AIAA JOURNAL
Vol. 43, No. 9, September 2005

Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities

Jeffrey P. Thomas,* Kenneth C. Hall,† and Earl H. Dowell‡
Duke University, Durham, North Carolina 27708-0300

A discrete adjoint approach is presented for computing steady and unsteady aerodynamic design sensitivities for compressible viscous flows about airfoil configurations. The nominal flow solver method is based on a harmonic balance solution technique, which is capable of modeling both steady and nonlinear periodic unsteady flows. The computer code for the discrete adjoint solver, which is derived from the nominal harmonic balance solver, has been generated with the aid of the advanced automatic differentiation software tool known as TAF (Transformation of Algorithms in FORTRAN).

Nomenclature

c	=	airfoil chord length
c_l, c_d, c_m	=	coefficient of lift, drag, and moment about elastic axis, respectively
$\hat{c}_{l_n}, \hat{c}_{d_n}, \hat{c}_{m_n}$	=	nth harmonic of unsteady lift, drag, and moment coefficient about elastic axis, respectively
h, α	=	airfoil plunge and pitch coordinate degree of freedom, respectively
$\hat{h}_n, \hat{\alpha}_n$	=	nth harmonic of airfoil plunge and pitch amplitude, respectively
J	=	cost function
M	=	number of dependent variables for computational fluid dynamics (CFD) model
M_∞	=	freestream Mach number
$\boldsymbol{N}$	=	vector operator defined by residual of discrete harmonic balance CFD solver
N_H	=	number of harmonics used in harmonic balance CFD flow solver method
$\boldsymbol{Q}$	=	vector representing discrete harmonic balance CFD flow solution
Re, Im	=	real and imaginary part, respectively
Re_∞	=	freestream Reynolds number
U_∞	=	freestream velocity
$\boldsymbol{x}$	=	vector representing computational mesh
$\omega, \bar{\omega}$	=	frequency and reduced frequency based on airfoil chord, $\bar{\omega} = \omega c / U_\infty$, respectively

Subscripts and Superscripts

T	=	transpose
$0, 1, \ldots, N_H$	=	zeroth, first, ..., N_Hth harmonic, respectively

Introduction

PRESENTED is a discrete adjoint method for modeling both steady and periodic unsteady aerodynamic design sensitivities for compressible viscous flows about airfoil configurations. Design sensitivities of airfoil steady or unsteady loading (e.g., lift, drag, or moment coefficient) with respect to arbitrary changes in airfoil shape can be rapidly determined with the methodology.

The discrete fluid-dynamic model is based on a modern high-fidelity harmonic-balance (HB)[1–3] computational fluid dynamics (CFD) solver for the Reynolds-averaged Navier–Stokes (RANS) equations. An advanced automatic differentiation software tool known as TAF (Transformation of Algorithms in FORTRAN)[4,5] is used to generate the necessary forward (tangent) and reverse (adjoint) partial gradient computer code required for implementing the discrete adjoint method for computing design sensitivities. Furthermore, for our particular HB/CFD flow solver, we have found that TAF can generate adjoint gradient code with a computational cost roughly a factor of three times the computational cost of the nominal solver.

Our ultimate objective is to develop a computational tool that will allow one to rapidly model aeroelastic design sensitivities such as the change in flutter onset dynamic pressure as a result of arbitrary changes in configuration geometry. A first step toward this goal is developing the computational methodology that will allow one to compute the change in the unsteady aerodynamic forces acting on a configuration as a results of arbitrary changes in geometric design. Modern compressible and viscous CFD methods are typically a necessity for accurately modeling unsteady aerodynamics for transonic Mach numbers, and it is for transonic Mach numbers that flutter onset is often a major concern. It is also often more convenient to do a flutter onset analysis when working in the frequency domain. In the frequency domain, one can typically solve directly for the flutter onset velocity and frequency, whereas in the time domain one must typically time march a number of time-domain solutions for different velocities to determine the velocity at which divergence occurs. Working in the frequency domain however requires a knowledge of the unsteady aerodynamics in the frequency domain. This is where an unsteady frequency domain version of the nominal CFD solver is useful. Such a solver can be used to directly compute the unsteady frequency-domain aerodynamic loading acting on a configuration. The harmonic-balance method provides a convenient method for creating such a frequency domain unsteady solver about an existing steady or time-domain unsteady CFD method. An automatic differentiation tool such as TAF can then be used to create the necessary computer code for the tangent and adjoint gradients of the nominal HB/CFD solver residual and chosen cost function as required for the overall discrete adjoint sensitivity technique.

After the necessary tangent and adjoint gradient code has been generated, one can proceed to determine design sensitivities as follows. First, using the nominal HB/CFD flow solver, one computes the steady or unsteady flow for the Mach number, mean angle of attack, Reynolds number, and if the flow is unsteady, the reduced frequency, and unsteady motion of interest. Next, one computes the

Presented as Paper 2003-0041 at the AIAA 41st Aerospace Sciences Meeting, Reno, NV, 6–9 January 2003; received 7 February 2003; revision received 2 March 2005; accepted for publication 4 March 2005.

