JOURNAL OF AIRCRAFT
Vol. 38, No. 4, July–August 2001

CFD-Based Aeroelastic Eigensolver for the Subsonic, Transonic, and Supersonic Regimes

Michel Lesoinne* and Charbel Farhat†
University of Colorado, Boulder, Colorado 80309-0429

We describe a linearized computational fluid dynamics (CFD) method for computing an arbitrary number of eigensolutions of a given aeroelastic problem. The proposed method is based on the reengineering of a three-way coupled formulation previously developed for the solution in the time domain of nonlinear transient aeroelastic problems. It is applicable in the subsonic, transonic, and supersonic flow regimes, and independently from the frequency or damping level of the target aeroelastic modes. It is based on the computation of the complex eigensolution of a carefully linearized fluid–structure interaction problem, relies on the inverse orthogonal iteration algorithm, and reutilizes existing unsteady flow solvers. This is a validated method with the flutter analysis of the AGARD Wing 445.6 for which experimental data are available.

Introduction

RECENT work in computational dynamic aeroelasticity has essentially focused on the development of advanced direct time-integration methods for the solution of the coupled fluid–structure equations of motion.[1–5] Such methods have been applied to flutter analysis and to the numerical investigation of active control devices attached to aeroelastic systems. The area of active flutter suppression has also received special attention. However, few recent works have addressed the problem of determining the coupled aeroelastic mode shapes, frequencies, and natural (open-loop) damping of an airframe structure. Existing methods[6–8] that tackle this problem are often restricted to critical subsonic and supersonic flow speeds and therefore can neither predict the negative or positive damping of an aeroelastic system away from the flutter speed or in the transonic regime, nor compute nonflutter mode shapes that may be required for designing an efficient active controller. Some other methods that also rely on a simplified linear flow theory can produce a more complete set of solutions;[9] however, their range of applications usually excludes the transonic regime. An alternative approach is to produce first a reduced-order aerodynamic operator and then find the eigensolutions of the coupled equations in which such an operator is used. However, in his work on model reduction, Dowell[10] requires the extraction of the complete set of complex eigensolutions at a cost that rapidly becomes prohibitive as the fluid mesh is refined. The method of approximate balancing proposed by Baker et al.[11] does not suffer from this drawback, but it does not have a guaranteed convergence and is harder to implement. Also, the sensitivity of the coupled eigensolution to the model reduction has yet to be understood.

In this paper, we propose a numerical method for computing an arbitrary number of eigensolutions of a given aeroelastic problem. This method is based on the linearization of the Euler equations and the reengineering of a three-way coupled formulation previously developed for the solution of nonlinear transient aeroelastic problems in the time domain.[12] Because it is CFD based, it is valid in principle in all flow regimes where the CFD method is valid (in our case, subsonic, transonic, and supersonic). Because it reutilizes existing unsteady flow solvers and nonlinear aeroelastic simulation capabilities, it expands their range of applications and simplifies code maintenance. Our method does not assume a harmonic motion and therefore is applicable independently from the frequency or damping level of the target aeroelastic modes. It is based on the computation of the complex eigensolution of a carefully linearized fluid–structure interaction problem, and it relies on the inverse orthogonal iteration algorithm.[13] It is applied directly to the unreduced coupled equations and avoids any prior reduction of the order of the fluid operator.

We also provide an overview of the computer implementation of the proposed numerical method, illustrate it with the stability analysis of a flat panel with infinite aspect ratio in supersonic airstreams, and validate it with the flutter analysis of the AGARD 445.6 Wing in the range $0.49 \leq M_\infty \leq 1.14$.

Three-Field Formulation of a Nonlinear Aeroelastic Problem

Governing Equations

In many aeroelastic applications where some of the fluid domain boundaries undergo a motion with a large amplitude, it is necessary to solve the flow equations on a moving and possibly deforming grid. Such a grid is commonly referred to in the computational aerodynamics literature as a dynamic mesh. Several approaches have been proposed in the past for solving fluid–structure interaction problems on moving and deforming meshes, among which we note the two closely related arbitrary Lagrangian Eulerian (ALE)[14,12] and dynamic mesh[15] methods. In the most general case, all of these methods can be used to formulate the fluid–structure problem of interest as a three-field problem[16]: the fluid, the structure, and the dynamic mesh that is often represented by a pseudostructural system. For example, in the case of the ALE method, a fluid–structure interaction problem can be described by the following coupled partial differential equations:

$$\left.\frac{\partial(J\boldsymbol{w})}{\partial t}\right|_{\xi} + J\nabla_x \cdot [\mathcal{F}(\boldsymbol{w}) - \dot{x}\boldsymbol{w}] = J\nabla_x \cdot \mathcal{R}(\boldsymbol{w}) \tag{1}$$

$$\rho_S \frac{\partial^2 u_S}{\partial t^2} - \mathrm{div}\{\sigma_S[\epsilon_S(u_S)]\} = b \tag{2}$$

$$\tilde{\rho}\frac{\partial^2 x}{\partial t^2} - \mathrm{div}[\tilde{E} : \tilde{\epsilon}(x)] = g(u_S) \tag{3}$$

Eq. (1) is the ALE nondimensional conservative form of the Navier–Stokes equations and describes viscous flows on dynamic meshes. Here, t denotes time; a dot designates a derivative with respect to time; $x(t)$ denotes the time-dependent position or displacement of a fluid grid point (depending on the context of the sentence

Presented as Paper 97-0647 at the 35th Aerospace Sciences Meeting and Exhibit, Reno, NV, 6–9 January 1997; received 6 April 1998; accepted for publication 10 November 2000.

*Assistant Professor, Department of Aerospace Engineering and Sciences and Center for Aerospace Structures. Member AIAA.

†Professor, Department of Aerospace Engineering and Sciences and Center for Aerospace Structures. Fellow AIAA.

and the equation); ξ its position in a reference configuration, $J = \det(\mathrm{d}x/\mathrm{d}\xi)$; $\boldsymbol{w}$ is the fluid-state vector using the conservative variables; and $\mathcal{F}$ and $\mathcal{R}$ denote, respectively, the convective and diffusive fluxes. Eq. (2) is an elastodynamic equation where u_S denotes the displacement field of the structure and ρ_S its density; σ_S and ϵ_S denote, respectively, the stress and strain tensors; and b represents the body forces acting on the given structure. Finally, Eq. (3) governs the dynamics of the fluid-moving grid. It is similar to the elastodynamic equation because the dynamic mesh is viewed here as a pseudostructural system. A tilde notation is used to indicate that $\tilde{\rho}$ is a fictitious density and $\tilde{E}$ is a fictitious tensor of elasticities.[17] Usually, the fluid mesh motion is triggered by the displacement or vibration of the structure, which is represented here by $g(u_S)$. The various Dirichlet and Neumann boundary conditions intrinsic to each of the fluid and structure problems are omitted for simplicity.

Equations (1), (2), and (3) are directly coupled. If u_F denotes the ALE displacement field of the fluid and p its pressure field, σ_S and σ_F denote the structure stress tensor and the fluid viscous stress tensor, Γ denotes the fluid–structure interface boundary (wet boundary of the structure), and n denotes the normal at a point to Γ, the fluid and structure equations are usually coupled by imposing that

$$\sigma_S \cdot n = -pn + \sigma_F \cdot n \qquad \text{on} \quad \Gamma \tag{4}$$

$$u_S = u_F \qquad \text{on} \quad \Gamma \tag{5}$$

The first of these two interface boundary conditions states that the tractions on the wet surface of the structure are in equilibrium with those on the fluid side of Γ. Equation (5) expresses the compatibility between the displacement fields of the structure and the fluid at the fluid–structure interface. For inviscid flows, this second equation is replaced by the slip-wall boundary condition:

$$\frac{\partial u_F}{\partial t} \cdot n = \frac{\partial u_S}{\partial t} \cdot n \qquad \text{on} \quad \Gamma \tag{6}$$

The structure and dynamic mesh motions are also coupled by the continuity conditions

$$x = u_S \qquad \text{on} \quad \Gamma \tag{7}$$

$$\frac{\partial x}{\partial t} = \frac{\partial u_S}{\partial t} \qquad \text{on} \quad \Gamma \tag{8}$$

In the sequel, only inviscid flows are considered.

