JOURNAL OF AIRCRAFT
Vol. 43, No. 5, September–October 2006

Error Estimation and Grid Adaptation Using Euler Adjoint Method

Hyoung-Jin Kim,[*] Yoshimichi Takano,[†] and Kazuhiro Nakahashi[‡]
Tohoku University, Sendai, 980-8579 Japan

DOI: 10.2514/1.19603

An adjoint-based error estimation and grid adaptation study is conducted for three-dimensional inviscid flows with unstructured meshes. The error in an integral output functional of interest is estimated by a dot product of the residual error and adjoint variable vector. To suppress excessive mesh refinement in unnecessary regions due to high-frequency noise contained in the residual vector, the flow residual is smoothed using a volume-weighted averaging process. Regions to be adapted are selected based on the error contribution of a local node to the global error. The adaptive regions are refined by the bisection refinement algorithm. The present procedure is applied to three-dimensional transonic inviscid flows around ONERA M6 wing and ONERA M5 airplane models. The same level of prediction accuracy for drag is achieved with much less mesh points than uniformly refined fine meshes. It is found that the residual smoothing strategy remarkably improves the accuracy of error estimation, and there exists an optimum number of smoothing for accurate error estimation.

I. Introduction

AMONG computational fluid dynamics (CFD) approaches for complex aerodynamic configurations, the unstructured grid approach has several advantages over structured grid approaches. It can treat complex geometries with more efficiency and less effort. It also has a greater flexibility in the adaptive mesh refinement so that the total number of grid points can be minimized for the required level of analysis accuracy. However, the complicated data structure of the unstructured grid approach requires more computational time and memory than other approaches do. Also, the unstructured approach suffers from more numerical diffusion than structured approaches. Thus employment of an adaptive grid refinement strategy is desirable for efficient CFD analysis with unstructured meshes.

The use of the grid adaptation allows reducing truncation errors with a limited number of grid points. One of the concerns for adaptive grid refinement is the selection of adequate refinement parameters for minimizing analysis errors. Conventional error indicators for adaptive mesh refinement are usually gradient or curvature information of flow variables such as pressure, temperature, or velocity [1–4]. However, functional outputs of engineering interests for flow analysis are usually integral outputs such as lift, drag, or moment rather than local flow errors. In general, the local error related with gradient or curvature of flow variables is not directly related to the global integral functional outputs.

The global error in the functional outputs can be estimated by introducing the adjoint variable, which is the sensitivity of the functional output of interest with respect to the local flow residual. This has been mainly applied to the finite element framework [5,6]. Recently, the error estimation method on global functional outputs was first applied to finite volume CFD by Giles and Pierce [7]. The local amount of the inner product components was used as an indicator for mesh refinement. Müller and Giles [8] applied the method to two-dimensional inviscid flows and suggested smoothing of the residual to remove high-frequency noise in the flow residual vector. Venditti and Darmofal [9–11] applied the method to quasi-one-dimensional inviscid, two-dimensional inviscid, and viscous problems with modified adaptation parameters presenting uncertainty in the error rather than the error itself. It was also shown that the adjoint-based mesh adaptation strategy produces much more accurate functional outputs than conventional mesh adaptations using featured-based local error indicators. The error estimation and adaptive mesh refinement method for finite volume CFD has been receiving much attention. However, most of the previous works are limited to one- and two-dimensional problems. Recently, Park [12,13] first applied the method to three-dimensional inviscid and viscous flows.

In this study, an adjoint-based error estimation and grid adaptation study is conducted for three-dimensional inviscid flows with unstructured meshes. At first, a flowfield is analyzed on a coarse mesh using a flow solver. A discrete adjoint analysis is then performed using the results of the flow analysis for a functional output of interest. With the flow and adjoint variables on the coarse mesh in hand, fine mesh variables are interpolated from the coarse mesh variables, where the fine mesh is obtained by a few bisection refinements of the coarse mesh. A flux residual vector is then calculated once by the flow solver with the interpolated flow variables on the fine mesh. To suppress excessive mesh refinement in unnecessary regions due to high-frequency noise contained in the residual vector, the flow residual is smoothed using a volume-weighted averaging process. By an inner product of the local residual vector and the adjoint variable vector, the error in the output function is estimated and regions to be adapted are determined. Finally, the adaptive regions are refined by the bisection refinement algorithm. The above procedure for adaptive mesh refinement is applied to transonic inviscid flows around ONERA M6 wing and ONERA M5 airplane models.

In Sec. II, detailed methodology of the present error estimation and mesh adaptation method is described. A brief description on the flow solvers and adjoint codes is given in Sec. III, which is followed by examples for three-dimensional configurations in transonic flow conditions in Sec. IV. Finally conclusions are made in Sec. V.

II. Methodology

A. Error Estimation

The formulations for error estimation are basically taken from [9]. A discrete residual vector for a fine mesh Ω_h can be expressed with

Presented as Paper 5336 at the 17th AIAA Computational Fluid Dynamics Conference, Toronto, Canada, 6–9 June 2005; received 22 August 2005; revision received 26 October 2005; accepted for publication 23 November 2005.

[*] Research Associate, Department of Aerospace Engineering, Member AIAA; hjkim@ad.mech.tohoku.ac.jp.

[†] Graduate Student, Department of Aerospace Engineering.

[‡] Professor, Department of Aerospace Engineering, Associate Fellow AIAA.

flux Jacobian and flow solution at a coarse mesh Ω_H using the Taylor series expansion:

$$R_h(Q_h) \cong R_h(Q_h^H) + \frac{\partial R_h}{\partial Q^h}\bigg|_{Q_h^H}(Q_h - Q_h^H) \tag{1}$$

where Q_h is a flow solution at a fine mesh Ω_h, and Q_h^H is a solution of a coarse mesh Ω_H interpolated into the fine mesh Ω_h. Since the residual vector should be zero at the fine mesh, i.e., $R_h(Q_h) = 0$, the following expression for the error in flow variables can be obtained:

$$\varepsilon = Q_h - Q_h^H \cong -\left[\frac{\partial R_h}{\partial Q_h}\bigg|_{Q_h^H}\right]^{-1} R_h(Q_h^H) \tag{2}$$

Using the above relation, the error in an integral output such as aerodynamic coefficients can be calculated as follows:

$$\begin{aligned} F_h(Q_h) - F_h(Q_h^H) &\cong \frac{\partial F_h}{\partial Q_h}\bigg|_{Q_h^H}(Q_h - Q_h^H) \\ &= -\frac{\partial F_h}{\partial Q_h}\bigg|_{Q_h^H}\left[\frac{\partial R_h}{\partial Q_h}\bigg|_{Q_h^H}\right]^{-1} R_h(Q_h^H) \end{aligned} \tag{3}$$

Since the adjoint variable is the solution of the following discrete adjoint equation:

$$\frac{\partial R^T}{\partial Q}\Lambda + \frac{\partial F^T}{\partial Q} = 0 \tag{4}$$

Eq. (3) can be expressed with the adjoint variable vector Λ as follows:

$$\begin{aligned} F_h(Q_h) - F_h(Q_h^H) &\cong (\Lambda_h|_{Q_h^H})^T R_h(Q_h^H) \\ &\cong (\Lambda_h^H)^T R_h(Q_h^H) = (J_h^H \Lambda_H)^T R_h(I_h^H Q_H) \end{aligned} \tag{5}$$

where I_h^H and J_h^H are prolongation operators from coarse to fine meshes for flow and adjoint variables, respectively.

