ICAS 2004
24TH INTERNATIONAL CONGRESS OF THE AERONAUTICAL SCIENCES

AERODYNAMIC DESIGN OPTIMIZATION USING UNSTRUCTURED NAVIER-STOKES ADJOINT METHOD

Hyoung-Jin Kim*, Kazuhiro Nakahashi**
Tohoku University, Sendai 980-8579 JAPAN

Keywords: *aerodynamic design, adjoint method, high lift device*

Abstract

For efficient aerodynamic design optimizations, an adjoint code is developed from an unstructured Navier-Stokes solver. The Navier-Stokes solver is based on a cell-vertex finite volume method and utilizes hybrid grid to accurately resolve wall boundary layers for high Reynolds number viscous flows. The adjoint code is developed by a discrete approach so that the Navier-Stokes flow solver and the adjoint code are exactly consistent with each other. An aerodynamic design tool is developed utilizing the flow solver, adjoint code and gradient-based optimizer and applied to a design example of High-Lift Device. Successful design results confirm validity and efficiency of the present design method.

1 Introduction

With the advances in computational fluid dynamics (CFD) and computing power of modern computers, aerodynamic design optimization methods utilizing CFD codes are more important than ever. Among several design optimization methods applicable to aerodynamic design problems, 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 practical aerodynamic design problems, one of major concerns is accurate and efficient calculation of sensitivity gradient of aerodynamic 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 one to calculate sensitivity information independently with the number of design variables.[1-9]

For complex aerodynamic configurations, the unstructured grid approach has several advantages over the structured grid approach. This approach 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, aerodynamic design examples are conducted to validate the performance of the developed design tool utilizing the adjoint code.

2 Flow and Adjoint Analysis

2.1 Flow Analysis

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

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV+\int_{\partial\Omega}\left(\mathbf{F}(\mathbf{Q})-\mathbf{G}(\mathbf{Q})\right)\cdot\mathbf{n}dS=\int_{\Omega}\mathbf{S}dV \qquad (1)$$

where $\mathbf{Q}=[\rho,\rho u,\rho v,\rho w,e,\rho\tilde{v}]^T$ 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 $\tilde{v}$ the variable for turbulence model equation. The vector $\mathbf{F}(\mathbf{Q})$ and

$\mathbf{G}(\mathbf{Q})$ represent the inviscid and viscous flux vector and $\mathbf{n}$ is the outward normal of $\partial\Omega$ which is the boundary of the control volume Ω. Eq.(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 Spallart-Almaras one-equation turbulence model.[10]

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;

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

$$\mathbf{f}(\mathbf{Q})_{ij} = \mathbf{F}(\mathbf{Q})_{ij}\cdot\mathbf{n}_{ij},$$

$$\mathbf{g}(\mathbf{Q})_{ij} = \mathrm{Re}^{-1}\,\mathbf{G}(\mathbf{Q})_{ij}\cdot\mathbf{n}_{ij}$$

, 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 $\mathbf{n}_{ij}$ 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 $\mathbf{f}(\mathbf{Q})_{ij}$ 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 $\mathbf{q} = [\rho, u, v, w, p, \tilde{\nu}]^{\mathrm{T}}$ inside the control volume using the following equation;

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

, where $\mathbf{r}$ 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.[12] Venkatakrishnan's limiter [13] is used for the function ψ in Eq.(3) because of its superior convergence properties.

In order to integrate Eq. (2) in time, the Lower-Upper Symmetric Gauss-Seidel(LU-SGS) implicit method [14] is adopted. With $\Delta\mathbf{Q}=\mathbf{Q}^{n+1}$ - $\mathbf{Q}^{n}$ and a linearization of numerical flux term as $\mathbf{f}_{ij}^{n+1} = \mathbf{f}_{ij}^{n} + \mathbf{A}_i^{+}\Delta\mathbf{Q}_i + \mathbf{A}_j^{-}\Delta\mathbf{Q}_j$, where $\mathbf{A} = \partial\mathbf{f}/\partial\mathbf{Q}$, Eq.(2) becomes the following equations.

