AIAA JOURNAL
Vol. 30, No. 10, October 1992

Methodology for Calculating Aerodynamic Sensitivity Derivatives

Arthur C. Taylor III,* Gene W. Hou,† and Vamshi Mohan Korivi‡
Old Dominion University, Norfolk, Virginia 23529

A general procedure is developed for calculating aerodynamic sensitivity coefficients using the full equations of inviscid fluid flow, where the focus of the work is the treatment of geometric shape design variables. Using an upwind cell-centered finite volume approximation to represent the Euler equations, sensitivity derivatives are determined by direct differentiation of the resulting set of coupled nonlinear algebraic equations that model the fluid flow. The technique is implemented and successfully tested in two dimensions for flow through a subsonic nozzle ($M_\infty = 0.85$) and also a supersonic inlet ($M_\infty = 2.0$). Specifically, the method is demonstrated by calculating the sensitivity of the aerodynamic loads (forces) on the interior walls of the nozzle/inlet to variations in the geometric parameters that define the shape. The sensitivity coefficients calculated using this approach compare very well with those calculated using the method of "brute force" (i.e., using finite differences to approximate the sensitivity derivatives) and are computationally less expensive to obtain.

Introduction

THE theme of this study is the development of procedures whereby fundamental principles of a discipline known as sensitivity analysis are merged with modern computational techniques for the governing equations of fluid mechanics. Through very rapid advances in recent years in a field now known as computational fluid dynamics (CFD), researchers working on supercomputers are now routinely generating detailed, accurate, numerical solutions to very complex flow problems. Modern CFD software is now being applied to practical flow problems by engineers who are using these codes as design tools, and the present research is motivated by the belief that the basic concepts of sensitivity analysis, if properly applied, could greatly enhance the power, efficiency, and effectiveness of these existing CFD codes in a design environment.

A sensitivity analysis can be defined as the calculation of slopes, known as sensitivity coefficients, where the derivatives of the responses of a particular system of interest are taken with respect to the design variables of interest. For the designer, an accurate knowledge of the sensitivity derivatives of the particular system under consideration can subsequently be exploited in several useful ways, with one of the best and most important examples being the use of the sensitivity derivatives in design optimization. In the field of computational structural mechanics, the subject of sensitivity analysis is relatively mature, and the references that are available are extensive. In the field of CFD, researchers have been and continue to be active, and Refs. 1-8 are relevant to the present study (although this list is by no means exhaustive).

The focus of this work will be the development of methods for efficiently and accurately calculating aerodynamic sensitivity coefficients, with emphasis on the treatment of variations of geometric shape. This emphasis is selected because, as a design variable, geometric shape is often one of the most important design variables and is also one of the most challenging variables to consider. Furthermore, calculations of the sensitivity derivatives of aerodynamic loads will be demonstrated in the test problems, because this also represents one of the most important examples. However, the methods that are to be presented are general in nature and are not restricted to this example. Finally, all techniques are presently shown in two dimensions using the Euler equations.

The Euler equations are represented in this study using a well-known upwind cell-centered finite volume approximation. Fundamental sensitivity equations are then derived using a method of direct differentiation (known in Refs. 2 and 4 as the quasianalytical method) of the resulting set of coupled nonlinear algebraic equations which model the Euler equations. A closely related well-known alternative method for computing sensitivity derivatives, known as the adjoint variable approach, is presented and discussed. Comparisons are made of the calculated sensitivity derivatives obtained using the methods discussed herein with sensitivity derivatives calculated using the method of brute force (i.e., the simplest method of calculating sensitivity derivatives, using finite difference approximations).

The rest of this article is organized as follows: After this introduction, the next section is a presentation of basic theory, which is further divided into five subsections to include 1) a review of the governing equations, 2) a review of the spatial discretization and implicit formulation, 3) the fundamental sensitivity equations, 4) terms involving grid sensitivities, and 5) special ancillary sensitivity equations. Following the presentation of theory, computational results are given from the application of the methods to two example problems, including 1) a subsonic nozzle ($M_\infty = 0.85$) and 2) a supersonic inlet ($M_\infty = 2.0$). The final major section is a brief summary of the work, where conclusions are given.

Presentation of Theory

Governing Equations

The governing equations employed here are the unsteady two-dimensional Euler equations, given as

$$\frac{1}{J}\frac{\partial Q}{\partial t} = R(Q) \tag{1}$$

Received Oct. 15, 1990; presented as Paper 91-1101 at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 8-10, 1991; revision received April 9, 1992; accepted for publication April 9, 1992.

*Assistant Professor, Department of Mechanical Engineering and Mechanics. Member AIAA.
†Associate Professor, Department of Mechanical Engineering and Mechanics. Member AIAA.
‡Graduate Research Assistant, Department of Mechanical Engineering and Mechanics.

where $R(Q)$, the residual (clearly zero in the steady state), is given as

$$R(Q) = -\frac{\partial \hat{F}(Q)}{\partial \xi} - \frac{\partial \hat{G}(Q)}{\partial \eta} \tag{2}$$

and Q is the usual four-element vector of conserved field variables. A transformation from physical (x, y) coordinates to generalized computational (ξ, η) coordinates is included in the residual, and J is the determinate of the Jacobian matrix of this transformation. The vectors $\hat{F}(Q)$ and $\hat{G}(Q)$ are the inviscid flux vectors in the ξ and η directions, respectively. The ideal gas law with a constant ratio of specific heats is used with these equations for closure.

Spatial Discretization and Implicit Formulation

Computationally, the Euler equations are solved here in their alternative integral conservation law form using an upwind cell-centered finite formulation (only an overview of this method is presented here, with additional details found in Refs. 9–15), where the residual at each cell is written as a balance of fluxes across cell interfaces. Upwind evaluation of the fluxes is accomplished by upwind interpolation of the field variables Q from the approximate cell centers to the cell interfaces, where the flux vector splitting procedure of van Leer[16] is employed. This results in a higher-order accurate [i.e., in this research, second-order accuracy is used in the streamwise (ξ) direction, and third-order accuracy is used in the normal (η) direction] algebraic approximate representation of the residual at each cell in the domain. When written for each cell (including boundary condition relationships) and assembled globally, this can be expressed as

$$\{R(Q^*)\} = \{0\} \tag{3}$$

where $\{Q^*\}$ is called the "root" (i.e., the steady-state value of the field variables). Therefore, Eq. (3) represents a large coupled system of nonlinear algebraic equations; thus, finding a steady-state solution to the Euler equations has been replaced (approximately) by the problem of finding the root, $\{Q^*\}$, of this set of algebraic equations.

