AIAA JOURNAL
Vol. 39, No. 6, June 2001

Aerodynamic Optimization of Supersonic Transport Wing Using Unstructured Adjoint Method

Hyoung-Jin Kim,* Daisuke Sasaki,† Shigeru Obayashi,‡ and Kazuhiro Nakahashi§
Tohoku University, Sendai 980-8579, Japan

An aerodynamic design method has been developed by using a three-dimensional Euler code and an adjoint code with a discrete approach. The method is applied to wing design problems of supersonic transports with wing-body-nacelle and wing-body configurations. Hicks–Henne shape functions are adopted for the surface geometry perturbation, and the elliptic equation method is employed for the interior grid modification during the design process. Interior grid sensitivities are neglected except those for design parameters associated with a body translation such as nacelles. The sequential quadratic programming method is used to minimize the drag with constraints on the lift and airfoil thickness. Successful design results confirm validity and efficiency of the present design method.

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 because of its well-developed numerical algorithms and relatively small computational burden. In the application of gradient-based methods to practical aerodynamic design problems, one of the major concerns is an accurate and efficient calculation of sensitivity derivatives of an aerodynamic objective function. The finite difference approximation is the simplest way to calculate the sensitivity information because it does not require any sensitivity code. However, the accuracy of such an approach depends critically on the perturbation size of design variables and the flow initialization.[1]

Sensitivity derivatives can be evaluated more robustly and efficiently by using a sensitivity analysis code based either on a direct method[2–4] or on an adjoint method.[2,5–14] An adjoint method is preferable in aerodynamic designs because it is more economical when the number of design variables is larger than the total number of an objective function and constraints. Reuther et al.,[8] for example, designed aircraft configurations using a continuous adjoint method with the Euler equations in a structured multiblock grid system.

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 reduced. Newman et al.[4] developed a direct sensitivity code via a discrete approach for the two- and three-dimensional Euler equations in the unstructured grid framework and demonstrated design examples of multielement airfoil in a subsonic flow and Boeing 747-200 in a transonic regime. Elliot and Peraire[11] reported a discrete adjoint method for the Euler equations with unstructured grids, which was applied to design a two-dimensional multielement airfoil, a three-dimensional wing, and a wing-body configuration. Recently, Nielson and Anderson[13] developed a discrete adjoint code for the three-dimensional Navier–Stokes equations with a one-equation turbulence model and examined numerical effects on the accuracy of sensitivity derivatives caused by the flux Jacobian simplification and turbulence model differentiation. Mohammadi[14] developed an unstructured adjoint code for the two-/three-dimensional Navier–Stokes equations with a two-equation turbulence model using an automatic differentiation tool with the reverse mode. Extensive previous works on aerodynamic design using structured/unstructured sensitivity analysis methods can be found in a recent review paper by Newman III et al.[9]

In this study direct and adjoint sensitivity codes have been developed from a three-dimensional unstructured Euler solver based on a cell-vertex finite volume method. With the resultant adjoint code designed are supersonic transport (SST) wings with wing-body-nacelle and wing-body configurations. Wing geometry is perturbed in an algebraic manner at five design sections. Interior grids are moved accordingly by the elliptic equation method. Grid sensitivities of interior nodes are neglected except those for design variables associated with nacelle translation in order to reduce required computational time for the mesh sensitivity calculation.

The rest of this paper presents a brief review on the flow solver and the direct and adjoint methods with a discrete approach. Sensitivity code validation is then given, followed by design methodologies including surface mesh deformation and interior mesh movement techniques. Design examples utilizing the resulting design method are then given for SST wings.

Flow Analysis

The Euler equations for compressible inviscid flows are written in an integral form as follows:

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{Q}\,\mathrm{d}V + \int_{\partial\Omega} \boldsymbol{F}(\boldsymbol{Q})\cdot\boldsymbol{n}\,\mathrm{d}S = 0 \tag{1}$$

where $\boldsymbol{Q} = [\rho, \rho u, \rho v, \rho w, e]^T$ is the vector of conservative variables; ρ the density; u, v, w the velocity components in the x, y, z directions; and e the total energy. The vector $\boldsymbol{F}(\boldsymbol{Q})$ represents the inviscid flux vector, and $\boldsymbol{n}$ is the outward normal of $\partial\Omega$, which is the boundary of the control volume Ω. This system of equations is closed by the perfect gas equation of state with a constant ratio of specific heats.

The equations are solved by a finite volume cell-vertex scheme. The control volume is a nonoverlapping dual cell. For a control volume Eq. (1) can be written in an algebraic form as follows:

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

where ΔS_{ij} is a segment area of the control volume boundary associated with edge connecting points i and j. This segment area

Received 25 July 2000; revision received 1 December 2000; accepted for publication 21 December 2000.

*Postdoctoral Research Fellow, Department of Aeronautics and Space Engineering; currently Senior Researcher, Turbomachinery Research Department, Korea Aerospace Research Institute, Taejun 305-600, Republic of Korea. Member AIAA.

†Graduate Research Assistant, Department of Aeronautics and Space Engineering.

‡Associate Professor, Department of Aeronautics and Space Engineering. Senior Member AIAA.

§Professor, Department of Aeronautics and Space Engineering. Associate Fellow AIAA.

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 term $\boldsymbol{h}$ is an inviscid numerical flux vector normal to the control volume boundary, and $\boldsymbol{Q}_{ij}^{\pm}$ are flow variables on both sides of the control volume boundary. The subscript of summation $j(i)$ means all node points are connected to node i.

The numerical flux $\boldsymbol{h}$ is computed using an approximate Riemann solver of Harten–Lax–van Leer–Einfeldt–Wada (HLLEW).[15] The second-order spatial accuracy is realized by a linear reconstruction of the primitive gas dynamic variables $\boldsymbol{q} = [\rho, u, v, w, p]^T$ inside the control volume using the following equation:

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

where $\boldsymbol{r}$ is a vector pointing to point (x, y, z) and i is the node index. The gradients associated with the control volume centroids are volume-averaged gradients computed by the surrounding grid cells. Venkatakrishnan's limiter[16] is used for the function ψ_i in Eq. (3) because of its superior convergence properties.

To integrate Eq. (2) in time, the lower-upper symmetric Gauss–Seidel (LU-SGS) implicit method[17] is adopted. With $\Delta \boldsymbol{Q} = \boldsymbol{Q}^{n+1} - \boldsymbol{Q}^n$ and a linearization of numerical flux term as $\boldsymbol{h}_{ij}^{n+1} = \boldsymbol{h}_{ij}^n + \boldsymbol{A}_i^+ \Delta \boldsymbol{Q}_i + \boldsymbol{A}_j^- \Delta \boldsymbol{Q}_j$, Eq. (2) becomes the following equations:

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

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

$$\boldsymbol{R}_i = -\sum_{j(i)} \Delta S_{ij} \boldsymbol{h}_{ij}^n \tag{5}$$

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 the left-hand side of Eq. (4). The final form of the LU-SGS method for the unstructured grid becomes the following.

Forward sweep:

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

Backward sweep:

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

where $\boldsymbol{D}$ is a diagonal matrix derived by Jameson–Turkel approximation of Jacobian[18] as $\boldsymbol{A}^{\pm} = 0.5(\boldsymbol{A} \pm \rho_A \boldsymbol{I})$, where ρ_A is a spectral radius of Jacobean $\boldsymbol{A}$:

$$\boldsymbol{D} = \left(\frac{V_i}{\Delta t} + 0.5 \sum_{j(i)} \Delta S_{ij} \rho_A\right) \boldsymbol{I} \tag{7}$$

The lower/upper splitting of Eq. (6) for the unstructured grid is realized by using a grid reordering technique[19] to vectorize the LU-SGS method and to improve the convergence.

