JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 20, No. 6, November–December 1997

Continuation Method for Calculation of Transonic Airfoil Flutter Boundaries

Philip S. Beran*
U.S. Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433-7765
and
Scott A. Morton†
U.S. Air Force Wright Laboratory, Wright–Patterson Air Force Base, Ohio 45433-7913

Flutter boundaries, through variations in aerodynamic and structural dynamic parameters, are computed for a NACA 64A006 airfoil at transonic freestream Mach numbers with pitch-and-plunge structural coupling. These flutter boundaries are obtained with a new continuation algorithm that specifically accounts for flutter onset. The algorithm, which is based on pseudoarclength continuation, solves an extended form of the Euler equations for the reduced velocity at flutter onset, the corresponding base-flow solution, the destabilizing mode at flutter onset, and the critical value of a secondary structural parameter, e.g., structural damping. The algorithm provides a relatively automated procedure for computing flutter boundaries directly, which avoids a trial-and-error, flutter-detection process typical of time-domain techniques. The current method is validated through comparison with independent flutter-boundary computations. Grid-sensitivity studies are carried out to assess the numerical accuracy of computed solutions.

Nomenclature

b, c	= airfoil half-chord and chord
C_l, C_m	= section lift and moment coefficients
D_α, D_h	= pitch and vertical axis spring damping coefficients
h	= vertical displacement (positive up)
I_α	= pitch axis moment of inertia
K_α, K_h	= pitch and vertical axis spring stiffness coefficients
m	= airfoil mass per unit span
p	= static pressure
r_α	= radius of gyration, $\sqrt{(I_\alpha/mb^2)}$
T	= temperature
t	= time
u, v	= velocity components in airfoil frame
$\bar{u}$	= reduced velocity, $V_\infty/(b\omega_\alpha)$
V_∞	= freestream velocity
x, y	= spatial coordinates in airfoil frame
α_{EQ}	= equilibrium angle of attack
Δ_t	= time step
ζ_α, ζ_h	= pitch and vertical axis structural damping ratios, $D_\alpha/[2\sqrt{(I_\alpha K_\alpha)}]$ and $D_h/[2\sqrt{(mK_h)}]$
Θ	= frequency of destabilizing eigenmode
μ_s	= airfoil-fluid mass ratio, $m/(\pi\rho_\infty b^2)$
ξ, η	= body conformal curvilinear coordinates
ρ	= density
ω_α, ω_h	= pitch and vertical axis natural frequencies, $\sqrt{(K_\alpha/I_\alpha)}$ and $\sqrt{(K_h/m)}$
ω_{SUR}	= underrelaxation parameter (<1)

Subscripts

ea	= referenced to elastic axis
i, j	= location of grid point
∞	= freestream quantity

Superscripts

n	= time level for unsteady quantity
ν	= iteration level (nontime)

Received Nov. 19, 1996; presented as Paper 97-0574 at the AIAA 35th Aerospace Sciences Meeting, Reno, NV, Jan. 6–9, 1997; revision received April 10, 1997; accepted for publication April 30, 1997. This paper is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

*Associate Professor, Department of Aeronautics and Astronautics, AFIT/ENY, 2950 P Street, Building 640. Senior Member AIAA.

†Captain, U.S. Air Force; Computational Aerodynamicist, Computational Fluid Dynamics Research Branch, WL/FIMC, 2645 Fifth Street, Building 450, Suite 7. Senior Member AIAA.

Introduction

ACCURATE and efficient determination of flutter boundaries is an important aspect of multidisciplinary design. These boundaries can be estimated through time integration of the governing aeroelastic equations. However, in a design environment, where a large number of configurations require examination and the location of flutter boundaries is unknown, time integration is unsystematic and prohibitively inefficient. An efficient, direct method of computing flutter boundaries for transonic flows would be of considerable value as a design tool and would strongly augment existing time-integration methods.

Morton and Beran[1,2] developed, implemented, and validated a direct method for the calculation of flutter points in two-dimensional transonic flow. The implicit methodology, which is more extensively documented in Ref. 3, provides the ability to compute directly the critical speed at which the fluid–structure system loses stability to periodic motion, i.e., limit-cycle oscillation. Using a direct method, the critical speed is precisely computed simultaneously with the computation of a single aeroelastic solution, thereby avoiding the trial-and-error application of a time-integration method to bracket the flutter-onset speed. With airfoil grids of modest size, Morton and Beran[2] demonstrated significant improvements in computational performance of the new, direct approach over an associated explicit, total variation diminishing scheme for flutter-speed prediction.

The majority of recent efforts in computational aeroelasticity have involved time integration of the transonic small-disturbance, Euler or Navier–Stokes equations. In the program CAP-TSD,[4] the transonic small-disturbance equations are integrated in time using approximate factorization for realistic aircraft configurations. A new version of this algorithm, CAP-TSDV,[5] exists for the aeroelastic analysis of wings in viscous, transonic flow and employs an interactive boundary-layer coupling method. Beyond potential-based methods, algorithms for simulating fully viscous flows about three-dimensional, aeroelastic configurations have been developed that are based on time integration of the Navier–Stokes equations. For example, there exist the finite difference algorithms ENSAERO[6] and ENS3DAE[7] and the finite element algorithm STARS.[8] Using similar time-integration stategies, techniques based on the Euler equations

have also been developed; an excellent review of such methods is given in Ref. 9.

Owing to the large computational requirements associated with computing unsteady flows about deforming three-dimensional configurations, recent work has been directed toward the modeling of unsteady aeroelastic behavior with relatively small numbers of important modes. This approach of reduced-order modeling, which has been led by Dowell,[10] is based on the efficient computation of eigenvalues and eigenvectors of a truncated system, ultimately derived through time integration of the governing equations for specified aerodynamic and structural dynamic conditions. In contrast to the calculation of a set of eigenmodes for specified flight conditions, direct methods compute the destabilizing eigenmode, the critical flight speed, and the corresponding aeroelastic state at flutter onset.

In constructing a flutter boundary, existing methods do not account for boundary shape. For example, in the computation of flutter boundaries, Ref. 2 used known bifurcating solutions at specific values of M_∞ as initial approximations to solutions at new values of M_∞. This approach is more efficient than using trivial initial approximations but does not account for local variation of critical solution behavior with freestream Mach number, a variation that can be rapid at subsonic freestream Mach numbers greater than about 0.85. As a result, small increments in M_∞ are required when tracing the flutter boundary in this regime, degrading the efficiency with which a boundary can be constructed. Time-integration methods suffer from this same initialization problem.

There are two primary objectives of this work. The first objective is to develop a technique for the efficient and direct computation of flutter boundaries. The second objective is to validate this procedure through comparison with other reported methods and through grid-sensitivity analyses. The procedure developed herein is an application of the pseudoarclength continuation (PAC) method[11] to an expanded system of equations satisfied at flutter onset. With PAC, system parameters are treated as variables that are computed using constraint equations as a solution path is traced. A new blocked-matrix procedure is developed to solve a reexpanded system consisting of the flutter equations and the PAC constraint equations. Although PAC is well suited to the computation of solution paths with folds,[12] in this paper we restrict attention to the new application of PAC to the computation of simple paths of flutter points. These paths will be traced through variation of structural parameters such as mass ratio.