Semidiscretization

The spatial approximation of the ALE nondimensional conservative form of the Euler equations by finite element or finite volume schemes leads to semidiscrete equations that can be written as

$$\frac{\partial}{\partial t}[\boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}] + \bar{\boldsymbol{F}}(\boldsymbol{w}, \boldsymbol{x}, \dot{\boldsymbol{x}}) = 0 \tag{9}$$

where a bold font designates the discrete counterpart of a field variable, and in the case of a finite volume semidiscretization, $\boldsymbol{A}$ is the diagonal matrix of the cell volumes and $\bar{\boldsymbol{F}}$ is the numerical flux approximating the integral of the physical flux function over the cell interfaces.

The finite element equations of structural equilibrium can be expressed as

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}, \dot{\boldsymbol{u}}) = \boldsymbol{f}_{\text{ext}}(\boldsymbol{u}, \boldsymbol{w}) \tag{10}$$

where $\boldsymbol{M}$ is the finite element mass matrix, $\boldsymbol{u}$ denotes the displacement vector, $\boldsymbol{f}_{\text{int}}$ is the vector of internal forces, and $\boldsymbol{f}_{\text{ext}}$ is the vector of external forces induced by the fluid on the structure.

Finally, if the dynamic mesh is assimilated with a pseudostructural model, the semidiscrete equations governing its motion can be written as

$$\tilde{\boldsymbol{M}}\ddot{\boldsymbol{x}} + \tilde{\boldsymbol{D}}\dot{\boldsymbol{x}} + \tilde{\boldsymbol{K}}\boldsymbol{x} = \boldsymbol{K}_c\boldsymbol{u} \tag{11}$$

where $\tilde{\boldsymbol{M}}$, $\tilde{\boldsymbol{D}}$, and $\tilde{\boldsymbol{K}}$ are the fictitious mass, damping, and stiffness matrices,[17] and $\boldsymbol{K}_c$ is a transfer matrix that describes the action of the motion of the structural side of the fluid–structure interface on the fluid dynamic mesh. These matrices can be readjusted at each time step.

Linearization of the Nonlinear Aeroelastic Problem

An important characteristic of aeroelastic vibration problems is that they can be analyzed by linearizing the governing equations around an equilibrium point and investigating the response of the target aeroelastic system to a small perturbation around that point.

The linearization with respect to the variables $\boldsymbol{w}$, $\dot{\boldsymbol{w}}$, $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\ddot{\boldsymbol{x}}$, $\boldsymbol{u}$, $\dot{\boldsymbol{u}}$, and $\ddot{\boldsymbol{u}}$ is performed around an equilibrium point characterized by the values $\boldsymbol{w}_0$, $\dot{\boldsymbol{w}}_0$, $\boldsymbol{x}_0$, $\dot{\boldsymbol{x}}_0$, $\ddot{\boldsymbol{x}}_0$, $\boldsymbol{u}_0$, $\dot{\boldsymbol{u}}_0$, and $\ddot{\boldsymbol{u}}_0$, respectively. We have

$$\boldsymbol{w} = \boldsymbol{w}_0 + \delta\boldsymbol{w}, \qquad \dot{\boldsymbol{w}} = \dot{\boldsymbol{w}}_0 + \delta\dot{\boldsymbol{w}} \tag{12}$$

$$\boldsymbol{x} = \boldsymbol{x}_0 + \delta\boldsymbol{x}, \qquad \dot{\boldsymbol{x}} = \dot{\boldsymbol{x}}_0 + \delta\dot{\boldsymbol{x}}, \qquad \ddot{\boldsymbol{x}} = \ddot{\boldsymbol{x}}_0 + \delta\ddot{\boldsymbol{x}} \tag{13}$$

$$\boldsymbol{u} = \boldsymbol{u}_0 + \delta\boldsymbol{u}, \qquad \dot{\boldsymbol{u}} = \dot{\boldsymbol{u}}_0 + \delta\dot{\boldsymbol{u}}, \qquad \ddot{\boldsymbol{u}} = \ddot{\boldsymbol{u}}_0 + \delta\ddot{\boldsymbol{u}} \tag{14}$$

The first term in Eq. (9) can be expanded as follows

$$\begin{aligned}\frac{\partial}{\partial t}[\boldsymbol{A}(\boldsymbol{x})\boldsymbol{w}] &= \boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{w}} + \frac{\partial \boldsymbol{A}(\boldsymbol{x})}{\partial t}\boldsymbol{w} = \boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{w}} + \dot{x}_k\frac{\partial \boldsymbol{A}}{\partial x_k}\boldsymbol{w} \\ &= \boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{w}} + \dot{x}_k\frac{\partial a_{ij}}{\partial x_k}w_j = \boldsymbol{A}(\boldsymbol{x})\dot{\boldsymbol{w}} + \boldsymbol{E}(\boldsymbol{x}, \boldsymbol{w})\dot{\boldsymbol{x}}\end{aligned} \tag{15}$$

where each element of the $\boldsymbol{E}$ matrix can be expressed as

$$e_{ik} = \frac{\partial a_{ij}}{\partial x_k}w_j \tag{16}$$

Its linearization is given by

$$\begin{aligned}&\boldsymbol{A}(\boldsymbol{x}_0)\dot{\boldsymbol{w}}_0 + \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\delta\boldsymbol{x}\right)\dot{\boldsymbol{w}}_0 + \boldsymbol{A}(\boldsymbol{x}_0)\delta\dot{\boldsymbol{w}} + \boldsymbol{E}(\boldsymbol{x}_0, \boldsymbol{w}_0)\dot{\boldsymbol{x}}_0 \\ &\quad + \left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}}\delta\boldsymbol{w}\right)\dot{\boldsymbol{x}}_0 + \boldsymbol{E}(\boldsymbol{x}_0, \boldsymbol{w}_0)\delta\dot{\boldsymbol{x}}\end{aligned} \tag{17}$$

where $(\partial \boldsymbol{E}/\partial \boldsymbol{x})\delta\boldsymbol{x}$ and $(\partial \boldsymbol{E}/\partial \boldsymbol{w})\delta\boldsymbol{w}$ are matrices with coefficients

$$\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{x}}\delta\boldsymbol{x}\right)_{ik} = \frac{\partial^2 a_{ij}}{\partial x_m \partial x_k}w_j\delta x_m \tag{18}$$

$$\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}}\delta\boldsymbol{w}\right)_{ik} = \frac{\partial a_{ij}}{\partial x_k}\delta w_j \tag{19}$$

Let

$$\begin{aligned}&\boldsymbol{A}_0 = \boldsymbol{A}(\boldsymbol{x}_0), && \boldsymbol{E}_0 = \boldsymbol{E}(\boldsymbol{x}_0, \boldsymbol{w}_0) \\ &\boldsymbol{F}_0 = \bar{\boldsymbol{F}}(\boldsymbol{w}_0, \boldsymbol{x}_0, \dot{\boldsymbol{x}}_0), && \boldsymbol{H}_0 = \frac{\partial \bar{\boldsymbol{F}}}{\partial \boldsymbol{w}}(\boldsymbol{w}_0, \boldsymbol{x}_0, \dot{\boldsymbol{x}}_0) \\ &\boldsymbol{G}_0 = \frac{\partial \bar{\boldsymbol{F}}}{\partial \boldsymbol{x}}(\boldsymbol{w}_0, \boldsymbol{x}_0, \dot{\boldsymbol{x}}_0), && \boldsymbol{C}_0 = \frac{\partial \bar{\boldsymbol{F}}}{\partial \dot{\boldsymbol{x}}}(\boldsymbol{w}_0, \boldsymbol{x}_0, \dot{\boldsymbol{x}}_0)\end{aligned} \tag{20}$$