Sensitivity Analysis

Direct Method

An aerodynamic sensitivity analysis begins with the fact that the discrete residual vector, Eq. (5) of the nonlinear flow equations, is null for a converged flowfield solution of steady problems, which can be written symbolically as

$$R_i[Q, X, \beta] = 0 \tag{8}$$

where $\boldsymbol{X}$ is the grid position vector, β the vector of design variables. Equation (8) can be directly differentiated via the chain rule with respect to β to yield the following equation:

$$\frac{\mathrm{d}R_i}{\mathrm{d}Q} = \left[\frac{\partial R_i}{\partial Q}\right]\left\{\frac{\mathrm{d}Q}{\mathrm{d}\beta}\right\} + \{C_i\} = 0$$

$$\text{where} \quad \{C_i\} = \left[\frac{\partial R_i}{\partial X}\right]\left\{\frac{\mathrm{d}X}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial R_i}{\partial \beta}\right\} \tag{9}$$

This equation is the direct sensitivity equation for the flow variable sensitivity $\{\mathrm{d}Q/\mathrm{d}\beta\}$. The vector $\{C_i\}$ has no relation with the $\{\mathrm{d}Q/\mathrm{d}\beta\}$ and, thus, is constant throutout the solution process of the sensitivity equation for a design variable β. $\{\mathrm{d}X/\mathrm{d}\beta\}$ in the $\{C_i\}$ is a vector of grid sensitivity, which can be calculated by a finite difference approximation or the direct differentiation of a routine for the grid generation or modification.

To find the solution $\{\mathrm{d}Q/\mathrm{d}\beta\}$ of Eq. (9) iteratively, a pseudotime term is added as follows to obtain the incremental form:

$$V_i \frac{\partial \boldsymbol{Q}_i'}{\partial t} = \left[\frac{\partial R_i}{\partial Q}\right]\left\{\frac{\mathrm{d}Q}{\mathrm{d}\beta}\right\}^{n+1} + \{C_i\} \tag{10}$$

where Q' represents the solution vector $\{\mathrm{d}Q/\mathrm{d}\beta\}$. The preceding system of equations is solved with the LU-SGS scheme that is used for the flow solver. By comparing Eqs. (2) and (10), one can obtain a direct sensitivity code by directly differentiating the right-hand side of the flow solver.

The Jacobian matrices $[\partial R/\partial Q]$ and $[\partial R/\partial X]$ in Eq. (9) are very large sparse matrices. Even for a two-dimenisonal grid system, if its banded structure is not considered, the memory requirement can be prohibitively large. To circumvent this problem, Newman et al.[4] adopted an efficeint matrix-vector product method. In the present direct sensitivity analysis, however, the terms $[\partial R/\partial Q]\{\mathrm{d}Q/\mathrm{d}\beta\}$ and $[\partial R/\partial X]\{\mathrm{d}X/\mathrm{d}\beta\}$ in Eq. (9) were calculated without any "matrix-vector product."[10] This could be done by directly differentiating those terms in the residual vector R that are explicit functions of the flow variable Q with respect to β for the $[\partial R/\partial Q]\{\mathrm{d}Q/\mathrm{d}\beta\}$ calculation. The same procedure is applied to $[\partial R/\partial X]\{\mathrm{d}X/\mathrm{d}\beta\}$; those terms in the residual vector R explicitly related with the grid position vector X are differentiated with respect to β.

When the flow variable sensitivity vector $\{\mathrm{d}Q/\mathrm{d}\beta\}$ is obtained, the total derivative of the objective function F can be calculated. The objective function F is usually aerodynamic coefficients such as C_D, C_L, C_M, or differences of surface pressures with specified target pressures. F is a function of flow variables $\boldsymbol{Q}$, grid position $\boldsymbol{X}$, and design variables $\boldsymbol{\beta}$, i.e.,

$$F = F[Q(\beta), X(\beta), \beta] \tag{11}$$

The sensitivity derivative of the cost function F with respect to a design variable β is given by

$$\left\{\frac{\mathrm{d}F}{\mathrm{d}\beta}\right\} = \left\{\frac{\partial F}{\partial Q}\right\}^T \left\{\frac{\mathrm{d}Q}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial F}{\partial X}\right\}^T \left\{\frac{\mathrm{d}X}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial F}{\partial \beta}\right\} \tag{12}$$

Adjoint Method

Because the total derivative of the flow equations in the steady state is null as can be seen in Eq. (9), we can introduce adjoint variables and combine Eqs. (9) and (11) to obtain

$$\left\{\frac{\mathrm{d}F}{\mathrm{d}\beta}\right\} = \left\{\frac{\partial F}{\partial Q}\right\}^T \left\{\frac{\mathrm{d}Q}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial F}{\partial X}\right\}^T \left\{\frac{\mathrm{d}X}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial F}{\partial \beta}\right\}$$

$$+ \{\lambda\}^T \left\{\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{\mathrm{d}Q}{\mathrm{d}\beta}\right\} + \{C\}\right\} \tag{13}$$

If one arranges Eq. (13) for the flow variable sensitivity vector $\{\mathrm{d}Q/\mathrm{d}\beta\}$ and sets coefficients of the sensitivity vector $\{\mathrm{d}Q/\mathrm{d}\beta\}$ equal to zero, one can obtain the following adjoint equation:

$$\{\lambda\}^T \left[\frac{\partial R}{\partial Q}\right] + \left\{\frac{\partial F}{\partial Q}\right\}^T = 0$$

or by transposing

$$\left[\frac{\partial R}{\partial Q}\right]^T \{\lambda\} + \left\{\frac{\partial F}{\partial Q}\right\} = 0 \tag{14}$$

If one finds the adjoint variable vector $\{\lambda\}$, which satisfies the preceding adjoint equation, one can obtain the sensitivity derivative of F with respect to β without any information about the flow variable sensitivity vector $\{\mathrm{d}Q/\mathrm{d}\beta\}$. This makes the computational cost for the sensitivity analysis remarkably reduced for a large number of design variables. Equation (13) eventually becomes the following form:

$$\left\{\frac{\mathrm{d}F}{\mathrm{d}\beta}\right\} = \left\{\frac{\partial F}{\partial X}\right\}^T \left\{\frac{\mathrm{d}X}{\mathrm{d}\beta}\right\} + \left\{\frac{\partial F}{\partial \beta}\right\} + \{\lambda\}^T\{C\} \qquad (15)$$

As Eqs. (2) and (10), the adjoint equation (11) is also converted to the following system of linear algebraic equations with a pseudotime term added and is solved with the LU-SGS scheme:

$$\left(\frac{V_i}{\Delta t}\boldsymbol{I} + \sum_{j(i)} \Delta S_{ij}\boldsymbol{A}_i^{+^T}\right)\Delta\lambda_i - \sum_{j(i)} \Delta S_{ij}\boldsymbol{A}_i^{-^T}\Delta\lambda_j = R\_\mathrm{adj}_i \qquad (16)$$

where R_adj_i is the adjoint residual defined as

$$R\_\mathrm{adj}_i = \left[\frac{\partial R}{\partial Q_i}\right]^T \{\lambda\} + \left\{\frac{\partial F}{\partial Q_i}\right\}$$

