JOURNAL OF COMPUTATIONAL PHYSICS 113, 336–346 (1994)

An Approximately Factored Incremental Strategy for Calculating Consistent Discrete Aerodynamic Sensitivity Derivatives

VAMSHI MOHAN KORIVI AND ARTHUR C. TAYLOR III

Department of Mechanical Engineering, Old Dominion University, Norfolk, Virginia 23529-0247

PERRY A. NEWMAN

Computational Sciences Branch, NASA Langley Research Center, Hampton, Virginia 23665-5225

GENE W. HOU

Department of Mechanical Engineering, Old Dominion University, Norfolk, Virginia 23529-0247

AND

HENRY E. JONES

U.S. Army Aviation & Troop Command, Computational Sciences Branch, NASA Langley Research Center, Hampton, Virginia 23665-5225

Received September 30, 1992; revised February 2, 1994

In this study involving advanced fluid flow codes, an incremental iterative formulation (also known as the "delta" or "correction" form), together with the well-known spatially split approximate factorization algorithm, is presented for solving the very large sparse systems of linear equations which are associated with aerodynamic sensitivity analysis. For smaller 2D problems a direct method can be applied to solve these linear equations in either the standard or the incremental form, in which case the two are equivalent. Iterative methods are needed for larger 2D and future 3D applications, however, because direct methods require much more computer memory than is currently available. Iterative methods for solving these equations in the standard form are generally unsatisfactory due to a lack of diagonal dominance and perhaps ill-conditioning of the coefficient matrix; this problem can be overcome when these equations are cast in the incremental form. These and other benefits are discussed herein. The methodology is successfully implemented and tested in 2D using an upwind, cell-centered, finite volume formulation applied to the thin-layer Navier–Stokes equations. Results are presented for two sample airfoil problems: (1) subsonic low Reynolds number laminar flow; and (2) transonic high Reynolds number turbulent flow. © 1994 Academic Press, Inc.

1. INTRODUCTION

For many complex flow fields of interest in practical engineering problems, accurate detailed analyses are now possible using supercomputers and advanced software; these codes have been developed in recent years through an intensive research effort focused in the discipline now known as computational fluid dynamics (CFD). For these advanced CFD codes to become more useful as practical *design* tools, additional software is needed which will efficiently provide accurate aerodynamic sensitivity derivatives which are consistent with the discrete flow solutions of the particular CFD code of choice. The theme of this study is the ongoing development of a methodology for calculating these derivatives.

A sensitivity derivative is defined as the derivative of a system response of interest (e.g., the lift or drag of an airfoil) with respect to an independent design variable of interest (e.g., a parameter which controls the shape of an airfoil). In a typical design environment, a very large number of analyses are often made in determining the "best" design. An efficient method for calculating accurate sensitivity derivatives can be applied in several different ways to significantly reduce the number and/or computational cost of these multiple analyses. This could be critical for the integration of advanced CFD codes into a systematic design methodology, where the computational cost of a single flow analysis can be extremely high, particularly in 3D.

One method of a very general yet conceptually simple nature for computing aerodynamic sensitivity derivatives is the method of "brute-force" finite differences. With this method, assuming forward finite difference approximations are used, the CFD flow analysis code is used to generate one converged flow solution for a slightly perturbed value of each design variable for which sensitivity derivatives are

0021-9991/94 $6.00

required. The principal drawback of this method is clearly that of computational cost, since the number of flow analyses required in a typical design problem can be extremely (i.e., prohibitively) large, particularly when the number of design variables is large.

As a typically less costly alternative to the finite difference approach, aerodynamic sensitivity derivatives can (in principle) be computed by direct differentiation of the governing equations which control the fluid flow. If the continuous governing equations are differentiated *prior* to their numerical discretization, the method is known as the "continuum" approach. In contrast, if the resulting algebraic equations which model the governing equations are differentiated *following* their discretization, the method is known as the "discrete" approach. In developing efficient methods for computing these sensitivity derivatives and their subsequent application to aerodynamic design problems, researchers have been and remain active; Refs. [1–25] are a representative (but not exhaustive) sample of papers which are germane to the present effort. Reference [8] addresses the distinction between the aforementioned "continuum" and "discrete" approaches, and Refs. [24, 25] are earlier studies upon which the present effort is based.

The present study represents an extension of the recent efforts of Refs. [13–23], where fundamental sensitivity equations are derived by direct differentiation of the system of discrete nonlinear algebraic equations which model either the Euler or thin-layer Navier–Stokes (TLNS) equations for 2D steady flow. This differentiation results in very large systems of *linear* algebraic sensitivity equations which must be solved to obtain these derivatives of interest. In Refs. [13–23], the fundamental sensitivity equations are solved in what shall be referred to as the "standard" (i.e., non-incremental) form. Furthermore, in these references, a direct solver method is applied to solve these equations; the single exception is Ref. [23], where a hybrid direct/iterative approach is adopted for an isolated airfoil example problem. There are some important advantages in using a direct method when feasible; these are discussed in the references and are also noted later in this article. However, the most serious disadvantage of a direct method is the extremely large computer storage requirement, which for practical 3D problems appears to be well beyond the current capacity of modern supercomputers; this capacity can even be exceeded in 2D on computational grids containing a large number of points.

In an effort to circumvent the computer storage limitation for the direct methods, this study focuses on fundamental algorithm development for the efficient *iterative* solution of the aerodynamic sensitivity equations. The principal motivation and objective is to develop a solid framework in 2D from which future extensions to 3D will be feasible. In general, one of the most serious difficulties encountered in the development and/or application of iterative techniques is lack of diagonal dominance and/or poor overall conditioning in the coefficient matrix. Unfortunately, this is a very common occurrence in the CFD coefficient matrices of interest; the severity varies greatly and depends on many factors. This problem can manifest itself in either poor performance or even complete failure (i.e., divergence) of an iterative algorithm.

A computationally useful property of the "incremental" form (also commonly known as the "delta" or "correction" form) can be effectively exploited to combat these problems of poor iterative algorithm performance. This property is that "approximations of convenience" can be introduced into the coefficient matrix operator of the equations, *without affecting the final converged values of the sensitivity derivatives*. The approximations must be "reasonable" enough so that the resulting iterative strategy is convergent. In contrast, if any approximations are made to the coefficient matrix operator of the equations in the standard form, then the computed sensitivity derivatives cannot be consistent discrete forms; that is, they will not be the correct derivatives of the nonlinear algebraic equations which are solved when generating the steady-state flow solution. In particular, it is proposed and successfully demonstrated numerically, that the identical diagonally dominant approximate coefficient matrix operator and algorithm, commonly associated with implicit methods for solving the nonlinear flow equations, can also be used to iteratively solve (in incremental form) the consistent discrete systems of linear equations for aerodynamic sensitivity analysis.