Analysis

The mathematical model for a pitch-and-plunge airfoil in inviscid flow is briefly summarized in this section. The original algorithm for flutter-point computation implemented by Morton[3] is known as TVDntiAE, whereas the new numerical procedure will be referred to as BIFAE. Algorithmic modifications to TVDntiAE are briefly described, including PAC, the chord method, and single-stage flux evaluation. The grids used in this study are also reviewed. More complete treatments of the mathematical and discrete models are provided in Ref. 13.

The model of a rigid, pitch-and-plunge (typical section) airfoil is depicted in Fig. 1. Two reference frames are defined: an inertial reference system $(\hat{i}, \hat{j})$ and an airfoil-fixed reference system $(\hat{e}_1, \hat{e}_2)$. The airfoil is constrained to move along the vertical inertial axis and to rotate about an elastic axis. The aeroelastic equations are placed in nondimensional form using c, V_∞, and T_∞ as length, velocity, and temperature scales, respectively. One exception is the relative position of the elastic axis $x_\alpha \equiv r/b$, where r is the distance from the elastic axis to the center of gravity (negative when the elastic axis is positioned between the center of gravity and the leading edge). The nondimensional structural equations in pitch and plunge are as follows:

$$\ddot{h} + (x_\alpha/2)\ddot{\alpha} + 2\zeta_h c_2 \dot{h} + c_2^2 h = (2/\mu_s \pi) C_l \tag{1}$$

$$x_\alpha \ddot{h} + \frac{r_\alpha^2}{2}\ddot{\alpha} + \frac{2\zeta_\alpha r_\alpha^2}{\bar{u}}\dot{\alpha} + c_1(\alpha - \alpha_0) = \frac{4}{\mu_s \pi} C_{m_{\text{ea}}} \tag{2}$$

where $c_1 \equiv 2r_\alpha^2/\bar{u}^2$, $c_2 \equiv 2\omega_h/(\bar{u}\omega_\alpha)$, and α is the angle of attack between the freestream velocity vector and the airfoil chordline. The structural equations are taken to be linear; the vertical and torsional springs are modeled with constant stiffness coefficients K_h and K_α and constant damping coefficients D_h and D_α. The static pretwist α_0 defines the unloaded position of the torsional spring and is specified to vanish in this study. Equations (1) and (2) are rewritten as a system of first-order differential equations by setting $y_1 = h$, $y_3 = \alpha$, $\dot{y}_1 = y_2$, and $\dot{y}_3 = y_4$ with $S \equiv [y_1, y_2, y_3, y_4]^T$. Following Ref. 3, the system of equations is expressed as $S_t = \mathcal{K}(S, C_l, C_{m_{\text{ea}}}; \bar{u})$, where $\mathcal{K}$ is a linear function of S and $\bar{u}$ is treated as a parameter.

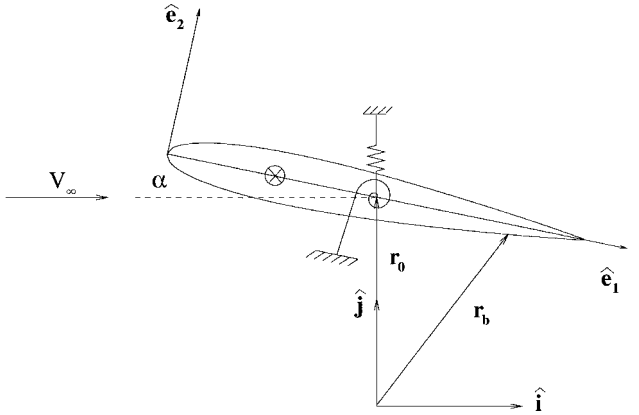

Fig. 1 Schematic of pitch-and-plunge airfoil.

The governing aerodynamic equations are the Euler equations, cast in nondimensional, strong-conservation form for a general curvilinear coordinate system. Following Morton,[3] the equations are expressed in terms of the array of conserved variables $\bar{U} \equiv [\rho, \rho u, \rho v, E_t]^T$:

$$\frac{\partial \hat{U}}{\partial t} + \frac{\partial \hat{E}(\bar{U})}{\partial \xi} + \frac{\partial \hat{F}(\bar{U})}{\partial \eta} = 0 \tag{3}$$

where $\hat{U} = \bar{U}/(\xi_x \eta_y - \xi_y \eta_x)$ and $\hat{E}$ and $\hat{F}$ are transformed flux arrays.[13] The aerodynamic equations are placed in discrete form using the upwind total variation diminishing (TVD) scheme of Harten[14] and Yee,[15] with a correction for grid-point motion. The discretization, of second-order spatial accuracy (away from extrema) and first-order temporal accuracy, is expressed as

$$\frac{\hat{U}_{i,j}^{n+1} - \hat{U}_{i,j}^{n}}{\Delta_t} = \tilde{\hat{F}}_{i,j-\frac{1}{2}}^{n} - \tilde{\hat{F}}_{i,j+\frac{1}{2}}^{n} + \tilde{\hat{E}}_{i-\frac{1}{2},j}^{n} - \tilde{\hat{E}}_{i+\frac{1}{2},j}^{n} \tag{4}$$

The arrays $\tilde{\hat{E}}$ and $\tilde{\hat{F}}$ are modified numerical fluxes that implement the second-order TVD formulation. Details of their construction are provided in Refs. 3 and 13. Previous work reported by Morton and Beran[1,2] employs the numerical fluxes in a two-stage, spatially split integration of $\hat{U}$, in distinction to this work's single-stage evaluation of the fluxes at time level n.

The far-field boundary conditions consist of inflow and outflow conditions. The inflow conditions are $\bar{U} = \bar{U}_\infty$, written in a time-evolutionary form, and the outflow conditions are time-evolutionary forms of the conditions $\rho_x = 0$, $(\rho u)_x = 0$, and $(\rho v)_x = 0$, as well as a freestream specification of total energy. The airfoil surface boundary conditions consist of impermeability, the vanishing normal pressure gradient $\partial p/\partial n = 0$, and the adiabatic wall condition $\partial T/\partial n = 0$.

Equation (4) is expressed in a field-collocated form as $U_t = F(U, \alpha, \dot{\alpha}, \dot{h}; \bar{u})$, where U is an ordered collocation of $\bar{U}$ at all interior and far-field grid points[1] and the reduced velocity is treated as a parameter. The three structural parameters α, $\dot{\alpha}$, and $\dot{h}$ affect the aerodynamics. The unsteady term U_t is a continuum representation of the left-hand side of Eq. (4), whereas the nonlinear function F represents a simple collocation of the right-hand sides of Eq. (4) over interior and far-field grid points. As implied earlier, flow variables at the airfoil surface and along the grid cut are not carried as dependent variables but are treated as temporary variables through repeated satisfaction of boundary and cut conditions. The conditions enforced along the grid cut are described later in this section.

