Unstructured Adjoint Method for Navier-Stokes Equations*

Hyoung-Jin KIM** and Kazuhiro NAKAHASHI**

For efficient aerodynamic design optimization, a discrete adjoint code is developed from an unstructured hybrid mesh Navier-Stokes solver. The adjoint code is verified by comparison of flux Jacobian and objective function gradient with a finite difference method. An aerodynamic design tool is developed utilizing the flow solver, adjoint code and a gradient-based optimizer and applied to a design example of a high-lift device. Use of prism layer grid sensitivities is suggested for more efficient gradient calculation from the adjoint analysis of Navier-Stokes equations with unstructured hybrid mesh.

Key Words: Aerodynamic Design, Adjoint, Sensitivity Analysis, High-Lift Device, Unstructured Mesh

1. Introduction

With the advances in computational fluid dynamics (CFD) and computing power, aerodynamic design optimization methods utilizing CFD codes are more important than ever. Among several design optimization methods applicable to aerodynamic designs, the gradient-based method has been used most widely due to its well-developed numerical algorithms and relatively small computational burden.

In the application of gradient-based methods to aerodynamic design problems, one of major concerns is accurate and efficient calculation of sensitivity gradient of objective functions. During the last decade, the adjoint method has grown much attention as an efficient sensitivity analysis method for aerodynamic optimization because it allows to calculate sensitivity information independently with the number of design variables[(1)–(7)].

The unstructured grid approach has several advantages over the structured grid approach. It can treat complex geometry with greater efficiency and less effort. It also has a greater flexibility in the adaptive grid refinement/unrefinement; thus the total number of grid points can be saved.

In this study, a discrete adjoint sensitivity code has been developed from a 3-D unstructured Navier-Stokes solver based on a cell-vertex finite volume method. With the resulting adjoint code, an aerodynamic design example is conducted to validate the performance of the developed design tool utilizing the adjoint code.

* Received 11th November, 2004 (No. 04-4218)

** Department of Aerospace Engineering, Tohoku University, Aobayama 01, Sendai 980–8579, Japan.
E-mail: naka@ad.mech.tohoku.ac.jp

2. Flow Analysis

The Navier-Stokes equations for compressible viscous flows are written in an integral form as follows;

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{Q} dV + \int_{\partial\Omega} (\boldsymbol{F}(\boldsymbol{Q}) - \boldsymbol{G}(\boldsymbol{Q}))\cdot \boldsymbol{n} dS = \int_{\Omega} \boldsymbol{S} dV, \tag{1}$$

where $\boldsymbol{Q} = [\rho, \rho u, \rho v, \rho w, e, \rho\tilde{\nu}]^T$ is the vector of conservative variables; ρ the density; u, v, w the velocity components in the x, y, z directions; e the total energy and $\tilde{\nu}$ the variable for turbulence model equation. The vector $\boldsymbol{F}(\boldsymbol{Q})$ and $\boldsymbol{G}(\boldsymbol{Q})$ represent the inviscid and viscous flux vector and $\boldsymbol{n}$ is the outward normal of $\partial\Omega$ which is the boundary of the control volume Ω. Equation (1) is closed by the perfect gas equation of state with a constant ratio of specific heats, and the laminar viscosity coefficient is obtained by the Sutherland's law. The turbulence viscosity is calculated using the Spalart-Allmaras one-equation turbulence model[(8)].

The equations are solved by a finite volume cell-vertex scheme. The control volume is a non-overlapping dual cell. For a control volume, Eq. (1) can be written as follows;

$$\frac{\partial \boldsymbol{Q}_i}{\partial t} = -\frac{1}{V_i}\sum_{j(i)} \Delta S_{ij}\left(\boldsymbol{f}(\boldsymbol{Q})_{ij} - \boldsymbol{g}(\boldsymbol{Q})_{ij}\right) + \boldsymbol{S}_i \tag{2}$$

$$\boldsymbol{g}(\boldsymbol{Q})_{ij} = Re^{-1}\boldsymbol{G}(\boldsymbol{Q})_{ij}\cdot \boldsymbol{n}_{ij}, \quad \boldsymbol{f}(\boldsymbol{Q})_{ij} = \boldsymbol{F}(\boldsymbol{Q})_{ij}\cdot \boldsymbol{n}_{ij},$$

where ΔS_{ij} is a segment area of the control volume boundary associated with edge connecting nodes i and j. This segment area ΔS_{ij} as well as its unit normal $\boldsymbol{n}_{ij}$ can be computed by summing up the contribution from each

tetrahedron sharing the edge. The subscript of summation, $j(i)$, means all node points connected to node i.

