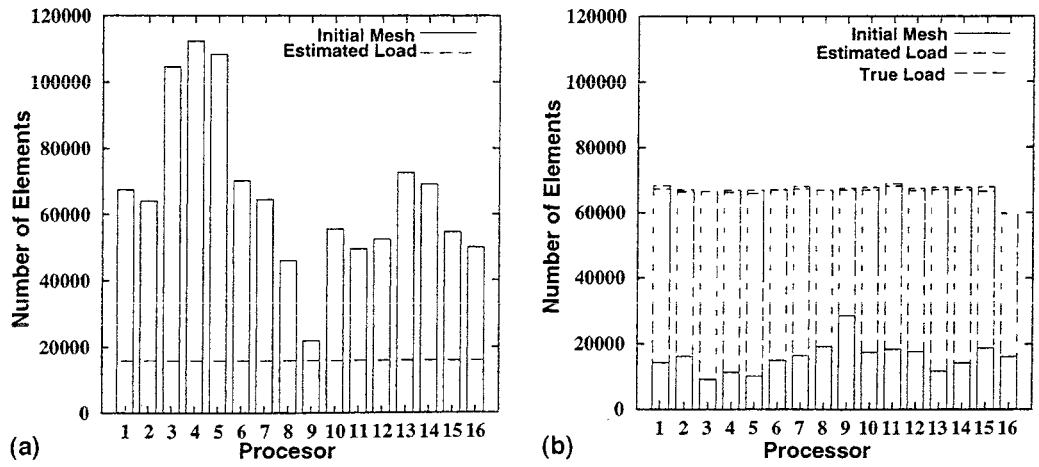

Fig. 3. Adaptive refinement without and without predictive load balancing (PLB): (a) without PLB; (b) with PLB.

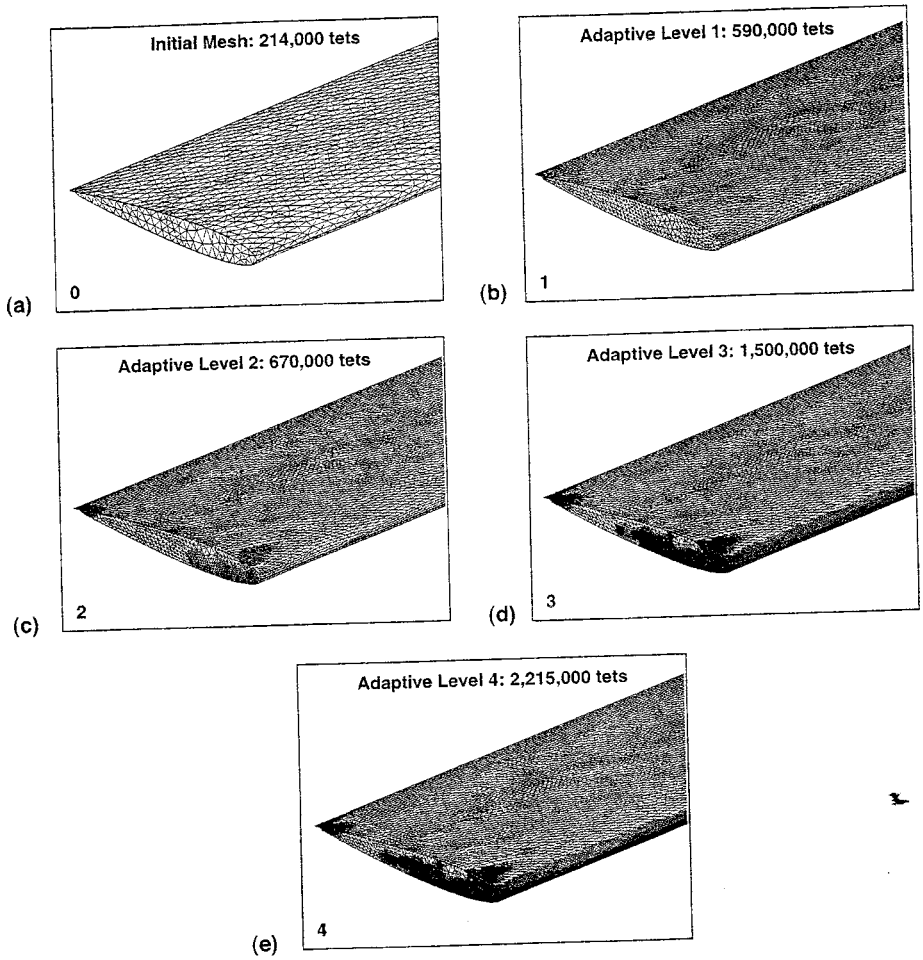

Fig. 4. Adaptive refinement levels for Caradonna–Tung blade: surface meshes.

Computed coefficient of pressure distributions at 5 radial locations are plotted in Fig. 6. When compared against experimental data, pressure distributions computed by adaptive procedure show noticeable improvement in comparison to initial mesh results. Especially, the suction peaks and the upper surface pressure plateau show remarkable improvement with adaptation.

Finally, to show the effect of adaptive refinement on the resolution of the tip vortex, tip vortex structure is rendered for initial mesh, and adaptive levels 2 and 4 in Fig. 7. Notice that with the adaptation a $180^{c}irc$ revolution of the tip vortex is captured successfully.

5.2. UH-60A blackhawk blade

As a second case, we selected the four bladed rotor system of the UH-60A Blackhawk. Due to periodicity of the flow-field, only one blade of the rotor system is considered. The geometric dimensions and features of the blade are taken from the scale model (1 : 5.73) used in the experiment discussed in [19]. Two airfoil profiles, SC-1095 and SC-1095R8, are used to construct the surface definition of the blade using a solid modeling system (Fig. 8). The built-in twist of the blade, which varies linearly over the first 80% radius of the blade and has a hook-type non-linear twist near the tip, is included in the model. Also, elastic twist of the blade is incorporated into the model using the measured deflections of the blade from reference [19]. An example blade pitch distribution, which at this point contains built-in twist, collective,