The aeroelastic equations are coupled into a single time-dependent system governing the interaction between structure and inviscid flowfield,

$$Y_t = G(Y; \bar{u}) \tag{5}$$

by defining $Y \equiv [U, S]^T$ and $G \equiv [F, \mathcal{K}]^T$. Equilibrium solutions of Eq. (5) satisfy the nonlinear system $G(Y; \bar{u}) = 0$ and are calculated with either Newton's method or the chord method [using lower–upper (LU) decomposition], with account taken of the internal structure of G_Y. Newton iterates are of the form $G_Y^{\nu} \Delta Y = -G^{\nu}$, where the correction ΔY is used to update the solution: $Y^{\nu+1} = Y^{\nu} + \Delta Y$. With the chord method, the Jacobian is frozen at iteration N_{chord}, such that iterates $\nu \geq N_{\text{chord}}$ take the form $G_Y^{N_{\text{chord}}-1} \Delta Y = -G^{\nu}$. Elements of G_Y are computed through second-order accurate numerical approximations. Convergence is evaluated with the stopping criterion $\|G\|_2 \leq \delta_{\text{EQ}}$, where δ_{EQ} is prescribed to be 10^{-5} and $\|G\|_2 \equiv \sqrt{(G^T G)}$. The aeroelastic equations are also solved through time integration of Eq. (5) to occasionally provide initial conditions for the flutter-point analysis and to verify the stability properties across computed Hopf points.[1,2,16]

At a Hopf point, a steady-state solution transitions to an oscillatory solution with zero amplitude,[17] a change in behavior used in this work to predict the onset of limit-cycle oscillation (LCO). For the pitch-and-plunge airfoils previously examined by Morton and Beran,[1,2] prediction of LCO is sufficient to describe the stability properties of the aeroelastic system. The Hopf-point solution is an oscillatory perturbation to an equilibrium solution Y^0 such that $Y = Y^0 + \epsilon P e^{\beta t}$ as $\epsilon \to 0$. The Jacobian at a Hopf point satisfies the eigenproblem $G_Y P = \beta P$, where $\beta = \pm i\Theta$ and $P \equiv P_1 + iP_2$. Two conditions constrain P to be nontrivial[17]: $q^T P_1 = 0$ and $q^T P_2 = 1$, where q is a constant vector. Thus, at the equilibrium points that are also Hopf points, an expanded system of equations $\mathcal{F}(\mathcal{X}) = 0$ is satisfied[3]:

$$\mathcal{F} = \begin{bmatrix} G_Y P_1 + \Theta P_2 \\ q^T P_1 \\ G \\ G_Y P_2 - \Theta P_1 \\ q^T P_2 - 1 \end{bmatrix} = 0, \qquad \mathcal{X} = \begin{bmatrix} P_1 \\ \bar{u} \\ Y \\ P_2 \\ \Theta \end{bmatrix} \tag{6}$$

The expanded system (6) has been solved previously for predicting stability changes in flowfields about nonmoving structures. Jackson[18] computed low Reynolds number transition to time-periodic flowfields for a number of simple geometries in two-dimensional, viscous flowfields. With a Galerkin finite element methodology, he obtained good agreement with experimental data for the shedding frequency and critical Reynolds number associated with flow about a circular cylinder. Using a finite difference formulation of the Navier–Stokes equations in two dimensions, as applied to the discretization of Eq. (6), Beran and Lutton[19] obtained critical Reynolds numbers and shedding frequencies for viscous flow about a NACA 0012 airfoil at various angles of attack.

Morton and Beran[2] solved the expanded system (6) for an airfoil with structural coupling in transonic flow to determine the flutter speed of the aeroelastic system. They used an underrelaxed, approximate Newton method, $\hat{\mathcal{F}}_{\mathcal{X}}^{\nu} \Delta\mathcal{X} = -\mathcal{F}^{\nu}$, where $\hat{\mathcal{F}}_{\mathcal{X}}$ is a defective Jacobian:

$$\hat{\mathcal{F}}_{\mathcal{X}} \equiv \begin{bmatrix} G_Y & (G_Y P_1)_{\bar{u}} & (G_Y P_1)_Y & \Theta\hat{I} & P_2 \\ q^T & 0 & 0 & 0 & 0 \\ 0 & G_{\bar{u}} & G_Y & 0 & 0 \\ -\Theta I & (G_Y P_2)_{\bar{u}} & (G_Y P_2)_Y & G_Y & -P_1 \\ 0 & 0 & 0 & q^T & 0 \end{bmatrix} \tag{7}$$

where I is the identity matrix and $\hat{I}$ is an incomplete identity matrix entirely composed of zero-value elements, except for diagonal elements of unity value corresponding to the four structural dynamics equations. The term $\hat{\mathcal{F}}_{\mathcal{X}}$ becomes equivalent to $\mathcal{F}_{\mathcal{X}}$ when $\hat{I}$ is replaced with I.

The use of a defective Jacobian greatly facilitates the computation of $\Delta\mathcal{X}$. However, to maintain numerical stability, underrelaxation of the correction is, in general, necessary: $\mathcal{X}^{\nu+1} = \mathcal{X}^{\nu} + \omega_{\text{SUR}} \Delta\mathcal{X}$. Morton and Beran[2] found ω_{SUR} to be constrained by flow character; values as large as 0.5 were acceptable for fully subsonic flow, but values below 0.1 were required in the transonic regime. Improved convergence properties were obtained in this study by only applying underrelaxation to the correction for P_2: $P_2^{\nu+1} = P_2^{\nu} + \omega_{\text{SUR}} \Delta P_2$. Iterates are continued until $\|\mathcal{F}_i^{\nu}\|_2 < \delta_{\text{HOPF}}$ $(i = 1, 2, 3, 4, 5)$, where δ_{HOPF} is typically set to 10^{-3} (sufficient to provide critical reduced velocity to three or four digits of precision). Regular points are computed in the neighborhood of a Hopf point and used as initial approximations to the solution vector. The initialization of eigenvector components is described in Ref. 13.

Solutions presented in this paper are limited to the computation of flutter points for a symmetric NACA 64A006 airfoil with $\alpha_0 = \alpha_{\text{EQ}} = 0$. (Flutter-point solutions for $\alpha_{\text{EQ}} \neq 0$ computed with TVDntiAE have been presented elsewhere[2,20].) As a consequence of this specialization in α_{EQ}, a more efficient version of the algorithm described earlier is implemented.[3] When $\alpha_0 = 0$, the static aeroelastic state is independent of $\bar{u}$ ($\alpha = h = 0$). Thus, $G_{\bar{u}} = 0$, thereby decoupling $G(Y) = 0$ from the remaining equations in the system (6). The equilibrium solution of Y is computed using either Newton's method or the chord method applied to $G(Y) = 0$. The remaining equations of the expanded system are solved with a reduced system Jacobian.[13]

Computing solutions of linear systems involving the matrix G_Y is inherent to the block calculation of $\Delta\mathcal{X}$ (Ref. 3). With LU decomposition of G_Y, these linear systems are solved to machine precision in a fully implicit manner, i.e., all unknowns are updated simultaneously. However, the use of LU decomposition is not a viable method for the analysis of three-dimensional flowfields, given the immense size of G_Y. Thus, in three dimensions, alternative methods of solving linear systems involving G_Y must be explored. As recently described by Beran and Carlson,[21] block relaxation of discrete equations for Hopf points can be performed to extend the methodology to three dimensions as long as the spatial decomposition is robust to physical changes in stability at Hopf points.

BIFAE includes the ability to freeze the global time step in both the equilibrium and Hopf-point computational procedures. In TVDntiAE, the time step is set at the start of each iteration to meet globally a Courant–Friedrichs–Lewy (CFL) stability criterion. When numerical approximations to Jacobian elements are computed, this time step is reevaluated as field variables are dynamically altered. In BIFAE, the expense of recomputing the global time step is avoided by freezing the time step at the start of each iterate in the computation of equilibrium points and over the entire flutter computation.