*Research Assistant Professor, Department of Mechanical Engineering and Materials Science. Senior Member AIAA.

†Professor and Department Chairperson, Department of Mechanical Engineering and Materials Science. Associate Fellow AIAA.

‡J. A. Jones Professor, Department of Mechanical Engineering and Materials Science, and Dean Emeritus, School of Engineering. Fellow AIAA.

solution for the adjoint variables corresponding to the nominal flow solution and the cost function(s) of interest. Finally, with the adjoint solution variables known, one can determine sensitivities of the chosen cost function with respect to arbitrary changes in the shape of the airfoil for virtually no cost. So for the computational expense of roughly three nominal HB/CFD flow solutions, one can compute an arbitrary number of design sensitivities. By contrast, the alternative numerical approach would be a finite difference method, where the overall cost would be proportional to the number of sensitivities desired.

The continuous and discrete adjoint methods are the two names commonly given to the two approaches that one can take in the development of an adjoint-based sensitivity technique. In the context of a CFD flow solver method, both are in fact discrete techniques. The difference is in how the adjoint operators are formulated. In the continuous approach, one proceeds by applying a CFD algorithm to the adjoint form of the governing equations, whereas in the discrete method one derives the adjoint operator directly from the original discrete CFD operator. The discrete method is in general considered more accurate because one is working directly from the nominal CFD method. However studies are currently underway to determine how much of a difference there is (for example, see Nadarajah and Jameson[6]).

Nielsen and Anderson[7,8] and colleagues at NASA Langley are pioneers in the discrete adjoint approach for large unstructured CFD flow solver methods. Nielsen and Anderson have derived their adjoint code by hand, which can be very challenging especially for a large CFD flow solver code. For our work, we have chosen to take advantage of recent advances that have been made in the field of automatic differentiation to automatically generate the necessary tangent and adjoint code of the nominal harmonic-balance CFD flow solver residual and chosen cost function.

Florea and Hall[9] and Giles et al.[10] have also recently studied using the discrete adjoint approach in the context of small disturbance unsteady CFD solvers for flows about two-dimensional airfoils and in turbomachinery configurations. Nadarajah et al.[11] have also recently reported a discrete adjoint approach for a harmonic-balance version of an inviscid Euler solver. In the following, the discrete adjoint method is applied to a viscous RANS HB/CFD solver with a complex turbulence model.

Theory

Harmonic-Balance CFD Method

The harmonic-balance flow solver technique provides an efficient method for modeling nonlinear aerodynamic effects for periodic unsteady finite-amplitude airfoil motions of a prescribed frequency. As is discussed in Refs. 1–3, the HB method can be easily implemented within the framework of an existing steady or time-domain unsteady CFD solver. The vector of unsteady flow conservation variables $\boldsymbol{U}(\boldsymbol{x}_i, t)$ at each computational mesh point $\boldsymbol{x}_i$ is approximated in a truncated Fourier expansion as

$$\boldsymbol{U}(\boldsymbol{x}_i, t) \approx \sum_{n=-N_H}^{N_H} \hat{\boldsymbol{U}}_n(\boldsymbol{x}_i) e^{jn\omega t} \tag{1}$$

where $\hat{\boldsymbol{U}}_n(\boldsymbol{x}_i)$ is the nth harmonic coefficient. McMullen et al.[12,13] are also investigating a similar expansion technique for unsteady Euler and Navier–Stokes flows.

In what we refer to as the "standard" harmonic-balance approach, Eq. (1) is substituted into the governing flow equations whereby one then goes through the process of "balancing" all of the resulting terms proportional to $e^{jn\omega t}$ for each n $(-N_H \le n \le N_H)$. This in turn yields $2N_H + 1$ equations for the $2N_H + 1$ harmonic coefficients $\hat{\boldsymbol{U}}_n$. This straightforward approach to the harmonic-balance formulation is however typically difficult to implement for complex systems of equations such as the Euler and Reynolds-averaged Navier–Stokes equations.

Hall et al.[1] however recently devised an alternative approach to the standard harmonic-balance approach whereby the method is formulated in terms of time-domain variables. That is, instead of working in terms of the Fourier coefficient variables $\hat{\boldsymbol{U}}_n(\boldsymbol{x}_i)$, one instead considers as dependent variables the flow solution stored at $2N_H + 1$ equally spaced subtime levels $[\boldsymbol{U}(\boldsymbol{x}_i, t_n)]$ over the time period of one cycle of unsteady motion. The Fourier and time-domain variables are related to one another via a constant Fourier transformation matrix.