The numerical flux $\boldsymbol{f}(\boldsymbol{Q})_{ij}$ is computed using an approximate Riemann solver of Harten-Lax-van Leer-Einfeldt-Wada (HLLEW)[11]. The second order spatial accuracy is realized by a linear reconstruction of the primitive gas dynamic variables $\boldsymbol{q} = [\rho, u, v, w, p, \tilde{\nu}]^T$ inside the control volume using the following equation;

$$\boldsymbol{q}(\boldsymbol{r}) = \boldsymbol{q}_i + \psi_i \nabla \boldsymbol{q}_i \cdot (\boldsymbol{r} - \boldsymbol{r}_i) \quad (0 \le \psi \le 1), \tag{3}$$

where $\boldsymbol{r}$ is a location vector, and i is the node index. The gradients associated with the control volume centroids are obtained by a least-squares method for the surrounding edges[10]. Venkatakrishnan's limiter[11] is used for the function ψ in Eq. (3).

In order to integrate Eq. (2) in time, the Lower-Upper Symmetric Gauss-Seidel (LU-SGS) method[12] is adopted. With $\Delta\boldsymbol{Q} = \boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n$ and a linearization of numerical flux term as $\boldsymbol{f}_{ij}^{n+1} = \boldsymbol{f}_{ij}^n + \boldsymbol{A}_i^+ \Delta\boldsymbol{Q}_i + \boldsymbol{A}_j^- \Delta\boldsymbol{Q}_j$, where $\boldsymbol{A} = \partial\boldsymbol{f}/\partial\boldsymbol{Q}$, Eq. (2) becomes as follows:

$$\left(\frac{V_i}{\Delta t}\boldsymbol{I} + \sum_{j(i)} \Delta S_{ij} \boldsymbol{A}_i^+\right)\Delta\boldsymbol{Q}_i + \sum_{j(i)} \Delta S_{ij} \boldsymbol{A}_j^- \Delta\boldsymbol{Q}_j = \boldsymbol{R}_i, \tag{4}$$

where $\boldsymbol{R}$ is a residual vector;

$$\boldsymbol{R}_i = -\sum_{j(i)} \Delta S_{ij} (\boldsymbol{f} - \boldsymbol{g})_{ij}^n + V_i \boldsymbol{S}_i \tag{5}$$

The LU-SGS method on unstructured grid can be derived by splitting node points $j(i)$ into two groups, $j \in L(i)$ and $j \in U(i)$, for the second summation in left-hand-side of Eq. (4). This lower/upper splitting for the unstructured grid is realized by using a grid reordering technique[13] to vectorize the LU-SGS method and to improve the convergence. More details of the flow solver can be found in Ref. (14).

3. Adjoint Method

3.1 Formulation

A discrete aerodynamic sensitivity analysis begins with the fact that a discrete residual vector of nonlinear flow equations is null for a converged flow solution of steady problems, which can be written symbolically as

$$\boldsymbol{R}(\boldsymbol{Q}(\beta), \boldsymbol{X}(\beta), \beta) = \boldsymbol{0}, \tag{6}$$

where $\boldsymbol{Q}$ is the flow variable vector, $\boldsymbol{X}$ the grid position vector and β a design variable. Equation (6) can be differentiated via the chain rule with respect to β to yield the following equation of sensitivity, which is also set to zero because the flow solver residual should be converged for the perturbation of β.

$$\frac{d\boldsymbol{R}}{d\beta} = \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}\frac{d\boldsymbol{Q}}{d\beta} + \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial\boldsymbol{R}}{\partial\beta} = \boldsymbol{0} \tag{7}$$