PAC, a well-known continuation technique,[11] is applied to the expanded system of equations to yield a reexpanded system for the flutter boundary. With PAC, a scalar equation is added to the expanded system (6) to allow an existing solution point on the flutter boundary ($\mathcal{X}_0$) to aid in the calculation of a neighboring solution point on the same boundary ($\mathcal{X}_1$). In this paper, we use PAC to obtain flutter boundaries in two structural parameters, damping and mass ratio.

PAC is based on the existence of a vector tangent $\boldsymbol{T}$ to the solution path at $\mathcal{X}_0$. This vector lies in the space spanned by Y and the structural-model parameter to be varied. The solution point $\mathcal{X}_1$ is constrained to lie in a hyperplane normal to the tangent vector a specified distance Δ_s away from $\mathcal{X}_0$ in the direction given by the tangent vector. The single degree of freedom made available by the addition of this constraint is the value of the structural-model parameter, designated here as γ. The parameter γ is treated as a variable distinct from $\bar{u}$, such that both γ and $\bar{u}$ are computed as part of the PAC process. By parameterizing $\mathcal{X}$ and γ by arclength along the solution path s, we express $\boldsymbol{T}$ as $[\dot{\mathcal{X}}_0(s), \dot{\gamma}_0(s)]^T$. The scalar constraint for $\mathcal{X}_1$ is then written as

$$\mathcal{N} \equiv \dot{\mathcal{X}}_0^T (\mathcal{X}_1 - \mathcal{X}_0) + \dot{\gamma}_0(\gamma_1 - \gamma_0) - \Delta_s = 0 \tag{8}$$

The expanded system (6) is reexpanded to include the scalar equation (8), leading to Newton linearizations of the form

$$\begin{bmatrix} \mathcal{F}_{\mathcal{X}} & \mathcal{F}_{\gamma} \\ \dot{\mathcal{X}}_0^T & \dot{\gamma}_0 \end{bmatrix}^{\nu} \begin{bmatrix} \Delta\mathcal{X}_1 \\ \Delta\gamma_1 \end{bmatrix} = -\begin{bmatrix} \mathcal{F} \\ \mathcal{N} \end{bmatrix}^{\nu} \tag{9}$$

Table 1 Values of grid parameters

Grid	Δ_{le}	Δ_{te}	Δ_w	R_{max}	I	J
G-A	0.0049	0.0010	0.015	10	104	15
G-B	0.0049	0.0010	0.015	10	104	31
G-1	0.0010	0.0005	0.0006	6	84	31
G-2	0.0010	0.0005	0.0006	12	84	34
G-3	0.0010	0.0005	0.0006	25	84	39
G-4	0.0010	0.0005	0.0006	6	124	31

Linear system (9) is solved for the corrections $\Delta\mathcal{X}_1$ and $\Delta\gamma_1$ with a new block-matrix methodology based on the existing procedure for solving $\hat{\mathcal{F}}^v_{\mathcal{X}}\Delta\mathcal{X} = -\mathcal{F}^v$. The term $\mathcal{F}_\gamma$ is numerically computed using second-order accurate approximations. Through continued iteration, a calculated value of γ_1 is obtained automatically. The components of $\boldsymbol{T}$ are approximated with backward-difference formulas using known points $\mathcal{X}_0$ and $\mathcal{X}_{-1}$, where $\mathcal{X}_0$ lies between $\mathcal{X}_{-1}$ and $\mathcal{X}_1$ on the solution path. The term $\boldsymbol{T}$ is approximated through $\dot{\mathcal{X}}_0 \approx (\mathcal{X}_0 - \mathcal{X}_{-1})/\Delta_s$ and $\dot{\gamma}_0 \approx (\gamma_0 - \gamma_{-1})/\Delta_s$, where Δ_s is initialized once with

$$\Delta_s = \sqrt{(\mathcal{X}_0 - \mathcal{X}_{-1})^T(\mathcal{X}_0 - \mathcal{X}_{-1}) + (\gamma_0 - \gamma_{-1})^2} \qquad (10)$$

The desired solution point $(\mathcal{X}_1, \gamma_1)$ is predicted with $\mathcal{X}_1 \approx \mathcal{X}_0 + \dot{\mathcal{X}}_0\Delta_s$ and $\gamma_1 \approx \gamma_0 + \dot{\gamma}_0\Delta_s$.

Grids with high degrees of smoothness and surface orthogonality were constructed for the NACA 64A006 airfoil using a hyperbolic O-grid generation algorithm developed by Kinsey and Barth.[22] The geometry of the rounded trailing edge was obtained by matching the slope of the analytical airfoil geometry with a circular arc at approximately 98% chord. Details are given in Ref. 3. Two grids developed by Morton,[3] designated G-A and G-B, were also used in this study for validation purposes. These grids and the new grids developed for this study used equivalent surface descriptions.

The primary parameters specified during the grid-generation process were leading-edge and trailing-edge spacing (Δ_{le} and Δ_{te}; spacing between nodes on the airfoil surface at the two edges), wall spacing (Δ_w; spacing between a surface node and the first node off the airfoil surface), domain size (R_{max}), and the grid dimensions I (around the airfoil in the ξ-coordinate direction) and J (normal to the airfoil in the η-coordinate direction). Parameter values for grids used in this study are provided in Table 1. Four new grids were generated by varying I and R_{max} to evaluate solution sensitivity to these parameters. Solution sensitivity to Δ_{le}, Δ_{te}, Δ_w, and J was not systematically investigated. However, for the values of these parameters assumed herein, Morton[3] reported equilibrium-point solutions for a NACA 0012 airfoil at angles of attack that were reasonably grid converged.

In contrast, Morton[3] verified a noteworthy sensitivity of C_l to R_{max} for equilibrium-point computations. Thus, domain size was varied in this study to determine the value of R_{max} required to obtain accurate flutter onset predictions in the flutter-dip regime. Grid dimension I was also varied to evaluate the role of shock capturing in predicting flutter speed in the flutter-dip regime. Grid G-4 was characterized by a value of I 50% greater than that of grids G-1–G-3. The extra nodes on the airfoil surface were clustered at midchord, near the position of strong shock formation above about Mach 0.85. In this region, the streamwise node spacing of grid G-4 was about a tenth that of the other grids.

Results

Results are presented for the computation of equilibrium points and flutter points. First, flutter boundaries in Mach number and mass ratio are compared with those reported in the literature to validate the new computational approach. The sensitivity of these solutions to variation of certain numerical parameters is examined. Performance of the chord method in obtaining equilibrium solutions is also assessed. Next, a variety of flutter boundaries are presented to establish the insensitivity of solutions to grid refinement. Finally, we describe a path of flutter points computed through variation of structural damping.

Results are obtained using a standardized set of values for aerodynamic, structural dynamic, and numerical parameters and using a baseline configuration of the BIFAE algorithm. These specifications will be assumed in the following discussions unless stated otherwise. First, it is assumed that $\alpha_{EQ} = 0$. The other structural parameter values are those used elsewhere[2,23,24]: no structural damping, a mass ratio of 125, a center of gravity at 37.5% chord with $x_\alpha = -0.25$, a frequency ratio $\omega_h/\omega_\alpha = 0.20$, and $r_\alpha^2 = 0.25$. The numerical scheme is specified to be in time-accurate form using a frozen time step, with a vanishing TVD dissipation parameter δ_1 (Ref. 15) and a global CFL number of 0.5.

