AERODYNAMIC OPTIMIZATION OF SUPERSONIC WING-NACELLE CONFIGURATION USING AN ADJOINT METHOD WITH THE UNSTRUCTURED-GRID APPROACH

Hyoung-Jin Kim*, Shigeru Obayashi†, and Kazuhiro Nakahashi‡
Tohoku University, Sendai, 980-8579 JAPAN

ABSTRACT

An aerodynamic design method has been developed by using a three-dimensional unstructured Euler code and an adjoint code with a discrete approach. The resulting adjoint code is applied to a wing design problem of supersonic transport with a wing-body-nacelle configuration. 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 nacelle translation. 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 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 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 since 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 are larger than the total number of an objective function and constraints. Reuther et al.[8,9], for example, designed aircraft configurations using a continuous adjoint method with the Euler equations in a structured multi-block grid system. Kim et al.[10] developed direct and adjoint sensitivity codes from 2-D Navier-Stokes code with an algebraic turbulence model in a structured 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 saved. Newman et al.[4] developed a direct sensitivity code via a discrete approach for the 2-D and 3-D Euler equations in the unstructured grid framework, and demonstrated design examples of multi-element 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 2-D multi-element airfoil, a 3-D wing, and a wing-body configuration. Recently, Nielson and Anderson [13] developed a discrete adjoint code for the 3-D Navier-Stokes equations with a one-equation turbulence model, and examined numerical effects on the accuracy of sensitivity derivatives due to the flux jacobian simplification and turbulence model differentiation. Mohammadi[14] developed an unstructured adjoint code for the 2D/3-D Navier-Stokes equations with a two-equation turbulence model using an automatic differentiation tool with the reverse mode.

In this study, direct and adjoint sensitivity codes have been developed from a 3-D unstructured Euler solver based on a cell-vertex finite volume method. With the resulting adjoint code, aerodynamic design of a Supersonic Transport (SST) wing with nacelle is conducted. 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. A design example utilizing the resulting design method is finally given for a supersonic transport (SST) wing in the wing-body-nacelle configuration.

[*] Postdoctoral Research Fellow, Dep't of Aeronautics and Space Eng.
[†] Associate Professor, Dep't of Aeronautics and Space Eng.
[‡] Professor, Dep't of Aeronautics and Space Eng.

Flow Analysis

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

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV+\int_{\partial\Omega}\mathbf{F}(\mathbf{Q})\cdot\mathbf{n}dS=0 \tag{1}$$

where $\mathbf{Q}=[\rho,\rho u,\rho v,\rho w,e]^T$ 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 $\mathbf{F(Q)}$ represents the inviscid flux vector and $\mathbf{n}$ 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 non-overlapping dual cell. For a control volume, Eq.(1) can be written in an algebraic form as follows;

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

where ΔS_{ij} is a segment area of the control volume boundary associated with edge connecting points *i* and *j*. This segment area 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 term $\mathbf{h}$ is an inviscid numerical flux vector normal to the control volume boundary, and $\mathbf{Q}_{ij}^{\pm}$ are flow variables on both sides of the control volume boundary. The subscript of summation, *j(i)*, means all node points connected to node *i*.

The numerical flux $\mathbf{h}$ 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 $\mathbf{q}=[\rho,u,v,w,p]^{\mathrm{T}}$ inside the control volume using the following equation;

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

where $\mathbf{r}$ 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.

In order to integrate Eq. (2) in time, the Lower-Upper Symmetric Gauss-Seidel(LU-SGS) implicit method [17] is adopted. With $\Delta\mathbf{Q}=\mathbf{Q}^{n+1}-\mathbf{Q}^{n}$ and a linearization of numerical flux term as $\mathbf{h}_{ij}^{n+1}=\mathbf{h}_{ij}^{n}+\mathbf{A}_i^{+}\Delta\mathbf{Q}_i+\mathbf{A}_j^{-}\Delta\mathbf{Q}_j$, Eq.(2) becomes the following equations.

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

where R is a residual vector;

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

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

Forward sweep:

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

Backward sweep:

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

where $\mathbf{D}$ is a diagonal matrix derived by Yoon and Jameson[17] with Jameson-Turkel approximation of Jacobian[18] as $\mathbf{A}^{\pm}=0.5\left(\mathbf{A}\pm\rho_A\mathbf{I}\right)$, where ρ_A is a spectral radius of Jacobean $\mathbf{A}$.