The remainder of this article is organized as follows. The next section, presentation of theory, is further subdivided into four subsections which review and discuss: (1) governing equations, spatial discretization, and implicit formulation; (2) fundamental aerodynamic sensitivity equations in standard form; (3) basic linear equation solving in incremental form; and (4) incremental solution of the equations of aerodynamic sensitivity analysis. In this last subsection, some significant implications of the incremental formulation are compared to the standard form. Following the presentation of theory section, computational results are presented which illustrate application of the methodology to two example airfoil problems: (1) subsonic low Reynolds number laminar flow; and (2) transonic high Reynolds number turbulent flow. The final section is a summary where conclusions are given.

2. PRESENTATION OF THEORY

2.1. *Governing Equations, Spatial Discretization, and Implicit Formulation*

The governing equations considered are the 2D thin-layer Navier–Stokes equations, which are solved numerically in integral conservation law form using an implicit upwind cell-centered finite volume formulation [26, 27]. Higher-order accurate approximate representation of the convective and pressure terms is accomplished using the popular flux-vector splitting upwind formulation of van Leer [28], and the thin-layer viscous terms are modeled using central "differences." At steady state, this discretization of the governing equations over the domain including the numerical boundary conditions is expressed as

$$\{R(Q^*)\}=\{0\}, \tag{1}$$

where Eq. (1) represents a large system of coupled nonlinear algebraic equations and the "root," $\{Q^*\}$, is the steady-state numerical solution for the field variables. In this equation and subsequently, the notation, "{ }", indicates a global column vector.

The well-known Newton linearized "incremental" strategy can in principle be applied to the root-finding problem in solving Eq. (1). Strict application of Newton's method, including consistent treatment of the boundary conditions, generally results in very large error reductions per iteration; quadratic convergence is achieved when the transient solution enters the range of attraction to the root [29, 30]. Newton's method presently appears infeasible for application to practical 3D problems because of the excessive computer storage which is required for a direct solution of the linear problem at each Newton iteration. While feasible in 2D, studies have also shown the direct Newton's method to be not necessarily the most efficient method with respect to overall CPU time, despite the large error reductions per iteration which are realized [29]. More commonly, therefore, CFD software has resorted to the use of *iterative* methods to solve these equations.

Implicit iterative methods for solving the Navier–Stokes equations are related to Newton's method and are represented here as

$$-\left[\frac{\widetilde{\partial R^n(Q)}}{\partial Q}\right]\{{}^n\Delta Q\}=\{R^n(Q)\} \tag{2}$$

$$\{Q^{n+1}\}=\{Q^n\}+\{{}^n\Delta Q\} \tag{3}$$

$$n=1,2,3,\ldots$$

where "n" is an iteration index, and $\{{}^n\Delta Q\}$ is the incremental change in the field variables from the present "known" (nth) iteration to the next (nth + 1) iteration level. An initial guess, $\{Q^1\}$, is required to initiate this iterative procedure, which in the present study is the freestream. The left-hand side coefficient matrix operator, $-[\widetilde{\partial R^n(Q)/\partial Q}]$, *approximates* the true Newton left-hand side coefficient Jacobian matrix operator. Typically the differences between the true Newton coefficient matrix operator and the approximate coefficient matrix operator and the approximate coefficient matrix operator of Eq. (2) include, but are not limited to,

(1) A "time-step" term is included (i.e., added) and thus significantly enhances each diagonal element of the coefficient matrix, $-[\widetilde{\partial R^n(Q)/\partial Q}]$, of Eq. (2). This is equivalent to the inclusion of under-relaxation in the true Newton's method, and under certain restrictions, can make the iterative procedure of Eqs. (2) and (3) "time-accurate."

(2) Simplifying linearization errors of various types are included in the construction of the approximate operator, $-[\widetilde{\partial R^n(Q)/\partial Q}]$. For example, consistent boundary condition linearization is typically neglected, and/or a first-order accurate upwind treatment of the inviscid terms might be used in this matrix operator, despite the higher-order accurate treatment of these terms in the vector $\{R^n(Q)\}$ on the right-hand side of the equations.

(3) Additional "approximations of convenience" are included in the matrix operator in order that a very efficient (in terms of computational work and computer storage) *approximate* solution of the linear problem can be generated at each iteration on the nonlinear problem. For example, with the popular spatially split approximate factorization (AF) method of Ref. [31], an approximate solution of Eq. (2) is produced at each nth iteration using alternating direction sweeps which involve the solution of a series of uncoupled sub-systems of block-tridiagonal linear equations in each sweep direction. It is this algorithm which is selected for use in the example problems of the present study. Additional well-known "iterative" algorithms which have been applied in solving the Navier–Stokes equations include (but are not limited to) LU (lower/upper) approximate factorizations [32], conventional relaxation methods [33], strongly implicit methods [34], and preconditioned conjugate gradient methods [35, 36].

Of course, these approximations result in far less error reduction per iteration than can be achieved with a faithful implementation of Newton's method; a converged steady-state solution generally requires hundreds or even thousands of iterations to achieve using the iterative methods. Because Eq. (2) is in incremental or "delta" form, however, at convergence the steady-state solution, $\{Q^*\}$, is independent of any and all approximations which are used in the left-hand side coefficient matrix operator.

2.2. *Fundamental Aerodynamic Sensitivity Equations in Standard Form*

In general, the jth aerodynamic system response, C_j, is functionally dependent on the steady-state field variables, $\{Q^*\}$; the vector of computational grid (x, y) coordinates, $\{\bar{X}\}$; and perhaps also explicitly on the vector of independent design variables, $\bar{\beta}$. That is,

$$C_j = C_j(Q^*(\bar{\beta}), \bar{X}(\bar{\beta}), \bar{\beta}). \tag{4}$$

The sensitivity derivative of C_j with respect to the kth design variable, β_k (i.e., the kth element of $\bar{\beta}$), is thus

$$\frac{dC_j}{d\beta_k} = \left\{\frac{\partial C_j}{\partial Q}\right\}^{\mathrm{T}} \left\{\frac{dQ^*}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^{\mathrm{T}} \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k}, \tag{5}$$

where superscript "T" denotes transpose.

The notation for a total derivative has been used on the left-hand side of Eq. (5) to indicate that the *total* rate of change of C_j with respect to β_k is included in the term and to distinguish it from the partial derivative on the right-hand side of the equation. Nevertheless, $dC_j/d\beta_k$ is a partial derivative in the sense that C_j is in general a function of multiple independent design variables, $\bar{\beta}$, as seen in Eq. (4). In Eq. (5), the term $\{d\bar{X}/d\beta_k\}$, known as the grid sensitivity vector, can be evaluated using any of several methods which have been suggested [23, 37, 38]; the strategy of Ref. [23] was selected in the present study. Another method, of course, would be to use the grid generation program to obtain "brute-force" finite differences for evaluating these terms. The grid sensitivity vector is null if the design variable, β_k, is not related to the geometric shape of the domain. The vector $\{dQ^*/d\beta_k\}$, which is the sensitivity of the steady-state field variables with respect to the kth design variable, is evaluated for use in Eq. (5) by solving a large system of coupled linear sensitivity equations which is derived subsequently.