The design objective function F is usually aerodynamic coefficients such as C_D, C_L, or differences between computed and target surface pressures. F is also a function of $\boldsymbol{Q}$, $\boldsymbol{X}$, and β:

$$F = F(\boldsymbol{Q}(\beta), \boldsymbol{X}(\beta), \beta) \tag{8}$$

The sensitivity gradient of cost function F with respect to β can also be obtained by chain rule as follows:

$$\frac{dF}{d\beta} = \frac{\partial F}{\partial\boldsymbol{Q}}\frac{d\boldsymbol{Q}}{d\beta} + \frac{\partial F}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial F}{\partial\beta} \tag{9}$$

Since the total derivative of the residual vector in the steady state is null as shown in Eq. (7), we can introduce adjoint variable vector $\boldsymbol{\Lambda}$ and combine Eqs. (9) and (7) to obtain

$$\frac{dF}{d\beta} = \frac{\partial F}{\partial\boldsymbol{Q}}\frac{d\boldsymbol{Q}}{d\beta} + \frac{\partial F}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial F}{\partial\beta} + \boldsymbol{\Lambda}^T\left(\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}\frac{d\boldsymbol{Q}}{d\beta} + \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial\boldsymbol{R}}{\partial\beta}\right) \tag{10}$$

Setting the sum of coefficients of the flow variable sensitivity vector $d\boldsymbol{Q}/d\beta$ as zero in Eq. (10) gives the following adjoint equation.

$$\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}}^T\boldsymbol{\Lambda} + \frac{\partial F}{\partial\boldsymbol{Q}}^T = \boldsymbol{0} \tag{11}$$

If one finds an adjoint variable vector $\boldsymbol{\Lambda}$ which satisfies the above adjoint equation, one can obtain the sensitivity gradient of F without any information of $d\boldsymbol{Q}/d\beta$. This makes the computational cost for sensitivity analysis independent of the number of design variables. Equation (10) eventually becomes the following form,

$$\frac{dF}{d\beta} = \frac{\partial F}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial F}{\partial\beta} + \boldsymbol{\Lambda}^T\left(\frac{\partial\boldsymbol{R}}{\partial\boldsymbol{X}}\frac{d\boldsymbol{X}}{d\beta} + \frac{\partial\boldsymbol{R}}{\partial\beta}\right) \tag{12}$$

$d\boldsymbol{X}/d\beta$ is the grid sensitivity, which can be calculated by a finite-difference approximation or the direct differentiation of a routine for grid generation or modification. The term $\partial\boldsymbol{R}/\partial\boldsymbol{X} \cdot d\boldsymbol{X}/d\beta$ in Eq. (12) is calculated without any matrix-vector product. Instead, this can be done by directly differentiating those terms in the residual vector $\boldsymbol{R}$ that are explicit functions of the grid vector $\boldsymbol{X}$ with respect to β.

The adjoint equation of Eq. (11) is converted to the following system of algebraic equations with a pseudo time term added and is solved with the LU-SGS scheme.

$$\left(\frac{V_i}{\Delta t}\boldsymbol{I} + \sum_{j(i)} \Delta S_{ij} \boldsymbol{A}_i^{+T}\right)\Delta\boldsymbol{\Lambda}_i + \sum_{j(i)} \Delta S_{ji} \boldsymbol{A}_i^{-T} \Delta\boldsymbol{\Lambda}_j = \boldsymbol{R}\_adj_i \tag{13}$$

where $\boldsymbol{R}\_adj_i$ is the adjoint residual vector defined as

$$\boldsymbol{R}\_adj_i = \frac{\partial\boldsymbol{R}}{\partial\boldsymbol{Q}_i}^T\boldsymbol{\Lambda} + \frac{\partial F}{\partial\boldsymbol{Q}_i}^T \tag{14}$$

Compared with Eq. (4), flux Jacobian matrix $\boldsymbol{A}^-$ in the second summation is calculated at node i instead of node j and the segment area ΔS_{ij} is changed by ΔS_{ji} in the second summation of Eq. (13). This is due to the fact that the flux Jacobian is transposed in the adjoint equation. However, the information on grid reordering used in the LU-SGS routine of the flow solver for convergence improvement and vectorization is still valid here for the adjoint analysis.

The flux Jacobian $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$ in Eq. (14) is a very large banded matrix. In the discrete adjoint method all the elements of Jacobian matrix should be calculated explicitly. If all of the calculated elements are stored in memory and utilized in the remaining iteration steps, computational time would be drastically reduced, but the memory requirement would become prohibitively large for three dimensional problems. On the other hand, if the elements are not stored but recalculated every iteration repetitively, the memory requirement can be remarkably reduced with increased computational costs. This demands a compromise which should be made considering avaliable computer resources.