The Euler equations are discretized in time using the Euler implicit method, followed by a Taylor's series linearization of the discrete equations in time about the known time level. This results in a large system of linear algebraic equations at each time step, written as

$$\left[\left[\frac{1}{J\Delta t}\right] - \left[\frac{\partial R^n(Q)}{\partial Q}\right]\right]\{{}^n\Delta Q\} = \{R^n(Q)\} \tag{4}$$

$$\{Q^{n+1}\} = \{Q^n\} + \{{}^n\Delta Q\}$$
$$n = 1, 2, 3, \ldots \tag{5}$$

Equations (4) and (5) represent the fundamental implicit formulation for integrating the Euler equations in time to steady-state. In these equations, n is the time iteration index, and $\{{}^n\Delta Q\}$ is the incremental change in the field variables between the known (nth) and the next (nth + 1) time levels. The matrix $[1/J\Delta t]$ is diagonal and contains the time terms. The large Jacobian matrix $[\partial R^n(Q)/\partial Q]$ is sparse and has a banded structure, with nine diagonals, the individual elements of which are 4×4 block coefficient matrices. In addition to its use in Eq. (4), this important Jacobian matrix plays another central role in this study, which will be shown later.

In principle, Eq. (4) can be repeatedly solved directly [using Eq. (5) to update the field variables] as the solution is advanced to steady-state, and for very large time steps the direct method represents Newton's root finding procedure for nonlinear equations. However, the direct method is not necessarily the most efficient procedure with respect to overall CPU time,[17] and the large storage requirements of the method make its use not feasible in three dimensions. Therefore, more commonly, an iterative algorithm is selected for use in the repeated solution of Eq. (4). Some popular choices of these iterative algorithms include approximate factorization[18] (AF), conventional relaxation algorithms,[13,14] the strongly implicit procedure[19] (SIP), and the preconditioned conjugate gradient method.[20]

Fundamental Sensitivity Equations

Consider the vector $\bar{\beta}$, the elements of which are independent variables that are typically called the design variables. Some, none, or all of the variables may be related to the geometric shape of the flow problem of interest, where, as stated, the emphasis of the present study will be that of geometric shape variation. Computationally, the geometric shape of the domain is defined by the mesh on which calculations are made, and the complete vector of (x, y) coordinates that defines the mesh is represented symbolically here as $\{\bar{X}\}$. For a steady-state solution the discrete residual vector given by Eq. (3) is rewritten in the following form:

$$\{R(Q^*(\bar{\beta}), \bar{X}(\bar{\beta}), \bar{\beta})\} = \{0\} \tag{6}$$

where in Eq. (6) the explicit dependence of the discrete residual on the computational mesh, $\{\bar{X}\}$, as well as its explicit dependence (if any) on $\bar{\beta}$ has now been emphasized. Direct differentiation of Eq. (6) with respect to β_k, the kth element of $\bar{\beta}$, yields

$$\underbrace{-\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{\partial Q}{\partial \beta_k}\right\}}_{\text{term 1}} = \underbrace{\left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{\partial \bar{X}}{\partial \beta_k}\right\}}_{\text{term 2}} + \underbrace{\left\{\frac{\partial R}{\partial \beta_k}\right\}}_{\text{term 3}} \tag{7}$$

Equation (7) is an exact derivative of the discrete algebraic residual vector and is known in Refs. 2 and 4 as the quasianalytical method. This equation represents the central and most general relationship on which relationships that follow in this section are based. The Jacobian matrix $[\partial R/\partial Q]$ of term 1 of Eq. (7) is identical to that found in the fundamental implicit formulation for numerical time integration [Eq. (4)] of the Euler equations and is thus well understood. The solution vector $\{\partial Q^*/\partial \beta_k\}$ is the sensitivity of the complete vector of field variables with respect to the kth design variable. The coefficient matrix $[\partial R/\partial \bar{X}]$ of term 2 is the Jacobian matrix of the discrete steady-state residual vector, with derivatives taken with respect to the complete vector of (x, y) grid coordinates and is documented in detail in Ref. 7. The vector $\{\partial \bar{X}/\partial \beta_k\}$ of term 2 contains what is referred to here as the grid sensitivity terms, which are the sensitivities of each x and y coordinate point of the entire computational mesh, with derivatives taken with respect to β_k. The grid sensitivity vector will be given special consideration later in the present study. The vector $\{\partial R/\partial \beta_k\}$ found in term 3 accounts for derivatives resulting from explicit dependencies (if any) of the residual vector on β_k. If β_k is *not* a design parameter related to the geometric shape, then term 2 of Eq. (7) will be zero, since in this case the vector $\{\partial \bar{X}/\partial \beta_k\}$ is null. If β_k *is* a geometric shape design parameter, its entire effect on the residual [Eq. (6)] will be felt through the grid, and term 3 of Eq. (7) will be zero.

It is strongly emphasized that all boundary condition relationships must be treated in a fully consistent manner and included in Eq. (7). Proper boundary condition treatment should be considered in the Jacobian matrices $[\partial R/\partial Q]$ and $[\partial R/\partial \bar{X}]$, as well as in the vector $\{\partial R/\partial \beta_k\}$. If accurate results are to be obtained using the present methods, it is critical that this is not neglected as it typically is, for example, in the coefficient matrix of the fundamental implicit formulation for integrating the Euler equations in time [i.e., Eq. (4)].

Note that Eq. (7) is a linear system of equations that in principle can be solved directly for the vector $\{\partial Q^*/\partial \beta_k\}$. Of course, the solution of Eq. (7) must be repeated for each

different element of $\bar{\beta}$ (i.e., for each design variable) for which sensitivity derivatives are desired. However, multiple solutions require only a single LU factorization of the coefficient matrix, which can be repeatedly reused in the forward and backward substitution operations, for an unlimited number of different design variables. The reuse of the LU factorization can represent a substantial savings in computational work, particularly when the linear system of Eq. (7) and/or the number of design variables of interest is large.

Typically the solution of Eq. (7) for the vector $\{\partial Q^*/\partial\beta_k\}$ is not the final goal, but rather the sensitivity derivatives of some specific system responses are sought instead (e.g., for an airfoil, the sensitivities of the lift, drag, and moment coefficients might be required). Consider therefore the jth system response of interest, C_j, which in general can be functionally dependent on the steady-state field variables, $\{Q^*\}$, the grid, $\{\bar{X}\}$, and can also depend explicitly on the design variables, $\bar{\beta}$; i.e.,

$$C_j = C_j(Q^*(\bar{\beta}),\ \bar{X}(\bar{\beta}),\ \bar{\beta}) \tag{8}$$

The total rate of change of the jth system response, C_j, with respect to the kth design variable β_k, is then given by

$$\frac{dC_j}{d\beta_k} = \underbrace{\left\{\frac{\partial C_j}{\partial Q}\right\}^T\left\{\frac{\partial Q^*}{\partial\beta_k}\right\}}_{\text{term 1}} + \underbrace{\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T\left\{\frac{\partial \bar{X}}{\partial\beta_k}\right\}}_{\text{term 2}} + \underbrace{\frac{\partial C_j}{\partial\beta_k}}_{\text{term 3}} \tag{9}$$

where in Eq. (9), terms 2 and/or 3 could be zero, depending on the particular system response (C_j) and design variable (β_k) of concern. Solution of Eq. (7) therfore provides the vector $\{\partial Q^*/\partial\beta_k\}$, which is needed in Eq. (9). Furthermore, for geometric shape sensitivity derivatives, the grid sensitivity vector $\{\partial\bar{X}/\partial\beta_k\}$ of Eq. (7) is reused, if needed, in Eq. (9). Specific ancillary sensitivity relationships of the type given by Eq. (9) that are used in the present study for computing sensitivity derivatives of aerodynamic force coefficients are presented in a later section.