Equation (5) implies that error in the functional output can be approximated by the inner product of adjoint and residual vectors or the weighted average of the residual vector components with adjoint variables as a weighting function.

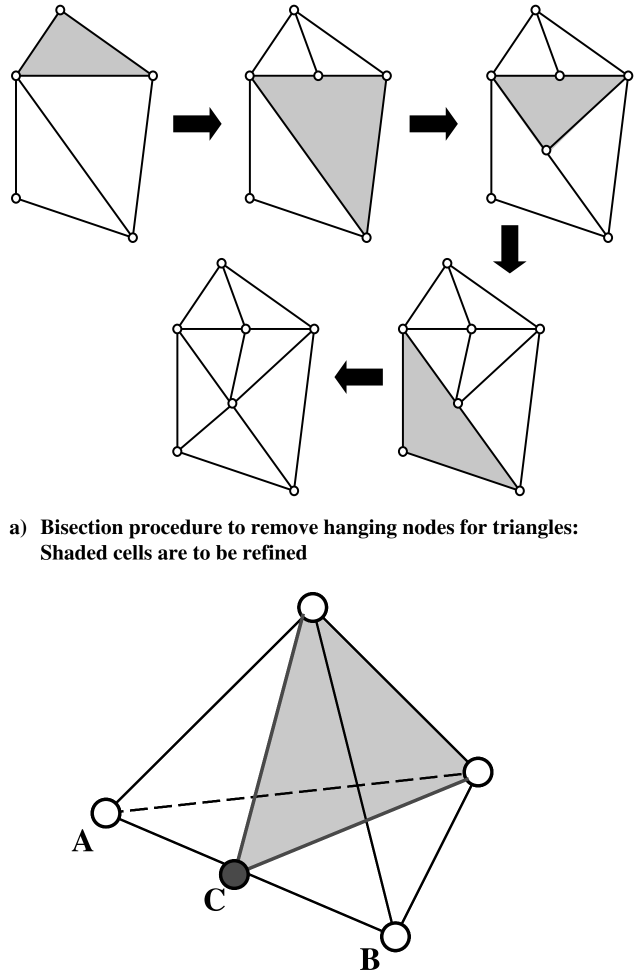
For the solution of the discrete adjoint equation (5), we use a discrete adjoint code developed in [14].

B. Mesh Refinement

In previous studies on adjoint-based error estimation and mesh adaptation, fine meshes are usually generated by regular refinement which inserts new nodes at the midpoint of existing edges creating four new triangles from an existing triangle for two-dimensional cases, and eight new tetrahedral from an existing tetrahedron for three-dimensional cases [10,12].

The refinement algorithm employed in this study is Rivara's longest-edge bisection algorithm [15,16]. A tetrahedron is refined into two "sons" by a bisecting plane passing through the midpoint of its longest edge and through the nodes opposite to this edge. The bisection algorithm divides the maximum angle into two leading to minimization of the maximum angle, which allows improving the quality of refined cell shapes over their "parent" cell shape. The bisection refinement process is illustrated in Fig. 1 for two-dimensional triangle cells and three-dimensional tetrahedral cells.

A drawback of the longest-edge bisection method is that tetrahedra resulting from the method are not guaranteed to have a bounded solid angle. However, Rivara and Levin showed that the minimum angle in the tetrahedron does not drop below a fixed point [16]. Some variants of the longest-edge bisection algorithm have also been suggested which are proven to be bounded in their shape measures [17]. Details of the three-dimensional bisection algorithm can be found in [3]



a) Bisection procedure to remove hanging nodes for triangles: Shaded cells are to be refined

b) Tetrahedron

Fig. 1 Longest-edge bisection refinement.

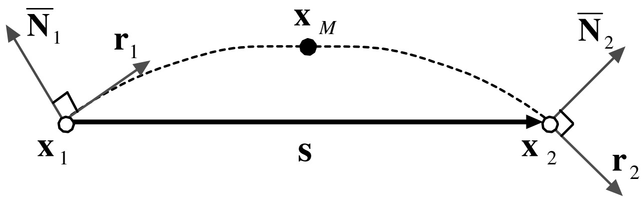
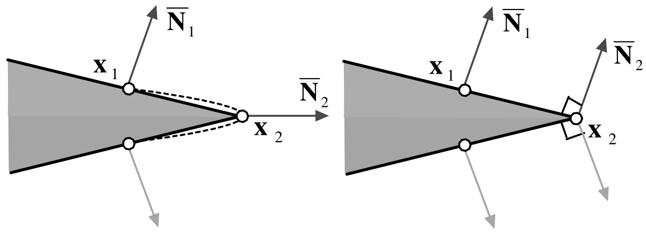


Fig. 2 Surface recovery: projection of newly added midpoint X_M on the surface.

As shown in Fig. 1a bisection refinement of one cell causes succeeding refinement of surrounding cells to remove all the hanging nodes. Thus, single refinement by the bisection algorithm creates a fine mesh with 3–4 times more mesh points than the initial coarse mesh. Since this fine mesh may not be "fine" enough for the error estimation and adaptation purposes, multiple refinements with level $N = 1$, 2, or 3 are tested as presented in the following sections.