In this study, among the elements of $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$, stored in memory are those calculated by the differentiation of $\psi_i \nabla \boldsymbol{q}_i$, the reconstruction and limiter terms (see Eq. (3)). Other parts obtained by the differentiation of inviscid and viscous fluxes are recalculated at all iterations of the adjoint analysis.

In order to simplify the differentiation process for $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$, the residual vector $\boldsymbol{R}$ is differentiated by primitive variable $\boldsymbol{q} = [\rho, u, v, w, p, \tilde{\nu}]^T$ rather than by the conservative variable $\boldsymbol{Q}$. Then, the flux Jacobian via the conservative variable can be obtained introducing the transformation matrix $\boldsymbol{M} = \partial \boldsymbol{Q}/\partial \boldsymbol{q}$;

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}^T = \left(\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\frac{\partial \boldsymbol{q}}{\partial \boldsymbol{Q}}\right)^T = \frac{\partial \boldsymbol{q}}{\partial \boldsymbol{Q}}^T \frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}^T = \boldsymbol{M}^{-1T}\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}^T \qquad (15)$$

In this study, the required differentiation process is conducted by human hand.

3.2 Validation of adjoint code

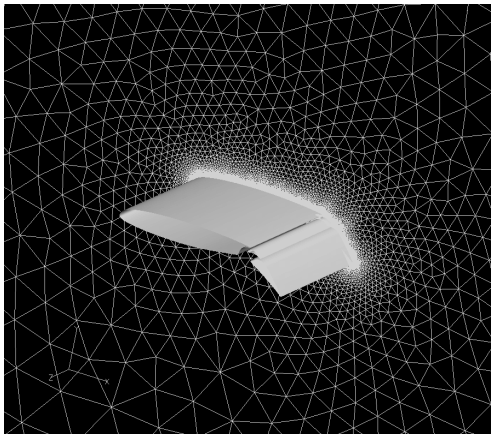
In order to validate the discrete adjoint code developed in this study, flow and sensitivity analyses are conducted for a typical high-lift device configuration with vane and flap. Flow conditions are $M_\infty = 0.2$, incidence angle = 8 degrees and Reynolds number of 8 million.

In order to conduct the two-dimensional problem with the three-dimensional tools, a two-dimensional mesh around the multi-element airfoil is spanwisely extended with only one cell stencil to build a three-dimensional mesh. Figure 1 shows initial geometry and mesh around it. Forty prism layers are located near the viscous wall. The mesh has 149 982 nodes, 526 172 edges, 117 191 tetrahedra, 108 648 prisms and 242 pyramids.

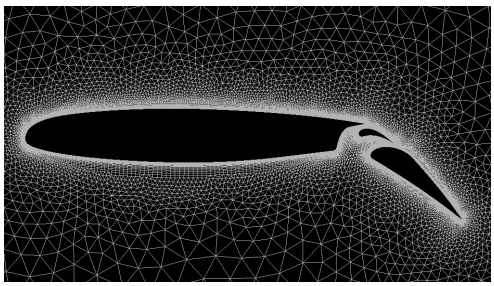
Firstly, for the verification of the flux Jacobian $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$ obtained by manual differentiation of residual vector, we compare the flux Jacobian calculated in the adjoint code with finite difference result. As can be seen in Eq. (14). $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$ can be obtained by setting the ith component of the adjoint vector, $\Lambda_i = 1$ and all other components of $\boldsymbol{\Lambda}$ as zero.