Flux Jacobian matrix A^- in the second summation of Eq. (16) is calculated at node i instead of node j and of negative sign. This shows that the wave propagation direction of the adjoint equations is opposite to that of the flow equations. 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 $[\partial R/\partial Q]^T$ in the right-hand side of Eq. (16) is a very large sparse matrix. In the adjoint method, unlike the direct method, all of the elements of the Jacobian matrix should be calculated explicitly. If all of the calculated elements are stored in memory, computational time can be drastically reduced, but the memory requirement would 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.[11] In this study, among the elements of $[\partial R/\partial Q]^T$, 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 differentiated HLLEW flux are recalculated every iterations 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 $\boldsymbol{h}$ 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 $[\partial R_i/\partial Q]^T\{\lambda\}$ is accumulated at all of 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 $R_{12}\,(= -\Delta S_{12}\boldsymbol{h}_{12})$, accumulation of the adjoint residual R_adj is made at nodes related with node 1 as follows:

$$R\_\mathrm{adj}_j <= R\_\mathrm{adj}_j + \left[\frac{\partial R_{12}}{\partial Q_j}\right]^T \lambda_j, \qquad j = 1, 2, 3, 4, 5, 6, \text{ and } 7 \qquad (17)$$

For nodes surrounding node 2,

$$R\_\mathrm{adj}_j <= R\_\mathrm{adj}_j - \left[\frac{\partial R_{12}}{\partial Q_j}\right]^T \lambda_j, \qquad j = 1, 2, 3, 7, 8, 9, \text{ and } 10$$

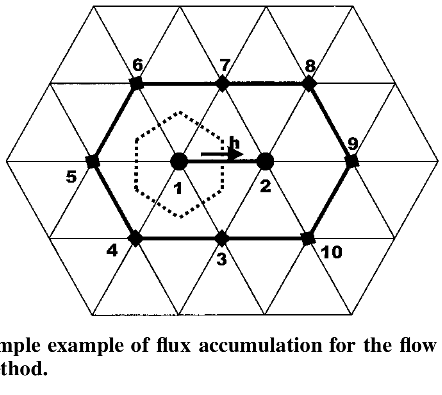


Fig. 1 Simple example of flux accumulation for the flow solver and adjoint method.

This causes small loops for the neighboring nodes to be inserted into the big loop for all edges. The length of the small loop was usually from 5 to 25 around a node point for a three-dimensional grid set for an inviscid computation depending on the grid structure. If the adjoint code is run on a vector machine, it would hamper the flux calculation routine of the adjoint code to be vectorized with the big loop of edges.

To simplify the differentiation process of $[\partial R/\partial Q]^T$, the residual vector R is differentiated by primitive variables $\boldsymbol{q} = [\rho, u, v, w, p]^T$ rather than by the conservative variables Q.[6] Then, the flux Jacobian via the conservative variable can be obtained introducing the transformation matrix $M = \partial Q/\partial \boldsymbol{q}$:

$$\left[\frac{\partial R}{\partial Q}\right]^T = \left(\left[\frac{\partial R}{\partial \boldsymbol{q}}\right]\left[\frac{\partial \boldsymbol{q}}{\partial Q}\right]\right)^T = \left[\frac{\partial \boldsymbol{q}}{\partial Q}\right]^T \left[\frac{\partial R}{\partial \boldsymbol{q}}\right]^T = M^{-1^T}\left[\frac{\partial R}{\partial \boldsymbol{q}}\right]^T \qquad (18)$$

The transformation matrices in a transposed form are given as

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

$$M^{-1^T} = \begin{bmatrix} 1 & -u/\rho & -v/\rho & -w/\rho & [(\gamma-1)(u^2+v^2+w^2)]/2 \\ 0 & 1/\rho & 0 & 0 & -(\gamma-1)u \\ 0 & 0 & 1/\rho & 0 & -(\gamma-1)v \\ 0 & 0 & 0 & 1/\rho & -(\gamma-1)w \\ 0 & 0 & 0 & 0 & (\gamma-1) \end{bmatrix} \qquad (19)$$

In this study the required differentiation process is conducted by human hand. Hand differentiation of a modern CFD code is a tedious job to do. However, if once done carefully, it provides an efficient sensitivity analysis tool.[10]

Boundary Conditions for Discrete Sensitivity Analysis

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:

$$\begin{bmatrix} \left[\dfrac{\partial R^i}{\partial Q^i}\right]^T & \left[\dfrac{\partial R^b}{\partial Q^i}\right]^T \\ \left[\dfrac{\partial R^i}{\partial Q^b}\right]^T & \left[\dfrac{\partial R^b}{\partial Q^b}\right]^T \end{bmatrix} \begin{Bmatrix} \{\lambda^i\} \\ \{\lambda^b\} \end{Bmatrix} + \begin{Bmatrix} \left\{\dfrac{\partial F}{\partial Q^i}\right\} \\ \left\{\dfrac{\partial F}{\partial Q^b}\right\} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ \{0\} \end{Bmatrix} \qquad (20)$$

or

$$\left[\frac{\partial R^i}{\partial Q^i}\right]^T \{\lambda^i\} + \left[\frac{\partial R^b}{\partial Q^i}\right]^T \{\lambda^b\} + \left\{\frac{\partial F}{\partial Q^i}\right\}^T = \{0\} \qquad (21a)$$

$$\left[\frac{\partial R^i}{\partial Q^b}\right]^T \{\lambda^i\} + \left[\frac{\partial R^b}{\partial Q^b}\right]^T \{\lambda^b\} + \left\{\frac{\partial F}{\partial Q^b}\right\}^T = \{0\} \qquad (21b)$$

where the superscript i presents values of inner node and b values of boundary nodes. For example, R^i is the residual at nodes in computational domain, and R^b is the residual of the boundary conditions at boundary nodes. Equation (21a) is solved in an incremental form of Eq. (16). The adjoint variable vector at boundary nodes $\{\lambda^b\}$ is calculated from Eq. (21b) with the adjoint variable vector at the interior nodes $\{\lambda^i\}$ of the earlier time level and the flux Jacobian $[\partial R^i / \partial Q^b]^T$.

An alternative way to impose boundary conditions of the discrete adjoint equations is to treat boundary conditions of a flow solver as an implicit manner. A discrete adjoint code developed from the flow solver with implicit boundary conditions would then automatically satisfy the boundary conditions for the adjoint equations.[4,11]

Sensitivity Code Validation

To validate the direct and adjoint sensitivity codes developed in this study, sensitivity analyses are conducted for a typical SST immersed in a supersonic flow. Flow conditions are $M_\infty = 2.0$ and $\alpha = 2.0$ deg. The number of nodes and cells for the adopted volume grid are about 260,000 and 1,390,000, respectively. All of the computations for the code validation were conducted with a single processor of a NEC SX-4 vector computer.

We used the following wedge-type shape function for the purpose of test:

$$\beta\colon\ y_{\text{new}} = y - \Delta\beta^* x \tag{22}$$

where x and y are coordinates of longitudinal and normal direction, respectively. The sensitivity derivatives are compared with those computed by the forward finite difference approximation with a step size $\Delta\beta$ of 10^{-7}. The residual of the flow solver is reduced by 17 orders from the initial value for the finite difference calculation. Table 1 compares the sensitivity derivatives by the adjoint, direct, and finite difference method. They compare very well with one another with errors less than 0.004%.