On the left-hand side of Eq. (9), the notation for a total derivative has been used, indicating that the total rate of change of C_j with respect to β_k is included in the expression and to distinguisy it from the partial derivative term (term 3) on the right-hand side of the equation. However, it should be understood that this derivative is still a partial derivative in the sense that C_j is in general a function of multiple independent design variables. For consistency, this notation will continue to be used throughout.

A closely related alternative procedure for computing sensitivity derivatives, known as the adjoint variable approach, is easily developed using the relationships presented thus far. This approach is begun by combining Eqs. (7) and (9) to yield

$$\frac{dC_j}{d\beta_k} = \left\{\frac{\partial C_j}{\partial Q}\right\}^T\left\{\frac{\partial Q^*}{\partial\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T\left\{\frac{\partial \bar{X}}{\partial\beta_k}\right\} + \frac{\partial C_j}{\partial\beta_k} + \{\lambda_j\}^T\left\{\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{\partial Q^*}{\partial\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{\partial \bar{X}}{\partial\beta_k}\right\} + \left\{\frac{\partial R}{\partial\beta_k}\right\}\right\} \tag{10}$$

The adjoint variable vector $\{\lambda_j\}$ is arbitrary at this point, since the inner product of $\{\lambda_j\}$ is taken with the null vector, from Eq. (7). Thus, there is no net change from Eq. (9) to Eq. (10), since the entire additional term on the right-hand side of Eq. (10) is zero, for any and all $\{\lambda_j\}$. Rearranging, Eq. (10) becomes

$$\frac{dC_j}{d\beta_k} = \left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{\partial \bar{X}}{\partial\beta_k}\right\} + \frac{\partial C_j}{\partial\beta_k} + \{\lambda_j\}^T\left\{\frac{\partial R}{\partial\beta_k}\right\} + \left(\left\{\frac{\partial C_j}{\partial Q}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial Q}\right]\right)\left\{\frac{\partial Q^*}{\partial\beta_k}\right\} \tag{11}$$

The necessity of evaluating the vector $\{\partial Q^*/\partial\beta_k\}$ using Eq. (7) is eliminated for all β_k by selecting the vector $\{\lambda_j\}$ such that the coefficient of $\{\partial Q^*/\partial\beta_k\}$ in Eq. (11) is null. That is, selection of $\{\lambda_j\}$ that satisfies

$$\left\{\frac{\partial C_j}{\partial Q}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial Q}\right] = \{0\}^T \tag{12}$$

implies

$$-\left[\frac{\partial R}{\partial Q}\right]\{\lambda_j\} = \left\{\frac{\partial C_j}{\partial Q}\right\} \tag{13}$$

Therefore, following the solution of Eq. (13) for this particular choice of the adjoint variable vector $\{\lambda_j\}$, the sensitivity derivatives of C_j with respect to all β_k are computed by

$$\frac{dC_j}{d\beta_k} = \left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{\partial \bar{X}}{\partial\beta_k}\right\} + \frac{\partial C_j}{\partial\beta_k} + \{\lambda_j\}^T\left\{\frac{\partial R}{\partial\beta_k}\right\} \tag{14}$$

Solution of the linear system of Eq. (13) for $\{\lambda_j\}$ is analogous to the solution of Eq. (7) for $\{\partial Q^*/\partial\beta_k\}$ in that the coefficient matrices are transposes of each other. A particular solution, $\{\lambda_j\}$, is valid only for a specific system response, C_j; thus, solution of Eq. (13) must be repeated for each different system response of interest. If Eq. (13) is solved directly, however, multiple solutions require only a single LU factorization of the coefficient matrix, which is repeatedly used for an unlimited number of right-hand side vectors, $\{\partial C_j/\partial Q\}$ (i.e., for an unlimited number of different system responses of interest).

It is simple to verify from the preceding equations, and it should be noted that each solution, $\{\partial Q^*/\partial\beta_k\}$, of Eq. (7) for a particular design variable can be used for an unlimited number of different system responses. In contrast, however, each solution $\{\lambda_j\}$ of Eq. (13) for a particular system response can be used for an unlimited number of different design variables. Therefore, in terms of computational work, if the number of system responses of interest is larger than the number of design variables, then sensitivity derivatives should be computed by solving Eq. (7). Otherwise, greater computational efficiency is obtained using the adjoint variable method. Despite the difference that has been noted between these two closely related procedures, it is emphasized that the two methods are *equivalent* in the sense that they will yield identical values (except for the machine truncation error) for the sensitivity derivatives, if properly implemented computationally. In the present study both methods have been successfully implemented in the example problems and have been verified to yield identical results (as expected).

The significance of the well-known difference in the computational efficiency of the two methods is greatly mitigated if a direct method is used to solve the linear systems [i.e., either Eq. (7) or Eq. (13)] through the previously mentioned highly efficient strategy of reusing the LU factored coefficient matrix for multiple right-hand side vectors. However, this distinction becomes very important if an iterative strategy is used to solve these linear systems, particularly if the difference between the number of design variables and the number of system responses of interest is very large. This, of course, is because with iterative methods, the computational work required for solution of each linear system (following the first solution) is approximately equal to the computational work required to solve the first one. Extension of the present methodology to full-scale problems in three dimensions on the supercomputers presently available will require the use of iterative methods to solve Eqs. (7) or (13) because of their size and the limitations in computer storage.

Grid Sensitivity Terms

In this section attention will be focused on the significance and the treatment of the vector $\{\partial\bar{X}/\partial\beta_k\}$, which must be evaluated on the right-hand side of the sensitivity equations, when geometric shape design variables are considered. Of central importance to note in the subsequent discussion is that these grid sensitivity terms are distinctly separate from the governing equations of fluid mechanics and are instead closely linked to the chosen system of generating the computational mesh on which the equations are solved. Therefore, these terms can be treated separately. Note that henceforth only geometric shape sensitivity derivatives are considered here; thus, in the subsequent discussion, $\bar{\beta}$ is taken to be a vector of only geometric shape design variables.

The geometric shape of the computational domain is defined completely by the (x, y) coordinates on the boundaries of the computational mesh. The vector of all (x, y) coordinates on the boundaries (or portions of boundaries) of interest, $\bar{X}_b$, is a relatively small subset of the much larger global vector, $\{\bar{X}\}$. In principle, each x and y coordinate of $\bar{X}_b$ could be taken to be a design variable. This would mean that $\bar{\beta}$ is defined to be equal to $\bar{X}_b$. With this approach, sensitivity derivatives are taken with respect to each individual x and y coordinate of $\bar{X}_b$. An important concern with this, however, is that the number of geometric shape design variables might likely be unmanageably large when the sensitivity derivatives are subsequently applied in a design setting. For this reason a second method is adopted here, the motivation being to significantly reduce the number of design variables. With this approach $\bar{\beta}$ is *not* taken to be equal to $\bar{X}_b$, but rather the assumption is made that the geometric shape of the boundaries of the domain can be defined a priori by relationships that involve a relatively small number of geometric shape (design) parameters. In short, it is assumed that the (x, y) coordinates of interest on the boundaries of the domain can be represented parametrically; i.e.,

$$\bar{X}_b = \bar{X}_b(\bar{\beta}) \tag{15}$$

Regardless of whether the coordinates on the boundary are taken to be the design variables, or whether these coordinates are represented parametrically in terms of the design variables, the methods described subsequently for computing the grid sensitivity terms are applicable.