C. Surface Recovery

When a newly added node point is located on the body surface, the new node needs to be projected onto the smooth surface geometry. This can be conducted either by a direct CAD interface [12] or a surface recovery method [18]. In the present study, we adopt the latter approach for a more portable and simple flow analysis tool. In this case, because new nodes are always on the midpoint of the edge, quadratic fitting for the edge is only needed instead of triangular surface cell reconstruction. A midpoint location $\mathbf{X}_M$ indicated in Fig. 2 is estimated by a Hermitian polynomial as



a) Single normal vector **b) Multiple normal vectors**

Fig. 3 Definition of normal vectors on ridges such as trailing edge for surface recovery: the single normal vector is not adequate for trailing edges.

$$X_M = 0.5(X_1 + X_2) + 0.125(\mathbf{r}_1 - \mathbf{r}_2) \tag{6}$$

where

$$\mathbf{r}_i = |\mathbf{s}| \cdot \frac{\bar{\mathbf{N}}_i \times \left(\mathbf{s} \times \bar{\mathbf{N}}_i\right)}{|\bar{\mathbf{N}}_i \times \left(\mathbf{s} \times \bar{\mathbf{N}}_i\right)|}, \qquad (i = 1, 2), \qquad \mathbf{s} = X_1 - X_2 \tag{7}$$

$\boldsymbol{X_1}$ and $\boldsymbol{X_2}$ are location vectors of both ends of the edge; $\bar{\mathbf{N}}_1$ and $\bar{\mathbf{N}}_2$ are unit normal vectors at each node.

If $\boldsymbol{X_1}$ or $\boldsymbol{X_2}$ is on a ridge such as the wing trailing edge, the midpoint location may deviate from the initial geometry as the dotted line in Fig. 3a. To circumvent this problem, multiple normal vectors are defined at ridge nodes when the angle between normal vectors of two adjacent surface cells is larger than 60 deg as shown in Fig. 3b [19].

D. Prolongation of Flow and Adjoint Variables

Flow and adjoint variables on the fine mesh Ω_h are interpolated from the variables on the coarse mesh Ω_H using prolongation operators I_h^H and J_h^H. For common nodes in both the coarse and the fine meshes, the flow and the adjoint variables are passed from coarse mesh to fine mesh nodes. For newly added nodes in the fine mesh like node C in Fig. 1b, interpolation from existing nodes is required. In this study, we adopt edge-based linear and quadratic interpolation operators [12], and the same operator is used for both flow and adjoint variables, i.e., $I_h^H = J_h^H$.

For linear interpolation, variables U_C at midpoint C in Fig. 1b is calculated by averaging variables at both ends of the edge AB as follows:

$$U_C = 0.5(U_A + U_B) \tag{8}$$

For quadratic interpolation, U_C is estimated by a Hermitian polynomial as

$$U_C = 0.5(U_A + U_B) + 0.125(\delta U_A - \delta U_B) \tag{9}$$

where $\delta U = \nabla U \cdot (X_B - X_A)$. ∇U is the gradient vector of U, and X is the location vector.

The gradient vector is calculated by a least squares method so that it is consistent with the reconstruction method of the flow solver. Actually, the gradient vector of the flow variables calculated in the flow solver with a limiter function is imported and used in the interpolation routine, and the gradient vector for adjoint variables is newly calculated without any limiter. ∇U at newly added nodes like node C is calculated by averaging gradients at both ends of the edge (nodes A and B) for cases where other new nodes are inserted on edges AC or BC to remove hanging nodes during the refinement process.

E. Residual Smoothing

With the variables $I_h^H Q_H$ and $J_h^H \Lambda_H$ projected from the coarse mesh Ω_H to the fine mesh Ω_h, the estimated error in the integral output can be calculated by Eq. (5). The residual vector $R_h(I_h^H Q_H)$ on Ω_H is calculated by single flux accumulation without any iterative process. However, the residual vector R_h calculated with interpolated flow variables contains high-frequency noises due to the inconsistency between interpolated and physical variables [8,9]. The high-frequency noises may degrade the accuracy of the error estimation and cause the waste of mesh points in regions where refinement is unnecessary.

To smooth out the noise components the smoothing process suggested in [8] is employed here. The residual at node i is distributed to elements E_i which contain node i, and the residual at tetrahedral element j is redistributed to its four nodes N_j. Also considered is the ratio between the element volume V_j and node control volume V_i so that the total sum of the residual is conserved during the smoothing process. The equations for the smoothing process can be written as follows:

$$R_j^{\text{elem}} = \sum_{i \epsilon N_j} \frac{1}{n} \frac{V_j}{V_i} R_i^{\text{node}} \tag{10a}$$

$$R_i^{\text{node}} = \sum_{j \epsilon E_i} \frac{1}{n} R_j^{\text{elem}} \tag{10b}$$

where n is the number of nodes of an element: three for triangle and four for tetrahedron, etc.

The above smoothing process can be applied repeatedly until high-frequency noise produced by the prolongation of flow variables is removed.

F. Adaptation Parameter

Giles and Pierce [7] suggested the use of the estimated error at each node as an adaptation parameter of the mesh adaptation for the accurate computation of integral outputs. Venditti and Darmofal [9] suggested the use of uncertainty in the estimated error quantity as an adaptation parameter instead of the error itself. While this approach is slightly better and more robust than that of Giles and Pierce, it requires the calculation of the adjoint residual for the fine mesh Ω_h. Although the residual calculation is made only once, it still largely increases the memory requirement of the total process, because the memory requirement of the adjoint solver is a few times larger than that of the flow solver [14]. Since we prefer a simple and portable solution package of the flow solver in this study, the local error shown in Eq. (5) is selected as the error indicator of the mesh adaptation suggested in [7] as follows:

$$\varepsilon_i = \left(J_h^H \Lambda_H\right)_i^T R_h \left(I_h^H Q_H\right)_i \tag{11}$$

The refinement criterion is determined such that when the absolute value of the local error falls in the following range, the node is flagged for refinement:

$$|\varepsilon_i| \geq (a\sigma + b) \frac{\varepsilon_t}{\varepsilon_g} \tag{12}$$

where $\varepsilon_g = \sum \varepsilon_i$ is the global error, σ is the standard deviation of local errors ε_i, and a and b are parameters adjusting the number of refined mesh points. In this study, $a = 1$ and $b = 0$ is employed.

After the flagging process for nodes to be refined on the fine mesh, the information of flagged nodes is transferred to the coarse mesh. In the fine mesh, if an *old* node, which also exists in the coarse mesh, is connected to a newly added node flagged for refinement, the old node is also flagged for refinement. Now in the coarse mesh, if a tetrahedron has at least one node flagged for refinement, the tetrahedron is marked for refinement. Then the bisection refinement algorithm is applied to the flagged tetrahedral in the coarse mesh.

Mesh unrefinement for cells with a very low level of error would also enhance the efficient use of mesh points. However, in this study, mesh unrefinement is not conducted in the test cases.