Working in terms of subtime level variables circumvents the necessity of having to go through the balancing step of the standard harmonic balance method formulation, and in fact the method allows one to easily formulate the harmonic-balance technique within the framework of an existing steady or time-domain unsteady CFD solver (see Refs. 1–3 for further details). Furthermore, if one wishes to use the technique for modeling a steady flow, one simply uses zero harmonics ($N_H = 0$) in the method, whereby the nominal steady CFD method is recovered. For the results presented here, the two-dimensional RANS HB/CFD method is based on a variant of the standard Lax–Wendroff scheme[14,15] in conjunction with the one-equation turbulence model of Spalart and Allmaras.[16]

Harmonic-Balance CFD Method Operator

Once constructed, the iterative HB/CFD solver can be viewed as a large vector operator of the form

$$\boldsymbol{\Delta Q}^m = \boldsymbol{Q}^{m+1} - \boldsymbol{Q}^m = \boldsymbol{N}(\boldsymbol{Q}^m, \boldsymbol{x}) \tag{2}$$

where $\boldsymbol{N}(\boldsymbol{Q}^m, \boldsymbol{x})$ is a vector representing one iteration of the residual operator of the nominal HB/CFD method.

To achieve solution convergence, one marches Eq. (2) in an iterative fashion until $\|\boldsymbol{\Delta Q}^m\| \approx 0$. More precisely, at convergence for an unsteady flow case the residual of the nominal HB/CFD solver represents

$$\boldsymbol{N}(\boldsymbol{Q}(\boldsymbol{x}), \boldsymbol{x}, N_H, \bar{\omega}, \hat{h}_n, \hat{\alpha}_n) \approx 0, \qquad n = 0, 1, \ldots, N_H \tag{3}$$

Discrete Adjoint Method Formulation for the HB/CFD Method

Let $J(\boldsymbol{Q}, \boldsymbol{x})$ be some cost function, for example, steady lift, drag, or moment coefficient $(\hat{c}_{l_0}, \hat{c}_{d_0}, \hat{c}_{m_0})$, or first harmonic unsteady lift, drag, or moment coefficient $(\hat{c}_{l_1}, \hat{c}_{d_1}, \hat{c}_{m_1})$. The objective of the discrete adjoint method sensitivity technique is to enable one to rapidly determine sensitivities of the chosen cost function J with respect to changes in the geometry of the configuration (or the computational mesh $\boldsymbol{x}$ in this case). The constraint is that the residual equation of the nominal discrete HB/CFD operator must be zero, that is, Eq. (3), $\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x}) = 0$ must be satisfied.

For the discrete adjoint method approach, one proceeds by introducing a vector of adjoint variables $\boldsymbol{\lambda}$. Because $\boldsymbol{N} = 0$, one can add the vector product of $\boldsymbol{\lambda}^T$ and $\boldsymbol{N}$, which is also equal to zero (scalar zero), to the cost function J. That is,

$$J = J + \boldsymbol{\lambda}^T \boldsymbol{N} \tag{4}$$

With $\boldsymbol{N}$ and J both being functions of $\boldsymbol{Q}$ and $\boldsymbol{x}$, one can then implicitly differentiate Eq. (4) to obtain

$$\mathrm{d}J = \left(\frac{\partial J}{\partial \boldsymbol{Q}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}}\right) \mathrm{d}\boldsymbol{Q} + \left(\frac{\partial J}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}}\right) \mathrm{d}\boldsymbol{x} \tag{5}$$

The main objective of the discrete adjoint method is to then determine a specific $\boldsymbol{\lambda}$ such that the term multiplied by $\mathrm{d}\boldsymbol{Q}$ in Eq. (5) equals zero, that is,

$$\frac{\partial J}{\partial \boldsymbol{Q}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}} = 0^T \tag{6}$$

or

$$\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}}^T \boldsymbol{\lambda} = -\frac{\partial J}{\partial \boldsymbol{Q}}^T \tag{7}$$

With a $\boldsymbol{\lambda}$ satisfying Eq. (7), the design sensitivity is then simply

$$\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{x}} = \frac{\partial J}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{x}} \tag{8}$$

Automatic Differentiation Software Tool

As for the required tangent and adjoint partial gradient code, there are four different quantities that one needs to be able to compute in order to carry out the discrete adjoint method for computing design sensitivities. These four quantities are as follows: 1) the partial gradient of the cost function J with respect to the computational grid $\boldsymbol{x}$, $(\partial J/\partial \boldsymbol{x})$; 2) the partial gradient of the cost function J with respect to the flow solution variables $\boldsymbol{Q}$, $(\partial J/\partial \boldsymbol{Q})$; 3) the partial gradient of the flow solver residual $\boldsymbol{N}$ with respect to the design variables $\boldsymbol{x}$, $(\partial \boldsymbol{N}/\partial \boldsymbol{x})$; and 4) the vector representing the adjoint of the matrix corresponding to the gradient of flow solver residual $\boldsymbol{N}$ with respect to the flow solution variables $\boldsymbol{Q}$ (i.e., $[\partial \boldsymbol{N}/\partial \boldsymbol{Q}]^T$) multiplied by the vector of the adjoint solution variables $\boldsymbol{\lambda}$ ($[\partial \boldsymbol{N}/\partial \boldsymbol{Q}]^T \boldsymbol{\lambda}$).