Figure 2 shows a comparison of convergence histories of the Euler solver, adjoint and direct sensitivity codes. All of them show similar convergence properties because they all have the same flux Jacobian matrices, and also they adopt the same implicit time-marching algorithm of LU-SGS scheme. The initial values of the sensitivity derivative $\{dQ/d\beta\}$ are obtained by differentiating the initial condition of the flow solver, and the initial adjoint variables $\{\lambda\}$ are set to zero. Table 2 compares required memory and computational time for the Euler solver and its sensitivity codes. The required memory for both direct and adjoint codes seems to be reasonable. The adjoint code costs somewhat large computational time per iteration because of the poor vectorization performance of the adjoint residual accumulation routine as mentioned in the preceding section. We also tested the ratio of computational time of the flow solver over the adjoint code at an Compaq α workstation, a scalar machine, and found that the adjoint code costs only 1.5 times the CPU time of the flow solver per iteration.

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

Derivative	Finite difference	Direct code (% error)	Adjoint code (% error)
$dC_L/d\beta$	1.308065	1.308050 (0.00115)	1.308056 (0.00069)
$dC_D/d\beta$	0.0983594	0.0983587 (0.000712)	0.0983557 (0.00376)

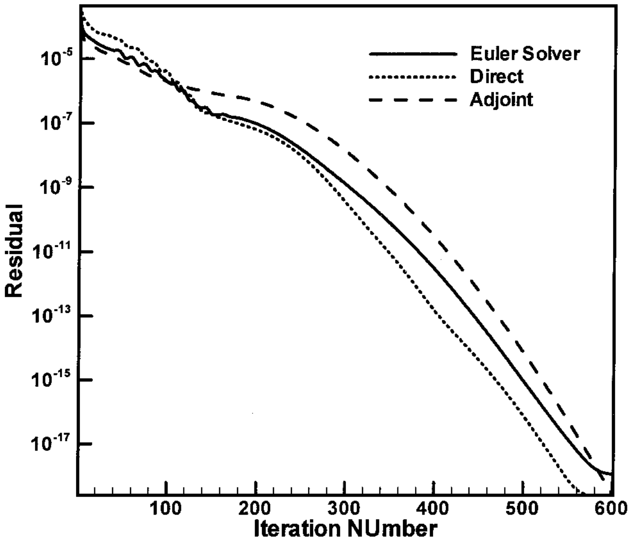


Fig. 2 Convergence history of flow solver and sensitivity codes.

Table 2 Comparison of memory and CPU time

Compared items	Flow solver	Direct code	Adjoint code
Required memory, MB	160	222 (1.39)[a]	360 (2.25)
Time per iteration, s	3.75	5.7 (1.52)	26.5 (7.07)

[a] Figures in parentheses are relative ratios to the flow solver.

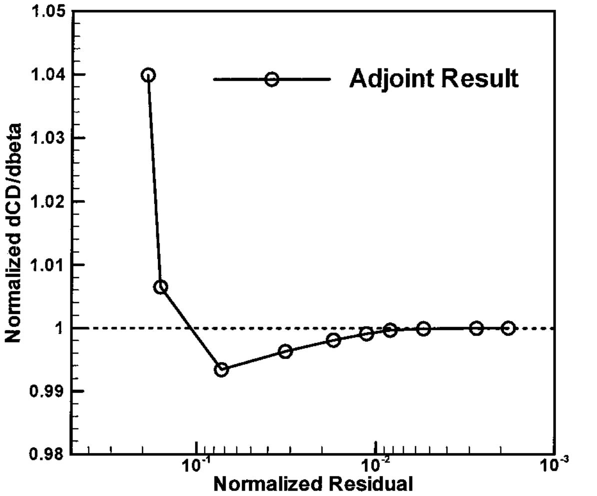


Fig. 3 Convergence trends of sensitivity derivative with respect to residual of the adjoint code.

Figure 3 shows convergence history of the C_D gradient as the adjoint code converges. Even only one-order reduction of the adjoint residual gives accurate gradient value within 1% error for the present design parameter. This is a well-known property of adjoint methods.[8]

Design Methodology

Design Objective

The objective of the present design study is defined as follows:

$$\text{Minimize}\quad C_D$$

$$\text{Subject to}\quad C_L = C_L^* \tag{23}$$

where C_D and C_L are drag and lift coefficients of an aircraft under consideration and C_L^* is specified target lift coefficient. If the lift constraint is dealt as an explicit constraint in an optimizer, it requires an additional adjoint code computation for the C_L derivatives. In this study, therefore, the lift constraint is satisfied running the flow solver in a fixed-lift mode, in which the incidence angle α is adjusted based on $C_{L\alpha}$. The incidence angle is modified every 20 iterations of the LU-SGS time integration after the residual is reduced by two orders of magnitude to obtain a lift coefficient satisfying the following conditions:

$$C_L^* \le C_L \le 1.003 C_L^*$$

Because we would like to minimize drag when $C_L = C_L^*$, i.e., at an adjusted incidence angle, the objective function $F = C_D$ should be modified as follows to consider the lift constraint consistently:

$$F = C_D^* = C_D + \frac{\partial C_D}{\partial \alpha}\Delta\alpha \tag{24}$$

where C_D is a drag coefficient without any incidence angle modification and $\Delta\alpha$ is a required incidence angle variation to match the lift with the target one. A similar relation can be written for the lift:

$$C_L^* = C_L + \frac{\partial C_L}{\partial \alpha}\Delta\alpha \tag{25}$$

where C_L is a lift coefficient without any incidence angle variation and C_L^* is the target lift coefficeint. If we arrange the preceding equation for $\Delta\alpha$ and input to Eq. (24), we obtain a modified objective function

$$F = C_D - \frac{(\partial C_D/\partial\alpha)}{(\partial C_L/\partial\alpha)}\left(C_L - C_L^*\right) \tag{26}$$

where the second term of lift acts as a penalty term, which prevents the design from reducing the drag by simply reducing the lift. The expression for the modified objective function was suggested in a variational form by Reuther et al.[8]

Design Parameters and Grid Modification Method

The wing section geometry is modified adding a linear combination of Hicks and Henne shape functions[20] f_k as follows:

$$y_{\text{new}} = y_{\text{initial}} + \sum_{k=1}^{nv} \beta_k \cdot f_k$$

$$f_k = \sin^3\left[\pi x^{e(k)}\right], \qquad e(k) = \frac{1n(0.5)}{in(x_k)} \tag{27}$$

where β_k are design variables, n_v the number of design variables, and x_k represents the peak location of f_k. Although these Hicks–Henne shape functions are not orthogonal, they have been widely used for aerodynamic design optimization problems with successful results.[3,8]

We used five design sections along an SST wing span and defined 20 Hicks–Henne design variables (coefficients of the shape functions) and one twist angle per a design section resulting in 105 design variables. Figure 4 shows 10 Hicks–Henne functions used for upper and lower surface perturbation. For the case with an engine nacelle, the height of diverter leading edge is also considered as a design parameter in addition to the 105 design variables. If the diverter height is varied during the design process, the nacelle is moved vertically as a solid body.

With the new geometry of design sections, vertical coordinates of wing surface node points are linearly interpolated. When the surface grid is modified, the interior grid points should 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 a more sophisticated grid movement method is needed.