On the other hand, $[\partial \boldsymbol{R}/\partial \boldsymbol{Q}]^T$ can be calculated by finite difference as follows:

$$\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}_j} = \left[\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{Q}}\right] z_j = \frac{\boldsymbol{R}(\boldsymbol{Q}+\varepsilon z) - \boldsymbol{R}(\boldsymbol{Q}-\varepsilon z)}{2\varepsilon}, \qquad (16)$$



(a) Geometry and boundary mesh



(b) Close view

Fig. 1 Mesh system for the High-Lift Device

Table 1 Comparison of 6×6 flux Jacobian matrix $[\partial \boldsymbol{R}_i/\partial \boldsymbol{Q}_j]^T$ by adjoint code and finite difference

FD	9.229E−07	2.373E−07	1.296E−08	−6.871E−10	−3.512E−05	4.515E−04
Adj	9.229E−07	2.373E−07	1.296E−08	−6.871E−10	−3.512E−05	4.515E−04
Diff.	*4.450E−14*	*−8.498E−14*	*−3.191E−13*	*−4.642E−16*	*−7.159E−13*	*−3.867E−11*
FD	−8.286E−05	6.448E−05	3.360E−05	5.632E−08	−2.861E−04	−1.685E−02
Adj	−8.286E−05	6.448E−05	3.360E−05	5.632E−08	−2.861E−04	−1.685E−02
Diff.	*5.713E−13*	*5.984E−13*	*6.837E−13*	*6.802E−16*	*3.124E−13*	*2.950E−11*
FD	−6.060E−05	4.467E−06	4.767E−05	3.763E−08	−2.257E−04	−1.721E−02
Adj	−6.060E−05	4.467E−06	4.767E−05	3.763E−08	−2.257E−04	−1.721E−02
Diff.	*−1.383E−13*	*5.805E−13*	*6.634E−14*	*−1.627E−15*	*−3.236E−13*	*−3.838E−11*
FD	3.826E−08	3.006E−09	−1.824E−08	4.542E−06	1.406E−07	6.515E−06
Adj	3.826E−08	3.006E−09	−1.824E−08	4.542E−06	1.406E−07	6.515E−06
Diff.	*−1.325E−13*	*−1.554E−13*	*−2.878E−13*	*−9.666E−15*	*1.311E−12*	*4.076E−11*
FD	6.276E−05	−3.541E−05	−4.633E−05	−1.957E−08	2.577E−04	1.474E−02
Adj	6.276E−05	−3.541E−05	−4.633E−05	−1.957E−08	2.577E−04	1.474E−02
Diff.	*1.606E−13*	*6.299E−13*	*3.062E−13*	*2.279E−16*	*3.888E−13*	*2.891E−11*
FD	0.000E+00	1.279E−10	3.827E−10	1.832E−13	−2.055E−10	1.522E−05
Adj	0.000E+00	1.279E−10	3.827E−10	1.828E−13	−2.058E−10	1.522E−05
Diff.	*0.000E+00*	*8.211E−15*	*3.929E−14*	*4.177E−16*	*3.398E−13*	*−1.949E−11*

where ε is set to 10^{-7}, and z_j is set to 1 and all other components are zero.

This comparison of flux Jacobian does not require any convergence of flow analysis. Table 1 shows comparison results for two neighboring nodes near flap trailing edge. Differences between the values by the adjoint code and finite difference are as low as machine zero.

Secondly, we directly compare the objective function gradient with respect to a simple geometric variable. If the flux Jacobian is accurate and the iterative solver of the adjoint code is implemented properly, one is supposed to get an accurate sensitivity gradient depending on the level of convergence and step size for the finite difference.

We used the following geometric parameter β for the purpose of test.

Table 2 Comparison of sensitivity derivatives: errors are with respect to the values of finite difference

	Finite Difference	Adjoint	Δ(%)
$dC_L/d\beta$	3.536	3.531	0.14

$$y_{new} = y - \Delta\beta \times x, \tag{17}$$

where x and y are coordinates of longitudinal and vertical directions, respectively. Sensitivity derivatives are compared with those computed by the central difference approximation with a step size $\Delta\beta$ of 10^{-4}. Table 2 compares sensitivity derivatives of aerodynamic coefficients by the finite-difference and adjoint methods. They compare very well with each other.

4. Design Example

4.1 Problem definition and design variables

As a design example, the high-lift device configuration presented in the previous section is optimized. The design objective is to maximize lift for a landing condition with a deflection angle of 35 degree.

The vane and flap are deployed with the same hinge location and deflection angle. Geometric constraints are imposed so that the vane and flap do not interfere with the main element at the cruise condition and during the hinged deployment between the cruise and landing configurations.

Design is conducted only for the unexposed region in the cruise configuration in order not to alter the cruise configuration. Thus geometry of vane and around flap leading edge is perturbed. The vane upper/lower surface and flap leading edge region is modified adding a linear combination of Hicks and Henne shape functions[15].