The linearization of the second term in Eq. (9) around the equilibrium position $(\boldsymbol{w}_0, \boldsymbol{x}_0, \dot{\boldsymbol{x}}_0)$ can be written as

$$\bar{\boldsymbol{F}}(\boldsymbol{w}, \boldsymbol{x}, \dot{\boldsymbol{x}}) = \boldsymbol{F}_0 + \boldsymbol{H}_0\delta\boldsymbol{w} + \boldsymbol{G}_0\delta\boldsymbol{x} + \boldsymbol{C}_0\delta\dot{\boldsymbol{x}} \tag{21}$$

Hence, the complete linearized fluid equation is

$$\begin{aligned}&\boldsymbol{A}_0(\dot{\boldsymbol{w}}_0 + \delta\dot{\boldsymbol{w}}) + \boldsymbol{E}_0(\dot{\boldsymbol{x}}_0 + \delta\dot{\boldsymbol{x}}) + \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\delta\boldsymbol{x}\right)_0\dot{\boldsymbol{w}}_0 + \left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{x}}\delta\boldsymbol{x}\right. \\ &\quad \left. + \frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}}\delta\boldsymbol{w}\right)_0\dot{\boldsymbol{x}}_0 + \boldsymbol{F}_0 + \boldsymbol{H}_0\delta\boldsymbol{w} + \boldsymbol{G}_0\delta\boldsymbol{x} + \boldsymbol{C}_0\delta\dot{\boldsymbol{x}} = 0\end{aligned} \tag{22}$$

If $\boldsymbol{K}_0$ and $\boldsymbol{D}_0$ denote, respectively, the finite element stiffness and damping matrices of the structure at the equilibrium point, the linearization of the left-hand side of the structural equation of motion [see Eq. (10)] can be expressed as

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_0 + \boldsymbol{M}\delta\ddot{\boldsymbol{u}} + \boldsymbol{f}_{\text{int}}(\boldsymbol{u}_0, \dot{\boldsymbol{u}}_0) + \boldsymbol{D}_0\delta\dot{\boldsymbol{u}} + \boldsymbol{K}_0\delta\boldsymbol{u} \tag{23}$$

The linearization of the external forces $\boldsymbol{f}_{\text{ext}}$ is given by

$$\boldsymbol{f}_{\text{ext}}(\boldsymbol{u}, \boldsymbol{w}) = \boldsymbol{f}_{\text{ext}}(\boldsymbol{u}_0, \boldsymbol{w}_0) + \boldsymbol{K}_{f_0}\delta\boldsymbol{u} + \boldsymbol{P}_0\delta\boldsymbol{w} \tag{24}$$

where

$$\boldsymbol{K}_{f_0} = \frac{\partial \boldsymbol{f}_{\text{ext}}}{\partial \boldsymbol{u}}(\boldsymbol{u}_0, \boldsymbol{w}_0), \qquad \boldsymbol{P}_0 = \frac{\partial \boldsymbol{f}_{\text{ext}}}{\partial \boldsymbol{w}}(\boldsymbol{u}_0, \boldsymbol{w}_0) \tag{25}$$

Hence, the complete linearized structural equation of motion can be written as

$$\boldsymbol{M}\delta\ddot{\boldsymbol{u}} + \boldsymbol{D}_0\delta\dot{\boldsymbol{u}} + (\boldsymbol{K}_0 - \boldsymbol{K}_{f_0})\delta\boldsymbol{u} = \boldsymbol{f}_{\text{ext}\,0} - \boldsymbol{M}\ddot{\boldsymbol{u}}_0 - \boldsymbol{f}_{\text{int}\,0} + \boldsymbol{P}_0\delta\boldsymbol{w} \tag{26}$$

By definition, the equilibrium point ($\boldsymbol{w}_0$, $\dot{\boldsymbol{w}}_0$, $\boldsymbol{x}_0$, $\dot{\boldsymbol{x}}_0$, $\ddot{\boldsymbol{x}}_0$, $\boldsymbol{u}_0$, $\dot{\boldsymbol{u}}_0$, $\ddot{\boldsymbol{u}}_0$) satisfies the governing equations (9–11) as well as the following steady-state conditions:

$$\dot{\boldsymbol{w}}_0 = 0, \qquad \boldsymbol{F}_0 = 0, \qquad \dot{\boldsymbol{x}}_0 = 0, \qquad \dot{\boldsymbol{x}}_0 = 0$$

$$\tilde{\boldsymbol{K}}\boldsymbol{x}_0 = \boldsymbol{K}_c\boldsymbol{u}_0, \qquad \ddot{\boldsymbol{u}}_0 = 0, \qquad \dot{x}_{0_k}\left(\frac{\partial a_{ij}}{\partial x_k}\right)_0 = 0 \tag{27}$$

It follows that

$$\boldsymbol{A}_0\dot{\boldsymbol{w}}_0 + \boldsymbol{E}_0\dot{\boldsymbol{x}}_0 + \boldsymbol{F}_0 = 0, \qquad \left(\frac{\partial \boldsymbol{A}}{\partial \boldsymbol{x}}\delta\dot{\boldsymbol{x}}\right)_0\dot{\boldsymbol{w}}_0 = 0$$

$$\dot{x}_{0_k}\left(\frac{\partial a_{ij}}{\partial x_k}\right)_0\delta w_j = 0, \qquad \left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}}\delta\boldsymbol{w}\right)_0\dot{\boldsymbol{x}}_0 = 0$$

$$\boldsymbol{M}\ddot{\boldsymbol{u}}_0 + \boldsymbol{f}_{\text{int}\,0} = \boldsymbol{f}_{\text{ext}\,0} \tag{28}$$

Therefore, the linearized flow equation can be rewritten as

$$\boldsymbol{A}_0\delta\dot{\boldsymbol{w}} + \boldsymbol{H}_0\delta\boldsymbol{w} + \boldsymbol{G}_0\delta\boldsymbol{x} + (\boldsymbol{E}_0 + \boldsymbol{C}_0)\delta\dot{\boldsymbol{x}} = 0 \tag{29}$$

and the linearized structural and mesh motion equations simplify to

$$\boldsymbol{M}\delta\ddot{\boldsymbol{u}} + \boldsymbol{D}_0\delta\dot{\boldsymbol{u}} + \boldsymbol{K}_{s_0}\delta\boldsymbol{u} = \boldsymbol{P}_0\delta\boldsymbol{w} \tag{30}$$

where

$$\boldsymbol{K}_{s_0} = \boldsymbol{K}_0 - \boldsymbol{K}_{f_0} \tag{31}$$

and

$$\tilde{\boldsymbol{M}}\delta\ddot{\boldsymbol{x}} + \tilde{\boldsymbol{D}}\delta\dot{\boldsymbol{x}} + \tilde{\boldsymbol{K}}\delta\boldsymbol{x} = \boldsymbol{K}_c\delta\boldsymbol{u} \tag{32}$$

It should be noted that the adjusted structural stiffness matrix $\boldsymbol{K}_{s_0}$ is not guaranteed to be positive definite. When it is not, the aeroelastic system will exhibit divergence.

In the sequel, the subscript 0 and the prefix δ are dropped to simplify the notation. This should not cause any confusion because in our subsequent analyses, all matrices are evaluated at the equilibrium point, and all vectors are perturbation quantities.

Finally, Eqs. (29–32) can be grouped into the following three-way coupled system:

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{\boldsymbol{M}} & 0 \\ 0 & 0 & \boldsymbol{M} \end{pmatrix}\begin{pmatrix} \ddot{\boldsymbol{w}} \\ \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} + \begin{pmatrix} \boldsymbol{A} & (\boldsymbol{E}+\boldsymbol{C}) & 0 \\ 0 & \tilde{\boldsymbol{D}} & 0 \\ 0 & 0 & \boldsymbol{D} \end{pmatrix}\begin{pmatrix} \dot{\boldsymbol{w}} \\ \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} + \begin{pmatrix} \boldsymbol{H} & \boldsymbol{G} & 0 \\ 0 & \tilde{\boldsymbol{K}} & -\boldsymbol{K}_c \\ -\boldsymbol{P} & 0 & \boldsymbol{K}_s \end{pmatrix}\begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{x} \\ \boldsymbol{u} \end{pmatrix} = 0 \tag{33}$$

Such a representation is, however, inconvenient for any practical purpose. Nevertheless, it reveals the unsymmetric nature of the coupled system and highlights the different orders of the differential equations governing the fluid and structural systems.