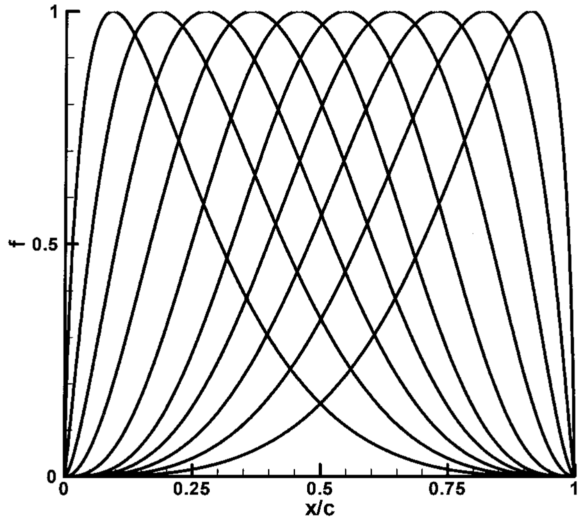


Fig. 4 Ten Hicks–Henne shape functions used in this study: SST wing-body-nacelle configuration.

For the movement of the grid points with the perturbed surface grid, we used the elliptic partial differential equation method proposed by Crumpton and Giles.[21] In the method the displacement $\delta\boldsymbol{x}$ from initial grid point $\boldsymbol{x}_0$ is prescribed by the following equation with Dirichlet boundary conditions:

$$\nabla\cdot(k\nabla\delta x) = 0 \tag{28}$$

Diffusion coefficient k is constant in each cell and is given by

$$k = 1/\max(\text{Vol}, \varepsilon) \tag{29}$$

where Vol is a control volume of each grid point and ε is a small positive number to prevent $\boldsymbol{k}$ from becoming negative. In the original form of the method in Ref. 21, Vol is obtained from deformed grid system $\boldsymbol{x}_0 + \delta\boldsymbol{x}$. In this study, however, the cell volume is calculated from initial grid $\boldsymbol{x}_0$ with an assumption that the cell volumes (or at least their relative ratios) do not change much through one iteration of the optimization process, which is often the case for aircraft wing section design problems. With this assumption the nonlinear elliptic equation becomes a linear one, which is much simpler and thus can be solved with a less computational time because the control volumes of the grid points do not need to be calculated during the iteration step. Although this caused no problem in the present design study, it might need to consider the original nonlinear equation for a robust grid modification if the geometry changes much throughout the design process. The elliptic equation (28) is discretized by a finite volume method, and subsequent linear algebraic equations are solved by the conjugate gradient method.[22] Required computational time to obtain converged solution $\delta\boldsymbol{x}$ was the same with that of a few iterations of the Euler solver.

Grid Sensitivity

The elliptic equation method for the interior grid movement is differentiated to be applied to the grid sensitivity calculation for the vector $\{C\}$ in Eq. (9) with respect to each geometric design variable. Because this requires almost the same computational cost with the grid movement procedure, the total computational burden would be a substantial amount if the number of design variables becomes large, say, more than 100.

One possible way to reduce the computational burden of the grid sensitivity calculation is to neglect the grid sensitivity of interior node points. In this study we made a comparison between the derivatives with and without the interior grid sensitivities in order to evaluate the accuracy of the simplification approach ignoring the interior grid movement. As a testbed for this comparison, we used an SST wing-body-nacelle configuration, which is an initial geometry for one of the design examples of this paper. Figure 5 shows surface shape and grids of the configuration. A flow-through type engine nacelle is attached to the lower surface of the wing via a diverter. The number of nodes and cells for the adopted volume grid are about 270,000 and 1,500,000, respectively.

Figure 6 compares the derivatives of the objective function obtained with and without the interior grid sensitivity information for the configuration. Derivatives with respect to the design variables show little difference between the two values except for design variables having indices of 21–30. The design variables with indices from 21–30 are defined on the lower surface of the second design section, which is located on the centerline of the diverter. Thus, they cause the nacelle to be translated vertically because the leading-edge height of the diverter is controlled only by the design variable for the height. The interior grid sensitivities are required for design variables associated with translation of the nacelle, and, on the other hand, the grid sensitivities can be ignored for other ordinary design variables, i.e., coefficients of shape function or twist angles, with little accuracy degradation.

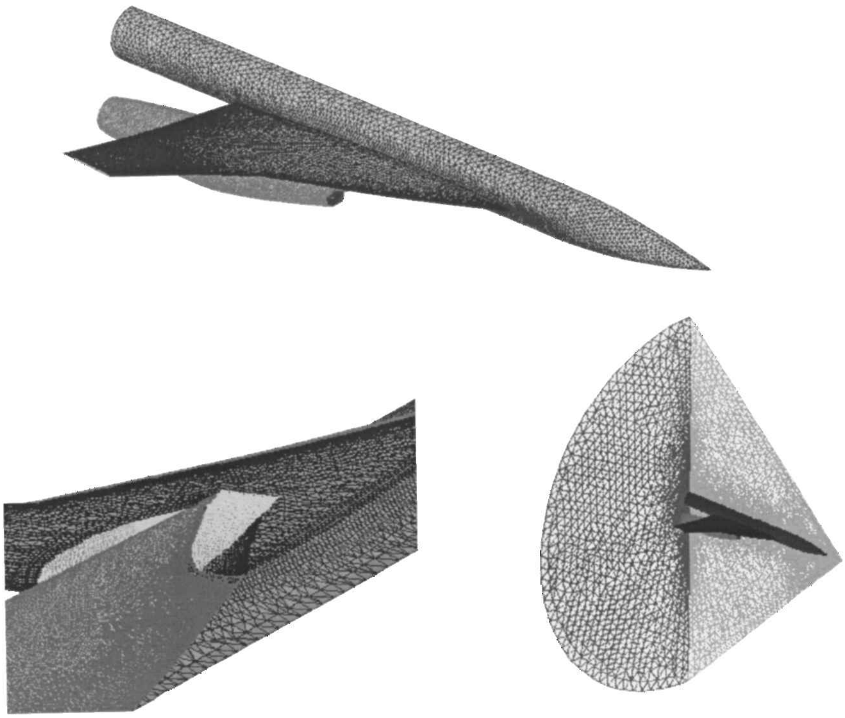


Fig. 5 Surface grids of National Aerospace Laboratory, Japan, experimental supersonic aircraft with nacelles.

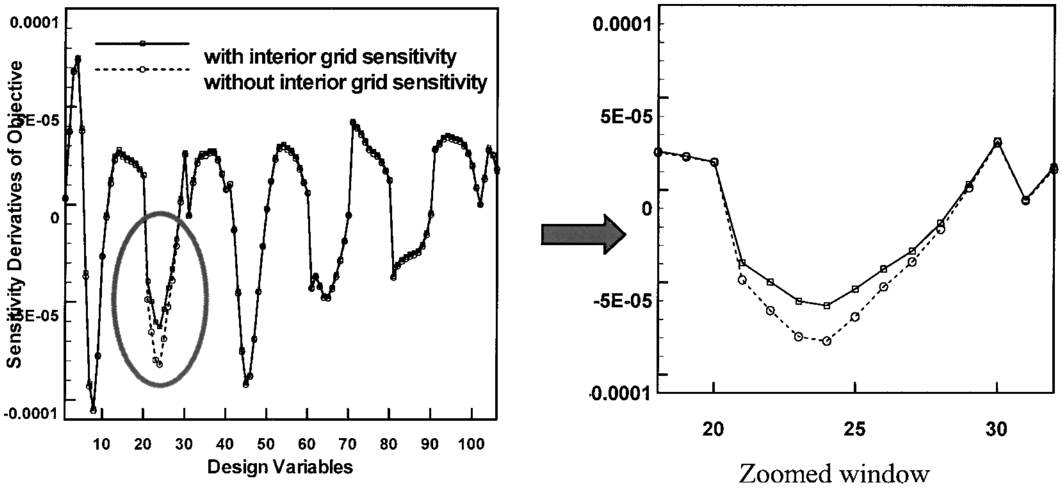


Fig. 6 Comparison of sensitivity derivatives with and without interior grid sensitivity information for SST wing-body-nacelle configuration.