A very general and conceptually simple method for computing the grid sensitivity terms is to make use of the grid generation program in approximating these terms by brute force finite differences. With this approach the geometric shape design parameters $\bar{\beta}$ are perturbed slightly, one at a time, and the mesh generation program (which was used to generate the initial mesh) is run repeatedly to generate perturbed grids. Then each grid sensitivity vector is calculated as

$$\left\{\frac{\partial\bar{X}}{\partial\beta_k}\right\} \approx \frac{\{\bar{X}(\beta_k + \Delta\beta_k)\} - \{\bar{X}(\beta_k - \Delta\beta_k)\}}{2\Delta\beta_k} \tag{16}$$

where, in the preceding equation, central finite difference approximations are used for greater accuracy, but require the generation of two perturbed meshes per design variable instead of the one, which is required when using forward finite differences. In using finite difference approximations to represent the elements of $\{\partial\bar{X}/\partial\beta_k\}$, it is suggested that the overall analytical nature of the fundamental sensitivity equations [e.g., Eq. (7)] is not severely diminished. Furthermore, it is believed that very accurate finite difference approximations of $\{\partial\bar{X}/\partial\beta_k\}$ can be obtained reliably and efficiently without introducing some of the difficulties that might be associated with using the brute force method on the fluids equations. This is because typically the equations of mesh generation are by design "smoother" than the equations of fluid flow and are far more efficient to solve.

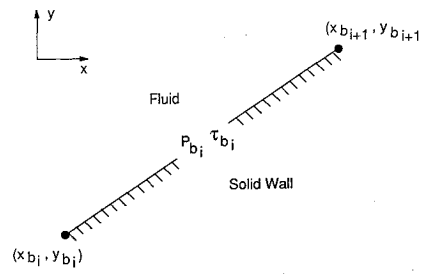

Fig. 1 Illustration of the *i*th element on a boundary.

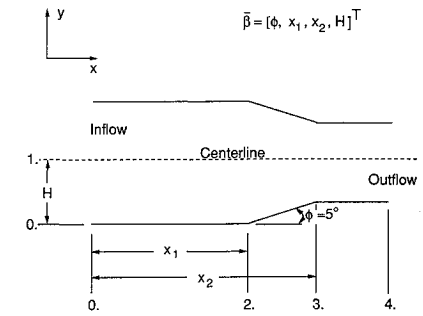

Fig. 2 Subsonic nozzle/supersonic inlet test geometry.

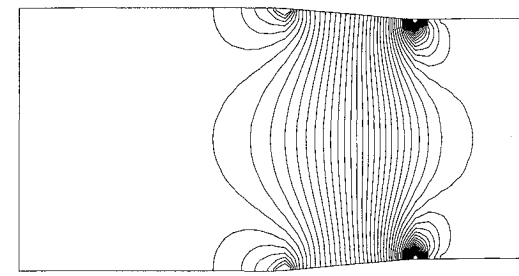

Fig. 3 Pressure contours, subsonic nozzle ($M_\infty = 0.85$) example problem.

As an alternative to the finite difference approach for computing the grid sensitivity terms, an exact analytical approach could be considered. To accomplish this, it is first necessary to determine analytical expressions to represent the "rules" that are built into and are inherent in the specific mesh generation scheme of choice by which the elements of $\{\bar{X}\}$ are generated. Of course, these expressions are written explicitly in terms of $\bar{\beta}$, so that all the terms of each vector $\{\partial\bar{X}/\partial\beta_k\}$ can be computed by direct differentiation of these expressions. The exact analytical evaluation of the grid sensitivity terms is implemented (as outlined in the preceding discussion) for the specific mesh generation strategy and for the specific elements of $\bar{\beta}$ that are associated with the example problems to be presented, and as a consistency check, these results are also successfully verified using the finite difference approach. The expected result is achieved that, as the finite difference of the independent variable (i.e., $\Delta\beta_k$) is made progressively smaller, there becomes no difference between the analytical and the finite difference methods in the computed values of $\{\partial\bar{X}/\partial\beta_k\}$. (This consistency check is done for each term of $\bar{\beta}$.)

The analytical method outlined in the preceding discussion for evaluating $\{\partial\bar{X}/\partial\beta_k\}$, although actually implemented in the present study (in order to demonstrate the method), has some significant shortcomings. First, the method lacks generality. That is, analytical expressions that represent $\{\bar{X}\}$ as an explicit function of $\bar{\beta}$ would have to be worked out for each

different mesh generation program and, more importantly, for each new flow problem having a different set of geometric shape design variables (i.e., for each new $\bar{\beta}$). Furthermore, for complex mesh generation programs, representing the rules by which grid points are distributed throughout the domain as analytical functions and differentiating them is likely to be very complicated. It is postulated that some of these difficulties that will probably be encountered in analytically computing the grid sensitivity terms could be mitigated through a strategy that employs a judicious use of the chain rule in computing these terms. Although this proposed strategy is not actually implemented in the present study, it is outlined and discussed subsequently in the remainder of this section.

Typically the computational mesh, $\{\bar{X}\}$, for a given problem is obtained as the principal output of a mesh generating program, where the principal input to the program are the coordinates of the boundaries of the domain, $\bar{X}_b$. (Additional input to the grid generating program might be the vector of parameters $\bar{\gamma}$, which exert some control over the placement of points in the interior of the domain, e.g., grid stretching near walls, but $\bar{\gamma}$ is usually not functionally dependent on $\bar{\beta}$.) Therefore, assuming a parametric representation of $\bar{X}_b$ in terms of $\bar{\beta}$, the operation of a grid generation program might be expressed as

$$\bar{X} = \bar{X}[\bar{X}_b(\bar{\beta}), \bar{\gamma}] \tag{17}$$

where $\bar{\gamma} \neq \bar{\gamma}(\bar{\beta})$. Direct differentiation of Eq. (17) using the chain rule yields

$$\left\{\frac{\partial \bar{X}}{\partial \beta_k}\right\} = \left[\frac{\partial \bar{X}}{\partial \bar{X}_b}\right]\left\{\frac{\partial \bar{X}_b}{\partial \beta_k}\right\} \tag{18}$$

The principal advantage of the approach represented by Eq. (18) is increased generality. For each completely new flow problem, the only terms that must be worked out for the specific problem at hand (either by direct differentiation of analytical expressions that are unique to the given problem or by the method of finite differences) is the vector $\{\partial \bar{X}_b/\partial \beta_k\}$, a relatively small subset of $\{\partial \bar{X}/\partial \beta_k\}$.