Aeroelastic Eigenvalue Problem

For simplicity, and without any loss of generality, we consider in the sequel the case where the dynamic fluid mesh is represented by a network of edge-springs as proposed by Batina.[15] This is a particular instance of the general formulation arrising from Eqs. (3) and (7), where $\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{D}} = 0$. We also define

$$\boldsymbol{K}^* = \tilde{\boldsymbol{K}}^{-1}\boldsymbol{K}_c \tag{34}$$

and eliminate $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ as follows:

$$\boldsymbol{x} = \tilde{\boldsymbol{K}}^{-1}\boldsymbol{K}_c\boldsymbol{u} = \boldsymbol{K}^*\boldsymbol{u}, \qquad \dot{\boldsymbol{x}} = \boldsymbol{K}^*\dot{\boldsymbol{u}} \tag{35}$$

Next, let us introduce the variable $\boldsymbol{y} = \dot{\boldsymbol{u}}$ and reduce the second-order coupled system in Eq. (33) to the following first-order one:

$$\begin{pmatrix} \boldsymbol{A} & 0 & 0 \\ 0 & \boldsymbol{M} & 0 \\ 0 & 0 & -\boldsymbol{I} \end{pmatrix}\begin{pmatrix} \dot{\boldsymbol{w}} \\ \boldsymbol{y} \\ \boldsymbol{u} \end{pmatrix} + \begin{pmatrix} \boldsymbol{H} & (\boldsymbol{E}+\boldsymbol{C})\boldsymbol{K}^* & \boldsymbol{G}\boldsymbol{K}^* \\ -\boldsymbol{P} & 0 & \boldsymbol{K}_s \\ 0 & \boldsymbol{I} & 0 \end{pmatrix}\begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{y} \\ \boldsymbol{u} \end{pmatrix} = 0 \tag{36}$$

This first-order system of differential equations can be rewritten as

$$\dot{\boldsymbol{q}} = \boldsymbol{N}\boldsymbol{q} \tag{37}$$

where

$$\boldsymbol{q} = \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{y} \\ \boldsymbol{u} \end{pmatrix}, \qquad \boldsymbol{N} = \begin{pmatrix} -\boldsymbol{A}^{-1}\boldsymbol{H} & -\boldsymbol{A}^{-1}(\boldsymbol{E}+\boldsymbol{C})\boldsymbol{K}^* & -\boldsymbol{A}^{-1}\boldsymbol{G}\boldsymbol{K}^* \\ \boldsymbol{M}^{-1}\boldsymbol{P} & 0 & -\boldsymbol{M}^{-1}\boldsymbol{K}_s \\ 0 & \boldsymbol{I} & 0 \end{pmatrix} \tag{38}$$

and its corresponding eigenvalue problem is

$$(\boldsymbol{N} - \lambda_i\boldsymbol{I})\boldsymbol{q}_i = 0 \tag{39}$$

where λ_i is an eigenvalue and $\boldsymbol{q}_i$ is an eigenvector. In general, the above eigenvalue problem has complex solutions. It is of great interest in aeroelasticity because the sign of the real part of each eigenvalue λ_i determines whether the aeroelastic system is stable or unstable. The main objective of this paper is to present a numerical method for solving this eigenvalue problem that is deemed more general than those published in the past. In particular, we are interested in the eigensolutions that are the least stable, that is, for which $\Re(\lambda_i)$ is the largest negative number or is positive.

Solution Method

Structural Model Reduction

If n_S denotes the number of structural degrees of freedom and n_F the number of fluid unknowns, the coupled fluid–structure eigenvalue problem in Eq. (39) admits $2n_S + n_F$ solutions. However, an analyst is in general interested in only a relatively small number of eigensolutions. For example, for flutter analysis, the eigenvalues of interest are those corresponding to a low frequency and/or small damping. Furthermore, the eigensolutions of interest commonly involve only a subset of the structural "dry" (uncoupled) modes. For these reasons, it is reasonable, but not mandatory, to represent the structure by its first m dry modes and therefore to switch from the displacement variable $\boldsymbol{u}$ to the modal variable (or generalized coordinate) $\boldsymbol{u}_m$ defined by

$$\boldsymbol{u} = \boldsymbol{\Psi}_m\boldsymbol{u}_m \tag{40}$$

where $\boldsymbol{\Psi}$ is an $n_S \times m$ matrix storing the m-chosen dry eigenvectors $\boldsymbol{\Psi}_i$. These eigenvectors are solution of the structural eigenvalue problem

$$\boldsymbol{K}\Psi_i = \omega_i^2\boldsymbol{M}\Psi_i \tag{41}$$

In general, m is relatively small compared to n_S (say, $10 \le m \le 30$) and therefore the reduced mass and stiffness matrices

$$\boldsymbol{M}_m = \boldsymbol{\Psi}_m^T\boldsymbol{M}\boldsymbol{\Psi}_m, \qquad \boldsymbol{K}_m = \boldsymbol{\Psi}_m^T\boldsymbol{K}_s\boldsymbol{\Psi}_m \tag{42}$$

and the reduced matrices

$$\boldsymbol{K}_m^* = \boldsymbol{K}^* \boldsymbol{\Psi}_m, \qquad \boldsymbol{P}_m = \boldsymbol{\Psi}_m^T \boldsymbol{P} \tag{43}$$

have small sizes. Substituting Eq. (40) into Eq. (36) leads to the reduced first-order system of differential equations

$$\dot{\boldsymbol{q}}_m = \boldsymbol{N}_m \boldsymbol{q}_m \tag{44}$$

where

$$\boldsymbol{q}_m = \begin{pmatrix} \boldsymbol{w} \\ \boldsymbol{y}_m \\ \boldsymbol{u}_m \end{pmatrix}$$

$$\boldsymbol{N}_m = \begin{pmatrix} -\boldsymbol{A}^{-1}\boldsymbol{H} & -\boldsymbol{A}^{-1}(\boldsymbol{E}+\boldsymbol{C})\boldsymbol{K}_m^* & -\boldsymbol{A}^{-1}\boldsymbol{G}\boldsymbol{K}_m^* \\ \boldsymbol{M}_m^{-1}\boldsymbol{P}_m & 0 & -\boldsymbol{M}_m^{-1}\boldsymbol{K}_m \\ 0 & \boldsymbol{I} & 0 \end{pmatrix} \tag{45}$$

The corresponding reduced eigenvalue problem is

$$(\boldsymbol{N}_m - \mu_i \boldsymbol{I})\boldsymbol{q}_{m_i} = 0 \tag{46}$$

Remark: It is also possible to use a modal representation for reducing the order of the fluid model[10] and therefore reducing further the size of the matrix $\boldsymbol{N}_m$. However, this reduction becomes prohibitive for a refined mesh because all the eigenvectors of the fluid operator have to be computed. An alternative is the approximate balancing of Baker,[11] but it requires computing the eigensolutions of a more complex system and suffers from lack of convergence guarantees because the matrix on which the power method is applied is varied at each iteration.