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

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 flow field solution of steady problems, which can be written symbolically as

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

where $\mathbf{X}$ 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.

$$\frac{dR_i}{dQ}=\left[\frac{\partial R_i}{\partial Q}\right]\left\{\frac{dQ}{d\beta}\right\}+\{C_i\}=0, \tag{9}$$

where $\{C_i\}=\left[\frac{\partial R_i}{\partial X}\right]\left\{\frac{dX}{d\beta}\right\}+\left\{\frac{\partial R_i}{\partial\beta}\right\}$.

This equation is the direct sensitivity equation for the flow variable sensitivity *{dQ/dβ}*. The vector $\{C_i\}$ has no relation with the *{dQ/dβ}*, and thus, is constant throuthout the solution process of the sensitivity equation for a design variable *β*. *{dX/dβ}* 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.

In order to find the solution *{dQ/dβ}* of Eq.(9) iteratively, a pseudo time term is added as follows to obtain the incremental form;

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

where Q′ represents the solution vector *{dQ/dβ}*. The above system of equations is solved with the LU-SGS scheme that is used for the flow solver. By comparing Eqs.(2) and (10), it is noted that one can obtain a direct sensitivity code by directly differentiating the right-hand side of the flow solver.

The Jacobian matrices [∂R/∂Q] and [∂R/∂X] in Eq.(9) are very large banded matrices. Even for a two-dimenisonal grid system, if its banded structure is not considered, the memory requirement easily exceeds sevearal Gbytes. In order to circumvent this problem Newman et al.[4] adopted an efficeint matrix-vector product method. In the present direct sensitivity analysis, however, the terms [∂R/∂Q] *{dQ/dβ}* and [∂R/∂X] *{dX/dβ}* in Eq.(9) was 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 [∂R/∂Q] *{dQ/dβ}* calculation. The same procedure is applied to [∂R/∂X] *{dX/dβ}*; those terms in the residual vector R explicitly related with the grid position vector X are differntiated with respect to β.

When the flow variable sensitivity vector *{dQ/dβ}* 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 **Q**, grid position **X**, and design variables **β,** i.e,

$$F = F(Q(\beta), X(\beta), \beta)\,. \qquad (11)$$

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

$$\left\{\frac{dF}{d\beta}\right\} = \left\{\frac{\partial F}{\partial Q}\right\}^T \left\{\frac{dQ}{d\beta}\right\} + \left\{\frac{\partial F}{\partial X}\right\}^T \left\{\frac{dX}{d\beta}\right\} + \left\{\frac{\partial F}{\partial \beta}\right\}, \qquad (12)$$

Adjoint Method

Since 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

$$\left\{\frac{dF}{d\beta}\right\} = \left\{\frac{\partial F}{\partial Q}\right\}^T \left\{\frac{dQ}{d\beta}\right\} + \left\{\frac{\partial F}{\partial X}\right\}^T \left\{\frac{dX}{d\beta}\right\} + \left\{\frac{\partial F}{\partial \beta}\right\}$$
$$+ \{\lambda\}^T \left\{\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ}{d\beta}\right\} + \{C\}\right\}. \qquad (13)$$

Coefficients of the flow variable sensitivity vector *{dQ/dβ}* form the following adjoint equation.

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

If one finds the adjoint variable vector {λ} which satisfies the above adjoint equation, one can obtain the sensitivity derivative of F with respect to β without any information about the flow variable sensitivity vector *{dQ/dβ}*. This makes the computational cost for the sensitivity analysis independent of the number of design variables. Eq.(13) eventually becomes to the following form,

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

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

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

where R_adj_i is the adjoint residual defined as

$$R\_adj_i = \left[\frac{\partial R}{\partial Q_i}\right]^T \{\lambda\} + \left\{\frac{\partial F}{\partial Q_i}\right\}.$$