As mentioned in the Introduction, the automatic differentiation software tool that we have utilized to generate the computer code for computing these four quantities for the HB/CFD solver is known as TAF.[4,5] In the following, we provide a brief explanation as to how one uses TAF to generate the computer code representing tangent or adjoint gradient of a computer algorithm (see www.FastOp.de for additional details).

To demonstrate, in this instance, we consider the case where one is interested in gradient of the flow solver solution residual vector $\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x})$ with respect to the flow solution variables $\boldsymbol{Q}$. One can determine the gradient of the scalar cost function J in a similar fashion. The TAF compiler operates on a computer server at the headquarters of FastOpt in Germany. After purchasing a usage license for TAF, FastOpt provides a computer shell script for running TAF. This shell script handles the transfer of instructions and source code files to the TAF compiler in Germany. One can in fact invoke this shell script from within the "Makefile" of the code that one is interested in differentiating. For the case of determining the gradient of the flow solver solution residual vector $\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x})$ with respect to the flow solver solution variables $\boldsymbol{Q}$, as arguments to the TAF shell script, one first specifies the name of the top-level subroutine representing one iteration of the governing CFD algorithm, that is, the computation of $\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x})$. Next, one specifies the names of the variables in the computer source code that represent the algorithm dependent $[\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x})]$ and the independent variables $(\boldsymbol{Q})$, respectively. The user then specifies whether they desire the computer code for the forward (tangent) or reverse (adjoint) partial gradient. The user also specifies a list of all of the source code computer files that comprise the computation of the CFD solver residual $\boldsymbol{N}(\boldsymbol{Q}, \boldsymbol{x})$.

If one specifies forward (tangent) mode, TAF then returns the computer code for the computation of the vector

$$\Delta \boldsymbol{n}_t = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}}\right] \Delta \boldsymbol{q}_t$$

where $\Delta \boldsymbol{n}_t$ and $\Delta \boldsymbol{q}_t$ are the vectors representing the forward partial gradient independent and dependent variables, respectively. Similarly, for the reverse (adjoint) mode TAF generates the computer code for the computation of the vector

$$\Delta \boldsymbol{n}_a = \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}}\right]^T \Delta \boldsymbol{q}_a$$

where $\Delta \boldsymbol{n}_a$ and $\Delta \boldsymbol{q}_a$ are the vectors representing the reverse partial gradient independent and dependent variables, respectively.

In a similar manner, one can generate the code representing the gradient of the cost function J with respect to the flow variables $\boldsymbol{Q}$ $(\partial J/\partial \boldsymbol{Q})$, the gradient of the cost function J with respect to the grid $\boldsymbol{x}$ $(\partial J/\partial \boldsymbol{x})$, and the gradient of the flow solver residual $\boldsymbol{N}$ with respect to the grid $\boldsymbol{x}$ $(\partial \boldsymbol{N}/\partial \boldsymbol{x})$.

After all of the necessary partial gradient computer code is created, one can then compute the partial derivatives required for the overall discrete adjoint sensitivity method. For instance, the mth element of the $1 \times M$ (M being the total number of dependent variables of the CFD model) partial gradient vector $\partial J/\partial \boldsymbol{Q}$ can be determined via a single call to the TAF-generated computer code for $\partial J/\partial \boldsymbol{Q}$, where in this instance the mth independent gradient variable of the $\partial J/\partial \boldsymbol{Q}$ gradient code is set to one and all other independent gradient variables are set to zero. For a cost function J being a quantity such as airfoil lift, which depends only on the flow variable located on the surface of the airfoil, one then only needs to compute the partial derivatives of the vector $\partial J/\partial \boldsymbol{Q}$ that are associated with the flow variables located on the airfoil surface. The remaining partial derivatives of the vector $\partial J/\partial \boldsymbol{Q}$ will be zero.

The partial derivatives $\partial J/\partial \boldsymbol{x}$ and $\partial \boldsymbol{N}/\partial \boldsymbol{x}$ can be determined in the same way as $\partial J/\partial \boldsymbol{Q}$. However, if one knows the specific design change $\Delta \boldsymbol{x}$ of interest, only one call to the gradient code for $\partial J/\partial \boldsymbol{x}$ and $\partial \boldsymbol{N}/\partial \boldsymbol{x}$, respectively, is all that is necessary to compute $\partial J/\partial \boldsymbol{x}$ and $\partial \boldsymbol{N}/\partial \boldsymbol{x}$. In this case, the gradient code independent variables are simply set to the design change $\Delta \boldsymbol{x}$ of interest.