Method of Inverse Orthogonal Iteration

We are interested in the low end of the spectrum of $\boldsymbol{N}_m$, which can be computed by a straightforward generalization of the inverse power iteration algorithm known as the method of inverse orthogonal iteration. However, rather than operating directly on $\boldsymbol{N}_m$, we select to operate on $(\boldsymbol{I} - h\boldsymbol{N}_m)$, $h > 0$, for reasons that will become clearer in the next section. The reader is reminded that $\boldsymbol{N}_m$ and $(\boldsymbol{I} - h\boldsymbol{N}_m)$ share the same eigenvectors, and that if ν_i is an eigenvalue of $(\boldsymbol{I} - h\boldsymbol{N}_m)$,

$$\mu_i = (1 - \nu_i)/h \tag{47}$$

is the corresponding eigenvalue of $\boldsymbol{N}_m$.

Let $r = 2m + n_F$. Given an $r \times p$ matrix $\boldsymbol{T}^{(0)}$ with $p \le m$ orthonormal columns, the following method of inverse orthogonal iteration generates a sequence of matrices $\boldsymbol{T}^{(k)} \in R^{r \times p}$, whose ranges $R[\boldsymbol{T}^{(k)}]$ converge to the subspace spanned by the p eigenvectors associated with the smallest eigenvalues of $(\boldsymbol{I} - h\boldsymbol{N}_m)$

For $k = 1, 2, \ldots$
Solve $(\boldsymbol{I} - h\boldsymbol{N})\boldsymbol{Z}^{(k)} = \boldsymbol{T}^{(k-1)}$
Orthonormalize $\boldsymbol{Z}^{(k)}$ using a QR factorization: $\boldsymbol{T}^{(k)}\boldsymbol{R}^{(k)} = \boldsymbol{Z}^{(k)}$

The matrix product $\boldsymbol{T}^{(k)^T}\boldsymbol{Z}^{(k)}$, where the superscript T designates a transpose, converges to a $p \times p$ block upper triangular with either 1×1 or 2×2 diagonal blocks. These diagonal blocks easily lead to the smallest eigenvalues ν_i, from which one can deduce the smallest eigenvalues μ_i. A good approximation of the eigenvectors $\boldsymbol{q}_{m_i}$, $i = 1, \ldots, p$ can then be obtained by computing the complete set of eigenvectors $\boldsymbol{S}_p$ of the $p \times p$ matrix $\boldsymbol{T}^{(k)^T}\boldsymbol{Z}^{(k)}$ and setting $\boldsymbol{Q}_{m_p} = \boldsymbol{T}^{(k)}\boldsymbol{S}_p$.

Reusing the Components from an Implicit Nonlinear Unsteady Solver

The computationally intensive kernel of the method of inverse orthogonal iteration is the one providing at each iteration the solution of a problem of the form

$$(\boldsymbol{I} - h\boldsymbol{N})\boldsymbol{z} = \boldsymbol{t} \tag{48}$$

We propose to solve the problem by a block-Jacobi scheme. In the next section, it is shown that it is possible to produce a good approximation of $\boldsymbol{z}$ that accelerates the convergence of this iterative solution procedure.

From Eq. (45), it follows that at each block-Jacobi iteration n, the flow problem associated with Eq. (48) is

$$(\boldsymbol{A} + h\boldsymbol{H})\boldsymbol{w}^{(n)} = \boldsymbol{A}\boldsymbol{w}^{(n-1)} - h(\boldsymbol{E}+\boldsymbol{C})\boldsymbol{K}^*\boldsymbol{y}_m^{(n-1)} - h\boldsymbol{G}\boldsymbol{K}^*\boldsymbol{u}_m^{(n-1)} \tag{49}$$

If h is identified with a time-step Δt, the problem becomes identical to that which arises at every time-step of the solution of a transient aeroelastic problem by an implicit scheme. Hence, each iteration of the eigensolver proposed in this paper can be carried out by reutilizing an implicit nonlinear unsteady flow solver. Given the major efforts that have been invested in the last decade to develop implicit Euler and Navier–Stokes unsteady flow solvers, this is a rather important feature of our aeroelastic eigensolver.

Accelerating the Convergence of the Block-Jacobi Algorithm

At the kth step of the method of inverse orthogonal iteration, $\boldsymbol{Z}^{(k-1)}$ is readily available, and the objective is to solve p systems of the form

$$(\boldsymbol{I} - h\boldsymbol{N})\boldsymbol{z}_i^{(k)} = \boldsymbol{t}_i^{(k-1)} \tag{50}$$

using a block-Jacobi scheme. A good initial guess for $\boldsymbol{z}_i^{(k)}$ can be constructed by the following Rayleigh–Ritz procedure. First, an initial guess is selected in the subspace spanned by $R[\boldsymbol{Z}^{(k-1)}]$

$$\boldsymbol{z}_i^{(k)^{(0)}} = \boldsymbol{Z}^{(k-1)}\boldsymbol{\lambda} \tag{51}$$

then $\boldsymbol{\lambda}$ is computed to minimize the initial block-Jacobi residual $||\boldsymbol{t}_i^{(k-1)} - (\boldsymbol{I} - h\boldsymbol{N})\boldsymbol{Z}^{(k-1)}\boldsymbol{\lambda}||_2$. This leads to

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{\lambda}}\left[\boldsymbol{\lambda}^T\boldsymbol{T}^{(k-2)^T}\boldsymbol{T}^{(k-2)}\boldsymbol{\lambda} - 2\boldsymbol{\lambda}^T\boldsymbol{T}^{(k-2)^T}\boldsymbol{t}_i^{(k-1)} + \boldsymbol{t}_i^{(k-1)^T}\boldsymbol{t}_i^{(k-1)}\right] = 0 \tag{52}$$

Given that $\boldsymbol{T}^{(k-2)}$ is an orthonormal matrix, it follows that the solution of Eq. (52) is

$$\boldsymbol{\lambda} = \boldsymbol{T}^{(k-2)^T}\boldsymbol{t}_i^{(k-1)} \tag{53}$$

Hence, the proposed initial guess for $\boldsymbol{z}_i^{(k)}$ is

$$\boldsymbol{z}_i^{(k)^{(0)}} = \boldsymbol{Z}^{(k-1)}\boldsymbol{T}^{(k-2)^T}\boldsymbol{t}_i^{(k-1)} \tag{54}$$

However, to minimize the storage requirements, the $r \times p$ matrix $\boldsymbol{Z}^{(k-1)}$ is not stored while applying the method of inverse orthogonal iteration; instead, the smaller $p \times p$ $\boldsymbol{R}^{(k-1)}$ matrix is stored. At each iteration of the eigensolver $\boldsymbol{Z}^{(k-1)} = \boldsymbol{R}^{(k-1)}\boldsymbol{T}^{(k-1)}$ is reconstructed, and therefore the initial guess $\boldsymbol{z}_i^{(k)^{(0)}}$ is computed as follows:

$$\boldsymbol{z}_i^{(k)^{(0)}} = \boldsymbol{R}^{(k-1)}\boldsymbol{T}^{(k-1)}\boldsymbol{T}^{(k-2)^T}\boldsymbol{t}_i^{(k-1)} \tag{55}$$

Using a matrix notation, the initial guesses for all p systems to be solved can be written as

$$\boldsymbol{Z}^{(k)^{(0)}} = \boldsymbol{R}^{(k-1)}\boldsymbol{T}^{(k-1)}\boldsymbol{T}^{(k-2)^T}\boldsymbol{T}^{(k-1)} \tag{56}$$

Solvers and Parallel Implementation

As stated earlier, a distinctive feature of the proposed aeroelastic eigensolver is that it can reuse an existing code for the solution of nonlinear transient aeroelastic response problems. In the flow solver of our aeroelastic heterogeneous software package,[4,17] the spatial discretization of the Euler or Navier–Stokes equations is carried out on unstructured meshes. It combines a Roe upwind scheme for computing the convective fluxes, and a Galerkin centered method for evaluating the viscous terms. Second-order spatial accuracy for the

convective terms is achieved through the use of a piecewise linear interpolation method that follows the principle of the MUSCL (monotonic upwind scheme for conservative laws) procedure.[18] Temporal discretization can be achieved either by a first-order time-accurate backward Euler scheme or a second-order time-accurate implicit scheme based on the defect correction method.[19] An ALE formulation is incorporated in the nonlinear unsteady flow solver to allow the grid points to displace in a Lagrangian fashion, or be held fixed in an Eulerian manner, or be moved in some specified way to give a continuous and automatic rezoning capability, depending on the needs of the physical problem to be solved. The numerical algorithms used with this ALE formulation satisfy the geometric conservation laws[20] that govern flow computations on dynamic meshes.