Flutter boundaries for variations in freestream Mach number are typically obtained in the following manner. For a specified grid, a starting solution is first obtained with BIFAE using time integration. The flowfield is initialized to a freestream state at Mach 0.3, and time integration is performed until residuals achieve an $L2$ norm on the order of 10^{-5}. Continuation begins, following initialization of the eigenvector P and constant vector q (Ref. 13). Flutter boundaries in M_∞ are not computed with the current version of the PAC software. Instead, solutions are obtained at regular Mach number intervals, using extrapolation to obtain starting approximations.

Code Validation

Solutions are now presented for grids G-A and G-B and compared with solutions obtained using identical grids reported elsewhere.[2,24] First, two flutter boundaries are constructed for variation in M_∞ between 0.75 and 0.854 and two different values of the damping ratios $\zeta_\alpha = \zeta_h$: 0.0 and 0.5. These boundaries are computed with grid G-A and compared with those reported by Morton and Beran.[2] Second, a flutter boundary is constructed for variation in mass ratio μ_s for Mach 0.85 using grid G-B. Results are compared with those reported by Buxton and Beran[24] and Wu et al.[23]

The flutter boundaries in M_∞ for grid G-A are shown in Fig. 2 along with the previous data computed with TVDntiAE.[2] Throughout most of the Mach number range, the flutter boundary calculations are in close agreement (within approximately 1%) for both sets of damping ratios. However, above Mach 0.84, differences are typically within 5%; a difference of 10% is observed at a single point (Mach 0.85, $\zeta_\alpha = \zeta_h = 0.5$). Very little of the difference is attributable to freezing the time step during the computation of the flutter point. When a second set of computations is performed with BIFAE using a dynamic time step, critical reduced velocities change by less than 0.1% in comparison with the original BIFAE data. We attribute the disparities between BIFAE and TVDntiAE to the use of a single-stage flux evaluation in BIFAE because that is the only remaining difference in the discretization of the governing equations between the two methods (the effect of flux evaluation on accuracy is addressed later). The BIFAE computations required approximately 2 h on a CPU for the case of vanishing structural damping vs 24 h for TVDntiAE.[2] Correcting for differences in workstation performance, BIFAE provides roughly a fivefold improvement in algorithmic efficiency over TVDntiAE. Note that BIFAE achieves a twofold improvement in the convergence rate when the time step is frozen.

To gain an independent validation of the current method, a flutter boundary computed with BIFAE for variation of mass ratio

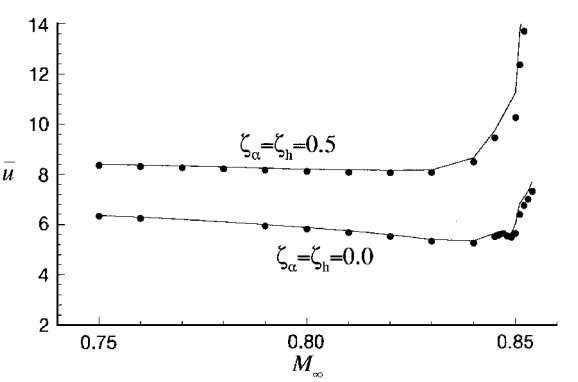

Fig. 2 Comparison of flutter boundaries for two different sets of damping ratios with grid G-A: ——, BIFAE (frozen time step) and ●, TVDntiAE.

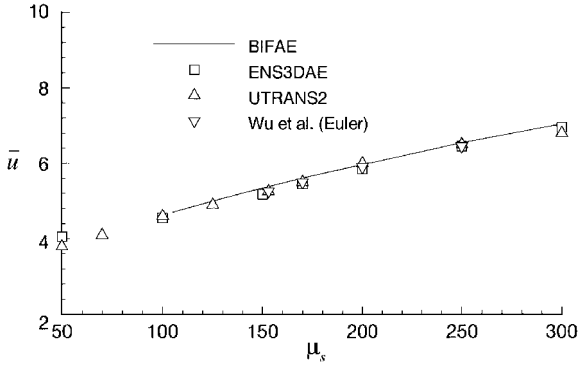

Fig. 3 Comparison of flutter boundaries for variation of mass ratio using four different algorithms (Mach 0.85, grid G-B).

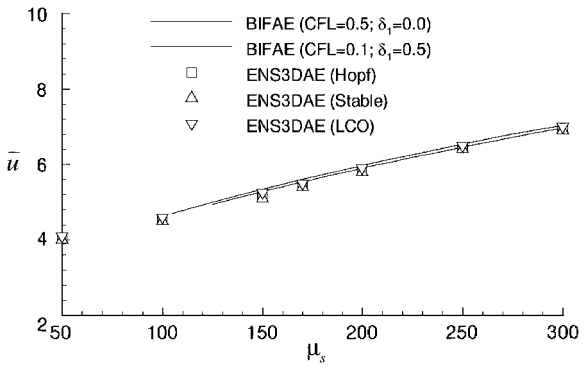

Fig. 4 Comparison of flutter boundaries for variation of mass ratio using different numerical parameter values and with END3DAE brackets (Mach 0.85, grid G-B).

is compared with that predicted by methods unrelated to BIFAE. Methodologies serving as a basis of comparison are the Wu et al.[23] Euler code, the UTRANS2 transonic small-disturbance code[25] as reported by Wu et al.,[23] and the ENS3DAE Euler/Navier–Stokes code[26] as reported by Buxton and Beran[24] for inviscid flow. Flutter results are obtained for Mach 0.85 and a variety of mass ratios between 100 and 300. Computations with ENS3DAE and BIFAE employ an identical grid, G-B. ENS3DAE estimates of flutter speed are reported as center values for a bracket about the point of stability exchange. The methods are compared in Fig. 3, and brackets are shown in Fig. 4; close agreement is achieved by all methods. In general, BIFAE predicts slightly higher values of flutter speed than ENS3DAE (about 3% higher than center values and about 2% higher than bracket tops). Note that the flutter boundary computed with BIFAE is composed of 40 solution points computed in about 5 CPU min each. In contrast, bracketing of solution points with ENS3DAE requires several CPU hours per point.

We now assess the effects of variation of three properties of the BIFAE algorithm on computed flutter speed: numerical dissipation, time step, and flux evaluation. Results from ENS3DAE serve as a baseline for comparison; all results are obtained for Mach 0.85. The numerical dissipation is adjusted by varying δ_1, a parameter appearing in the entropy correction of the TVD scheme. In general, the scheme is more dissipative when δ_1 is increased and least dissipative when the parameter vanishes.[15] Three values of δ_1 are examined: 0, 0.5, and 1.0. Results are computed for $\mu_s = 150$ and grid G-B; flutter speeds are reported in Table 2. As δ_1 is increased from 0, $\bar{u}$ is observed to drop slightly, closer to the value bracketed by ENS3DAE. When $\delta_1 = 0$, the default condition, and the CFL number is reduced from 0.5 to 0.1, $\bar{u}$ again drops slightly, closer to the value bracketed by ENS3DAE. However, the CFL number used in the ENS3DAE calculation is 9.0 (Ref. 24), thus suggesting that improved comparison with ENS3DAE through CFL reduction is coincidental. Also, the flutter speed predicted with grid G-B for $\mu_s = 150$ is nearly unchanged when the calculation is repeated using grid G-1 (with $\delta_1 = 0.0$ and a CFL number of 0.5). As provided in Table 2, the critical value of $\bar{u}$ for grid G-1 is found to be 5.358. Flutter boundaries computed with 1) the default set of numerical parameters and 2) the CFL number set to 0.1 with $\delta_1 = 0.5$ are compared with the END3DAE results in Fig. 4. The ENS3DAE bracket is very tight, typically covering a difference in $\bar{u}$ of 0.1. The second flutter boundary computed with BIFAE provides a match superior to that obtained with ENS3DAE over all values of mass ratio examined.