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

, where $\mathbf{R}$ is a residual vector;

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

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 LHS of Eq.(4). The final form of the LU-SGS method for the unstructured grid becomes,

Forward sweep:

$$\Delta\mathbf{Q}_i^{*} = \mathbf{D}^{-1}\left[\mathbf{R}_i - \sum_{j\in L(i)}\Delta S_{ij}A_j^{-}\Delta\mathbf{Q}_j^{*}\right] \qquad (6a)$$

Backward sweep:

$$\Delta\mathbf{Q}_i = \Delta\mathbf{Q}_i^{*} - \mathbf{D}^{-1}\sum_{j\in U(i)}\Delta S_{ij}A_j^{-}\Delta\mathbf{Q}_j \qquad (6b)$$

$$\mathbf{D} = \left(\frac{V_j}{\Delta t} + 0.5\sum_{j(i)}\Delta S_{ij}\rho_A\right)\mathbf{I} \qquad (7)$$

The term $\mathbf{D}$ is a diagonal matrix derived by Jameson-Turkel approximation of Jacobian[15] as $\mathbf{A}^{\pm} = 0.5(\mathbf{A} \pm \rho_A\mathbf{I})$, where ρ_A is a spectral radius $|U| + a$ of Jacobean $\mathbf{A}$ added with the substitute of viscous flux Jacobian $\partial\mathbf{g}/\partial\mathbf{Q}$ as follows;

$$\rho_A = \chi(|U| + a) + 2\frac{\mu}{\mathrm{Re}\cdot\rho\cdot h}$$

, where h is a distance between the nodes i and j, and χ is chosen to be 1.01.

The lower/upper splitting of Eq.(6) for the unstructured grid is realized by using a grid reordering technique [16] to vectorize the LU-SGS method and to improve the convergence. More details of the flow solver can be found in Ref.[17]

2.2 Adjoint Method

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

$$\mathbf{R}\left(\mathbf{Q}(\beta),\mathbf{X}(\beta),\beta\right)=\mathbf{0}\,, \qquad (8)$$

where **Q** is the flow variable vector, **X** the grid position vector and *β* a design variable. Equation (8) 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*.

$$\frac{d\mathbf{R}}{d\beta}=\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\frac{d\mathbf{Q}}{d\beta}+\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\frac{d\mathbf{X}}{d\beta}+\frac{\partial\mathbf{R}}{\partial\beta}=\mathbf{0} \qquad (9)$$

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

$$F=F\left(\mathbf{Q}(\beta),\mathbf{X}(\beta),\beta\right) \qquad (10)$$

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

$$\frac{dF}{d\beta}=\frac{\partial F}{\partial\mathbf{Q}}^{T}\frac{d\mathbf{Q}}{d\beta}+\frac{\partial F}{\partial\mathbf{X}}^{T}\frac{d\mathbf{X}}{d\beta}+\frac{\partial F}{\partial\beta} \qquad (11)$$

Since the total derivative of the residual vector in the steady state is null as shown in Eq.(9), we can introduce adjoint variable vector **Λ** and combine Eqs.(11) and (9) to obtain

$$\begin{aligned}\frac{dF}{d\beta}&=\frac{\partial F}{\partial\mathbf{Q}}^{T}\frac{d\mathbf{Q}}{d\beta}+\frac{\partial F}{\partial\mathbf{X}}^{T}\frac{d\mathbf{X}}{d\beta}+\frac{\partial F}{\partial\beta}\\&+\mathbf{\Lambda}^{T}\left(\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\frac{d\mathbf{Q}}{d\beta}+\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\frac{d\mathbf{X}}{d\beta}+\frac{\partial\mathbf{R}}{\partial\beta}\right)\end{aligned} \qquad (12)$$

Setting the sum of coefficients of the flow variable sensitivity vector *d***Q**/*dβ* as zero in Eq.(12) gives the following adjoint equation.

$$\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}^{T}\mathbf{\Lambda}+\frac{\partial F}{\partial\mathbf{Q}}=0 \qquad (13)$$

If one finds an adjoint variable vector **Λ** which satisfies the above adjoint equation, one can obtain the sensitivity gradient of F without any information of *d***Q**/*dβ*. This makes the computational cost for the sensitivity analysis independent of the number of design variables. Eq.(21) eventually becomes the following form,

$$\frac{dF}{d\beta}=\frac{\partial F}{\partial\mathbf{X}}^{T}\frac{d\mathbf{X}}{d\beta}+\frac{\partial F}{\partial\beta}+\mathbf{\Lambda}^{T}\left(\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\frac{d\mathbf{X}}{d\beta}+\frac{\partial\mathbf{R}}{\partial\beta}\right) \qquad (14)$$

*d***X**/*dβ* is the grid sensitivity, which can be calculated by a finite-difference approximation or the direct differentiation of a routine for the grid generation or modification. The term $\frac{\partial\mathbf{R}}{\partial\mathbf{X}}\frac{d\mathbf{X}}{d\beta}$ in Eq.(14) is calculated without any matrix-vector product. Instead, this can be done by directly differentiating those terms in the residual vector **R** that are explicit functions of the grid vector **X** with respect to *β*.
The adjoint equation of Eq.(13) is converted to the following system of algebraic equations with a pseudo time term added and is solved with the LU-SGS scheme.

$$\left(\frac{V_i}{\Delta t}\mathbf{I}+\sum_{j(i)}\Delta S_{ij}\mathbf{A}_i^{+T}\right)\Delta\lambda_i+\sum_{j(i)}\Delta S_{ji}\mathbf{A}_i^{-T}\Delta\lambda_j=R\_adj_i \qquad (15)$$

where *R_adj*$_i$ is the adjoint residual defined as

$$R\_adj_i=\frac{\partial\mathbf{R}}{\partial\mathbf{Q}_i}^{T}\mathbf{\Lambda}+\frac{\partial F}{\partial\mathbf{Q}_i} \qquad (16)$$

Flux Jacobian matrix 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.(15). 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 the convergence improvement and vectorization is still valid here for the adjoint equations.