The structure is represented by a finite element model, and its dynamic behavior can be predicted using either the true displacement, velocity, and acceleration degrees of freedom or the modal coordinates discussed in solution method section.

In this work, the unstructured dynamic fluid mesh is represented by a network of edge-based springs ($\tilde{\boldsymbol{M}} = \tilde{\boldsymbol{D}} = 0$). The grid points located on the upstream and downstream boundaries are held fixed. The motion of those points located on the fluid–structure boundary Γ is determined from the structure surface motion and/or deformation.

In general, the fluid and structure meshes have two independent representations of the physical fluid–structure interface, which often leads to fluid and structure meshes that are incompatible at their interface boundaries. In that case, we use the Matcher software and algorithms described in previous work by the authors[21,22] for properly exchanging the aerodynamic and elastodynamic data between the flow and structural solvers.

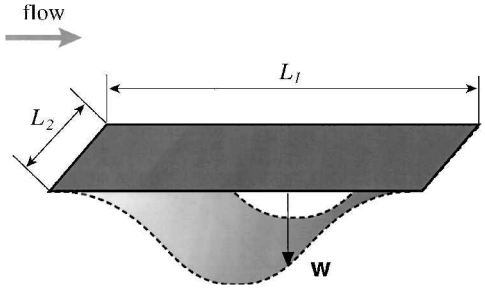

Fig. 1 Aeroelastic panel problem.

Our flow, structure, and dynamic mesh solvers are implemented as separate programs that communicate via a software communication library. Each of these three codes is parallelized using the mesh partitioning and message-passing paradigms.[23,24] The message passing interface (MPI) is used for interprocessor communication within each single discipline code as well as between the fluid, structure, and dynamic mesh codes.

Finally, we note that in this work, the derivatives incurred by the linearization process have been either computed numerically or generated with the software ADIFOR.[25]

Applications

First, we consider the three-dimensional aeroelastic analysis on an IBM SP2 parallel processor of a flat panel with infinite aspect ratio in supersonic airstreams. The panel has a length $L = 0.5$ m, a uniform thickness $h = 1.35 \times 10^{-3}$ m, a Young modulus $E = 7.728 \times 10^{10}$ N/m^2, a Poisson ratio $\nu = 0.33$, a density $\rho = 2710$ kg/m^3, and it is clamped at both ends (Fig. 1). Its finite element discretization contains 100 triangular shell elements, 102 nodes, and 612 DOF. Only the first 10 dry modes associated with this finite element model are retained for the aeroelastic stability analysis reported here. The flow domain above this panel is discretized into 20,250 vertices and 94,080 unstructured tetrahedra. A slip condition is imposed at the fluid–structure boundary. The inflow conditions are set to $\rho_\infty = 0.4$ kg/m^3 and $c_\infty = 300$ m/s. The Mach number is varied between 1.8 and 2.45.

The analytical solution of the instability problem of flat panels with infinite aspect ratio in supersonic airstreams assumes a shallow shell theory for the structure and a linearized formulation for the flow problem (piston theory). Within this classical approach, the dynamics of the focus-coupled fluid–structure system are governed by a fourth-order partial differential equation, and the flutter condition is obtained by analyzing the roots of the corresponding characteristic equation.[26] For the panel considered in this section, the classical linear theory predicts flutter at the critical Mach number $M_\infty^{\mathrm{cr}} = 2.27$. Here we apply the proposed aeroelastic eigenvalue formulation and solution procedure to compute the stability limit of this panel. More specifically, we use six trial vectors to initialize the method of inverse orthogonal iteration.

Figures 2 and 3 show the real and imaginary parts of the first two aeroelastic modes of the panel as a function of the Mach number. For comparison, we note that the frequencies of the first two dry modes are $\omega_1 = 197.1$ rad/s and $\omega_2 = 541.8$ rad/s. The continuous

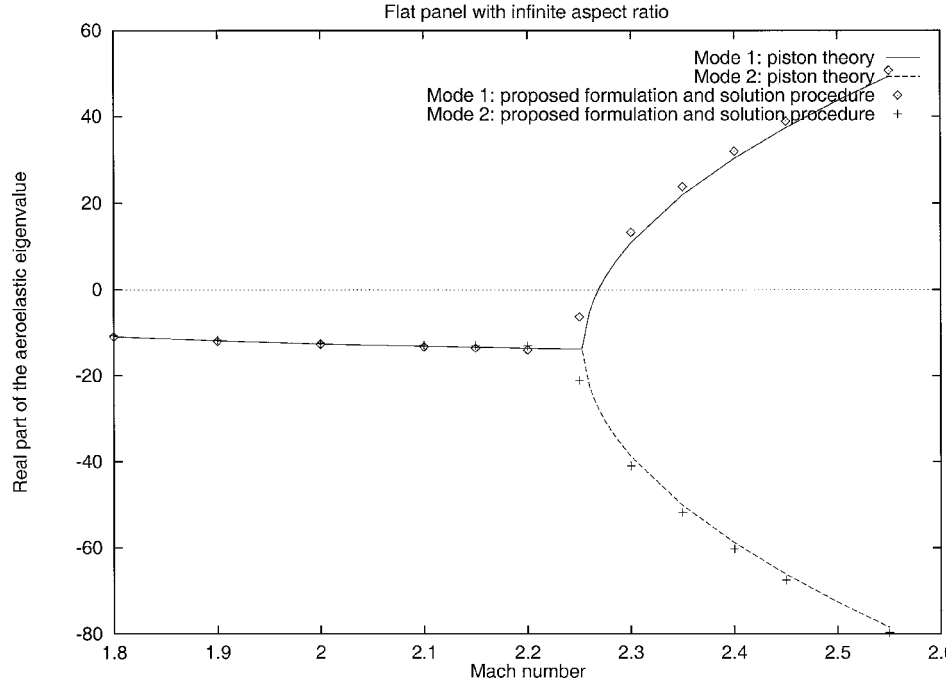

Fig. 2 Variation of the real part of the aeroelastic eigenvalue with the Mach number.

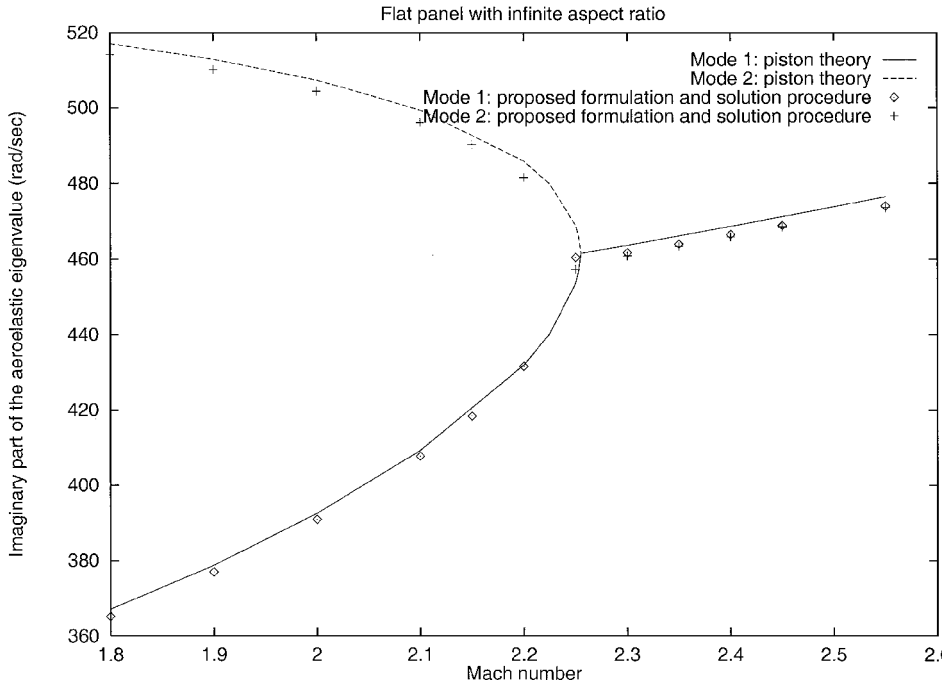

Fig. 3 Variation of the imaginary part of the aeroelastic eigenvalue with the Mach number.