The adjoint solver, which consists of the generated tangent and adjoint code, thus represents the residual of Eq. (7). In a manner similar to how the nominal HB flow solver operates, one then marches the adjoint solver residual in an iterative fashion to determine the converged solution for the adjoint variables $\boldsymbol{\lambda}$. That is, one marches

$$\Delta \boldsymbol{\lambda}^m = \boldsymbol{\lambda}^{m+1} - \boldsymbol{\lambda}^m = \frac{\partial J}{\partial \boldsymbol{Q}}^T + \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{Q}}^T \boldsymbol{\lambda}^m \tag{9}$$

until $\|\Delta \boldsymbol{\lambda}^m\| \approx 0$.

Model Configuration

NLR 7301 Airfoil Section

To demonstrate the unsteady HB/CFD discrete adjoint sensitivity approach, for this preliminary analysis the configuration considered is the NLR 7301 supercritical airfoil section. The reason we have chosen this particular airfoil is that we eventually would like to be able to use the discrete adjoint method sensitivity technique to determine design sensitivities of aeroelastic quantities such as the flutter onset dynamic pressure or limit-cycle-oscillation (LCO) amplitude with respect to changes in airfoil design. Schewe et al.[17] have conducted transonic two-degree-of-freedom aeroelastic experimental studies for an NLR 7301 constant airfoil section wing model at various Mach numbers and angles of attack. In some instances, aeroelastic LCO was observed. Transonic experimental aeroelastic data are quite limited, especially for two-dimensional configurations, yet this is a two-dimensional model where experimental aeroelastic LCO data are available.

Computational Mesh

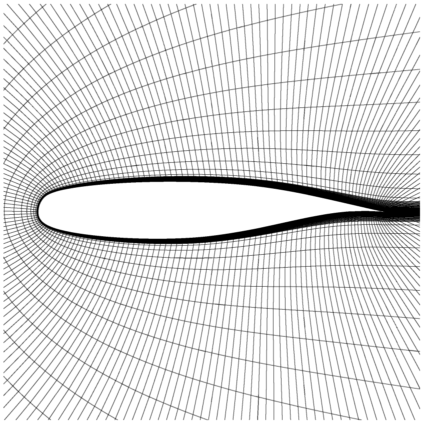
Figure 1 shows a sample viscous computational grid for the NLR 7301 airfoil configuration. A structured c-grid mesh topology is used, which consists of 49 mesh points radially and 191 mesh points circumferentially, with a total of 143 mesh points surrounding the airfoil surface, and the remainder of the circumferential mesh points being distributed in the wake. The outer boundary extends to a distance of 10 chord lengths from the center of the airfoil.

Results

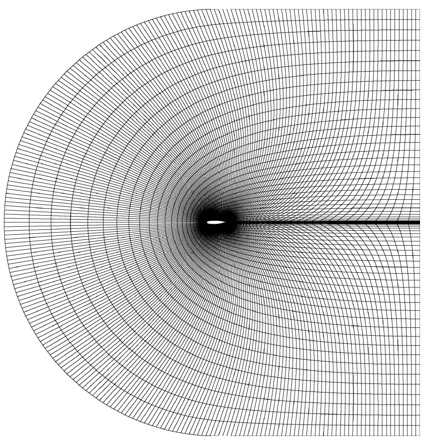
Nominal Solution

As mentioned in the Introduction, for the discrete adjoint method sensitivity procedure one first computes the nominal, in this case unsteady, HB/CFD flow solution for the case of interest. In this instance, the mean (zeroth-harmonic) flow conditions are set as $M_\infty = 0.75$, $\hat{\alpha}_0 = 0.4$ deg, and $Re_\infty = 1.727 \times 10^6$. After wind-tunnel interference effects are factored in, this flow condition is near the experimental wind-tunnel flow condition referred to by Schewe et al.[17] as "measured-point" condition MP77, where a very low-amplitude LCO was observed in the experiment.

For the unsteadiness, we consider a small-amplitude unsteady first harmonic pitch motion of $\hat{\alpha}_1 = 0.001$ deg about the quarter-chord. This small amplitude is chosen to simulate linear small disturbance unsteady aerodynamics, which is a requirement for a linear flutter onset aeroelastic analysis. Because the amplitude of the unsteadiness is very small, only one harmonic $(N_H = 1)$ is used for the HB/CFD solver. Figure 2 shows computed mean flow Mach contours.



a) Mesh close-up



b) Mesh overall

Fig. 1 Computational grid used for the viscous NLR 7301 airfoil configuration.