Table 2 Flutter speeds for different numerical conditions at Mach 0.85 with $\mu_s = 150$ (flutter speed estimated with ENS3DAE is 5.175)

Grid	Flux	δ_1	CFL no.	$\bar{u}$
G-B	Single	0.0	0.5	5.342
G-B	Single	0.5	0.5	5.313
G-B	Single	1.0	0.5	5.306
G-B	Single	0.0	0.1	5.309
G-B	Single	0.5	0.1	5.276
G-B	Single	1.0	0.1	5.274
G-1	Single	0.0	0.5	5.358
G-A	Single	0.0	0.5	6.437
G-A	Single	0.0	0.1	6.518
G-A	Single	0.0	0.01	6.539
G-A	Single	0.0	0.001	6.542
G-A	Dual	0.0	0.5	6.030
G-A	Dual	0.0	0.1	6.420
G-A	Dual	0.0	0.01	6.517
G-A	Dual	0.0	0.001	6.528

For a given flowfield state, the CFL number is proportional to Δ_t. Thus, decreases in CFL number increase the temporal accuracy of time integration of the discrete equations. As a consequence, the flutter speed predicted by BIFAE should be convergent to a more accurate estimate as CFL is reduced because the time integration and flutter analysis modes of BIFAE are fully consistent.[2] As seen in Table 2 for grid G-A, the sequence of flutter speeds is convergent to about 6.54 as the CFL number is reduced to 0.001. We emphasize that the computational work performed by the flutter analysis mode of BIFAE is nearly independent of the CFL number, whereas the computational work involved in time integration is inversely proportional to the CFL number.

As proposed earlier, differences in the flux-evaluation scheme affect computed values of flutter speed. We evaluate differences between the single-stage (BIFAE) and dual-stage (TVDntiAE) flux-evaluation schemes by varying the CFL number to determine the asymptotic values of $\bar{u}$ for a small time step. Convergence should be to equivalent asymptotic values (limited only by the discrete manner in which the Jacobian matrices are computed and stored) because both flux-evaluation schemes are consistent. Results are compared in Table 2 for grid G-A. When the CFL number is reduced to 0.001, BIFAE predicts $\bar{u} \to 6.542$, whereas TVDntiAE predicts $\bar{u} \to 6.528$. These nearly convergent values of $\bar{u}$ are in excellent agreement. Furthermore, BIFAE appears to converge faster to an asymptotic value of $\bar{u}$, indicating that, when the CFL number is 0.5, BIFAE provides a more accurate prediction of flutter speed than TVDntiAE.

The chord method is used to compute equilibrium-point solutions, which define the aerodynamic state during the computation of flutter points. As described earlier, chord iterates are applied in place of Newton iterates at a specified starting iterate N_{chord}. The specification of N_{chord} impacts the efficiency with which equilibrium points are computed. The dependence of convergence rate on N_{chord} is evaluated by computing two equilibrium points with a series of different starting iterates. These calculations are carried out with grid G-1 for two Mach numbers, 0.31 and 0.84. Initial conditions for the two cases are solutions at Mach 0.30 and 0.83, respectively.

Convergence histories for the chord method and Newton's method, as applied to the computation of the equilibrium point at Mach 0.31, are shown in Fig. 5. Convergence is evaluated using the $L2$ norm of the residual G. For Mach 0.31, the convergence of Newton's method is found to be superlinear between six and nine iterations. However, beyond the ninth iteration, the convergence rate degrades to linear, ultimately achieving a convergence rate of 0.19 at a residual norm of approximately 10^{-9} on the 12th iteration (not shown). The convergence properties for three values of N_{chord}, 5, 6, and 7, are also shown in Fig. 5 (divergence is established for $N_{\text{chord}} = 4$). Convergence to a residual norm of 10^{-5} is achieved in

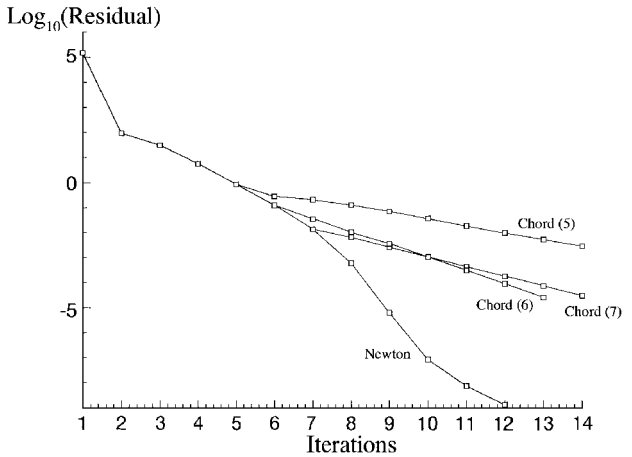

Fig. 5 Convergence histories for regular point computation at Mach 0.31 using Newton's method and the chord method (values of N_{chord} given in parentheses).

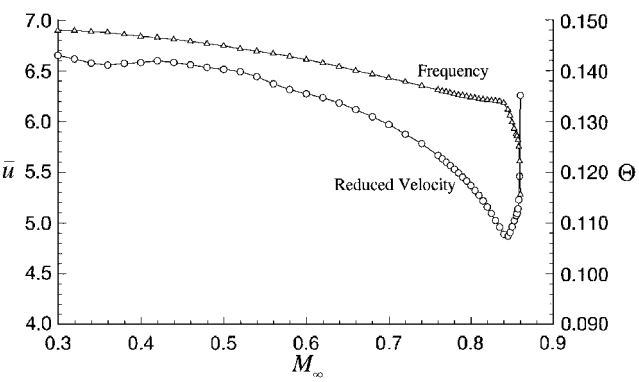

Fig. 6 Flutter boundary for variation of M_∞ (grid G-1).

all three cases in 23, 13, and 15 iterations, respectively (convergence to the same norm is obtained with Newton's method in 8 iterations). Best performance is observed for $N_{\text{chord}} = 6$. For this value of N_{chord}, residual norms of 10^{-5} and 10^{-9} are obtained with 75 and 50% the computational effort of Newton's method, respectively.

At Mach 0.84, the chord method again is more efficient than Newton's method in achieving two specified residual norms. For a residual norm of 10^{-5}, convergence is achieved with Newton's method in 9 iterations and in 12 and 9 iterations, respectively, for $N_{\text{chord}} = 7$ and 8. For $N_{\text{chord}} = 8$, residual norms of 10^{-5} and 10^{-7} are obtained with 80 and 20% of the computational effort of Newton's method. In the latter case, the relatively poor performance of Newton's method reflects a very slow asymptotic convergence rate (0.93) at a residual norm of 10^{-7}.