As mentioned earlier, the flux Jacobian $[\partial\mathbf{R}/\partial\mathbf{Q}]^T$ in the RHS of Eq.(16) 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 $[\partial \mathbf{R}/\partial \mathbf{Q}]^T$, stored in memory are those calculated by the differentiation of $\psi_i \nabla \mathbf{q}_i$, the reconstruction and limiter terms (see Eq.(3)). Other parts obtained by the differentiation of inviscid and viscous fluxes are recalculated at every iteration of the adjoint analysis instead of being stored in memory.

Figure 1 compares a two-dimensional example of flux accumulation for the flow solver and the adjoint method. In the flow solver, primitive flow variables are reconstructed at the control volume surface using surrounding node point values. Then the flux **f** through the control volume surface is calculated and acculuated at both nodes 1 and 2. This is repeated for all edges to obtain flux residual for the control volume. On the other hand, in the adjoint method, the adjoint flux $\frac{\partial R_i}{\partial Q}^T \Lambda$ is accumulated at all the node points that have effects on the reconstructed flow variables at the control volume surface. For example, if we set the flux for the edge connecting node 1 and node 2 as R_{12}, accumulation of the adjoint residual R_adj is made at nodes related with node 1 as follows.

$$\text{R\_adj}_j = \text{R\_adj}_j + \frac{\partial R_{12}}{\partial Q_j}^T \Lambda_1, \quad \text{j=1, 2, 3, ..,7}$$

For nodes surrounding node 2,

$$\text{R\_adj}_j = \text{R\_adj}_j - \frac{\partial R_{12}}{\partial Q_j}^T \Lambda_2, \quad \text{j=1, 2, 3, 7, 8, 9, 10.}$$

This causes small loops for the neighboring nodes to be inserted into the big loop for all edges. The length of the small loop is usually from 5 to 30 around a node point for a three dimensional grid depending on the grid topology. If the adjoint code is run on a vector machine, the small loop of neighbor nodes should be enrolled and another edge coloring is required for the vectorization with the big loop of edges.

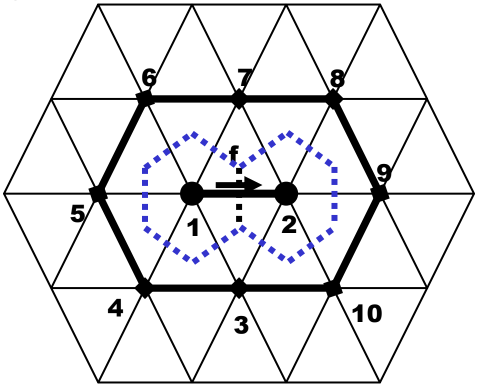


Fig.1 A simple example of flux accumulation for the adjoint method

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

$$\frac{\partial \mathbf{R}}{\partial \mathbf{Q}}^T = \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \mathbf{Q}} \right)^T = \frac{\partial \mathbf{q}}{\partial \mathbf{Q}}^T \frac{\partial \mathbf{R}}{\partial \mathbf{q}}^T = \mathbf{M}^{-1^T} \frac{\partial \mathbf{R}}{\partial \mathbf{q}}^T \qquad (17)$$

The transformation matrices in a transposed form are given as

$$\mathbf{M}^T = \begin{bmatrix} 1 & u & v & w & (u^2+v^2+w^2)/2 & \tilde{\nu} \\ 0 & \rho & 0 & 0 & \rho u & 0 \\ 0 & 0 & \rho & 0 & \rho v & 0 \\ 0 & 0 & 0 & \rho & \rho w & 0 \\ 0 & 0 & 0 & 0 & 1/(\gamma-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho \end{bmatrix},$$

$$\mathbf{M}^{-1^T} = \begin{bmatrix} 1 & -u/\rho & -v/\rho & -w/\rho & (\gamma-1)(u^2+v^2+w^2)/2 & -\tilde{\nu}/\rho \\ 0 & 1/\rho & 0 & 0 & -(\gamma-1)u & 0 \\ 0 & 0 & 1/\rho & 0 & -(\gamma-1)v & 0 \\ 0 & 0 & 0 & 1/\rho & -(\gamma-1)w & 0 \\ 0 & 0 & 0 & 0 & (\gamma-1) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\rho \end{bmatrix}$$