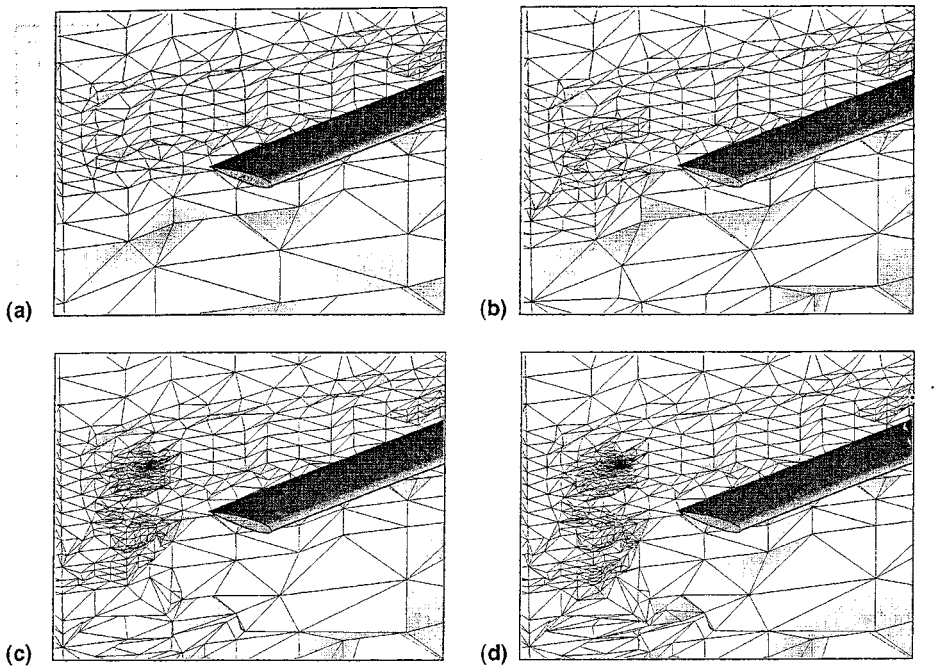

Fig. 5. Adaptive refinement levels for Caradonna–Tung blade: interior: (a) first level: 590,000 tets; (b) second level: 670,000 tets; (c) third level: 1,500,000 tets; (d) fourth level: 2,215,000 tets.

and elastic twist is plotted against non-dimensional radius and compared with experimental measurements in Fig. 9. The flow-field of the UH-60A rotor is very interesting due to the swept-tip and non-linear twist. Two flow-field conditions are chosen to study the adaptive solution procedure of this blade. The first case studied with the UH-60A is at zero thrust and the second case is at high thrust (design) conditions. Both of these cases are reported to have neither separation nor stall in the experimental results [19].

5.2.1. UH-60A at zero thrust

The flow-field conditions used for this case are as follows: tip Mach number $M_t = 0.628$, rotor speed $\Omega = 1427$ rpm, thrust coefficient $C_T/\sigma = 0.0$, collective setting $\theta_{0.75} = 0.11°$ and coning $\beta_0 = -0.20°$.

Before doing any adaptive solutions, a mesh sensitivity study is done to establish initial element size that can be used as a starting point. First, a mesh containing 760,000 tetrahedral elements (see Fig. 10), is used to calculate the flow-field. For this mesh the average global element size is about 0.125R and it is more or less uniform throughout the domain with the exception of the blade surface which has a finer (down to 0.001R) mesh resolution than the rest of the domain. On the blade surface, the mesh is clustered near the leading edge and trailing edge regions to capture the stagnation pressure correctly. Second, a finer mesh, which is refined near the tip-path-plane of the blade, is generated (see Fig. 11). To create this mesh, a slice of the computational domain that encloses the blade is targeted for refinement. The initial size of this second mesh is 1,100,400 tetrahedral elements. The average element size in the refined domain is about 0.002R. Computed pressure distributions at selected radial locations on the blade surface are compared against experimental data [19] in Fig. 12. There comparisons reveal that the uniform (coarse) mesh did not capture some of the key aspects of the flow-field as observed in experiment. As anticipated, results with the finer mesh are considerably more accurate than the coarse mesh results. Although the fine mesh does not resolve all the features of the flow-field, it is a good starting point for adaptive refinement. Next, the adaptive procedure is applied to the fine initial grid results to improve the solution further and to locate the vortex structure. With the adaptive procedure, the mesh is refined in two levels, the first level using the

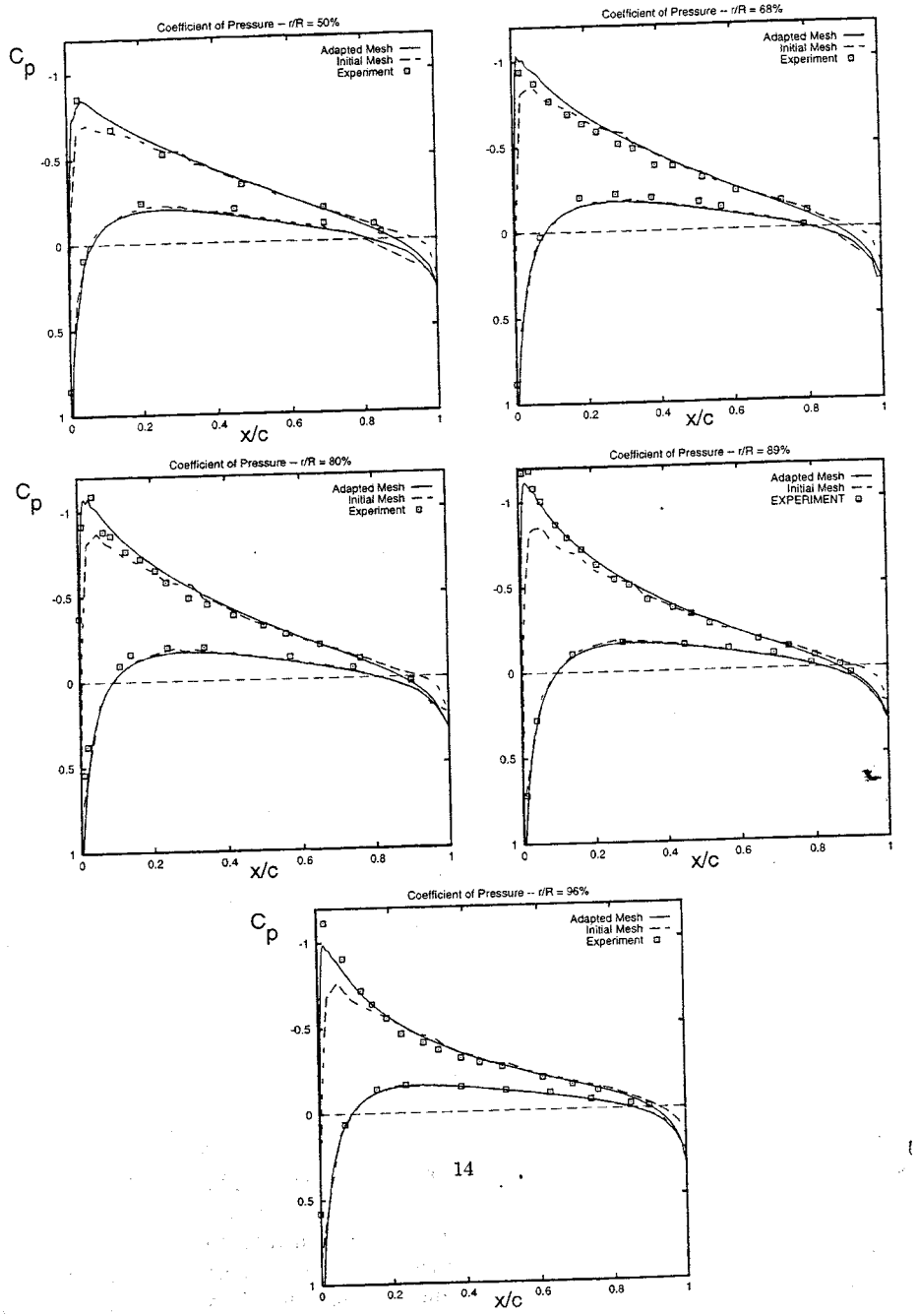

Fig. 6. Comparison of computed pressure coefficient with experimental data: Caradonna–Tung blade, $M_t = 0.439$, $C_T = 0.00459$ and $\theta_c = 8°$.

interpolation error estimate and the second level using the vortex core detection technique. At this point, the mesh size reached 2,164,704 elements.