Solution Sensitivity to Grid Refinement

Solutions are now presented for grid G-1, a grid with a relatively small computational domain size. A flutter boundary, composed of over 50 solution points, is obtained for M_∞ ranging between 0.30 and 0.86 and is shown in Fig. 6 in terms of critical reduced velocity and frequency Θ. The profile of reduced velocity is characterized by a shallow decline in $\bar{u}$ with increasing Mach number for values of M_∞ less than about 0.6. With further increase in M_∞, the critical reduced velocity declines at a faster rate, ultimately achieving a minimum value (over the range of M_∞ examined) of 4.87 at $M_\infty = 0.845$. The term $\bar{u}$ is observed to increase with M_∞ between Mach 0.8475 and 0.86. (Solutions beyond Mach 0.86 were not computed because the value of ω_{SUR} required for convergence became too small for efficient computation.)

The effects of structural damping on flutter boundary position are assessed by computing flutter boundaries for variation in M_∞ using grid G-1 with different values of the structural damping ratios: ζ_α and ζ_h (assumed equal). Four sets of ratio values are compared in Fig. 7 (0.0, 0.25, 0.50, and 1.0), along with the flutter boundaries described earlier for grid G-A. As seen in Fig. 7, reduction of structural damping reduces $\bar{u}$. Also, grid refinement (grid G-A to grid G-1) reduces flutter speed, typically by at least 10%, suggesting that the grid independence of flutter solutions needs to be established.

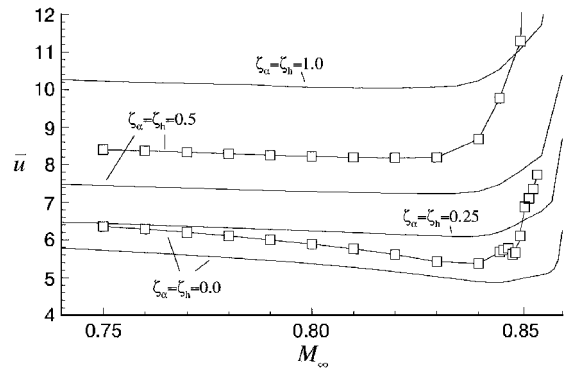

Fig. 7 Variation of structural damping for two grids: □, G-A and ——, G-1.

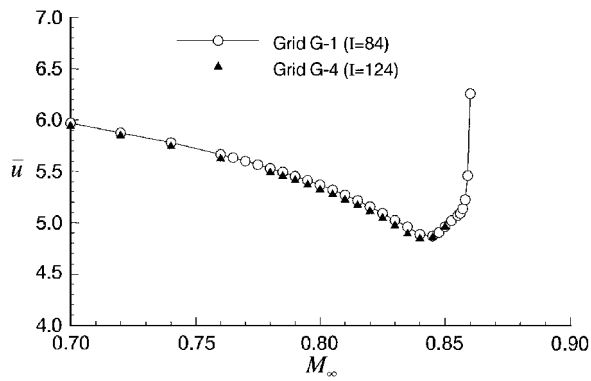

Fig. 8 Comparison of flutter boundaries for variation of node density in ξ-coordinate direction.

The evaluation of solution sensitivity to grid refinement is limited to the variation of two grid parameters: the number of nodes composing the airfoil surface I and the domain size $R_{\max}$. When the latter parameter is varied, the number of nodes normal to the airfoil J is adjusted to maintain a constant grid structure near the airfoil. Results are presented first for variation of I in Fig. 8 using two grids: G-1 ($I = 84$) and G-4 ($I = 124$). The flutter boundaries are computed with no structural damping and are shown between Mach 0.70 and Mach 0.86. Extension of the boundaries to higher Mach numbers is limited by the convergence properties of BIFAE in this regime. Results obtained with the two grids are in excellent agreement through the flutter-dip region, with grid G-1 yielding critical values of $\bar{u}$ approximately 1% larger than those predicted using grid G-4. These results suggest that I need not exceed 84 to capture accurately the transonic dip phenomenon.

Solution sensitivity to variations in domain size is found to be dependent on structural damping. This sensitivity is evaluated using two grids: G-1 ($R_{\max} = 6$) and G-2 ($R_{\max} = 12$). Flutter boundaries are obtained for three sets of damping ratios ($\zeta_\alpha = \zeta_h$ assumed): 0.0, 0.25, and 0.5. Results are shown in Fig. 9. For the largest value of structural damping (0.5), the two grids yield nearly identical results over the entire Mach number range. When $\zeta_\alpha = \zeta_h = 0.25$, grid G-1 yields critical values of $\bar{u}$ within 0.5% of those predicted with grid G-2 over the available Mach range. This maximum difference grows to 5% when damping is reduced to zero. Comparisons at Mach numbers below 0.7 (not shown) are in relatively good agreement for all damping ratios. Solutions using grid G-2 were not obtained for damping ratios below 0.5 and above Mach 0.815 because of the decreasing efficiency of BIFAE ($\omega_{\text{SUR}} < 0.05$) under these conditions.

At a single Mach number, 0.30, flutter speeds for grids G-1–G-4 are compared. The critical values of $\bar{u}$ are found to be 6.685 (G-1), 6.740 (G-2), 6.742 (G-3), and 6.580 (G-4). In this low Mach number regime, critical $\bar{u}$ is seen to have little sensitivity to $R_{\max}$

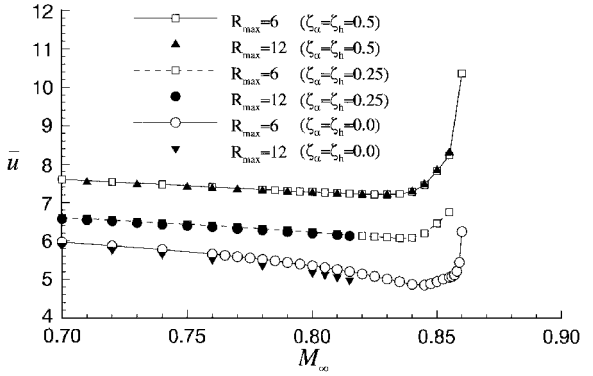

Fig. 9 Comparison of flutter boundaries for variation of domain size.

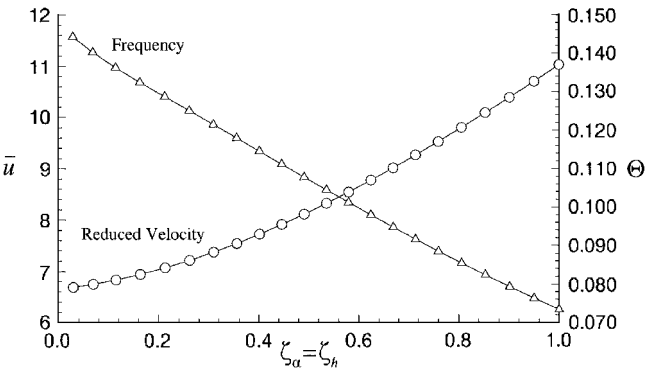

Fig. 10 Computation of flutter boundary with PAC for structural damping ratios between 0 and 1.

(1% increase after 300% increase in $R_{\max}$) and I (2% decrease after a 50% increase in I).