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

2.2 Boundary Conditions for Discrete Adjoint Method

Boundary conditions for the direct method can be simply imposed by differentiating the boundary conditions for the flow equations. This section is thus mainly devoted to the boundary conditions for the discrete adjoint method. The adjoint equation (14) can be written in a more detail form containing boundary conditions as follows.

$$\begin{bmatrix} \dfrac{\partial \mathbf{R}^{i}}{\partial \mathbf{Q}^{i}}^{T} & \dfrac{\partial \mathbf{R}^{b}}{\partial \mathbf{Q}^{i}}^{T} \\ \dfrac{\partial \mathbf{R}^{i}}{\partial \mathbf{Q}^{b}}^{T} & \dfrac{\partial \mathbf{R}^{b}}{\partial \mathbf{Q}^{b}}^{T} \end{bmatrix} \begin{Bmatrix} \mathbf{\Lambda}^{i} \\ \mathbf{\Lambda}^{b} \end{Bmatrix} + \begin{Bmatrix} \dfrac{\partial F}{\partial \mathbf{Q}^{i}} \\ \dfrac{\partial F}{\partial \mathbf{Q}^{b}} \end{Bmatrix} = \mathbf{0} \qquad (18)$$

or

$$\frac{\partial \mathbf{R}^{i}}{\partial \mathbf{Q}^{i}}^{T} \mathbf{\Lambda}^{i} + \frac{\partial \mathbf{R}^{b}}{\partial \mathbf{Q}^{i}}^{T} \mathbf{\Lambda}^{b} + \frac{\partial F}{\partial \mathbf{Q}^{i}} = \mathbf{0} \qquad (19a)$$

$$\frac{\partial \mathbf{R}^{i}}{\partial \mathbf{Q}^{b}}^{T} \mathbf{\Lambda}^{i} + \frac{\partial \mathbf{R}^{b}}{\partial \mathbf{Q}^{b}}^{T} \mathbf{\Lambda}^{b} + \frac{\partial F}{\partial \mathbf{Q}^{b}} = \mathbf{0} \qquad (19b)$$

where the superscript i presents values of inner node, and b values of boundary nodes. For example, $\mathbf{R}^{i}$ is the residual at nodes in computational domain, and $\mathbf{R}^{b}$ is the residual of the boundary conditions at boundary nodes. The boundary conditions can be treated explicit or implicit manner. In the explicit manner, the adjoint variable vector at boundary nodes, Λ^{b} is calculated from Eq.(19b) with Λ^{i} of the previous time level and the flux Jacobians of $\left[\partial \mathbf{R}^{i} / \partial \mathbf{Q}^{b}\right]^{T}$ and $\left[\partial \mathbf{R}^{b} / \partial \mathbf{Q}^{b}\right]^{T}$. And then the calculated value of Λ^{b} is used in Eq.(19a). In the implicit way, the Eq(19b) itself is arranged for Λ^{b} and inserted into Eq(19a) without explicit calculation of the values of Λ^{b}. Equation (19a) is solved in an incremental form of Eq.(15).

2.3 Validation of Adjoint code

In order to validate the direct and adjoint sensitivity codes developed in this study, sensitivity analyses are conducted for an High-Lift device which is selected for a design example in section 3. Flow conditions are $M_{\infty} = 0.2$, $\alpha = 8$ degree and Reynolds number of $8\text{x}10^{6}$. We used the following geometric parameter β for the purpose of test.

$$y_{new} = y - \Delta\beta * x, \qquad (20)$$

, 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 Δβ of 10^{-4}. Table 1 compares sensitivity derivatives of aerodynamic coefficients by the finite-difference and adjoint methods. They compare very well with one another with errors less than 0.14 %.

Table 1 Comparison of sensitivity derivatives: errors are with respect to the values of FD

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

3 Design Example

3.1 Definition of Design Problem

As a design example, a typical high-lift device configuration is optimized. The high-lift device is a multi-element airfoil with a vane and flap as shown in Fig.2. The vane and flap are assumed to be deployed with the same rotation angle.

