J. KSIAM **Vol. 3**, No.2, 161-171, 1999

AERODYNAMIC SENSITIVITY ANALYSIS FOR NAVIER-STOKES EQUATIONS

Hyoung-Jin Kim, Chongam Kim, and Oh-Hyun Rho
Ki Dong Lee

Abstract

Aerodynamic sensitivity analysis codes are developed via the hand-differentiation using a direct differentiation method and an adjoint method respectively fromdiscrete two-dimensional compressible Navier-Stokes equations. Unlike previous other researches, Baldwin-Lomax algebraic turbulence model is also differentiated by hand to obtain design sensitivities with respect to design variables of interest in turbulent flows. Discrete direct sensitivity equations and adjoint equations are efficiently solved by the same time integration scheme adopted in the flow solver routine. The required memory for the adjoint sensitivity code is greatly reduced at the cost of the computational time by allowing the large banded flux jacobian matrix unassembled. Direct sensitivity code results are found to be exactly coincident with sensitivity derivatives obtained by the finite difference. Adjoint code results of a turbulent flow case show slight deviations from the exact results due to the limitation of the algebraic turbulence model in implementing the adjoint formulation. However, current adjoint sensitivity code yields much more accurate sensitivity derivatives than the adjoint code with the turbulence eddy viscosity being kept constant, which is a usual assumption for the prior researches.

1. Introduction

With the advances in computational fluid dynamics, design optimization methods in the aerodynamic design are more important than ever. In the application of gradient-based optimization methods to aerodynamic design problems, one of the major concerns is an accurate and efficient calculation of sensitivity derivatives of system responses of interest, which are usually aerodynamic coefficients or surface pressure distributions with respect to design variables.

The finite difference approximation approach is the easiest way to use since it does not require any development of a sensitivity code. However, the accuracy of finite difference approach depends critically on the perturbation size of design variables and the flow initialization.[1]

A robust way of computing sensitivity derivatives is to build a sensitivity analysis code. A sensitivity analysis code can be developed by direct differentiation methods[2-5] or adjoint variable methods[6-8]. Direct differentiation methods are more economical than adjoint variable methods when the numbers of objectives and constraints are larger than the number of design variables.

Key Words: Sensitivity derivative, direct differentiation, adjoint variable

Adjoint variable methods are preferable in the opposite case.

Both methods can be dealt with either discreteor continuous approach. In the discrete approach, discretized flow equations are differentiated, while the flow equations are differentiated before they are discretized in the continuous approach. The discrete approach can be advantageous in the sense that the derivatives obtained are consistent with finite-difference derivatives regardless of a computational grid size. On the other hand, the continuous approach provides a clear insight into the nature of the sensitivity solution.

Previous works on the sensitivity analysis by human hand in both direct differentiation and adjoint variable methods have shown troubles in differentiating viscosity terms, which reflects the variation of laminar and/or turbulent viscosities with respect to the variation of design variables.[5,6] Automatic differentiation tools such as ADIFOR[3,4] and Odyssée[7,8] have been successfully used for a sensitivity code generation from Navier-Stokes codes including turbulence models. However, the sensitivity code generated by the automatic differentiation is much less efficient than a hand-differentiated one.[3,8]

Another problem with adjoint codes generated by automatic differentiation in reverse mode[7,8] or by human hand[2] is a memory problem. Actually, they require much more memory than the original flow solver and are prohibitive for large two-dimensional problems and all three-dimensional problems.

In this study, a Navier-Stokes solver with the Baldwin–Lomax algebraic turbulence model is directly differentiated by hand, and a corresponding adjoint code is developed from the direct-differentiated sensitivity code. The required memory for the adjoint sensitivity code is greatly reduced at the cost of the computational time by allowing the large banded flux jacobian matrix unassembled. Sensitivity derivatives obtained by the sensitivity codes developed herein are compared with those calculated using the finite difference approximation.

The rest of this paper presents a brief review on the flow solver used in this study, and a basic theory of the direct differentiation method and the adjoint variable method in the discrete approach. Computational results are then given for example problems, including subsonic and transonic laminar and turbulent flows around NACA0012 airfoil.