Flux Jacobian matrix A^- in the second summation is calculated at node i instead of node j and of negative sign. This shows that 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 RHS of Eq.(16) is a very large banded matrix. In the adjoint method, unlike the direct method, all 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 $\psi_i \nabla \mathbf{q}_i$, 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 **h** through the control volume surface is calculated and accumulated 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 $\left[\frac{\partial R_i}{\partial Q}\right]^T \{\lambda\}$ 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} ($= -\Delta S_{12}\mathbf{h}_{12}$), accumulation of the adjoint residual R_adj is made at nodes related with node 1 as follows.

$$\mathrm{R\_adj}_j \Leftarrow \mathrm{R\_adj}_j + \left[\frac{\partial R_{12}}{\partial Q_j}\right]^T \lambda_j, \mathrm{j} = 1, 2, 3, .., 7 \qquad (17)$$

For nodes surrounding node 2,

$$\mathrm{R\_adj}_j \Leftarrow \mathrm{R\_adj}_j - \left[\frac{\partial R_{12}}{\partial Q_j}\right]^T \lambda_j, \mathrm{j} = 1, 2, 3,7,8,9,10.$$

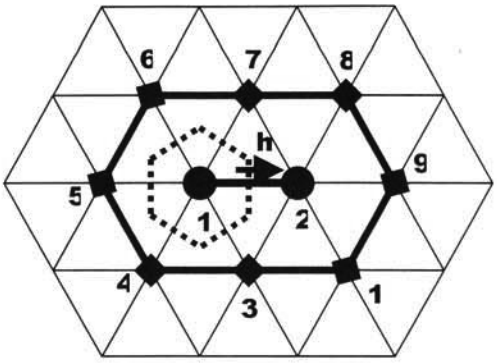


Fig.1 A 2-D example of flux accumulation for the flow solver and the adjoint method

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 Euler grid 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.

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

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

The transformation matrices in a transposed form are given as

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

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

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

Boundary Conditions for the 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.

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

or

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

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

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 previous 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

In order to validate the direct and adjoint sensitivity codes developed in this study, sensitivity analyses are conducted for a typical Supersonic Transport (SST) immersed in a supersonic flow. Flow conditions are M_∞ = 2.0 and α = 2.0 degree. All the computations for the code validation were conducted with a single processor of a NEC SX-4 vector computer.

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

$$\beta : \quad y_{new} = y - \Delta\beta * x, \quad (22)$$

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 Δβ of 10^{-7}. The residual of the flow solver is reduced to nearly machine zero 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

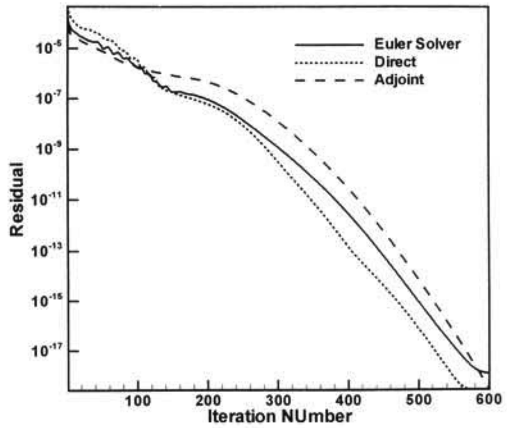


Fig.2 Convergence histories of flow solver and sensitivity codes

since 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/dβ}* are obtained by differentiating those of the flow solver, and the initial adjoint variables {λ} 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 due to the poor vectorization performance of the adjoint residual accumulation routine as mentioned in the previous section. We also tested the ratio of computational time of the flow solver over the adjoint code at a 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.

Figure 3 shows convergence history of the C_D gradient as the adjoint code converges. It should be noted that even only one-order reduction of the adjoint residual gives accurate gradient value within 1 % error for the present design parameter.