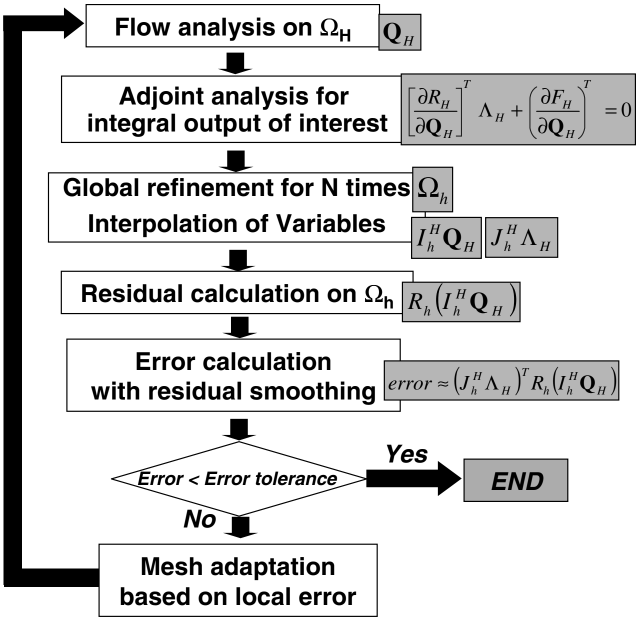

Flow analysis on Ω_H
$\mathbf{Q}_H$
Adjoint analysis for integral output of interest
$\left[\frac{\partial R_H}{\partial \mathbf{Q}_H}\right]^T \Lambda_H + \left(\frac{\partial F_H}{\partial \mathbf{Q}_H}\right)^T = 0$
Global refinement for N times
Ω_h
Interpolation of Variables
$I_h^H \mathbf{Q}_H$
$J_h^H \Lambda_H$
Residual calculation on Ω_h
$R_h\left(I_h^H \mathbf{Q}_H\right)$
Error calculation with residual smoothing
$error \approx \left(J_h^H \Lambda_H\right)^T R_h\left(I_h^H \mathbf{Q}_H\right)$
Error < Error tolerance
Yes
END
No
Mesh adaptation based on local error

Fig. 4 Flow chart for error estimation and mesh refinement.

G. Overall Procedure

Figure 4 shows the overall procedure of error estimation and mesh refinement employed in this study. For a coarse mesh Ω_H flowfield is analyzed with the flow solver. The adjoint equations are solved with the flow solution Q_H for integral output of interest F. Then the coarse mesh Ω_H is uniformly refined for N times to create a fine-enough reference mesh Ω_h on which the error estimation is conducted. During the uniform refinement process, flow and adjoint variables are also interpolated simultaneously by either the linear or the quadratic interpolation. With the interpolated flow variables in Ω_h, residual of flow equations is computed with a single flux quadrature. The error in the integral output of interest can now be calculated by a dot product of the residual vector and the adjoint vector. To remove high-frequency noises inherent in the residual, the residual smoothing is conducted repetitively until the estimated error reaches its optimum value.

III. Flow and Adjoint Analysis

A. Flow Analysis

A three-dimensional unstructured Euler solver is used as a flow solver. The governing Euler equations are solved by the cell-vertex finite volume scheme. Control volumes are nonoverlapping dual cells constructed around each node. Each edge connecting two nodes is associated with an area vector of the control surface, and gas dynamic fluxes are computed through the areas. To enhance the accuracy of the scheme, a least squares reconstruction of the primitive gas dynamic variables inside the control volume is used in conjunction with a flux limiter. The flux is computed using an approximate Riemann solver. For the time integration, the lower-upper symmetric Gauss–Seidel (LU-SGS) implicit method is adopted [20]. Details of the flow solver can be found in [21].

B. Adjoint Analysis

With the solution of the flow equations, the linear adjoint equation of (4) is solved by a discrete adjoint solver, which was developed in a previous study for efficient aerodynamic shape optimization [14]. A pseudotime term is added to Eq. (4), which is then solved by the same time integration scheme as the flow solver.

IV. Results

As examples of the error estimation and adaptive mesh refinement in three-dimensional flows, ONERA-M6 wing and ONERA-M5 airplane configurations are tested in transonic flow conditions. The meshes are refined for accurate calculation of drag by surface integration.

The surface meshes on the configurations are generated by a direct advancing front method coupled with the geometrical feature extraction on the STL (Stereolithography) data format [19]. The tetrahedral volume meshes are generated by a Delaunay-type generation method [22].

A. ONERA-M6 Wing

The initial coarse mesh around the ONERA-M6 wing configuration has 54,349 nodes and 292,452 tetrahedral elements. Flow conditions are M_∞ of 0.84 and an incidence angle of 3.06 deg.

First, the effect of residual smoothing is presented for the initial mesh. The inherent noise in the residual due to the interpolation of flow variables may cause the waste of nodes in regions where refinement is not necessary. Figure 5 shows the variation of the estimated error in drag for the iteration number of the smoothing process. The result is obtained from the initial mesh with linear interpolation and $N = 3$. The error between the drag for initial mesh and an extrapolated value from uniformly refined fine meshes is 67 counts, and the error estimation result with residual smoothing plotted in Fig. 5 estimates the error as about 59 counts with 13 smoothing iterations, whereas without smoothing the estimated error is only 35 counts.

In all the cases conducted in this study there exists an optimum number of smoothing which results in a most accurate error estimation. Similar trends are obtained for other interpolation methods and values of N, although not shown here. In the initial stage of the iterative smoothing, high-frequency noises are being removed resulting in more accurate results, and after an optimum number of smoothing, the main feature of the residual is being smeared out by the smoothing resulting in a gradual decrement of accuracy.

Figure 6 compares contours of the residual vector $R_h(I_h^H Q_H)$ before and after the smoothing operation. The noise is removed by the smoothing, without which the regions to be adapted would be largely affected by the noisy residual.

Figure 7 shows error-corrected drag coefficients on the initial coarse mesh for various options in the error estimation: interpolation methods, with or without residual smoothing, and the number of refinement N for the fine mesh creation. The error estimation is more accurate with residual smoothing, higher order interpolation, and larger N.