2. Flow Analysis

A two-dimensional Navier-Stokes solver developed and validated in Ref.[9,10] was used for the flow analysis. Reynolds averaged two-dimensional compressible Navier-Stokes equations in generalized coordinates are used in the conservation form based on a cell-centered finite volume approach, given as

$$\frac{1}{J}\frac{\partial Q}{\partial t} + R = 0, \tag{1}$$

where R is the residual vector, Q is a four-element vector of conserved flow variables. Navier-Stokes equations are discretized in time using the Euler implicit method and linearized by employing the flux jacobian. This results in a large system of linear equations in delta form at each time step as

$$\left(\frac{I}{J\Delta t} + \left[\frac{\partial R}{\partial Q}\right]^n\right)\Delta Q = -R^n \,. \tag{2}$$

Roe's Flux Difference Splitting (FDS) scheme was adopted for the space discretization in the inviscid flux terms of the residual vector on the right-hand side; MUSCL approach with Koren limiter is employed to obtain a third order accuracy. The central difference method is used for viscous flux terms of the residual vector. In the implicit part, Beam & Warming's Alternating Direction Implicit (ADI) method is used, and van Leer's Flux Vector Splitting (FVS) is employed with a first order accuracy for the flux jacobian. The flux jacobian for the viscous part is neglected in the implicit part since it does not influence the solution accuracy. Turbulence effects were considered using the Baldwin-Lomax algebraic model with a relaxation technique. All boundary conditions were specified explicitly. One-dimensional characteristic conditions were used for the inflow and outflow boundaries. The no-slip condition and adiabatic wall condition were specified on the solid wall, and local time stepping was used.

A C-type grid system around the airfoil was generated by a conformal mapping technique, giving 135 points in the chordwise direction, 41 points in the normal direction and 95 points on the airfoil surface.

3. Sensitivity Analysis

Direct Differentiation Method

The discrete residual vector of nonlinear aerodynamic analysis for steady problems can be written symbolically as

$$R[Q(\beta), X(\beta), \beta] = 0\,, \tag{3}$$

where **X** is the grid position vector, and **b** is the vector of design variables. Boundary conditions are also included in the residual vector **R.**

Eq. (3) is directly differentiated with respect to β_k to yield the following equation.

$$\left\{\frac{dR}{d\beta_k}\right\} = \left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial X}\right]\left\{\frac{dX}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\} = 0\,. \tag{4}$$

The grid sensitivity vector $\{dX/d\beta_k\}$ can be calculated by differentiating the grid generation code or simply by applying the finite difference approxiation. However, it can be obtained analytically, if the grid points are analytically modified during the design process.

In order to find the solution $\{dQ/d\beta_k\}$ of Eq. (4), a pseudo time term is added and the same time integration scheme with the flow solver is adopted. Applying Euler implicit method followed by the linearization with van Leer flux jacobian of a first-order accuracy gives the following system of linear algebraic equations.

$$\left(\frac{I}{J\Delta t} + \left[\frac{\partial R}{\partial Q}\right]\right)\Delta\left\{\frac{dQ}{d\beta_k}\right\} = -\left\{\frac{dR}{d\beta_k}\right\}^n\,. \tag{5}$$

The above system of equations is solved with ADI scheme which is used for the flow solver.

By comparing Eq. (2) and (5), it can be noted that one can obtain a direct sensitivity code by directly differentiating the right-hand side of the discretized flow equations. All of the derivative terms in Eq. (4) are differentiated by hand except grid sensitivity vector $\{dX/d\beta_k\}$, which are calculated from a grid generation code.

The jacobian matrices $[\partial R/\partial Q]$ and $[\partial R/\partial X]$ in Eq. (4) include Roe's FDS flux jacobian and viscous flux jacobian and are very large banded matrices as the inviscid and viscous fluxes are third- and second- order accurate, respectively. In order to avoid this problem, the terms $[\partial R/\partial Q]\{dQ/d\beta_k\}$ and $[\partial R/\partial X]\{dX/d\beta_k\}$ of Eq. (4) are calculated without the explicit formulation of the very large Jacobian matrices $[\partial R/\partial Q]$ and $[\partial R/\partial X]$. Van Leer's flux Jacobian in the LHS is frozen at the steady-state value. This leads to the required memory increase and computational time reduction.