This result agrees well with Ref. 7 in which it was shown that for geometry with singularity such as sharp trailing edges interior mesh sensitivities must be included for the calculation of the derivatives associated with translation. In the present case the nacelle inlet and outlet have sharp edges, which seems to cause the derivatives calculated without interior mesh sensitivities to be deviated from those values obtained with the mesh sensitivities. Also, Eyi and Lee[3] defined grid sensitivities on the body surface only by ignoring the movement of interior grid points in their study on direct sensitivity analysis with two-dimensional Euler equations. Although they did not present an explicit accuracy comparison, they reported that the simplification approach does not affect the accuracy of the resulting sensitivity.

Recently, on the other hand, Anderson and Bonhaus[12] 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 the reference 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. Reminding the fact that the present study deals with an Euler solver with a relatively coarse grid near-wall boundary compared to those for viscous computations, this disagreement seems to be caused by the effects of the near-wall fine grid, viscosity, or turbulence model considered in the reference. This issue was discussed well by Nielson and Anderson.[13]

In this study interior grid sensitivities for the 10 design variables (21–30) of the case with nacelle are calculated by the elliptic equation method, whereas for other design variables only the surface grid sensitivities are defined. This simplification approach reduced the computational time for the vector $\{C\}$ in Eq. (15) by about 75% compared with the approach considering all of the interior grid sensitivities.

Optimization Method

For the minimization of the objective function with specified constraints, the ADS program[23] was used as an optimizer. The sequential quadratic programming (SQP) method[24] is adopted in

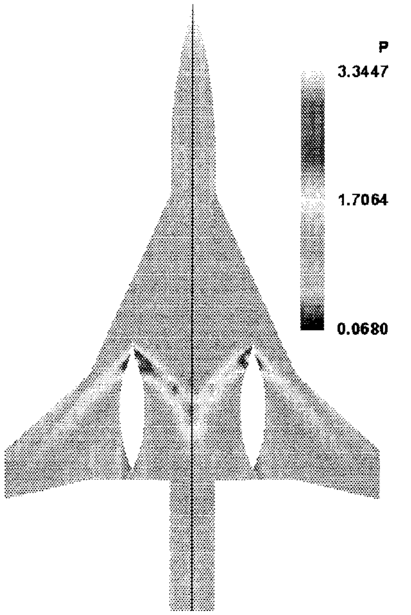


Fig. 7 Lower surface contours of nondimensional pressure of initial and design configuration: left-hand side = initial, and right-hand side = design.

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. The 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 formula. Detailed algorithms and methodologies of the SQP method are described in Ref. 24.

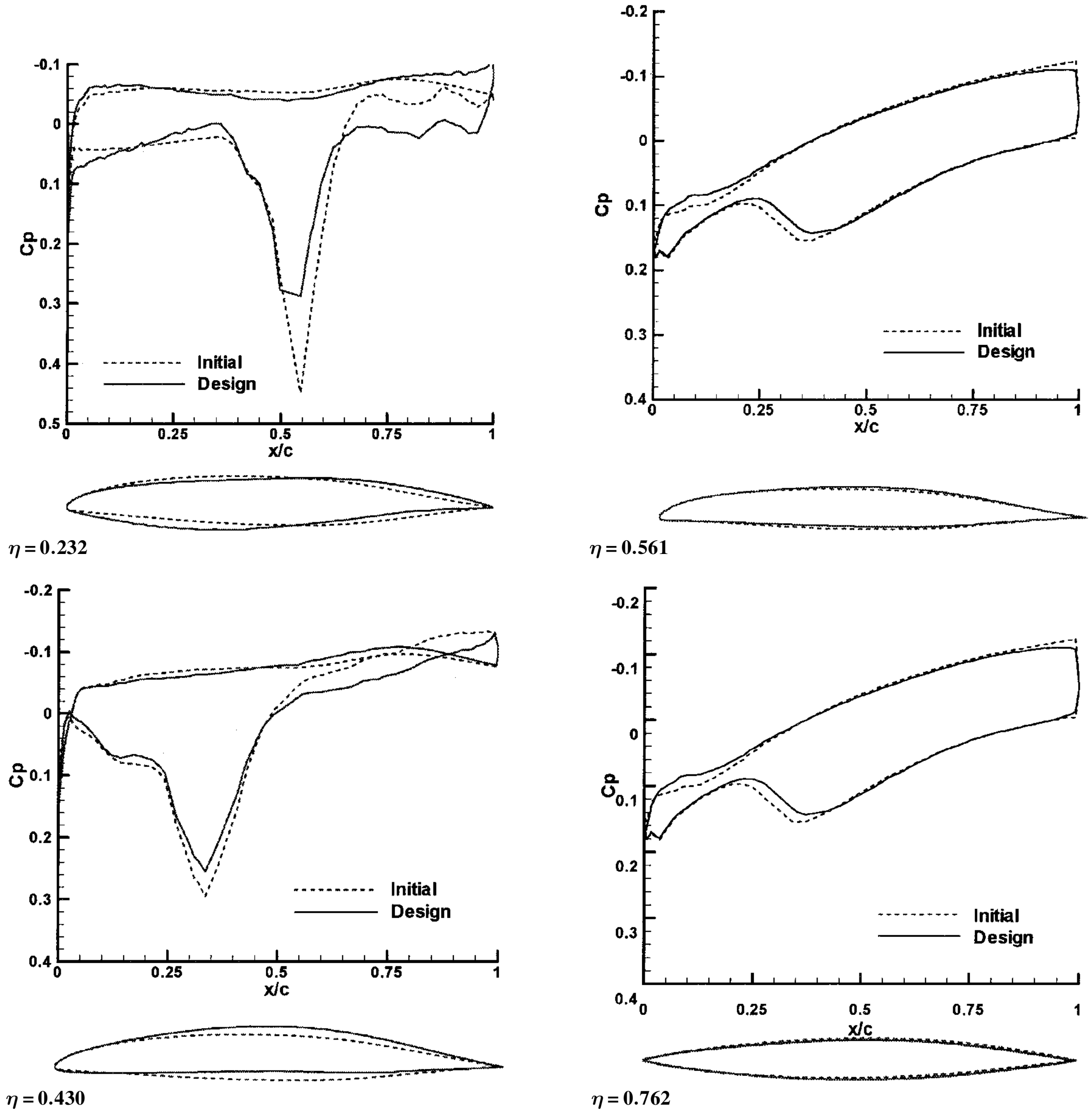
Design Results

The present design method using the unstructured Euler solver and the adjoint code is applied to two experimental supersonic transports. One is wing-body-nacelle and the other wing-body configuration, both of which are under development by National Aerospace Laboratory of Japan as basic studies for the next generation supersonic transport.[25]

Design I: Wing-Body-Nacelle Configuration

Surface shape and grids of the configuration are shown in Fig. 5 Design conditions are a freestream Mach number of 2.0 and C_L of 0.100. The number of nodes and cells for the adopted volume grid are about 270,000 and 1,500,000, respectively.

As mentioned earlier in the section on design parameters, the total number of design variables is 106 for this design example.