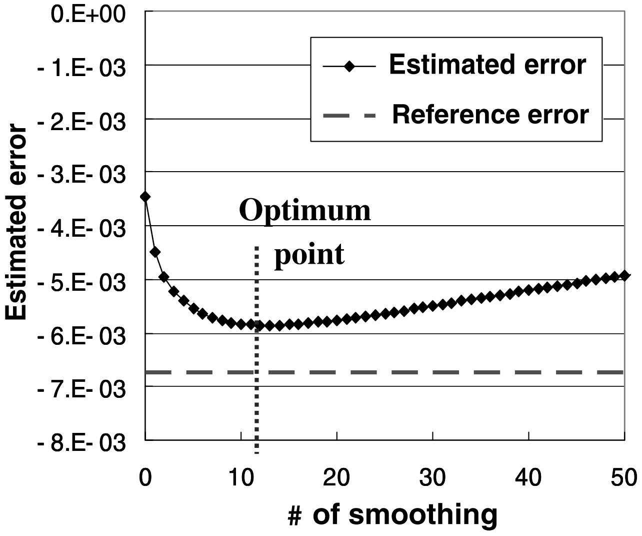
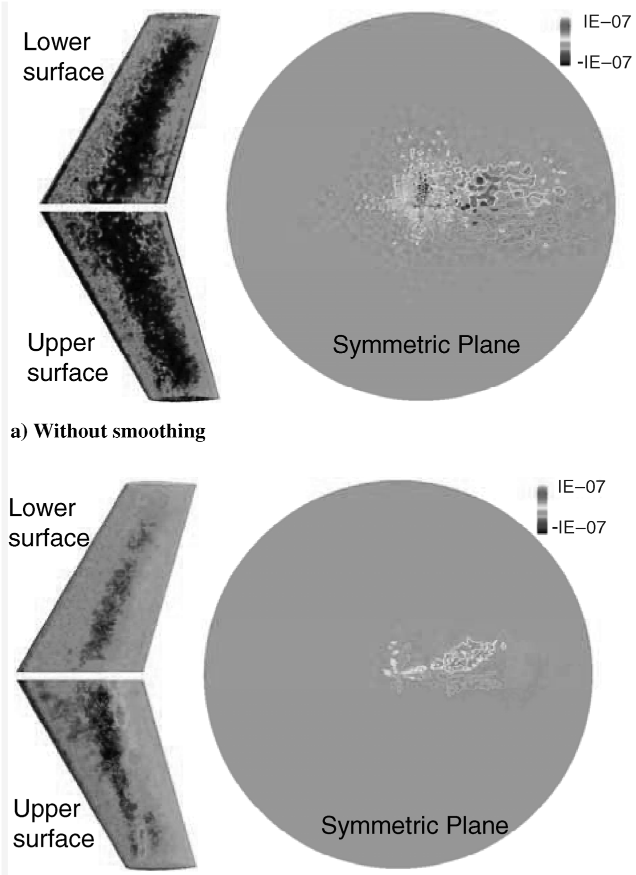


Fig. 5 The effect of the number of smoothing on the estimated error in C_D of ONERA M6 at initial mesh [linear prolongation, N(level of fine mesh) = 3]. The reference error is the error compared with the Richardson extrapolation. There exists an optimum number of residual smoothing which renders a most accurate estimation of the error.



a) Without smoothing

b) With smoothing (13 times)

Fig. 6 Contours of residual R_5 with and without smoothing for ONERA M6. Smoothing of the residual of the prolongated flow variable Q_h^H removes high-frequency noise.

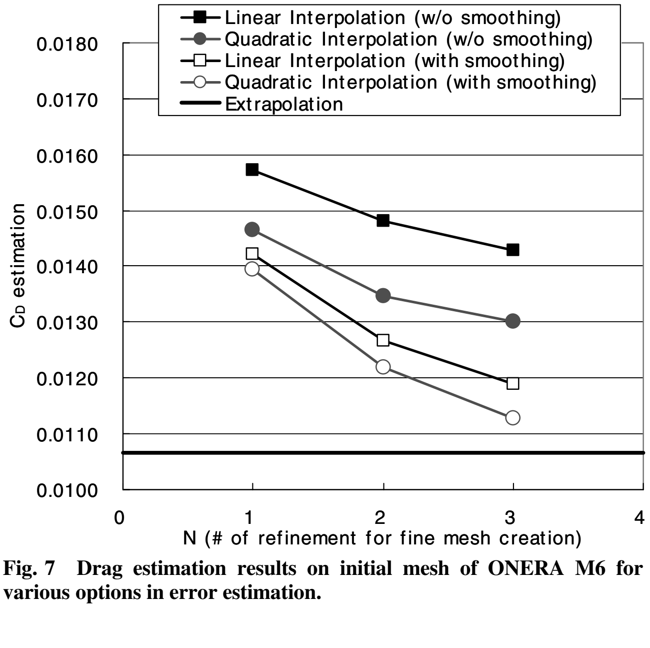


Fig. 7 Drag estimation results on initial mesh of ONERA M6 for various options in error estimation.

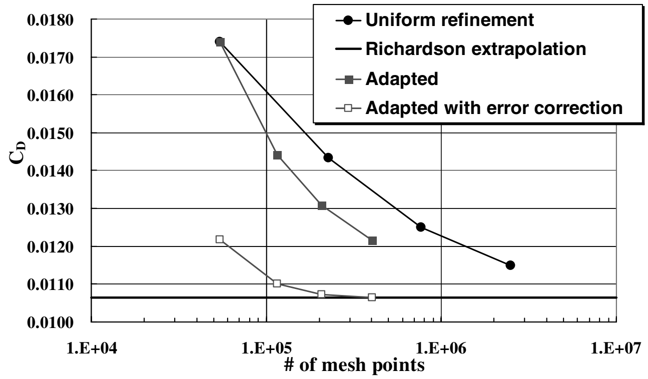
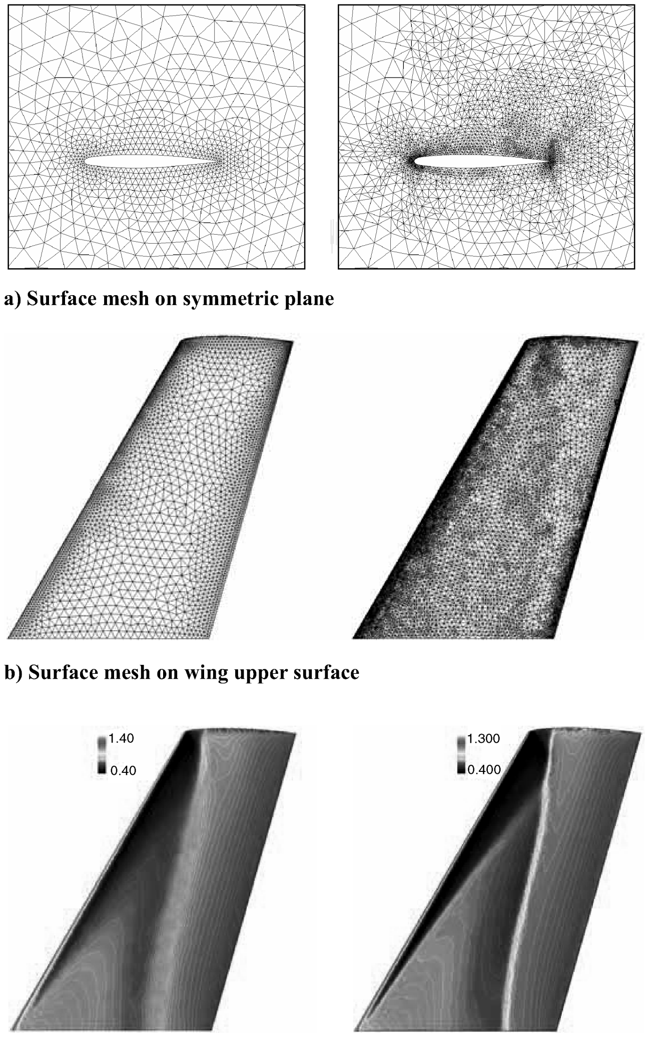