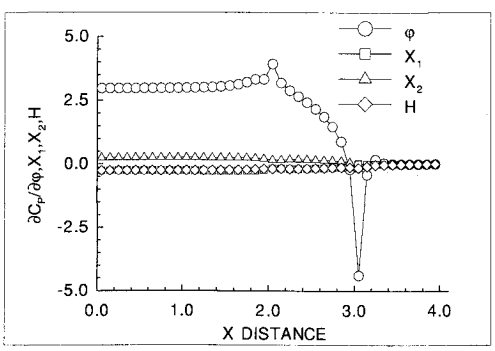

Fig. 4 Pressure coefficient sensitivity derivatives, lower wall, subsonic nozzle problem.

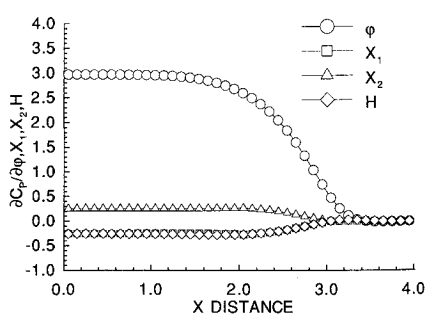

Fig. 5 Pressure coefficient sensitivity derivatives, centerline, subsonic nozzle problem.

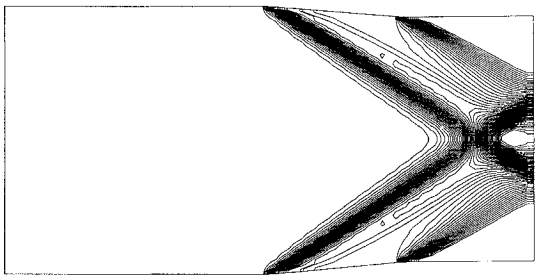

Fig. 6 Pressure contours, supersonic inlet ($M_\infty = 2.0$) example problem.

The global Jacobian matrix $[\partial \bar{X}/\partial \bar{X}_b]$ is strictly associated with the functional relationships that are used to distribute the (x, y) coordinates throughout the entire computational mesh and are built into the specific mesh generation program of choice. This matrix $[\partial \bar{X}/\partial \bar{X}_b]$, which is associated with a particular set of grid generation rules, is analogous to and similar in nature to the global Jacobian matrices $[\partial R/\partial Q]$ and $[\partial R/\partial \bar{X}]$ (discussed previously), where these two matrices are associated with the specific numerical approximation of choice used to model the governing fluids equations (i.e., the Euler equations). For this reason, if an analytical approach is chosen for evaluating the terms of $[\partial \bar{X}/\partial \bar{X}_b]$, it would only be necessary to work out these derivatives analytically one time for the given mesh generation program of choice, and the results could be included as a "sensitivity module" within the grid generation program itself. Alternatively, the terms of this matrix could be approximated by finite differences and evaluated within the mesh generation program.

Ancillary Sensitivity Relationships

The purpose of this section is to present some additional terms and specific relationships of the type given by Eq. (9) that are used to compute sensitivity derivatives with respect to geometric shape in the example problems. Specifically, expressions are given for aerodynamic force coefficients in two dimensions and their sensitivity derivatives. In addition, terms developed in this section are also useful in the construction of the right-hand sides of Eqs. (13) and (14), of the adjoint variable approach.

Figure 1 illustrates the ith element (oriented at an arbitrary angle in space), which is located on the boundary of the geometric shape of interest, over/through which the fluid is passing. In the figure the coordinates (x_{b_i}, y_{b_i}) and $(x_{b_{i+1}}, y_{b_{i+1}})$ are the physical (x, y) coordinates at either end of this ith element and are assumed to be nondimensionalized by L_{ref}, a reference length. The convention is established that, as one moves along the surface in the direction of increasing the index i, then the solid surface is on the right, and the fluid is on the left. Also noted in the figure are the pressure and shear stress, P_{b_i} and τ_{b_i}, respectively, which are associated with this ith element on the boundary (where the shear stress is, of course, zero in the present study, since only inviscid flow is considered here).

A nondimensional pressure coefficient C_{p_i}, which is associated with this ith element on the boundary is defined as follows:

$$C_{p_i} = P_{b_i}/(\tfrac{1}{2}\rho_\infty U_\infty^2) \tag{19}$$

where $\frac{1}{2}\rho_\infty U_\infty^2$ is the dynamic pressure of the freestream. Note that the conventional definition of pressure coefficient as a pressure difference with the freestream pressure has not been adopted. Using the (x, y) coordinate system established in

Fig. 1, nondimensional force coefficients in the x and y directions, C_{x_i} and C_{y_i}, respectively, for the ith surface element, are given as

$$C_{x_i} = C_{p_i}(y_{b_{i+1}} - y_{b_i})$$
$$C_{y_i} = C_{p_i}(x_{b_i} - x_{b_{i+1}}) \tag{20}$$

The total force coefficients in the x and y directions are given by summing Eqs. (20) over the total number of elements of interest, NE. The result is

$$C_x = \sum_{i=1}^{NE} C_{x_i}$$
$$C_y = \sum_{i=1}^{NE} C_{y_i} \tag{21}$$

Sensitivity derivatives of these force coefficients taken with respect to β_k are given as

$$\frac{dC_x}{d\beta_k} = \sum_{i=1}^{NE} \left\{ \frac{\partial C_{p_i}}{\partial \beta_k}(y_{b_{i+1}} - y_{b_i}) + C_{p_i}\left(\frac{\partial y_{b_{i+1}}}{\partial \beta_k} - \frac{\partial y_{b_i}}{\partial \beta_k} \right) \right\}$$
$$\frac{dC_y}{d\beta_k} = \sum_{i=1}^{NE} \left\{ \frac{\partial C_{p_i}}{\partial \beta_k}(x_{b_i} - x_{b_{i+1}}) + C_{p_i}\left(\frac{\partial x_{b_i}}{\partial \beta_k} - \frac{\partial x_{b_{i+1}}}{\partial \beta_k} \right) \right\} \tag{22}$$

Note in expressions (22) that terms such as $\partial x_{b_i}/\partial \beta_k$ and $\partial y_{b_i}/\partial \beta_k$ are evaluated as being elements of the global grid sensitivity vector $\{\partial \bar{X}/\partial \beta_k\}$ and that the terms $\partial C_{p_i}/\partial \beta_k$ (i.e., the pressure sensitivity derivatives) are obtained from the global vector $\{\partial Q^*/\partial \beta_k\}$ following the solution of Eq. (7). This is accomplished using the chain rule, noting that pressure is a function of the conserved variables Q through the ideal gas law. Finally, note that for airfoil calculations the sensitivity derivatives of the lift C_L and drag C_D coefficients could be calculated using simple combinations of these expressions [i.e., Eqs. (22)], since C_L and C_D are simple functions of C_x, C_y, and angle of attack.