The large system of coupled nonlinear algebraic equations which model the flow, given previously by Eq. (1), can be expressed in general as

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

where now, the dependence of these equations on the grid, $\{\bar{X}\}$, and on the design variables, $\bar{\beta}$, is noted. In addition, Eq. (6) includes the possibility of an explicit dependence on the steady-state lift coefficient, C_L. This explicit dependence is found in the far-field boundary conditions of an isolated lifting airfoil when the accurate "lift-corrected" far-field boundary conditions of Ref. [39] have been used, as in the example problems of this study. Note that C_L itself depends on the field variables, $\{Q^*\}$; the grid, $\{\bar{X}\}$; and possibly explicitly on the design variables, $\bar{\beta}$; in the manner expressed by Eq. (4). The explicit dependence on C_L noted in Eq. (6) might therefore appear redundant; however, the computational advantages of this particular grouping of terms is discussed in detail in Ref. [23] and will become apparent.

Differentiation of Eq. (6) with respect to β_k yields

$$\left\{\frac{dR}{d\beta_k}\right\} = \left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ^*}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\} + \left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k} = \{0\}, \tag{7}$$

where in Eq. (6) the term $dC_L/d\beta_k$ is evaluated using a relationship of the form given by Eq. (5). Note that the vector $\{\partial R/\partial C_L\}$ is very sparse; nonzero contributions to it arise only from the "lift-corrected" far-field boundary condition equations. Equation (7) is thus a large system of coupled linear equations which in principle can be solved for the unknown vector, $\{dQ^*/d\beta_k\}$, one such solution for each design variable, β_k. This method is known as the *quasi-analytical method* for computing sensitivity derivatives.

The matrix $[\partial R/\partial Q]$ of Eq. (7) is the Jacobian of the nonlinear flow equations (evaluated at steady-state) with respect to the field variables and includes consistent treatment of all boundary conditions; an exception is that contribution resulting from the explicit dependence of the lift-corrected far-field boundary condtions on C_L. Substitution of Eq. (5) for $dC_L/d\beta_k$ into Eq. (7) reveals that this contribution to $[\partial R/\partial Q]$ is given by the very sparse matrix, $\{\partial R/\partial C_L\}\{\partial C_L/\partial Q\}^{\mathrm{T}}$. The matrix $[\partial R/\partial \bar{X}]$ of Eq. (7) is the Jacobian of the flow equations (evaluated at the steady state and including all boundary conditions) with respect to the grid coordinates [17, 18]; again the exception is the contribution from the explicit dependence of the far-field boundary conditions on C_L. Here this contribution is given by the very sparse matrix $\{\partial R/\partial C_L\}\{\partial C_L/\partial \bar{X}\}^{\mathrm{T}}$. The vector $\{\partial R/\partial \beta_k\}$ of Eq. (7) accounts for explicit dependencies (if any) of the flow equations, including boundary conditions, on β_k; the contribution to this vector from the C_L dependence of the far-field boundary conditions is given by the vector $\{\partial R/\partial C_L\}(\partial C_L/\partial \beta_k)$.

A well-known closely related alternative strategy for computing sensitivity derivatives, known as the *adjoint variable method*, is easily developed using expressions which have been presented thus far. This begins by combining Eqs. (5) and (7) to yield

$$\begin{aligned}\frac{dC_j}{d\beta_k} = {} & \left\{\frac{\partial C_j}{\partial Q}\right\}^{\mathrm{T}} \left\{\frac{dQ^*}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^{\mathrm{T}} \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k} \\ & + \{\lambda_j\}^{\mathrm{T}}\left(\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ^*}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} \right. \\ & \left. + \left\{\frac{\partial R}{\partial \beta_k}\right\} + \left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k}\right).\end{aligned} \tag{8}$$

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. (5) to Eq. (8) since the entire additional term on the right-hand side of Eq. (8) is zero, for any and all $\{\lambda_j\}$. Expanding and rearranging, Eq. (8) becomes

$$\frac{dC_j}{d\beta_k}=\left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^{\mathrm{T}}+\{\lambda_j\}^{\mathrm{T}}\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{d\bar{X}}{d\beta_k}\right\}+\frac{\partial C_j}{\partial \beta_k}$$
$$+\{\lambda_j\}^{\mathrm{T}}\left\{\frac{\partial R}{\partial \beta_k}\right\}+\{\lambda_j\}^{\mathrm{T}}\left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k}$$
$$+\left(\left\{\frac{\partial C_j}{\partial Q}\right\}^{\mathrm{T}}+\{\lambda_j\}^{\mathrm{T}}\left[\frac{\partial R}{\partial Q}\right]\right)\left\{\frac{dQ^*}{d\beta_k}\right\}. \tag{9}$$

The necessity of evaluating the vector, $\{dQ^*/d\beta_k\}$, using Eq. (7) is eliminated for all β_k by selecting the vector, $\{\lambda_j\}$, such that the coefficient of $\{dQ^*/d\beta_k\}$ in Eq. (9) is null. That is, selection of $\{\lambda_j\}$ which satisfies

$$\left\{\frac{\partial C_j}{\partial Q}\right\}^{\mathrm{T}}+\{\lambda_j\}^{\mathrm{T}}\left[\frac{\partial R}{\partial Q}\right]=\{0\}^{\mathrm{T}} \tag{10}$$

or

$$\left[\frac{\partial R}{\partial Q}\right]^{\mathrm{T}}\{\lambda_j\}+\left\{\frac{\partial C_j}{\partial Q}\right\}=\{0\}. \tag{11}$$

Therefore, following the solution of Eq. (11) 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

$$\frac{dC_j}{d\beta_k}=\left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^{\mathrm{T}}+\{\lambda_j\}^{\mathrm{T}}\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{d\bar{X}}{d\beta_k}\right\}+\frac{\partial C_j}{\partial \beta_k}$$
$$+\{\lambda_j\}^{\mathrm{T}}\left\{\frac{\partial R}{\partial \beta_k}\right\}+\{\lambda_j\}^{\mathrm{T}}\left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k}. \tag{12}$$

Note that Eq. (12) can be solved for $dC_j/d\beta_k$ only if $dC_L/d\beta_k$ is known *or* if $C_j=C_L$. Therefore, when the lift-corrected far-field boundary conditions are treated in the manner described, then $dC_L/d\beta_k$ must be the *first* sensitivity derivative which is calculated (for any and all β_k of concern), regardless of whether the sensitivity of C_L is of actual interest. (Normally, of course, the sensitivity derivatives of C_L will be of interest in a typical problem.) A particular solution, $\{\lambda_j\}$, is valid only for a specific system response, C_j, and thus solution of Eq. (11) must be repeated for each different system response of interest.