Fig. 8 Results of adaptive mesh refinement for an accurate C_D calculation of ONERA M6 (quadratic prolongation, $N = 2$, with residual smoothing).



a) Surface mesh on symmetric plane

b) Surface mesh on wing upper surface

c) Pressure contours on wing upper surface

Fig. 9 Comparison of initial (left) and adaptively refined (right) ONERA M6 meshes and flow solutions on the meshes.

Meanwhile, N is the refinement number that mostly affects the required memory. Table 1 shows the number of mesh points for $N = 1$–3. With $N = 3$, the number of mesh points is about 46 times that of the initial mesh. The memory requirements of the refinement codes can easily reach more than several gigabytes in three-dimensional cases. Thus N should be selected carefully considering available memory storage and estimation accuracy.

In [9], on the other hand, it was reported that the number of mesh points in the fine mesh did not affect the accuracy of error estimation for the quasi-one-dimensional case. This inconsistency between the present study and [9] might be caused because the fine meshes tested in [9] correspond to larger N's than tested in this study. As can be seen in Fig. 7, the amount of estimated drag reduction is being decreased as N increases. Thus for larger N's such as 5 and 6, there might be little difference between the estimated drags. In the present study, this could not have been tested because of insufficient

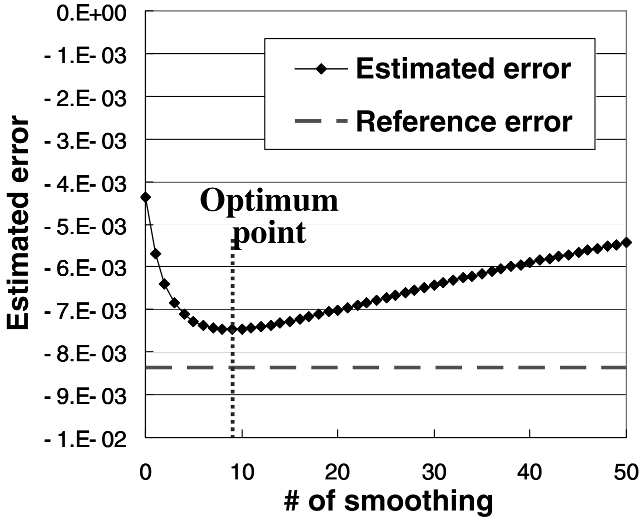


Fig. 10 The effect of the number of smoothing on the error estimation in C_D of ONERA M5 at initial mesh [quadratic prolongation, N(level of fine mesh) = 2]. The reference error is the error compared with the Richardson extrapolation. There exists an optimum number of residual smoothing which renders a most accurate estimation of the error.

Table 1 Number of mesh points by uniform refinement from initial ONERA M6 mesh

N	0 (initial)	1	2	3
No. of mesh points	54,349	227,336	764,980	2,492,882
Relative ratio to the initial mesh	1	4.2	14.1	45.9

available memory for the mesh refinement process.

The mesh adaptation results are presented in Fig. 8. The drag coefficient is plotted for the total number of mesh points. For purposes of comparison, drag coefficients for uniformly refined meshes are also computed and plotted together with the adaptively refined mesh results. The functional results by uniformly refined meshes are extrapolated to be used as a reference value. Considering the two finest meshes with indices 1 and 2 whose characteristic cell sizes are h_1 and h_2, respectively, extrapolated functional f can be written as

$$f = \frac{kf_1 - f_2}{k - 1}, \qquad k = \left(\frac{h_2}{h_1}\right)^2 \tag{13}$$

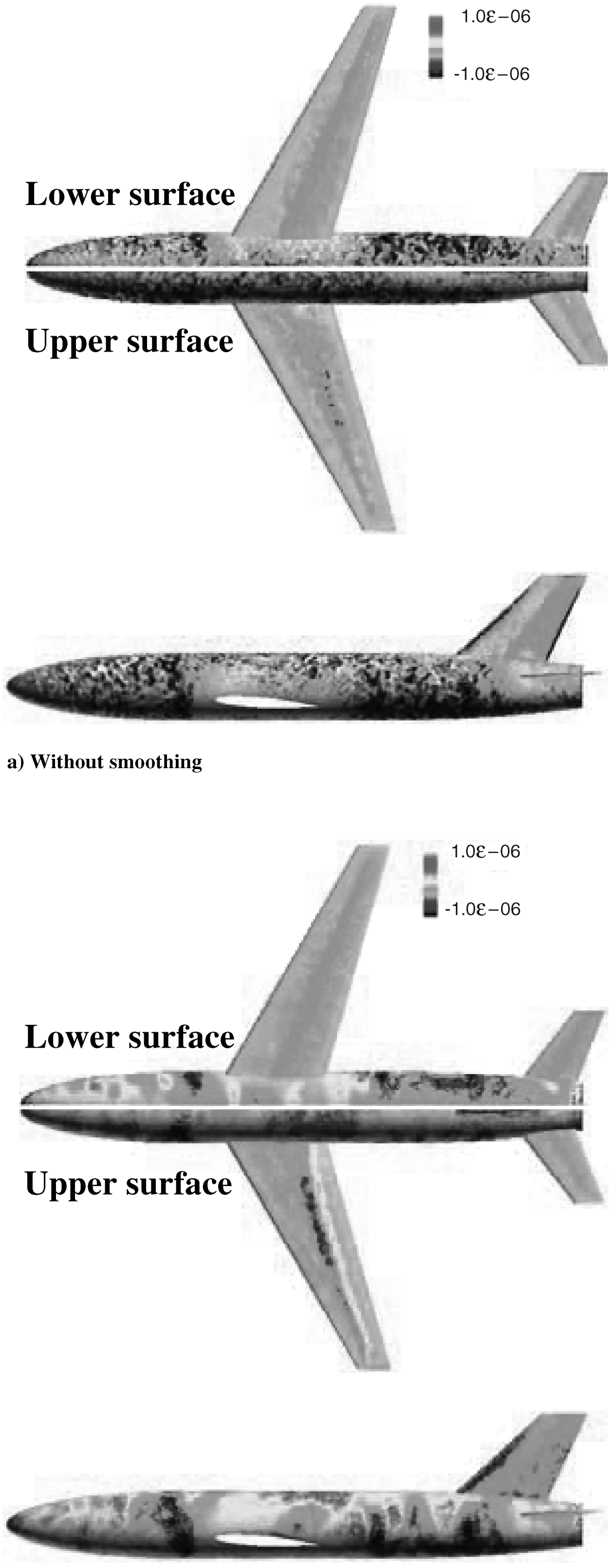
where h_1 and h_2 are proportional to the inverse of the cubic root of the number of cells for three-dimensional cases [9,12].