Computed pressure coefficient distributions are compared with the experiment and the coarse initial grid in Fig. 12. The suction peaks for radial stations 55%, 67.5% and 77.5% are missed by the computational results. Further refinement of the mesh near the leading edge of the blade is required to

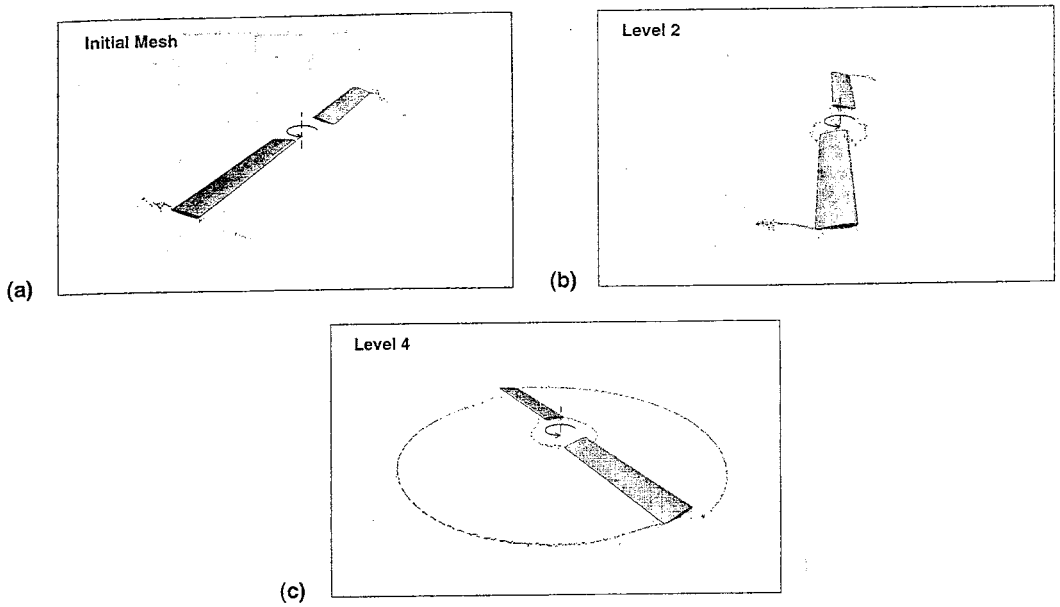

Fig. 7. Progress of tip vortex geometry with adaptive refinement: Caradonna–Tung blade.

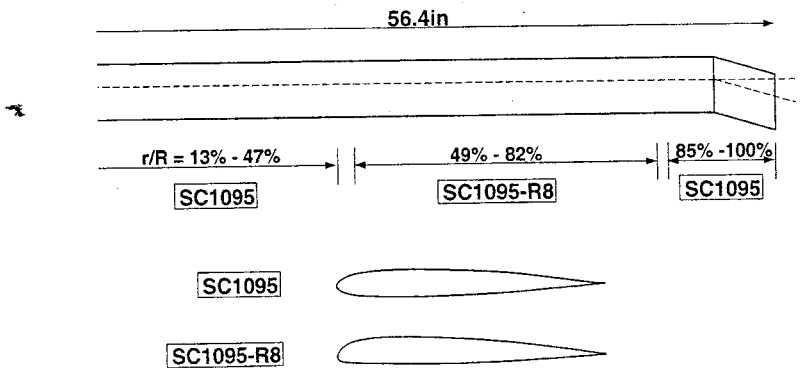

Fig. 8. UH-60A Blackhawk rotor blade planform.

improve the current results. However, overall quality of this case with the adapted mesh is considered satisfactory.

Figs. 13 and 14 show computed sectional thrust and torque distributions, respectively. Notice that the adapted grid enhances the accuracy of the thrust distribution in the neighborhood of the inflection point near the tip. The integrated value of C_T/σ for the adapted mesh is 0.00039.

Finally, Fig. 15 shows the predicted vortex flow structure which is calculated by the vortex core detection technique. Note that there is an unexpected vortex tube located between 75% and 80% radius of the blade. This vortex tube is independent of the tip vortex. The reason for the existence of a vortex tube at 75% radius is hypothesized to be the differential change in thrust loading from positive to negative as shown in Fig. 13. The inboard shed wake, in this case, should have a discontinuity around 80% radial span. Existence of this inboard vortex tube clearly demonstrates a deviation from empirical wake geometry studies [17] where the inboard shed wake is assumed continuous between the root and the tip of the blade. Finally, there is also a vortex tube emanating from the tip of the blade, but it does not seem to have a very strong interaction with the blade tip.

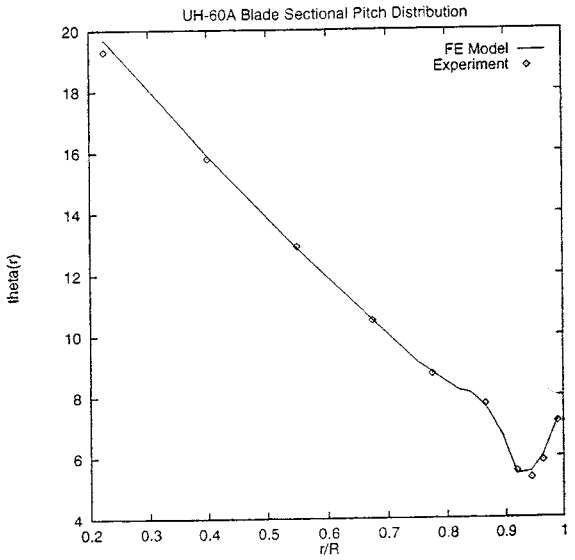

Fig. 9. UH-60A blade pitch distribution.

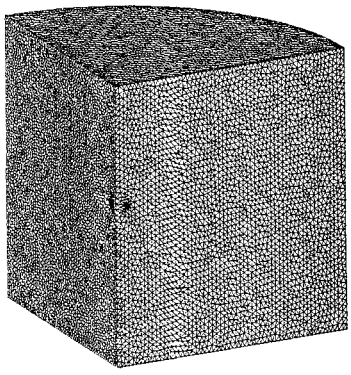

Fig. 10. Coarse initial mesh for UH-60A blade at zero thrust.