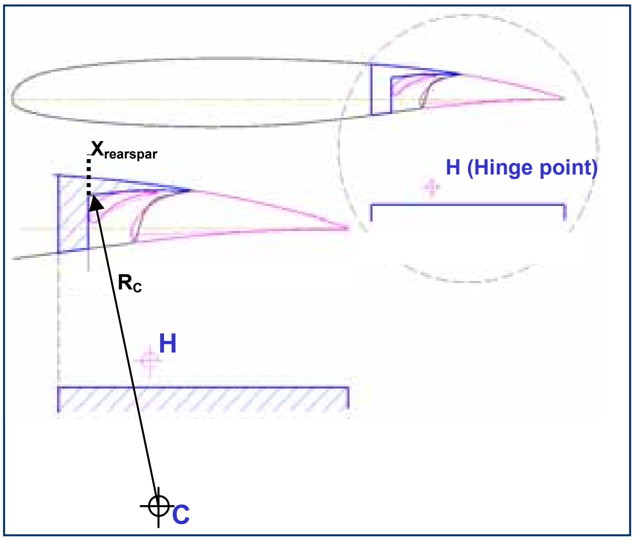


Fig.2 Cruise configuration with retracted vane and flap

The design objective is to maximize lift for a landing condition with a deflection angle of 35 degree. Geometric constraints are imposed so that the vane and flap not interfere with the main element at a cruise condition and also for the hinged deployment between the cruise and landing configurations. The geometric constraints can be represented as follows

1) $X_{\text{rear spar}}$ - $X_{\text{vane}} \le 0$

The vane leading edge should not intrude the main element.

2) Y_{FLAP}-$Y_{\text{main\_TE}} \le 0$

Y(vertical) coordinates of flap surface nodes should not touch main element trailing edge.

3) $|\mathbf{X}_{\mathbf{vane}}$ $\mathbf{-C}|$ $\mathbf{-R_C} \le 0$

Vane upper surface should not intrude the main element lower surface which has a curvature radius of R_C with center C.

4) $|\mathbf{X}_{\mathbf{vane}}$ $\mathbf{-H}|$ $\mathbf{-(R_C\text{-}R_H)} \le 0$

Vane upper surface should not intrude the main element during deployment.

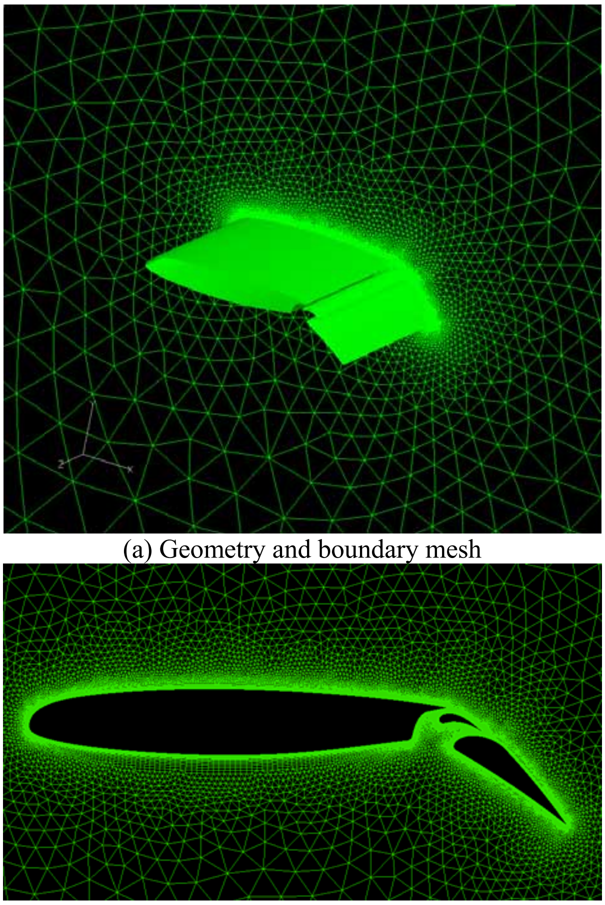
Constraints 1~3 are for the retracted cruise condition, Constraint 4 is for the deployment of vane and flap.

Design conditions are free-stream Mach number of 0.2 and incidence angle of 8 degree. In order to conduct the two-dimensional problem with the three-dimensional tools, the two-dimensional mesh is extended spanwisely with only one cell stencil.

Figure 3 shows initial geometry and mesh around it. The mesh has 149,982 nodes, 526,172 edges, 117,191 tetrahedra, 108,648 prisms and 242 pyramids.

3.2 Design Variables

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 function[18], f_k as follows.



(a) Geometry and boundary mesh

(b) close view

Fig.3 Mesh system for the High-Lift Device

$$y_{new} = y_{initial} + \sum_{k=1}^{n_v} \beta_k \cdot f_k \, , \qquad (20)$$

$$f_k = \sin^3[\pi x^{e(k)}], \quad e(k) = \frac{1n(0.5)}{in(x_k)},$$

where β_k are design variables, n_v the number of design variables, and x_k represents the peak location of f_k. Figure 4 shows adopted ten Hicks-Henne shape functions.