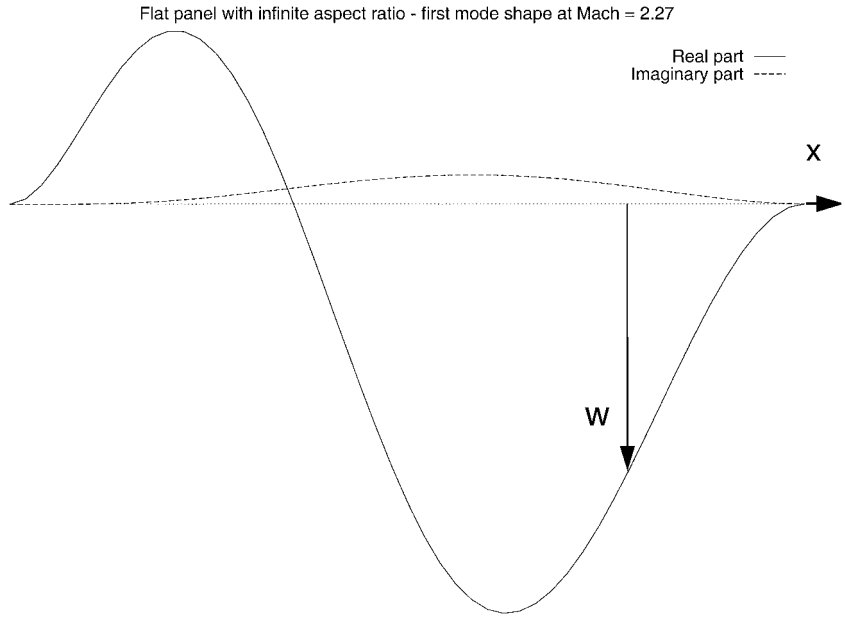

Fig. 4 Real and imaginary parts of the first aeroelastic mode shape.

curves correspond to the solutions obtained by the piston theory, and the superimposed discrete points correspond to those computed by our new method. These figures demonstrate an excellent correlation between our computational results and the analytical solution, and a relative error in the neighborhood of 0.5.% The reader can observe that the first two aeroelastic eigenvalues coalesce before the flutter speed. Before coalescence, these eigenvalues reveal the same amount of damping. After coalescence, the frequencies of the two modes become equal, but one mode becomes unstable and the other witnesses an increase in damping. Because the structural model constructed with finite elements is slightly more flexible than that derived by the analytical method, the computational methodology proposed in this paper predict coalescence and flutter slightly earlier than predicted by the piston theory. However, as already stated, the relative error between the results produced by our new method and those of the piston theory is less than 0.5%.

Finally, Fig. 4 shows the real and imaginary parts of the first wet mode shape of the panel computed at $M_\infty = 2.27$. Unlike in the dry case, this mode shape is not symmetric. It is not antisymmetric, either. Plotting complex eigenvectors is difficult in general because of the lack of standard normalization procedures. However, the fact that we have been able to find a normalization scheme where the amplitude of the imaginary part of the first wet mode shape is very small compared with that of the real part indicates that the first aeroelastic mode does not contribute an important phase shift to the motion of the panel.

Next, we consider the flutter analysis of the AGARD Wing 445.6 (Fig. 5) (Ref. 27.) This wing is an AGARD standard aeroelastic configuration with a 45 deg quarter-chord sweep angle, a panel aspect ratio of 1.65, a taper ratio of 0.66, and a NACA 65A004 airfoil section. The model selected here is the so-called 2.5-ft weakened model 3, whose measured modal frequencies and wind-tunnel flutter

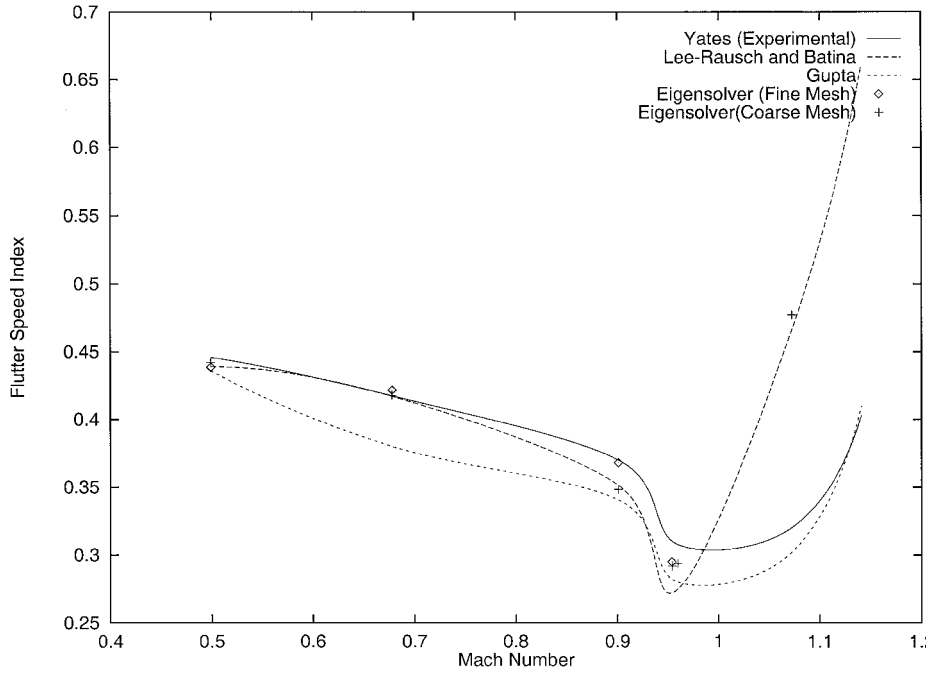

Fig. 5 Flutter speed index (AGARD Wing 445.6).

test results are reported in Yates[27] and for which computational aeroelastic data can be found in Lee-Rausch and Batina[28] and in Gupta.[29]

An undamped finite element model of the wing with 800 triangular composite shell elements and 2646 DOF is constructed using the information given in Yates.[27] It yields natural mode shapes and frequencies that are similar to those derived experimentally. More specifically, the frequencies associated with the first four natural modes of this finite element model are, respectively, 9.83, 39.54, 50.50, and 96.95 Hz. They differ from the experimental ones by only 2.5, 3.6, 4.5, and 5.9%, respectively.

Two three-dimensional unstructured tetrahedral CFD Euler meshes were generated using GHS3D[30] and Gridgen, respectively. The first mesh has only 22,014 vertices. The second mesh is finer: it contains 199,680 vertices and serves the purpose of checking mesh convergence.

All computations discussed next are carried out on an IBM SP2 parallel computer. A single processor is assigned to the finite element structural model, while 3 and 10 processors are assigned the coarse and fine CFD meshes, respectively.

We apply the eigensolution method proposed in this paper to the prediction of the flutter speed index as a function of the Mach number. More specifically, we represent the structure by its first 10 dry modes and employ 6 trial vectors to initialize the method of inverse orthogonal iteration. For each target Mach number, we set the freestream density as in Ref. 27, and vary the freestream pressure until we observe the onset of flutter characterized by a vanishing real part of a computed eigenvalue. We report our results in Fig. 5 and compare them with the experimental data published in Ref. 27, and the computational results published in references from Lee-Rausch and Batina[28] and Gupta.[29] Note that the flutter envelopes reported by Lee-Rausch and Batina and by Gupta are derived from the solution of nonlinear transient aeroelastic response problems where the unsteady flow computations are performed using an Euler solver.