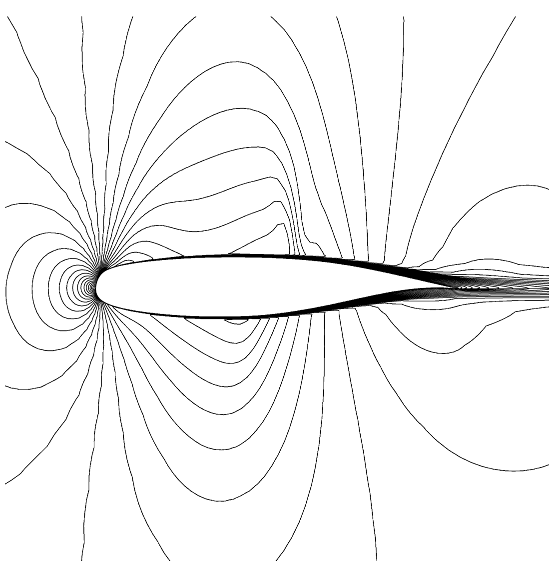


Fig. 2 Mean flow Mach contours for viscous NLR 7301 airfoil configuration.

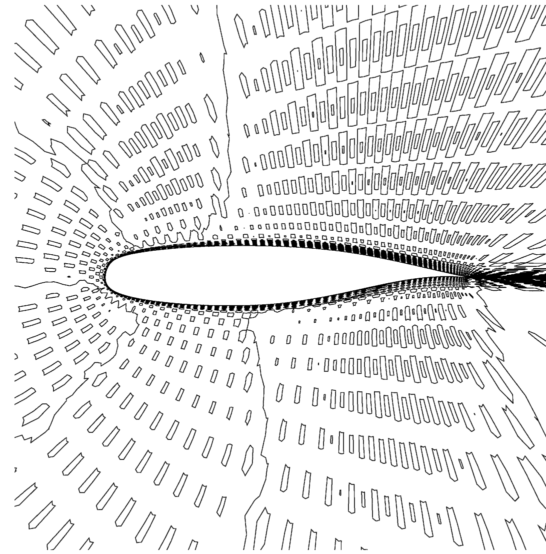


Fig. 3 Adjoint solution contours for viscous NLR 7301 airfoil configuration.

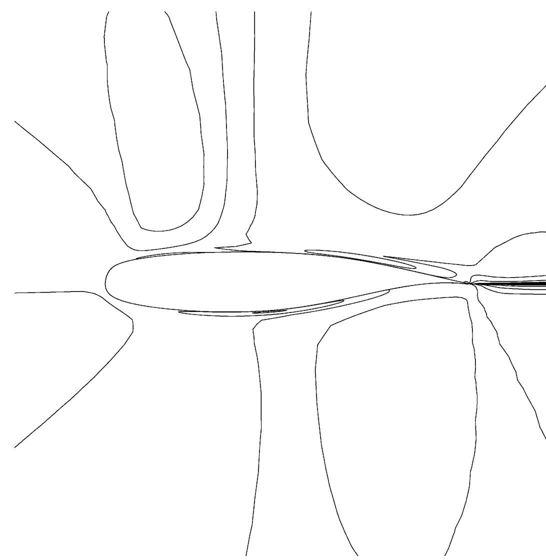


Fig. 4 Adjoint solution contours for viscous NLR 7301 airfoil configuration when running nominal and adjoint solvers without multigrid.

Adjoint Solution

Figure 3 shows the computed adjoint solution [Eqs. (7) and (9)] for the adjoint variable corresponding to the total energy equation of the first subtime level of the nominal HB/CFD solver. For demonstration purposes, the cost function is chosen to be the imaginary part of the unsteady moment coefficient normalized by the unsteady pitch amplitude ($\mathrm{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1$). The cost function can, however, be any one of the forces (i.e., lift, drag, or moment), for any harmonic of interest (e.g., the zeroth, first, second, etc.), and for either the real or imaginary part of harmonics higher than the zeroth harmonic.

The checkerboard pattern for the adjoint solution contours is because of the multigrid convergence acceleration routines of the nominal solver. Figure 4 shows the computed adjoint solution contours when the nominal and adjoint solvers are run without multigrid.

The reason for the difference between the no-multigrid and multigrid adjoint solutions is because the underlying Lax–Wendroff CFD method used in the present study has a slight dependence on the pseudo-time-step sizes used for marching steady-state problems, which is the case for the harmonic-balance frequency-domain method, to convergence. Because the time-step sizes change during a multigrid sweep, the final no-multigrid and multigrid nominal harmonic-balance solutions will differ slightly, and, as such, so also will the adjoint solutions based on the separate no-multigrid and multigrid nominal harmonic-balance solutions. The nice aspect of the discrete adjoint method is that it is able to account for this time-step dependency of the Lax–Wendroff method. The finite difference and sensitivity comparison results shown in the rest of this paper are for the multigrid version of the Lax–Wendroff CFD method.