It is simple to verify from the preceding equations, and significant to note, that each solution, $\{dQ^*/d\beta_k\}$, of Eq. (7) for a particular design variable can be used for an unlimited number of different system responses. In constrast, however, each solution, $\{\lambda_j\}$, of Eq. (11) for a particular system response can be used for an unlimited number of different design variables. Therefore, the total number of large linear systems which must be solved for a particular problem can be minimized through a judicious selection of one of these two methods, depending on whether the number of system responses of interest or the number of design variables of interest is larger.

In terms of computational efficiency, the significance of the well-known difference in the two methods is mitigated greatly if a direct method is used to solve these linear systems (i.e., either Eq. (7) or Eq. (11)), because with either method the LU factorization must only be done once and it is then repeatedly reused for multiple right-hand side vectors. However, this distinction can become 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. Despite this difference, it is emphasized that these two methods are *equivalent* in the sense that they yield identical values for the sensitivity derivatives, if properly implemented computationally.

Summarizing briefly, it has been shown that calculating aerodynamic sensitivity derivatives using the discrete direct differentiation method requires the direct or iterative solution of large linear systems of equations of the type given by either Eq. (7) or Eq. (11). These two systems of linear equations are referred to as the *aerodynamic sensitivity equations in standard form*. Fundamental algorithm development for the solution of one of these two linear systems is easily extended and applied to the other, since their respective coefficient matrices, $[\partial R/\partial Q]$ and $[\partial R/\partial Q]^{\mathrm{T}}$, are transposes of each other. When the standard form equations are solved, *no* approximations can be introduced into any of the terms, without simultaneously introducing error into the resulting sensitivity derivatives. In this form, the framework to support the development of iterative methods is thus rigid and restrictive.

As a consequence of the preceding discussion, given the choice of a higher-order accurate upwind approximation for the spatial discretization for the flow analysis, a consistent higher-order accurate upwind spatial discretization including a fully consistent treatment of all boundary conditions is *required* in the coefficient matrix operator of the sensitivity equations (in standard form). Furthermore, there can be no "time term" added here to enhance each element of the diagonal, as is used (in contrast) in the implicit formulation of Eq. (2) for solving the nonlinear flow equations. Unfortunately, the resulting coefficient matrix (either $[\partial R/\partial Q]$ or $[\partial R/\partial Q]^{\mathrm{T}}$) of the linear sensitivity equations in standard form in this case is not block-diagonally dominant [33], and consequently the computational performance of traditional iterative methods for solving these equations in this standard form is expected to be poor, or even to fail [23]. Therefore, it is this particular difficulty (i.e., the lack of sufficient diagonal dominance) and its resolution which is of principal concern in the development of the incremental form of the equations in the following sections.

2.3. *Basic Linear Equation Solving in Incremental Form*

Consider the linear system of algebraic equations in the general form

$$[A]\{Z^*\} + \{B\} = \{0\}, \tag{13}$$

where $\{Z^*\}$ is the solution vector. In treating the problem of solving Eq. (13), in essence a "root finding" problem, application of Newton's method (traditionally used in root finding for nonlinear equations) to the linear problem yields the basic two-step iterative incremental formulation

$$-[\tilde{A}]\{{}^m\Delta Z\} = [A]\{Z^m\} + \{B\} \tag{14}$$

$$\{Z^{m+1}\} = \{Z^m\} + \{{}^m\Delta Z\} \tag{15}$$

$$m = 1, 2, 3, ...,$$

where "m" is an iteration index, and $\{{}^m\Delta Z\}$ is the incremental change in the solution from the known (mth) to the next (mth + 1) iteration level. An initial guess, $\{Z^1\}$, is required to begin the procedure, which in the present study is taken everywhere as zero. If Newton's method is applied strictly, the coefficient matrix $[\tilde{A}]$ is equal to the matrix $[A]$, and clearly the two-step iterative strategy of Eqs. (14) and (15) for the linear problem converges on the first iteration, for any initial guess. Therefore, in this case, solution of the linear system in the standard form (Eq. (13)) and solution in the incremental form (Eqs. (14) and (15)) are equivalent.

More generally, however, the matrix $[\tilde{A}]$ is not necessarily equal to the matrix $[A]$. The matrix $[\tilde{A}]$ can be *any convenient approximation* of the matrix $[A]$ with the restriction that $[\tilde{A}]$ must approximate $[A]$ well enough so that the two-step iterative procedure (Eqs. (14) and (15)) converges (or, at the very least, can be forced to converge by including a strategy such as under-relaxation). Simply stated, $[\tilde{A}]$ should capture the essence of $[A]$. Furthermore, because the equations have been cast in "delta" form, the incremental method produces the unique solution of Eq. (13), $\{Z^*\}$, if convergent. In this formulation, the purpose of the left-hand side operator is to drive the right-hand side vector to zero.

2.4. *Incremental Solution of the Equations of Aerodynamic Sensitivity Analysis*

Application of the fundamental incremental formulation for linear equation solving, Eqs. (14) and (15), to the linear system of Eq. (7) (i.e., the quasi-analytical method) for computing aerodynamic sensitivity derivatives, gives

$$-\left[\widetilde{\frac{\partial R}{\partial Q}}\right]\left\{{}^m\Delta\frac{dQ}{d\beta_k}\right\} = \left\{\frac{dR^m}{d\beta_k}\right\} \tag{16}$$

$$\left\{\frac{dQ^{m+1}}{d\beta_k}\right\} = \left\{\frac{dQ^m}{d\beta_k}\right\} + \left\{{}^m\Delta\frac{dQ}{d\beta_k}\right\} \tag{17}$$

$$m = 1, 2, 3, ...,$$

where

$$\begin{aligned}\left\{\frac{dR^m}{d\beta_k}\right\} &= \left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ^m}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\}\\ &\quad + \left\{\frac{\partial R}{\partial \beta_k}\right\} + \left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L^m}{d\beta_k}\\ \frac{dC_L^m}{d\beta_k} &= \left\{\frac{\partial C_L}{\partial Q}\right\}^{\mathrm{T}}\left\{\frac{dQ^m}{d\beta_k}\right\} + \left\{\frac{\partial C_L}{\partial \bar{X}}\right\}^{\mathrm{T}}\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_L}{\partial \beta_k},\end{aligned} \tag{18}$$

where the left-hand side coefficient matrix operator $[\widetilde{\partial R/\partial Q}]$ approximates the matrix $[\partial R/\partial Q]$ and will be discussed subsequently, in greater detail. The vector $\{dR^m/d\beta_k\}$ represents the mth iteration on the *total* derivative of the discrete steady-state nonlinear flow equations, Eq. (6), with respect to β_k. From Eq. (7), clearly this vector must be driven to zero in order to find the solution, $\{dQ^*/d\beta_k\}$, of Eq. (7), which is of course the objective of the incremental strategy of Eqs. (16), (17), and (18). Approximations must *not* be made to any terms of the vector, $\{dR^m/d\beta_k\}$, taking particular care that a consistent treatment of all boundary conditions is included, if the converged solution is to yield the correct, consistent, discrete sensitivity derivatives. The final solution at convergence depends only on the terms of this right-hand side vector.