The adaptive refinement is conducted 3 times and the estimated error is 16 counts on the final mesh. The error-corrected drag is also plotted at the same number of mesh points with the adapted mesh. The final mesh has 401,437 nodes and 106 counts of corrected drag and 122 counts of uncorrected drag. The error-corrected drag almost exactly coincides with the extrapolated drag from fine meshes. Including error correction, the same accuracy of the functional output as the uniformly refined meshes is achieved with about one-and-half order less mesh points.

The final mesh of the adaptive refinement is compared with the initial mesh in Fig. 9. The leading and trailing edge regions are mainly refined, and the shock wave region is slightly refined by the adaptive refinement for accurate drag computation.

B. ONERA M5

The initial coarse mesh around the ONERA M5 configuration has 113,124 nodes and 606,110 tetrahedral elements. Flow conditions are M_∞ of 0.84 and an incidence angle of −1 deg.

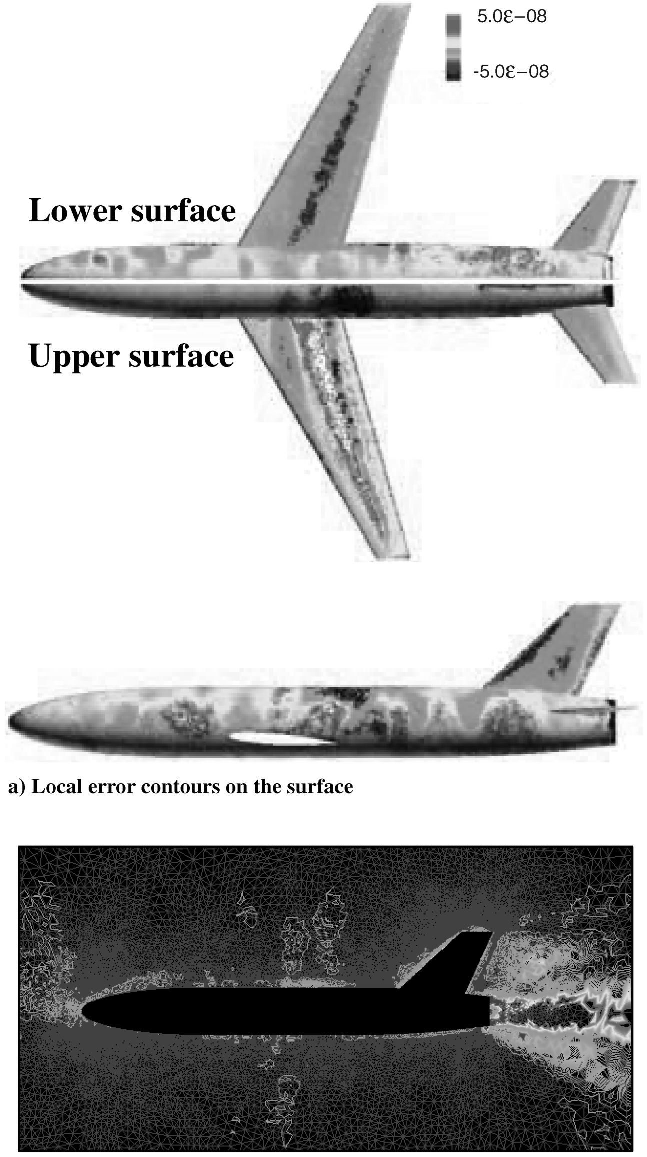


a) Without smoothing

b) With smoothing

Fig. 11 Contours of residual R_5 with and without smoothing for ONERA M5, (The smoothing of the residual of the prolongated flow variable Q_h^H removes high-frequency noise.)

First, the effect of residual smoothing is presented here for the initial mesh. Figure 10 shows the variation of the estimated error in drag for the iteration number of the smoothing process. The result is obtained from the initial mesh with quadratic interpolation and



a) Local error contours on the surface

b) Local error contours on the symmetric plane

Fig. 12 Contours of local error in C_D for ONERA M5.

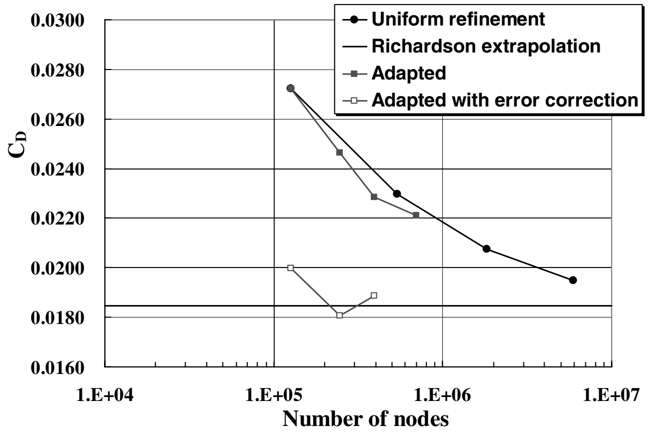
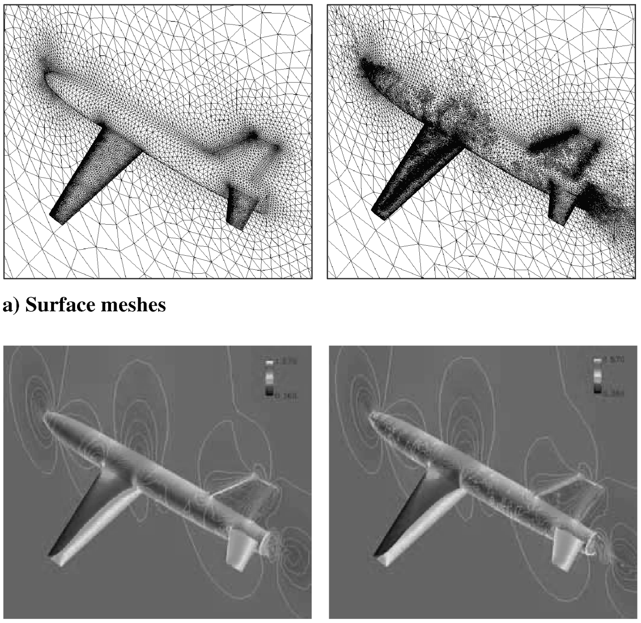


Fig. 13 ONERA M5 drag estimation and adaptive mesh refinement (quadratic interpolation, $N = 2$, with residual smoothing).



a) Surface meshes

b) Pressure contours

Fig. 14 Comparison of initial (left) and adaptively refined (right) ONERA M5 meshed and flow solutions.