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

1) Ten Hicks-Henne shape functions for vane upper/lower and flap leading edge, 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, and two designs are conducted: with and without deflection angle as a design variable. And thus 36 and

37 design variables are used respectively for each design.

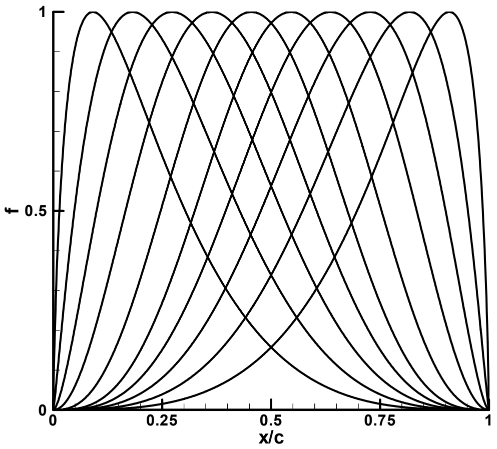


Fig.4 Hicks-Henne shape functions

3.3 Grid Modification

When the surface grid is modified by the design parameters, the interior volume grid points should also be moved accordingly. In the structured grid approach, the interior grid positions can be moved with a relative ease using an algebraic mesh movement strategy which modifies the grid point coordinates along a grid line of the same index. In the unstructured grid method, however, such a simple grid modification method cannot be applied, and an efficient and robust grid movement method is required.

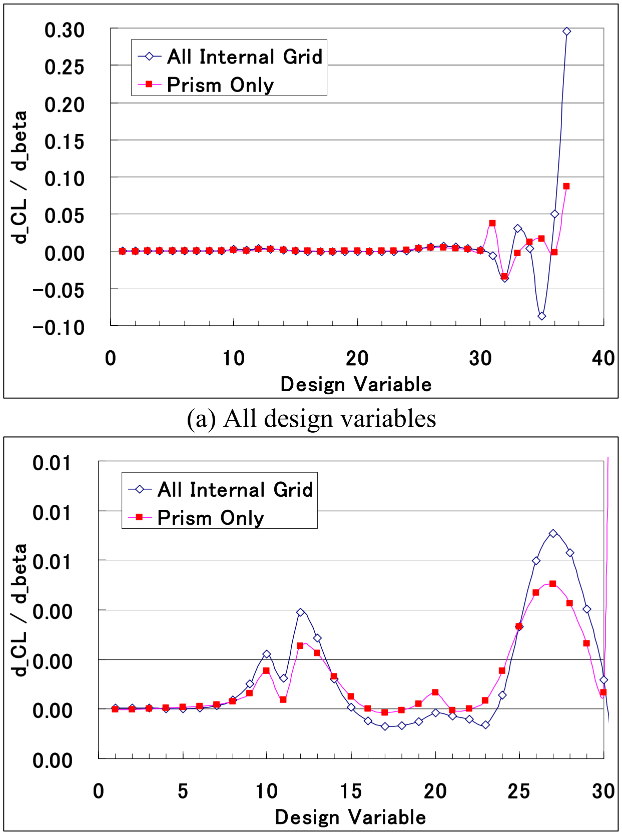
For the movement of the grid points with the perturbed surface grid, we employed the mesh point movement method using spring analogy proposed by Murayama et al.[19], in which spring coefficients are specified so that prism layers around the geometry are moved almost rigidly.

3.4 Grid Sensitivity

The grid sensitivity $d\mathbf{X}/d\beta$ in Eq.(14) 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 a minute at a Compaq Alpha workstation 500MHz for the present design example, the total computational burden would be a substantial amount if the number of design variables becomes large; say, 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.[5,8], in inviscid flows the interior grid sensitivities are required for design variables associated with translation of the geometries with sharp edges, and, on the other hand, the grid sensitivities can be ignored for other ordinary design variables.

On the other hand, Anderson and Bonhaus[6] compared the accuracy of sensitivity derivatives with and without interior grid sensitivities with an adjoint code for Navier-Stokes equations with a one equation turbulence model. In their work, it was reported that derivatives with and without the grid sensitivities differ significantly, and therefore, the design could fail if the grid sensitivity terms were not included.



(a) All design variables

(b) Zoomed view for 1~30 design variables

Fig.5 Effect of grid sensitivity simplification on sensitivity derivatives

In this study, we compared the accuracy of sensitivity derivatives without 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. Figure 5 shows the comparison results. The sensitivity derivatives with surface grid sensitivity only was also calculated, but not presented here because they are totally different in magnitude and trend. It can be noted in Fig.5 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 are the coefficient of Hicks-Henne shape function 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 exact results.