$\eta = 0.232$

$\eta = 0.561$

$\eta = 0.430$

$\eta = 0.762$

Fig. 8 Comparison of wing section shapes and pressure distributions; SST wing-body-nacelle configuration.

Constraints are imposed so that wing section thickness values at front (5% chord), rear (80% chord) spar position and maximum thickness position (50% chord) should be larger than those of the initial geometry. In addition, the diverter leading-edge height is also constrained to be larger than the initial value. This lower side constraint is to prevent boundary-layer flow from being entrained into the engine nacelle, which might occur if the height of the diverter leading edge becomes smaller than the initial value.

The SQP optimization was run for five iterations in order to minimize the modified objective function of Eq. (26) with the geometric constraints. However, no further performance improvement was made after design iteration three. As can be noted in Table 3, the drag coefficient was reduced by 16 counts from 0.0205 to 0.0189 retaining the lift coefficient as the specified value. The thickness constraints were also satisfied by the optimizer.

Figure 7 shows the surface pressure contours on the wing lower surface. The strength of the impinging shock wave on the wing lower surface generated by the diverter leading edge is greatly reduced through the design procedure. The strength of the expansion wave at the trailing edge of the diverter has also been remarkably reduced. Figure 8 compares wing section shapes and pressure distributions at design sections. The wing section shapes are enlarged by a factor of three in the normal direction. Section pressure distributions also show that the shock strength on the lower surface has been remarkably reduced.

The leading-edge height of the diverter remained the same as the initial value because the gradient of the objective function with respect to the height is positive throughout the design iterations. This is quite natural in a sense that the diverter height increment will increase the aircraft volume and also the pressure drag accordingly.

Because the present design study is based on the Euler equations, the estimated amount of the pressure drag reduction might be deviated from the realistic value, especially for this kind of case with strong shock/boundary-layer interaction. To consider the viscous effects in the design process, employment of a Navier–Stokes solver and its adjoint code is necessary.

Design II: Wing-Body Configuration

Figure 9 shows the initial wing-body geometry for the second design example. Design conditions are the same with design example I; freestream Mach number of 2.0 and C_L of 0.100. The initial wing geometry has been designed by an inverse design method with the natural laminar flow (NLF) concept, and possesses very good aerodynamic performance at the design condition.[25] The number of nodes and cells for the adopted volume grid are about 260,000 and 1,390,000, respectively. In the present optimization the same design variables are employed as the design example I except the diverter leading-edge height. Therefore, totally 105 design variables are used for the wing section shape modification and twist angles variation. The same thickness constraints are also imposed as in example I.

Table 3 Design results for SST wing-body-nacelle configuration

Aerodynamic performance	Initial	Design	Δ, %
C_L	0.10017	0.10020	+0.03
C_D	0.020513	0.018918	+7.78
L/D	4.883	5.297	+8.48

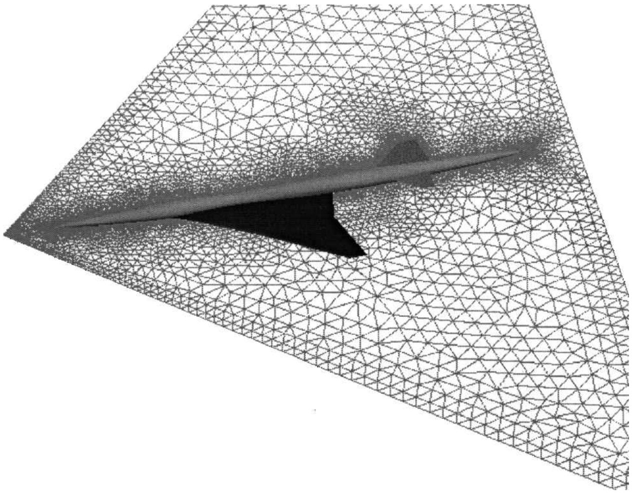


Fig. 9 Surface geometry of SST wing-body configuration and grids on symmetric plane.

Table 4 Design results for SST wing-body configuration

Aerodynamic performance	Initial	Design	Δ, %
C_L	0.10002	0.09993	−0.088
C_D	0.006349	0.006242	1.675
L/D	15.75	16.01	1.614

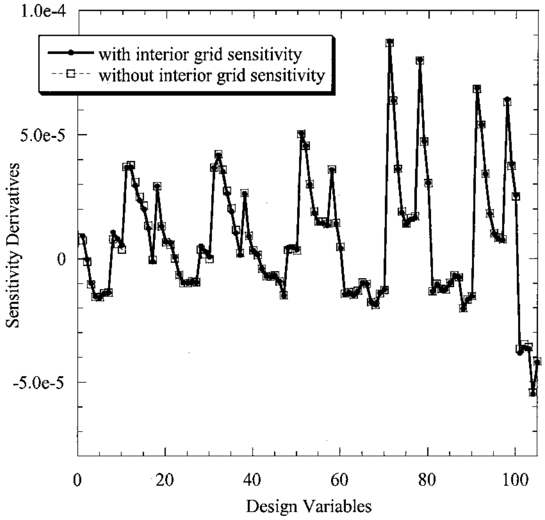
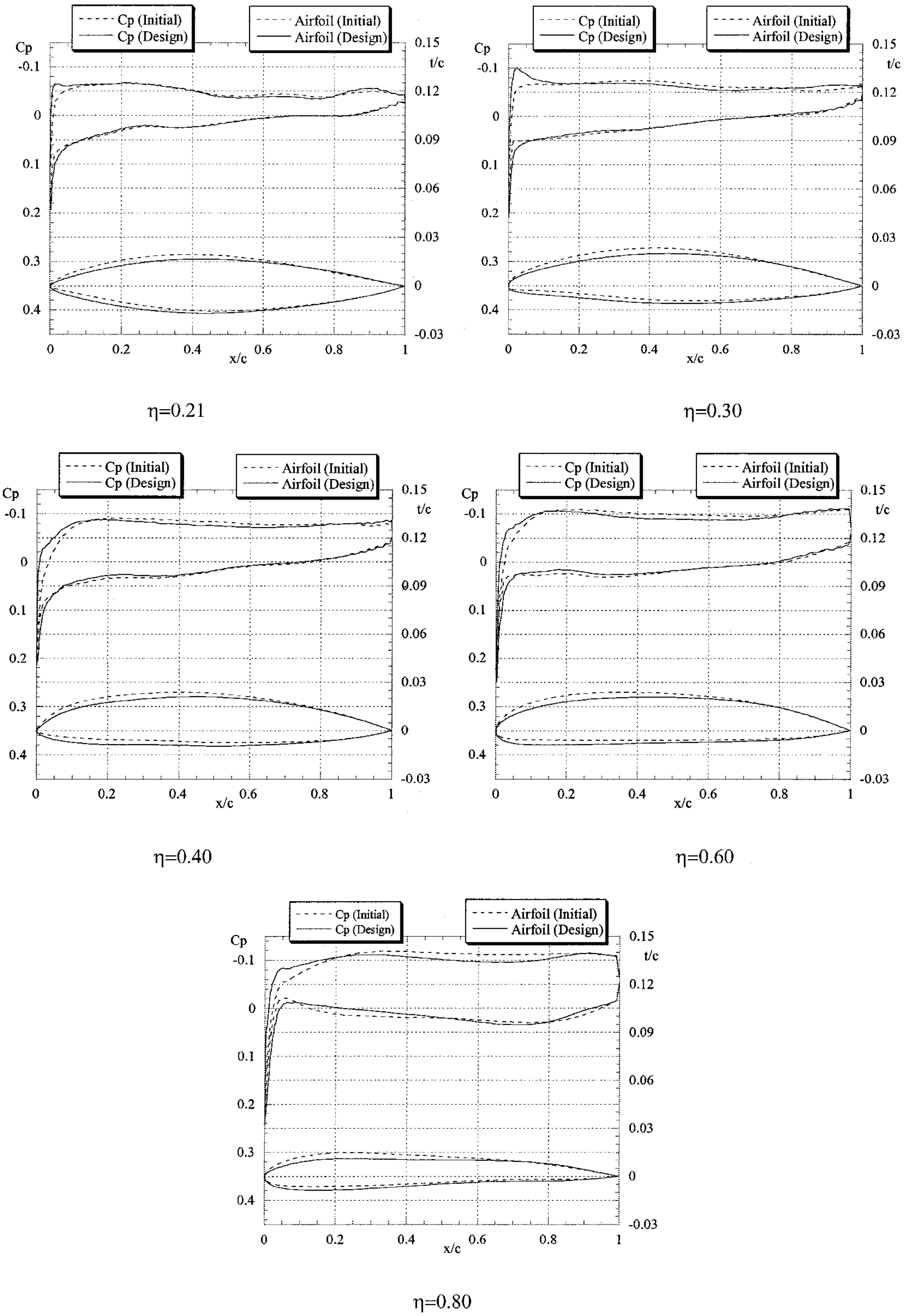