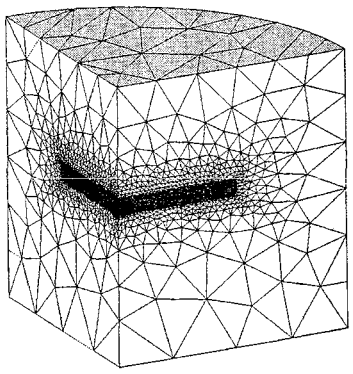

Fig. 11. Fine initial mesh for the UH-60A blade at zero thrust.

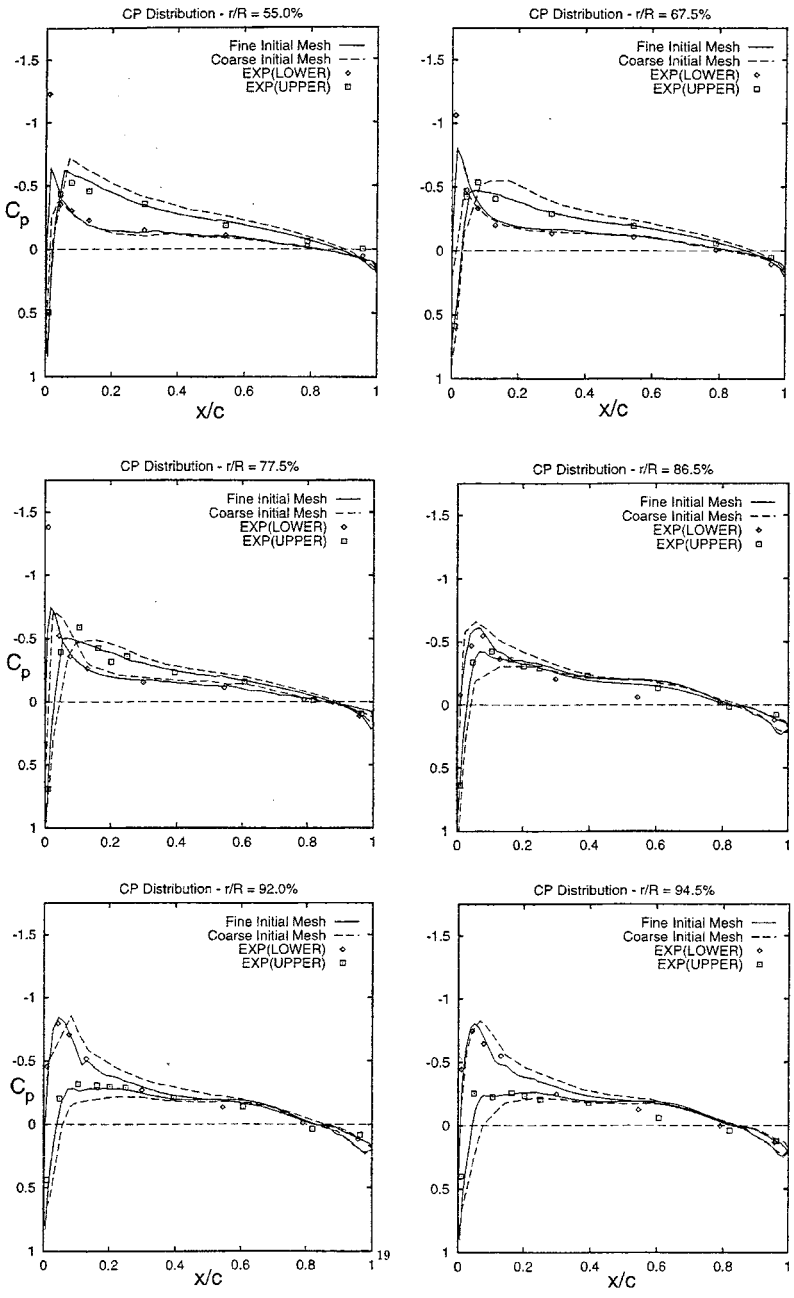

Fig. 12. Computed coefficient of pressure distributions: $C_T/\sigma = 0.0$.

5.2.2. UH-60A at design thrust

The final and the most challenging case analyzed for the UH-60A blade is a design thrust case where the flow conditions are: tip Mach number $M_t = 0.628$, rotor speed $\Omega = 1425$ rpm, thrust coefficient $C_T/\sigma = 0.085$, collective setting $\theta_{0.75} = 10.47°$, coning $\beta_0 = 2.31°$ and $FM = 0.73$. This particular case offers a stronger tip vortex structure than the zero thrust case.

Similar to the zero thrust case, an initial mesh with 1,200,000 tetrahedral elements is generated. Figs. 16 and 17 show the outer boundary and blade tip close-up views of this mesh, respectively. Notice

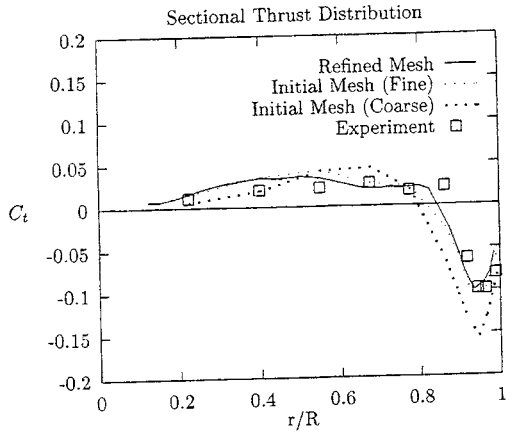

Fig. 13. Sectional thrust distribution at $C_T/\sigma = 0.0$.

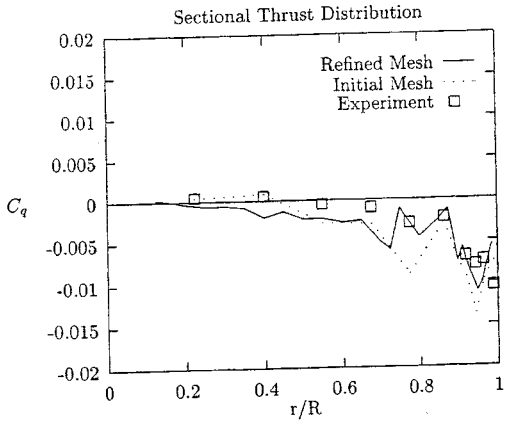

Fig. 14. Sectional torque distribution at $C_T/\sigma = 0.0$.

that the mesh resolution is finer at the outer boundaries in comparison to the zero thrust case (see Fig. 11). A finer mesh resolution at the outer boundaries is needed to account for the non-zero induced velocity field (due to potential sink) applied as the boundary condition. In the zero thrust case, the induced velocities at the outer boundaries were zero, thus, a fine mesh was not necessary. Also note that we again generated a finer mesh enclosing the rotor disk to capture the initial formation of the blade wake.