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

	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)

Table 2 Comparison of memory and CPU time; numbers in the parentheses are relative ratios to the flow solver

	Flow Solver	Direct code	Adjoint code
Required Memory(MB)	160	222 (1.39)	360 (2.25)
Time per Iteration (sec.)	3.75	5.7 (1.52)	26.5 (7.07)

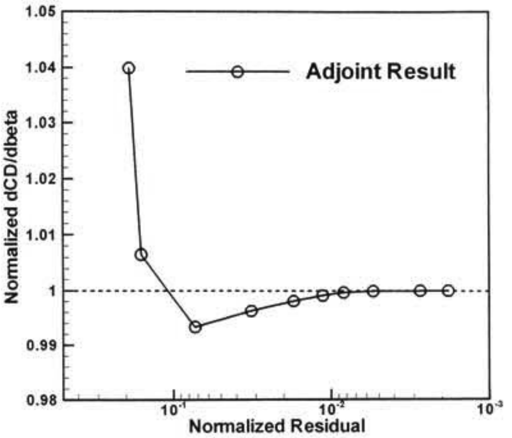


Fig.3 Convergence trends of sensitivity derivative with respect to residual of the adjoint code

Design Methodology

Design Objective

The present design method using the unstructured Euler solver and the adjoint method is applied to an experimental SST wing with a flow-through type engine nacelle attached on its lower surface, which is under development by National Aerospace Laboratory of Japan as a basic study for the next generation supersonic transport.[20]

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

$$\text{Minimize } C_D \qquad (23)$$
$$\text{Subject to } C_L = C_L^*$$

where C_D and C_L are drag and lift coefficients, respectively, and C_L^* is specified. 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 iteration of the LU-SGS time integration after the residual is reduced by 2 orders of magnitude to obtain a lift coefficient satisfying the following inequality conditions.

$$C_L^* \leq C_L \leq 1.003\, C_L^*$$

Since 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,

$$F = C_D^* = C_D + \frac{\partial C_D}{\partial \alpha}\Delta\alpha, \qquad (24)$$

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 a target one. Similar relation can be written for the lift.

$$C_L^* = C_L + \frac{\partial C_L}{\partial \alpha}\Delta\alpha, \qquad (25)$$

where C_L is a lift coefficient without any incidence angle variation, and C_L^* is the target lift coefficeint, which is 0.100 for this case. If we arrange the above equation for

$\Delta\alpha$ and input to Eq.(24), we obtain a modified objective function

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

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 same 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[21], f_k as follows.

$$y_{new} = y_{initial} + \sum_{k=1}^{n_v} \beta_k \cdot f_k, \tag{27}$$

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

where β_k are design variables, n_v the number of design variables, and x_k represents the peak location of f_k. Although this 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 the SST wing span and defined 20 Hicks-Henne design variables and one twist angle per a design section. Figure 4 shows ten Hicks-Henne functions used for upper and lower surface perturbation. In addition to the 105 design variables, the height of diverter is also considered as a design parameter. With the new geometry of the design sections, node points on the wing surface 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.

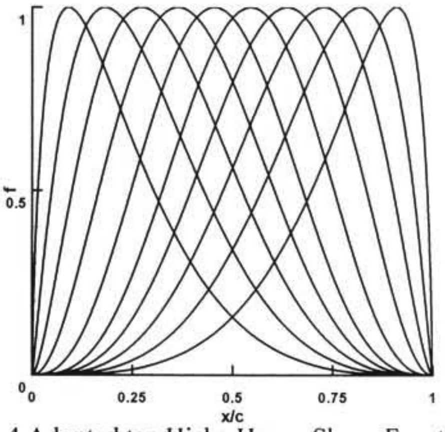


Fig.4 Adopted ten Hicks-Henne Shape Functions

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[22]. In the method, the displacement $\boldsymbol{\delta x}$ 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

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

where *Vol* is a control volume of each grid point and $\boldsymbol{\varepsilon}$ is a small positive number to prevent $\boldsymbol{k}$ from becoming negative. In the original form of the method in Ref.22, *Vol* is obtained from deformed grid system $\boldsymbol{x_0} + \boldsymbol{\delta x}$. 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 since 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[23]. Required computational time to obtain converged solution $\boldsymbol{\delta x}$ was 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. Since 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 one hundred.

One possible way to reduce the computational burden of the grid sensitivity calculation is to neglect the grid sensitivity of interior node points. 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 2-D 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.