$N = 2$. The error between the drag for initial mesh and an extrapolated value from uniformly refined fine meshes is 84 counts, and the error estimation result with residual smoothing presented in Fig. 11 estimates the optimum error as about 75 counts with nine smoothing iterations, whereas the estimated error is only 44 counts without smoothing.

Figure 11 compares contours of the residual vector $R_h(I_h^H Q_H)$ with and without the smoothing process. The noise appears to be very severe especially on the fuselage surface.

The error distribution on the fine mesh obtained by the dot product of the adjoint vector and the smoothed residual in Eq. (11) is shown in Fig. 12. Although it might appear that large errors are located on the fuselage surface, the largest amount of errors exist around the base region as shown in the contours on the symmetric plane.

The mesh adaptation results for ONERA M5 are presented in Fig. 13. The adaptive refinement is conducted 3 times, and the resulting drag coefficients are compared with those computed on uniformly refined meshes. The error-corrected results are converging to the extrapolated drag as the number of the adaptation number increases.

The final mesh has 702,419 nodes and the comparison of initial and final mesh and pressure contours are shown in Fig. 14. The airplane head and base, around the shock wave region, and the leading/trailing edges of the main and tail wings are mainly refined by the adaptation procedure.

V. Conclusions

An adjoint-based error estimation and grid adaptation study has been conducted for three-dimensional inviscid flows with unstructured meshes. A fine mesh was created by a few bisection refinements of the coarse mesh. The error was estimated by the dot product of the adjoint vector and the smoothed residual vector on a fine mesh. The local error term was used as an adaptation parameter for mesh refinement. The present procedure for adaptive mesh refinement has been applied to transonic inviscid flows around ONERA M6 wing and ONERA M5 airplane models. The same level of prediction accuracy for drag was achieved with much less mesh points than uniformly refined fine meshes. It was found that the residual smoothing strategy improves the accuracy of the error estimation by removing noise in the residual vector, and there exists an optimum number of smoothing for the accurate error estimation.

References

[1] Castro-Diaz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., "Anisotropic Unstructured Mesh Adaptation for Flow Simulations," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 4, 1997, pp. 475–491.

[2] R. Löhner, "Generation of Unstructured Grids Suitable for RANS Calculations," AIAA 99-0662, Jan. 1999.

[3] Sharov, D., and Fujii, K., "Three-Dimensional Adaptive Bisection of Unstructured Grids for Transient Compressible Flow Computations," AIAA Paper 95-1708, June 1995.

[4] Mavriplis, D. J., "Adaptive Meshing Techniques for Viscous Flow Calculations on Mixed Element Unstructured Meshes," *International Journal for Numerical Methods in Fluids*, Vol. 34, No. 2, Sept. 2000, pp. 93–111.

[5] Monk, P., and Suli, E., "The Adaptive Computations of Far-Field Patterns by A Posteriori Error Estimation of Linear Functionals," *SIAM Journal on Numerical Analysis*, Vol. 36, No. 1, 1998, pp. 251–274.

[6] Parashivoiu, M., Peraire, J., and Patera, A., "A Posteriori Finite Element Bounds for Linear-Functional Outputs of Elliptic Partial Differential Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 150, No. 3, 1997, pp. 289–312.

[7] Pierce, N. A., and Giles, M. B., "Adjoint Recovery of Superconvergent Functionals from PDE Approximations," *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.

[8] Müller, J.-D., and Giles, M. B., "Solution Adaptive Mesh Refinement Using Adjoint Error Analysis," AIAA 2001-2550, June 2001.

[9] Venditti, D. A., and Darmofal, D. L., "Adjoint Error Estimation and Grid Adaptation for Functional Outputs: Application to Quasi-One-Dimensional Flow," *Journal of Computational Physics*, Vol. 164, No. 1, 2000, pp. 204–227.

[10] Venditti, D. A., and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flows," *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–69.

[11] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.

[12] Park, M. A., "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," *AIAA Journal*, Vol. 42, No. 9, 2004, pp. 1854–1862; also AIAA 2002-3286, June 2002.

[13] Park, M. A., "Three-Dimensional Turbulent RANS Adjoint-Based Error Correction," AIAA 2003-3849, June 2003.

[14] Kim, H. J., Sasaki, D., Obayashi, S., and Nakahashi, K., "Aerodynamic Optimization of Supersonic Transport Wing Using Unstructured Adjoint Method," *AIAA Journal*, Vol. 39, No. 6, June 2001, pp. 1011–1020.

[15] Rivara, M. C., "Selective Refinement/Derefinement Algorithms for Sequences of Nested Triangulations," *International Journal of Numerical Methods in Engineering*, Vol. 28, No. 12, 1989, pp. 2889–2906.

[16] Rivara, M. C., and Levin, C., "A 3D Refinement Algorithm Suitable for Adaptive and Multi-Grid Techniques," *Communications in Applied Numerical Methods*, Vol. 8, No. 5, 1992, pp. 281–290.

[17] Jones, M. T., and Plassmann, P. E., "Adaptive Refinement of Unstructured Finite-Element Meshes," *Finite Elements in Analysis and Design*, Vol. 25, No. 1, 1997, pp. 41–60.

[18] Löhner, R., "Regridding Surface Triangulations," *Journal of Computational Physics*, Vol. 126, No. 1, 1996, pp. 1–10.

[19] Ito, Y., and Nakahashi, K., "Direct Surface Triangulation Using Stereolithography Data," *AIAA Journal*, Vol. 40, No. 3, 2002, pp. 490–496.

[20] Sharov, D., and Nakahashi, K., "Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations," *AIAA Journal*, Vol. 36, No. 3, 1998, pp. 484–486.

[21] Nakahashi, K., Kano, S., Kodera, M., and Sharov, D., "Applications of Unstructured Hybrid Grid Method to High-Reynolds Number Viscous Flows," *International Journal for Numerical Methods in Fluids*, Vol. 31, No. 1, 1999, pp. 97–111.

[22] Sharov, D., and Nakahashi, K., "A Boundary Recovery Algorithm for Delaunay Tetrahedral Meshing," *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, Mississippi University, April 1996, pp. 229–238.