Comparisons of computed pressure distributions with experimental data [19] are given in Fig. 18. The refined mesh results are obtained using the two-level adaptive procedure starting from the initial mesh. During the adaptive solution of this case, the mesh is refined once using the interpolation error estimate and four times using the vortex core detection technique to resolve the global features of the flow-field and adaptively capture the tip vortex. The final refined mesh contains 2,300,000 tetrahedral elements. As seen in Fig. 18, the computed pressure distribution with the refined mesh has an overall better quality in capturing both the leading edge suction peaks and the mid-chord pressure plateau. The pressure distributions at radial stations 94.5%, 96.5% and 99% show noticeable improvement with mesh adaptation. This improvement is primarily due to a more accurate representation of the tip vortex release from the previous blade. The computed sectional thrust and torque coefficient distributions are presented in Figs. 19 and 20, respectively. The initial mesh, which could not resolve the tip vortex for the first 90° azimuth, resulted in a poor sectional thrust distribution particularly at the outboard portion of the blade. We emphasize here that

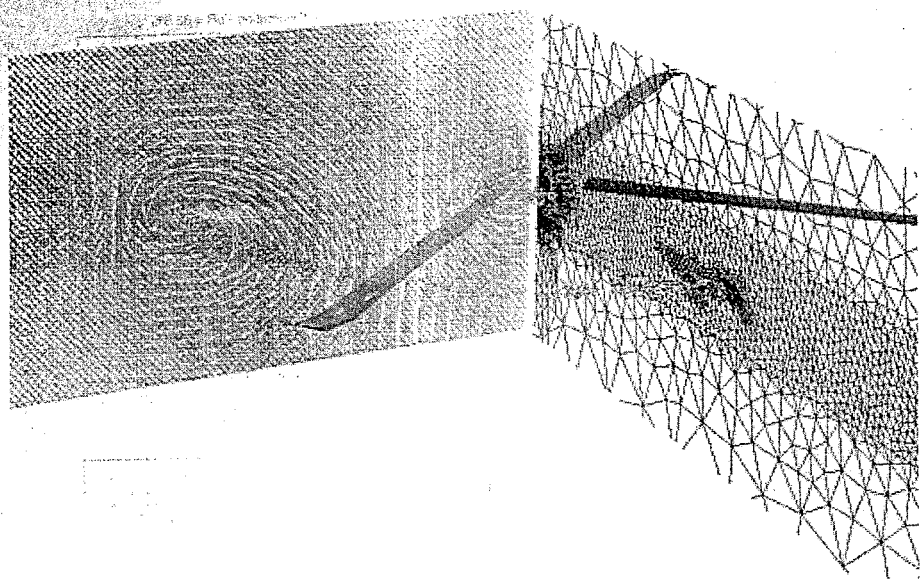

Fig. 15. Computed vortex flow structure for the UH-60A blade at zero thrust.

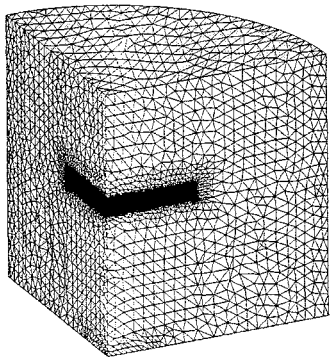

Fig. 16. Initial mesh for the UH-60A blade outer boundaries: $C_T/\sigma = 0.085$.

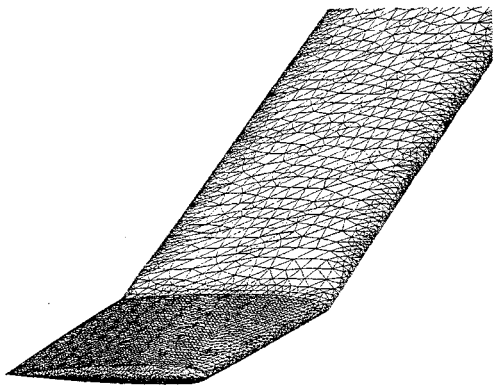

Fig. 17. Initial mesh for the UH-60A blade tip: $C_T/\sigma = 0.085$.

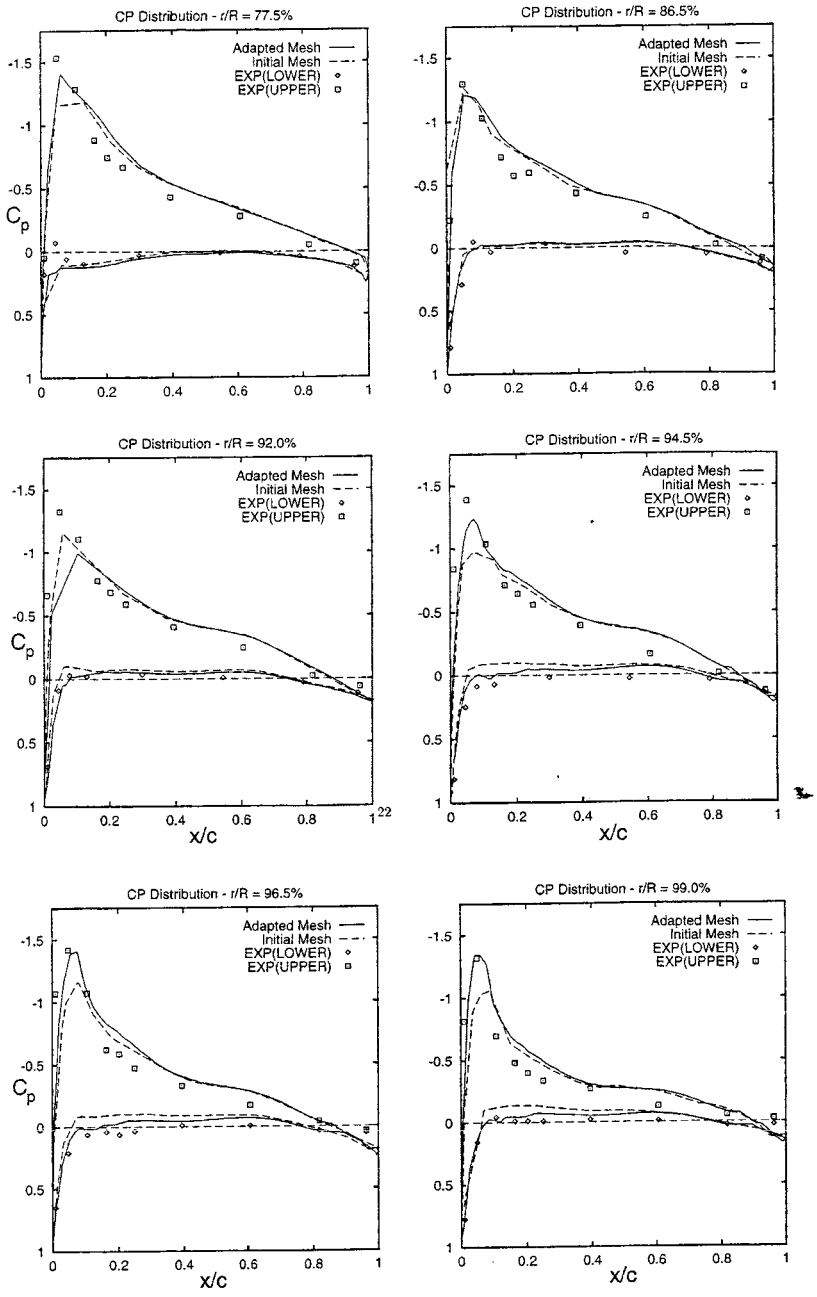

Fig. 18. Computed coefficient of pressure distributions: $C_T/\sigma = 0.085$.