When the flow variable sensitivity vector $\{dQ/d\beta_k\}$ is obtained, The total derivative of the system response of interest, C_j can be calculated. C_j is a function of flow variables **Q**, grid position **X**, and design variables **b**; i.e.,

$$C_j = C_j\big(Q(\beta), X(\beta), \beta\big)\,. \tag{6}$$

The sensitivity derivative of the aerodynamic coefficient C_j with respect to the k^{th} design variable β_k is given by

$$\left\{\frac{dC_j}{d\beta_k}\right\} = \left\{\frac{\partial C_j}{\partial Q}\right\}^T \left\{\frac{dQ}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial X}\right\}^T \left\{\frac{dX}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \beta_k}\right\}. \tag{7}$$

Adjoint Variable Method

As the total derivative of the residual vector $\{dR/d\beta_k\}$ is null in the steady state, we can introduce adjoint variables and combine Eq. (4) and (7) to obtain

$$\left\{\frac{dC_j}{db_k}\right\} = \left\{\frac{\partial C_j}{\partial Q}\right\}^T \left\{\frac{dQ}{db_k}\right\} + \left\{\frac{\partial C_j}{\partial X}\right\}^T \left\{\frac{dX}{db_k}\right\} + \left\{\frac{\partial C_j}{\partial b_k}\right\} + \{\lambda_j\}^T \left\{\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial X}\right]\left\{\frac{dX}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\}\right\}. \tag{8}$$

If we find $\{l_j\}^T$ that satisfies the following adjoint equation

$$\left[\frac{\partial R}{\partial Q}\right]^T \{l_j\} + \left\{\frac{\partial C_j}{\partial Q}\right\} = 0 \quad , \tag{9}$$

we can obtain the sensitivity derivative of C_j with respect to β_k by the following equation.

$$\left\{\frac{dC_j}{d\beta}\right\} = \left\{\frac{\partial C_j}{\partial X}\right\}^T \left\{\frac{dX}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \beta_k}\right\} + \{\lambda_j\}^T \left\{\left[\frac{\partial R}{\partial X}\right]\left\{\frac{dX}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\}\right\}. \tag{10}$$

The adjoint equation (9) is also converted to the following system of linear algebraic equations and is solved by the ADI scheme.

$$\left(\frac{I}{J\Delta t} + \left[\frac{\partial R}{\partial Q}\right]^T\right)\Delta\{\lambda_j\} = -\left\{\left[\frac{\partial R}{\partial Q}\right]^T \{\lambda_j\} + \left\{\frac{\partial C_j}{\partial Q}\right\}\right\}^n . \tag{11}$$

The transposed flux Jacobian $[\partial R/\partial Q]^T$ in the LHS of Eq. (11) is van Leer's FVS flux jacobian that is frozen at the steady-state value. The transposed flux Jacobian $[\partial R/\partial Q]^T$ in the RHS includes Roe's FDS flux Jacobian and viscous flux Jacobian and is a very large banded matrix. Unlike the flux Jacobian $[\partial R/\partial Q]$ of the direct differentiation method, all the element of $[\partial R/\partial Q]^T$ should be explicitly calculated. In the prior researches on the discrete adjoint variable methods[2,7], all the elements of the jacobian matrix $[\partial R/\partial Q]^T$ were calculated and assembled at the cost of very large memory requirement. Although the computational time can be decreased as the elements of the jacobian matrix are calculated only once, the memory requirement is prohibitive for large two-dimensional problems and all three-dimensional problems.

In the present study, elements of $[\partial R/\partial Q]^T$ are calculated and multiplied by the corresponding element of the adjoint vector $\{\lambda_j\}$, and thus the large matrix $[\partial R/\partial Q]^T$ need not to be assembled. This increases the computational time as the elements of the flux jacobian matrix $[\partial R/\partial Q]^T$ have to be calculated every iteration. However, the required memory can be remarkably reduced to the order of the required memory of the flow solver.

In Ref.[3] it was reported that a hand-differentiated sensitivity code may take six man-months to two man-years, or even longer to be generated, whereas it takes about one man-week to generate a direct-differentiation sensitivity code by the automatic differentiation. According to authors' experience, however, less than two man-weeks were required to build a hand-differentiated direct differentiation sensitivity code, and one man-month to build an adjoint sensitivity code for two-dimensional Navier-Stokes equations. This indicates that the hand-differentiation is a viable approach compared to the automatic differentiation in the required time for the sensitivity code generation.