It is proposed that the identical approximate left-hand side coefficient matrix operator $[\widetilde{\partial R/\partial Q}]$ and algorithm which are used in solving Eq. (2) for the flow variables also be used (when evaluated at the steady state) as the approximate left-hand side operator and algorithm which are used in solving Eq. (16) for the flow sensitivities. That is, a *first-order* accurate upwind spatial discretization of the inviscid terms is used in this operator as an approximation here to the higher-order accurate upwind discretization of these terms. It is most significant to note that by design in this choice, block-diagonal dominance is now obtained and maintained in the left-hand side coefficient matrix. In addition, a false "time term" is included (i.e., added) so each diagonal element of the matrix, $[\widetilde{\partial R/\partial Q}]$, is further enhanced; this is equivalent to under-relaxation in the incremental strategy of Eqs. (16), (17), and (18). The boundary conditions are not linearized in a fully consistent manner in this approximate matrix operator; far off-diagonal contributions from the "periodic" boundary conditions which arise when calculations are performed on a "C" or "O" mesh are neglected. However, these "periodic" boundary conditions cause special computational difficulties for the standard form equations which require a consistent treatment in the left-hand side matrix operator [23, 25]. Finally, the well-known spatially split approximate factorization (AF) algorithm [31] (also used here in solving the nonlinear flow equations) is used to solve Eq. (16) (approximately) at each mth iteration. If the resulting block-tridiagonal coefficient matrices are stored over the entire domain, only a single LU factorization of each of them is required, which can be repeatedly reused for *all iterations and all design variables*; this strategy is implemented in the large 2D example problems which are presented.

If the adjoint variable formulation for computing the sensitivity derivatives is preferred, then application of the incremental formulation for linear equation solving, Eqs. (14) and (15), to the linear system of Eq. (11) for computing the adjoint variable vector, $\{\lambda_j\}$, yields

$$-\widetilde{\left[\frac{\partial R}{\partial Q}\right]}^{\mathrm{T}}\{{}^m\Delta\lambda_j\} = \left[\frac{\partial R}{\partial Q}\right]^{\mathrm{T}}\{\lambda_j^m\} + \left\{\frac{\partial C_j}{\partial Q}\right\} \qquad (19)$$

$$\{\lambda_j^{m+1}\} = \{\lambda_j^m\} + \{{}^m\Delta\lambda_j\} \qquad (20)$$

$$m = 1, 2, 3, \ldots.$$

For application in Eq. (19), it is straightforward to transpose the approximate left-hand side coefficient matrix operator and algorithm which were described previously for use in Eq. (16). Again, only a single LU factorization of the globally stored block-tridiagonal coefficient matrices is required.

3. COMPUTATIONAL RESULTS

3.1. *Subsonic Airfoil, Low Reynolds Number Laminar Flow*

The first example problem is subsonic low Reynolds number constant viscosity laminar flow over a NACA 1406 airfoil. Flow is considered at a freestream Mach number, $M_\infty = 0.6$, angle of attack, $\alpha = 1.0°$, and Reynolds number, $\mathrm{RE} = 5.0 \times 10^3$. A computational grid, a "C" mesh, of 257×65 points is used, with the "lift-corrected" far-field boundary placed five chords from the airfoil; points are clustered near the airfoil surface to assist with the resolution of gradients in this vicinity. The spatially split AF algorithm is used to achieve the converged (i.e., the average global error is reduced to machine-zero) steady-state solution, $\{Q^*\}$, to the discrete nonlinear flow equations, Eq. (1). Figure 1 is a plot of the computed steady-state pressure coefficient, C_p, on the surface of the airfoil. The computed lift, drag, and pitching moment coefficients obtained are $C_L = 0.18148$, $C_D = 0.41703$ E-01, and $C_M = -0.23718$ E-01.

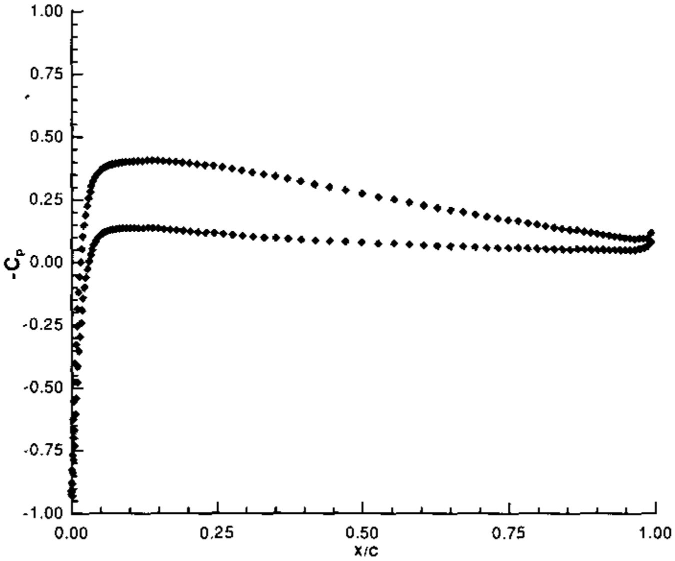

FIG. 1. Chordwise distribution of surface pressure coefficient, NACA 1406 airfoil, $M_\infty = 0.6$, angle of attack, $\alpha = 1.0°$, and Reynolds number, $\mathrm{RE} = 5.0 \times 10^3$, laminar flow.

Sensitivity derivatives of C_L, C_D, and C_M are computed with respect to six independent design variables: (1) airfoil maximum thickness, T; (2) airfoil maximum camber, C; (3) location of maximum camber, L; (4) angle of attack, α; (5) freestream Mach number, M_∞; and (6) Reynolds number, RE. The three geometric shape related design variables (T, C, and L) are parameters which together with well-known analytical expressions (given, for example, in Refs. [2 or 23]) define the "x" and "y" coordinates on the surface (and hence the shape) of the NACA 4-digit airfoil. Sensitivity derivatives are computed using three methods: (1) the quasi-analytical method; (2) the adjoint variable method; and (3) the "brute force" finite difference method. Application of these three methods is described subsequently in greater detail; computational result comparisons are summarized in Table I. For the quasi-analytical and adjoint variable methods, it is noted that the direct solver approach was abandoned, because for this large computational grid the conventional ("in-core" storage) banded matrix solver algorithm exceeded the maximum 40 megaword storage computer facility restriction.