Fig. 10 Comparison of sensitivity derivatives with and without interior grid sensitivity information for SST wing-body configuration.

Before the design process is conducted, a comparison is made between the derivatives with and without the interior grid sensitivities in order to evaluate the accuracy of the simplification approach ignoring the interior grid movement. Figure 10 compares the derivatives of the objective function obtained with and without the interior grid sensitivity information for the configuration. Derivatives with respect to the design variables agree well one another. The grid sensitivities are, therefore, ignored for all design variables adopted in this case.

Table 4 summarizes the design results. The SQP optimizer was run for 15 iterations to obtain a drag coefficient reduced by only one count from 0.006349 to 0.006242 retaining the lift coefficient as the specified value and satisfying imposed thickness constraints.

Figure 11 compares wing section shapes and pressure distributions at wing sections. Section pressure distributions show that the suction peak at the leading edge has been increased by the design with other features of pressure distributions being almost the same although the section shapes have been changed remarkably. This implies that the initial shape is already near optimum design; thus, further performance improvement is hard to be made from the initial one. Furthermore, the NLF concept requires the surface pressure not to increase along streamlines so that the boundary-layer instability does not grow up. The increased suction peaks of the design wing, therefore, do not meet the NLF concept that was employed as a design philosophy for the initial wing. To consider the design requirement for the NLF, a Navier–Stokes-based design should be employed.



η=0.21

η=0.30

η=0.40

η=0.60

η=0.80

Fig. 11 Comparison of wing section shapes and pressure distributions; SST wing-body configuration.

Conclusions

An aerodynamic design optimization system is developed using the unstructured Euler solver and the discrete adjoint method. The interior grid position movement is made by the elliptic equation method. For an efficient calculation of terms related with the grid sensitivities, grid sensitivities of interior node points are ignored except those for the design variables associated with nacelle translation. The present method is successfully applied to design the SST wing-body-nacelle and wing-body configurations. For the wing-body-nacelle configuration the impinging shock wave from the diverter on the wing lower surface has been greatly reduced, and as a consequence, drag is remarkably reduced by five iterations of the SQP optimizer. On the other hand, the initial shape of the wing-body configuration was near optimum itself, and thus only one count drag reduction was made by 15 design iterations. Through the design examples the present design optimization tool including the adjoint method was shown to be an efficient and reliable design tool for complex aerodynamic vehicles.

Acknowledgments

The first author was supported by the Japan Society for Promotion of Science. The authors wish to thank to Y. Itoh, a graduate student of Tohoku University, for his help to generate the surface grid. The authors also thank T. Iwamiya, National Aerospace Laboratory (NAL), for providing the surface geometries of NAL experimental supersonic transports.

References

[1] Eyi, S., and Lee, K. D., "Effect of Sensitivity Calculation on Navier–Stokes Design Optimization," AIAA Paper 94-0060, Jan. 1994.

[2] Sherman, L. L., Taylor, A. C., III, Green, L. L., Newman, P. A., Hou, G. J., and Korivi, V. M., "First- and Second-Order Aerodynamic Sensitivity Derivatives via Automatic Differentiation with Incremental Iterative Methods," AIAA Paper 94-4262, Sept. 1994.

[3] Eyi, S., and Lee, K. D., "Effects of Sensitivity Derivatives on Aerodynamic Design Optimization," *Inverse Problems in Engineering Journal*, Vol. 3, No. 2, 1996, pp. 213–225.

[4] Newman, J. C., III, Taylor, A. C., III, and Barnwell, R. W., "Aerodynamic Shape Sensitivity Analysis and Design Optimization of Complex Configuration Using Unstructured Grids," AIAA Paper 97-2275, Jan. 1997.

[5] Sung, C.-H., and Kwon, J. H., "Accurate Aerodynamic Sensitivity Analysis Using Adjoint Equations," *AIAA Journal*, Vol. 38, No. 2, 2000, pp. 243–250.

[6] Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamic Design Using the Navier–Stokes Equations," AIAA Paper 97-0101, Jan. 1997.

[7] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 97-0643, Jan. 1997.

[8] 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," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[9] Newman, J. C., III, Taylor, A. C., III, Barnwell, R. W., Newman, P. A., and Hou, G. J.-W., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96.

[10] Kim, H. J., Kim, C., Rho, O. H., and Lee, K., "Aerodynamic Sensitivity Analysis for Navier–Stokes Equations," AIAA Paper 99-0402, Jan. 1999.

[11] Elliot, J., and Peraire, J., "Aerodynamic Optimization on Unstructured Meshes with Viscous Effects," AIAA Paper 97-1849, June 1997.

[12] Anderson, W. K., and Bonhaus, D. L., "Airfoil Design on Unstructured Grids for Turbulent Flows," *AIAA Journal*, Vol. 37, No. 2, 1999, pp. 185–1191.

[13] Nielson, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations," *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.

[14] Mohammadi, B., "Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence," AIAA Paper 97-0099, Jan. 1997.

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

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

[17] Jameson, A., and Yoon, S., "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," *AIAA Journal*, Vol. 25, No. 7, 1987, pp. 929–935.

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

[19] 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.

[20] Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[21] Crumpton, P. I., and Giles, M. B., "Implicit Time Accurate Solutions on Unstructured Dynamic Grids," AIAA Paper 95-1671, June 1995.

[22] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in Fortran*, 2nd ed., Cambridge Univ. Press, Cambridge, England, U.K., 1992, pp. 77–82.

[23] Vanderplaats, G. N., *ADS—A Fortran Program for Automated Design Synthesis Version 3.00; Users' Guide*, Engineering Design Optimization, Inc., Santa Barbara, CA, 1987.

[24] Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design: with Applications*, McGraw–Hill, New York, 1984, pp. 195–199.

[25] Iwamiya, T., "NAL SST Project and Aerodynamic Design of Experimental Aircraft," *Proceedings of Computational Fluid Dynamics '98*, ECCOMAS 98, Vol. 2, Wiley, New York, 1998, pp. 580–585.

E. Livne
Associate Editor