all of our attempts to capture the correct distribution of sectional thrust before resolving the tip vortex properly failed. Progressive adaptive refinement steps, using both interpolation error estimate and the vortex core detection technique, resulted in a correct prediction of tip vortex structure. The sectional thrust distribution with the final adapted mesh shows a remarkable improvement with respect to initial mesh results. Therefore, it has been concluded that computing at least the first 90° azimuth travel of the tip vortex is essential for this high thrust UH-60A blade case. Fig. 20 compares the sectional torque coefficient for the refined and initial meshes against experimental data. Notice that the experimental torque distribution

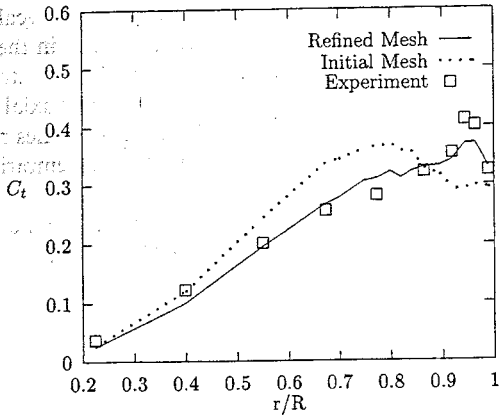

Fig. 19. Sectional thrust distribution at $C_T/\sigma = 0.085$.

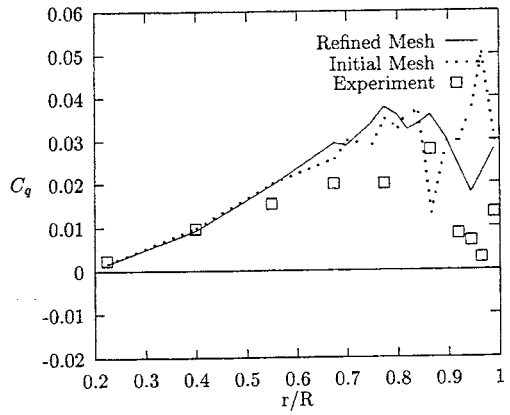

Fig. 20. Sectional torque distribution at $C_T/\sigma = 0.085$.

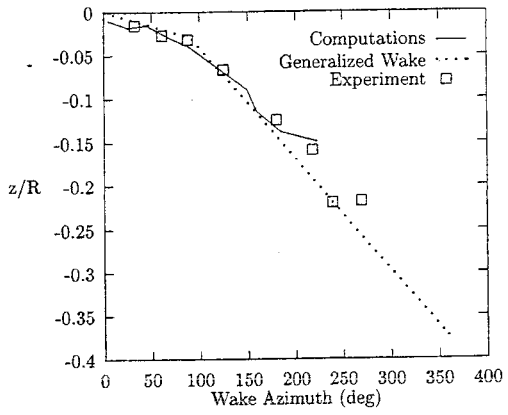

Fig. 21. Axial displacement of tip vortex: $C_T/\sigma = 0.085$.

contains both profile and skin friction torque, whereas numerical calculations reflect only the profile torque due to inviscid flow approximation. The integrated values of C_T/σ and figure of merit for the adapted mesh are 0.07946 and 0.72, respectively.

Further investigation of the wake geometry for this high thrust case reveals that the tip vortex, which is released from the previous blade, descends only about 3% blade radius in the axial direction by the time it passes under the next blade. Due to its close proximity to the blade, this strong vortex alters the velocity and pressure field of the blade tip significantly. The computed tip vortex axial and radial displacement plots are given in Figs. 21 and 22, respectively. In Figs. 21 and 22, the dashed lines represent the generalized wake formulation [17], where the tip vortex geometry is represented by an empirical formula based on experimental observations.

Fig. 23 shows the progress of mesh adaptation for the tip vortex. The images in Fig. 23 depict the UH-60A blade surface and a planar section of the mesh taken at 15° behind the blade. The planar sections in

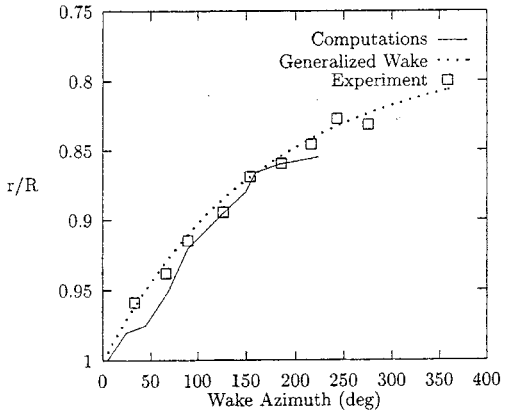

Fig. 22. Radial displacement of tip vortex: $C_T/\sigma = 0.085$.