For the quasi-analytical method, sensitivity derivatives are calculated through the iterative solution of the incremental form (i.e., Eqs. (16), (17), and (18)) of six large systems of linear equations, one such linear system for each of the six design variables considered here. The well-known spatially split AF algorithm [31] is used, with a constant Courant number of 45 (i.e., local time-stepping is used), which is found by numerical experimentation to be about the optimum for computational efficiency in this example

TABLE I

Summary of the Computational Results for the NACA 1406 Airfoil: Subsonic Laminar Low Reynolds Number Example Problem

Solution method	Total CPU time (s)[a]	Design variable β_k	$\frac{dC_L}{d\bar{\beta}}$	$\frac{dC_D}{d\bar{\beta}}$	$\frac{dC_M}{d\bar{\beta}}$
Quasi-	458				
analytical		T	−1.392 E+00	+2.019 E−01	+1.805 E−01
method,		C	+6.583 E+00	+7.583 E−02	−2.240 E+00
approximately		L	−1.154 E−02	+5.544 E−05	−2.122 E−02
factored		α	+6.122 E+00	+9.181 E−02	−3.168 E−02
incremental		M_∞	+5.428 E−03	+1.628 E−02	−4.732 E−03
scheme		RE	+5.958 E−06	−4.912 E−06	−6.564 E−07
Adjoint	579				
variable		T	−1.392 E+00	+2.019 E−01	+1.805 E−01
method,		C	+6.583 E+00	+7.583 E−02	−2.240 E+00
approximately		L	−1.154 E−02	+5.544 E−05	−2.122 E−02
factored		α	+6.122 E+00	+9.181 E−02	−3.168 E−02
incremental		M_∞	+5.428 E−03	+1.628 E−02	−4.732 E−03
scheme		RE	+5.958 E−06	−4.912 E−06	−6.564 E−07
"Brute force"	7404	T	−1.392 E+00	+2.019 E−01	+1.805 E−01
finite		C	+6.583 E+00	+7.583 E−02	−2.240 E+00
difference		L	−1.154 E−02	+5.548 E−05	−2.122 E−02
method		α	+6.122 E+00	+9.181 E−02	−3.168 E−02
		M_∞	+5.426 E−03	+1.628 E−02	−4.732 E−03
		RE	+5.958 E−06	−4.912 E−06	−6.564 E−07

[a] All calculations performed on a Cray-2 computer.

problem. An eight order-of-magnitude reduction in the average global error is the specified convergence criterion in solving each of these six linear systems; an average of 683 iterations is required in each case to achieve this convergence criterion.

For the adjoint variable method, sensitivity derivatives are calculated through the iterative solution of the incremental form (i.e., Eqs. (19) and (20)) of three large systems of linear equations, one such linear system for each of the three system responses considered here. Again the AF algorithm is used, and a constant Courant number of 45 is found to be about the optimum. An average of 1743 iterations is required to achieve an eight order-of-magnitude average global error reduction, the required convergence criterion for each of these three linear system solutions.

In application of the "brute-force" finite difference method, central finite differences are used, with a forward and backward perturbation of each design variable, $\Delta\beta_k = \pm 5.0\text{E}-0.6 \times \beta_k$. Machine-zero converged steady-state solutions of the discrete nonlinear flow equations are obtained for each forward and backward perturbation of each design variable; thus for six design variables, a total of 12 solutions to the nonlinear flow equations are produced. The AF algorithm is again used to solve the flow equations; in order to reduce computational work during these solutions, the LU factored block-tridiagonal systems are stored over the domain and are repeatedly reused for 10 iterations prior to each re-evalution of these terms. (See Ref. [40] for additional details concerning this strategy; there it was shown via numerical studies to be near optimum.)

In comparing the sensitivity derivatives calculated using the quasi-analytical method with the adjoint variable method, the results are seen to agree, as expected. Unexpectedly, however, the computational work required by the latter method (where a total of three linear systems are solved) is seen to exceed that of the former (where a total of six linear systems are solved); the convergence rates obtained with the latter method were significantly slower than those for the former method in this particular example problem. In comparing the sensitivity derivatives calculated using the method of finite differences with the other two methods, excellent agreement is obtained, as expected. The "brute-force" finite difference method is seen to be very much more costly computationally than either the quasi-analytical or adjoint variable methods.

3.2. *Transonic Airfoil, High Reynolds Number Turbulent Flow*

The second example problem is transonic high Reynolds number turbulent flow over a NACA 1406 airfoil. The variation of the molecular viscosity with temperature is computed using Sutherland's law, and turbulence is simulated using the well-known algebraic model of Baldwin and Lomax [41]. Flow is considered at a freestream Mach number, $M_\infty = 0.8$, angle of attack, $\alpha = 1.0°$, and Reynolds number, RE $= 5.0 \times 10^6$. A "C" mesh with 257×65 grid points is again used with the "lift-corrected" far-field boundary placed five chords from the airfoil; clustering of points near the surface is tighter in the present example than in the previous example because of the higher Reynolds number. The spatially split AF algorithm is used to achieve a machine-zero converged steady-state solution. Figure 2 is a plot of the computed steady-state pressure coefficient, C_p, on the surface of the airfoil, and Fig. 3 is a complete contour plot of the static pressure, which clearly shows the presence of a shock wave on the suction surface of the airfoil. The computed lift, drag, and pitching moment coefficients are $C_L = 0.41662$, $C_D = 0.77501$ E-02, and $C_M = -0.45633$ E-01.

Sensitivity derivatives of C_L, C_D, and C_M are computed with respect to the same six independent design variables previously considered. The quasi-analytical, the adjoint variable, and the "brute-force" finite difference methods are also applied in computing these sensitivity derivatives. However, for the quasi-analytical and adjoint variable methods, the approximation of neglecting the variation of the laminar and turbulent viscosities with respect to the field variables, $\{Q^*\}$, and the computational grid, $\{\bar{X}\}$, is made. That is, in the analytical construction of all the derivatives (including the Jacobian matrices, $[\partial R/\partial Q]$ and $[\partial R/\partial \bar{X}]$) which are used to calculate the sensitivity derivatives, the approximation is made that both laminar and turbulent viscosities are constant. For this reason, the quasi-analytical or the adjoint variable methods cannot give sensitivity derivatives which are *exactly* consistent discrete forms; the results from the "brute-force" finite difference procedure are thus considered to be more accurate in this example. This approximation is used, of course, because of the great complexity involved in consistently treating the derivatives of the turbulent viscosity; in fact, a fully consistent treatment of these terms is not possible at points where this turbulence model is not continuously differentiable. Application of the three methods is described subsequently in greater detail; computational result comparisons are summarized in Table II.