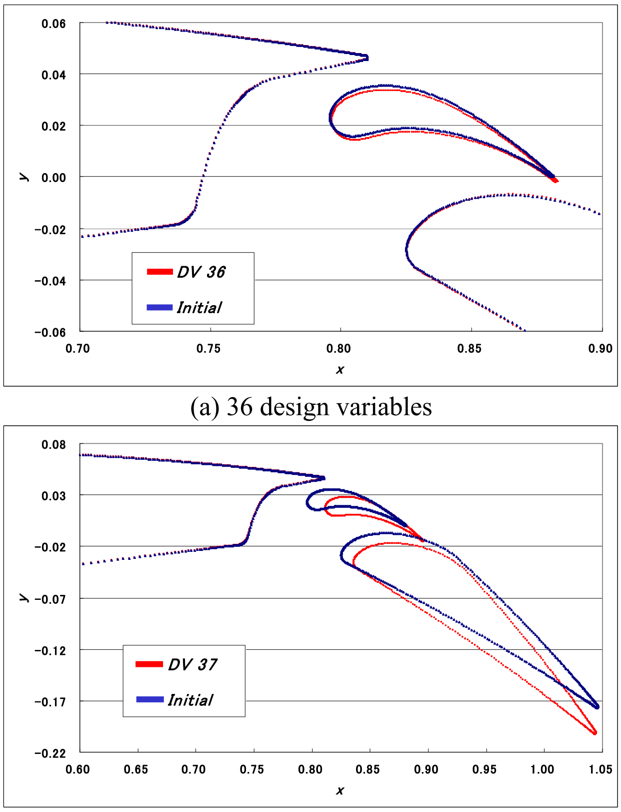
The accuracy of sensitivity derivatives with prism layer grid sensitivity would of course be dependent on geometry and flow conditions under design, number of prism layers, etc. In the present study we used the exact sensitivity derivatives because the computational cost for the grid sensitivity is negligible for the present study with only 37 design variables. However, the simplified grid sensitivity taking advantage of the hybrid grid needs to be investigated further for more efficient design of three dimensional complex geometries with very large design variables.

3.5 Design Algorithm

For the minimization of the objective function with specified constraints, the ADS program[19] was used as an optimizer. The Sequential Quadratic Programming (SQP) method[20] is adopted in which the objective is approximated by a quadratic Taylor series expansion to create a direction-finding problem. This subproblem is solved using the Modified Method of Feasible Directions. Lagrangian multipliers are calculated at the optimum of the subproblem. Then one-dimensional search is conducted using quadratic polynomial interpolation. When the one design iteration is complete, the approximated Hessian matrix is updated by the Broydon-Fletcher-Goldfarb-Shanno(BFGS) formula. Detailed algorithms and methodologies of the SQP method is described in Ref.20.

3.6 Design Results

Figure 6 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 was limited as 5 degrees, and as expected, the design result has the maximum 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.



(a) 36 design variables

(b) 37 design variables

Fig.6 Design geometry

Figure 7 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 slight change for the 37 design variable case, which however, does not seem to distribute to the lift increase. Pressure contours around the initial design airfoils are shown in Fig.8. Low pressure zone by leading edge suction is remarkably visible in the result of 37 design variables.

Table 2 Comparison of lift coefficients

	Initial	DV36	DV37
C_l	3.478	3.491	3.602

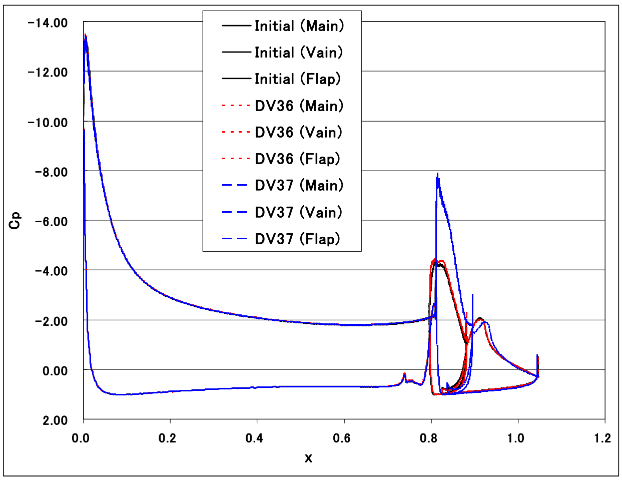
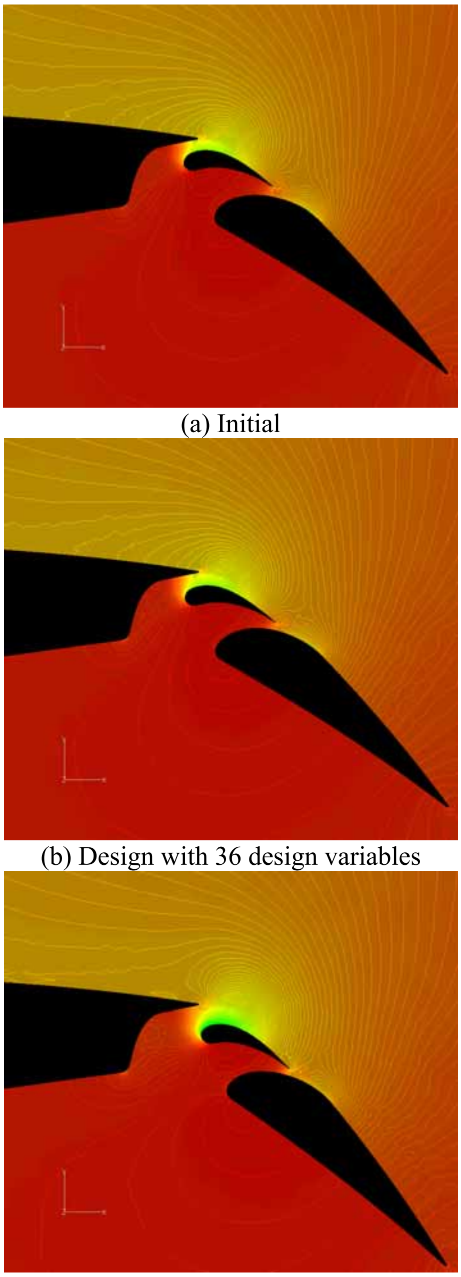