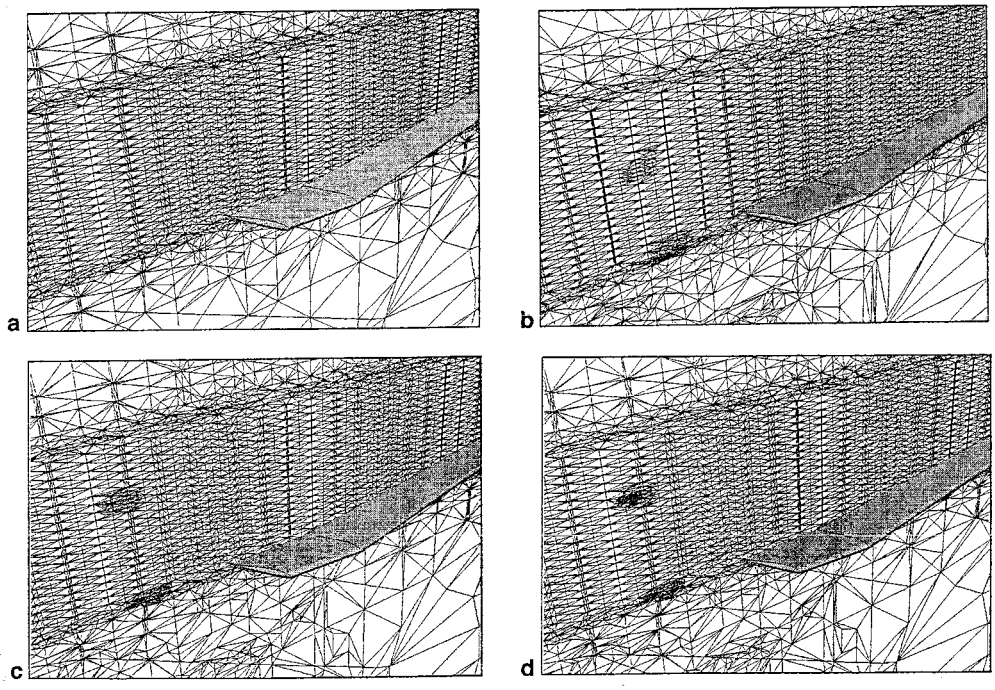

Fig. 23. Adaptive refinement levels for UH-60A at design thrust: (a) initial mesh: 1,200,000 tets; (b) first level: 1,750,000 tets; (c) second level: 2,100,000 tets; (d) third level: 2,300,000 tets.

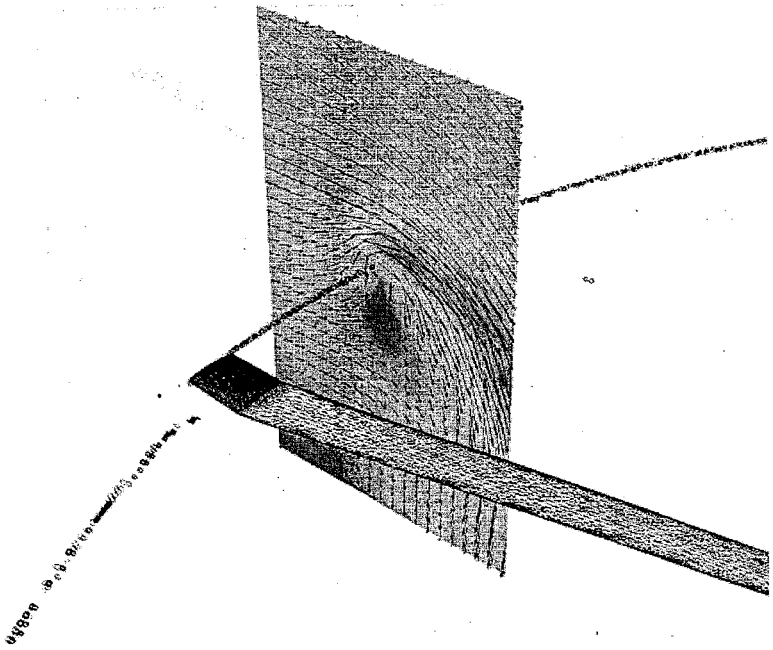

Fig. 24. Computed tip vortex flow structure for the UH-60A blade at $C_T/\sigma = 0.085$.

these images reflect the interpolated mesh connectivity. Notice that, as the refinement levels progress, the mesh resolution increases at the points on this 15° azimuthal plane where the tip vortex intersects the plane. We emphasize here that the adaptation procedure combined with the vortex core detection technique is very effective and efficient. A typical mesh adaptation level using vortex core detection technique increases the mesh size only by 5–7%. This is mainly due to that fact that the adaptation scheme refines the mesh only in the areas where there is an indication of a vortex flow. The vortex core detection technique is also used as a tool to render the tip vortex geometry. The adapted vortex flow near the tip of the blade is shown in Fig. 24. Finally, Fig. 25 shows the progressive adaptation of the tip vortex with increasing refinement levels.

6. Conclusions

An adaptive refinement procedure has been developed for computing vortical flows encountered in rotor aerodynamics. An error indicator based on interpolation error estimate is formulated and coded into the parallel adaptive finite element framework [5]. It has been shown that the error indicator based on interpolation error estimate is effective in resolving the global features of the flow-field. Along with the first error indicator it has been found that a second error indicator aids in the efficient resolution of small scale features of the flow such as vortex tubes. For this purpose a topology based vortex core detection technique has been used to capture vortex tubes for the rotor blade in hover conditions. It has been shown that the combination of the two error indicators used in refining the flow shows promise for computing rotor-blade flows effectively and efficiently with an acceptable level of user interference during the adaptation procedure.

For the Caradonna–Tung blade, the tip vortex is resolved 180° within four levels of adaptive refinement. Overall quality of the computed surface pressure distributions with the adaptation is very good. Especially, using the error indicator formulated in this paper, the leading edge suction peaks are captured successfully.

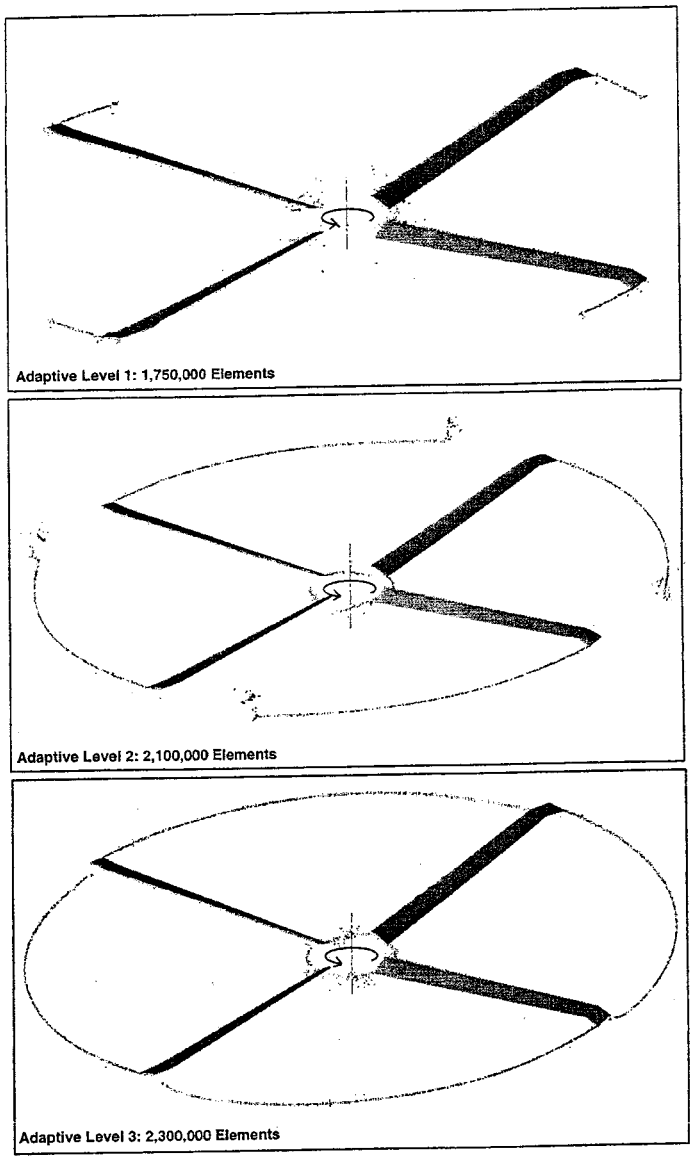

Fig. 25. Adaptive progress of computed tip vortex flow structures for the UH-60A blade at $C_T/\sigma = 0.085$.