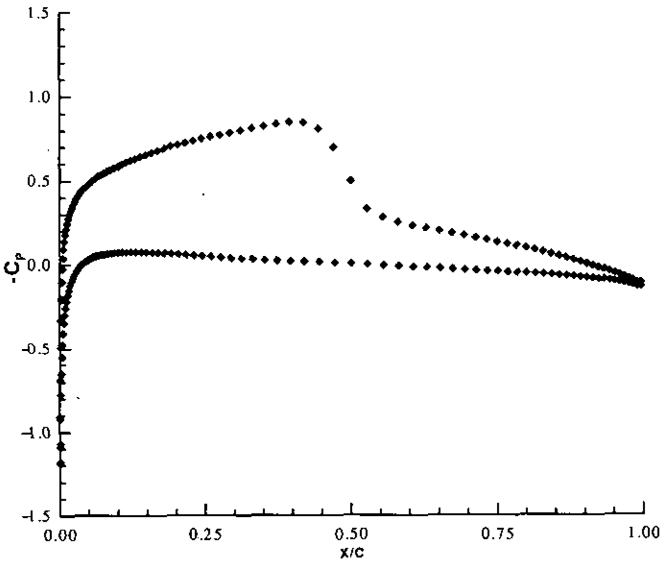

FIG. 2. Chordwise distribution of surface pressure coefficient, NACA 1406 airfoil, $M_\infty = 0.8$, angle of attack, $\alpha = 1.0°$, and Reynolds number, RE $= 5.0 \times 10^6$, turbulent flow.

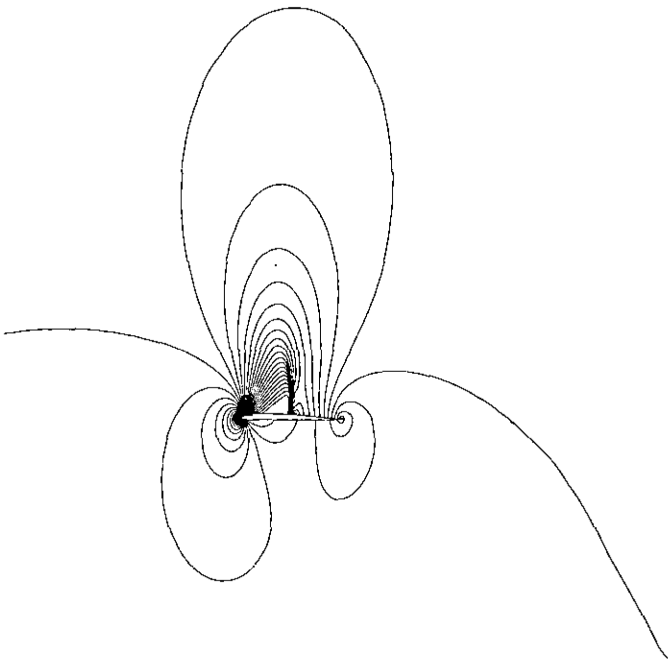

FIG. 3. Static pressure contour plot, NACA 1406 airfoil, $M_\infty = 0.8$, angle of attack, $\alpha = 1.0°$, and Reynolds number, RE $= 5.0 \times 10^6$, turbulent flow.

For the quasi-analytical and adjoint variable methods, the sensitivity derivatives are computed using the spatially split AF algorithm to iteratively solve in incremental form the required linear systems which have been described. With both methods, a constant Courant number of 30 is used (approximately the optimum value, as determined numerically); in all cases an eight order-of-magnitude reduction in the average global error is the convergence criterion enforced. For the quasi-analytical method, an average of 1619 iterations is needed to achieve convergence; for the adjoint variable method, an average of 1798 iterations is required. Finally, the "brute force" finite difference method is applied here in a manner identical to that described in the previous example problem.

In comparing the sensitivity derivatives calculated using the quasi-analytical method with the adjoint variable method, the results are seen to agree, as expected. In addition, it is noted that the total computational cost of the for-

TABLE II

Summary of the Computational Results for the NACA 1406 Airfoil: Transonic Turbulent High Reynolds Number Example Problem

Solution method	Total CPU time (s)[a]	Design variable β_k	$\frac{dC_L}{d\bar{\beta}}$	$\frac{dC_D}{d\bar{\beta}}$	$\frac{dC_M}{d\bar{\beta}}$
Quasi-	1052				
analytical		T	+2.275 E−01	+2.654 E−01	−3.124 E−01
method,		C	+1.942 E+01	+6.511 E−01	−5.516 E+00
approximately		L	+1.338 E−01	−1.151 E−02	−5.589 E−02
factored		α	+1.198 E+01	+4.200 E−01	−4.675 E−01
incremental		M_∞	+1.772 E+00	+1.921 E−01	−5.430 E−01
scheme		RE	+4.145 E−09	−4.881 E−10	−4.397 E−10
Adjoint	586				
variable		T	+2.275 E−01	+2.654 E−01	−3.124 E−01
method,		C	+1.942 E+01	+6.511 E−01	−5.516 E+00
approximately		L	+1.338 E−01	−1.151 E−02	−5.589 E−02
factored		α	+1.198 E+01	+4.200 E−01	−4.675 E−01
incremental		M_∞	+1.772 E+00	+1.921 E−10	−5.430 E−01
scheme		RE	+4.145 E−09	−4.881 E−10	−4.397 E−10
"Brute force"	8526	T	+7.919 E−01	+2.744 E−01	−4.153 E−01
finite		C	+2.063 E+01	+6.776 E−01	−5.770 E+00
difference		L	+1.107 E−01	−1.174 E−02	−5.350 E−02
method		α	+1.299 E+01	+4.346 E−01	−6.328 E−01
		M_∞	+2.040 E+00	+1.969 E−01	−5.972 E−01
		RE	−1.185 E−09	−2.829 E−10	+1.497 E−10

[a] All calculations performed on a Cray-2 computer.

mer method is approximately twice the cost of the latter, as expected (since in the former method six linear systems are solved compared to only three in the latter method, and the average number of iterations is comparable). In comparing the sensitivity derivatives calculated using the method of finite differences with the other two methods, there is some discrepancy in the results, as expected, because of the neglected consistent treatment of the viscosity. For the most part, the agreement between these calculated derivatives is good. The most significant discrepancy is noted in the sensitivity derivatives of C_L with respect to maximum airfoil thickness, T, where the derivatives differ by a factor of about three or four. However, this sensitivity derivative is smaller in magnitude than the largest derivatives. As in the first example problem, the "brute-force" finite difference method is seen here to be very much more costly computationally than either the quasi-analytical or adjoint variable methods.

4. SUMMARY AND CONCLUSIONS

An incremental strategy has been presented for iteratively solving the very large systems of linear equations which are associated with aerodynamic sensitivity derivatives for advanced CFD codes. The method permits use of an approximate left-hand side coefficient matrix operator of convenience, which at convergence yields the consistent discrete sensitivity derivatives of interest. In the present research, it is shown that the identical left-hand side matrix operator and well-known spatially split approximate factorization algorithm used to solve the nonlinear flow equations can also be successfully used to efficiently solve the linear sensitivity equations. The procedures are demonstrated on two example airfoil problems: subsonic low Reynolds number laminar flow and transonic high Reynolds number turbulent flow.