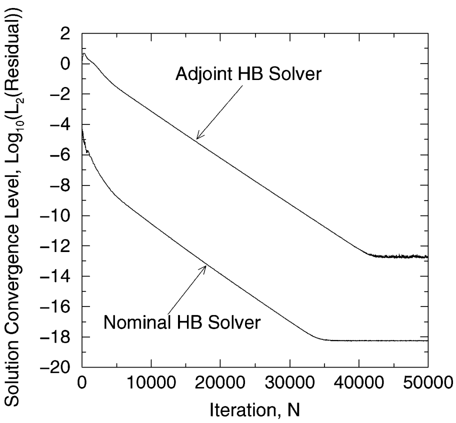


Fig. 5 Nominal HB solver and adjoint HB solver convergence histories.

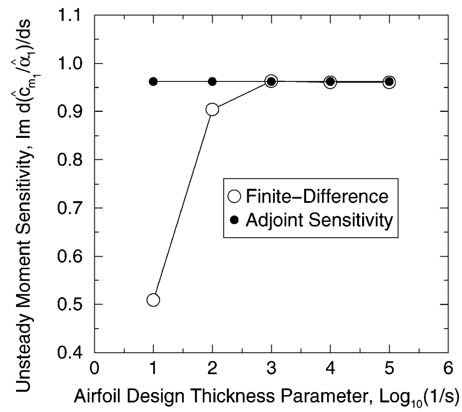


Fig. 6 Comparison of forward finite difference and discrete adjoint method computed design sensitivities for the imaginary part of the first harmonic unsteady moment coefficient because of change in airfoil thickness.

Solution Convergence Histories

Figure 5 shows the solution residual convergence histories for both the nominal and adjoint solvers. One can see that slopes of these curves, as they approach their respective double-precision convergence limits, are the same. This is in fact a way that one can get an initial sense if their adjoint solver is set up correctly. The adjoint and nominal solvers should have the same convergence rates as their residuals go to machine zero.

Sensitivity Computation

Once the adjoint solution has been computed, one can then rapidly compute design sensitivities using Eq. (8). To demonstrate, in this instance, we calculate the sensitivity of $\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1$ caused by a change in the airfoil thickness. We also compare this discrete adjoint method sensitivity gradient to a finite difference gradient computation.

More specifically, we compute the derivative of $\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1$ with respect to a design variable we have denoted as s, which governs the thickness of the airfoil. In this instance, the upper $[z_u(x)]$ and lower $[z_l(x)]$ airfoil section surface z coordinates are given by

$$z_u(x) = (1+s)z_{u_{\text{nom}}}(x), \qquad z_l(x) = (1+s)z_{l_{\text{nom}}}(x)$$

where the x coordinate is measured along the chord of the airfoil section and $z_{u_{\text{nom}}}(x)$ and $z_{l_{\text{nom}}}(x)$ are the upper and lower airfoil section z coordinates for the nominal airfoil section design. Our objective is to compute the derivative

$$\frac{\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)}{\mathrm{d}s}$$

Our discrete adjoint solver is set up to determine design sensitivities as a result of changes in the computational mesh $\boldsymbol{x}$. Thus in order to evaluate the derivative $\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)/\mathrm{d}s$, we compute the product of the derivatives

$$\frac{\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)}{\mathrm{d}\boldsymbol{x}}, \qquad \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s}$$

that is,

$$\frac{\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)}{\mathrm{d}s} = \frac{\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)}{\mathrm{d}\boldsymbol{x}}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s}$$

As mentioned in the preceding sections, the adjoint computer code generated by TAF enables one to compute the gradient of a chosen cost function with respect to changes in the computational mesh, which will also change when the design of the airfoil changes. This is a nice feature in that one can then determine the sensitivity for any arbitrary change in the geometry based on the new computational grid, which reflects the change in the airfoil geometry.

The derivative $\mathrm{d}(\text{Im}\,\hat{c}_{m_1}/\hat{\alpha}_1)/\mathrm{d}\boldsymbol{x}$ is based on the discrete adjoint methodology. The derivative $\mathrm{d}\boldsymbol{x}/\mathrm{d}s$, on the other hand, is approximated using a forward finite difference, that is,

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} \approx \frac{\boldsymbol{x}(s)-\boldsymbol{x}(0)}{s} \qquad \text{for small } s \tag{10}$$

where $\boldsymbol{x}(s)$ is an alternate computational mesh based on a small value of s. In the limit as s goes to zero, the finite difference approximation of Eq. (10) will approach the analytic value for the derivative $\mathrm{d}\boldsymbol{x}/\mathrm{d}s$. In this particular study, by design we have constructed the alternate meshes $\boldsymbol{x}(s)$ such that

$$\boldsymbol{x}(s) = \boldsymbol{x}(0) + s\tau$$

that is, the alternate meshes $\boldsymbol{x}(s)$ are linearly proportional to s, so that $\mathrm{d}\boldsymbol{x}/\mathrm{d}s = \tau$, even via a finite difference computation.