4. Results & Discussion

In order to evaluate the performance of the direct and adjoint sensitivity codes to accurately calculate sensitivity derivatives, sensitivity analyses are conducted for NACA0012 airfoil in the laminar/turbulent flow field of subsonic and transonic regime.

The results are compared with the derivatives computed by the following finite difference approximation.

$$\frac{dC_j}{d\beta_k} \cong \frac{C_{j,\beta_k+\Delta\beta_k} - C_{j,\beta_k}}{\Delta\beta_k} \quad . \qquad (12)$$

The residual of the flow solver is reduced to 10^{-11} from the freestream value for the finite difference calculation. The step size $\Delta\beta_k$ is 10^{-6} or 10^{-7} depending on the design variable and the flow condition.

The residuals of the sensitivity codes are reduced to 10^{-7} from the initial value of the residual. The initial value of the sensitivity derivatives $\{dQ/d\beta_k\}$ and adjoint variables $\{\lambda_j\}$ is set to zero.

In order to consider the sensitivity derivative with respect to the geometry change, one of the Hicks-Henne functions is adopted as a shape function as follows.

$$F(x) = \sin^3\left[px^{\frac{\ln(0.5)}{\ln(0.6)}} \right] \quad . \qquad (13)$$

The upper surface of NACA0012 airfoil is modified as follows.

$$Y_{new} = Y_{old} + \beta F(x) \quad . \qquad (14)$$

Where β is a design variable. The incidence angle α is the other design variable in this study.

Direct Differentiation Approach

In order to validate the sensitivity code by the direct differentiation method, sensitivity derivatives obtained by the direct sensitivity code is compared to those by the finite difference method. Computations are conducted for both subsonic and transonic turbulent flows.

The first example is a subsonic turbulent flow case. The flow condition is M_∞ = 0.6, α = 2° and Reynolds number = 6,500,000. Fig.1 shows pressure derivative ($p' = dp/d\alpha$) contours. Streamlines of velocity sensitivity derivatives (u', v') in Fig. 2 show the effect of the incidence angle increment. Table 1 shows the computed sensitivity derivatives of aerodynamic coefficients with respect to incidence angle α and the coefficient β of the shape function. The sensitivity derivatives of aerodynamic coefficients by the direct differentiation and the finite difference method compare very well with each other up to four or five significant digits.

Table 1 Validation of direct sensitivity code:
subsonic turbulence case (M_∞ = 0.60, α = 2° and Re = 6,500,000)

	$dC_l/d\alpha$	$dC_d/d\alpha$	$DC_m/d\alpha$
FD ($\Delta\alpha=10^{-6}$)	7.379933	0.05013069	0.1343019
DD	7.379935	0.05013063	0.1343012
	$dC_l/d\beta$	$dC_d/d\beta$	$dC_m/d\beta$
FD($\Delta\beta=10^{-6}$)	3.65374	0.0217594	-1.26809
DD	3.65373	0.0217605	-1.26808

Table 2 Validation of direct sensitivity code:
transonic turbulence case (M_∞ = 0.75, α = 2° and Re = 6,500,000)

	$dC_l/d\alpha$	$dC_d/d\alpha$	$dC_m/d\alpha$
FD ($\Delta\alpha=10^{-6}$)	8.37948	0.486983	0.222065
DD	8.37972	0.486994	0.222014
	$dC_l/d\beta$	$dC_d/d\beta$	$dC_m/d\beta$
FD ($\Delta\beta=10^{-7}$)	7.13337	-0.195777	-1.85788
DD	7.13346	-0.195774	-1.85790

The second example is a transonic turbulent flow case. The flow condition is M_∞ = 0.75, α = 2° and Reynolds number = 6,500,000. A strong shock wave is formed on the upper surface of the airfoil. Fig. 3 shows pressure sensitivity contours with respect to incidence angle α, which represents a drastic variation of the pressure around the shock wave. Streamlines of the velocity derivative are similar to the subsonic case, but it is getting close to the flow separation downstream of the shock wave as can be seen in Fig. 4. Table 2 presents the sensitivity derivatives of aerodynamic coefficients with respect to two design variables. As in the subsonic case, the sensitivity derivatives of aerodynamic coefficients by the direct differentiation method and the finite difference method are identical with each other up to four or five significant digits.