REFERENCES

1. E. C. Yates Jr. and R. Desmarais, "Boundary Integral Method for Calculating Aerodynamic Sensitivities with Illustration for Lifting Surface Theory," in *Proceedings, International Symposium of Boundary Element Methods (IBEM 89), Oct. 2–4, 1989, East Hartford, Conn.*, (Springer-Verlag, New York, 1990).
2. H. M. Elbanna and L. A. Carlson, *J. Aircraft* **27** (6) 507; (1990), AIAA Paper 89-0532.
3. E. C. Yates Jr., NASA TM-100502, September 1987 (unpublished).
4. J. Sobieszczanski-Sobieski, "The Case for Aerodynamic Sensitivity Analysis," in *Sensitivity Analysis in Engineering:* NASA CP-2457, 1987; NASA, Washington, DC, 1987.
5. D. R. Bristow and J. D. Hawk, NASA CP-3528, 1982 (unpublished).
6. A. Jameson, NASA CR-181749; ICASE Report No. 88–64, November 1988 (unpublished).
7. P. D. Frank and G. R. Shubin, *J. Comput. Phys.* **98** (1), 74 (1992).
8. G. R. Shubin and P. D. Frank, Applied Mathematics and Statistics

Technical Report, AMS-TR-163, Boeing Computer Services, Seattle, WA, April 1991 (unpublished).

9. G. R. Shubin, Applied Mathematics and Statistics Technical Report, AMS-TR-164, Boeing Computer Services, Seattle, WA, June 1991 (unpublished).
10. T. M. Sorensen, M. S. thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 1991 (unpublished).
11. G. S. Dulikravich, *J. Aircraft* **29** (6), 1020 (1992).
12. S. Ta'asan, G. Kuruvila, and M. D. Salas, AIAA Paper 92-0025, January 1992 (unpublished).
13. O. Baysal and M. E. Eleshaky, *ASME J. Fluids Eng.* **113** (4), (1991); in *Recent Advances and Applications in CFD*, edited by O. Baysal, ASME-FED, Vol. 103; 11th Winter Annual Meeting, November 1990, p. 191.
14. O. Baysal and M. E. Eleshaky, *AIAA J.* **30** (3), 718 (1992); AIAA Paper 91-0471, January 1991.
15. O. Baysal, M. E. Eleshaky, and G. W. Burgreen, in *Proceedings, AIAA 10th Computational Fluid Dynamics Conference, June 24–26, 1991, Honolulu, Hawaii*; AIAA Paper 91-1577.
16. M. E. Eleshaky and O. Baysal, *ASME J. Fluids Eng.* **115** (1) 75 (1993); in *Multidisciplinary Applications of Computational Fluid Dynamics*, edited by O. Baysal; ASME-FED, Vol. 129, 12th Winter Annual Meeting, December 1991, p. 27.
17. A. C. Taylor III, V. M. Korivi, and G. W. Hou, *AIAA J.* **30** (8), 2163 (1992); AIAA 91-0173, January 1991.
18. A. C. Taylor III, G. W. Hou, and V. M. Korivi, *AIAA J.* **30** (10), 2411 (1992); in *Proceedings, AIAA/ASME/ASCE/AHS/ACS 32nd Structures, Structural Dynamics, and Materials Conference, April 1991*, 477, AIAA Paper 91-1101-CP.
19. V. M. Korivi, M. S. thesis, Old Dominion University, Norfolk, VA, August 1991.
20. A. C. Taylor III, V. M. Korivi, and G. W. Hou, in *Proceedings, AIAA 10th Computational Fluid Dynamics Conference, June 24–26, 1991, Honolulu, Hawaii*; AIAA Paper 91-1569.
21. A. C. Taylor III, G. W. Hou, and V. M. Korivi, AIAA Paper 91-1680, June 1991 (unpublished).
22. G. W. Hou, A. C. Taylor III, and V. M. Korivi, AIAA Paper 91-2259, June 1991; *Int. J. Numer. Methods Eng.*, to appear.
23. A. C. Taylor III, G. W. Hou, and V. M. Korivi, AIAA Paper 91-3083, September 1991.
24. P. A. Newman, G. J.-W. Hou, H. E. Jones, A. C. Taylor III, and V. M. Korivi, NASA TM-104206, February 1992; in *AIAA CP-9213, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992*, p. 531; AIAA Paper 92-4753-CP (unpublished).
25. V. M. Korivi, A. C. Taylor III, P. A. Newman, G. J.-W. Hou, and H. E. Jones, NASA TM-104207, February 1992; in *AIAA CP-9213, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992*, p. 465; AIAA Paper 92-4746-CP.
26. R. W. Walters and J. L. Thomas, in *State of the Art Surveys of Computational Mechanics*, edited by A. K. Noor (ASME, New York, 1989), p. 145.
27. J. L. Thomas, B. Van Leer, and R. W. Walters, *AIAA J.* **28** (6) 973 (1990); AIAA Paper 85-1680.
28. B. Van Leer, ICASE Report 82-30, September 1982; *Lecture Notes in Phys.*, Vol. 170 (Springer-Verlag, New York/Berlin, 1982), p. 507.
29. D. W. Riggins, R. W. Walters, and D. Pelletier, AIAA Paper 88-0229, January 1988 (unpublished).
30. V. Venkatakrishnan, *AIAA J.* **27** (7), 885 (1989); AIAA Paper 88-0413.
31. R. M. Beam and R. F. Warming, *AIAA J.* **16**, 393 (1978).
32. S. Yoon and A. Jameson, AIAA Paper 87-0600, January 1987 (unpublished).
33. J. L. Thomas and R. W. Walters, *AIAA J.* **25** (4), 527 (1987).
34. R. W. Walters, D. L. Dwoyer, and H. A. Hassan, *AIAA J.* **24** (1), 6 (1986).
35. K. Ajmani, W. F. Ng, and M. S. Liou, in *Proceedings, AIAA 10th Computational Fluid Dynamics Conference, June 24–26, 1991, Honolulu, Hawaii*; AIAA Paper 91-1556.
36. V. Venkatakrishnan, *AIAA J.* **29** (7), 1092 (1991); AIAA Paper 90-0586.
37. R. E. Smith, Jr. and I. Sadrehaghighi, in *Proceedings 4th International Symposium of Computational Fluid Dynamics, September 9–12, 1991, University of California-Davis*, p. 1071.
38. I. Sadrehaghighi, R. E. Smith, Jr., and S. N. Tiwari, AIAA Paper 92-0660, January 1992 (unpublished).
39. J. L. Thomas and M. D. Salas, *AIAA J.* **24** (7), 1074 (1986); AIAA Paper 85-0020.
40. A. C. Taylor III, W. F. Ng, and R. W. Walters, *J. Comput. Phys.* **99**, (1), 159 (1192); *Proceedings, AIAA 9th Computational Fluid Dynamics Conference, June 13–15, 1989, Buffalo, New York*; AIAA Paper 89-1954.
41. B. Baldwin and H. Lomax, AIAA Paper 78-0257, January 1978 (unpublished).