Continuation in Structural Damping

A flutter boundary is now presented for variation of structural damping. In this PAC calculation, the following conditions are assumed: $\zeta_h = \zeta_\alpha$, $M_\infty = 0.30$, $\omega_{\text{SUR}} = 0.25$, and $\Delta_s = 0.05$ (initially). The boundary is computed with grid G-1, starting at $\zeta_h = \zeta_\alpha = 1.0$, continuing in the direction of decreased damping, and extending over the range of positive damping values. For these conditions, a flutter boundary composed of 22 points is obtained. Results are shown in Fig. 10 in terms of the computed flutter speed and frequency. Frequency is observed to increase and reduced velocity is observed to decrease as damping is reduced. With PAC, most points are computed in six approximate Newton iterates of the expanded system. In comparison, if PAC is not used, and neighboring solution points are used to provide initial estimates ($\mathcal{X}_1 = \mathcal{X}_0$), then 16–17 iterates per flutter solution are required to achieve convergence.

Conclusions

A continuation procedure was developed to trace the flutter boundaries associated with a pitch-and-plunge airfoil in the subsonic and transonic regimes. The pseudoarclength continuation procedure was applied to the computation of flutter boundaries by variation of airfoil-fluid mass ratio and structural damping. Results were validated against those reported in the literature and computed with other computational procedures. The analysis was limited to the Euler equations, thus rendering the method incapable of predicting the effects of viscosity on flutter speed. New features for flutter-point computation, including single-stage flux evaluation, time-step freezing, and the chord method, were successfully implemented and tested, resulting in improved algorithmic performance. Solution insensitivity to grid refinement was established in the transonic flutter-dip region, with maximum differences less than 5% (typically about 1%). Differences attributable to time discretization were observed to be less than 2%; differences attributable to numerical dissipation were found to be even less significant. Thus, given the efficiency with which flutter boundaries were computed (a few CPU hours for the variation of structural parameters), the procedure achieved high accuracy.

Acknowledgment

This work was sponsored by the the Air Force Office of Scientific Research (AFOSR/NA) through Project/Task 2300/AS.

References

[1]Morton, S. A., and Beran, P. S., "Nonlinear Analysis of Airfoil Flutter at Transonic Speeds," *Proceedings of the AIAA 13th Applied Aerodynamics Conference*, AIAA, Washington, DC, 1995 (AIAA Paper 95-1905).

[2]Morton, S. A., and Beran, P. S., "Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds," AIAA Paper 96-0060, Jan. 1996.

[3]Morton, S. A., "Nonlinear Analysis of Airfoil Flutter at Transonic Speeds," Ph.D. Thesis, U.S. Air Force Inst. of Technology, Wright–Patterson AFB, OH, May 1996.

[4]Batina, J. T., "Unsteady Transonic Algorithm Improvements for Realistic Aircraft Applications," *Journal of Aircraft*, Vol. 26, No. 2, 1989, pp. 131–139.

[5]Edwards, J. W., "Transonic Shock Oscillations and Wing Flutter Calculated with an Interactive Boundary Layer Coupling Method," NASA TM-110284, Aug. 1996.

[6]Guruswamy, G. P., and Tu, E. L., "Navier–Stokes Computations on Flexible Advanced Transport Wings in Transonic Regime," *Journal of Aircraft*, Vol. 33, No. 3, 1996, pp. 576–581.

[7]Schuster, D., Vadyak, J., and Atta, E., "Static Aeroelastic Analysis of Fighter Aircraft Using a Three-Dimensional Navier–Stokes Algorithm," *Journal of Aircraft*, Vol. 27, No. 9, 1996, pp. 820–825.

[8]Gupta, K. K., "Development of a Finite Element Aeroelastic Analysis Capability," *Journal of Aircraft*, Vol. 33, No. 5, 1996, pp. 995–1002.

[9]Edwards, J. W., and Malone, J. B., "Current Status of Computational Methods for Transonic Unsteady Aerodynamics and Aeroelastic Applications," NASA TM-104191, Jan. 1992.

[10]Dowell, E. H., "Eigenmode Analysis in Unsteady Aerodynamics: Reduced-Order Models," *AIAA Journal*, Vol. 34, No. 8, 1996, pp. 1578–1583.

[11]Seydel, R., *From Equilibrium to Chaos: Practical Bifurcation and Stability Analysis*, Elsevier, New York, 1988, pp. 109–127.

[12]Keller, H. B., "Continuation Methods in Computational Fluid Dynamics," *Numerical and Physical Aspects of Aerodynamics Flows*, edited by T. Cebeci, Springer–Verlag, New York, 1982, pp. 3–14.

[13]Beran, P. S., and Morton, S. A., "A Continuation Method for the Calculation of Airfoil Flutter Boundaries," AIAA Paper 97-0574, Jan. 1997.

[14]Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, No. 3, 1983, pp. 357–393.

[15]Yee, H. C., "A Class of High-Resolution Explicit and Implicit Shock Capturing Methods," NASA TM-101088, Feb. 1989.

[16]Lutton, M. J., and Beran, P. S., "Hopf Bifurcation in Viscous Low Speed Flows About an Airfoil with Structural Coupling," *Computers and Fluids*, Vol. 23, No. 2, 1994, pp. 323–345.

[17]Griewank, A., and Reddien, G., "The Calculation of Hopf Points by a Direct Method," *IMA Journal of Numerical Analysis*, Vol. 3, No. 3, 1983, pp. 295–303.

[18]Jackson, C. P., "A Finite-Element Study of the Onset of Vortex Shedding in Flow Past Variously Shaped Bodies," *Journal of Fluid Mechanics*, Vol. 182, July 1987, pp. 23–45.

[19]Beran, P. S., and Lutton, M. J., "Hopf Bifurcation in Viscous Flows About Airfoils at Low Speeds," AIAA Paper 91-1807, June 1991.

[20]Morton, S. A., and Beran, P. S., "Effects of Structural Nonlinearity in the Bifurcation Analysis of Transonic Airfoil Flutter," AIAA Paper 96-1975, June 1996.

[21]Beran, P. S., and Carlson, C. D., "Domain-Decomposition Methods for Bifurcation Analysis," AIAA Paper 97-0518, Jan. 1997.

[22]Kinsey, D. W., and Barth, T. J., "Description of a Hyperbolic Grid Generating Procedure for Arbitrary Two-Dimensional Bodies," U.S. Air Force Wright Aeronautical Labs., AFWAL-TM-84-FIMM, Wright–Patterson AFB, OH, July 1984.

[23]Wu, J.-C., Kaza, K. R. V., and Sankar, L. N., "A Technique for the Prediction of Airfoil Flutter Characteristics in Separated Flow," AIAA Paper 87-0910, April 1987.

[24]Buxton, B. J., and Beran, P. S., "Validation of Two Shock-Capturing Methods for Calculaton of Transonic Airfoil Flutter," AIAA Paper 97-0834, Jan. 1997.

[25]Farr, J. L., Traci, R. M., and Albano, E. D., "Computer Programs for Calculating Small Disturbance Transonic Flows About Oscillating Airfoils," U.S. Air Force Flight Dynamics Lab., AFFDL-TR-74-135, Wright–Patterson AFB, OH, Nov. 1974.

[26]Smith, M. D., Schuster, D., Huttsell, L., and Buxton, B. J., "Development of an Euler/Navier–Stokes Aeroelastic Method for Three-Dimensional Vehicles with Multiple Flexible Structures," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 37th Structures, Structural Dynamics, and Materials Conference*, AIAA, Reston, VA, 1996, pp. 1733–1746 (AIAA Paper 96-1513).