The results reported in Fig. 5 show that in the range $0.499 \leq M_\infty \leq 0.960$, the flutter speed indices predicted by our computational methodology compare favorably with the experimental data, and with both computational results published in Refs. 28 and 29. In the supersonic regime, our results compare more favorably with experimental data than those of Lee-Rausch and Batina, but less favorably than those of Gupta. Indeed, in the supersonic regime, our computational results validate those of Lee-Rausch and Batina. As can be seen, the finer mesh seems to indicate an onset of the transonic dip at a very slightly higher Mach number than the coarser mesh. Otherwise, the results predicted by our eigensolver using the coarse mesh agree with those obtained using the finer mesh, which establishes mesh convergence.

Conclusions

In this paper, we have described a CFD-based linearized method for computing an arbitrary number of eigensolutions of a given aeroelastic problem. The proposed method can be implemented by reengineering the computational modules of a nonlinear transient aeroelastic simulation capability. In particular, it reutilizes existing CFD-based unsteady flow solvers. It is applicable in subsonic, transonic, and supersonic flow regimes, independently from the frequency or damping level of the target aeroelastic modes. It is based on the computation of the complex eigensolution of a carefully linearized fluid–structure interaction problem by the inverse orthogonal iteration algorithm. We have illustrated this method with the stability analysis of a flat panel with infinite aspect ratio in supersonic airstreams. For this academic aeroelastic problem, the results produced by our computational method are in excellent agreement with those predicted analytically by the piston theory. We have also applied it to the flutter analysis of the AGARD Wing 445.6 and predicted flutter speed indices that are in very good agreement with experimental data in the range $0.499 \leq M_\infty \leq 0.960$ and in excellent agreement with the computational results obtained by another investigator[28] in the range $0.499 \leq M_\infty \leq 1.14$ by solving nonlinear response problems in the time domain. In particular, the proposed linearized method captured well the transonic dip.

References

[1]Rizetta, D. P., and Visbal, M. R., "Comparative Numerical Study of Two Turbulence Models for Airfoil Static and Dynamic Stall," AIAA Paper 92-4649, June 1992.

[2]Pramono, E., and Weeratunga, S. K., "Aeroelastic Computations for Wings Through Direct Coupling on Distributed-Memory MIMD Parallel Computers," AIAA Paper 94-0095, Jan. 1994.

[3]Piperno, S., Farhat, C., and Larrouturou, B., "Partitioned Procedures for the Transient Solution of Coupled Aeroelastic Problems," *Comput. Meths. Appl. Mech. Engrg.* Vol. 124, 1995, pp. 79–11.

[4]Farhat, C., Lesoinne, M., Chen, P. S., and Lantéri, S., "Parallel Heterogeneous Algorithms for the Solution of Three-Dimensional Transient Coupled Aeroelastic Problems," AIAA Paper 95-1290, April 1995.

[5]Morton, S. A., and Beran, P. S., "Nonlinear Analysis of Airfoil Flutter at Transonic Speeds," AIAA Paper 95-1905, June 1995.

[6]Dat, R., and Meurzec, J. L., "On the Flutter Calculations by the So-Called Reduced Frequency Scanning Method," *Recherche Aérospatiale*, Vol. 133, Nov.–Dec. 1969.

[7]Richardson, J. R., "A More Realistic Method for Routine Flutter Calculations," *Symposium on Structural Dynamics and Aeroelasticity*, AIAA, Boston, MA, 1965.

[8]Hassig, H., "An Approximate True Damping Solution of the Flutter Equation by Determinant Iteration," *Journal of Aircraft*, Vol. 8, No. 11, 1971, pp. 885–889.

[9]Edwards, J. W., "Unsteady Aerodynamic Modeling and Active Aeroelastic Control," Stanford University Dept. of Aeronautics and Astronautics Rept. 504, Ph.D. Dissertation, Stanford Univ., Stanford, CA, 1977.

[10]Dowell, E. H., "Eigenmode Analysis in Unsteady Aerodynamics: Reduced Order Models," AIAA Paper 95-1450, April 1995.

[11]Baker, M. L., Mingori, D. L., and Goggins, P. J., "Approximate Subspace Iteration for Constructing Internally Balanced Reduced Order Models of Unsteady Aerodynamic Systems," AIAA Paper 96-1441, April 1996.

[12]Farhat, C., Lesoinne, M., and Maman, N., "Mixed Explicit/Implicit Time Integration of Coupled Aeroelastic Problems: Three-Field Formulation, Geometric Conservation and Distributed Solution," *International Journal for Numerical Methods in Fluids*, Vol. 21, 1995, pp. 807–835.

[13]Golub, G. H., and Van Loan, C. F., *Matrix Computations*, Johns Hopkins Univ. Press, Baltimore, MD, 1983.

[14]Donea, J., "An Arbitrary Lagrangian–Eulerian Finite Element Method for Transient Fluid-Structure Interactions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 33, 1982 pp. 689–723.

[15]Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," AIAA Paper 89-0115, Jan. 1989.

[16]Lesoinne, M., and Farhat, C., "Stability Analysis of Dynamic Meshes for Transient Aeroelastic Computations," AIAA Paper 93-3325, July 1993.

[17]Farhat, C., "High Performance Simulation of Coupled Nonlinear Transient Aeroelastic Problems," AGARD Rept. R-807, *Special Course on Parallel Computing in CFD (l'Aérodynamique numérique et le calcul en parallèle)*, North Atlantic Treaty Organization, Aalborg, Oct. 1995.

[18]Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V: A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, 1979, pp. 101–136.

[19]Martin, R., and Guillard, H., "A Second-Order Defect Correction Scheme for Unsteady Problems," *Computers and Fluids Journal*, Vol. 25, No. 1, 1996, pp. 9–27.

[20]Lesoinne, M., and Farhat, C., "Geometric Conservation Laws for Aeroelastic Computations Using Unstructured Dynamic Meshes," AIAA Paper 95-1709, June 1995. *Computer Methods in Applied Mechanics and Engineering*, Vol. 134, 1996, pp. 71–90.

[21]Maman, N., and Farhat, C., "Matching Fluid and Structure Meshes for Aeroelastic Computations: A Parallel Approach," *Computer and Structure*, Vol. 54, 1995, pp. 779–785.

[22]Farhat, C., and Lesoinne, M., "A Conservative Algorithm for Exchanging Aerodynamic and Elastodynamic Data in Aeroelastic Systems," AIAA Paper 98-0515, Jan. 1998.

[23]Farhat, C., and Lantéri, S., "Simulation of Compressible Viscous Flows on a Variety of MPPs: Computational Algorithms for Unstructured Dynamic Meshes and Performance Results," *Computer Methods in Applied Mechanics and Engineering*, Vol. 119, 1994, pp. 35–60.

[24]Farhat, C., Lantéri, S., and Simon, H. D., "TOP/DOMDEC, a Software Tool for Mesh Partitioning and Parallel Processing," *Journal of Computing Systems in Engineering*, Vol. 6, 1995, pp. 13–26.

[25]Corliss, G., Griewank, A., and Hovland, P., "ADIFOR—Generating Derivative Codes from FORTRAN Programs," *Scientific Programming*, Vol. 1, No. 1, Fall 1992, pp. 11–30.

[26]Bisplinghoff, R. L., and Ashley, H., "Principles of Aeroelasticity," Dover, New York, 1962.

[27]Yates, E. C., "AGARD Standard Aeroelastic Configuration for Dynamic Response, Candidate Configuration I.—Wing 445.6," NASA TM-100492, 1987.

[28]Lee-Rausch, E. M., and Batina, J. T., "Wing-Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method," AIAA Paper 93-1422, Aug. 1993.

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

[30]George, P. L., "Improvement on Delaunay Based 3D Automatic Mesh Generator," *Finite Elements in Analysis and Design*, Vol. 25, 1997, pp. 297–317.