Additional design variables are location of vane and hinge, vane angle and length, and so on. Employed design variables are summarized as follows.

1) Ten Hicks-Henne shape functions for vane upper/lower surface and flap leading edge region, respectively

2) ΔX, ΔY for Hinge location

3) ΔX, ΔY for Vane movement

4) Vane angle and length

5) Deflection angle of vane and flap

Total number of design variables is 37. Two design cases are conducted: with and without deflection angle as a design variable. Therefore 36 and 37 design variables are used respectively for each design.

4.2 Grid modification and optimizer

When the surface grid is modified by the design parameters, the interior volume grid points should also be moved accordingly. We employed a robust mesh point movement method using spring analogy proposed by Murayama et al.[16], in which spring coefficients are specified so that prism layers around the geometry are moved almost rigidly.

For the minimization of the objective function with the geometric constraints, the Sequential Quadratic Programming (SQP) method[17] implemented in the ADS program[18] was used as an optimizer.

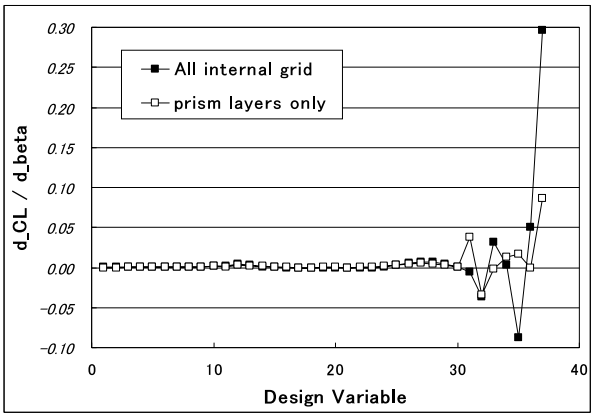
4.3 Grid sensitivity

The grid sensitivity $d\boldsymbol{X}/d\beta$ in Eq. (12) can be calculated either by differentiating the spring analogy method for the interior grid movement or by finite difference method. Although the computational cost with the grid movement procedure is less than one minute at a Compaq Alpha workstation 500 MHz for the present design example, the total computational burden would be a substantial amount if the number of design variables becomes large; e.g. more than one hundred for three-dimensional problems.

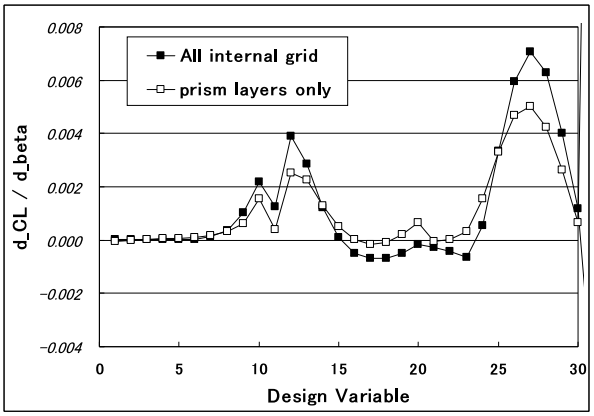
One possible way to reduce the computational burden of the grid sensitivity calculation is to neglect the grid sensitivity of interior node points. As reported in Refs. (5) and (7), in inviscid flows the interior grid sensitivities are required only for design variables associated with translation of the geometries with sharp edges, while grid sensitivities can be ignored for other ordinary design variables for surface perturbation. On the other hand, Anderson and Bonhaus[6] reported that sensitivity derivatives with and without the grid sensitivities differ significantly for viscous flows, and therefore, the design could fail if the grid sensitivity terms were not included.

In this study, we compared the accuracy of sensitivity derivatives with all interior grid sensitivities and with prism layers only. The grid sensitivity of node points in the prism layers can be easily obtained without any iterative calculation because the prism layers go under a rigid motion for surface perturbations. Figure 2 shows the comparison results. The sensitivity derivatives using only the surface grid sensitivities were also calculated but not presented here because they are totally different in magnitude and trend. It can be noted in Fig. 2 that for design variables 1 – 30, the sensitivity derivatives with prism layer grid sensitivities are in the same trend with the results with all the internal grid sensitivities. These design variables 1 – 30 are coefficients of Hicks-Henne shape functions and therefore are not related with geometry translation or rotation. On the other hands, remaining design variable 31 – 37 are related with body translation or rotation, and thus not in agreement with the results using all interior grid sensitivities.