Fig.7 Comparison of surface pressure distributions

4 Concluding Remarks

An aerodynamic design method has been developed using a three-dimensional unstructured Navier-Stokes code and its adjoint code. The adjoint code was developed by a discrete approach so that the Navier-Stokes flow solver and the adjoint code are exactly consistent with each other. The developed design tool was applied to a High-Lift Device design example. Successful design results confirmed validity and efficiency of the present design method.



(a) Initial

(b) Design with 36 design variables

(c) Design with 37 design variables

Fig.8 Pressure contours around vane and flap

Acknowledgement

The authors wish to thank to Takumi Kogure of Tohoku University, for providing the computational mesh for the high lift device.

References

[1] Jameson, A., Aerodynamic Shape Optimization Using the Adjoint Method, *Lecture series*, Von Karman Institute for Fluid Dynamics, Feb. 2003.

[2] Kim, S., Alonso, J.J., Jameson, A., Design Optimization of High-Lift Configurations Using a Viscous Continuous Adjoint Method, AIAA 2002-0844, *40th AIAA Aerospace Science Meeting and Exhibit*, Jan. 2002, Reno, NV.

[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, pp1411-1419.

[5] Anderson, W. K. and Venkatakrishnan, V., Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation, *Computers & Fluids*, Vol.28, No.4-5, 1999, pp443-480.

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

[7] Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1, *AIAA J. of Aircraft*, vol. 36, No.1, 1999, pp51-60.

[8] 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, June 2001, pp.1011-1020.

[9] Kim, H.J., Kim, C., Rho, O.H., and Lee, K. Aerodynamic Sensitivity Analysis for Navier-Stokes Equations, AIAA 99-0402, *37th AIAA Aerospace Science Meeting and Exhibit*, Jan. 1999, Reno, NV.

[10] Spalart, P.R. and Allmaras, S.R., A One-Equation Turbulence Model for Aerodynamic Flows, Recherche Aerospatiale, Vol.1, 1994, pp.5-21.

[11] Obayashi, S., and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier-Stokes Solver", *AIAA J.* Vol.33, No.6, pp.1134-1141, 1995.

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

[13] Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions," AIAA Paper 93-0880, January 1993.

[14] Jameson, A., and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA J.*, Vol.25, pp.929-935, 1987.

[15] Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions," Mathematics of Computation, Vol.37, No.156, pp.385-397, 1981.

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

[17] 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*, John Wiley & Sons, No. 31, pp.97-111, 1999.

[18] Hicks, R. M. and Henne, P. A., Wing Design by Numerical Optimization, *AIAA J. of Aircraft*, Vol.15, No.7, pp. 407-412. 1978.

[19] Murayama, M., Nakahashi, K., and Matsushima, K., Unstructured Dynamic Mesh for Large Movement and Deformation, AIAA 2002-0122, *40th AIAA Aerospace Science Meeting and Exhibit*, Jan. 2002.

[20] Vanderplaats, G. N., *ADS – A Fortran Program For Automated Design Synthesis version 3.00; Users' Guide*, Engineering Design Optimization, INC., 1987.

[21] Vanderplaats, G. N., Numerical Optimization Techniques for Engineering Design: With Applications, McGraw Hill, N.Y., 1984.