For the UH-60A rotor blade, a zero thrust and a high thrust hover cases are studied. The effect of tip vortex for the zero thrust UH-60A computations is found to be minimal. However for this case, another bound vortex in strong interaction with the blade is identified at 75–80% radius and it is believed that this vortex tube exists due to an abrupt differential change in thrust loading. The inboard shed wake, in this case, should have a discontinuity around 80% radial span. Existence of this inboard vortex tube clearly demonstrated a deviation from empirical wake geometry studies [17] where the inboard shed wake was assumed continuous between the root and the tip of the blade. Finally, the sectional thrust and torque distributions computed by the adapted mesh are in good agreement with experimental data.

For the high-thrust case, it is found that the tip vortex interaction is strong and has to be computed as accurately as possible to get good agreement between computed and measured performance characteristics of the UH-60A blade. Progressive adaptive refinement steps, using both interpolation error estimate and the vortex core detection technique, resulted in a correct prediction of tip vortex structure. The sectional

thrust distribution with the final adapted mesh showed a remarkable improvement with respect to initial mesh where the tip vortex could not be resolved.

Acknowledgements

The authors would like to thank for the computer support provided by RPI Numerical Intensive Computing (NIC) services, Maui High Performance Computing Center (MHPCC) and NASA Ames Parallel Computing Center. The efforts of Dr. Carlo Bottasso, in coding the initial versions of the finite element program used here, is acknowledged. Furthermore, first author would like to thank to Dr. Hugues Decougny, Dr. Can Ozturan, Mark Beall, Jim Teresco and Louis Ziantz for providing their technical knowledge and skills into the progress of the work being presented in this paper. Finally, the authors want to thank the Army Research Office for its partial funding support under grant number DAAH04-940G-0073.

References

[1] R.H. Abraham, C.D. Shaw, Dynamics – The Geometry of Behavior, Part 2: Chaotic behavior, vol. 2, Aerial Press, 1985, pp. 27–30.
[2] M. Aiffa, Adaptive *hp*-refinement methods for singularly-perturbed elliptic and parabolic systems, Ph.D. Dissertation, Department of Mathematics Science, Rensselaer Polytechnic Institute, Troy, 1997.
[3] D.C. Arney, J.E. Flaherty, An adaptive method with mesh moving and local mesh refinement for time-dependent partial differential equations, Department of Computer Science, Rensselaer Polytechnic Institute, 1986.
[4] I. Babuska, M. Suri, The optimal convergence Rate of the p-version of the finite element method, SIAM J. Numer. Anal. 24 (1987) 750–776.
[5] C.L. Bottasso, M.S. Shephard, A parallel adaptive finite element solver for rotary wing aerodynamics, AIAA J. 35 (6) (1997) 1–8.
[6] F.X. Caradonna, C. Tung, Experimental and analytical studies of a model helicopter rotor in hover, NASA TM 81232, 1980.
[7] F. Chalot, T.J.R. Hugues, F. Shakib, Symmetrization of conservation laws with entropy for high-temperature hypersonic flows, Comput. Systems Engrg. 1 (1990) 495–521.
[8] M.S. Chong, A.E. Perry, B.J. Cantwell, A general classification of three-dimensional flow fields, Phys. Fluids A 2 (5) (1990) 765.
[9] K.D. Devine, J.E. Flaherty, R.M. Loy, S.R. Wheat, Parallel partitioning strategies for the adaptive solution of conservation laws, in: I. Babuska, J.E. Flaherty (Eds.), Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations, Springer, Berlin, vol. 75, 1995, pp. 215–242.
[10] S. Dey, M.S. Shephard, J.E. Flaherty, Geometry-based issues associated with p-version finite element computationsl, Comput. Meth. Appl. Mech. Engrg. 150 (1997) 39–50.
[11] J.E. Flaherty, R.M. Loy, P.C. Scully, M.S. Shephard, B.K. Szymanski, J.D. Teresco, L.H. Ziantz, Load Balancing and Communication Optimization for Parallel Adaptive Finite Element Methods, SCCC '97, IEEE, New York, 1997, pp. 246–255.
[12] J.E. Flaherty, R.M. Loy, C. Ozturan, M.S. Shephard, B.K. Szymanski, J.D. Teresco, L.H. Ziantz, Parallel structures and a dynamic load balancing for adaptive finite element computations, Appl. Numer. Math. 26 (1/2) (1998) 241–263.
[13] M. Georges, M.S. Shephard, Automatic three-dimensional mesh generation by the finite octree technique, Int. J. Num. Meth. Eng. 32 (4) (1991) 709–749.
[14] G. Goodsell, Pointwise super-convergence of gradient for the linear tetrahedral element, Num. Meth. Partial Differential Equations 10 (1994) 651–666.
[15] T.J.R. Hugues, L.P. Franca, G.M. Hulbert, A new finite element formulation for fluid dynamics: VIII. The Galerkin/least-squares method for advective-diffusive equations, J. Appl. Mech. 73 (1989) 173–189.
[16] D. Kenwright, R. Haimes, Vortex identification – applications in aerodynamics, in: IEEE/ACM Proc. Visualization '97, Phoenix AZ, ACM Press, New York, October 1997.
[17] A.J. Landgrebe, The Wake Geometry of a Hovering helicopter rotor and its influence on rotor performance, in: 28th Annual National Forum of the AHS, May 1972, pp. 3–15.
[18] R. Lohner, Mesh adaptation in fluid mechanics, Engrg. Fracture Mech. 50 (1995) 819–847.
[19] P.F. Lorber, R.C. Stauter, M.J. Pollack, A.J. Landgrebe, A Comprehensive Hover Test of Airloads and Airflow of an Extensively Instrumented Model Helicopter Rotor, vols. I–V, USAAVSCOM TR 91-D-16E, 1991.
[20] J.T. Oden, J.N. Reddy, An Introduction to the Mathematical Theory of Finite Elements, Wiley, New York, 1976.
[21] J.T. Oden, T. Strouboulis, P. Devloo, Adaptive finite element methods for the analysis of inviscid compressible flow: Part I. fast refinement/unrefinement and moving mesh methods for unstructured meshes, Comput. Meth. Appl. Mech. Engrg. 59 (1986) 327–362.
[22] R.C. Strawn, T.J. Barth, A finite-volume Euler solver for computing rotary-wing aerodynamics on unstructured meshes, in: AHS 48th Annual Forum Proceedings, June 1992, pp. 419–428.