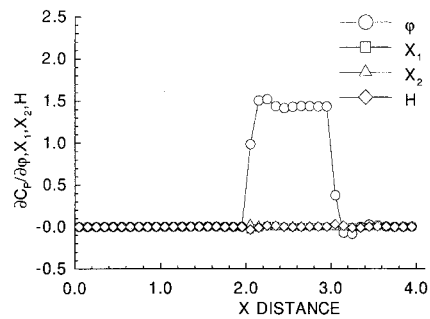

Fig. 7 Pressure coefficient sensitivity derivatives, lower wall, supersonic inlet problem.

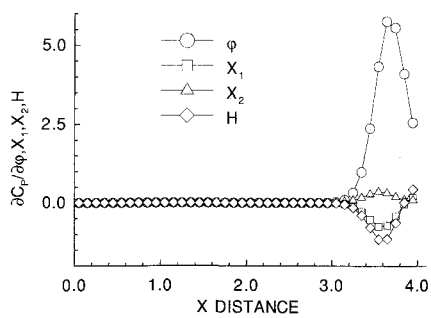

Fig. 8 Pressure coefficient sensitivity derivatives, centerline, supersonic inlet problem.

Computational Results

Description of the Example Problems

There are two example problems that are investigated in this study: 1) a subsonic nozzle ($M_\infty = 0.85$) and 2) a supersonic inlet ($M_\infty = 2.0$). The geometry of the two problems is identical and is illustrated in Fig. 2. The only difference between these two example problems is found in the inflow/outflow boundary conditions, to be discussed later.

There are four geometric shape (design) parameters to be considered in each of the two example problems. These four parameters, illustrated in Fig. 2, are therefore the elements of the vector $\bar{\beta}$, given as

$$\bar{\beta} = [\phi, x_1, x_2, H]^T \tag{23}$$

where ϕ is the angle of the inclined ramp, x_1 is the x coordinate of the start of the inclined ramp, x_2 is the x coordinate of the end of the ramp, and H (a reference length) is one-half of the height of the entrance to the nozzle/inlet.

In both the $M_\infty = 0.85$ subsonic nozzle and the $M_\infty = 2.0$ supersonic inlet, sensitivity derivatives are taken about a baseline steady-state solution, where the geometric parameters that define the shape of the nozzle/inlet are taken to be $\phi = 5.0$ deg, $x_1 = 2.0$, $x_2 = 3.0$, and $H = 1.0$. The length of the nozzle/inlet (an additional geometric shape parameter that is not considered with respect to sensitivity derivatives in the present study) is taken to be 4.0.

The centerline of the nozzle/inlet is a plane of symmetry; thus, it is only necessary to compute either the lower or upper half of the flowfield (the lower half was chosen in the present work). A computational grid with 41 grid lines in the streamwise direction, and 31 grid lines in the normal direction is chosen. This 41 × 31 point mesh is generated under the following set of rules for the distribution of grid points throughout the domain:

1) At a fixed x location, the 31 horizontal grid lines are equally spaced.

2) Between $x = 0$ and $x = x_1$, the vertical grid lines are evenly spaced, with a total of 20 vertical columns of cells between these two x stations.

3) Between $x = x_1$ and $x_1 = x_2$, the vertical grid lines are evenly spaced, with a total of 10 vertical columns of cells between these two x stations.

4) Between $x = x_2$ and the last vertical grid line at $x = 4.0$, the vertical grid lines are evenly spaced, with a total of 10 vertical columns of cells between these two x stations.

These preceding four rules of grid generation are expressed as functional relationships explicitly in terms of $\bar{\beta}$, and derivatives are taken of these resulting expressions with respect to each element of $\bar{\beta}$, so that all elements of the grid sensitivity vector $\{\partial \bar{X}/\partial \beta_k\}$ can be evaluated analytically (for each of the four design variables). The details of these expressions and their derivatives are omitted here because they lack generality (i.e., they are very specific to the particular example geometry and mesh generation rules of the present study). As noted earlier, the computed values of all the elements of the vector $\{\partial \bar{X}/\partial \beta_k\}$, using this analytical approach, are verified by the more general procedure of using finite difference approximations to represent these grid sensitivity terms, where, as expected, the agreement in the results obtained using these two methods is essentially perfect.

Subsonic Nozzle Example Results

A steady-state conventional numerical solution to the Euler equations is obtained for the $M_\infty = 0.85$ subsonic nozzle for the previously described baseline geometry, where the L_2 norm of the residual is reduced to machine zero (approximately a 12-order-of-magnitude reduction) using the AF time integration algorithm.[18] The pressure contours for this steady-

state baseline solution are shown in Fig. 3. Boundary conditions are specified in this example problem as follows:

1) On the lower wall boundary, flow tangency is enforced.

2) On the upper boundary of the computational domain (i.e., on the centerline of the nozzle), flow symmetry is enforced. (Note that, in reality, the boundary conditions that are enforced on the centerline are identical to those of the lower wall).

3) On the inflow boundary, both entropy and stagnation enthalpy are held constant to be that of the freestream, and the v component of velocity is fixed at zero. The static pressure is extrapolated.

4) On the outflow boundary, density as well as both components of velocity are extrapolated. The static pressure is held constant and is specified to be that of the freestream (i.e., P/P_∞ is specified to be 1.0).

Equation (7) is solved directly four times (using a single LU factorization of the coefficient matrix followed by four forward and backward substitutions), where for each solution a different grid sensitivity vector $\{\partial \bar{X}/\partial \beta_k\}$ is used on the right-hand side of the linear system (one for each of the four design variables). These direct solutions of Eq. (7) are obtained using a conventional vectorized banded matrix solver that takes advantage of the fact (in terms of computational work and computer storage) that, outside of the main bandwidth, all of the elements are zero.[17] Each of the four resulting solution vectors, $\{\partial Q^*/\partial \phi\}$, $\{\partial Q^*/\partial x_1\}$, $\{\partial Q^*/\partial x_2\}$, and $\{\partial Q^*/\partial H\}$, represents the sensitivities of the complete vector of field variables, with derivatives taken with respect to each of the four respective geometric shape parameters. From each of these four global solution vectors, a specific subset of information of interest is extracted: the subset of the sensitivity derivatives of C_p (i.e., $\partial C_p/\partial \phi$, $\partial C_p/\partial x_1$, $\partial C_p/\partial x_2$, and $\partial C_p/\partial H$) along the lower wall and also the centerline of the nozzle. As mentioned previously, pressure sensitivities are computed from the sensitivity derivatives of the conserved field variables using the ideal gas law and the chain rule.

In addition to the computation of pressure coefficient sensitivities by the solution of Eq. (7), for validation and comparison purposes, this information is computed by the use of finite difference approximations. Specifically, central difference representations of these pressure sensitivities are obtained by computing two conventional steady-state numerical solutions to the Euler equations (where in each case the L_2 norm of the residual is again reduced to machine zero), where these two steady-state solutions are centered across the baseline steady-state solution, and in all cases the incremental change in the independent variable (i.e., $\Delta\phi$, Δx_1, Δx_2, and ΔH) is taken to be ± 0.00001. The baseline solution is used as the initial guess during startup in generating these perturbed solutions.