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. Figure 5 compares the derivatives of the objective function ob-

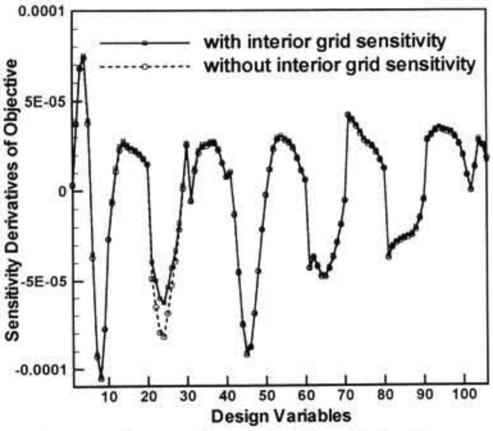


Fig.5 Comparison of sensitivity derivatives with and without interior grid sensitivity information

tained with and without the interior grid sensitivity information. Derivatives with respect to the design variables have little difference between the two values except those of 21 ~ 30 in the design variables. The design variables with indices from 21 to 30 are defined on the lower surface of the second design section, which is located at the centerline of the diverter. Thus, they cause the nacelle to be translated vertically because of the constraint on the leading edge height which will be mentioned in a following section. It has been shown in Ref.7 that for geometries with singularity such as sharp trailing edges, interior mesh sensitivities must be included for the calculation of the derivatives associated with translation. In this case, the nacelle inlet and outlet have sharp edges, which causes the derivatives calculated without interior mesh sensitivities to be deviated from those values obtained with the mesh sensitivities. It can be noted here that 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, without major accuracy degradation.

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 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. Reminding that the present study deals with the Euler equations, this disagreement seems to be caused by the effects of the viscosity and/or turbulence model considered in the reference. However, further research is required to reveal the exact reason of the disagreement.

In this study, interior grid sensitivities for the ten design variables (21~30) are calculated by the elliptic equation method, while for other design variables, only the surface grid sensitivities are defined. This simplification approach required only a quarter of the computational time for the approach computing all the interior grid sensitivities.

Optimization Method

For the minimization of the objective function with specified constraints, the ADS(Automated Design Synthesis) program[24] was used as an optimizer. The Sequential Quadratic Programming (SQP) method[25] 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 formula. Detailed algorithms and methodologies of the SQP method is described in Ref.25.

Design Results

Design conditions are a freestream Mach number of 2.0 and C_L of 0.100. Figure 6 shows the wing-nacelle configuration and surface grids of initial geometry. The number of nodes and cell for the adopted volume grid are about 270,000 and 1,500,000, respectively. The initial geometry has a drag coefficient of 0.02051 and L/D of 4.883, which is much smaller than general SST configurations. This is because the size of NAL experimental aircraft is roughly 10% scale of the assumed actual size SST, and thus the relative size of an engine nacelle is comparatively larger than that of the actual SST.

In the present optimization the diverter leading edge height is also constrained to be larger than the initial value. This lower side constraint is to prevent the boundary layer flow from being entrained into the engine, which might occur if the height of the diverter leading edge becomes smaller than the initial value. Additional 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 initial geometry.