The accuracy of sensitivity derivatives with prism layer grid sensitivity would of course be dependent on geometry and flow conditions under design, number of prism layers, etc. In the present study we used all the interior grid sensitivities because the computational cost for grid sensitivities is negligible for the present study with only 37 design variables. However, the use of prism layer grid

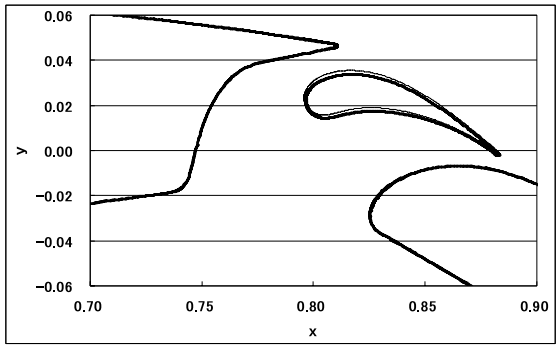


(a) All design variables

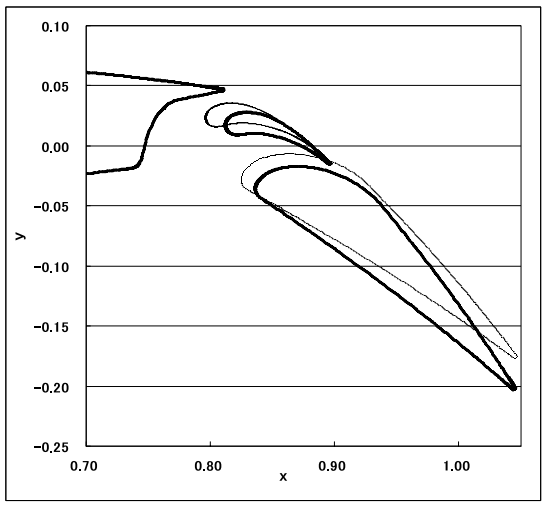


(b) Zoomed view for 1 – 30 design variables

Fig. 2 Effect of grid sensitivity simplification on sensitivity derivatives



(a) 36 design variables



(b) 37 design variables

Fig. 3 Design geometry (thin: initial, thick: design)

Table 3 Comparison of lift coefficients

	Initial	DV36	DV37
C_l	3.478	3.491	3.602

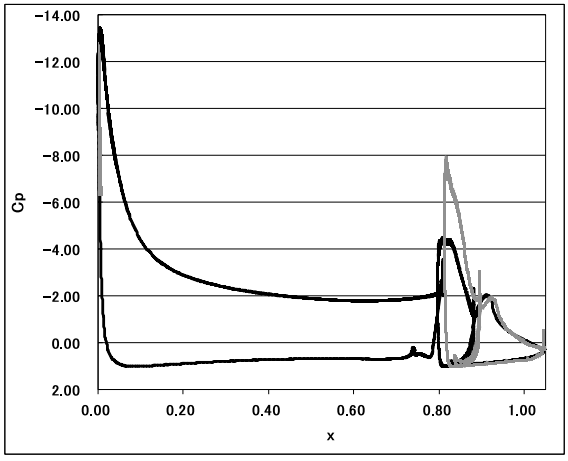


Fig. 4 Comparison of surface pressure distributions (thin: initial, thick: 36 dv, gray: 37 dv)

sensitivities taking advantage of the hybrid grid needs to be investigated further for more efficient sensitivity analysis of three dimensional complex geometries with a large number of design variables.

4.4 Design results

Figure 3 compares the initial and design configurations by 36 and 37 design variables, respectively. Table 3 summarizes the design results. The lift coefficient was increased by 130 counts for the design with 36 design variables and by 1 240 counts for the design with 37 design variables while both meeting all the geometric constraints. The maximum deflection angle variation was limited as five degrees, and as expected, the design result has the maximum allowed angle. The design results imply that flap deflection angle has the dominant effect on the lift, and the amount of lift increment without flap deflection angle is very limited.