Adjoint Variable Method

In order to validate the sensitivity code by the adjoint variable method, the sensitivity derivatives calculated by using the adjoint sensitivity code are compared to those by the direct sensitivity code. Computations are conducted for both subsonic laminar and transonic turbulent flows.

The first example is a subsonic laminar case. The flow condition is M_∞ = 0.6, α = 1° and Reynolds number = 5,000. Fig.5 shows λ_{m1} contours and streamlines of (λ_{m2},λ_{m3}). The subscript m means that the system response of interest C_j was C_m in the adjoint equations. The integer subscript figures 1, 2 and 3 represents elements of the adjoint variable vector $\{l_j\}^T$ $(=\{\lambda_{j1},\lambda_{j2},\lambda_{j3},\lambda_{j4}\}^T)$ corresponding to the conservative flow variable vector $\mathbf{Q}$ (= $\{\rho, \rho u, \rho v, e\}^T$). The λ_{m1} has a large gradient in the boundary layer region on the airfoil surface and around the stagnation streamline upstream of the airfoil leading edge. Streamlines of the vector (λ_{m2},λ_{m3}) show discontinuity of the vector direction around the flow stagnation streamline upstream of the airfoil leading edge. Streamlines of the vector (λ_{m2},λ_{m3}) look like circulation lines with a negative lift or the opposite flow direction.

In Table 3, we can note that the total derivatives of aerodynamic coefficients with respect to geometric perturbation by the direct and adjoint sensitivity codes are identical with each other up to 6 significant digits.

The second example is a transonic turbulent flow case. The flow condition is $M_\infty = 0.75$, $\alpha = 2°$ and Reynolds number = 6,500,000. The adjoint sensitivity code was run in two modes, one with turbulence eddy viscosity (μ_t) terms differentiated and the other with constant turbulence eddy viscosity (μ'_t=0). λ_{l1} contours around NACA0012 airfoil are plotted in Fig. 6. The λ_{l1} has a large gradient around the stagnation streamline and in the boundary layer on the airfoil surface.

Table 3 Validation of adjoint sensitivity code:
subsonic laminar case ($M_\infty = 0.60$, $\alpha = 1°$ and Re = 5,000)

	$dC_l/d\beta$	$dC_d/d\beta$	$dC_m/d\beta$
AV	-1.2065570	0.08362676	-0.12162388
FD	-1.2065569	0.08362679	-0.12162394

discontinuity of the adjoint variables around the stagnation streamline in Fig. 5 and 6 is due to the existence of singularity crossing the incoming stagnation streamline upstream of the airfoil leading edge.[11]

Table 4 Validation of Adjoint Sensitivity Code:
transonic turbulence case ($M_\infty = 0.75$, $\alpha = 2°$ and Re = 6,500,000)

	$dC_l/d\alpha$	$dC_d/d\alpha$	$dC_m/d\alpha$
DD	8.37972	0.486994	0.222014
AV	8.28911 (0.9892)	0.486983 (0.9948)	0.242023 (1.0901)
AV (μ'_t=0)	7.73075 (0.9226)	0.483649 (0.9931)	0.342835 (1.544)
	$dC_l/d\beta$	$dC_d/d\beta$	$dC_m/d\beta$
DD	7.13346	-0.195774	-1.85790
AV	7.17228 (0.9946)	-0.190391 (0.9725)	-1.86124 (1.0018)
AV (μ'_t=0)	9.37707 (1.3145)	-0.123950 (0.6331)	-2.233926 (1.2024)

Table 4 compares the sensitivity derivatives of aerodynamic coefficients with respect to two design variables. The values in the parentheses are sensitivity derivative ratios that are the sensitivity derivatives via the adjoint sensitivity code normalized by the respective sensitivity derivatives via the direct differentiation code. The sensitivity derivatives of the adjoint sensitivity code with differentiated eddy viscosity terms show about 0.18~9 % deviation from those of the direct differentiation code. On the other hand, the adjoint code results with μ'_t=0 show much larger deviations(0.5~54%).