Figure 4 is an illustration of the pressure coefficient sensitivity derivatives obtained from the four direct solutions of Eq. (7). In this figure the sensitivity derivatives of the pressure coefficient with respect to each of the four design variables along the lower wall of the nozzle are plotted vs x. Figure 5 is similar to Fig. 4, except that pressure sensitivities along the centerline of the nozzle are plotted vs x. Omitted for the sake of keeping these figures uncluttered is the comparison with the results of the computations obtained using the brute force finite difference method, where, in all cases, the agreement in the results obtained using the two methods is essentially perfect. In Ref. 8, where these same results are presented using one figure for each variable, this comparison is shown in detail.

Following these direct solutions of Eq. (7), Eqs. (22) are used to compute sensitivity derivatives of the aerodynamic force coefficients, C_x and C_y, with respect to the parameters ϕ, x_1, x_2, and H for the lower wall and then for the centerline of the nozzle. (Of course, there is no *net* aerodynamic force in either direction at the centerline of the nozzle; nevertheless, these force coefficients are calculated as if the centerline were a solid wall.) In addition, these sensitivity derivatives are computed using the adjoint variable approach [i.e., using equations of the type given by Eq. (14), after solving Eq. (13)

Table 1 Force coefficient sensitivities, lower surface, subsonic nozzle problem

	$\frac{dC_x}{d\phi, x_1, x_2, H}$		$\frac{dC_y}{d\phi, x_1, x_2, H}$	
Sensitivities	Direct differentiation	Finite difference	Direct differentiation	Finite difference
$\frac{dC_x, C_y}{d\phi}$	2.4140	2.4140	−7.7588	−7.7588
$\frac{dC_x, C_y}{dx_1}$	−0.2072	−0.2072	0.5232	0.5232
$\frac{dC_x, C_y}{dx_2}$	0.2069	0.2069	−0.8943	−0.8943
$\frac{dC_x, C_y}{dH}$	−0.0143	−0.0143	0.6841	0.6841

Table 2 Force coefficient sensitivities, centerline, subsonic nozzle problem

	$\frac{dC_x}{d\phi, x_1, x_2, H}$		$\frac{dC_y}{d\phi, x_1, x_2, H}$	
Sensitivities	Direct differentiation	Finite difference	Direct differentiation	Finite difference
$\frac{dC_x, C_y}{d\phi}$	0.0000	0.0000	7.7582	7.7582
$\frac{dC_x, C_y}{dx_1}$	0.0000	0.0000	−0.5232	−0.5232
$\frac{dC_x, C_y}{dx_2}$	0.0000	0.0000	0.8942	0.8942
$\frac{dC_x, C_y}{dH}$	0.0000	0.0000	−0.6843	−0.6843

Table 3 Force coefficient sensitivities, lower surface, supersonic inlet problem

	$\frac{dC_x}{d\phi, x_1, x_2, H}$		$\frac{dC_y}{d\phi, x_1, x_2, H}$	
Sensitivities	Direct differentiation	Finite difference	Direct differentiation	Finite difference
$\frac{dC_x, C_y}{d\phi}$	0.5936	0.5936	−1.4295	−1.4295
$\frac{dC_x, C_y}{dx_1}$	−0.0411	−0.0411	0.1127	0.1127
$\frac{dC_x, C_y}{dx_2}$	0.0411	0.0411	−0.1124	−0.1124
$\frac{dC_x, C_y}{dH}$	−0.000239	−0.000239	0.000729	0.000729

Table 4 Force coefficient sensitivities, centerline, supersonic inlet problem

	$\frac{dC_x}{d\phi, x_1, x_2, H}$		$\frac{dC_y}{d\phi, x_1, x_2, H}$	
Sensitivities	Direct differentiation	Finite difference	Direct differentiation	Finite difference
$\frac{dC_x, C_y}{d\phi}$	0.0000	0.0000	2.5978	2.5978
$\frac{dC_x, C_y}{dx_1}$	0.0000	0.0000	−0.2766	−0.2766
$\frac{dC_x, C_y}{dx_2}$	0.0000	0.0000	0.0305	0.0305
$\frac{dC_x, C_y}{dH}$	0.0000	0.0000	−0.3932	−0.3932

directly]. As expected, perfect agreement (except for very small machine truncation error) is seen in the results obtained using these two methods; therefore, the identical results of both of these methods are presented and referred to as the single method of direct differentiation in Table 1 for the lower wall and Table 2 for the centerline. Also presented in these tables are the results that are obtained for these sensitivity derivatives using the finite difference approach, where clearly the agreement is excellent with the results of the method of direct differentiation.

Supersonic Inlet Example Problem

A steady-state conventional numerical solution to the Euler equations is obtained for the $M_\infty = 2.0$ supersonic inlet for the previously described baseline geometry, where the L_2 norm of the residual is reduced to machine zero using the vertical line Gauss-Seidel (VLGS) relaxation algorithm.[13-15] The pressure contours for this steady-state baseline solution are shown in Fig. 6. Boundary conditions are specified in this example problem as follows:

1) On the inflow boundary, all variables are held fixed, and specified to be that of the freestream.

2) On the outflow boundary, all variables are extrapolated.

3) Boundary conditions on the upper and lower surface of the computational domain are the same as those for the subsonic example problem.

In the special case (which is applicable in this example) of a fully supersonic inviscid flow where a fully upwind flux balance is used in the streamwise direction, a direct solution of Eq. (7) is accomplished very efficiently using a single sweep of the VLGS algorithm in the positive streamwise direction. (A full explanation of this is found in Ref. 15.) If the adjoint variable approach is preferred, direct solution of Eq. (13) can be obtained using a single reverse sweep of this algorithm.

The results illustrated in Figs. 7 and 8 for the present supersonic example are similar to the results that have been previously discussed in greater detail and illustrated in Figs. 4 and 5, respectively, for the subsonic example. In addition, Tables 3 and 4 for the supersonic example show similar results to those shown in Tables 1 and 2, respectively, of the subsonic example. In particular, note in Tables 3 and 4 that the agreement is again excellent in the results when the method of direct differentiation is compared with the finite difference approach.

Discussion

The sensitivity derivatives that are calculated and presented previously are generally consistent with the nature of the flow physics of the chosen example problems. However, there are exceptions that should be noted. The most obvious example of calculated sensitivity derivatives that are inconsistent with the flow physics are dC_x/dH and dC_y/dH of the lower surface of the supersonic example problem. (See the last entry of Table 3.) Although these computed aerodynamic sensitivity coefficients are relatively small, yet nonzero, they should, in fact, be exactly zero to be perfectly consistent with the flow physics. That is, changing the height of this inlet should not affect the flowfield variables anywhere along the lower surface of the inlet (for an exact solution to the Euler equations), because for this example geometry the shock reflection off of the centerline does not strike the lower surface. The aforementioned inconsistency is attributed to the discretization error associated with the discrete numerical model which approximates the Euler equations. This explanation is supported by a grid refinement study in which when the number of cells is doubled in both directions, the phenomenon is found to become much smaller.