Figure 4 compares the surface pressure distributions. In both of the design results, lift increments are caused by increased suction peak at the vane upper surface. Main surface pressures show little difference, and flap surface pressures only have visible changes for the 37 design variable case.

5. Concluding Remarks

An aerodynamic design method has been developed

using a three-dimensional unstructured Navier-Stokes code and its discrete adjoint code. The adjoint code is verified by comparison of flux Jacobian and objective function gradient with a finite difference. The developed design tool was successfully applied to a High-Lift Device design example. Use of prism layer grid sensitivities is suggested for more efficient gradient calculation from the Navier-Stokes adjoint analysis with unstructured hybrid mesh.

References

(1) Jameson, A., Aerodynamic Shape Optimization Using the Adjoint Method, Lecture Series, Von Karman Institute for Fluid Dynamics, February, (2003).

(2) Kim, S., Alonso, J.J. and Jameson, A., Design Optimization of High-Lift Configurations Using a Viscous Continuous Adjoint Method, AIAA 2002-0844, January, Reno, NV, (2002).

(3) Kim, C.S., Kim, C. and Rho, O.H., Sensitivity Analysis for the Navier-Stokes Equations with Two-Equation Turbulence Models, AIAA J., Vol.39, No.5 (2001), pp.838–845.

(4) Nielson, E.J. and Anderson, W.K., Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations, AIAA J., Vol.37, No.11 (1999), pp.1411–1419.

(5) Anderson, W.K. and Venkatakrishnan, V., Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation, Computers and Fluids, Vol.28, No.4-5 (1999), pp.443–480.

(6) Anderson, W.K. and Bonhaus, D.L., Airfoil Design on Unstructured Grids for Turbulent Flows, AIAA J., Vol.37, No.2 (1999), pp.1185–1191.

(7) Kim, H.J., Sasaki, D., Obayashi, S. and Nakahashi, K., Aerodynamic Optimization of Supersonic Transport Wing Using Unstructured Adjoint Method, AIAA J., Vo.39, No.6 (2001), pp.1011–1020.

(8) Spalart, P.R. and Allmaras, S.R., A One-Equation Turbulence Model for Aerodynamic Flows, Recherche Aerospatiale, Vol.1 (1994), pp.5–21.

(9) Obayashi, S. and Guruswamy, G.P., Convergence Acceleration of an Aeroelastic Navier-Stokes Solver, AIAA J., Vol.33, No.6 (1995), pp.1134–1141.

(10) Haselbacher, A. and Blazek, J., On the Accurate and Efficient Discretization of the Navier-Stokes Equations on Mixed Grids, AIAA 99-3363, (1999).

(11) Venkatakrishnan, V., On the Accuracy of Limiters and Convergence to Steady State Solutions, AIAA Paper 93-0880, January, (1993).

(12) Yoon, S. and Jameson, A., Lower-Upper Symmetric-Gauss-Seidel Method for the Euler and Navier-Stokes Equations, AIAA J., Vol.26, No.9 (1988), pp.1025–1026.

(13) Sharov, D. and Nakahashi, K., Reordering of Hybrid Unstructured Grids for Lower-Upper Symmetric Gauss-Seidel Computations, AIAA J., Vol.36, No.3 (1998), pp.484–486.

(14) Nakahashi, K., Kano, S., Kodera, M. and Sharov, D., Applications of Unstructured Hybrid Grid Method to High-Reynolds Number Viscous Flows, International J. for Numerical Methods in Fluids, No.31, (1999), pp.97–111, John Wiley & Sons.

(15) Hicks, R.M. and Henne, P.A., Wing Design by Numerical Optimization, J. of Aircraft, Vol.15, No.7 (1978), pp.407–412.

(16) Murayama, M., Nakahashi, K. and Matsushima, K., Unstructured Dynamic Mesh for Large Movement and Deformation, AIAA 2002-0122, January, (2002).

(17) Vanderplaats, G.N., Numerical Optimization Techniques for Engineering Design: With Applications, (1984), pp.3641–3646, McGraw-Hill, N.Y.

(18) Vanderplaats, G.N., ADS — A Fortran Program for Automated Design Synthesis Version 3.00; Users' Guide, Engineering Design Optimization, INC., (1987).