Finally, Fig. 6 shows a comparison of forward finite difference and discrete adjoint method design sensitivities for different magnitudes of airfoil thickness parameter s. As can be seen, for values for s greater than 0.001, the finite difference sensitivities demonstrate that there are nonlinear effects brought about by such large changes in airfoil design thickness. However, for values of s less than 0.001, it can be seen that the finite difference value for the derivative remains nearly constant and that the computed adjoint method value for this derivative, which is a constant, matches very well to the finite difference value.

Conclusions

Presented is a discrete adjoint method for modeling steady and periodic unsteady aerodynamic design sensitivities. The flow solver is based on an unsteady nonlinear harmonic balance technique for the Reynolds-averaged Navier–Stokes equations. The source code for the adjoint solver is generated from the nominal harmonic balance flow solver with the aid of the automatic differentiation compiler TAF (Transformation of Algorithms in FORTRAN). The resulting adjoint solver has a computational cost of roughly three times a nominal solution. Once the adjoint solution is determined, one can then obtain steady or periodic unsteady aerodynamic design sensitivities for a large number of design changes at essentially no additional computational cost. The method has been demonstrated for design sensitivities of unsteady loading for transonic viscous flow about the NLR 7301 supercritical airfoil section.

Acknowledgments

The authors thank Ralf Giering and Thomas Kaminski for their efforts in developing the TAF adjoint compiler, which has made the research presented in this paper possible.

References

[1]Hall, K. C., Thomas, J. P., and Clark, W. S., "Computation of Unsteady Nonlinear Flows in Cascades Using a Harmonic Balance Technique," *AIAA Journal*, Vol. 40, No. 5, 2002, pp. 879–886.

[2]Thomas, J. P., Dowell, E. H., and Hall, K. C., "Nonlinear Inviscid Aerodynamic Effects on Transonic Divergence, Flutter and Limit Cycle Oscillations," *AIAA Journal*, Vol. 40, No. 4, 2002, pp. 638–646.

[3]Thomas, J. P., Dowell, E. H., and Hall, K. C., "Modeling Viscous Transonic Limit Cycle Oscillation Behavior Using a Harmonic Balance Approach," *Journal of Aircraft*, Vol. 41, No. 6, 2004, pp. 1266–1274.

[4]Giering, R., and Kaminski, T., "Recipes for Adjoint Code Construction," *Association for Computing Transactions on Mathematical Software*, Vol. 24, No. 4, 1998, pp. 437–474.

[5]Giering, R., Kaminski, T., and Slawig, T., "Generating Efficient Derivative Code with TAF: Adjoint and Tangent Linear Euler Flow Around an Airfoil," *Future Generation Computer Systems* (to be published).

[6]Nadarajah, S. K., and Jameson, A., "Studies of the Continuous and Discrete Adjoint Approaches to Viscous Automatic Aerodynamic Shape Optimization," AIAA Paper 2001-2530, 2001.

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

[8]Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," AIAA Paper 2001-0596, Jan. 2001.

[9]Florea, R., and Hall, K. C., "Sensitivity Analysis of Unsteady Inviscid Flow Through Turbomachinery Cascades," *AIAA Journal*, Vol. 39, No. 6, 2001, pp. 1047–1057.

[10]Giles, M. B., Duta, M. C., Müller, J. D., and Pierce, N. A., "Algorithm Developments for Discrete Adjoint Methods," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 198–205.

[11]Nadarajah, S. K., McMullen, M. S., and Jameson, A., "Optimum Shape Design for Unsteady Flows Using Time Accurate and Non-Linear Frequency Domain Methods," AIAA Paper 2003-3875, 2003.

[12]McMullen, M., Jameson, A., and Alonso, J. J., "Acceleration of Convergence to Period Steady State in Turbomachinery Flows," AIAA Paper 2001-0152, Jan. 2001.

[13]McMullen, M., Jameson, A., and Alonso, J. J., "Application of a Non-Linear Frequency Domain Solver to the Euler and Navier–Stokes Equations," AIAA Paper 2002-0120, Jan. 2002.

[14]Ni, R., "A Multiple Grid Scheme for Solving the Euler Equations," *AIAA Journal*, Vol. 20, No. 11, 1982, pp. 1565–1571.

[15]Saxor, A. P., "A Numerical Analysis of 3-D Inviscid Stator/Rotor Interactions Using Non-Reflecting Boundary Conditions," Gas Turbine Lab., Massachusetts Inst. of Technology, Rept. 209, Cambridge, MA, March 1992.

[16]Spalart, P. R., and Allmaras, S. R., "A One Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, Jan. 1992.

[17]Schewe, G., Mai, H., and Dietz, G., "Nonlinear Effects in Transonic Flutter with Emphasis on Manifestations of Limit Cycle Oscillations," *Journal of Fluids and Structures*, Vol. 18, No. 1, 2003, pp. 3–22.

E. Livne
Associate Editor