Required Computational Time and Memory

Table 5 compares computational time per iteration and memory for the sensitivity codes developed herein with those for the codes generated by the automatic differentiation tools. The presented values are normalized by the time and memory required for the original flow solver. The sensitivity codes developed in this study require less computational time and memory than those by automatic differentiation. Although computational time per iteration for an Odyssée-generated adjoint code was not available, it is significantly slower than a hand-differentiated code by a factor of 5 in CPU time.[8]

The direct and adjoint sensitivity codes have similar convergence ratio with the flow solver as they all employ van Leer's flux jacobian in the implicit part. The direct sensitivity code requires much less computational time than finite difference approximations as its convergence tolerance to obtain accurate sensitivity derivatives is larger than that of the flow solver by four orders of magnitude approximately.

Table5 Comparison of computational time and memory

	Flow Solver	Sensitivity code			
		Present		AD	
		DD	AV	DD[a]	AV[b]
Time Per iteration	1	1.1	2.4	2~3	NA
Memory	1	1.8	1.8	2	10

a: by ADIFOR[4] b: by Odyssée[7], NA : Not Available

The adjoint sensitivity code requires more than two times the computational time of the direct sensitivity code because the elements of the residual vector R are differentiated by the four elements of the flow variable **Q** instead of the design variable β_k as in the direct code. Thus, the adjoint code would be more economical than the direct sensitivity code to calculate sensitivity derivatives if the number of design variables is larger than twice the number of objectives and constraints.

5. Concluding Remarks

The direct differentiation approach and the adjoint variable approach are applied respectively to the discrete flow equations to develop aerodynamic sensitivity analysis codes for turbulent flows. A Navier-Stokes solver with Baldwin-Lomax turbulence model is differentiated by hand to obtain design sensitivities with respect to design variables of interest efficiently. Direct sensitivity equations and adjoint equations are efficiently solved by the same time integration scheme with the flow solver. The required memory for the adjoint sensitivity code is greatly reduced at the cost of the computational time by allowing the large banded flux jacobian matrix unassembled.

Sensitivity derivatives computed by the sensitivity codes almost exactly coincide with those of the finite difference method. Although adjoint code results of the turbulent flow case show slight deviations from the exact results, the adjoint sensitivity code gives much more accurate sensitivity derivatives than the adjoint code with the turbulence eddy viscosity being kept constant, which is a usual assumption for the prior researches.

The strategy adopted in this research shows a promise for the extension to three-dimensional problems as it is efficient, accurate and requires much less memory than the prior approaches.

6. References

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

[2] Eleshaky, M. E. and Bayal, O. "Aerodynamic Shape Optimization Using Sensitivity Analysis

on Viscous Flow Equations," J. of Fluid Engineering, Vol.115, No.3, 1993, pp75-84

[3] 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," AIAA-94-4262-CP, Sep. 1994.

[4] Taylor III, A. C., Oloso, A., " Aerodynamic Design Sensitivities By Automatic Differentiation," AIAA98-2536, June, 1998

[5] Ajmani, K., and Taylor, III, A. C., "Discrete Sensitivity Derivatives of the Navier-Stokes Equations with a Parallel Krylov Solver," AIAA 94-0091, Jan. 1994.

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

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

[8] Malé, J. M., Mohammadi, N., Schmidt, R., " Direct and Reverse Modes of Automatic Differentiation of Programs for Inverse Problems," Application to Optimum Shape Design Proc. 2nd Int. SIAM Workshop on Computational Differentiation, Santafe, 1996

[9] S. W. Hwang, " Numerical Analysis of Unsteady Supersonic Flow over Double Cavity," Ph.D. Thesis, Seoul National Univ., Seoul, Korea, 1996.

[10] Kim, H. J. and Rho, O. H.," Dual-Point Design of Transonic Airfoils using the Hybrid Inverse Optimization Method," J. of Aircraft vol.34 No.5 pp612-618, 1997.