CPU Time

It is difficult to quantify precisely the relative computational efficiency of the present methodology compared to the finite difference method of computing aerodynamic sensitivity derivatives. This is because the CPU time that is used in implementing the finite difference technique can vary greatly depending on a number of factors, including

1) the use of central vs the use of forward (or backward) difference approximations (the former being far more accurate, at twice the computational cost of the latter);

2) the strictness of the convergence criterion used to find steady-state solutions to the Euler equations (for subsequent use in computing the sensitivity derivatives by finite differences);

3) the efficiency of the algorithm used to find steady-state solutions to the Euler equations;

4) the type of initial guess used to obtain the neighboring steady-state solutions to the Euler equations; and

5) the particular flow problem under consideration.

Despite these difficulties in assessing the relative computational efficiency, it is noted that the sensitivity derivatives that are calculated in the subsonic example problem using the method of brute force finite differences are about 47 times more costly to generate (in terms of total CPU time) than the results that are generated by direct solution of Eq. (7). In the supersonic example the finite difference approach is only about nine times more costly, which is attributed to the fact that the CFD solver is very much more efficient in this special case of fully supersonic inviscid flow. These CPU time comparisons with the finite difference method do not change significantly by use of the adjoint variable method in computing the sensitivity derivatives shown in Tables 1-4, since in each of the two example problems the number of system responses of interest (i.e., C_x and C_y on the lower wall and centerline) is equal to the number of design variables considered. The computational work that is used in generating the results using the finite difference approach should be taken as the "worst case" situation, however, considering the strict manner in which this method is applied in the present study, and could probably be much less costly in a practical design application.

Summary and Conclusions

A methodology of a general nature is developed for computing aerodynamic sensitivity derivatives, where the procedures include the full capability of computing these derivatives with respect to geometric shape design variables on "body-oriented" computational grids. The methods are successfully demonstrated in two dimensions for the Euler equations using an upwind cell-centered finite volume approach, where sensitivity derivatives are computed by direct differentiation of the nonlinear algebraic equations that model the Euler equations.

The procedures are applied in calculating the sensitivity derivatives of the aerodynamic loads on the interior walls of a subsonic nozzle ($M_\infty = 0.85$) and also a supersonic inlet ($M_\infty = 2.0$) with respect to the parameters that define the geometric shape of these devices. The computed results obtained using these techniques compare essentially perfectly in all cases with the computed results obtained through use of the method of brute force (finite differences). In addition, the present methodology is observed to be significantly more computationally efficient in the example problems than the method of finite differences (although a precise comparison of the relative computational efficiency is difficult to quantify).

Acknowledgments

This research was supported in part by Grant DMC-865-7917 from the National Science Foundation and in part by Grant NAG-1-1265 from NASA Langley Research Center, with Henry E. Jones serving as the technical monitor for the latter grant. The authors thank Perry A. Newman and E. Carson Yates, Jr., of NASA Langley Research Center for the many helpful discussions that contributed to this work.

References

[1]Yates, E. C., Jr., and Desmarais, R., "Boundary Integral Method for Calculating Aerodynamic Sensitivities with Illustration for Lifting Surface Theory," *Proceedings of the International Symposium of Boundary Element Methods (IBEM 89),* Springer-Verlag, New York, Oct. 2-4, 1989.

[2]Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients in the Transonic and Supersonic Regimes," *Journal of Aircraft,* Vol. 27, No. 6, 1990, pp. 507-518; also AIAA Paper 89-0532, 1989.

[3]Yates, E. C., Jr., "Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady, Nonplanar Lifting-Surface Theory," NASA TM-100502, Sept. 1987.

[4]Sobieszczanski-Sobieski, J., "The Case For Aerodynamic Sensitivity Analysis," *Sensitivity Analysis in Engineering,* NASA CP-2457, Feb. 1981.

[5]Bristow, D. R., and Hawk, J. D., "Subsonic Panel Method For The Efficient Analysis of Multiple Geometry Perturbations," NASA CR-3528, March 1982.

[6]Jameson, A., "Aerodynamic Design Via Control Theory," NASA CR-181749 and Institute for Computer Applications in Science and Engineering Rept. 88-64, Nov. 1988.

[7]Taylor, A. C., III, Korivi, V. M., and Hou, G. W., "Sensitivity Analysis Applied to the Euler Equations: A Feasibility Study with Emphasis on Variation of Geometric Shape," AIAA Paper 91-0173, Jan. 1991.

[8]Taylor, A. C., III, Hou, G. W., and Korivi, V. M., "A Methodology for Determining Aerodynamic Sensitivity Derivatives With Respect to Variation of Geometric Shape," AIAA Paper 91-1101, *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference,* Baltimore, MD, April 8-10, 1991.

[9]Walters, R. W., and Thomas, J. L., "Advances in Upwind Relaxation Methods," *State of the Art Surveys of Computational Mechanics,* edited by A. K. Noor, American Society of Mechanical Engineers, New York, 1989, pp. 145-183.

[10]Thomas, J. L., Van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," *AIAA Journal,* Vol. 28, No. 6, 1990, pp. 973-974; also AIAA Paper 85-1680, July 1985.

[11]Walters, R. W., and Dwoyer, D. L., "An Efficient Iteration Strategy for the Solution of the Euler Equations," AIAA Paper 85-1529, July 1985.

[12]Newsome, R. W., Walters, R. W., and Thomas, J. L., "An Efficient Iteration Strategy for the Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," *AIAA Journal,* Vol. 27, No. 9, 1989, pp. 1165-1166; also AIAA Paper 87-1113, June 1987.

[13]Thomas, J. L., and Walters, R. W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," *AIAA Journal,* Vol. 25, No. 4, 1987, pp. 527-534.

[14]Napolitano, M., and Walters, R. W., "An Incremental Block-Line-Gauss-Seidel Method for the Navier-Stokes Equations," *AIAA Journal,* Vol. 24, No. 5, 1986, pp. 770-776.

[15]Walters, R. W., and Dwoyer, D. L., "Efficient Solutions to the Euler Equations for Supersonic Flow with Embedded Subsonic Regions," NASA TP 2523, Jan. 1987.

[16]Van Leer, B., "Flux-Vector Splitting for the Euler Equations," Institute for Computer Applications in Science and Engineering Rept. 82-30, Sept. 1982; also *Lecture Notes in Physics,* Vol. 170, 1982, pp. 507-512.

[17]Riggins, D. W., Walters, R. W., and Pelletier, D., "The Use of Direct Solvers for Compressible Flow Computations," AIAA Paper 88-0229, Jan. 1988.

[18]Beam, R. M., and Warming, R. F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal,* Vol. 16, No. 4, 1978, pp. 393-402.

[19]Walters, R. W., Dwoyer, D. L., and Hassan, H. A., "A Strongly Implicit Procedure for the Compressible Navier Stokes Equations," *AIAA Journal,* Vol. 24, No. 1, 1986, pp. 6-12.

[20]Venkatakrishnan, V., "Preconditioned Conjugate Gradient Method For The Compressible Navier-Stokes Equations," *AIAA Journal,* Vol. 29, No. 7, 1991, pp. 1092-1100; also AIAA Paper 90-0586, Jan. 1990.