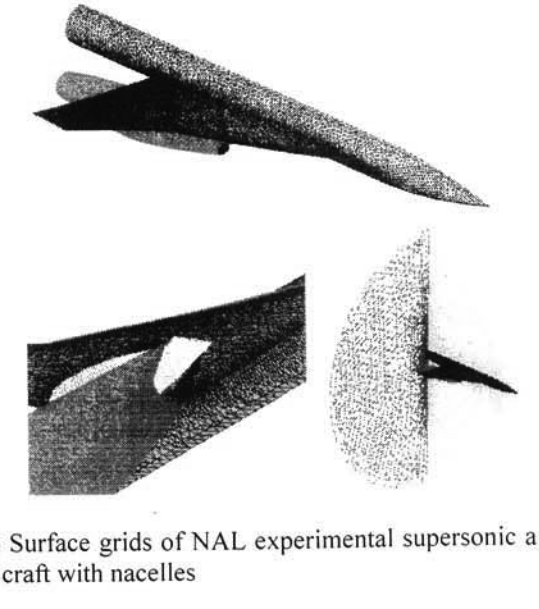


Fig.6 Surface grids of NAL experimental supersonic aircraft with nacelles

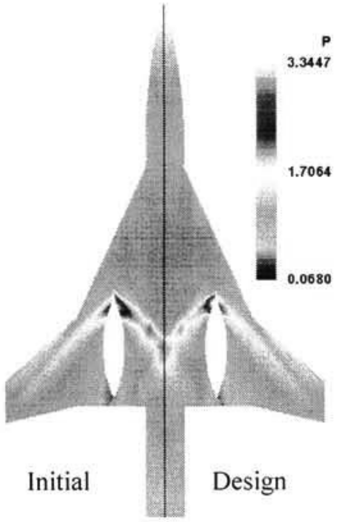


Fig.7 Comparison of lower surface pressure contours

Table 3 Design results: SST wing-nacelle configuration

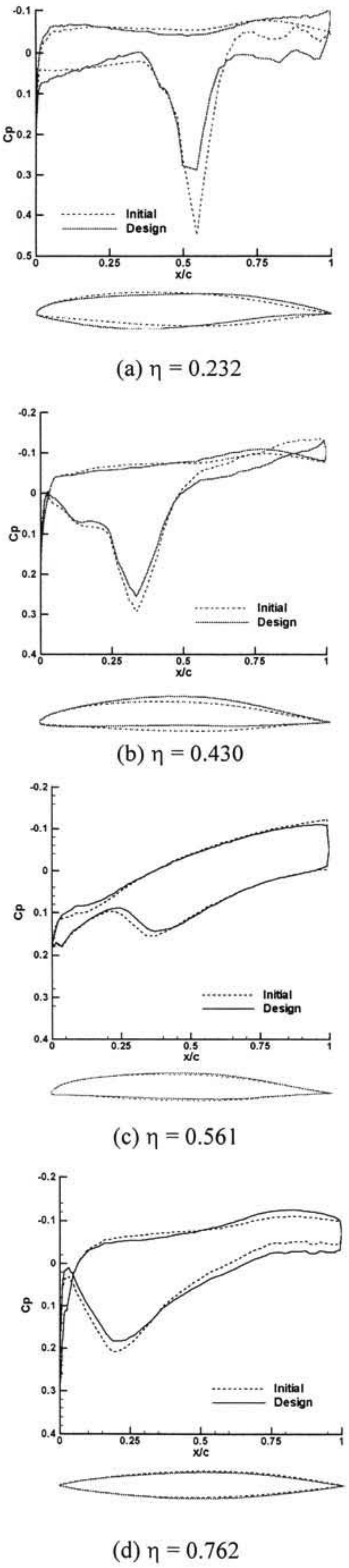
	Initial	Design	Δ (%)
C_L	0.10017	0.10020	+0.03
C_D	0.020513	0.018918	+7.78
L/D	4.883	5.297	+8.48

The density residual of the Euler solver was reduced by four orders from the initial value, and that of the adjoint code by two orders. The SQP optimization iterations converged with three iterations to obtain a drag coefficient reduced by 16 counts from 0.0205 to 0.0189 retaining the lift coefficient as the specified value and satisfying imposed thickness constraints. Table 3 summarizes the design results. During the design process, the Euler solver was run three times and the adjoint code also three times, which is equivalent to about less than six analyses of the Euler solver in computational time.

Figure 7 shows the surface pressure contours on the wing lower surface. It can be noted that 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. Also the strength of the expansion wave at the trailing edge of the diverter has been remarkably reduced. Figure 8 compares wing section shapes and pressure distributions at design sections. The wing section shapes are elongated 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, since the gradient of the objective function with respect to the height is positive throughout the design iteration. This is quite natural in a sense that the volume of the aircraft will be increased and therefore the pressure drag will be increased if the diverter height increases.

Since the present design study is based on the Euler equations, the estimated drag might be deviated from the realistic value, especially for this kind of cases with a strong interaction between shock wave and boundary layer. In order to consider the viscous effects in the design process, employment of the Navier-Stokes



(a) η = 0.232

(b) η = 0.430

(c) η = 0.561

(d) η = 0.762

Fig.7 Design results: section shapes and pressure distributions at design sections; ---- initial, — design.

equations would be necessary, and also, development of a discrete adjoint code for a 3-D unstructured Navier-Stokes solver with a turbulence model would be required.