[11] Giles, M. B. and Pierce , N. A. "Adjoint Equations in CFD: Duality, Boundary Conditions and Solution Behavior," AIAA 97-1850, June, 1997

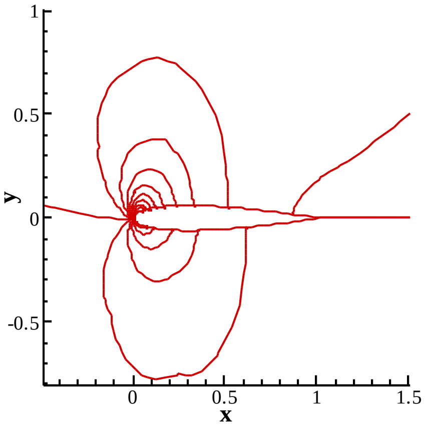

Fig.1 Direct differentiation sensitivity analysis: dp/dα contour, subsonic turbulence case (M_∞ = 0.60, α = 2°, Re = 6,500,000)

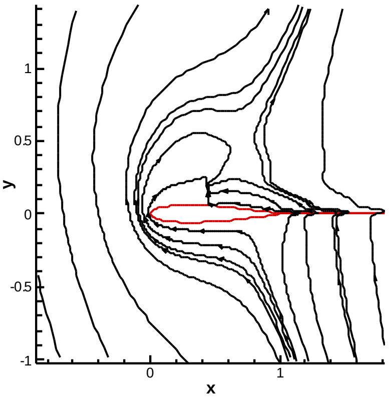

Fig.2 Direct differentiation senitivity analysis: velocity sensitivity streamline for α subsonic turbulence case (M_∞ = 0.60, α = 2°, Re = 6,500,000)

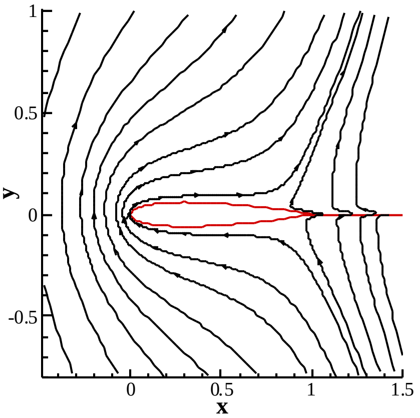

Fig.3 Direct differentiation sensitivity analysis: pressure sensitivity contour for α transonic turbulence case (M_∞ = 0.75, α = 2°, Re = 6,500,000)

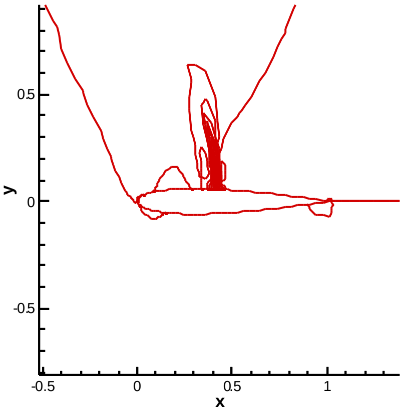

Fig.4 Direct differentiation sensitivity analysis: velocity sensitivity streamline for α transonic turbulence case (M_∞ = 0.75, α = 2°, Re = 6,500,000)

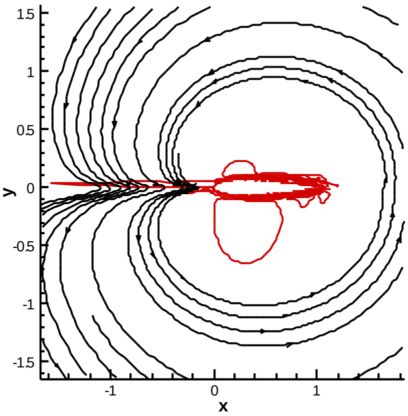

Fig.5 Adjoint sensitivity analysis:
λ_{m1} contour and (λ_{m2},λ_{m3}) streamlines
subsonic laminar case
(M_{∞} = 0.60, α = 1°, Re = 5,000)

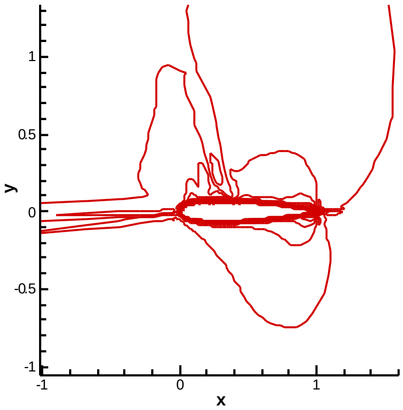

Fig.6 Adjoint sensitivity analysis : λ_{11} contour
transonic turbulence case
(M_{∞} = 0.75, α = 2°, Re = 6,500,000)