Concluding Remarks

An aerodynamic design optimization system is developed using the unstructured Euler solver and the discrete adjoint method. Surface geometry is perturbed by simple algebraic shape functions and a twist angle variation. 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 a SST wing with nacelles. 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 three iterations of the SQP optimizer.

Acknowledgement

The first author was supported by the Japan Society for Promotion of Science(JSPS). The authors also wish to thank Mr. Itoh, a graduate student of Tohoku University, for his help to generate the surface grid.

References

1. Eyi, S. and Lee, K. D., "Effect of Sensitivity Calculation on Navier-Stokes Design Optimization," AIAA 94-0060, Jan. 1994.
2. Sherman, L. L., Taylor, III, A. C., 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," AIAA94-4262, Sep. 1994.
3. Eyi, S. and Lee, K. D., "Effect of Sensitivity Analysis on Aerodynamic Design Optimization," FED-Vol.232, CFD for Design and Optimization, ASME 1995.
4. Newman III, J. C., Taylor III, A. C., and Barnwell, R. W., "Aerodynamic Shape Sensitivity Analysis and Design Optimization of Complex Configuration Using Unstructured Grids," AIAA 97-2275, Jan. 1997.
5. Eleshaky, M. E. and Baysal, O., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *Journal of Fluid Engineering*, Vol.115, No.3, pp75-84, 1993.
6. Jameson, A., Pierce, N. A., Martinelli, L., "Optimum Aerodynamic Design using the Navier-Stokes Equations," AIAA 97-0101, Jan. 1997.
7. Anderon, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA 97-0643, 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." J. of Aircraft, vol. 36, No.1, pp51-60, 1999.
9. 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 2." J. of Aircraft, vol. 36, No.1, pp61-74, 1999.
10. Kim, H. J., Kim, C., Rho, O. H., and Lee, K., "Aerodynamic Sensitivity Analysis for Navier-Stokes Equations," AIAA 99-0402, Jan. 1999.
11. Elliot, J. and Peraire, J., "Aerodynamic Optimization on Unstructured Meshes with Viscous Effects." AIAA 97-1849, June 1997.
12. Anderson, W. K. and Bonhaus, D. L., "Airfoil Design on Unstructured Grids for Turbulent Flows." AIAA J. Vol.37, No.2, pp185-1191, 1999.
13. Nielson, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier-Stokes Equations," AIAA J. Vol.37, No.11, pp1411-1419, 1999.
14. Mohammadi, B. " Optimal Shape Design, Reverse Mode of Automatic Differentiation and Turbulence." AIAA 97-0099, Jan. 1997.
15. Obayashi, S., and Guruswamy, G. P., "Convergence Acceleration of an Aeroelastic Navier-Stokes Solver", AIAA J. Vol.33, No.6, pp.1134-1141, 1995.
16. Venkatakrishnan, V., "On the Accuracy of Limiters and Convergence to Steady State Solutions." AIAA Paper 93-0880, January 1993.
17. Yoon, S. and Jameson, A., "Lower-Upper Symmetric-Gauss-Seidal Method for the Euler and Navier-Stokes Equations," AIAA J., Vol.26, No.9 pp. 1025-1026, 1988.
18. Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions," Mathematics of Computation, Vol.37, No.156, pp.385-397, 1981.
19. 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.
20. Iwamiya, T., "NAL SST Project and Aerodynamic Design of Experimental Aircraft," Proc. Computational Fluid Dynamics '98, Vol.2, ECCOMAS 98, John Wiley & Sons, Ltd. pp.580-585, 1998.
21. Hicks, R. M. and Henne, P. A., "Wing Design by Numerical Optimization," J. of Aircraft, Vol.15, No.7, pp. 407-412. 1978.
22. Crumpton, P. I. and Giles, M. B. "Implicit Time Accurate Solutions on Unstructured Dynamic Grids," AIAA 95-1671, June, 1995.
23. Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., "Numerical Recipes in Fortran." 2nd ed. Cambridge Univ. Press, Cambridge, England, UK, 1992.
24. Vanderplaats, G. N., "ADS – A Fortran Program For Automated Design Synthesis version 3.00." Engineering Design Optimization, INC., 1987.
25. Vanderplaats, G. N., "Numerical Optimization Techniques for Engineering Design: With Applications," McGraw Hill, N.Y., 1984.