JOURNAL OF AIRCRAFT
Vol. 38, No. 2, March–April 2001

Calculation of Wing Flutter by a Coupled Fluid-Structure Method

F. Liu,* J. Cai,† and Y. Zhu‡
University of California, Irvine, Irvine, California 92697-3975
H. M. Tsai§ and A. S. F. Wong¶
DSO National Laboratories, Singapore 118230, Republic of Singapore

An integrated computational fluid dynamics (CFD) and computational structural dynamics (CSD) method is developed for the simulation and prediction of flutter. The CFD solver is based on an unsteady, parallel, multiblock, multigrid finite volume algorithm for the Euler/Navier–Stokes equations. The CSD solver is based on the time integration of modal dynamic equations extracted from full finite element analysis. A general multiblock deformation grid method is used to generate dynamically moving grids for the unsteady flow solver. The solutions of the flowfield and the structural dynamics are coupled strongly in time by a fully implicit method. The coupled CFD–CSD method simulates the aeroelastic system directly on the time domain to determine the stability of the aeroelastic system. The unsteady solver with the moving grid algorithm is also used to calculate the harmonic and/or indicial responses of an aeroelastic system, in an uncoupled manner, without solving the structural equations. Flutter boundary is then determined by solving the flutter equation on the frequency domain with the indicial responses as input. Computations are performed for a two-dimensional wing aeroelastic model and the three-dimensional AGARD 445.6 wing. Flutter boundary predictions by both the coupled CFD–CSD method and the indicial method are presented and compared with experimental data for the AGARD 445.6 wing.

I. Introduction

As in many other areas of aerodynamics, the panel method, methods based on the transonic small disturbance equation, and methods based on the full potential equation, successively found their ways into the simulation and prediction of flutter of airfoils and wings since computational fluid dynamics (CFD) was introduced as a tool for aerodynamic research and design. As CFD and computer technology progress, higher-order methods based on the Euler and the Navier–Stokes equations become more attractive because they are able to model more accurately transonic, nonlinear, and viscous effects. Computations have also advanced from two-dimensional problems to fully three-dimensional problems with or without coupled solution of the structural equations. Ballhaus and Goorjian[1] used a two-dimensional Euler code to calculate the indicial response of an airfoil, where the flow calculations were performed decoupled, that is, separately from the solution of the structural equations. Bendiksen and Kousen[2] and Kousen and Bendiksen[3] used an explicit time-accurate Euler code to study nonlinear effects in transonic flutter. With their Euler model coupled with the solution of a two-degrees-of-freedom (DOF) airfoil structural model, they demonstrated the possibility of limit cycle oscillations (LCO) in a transonic flow. Lee-Rausch and Batina[4,5] developed three-dimensional methods for the Euler and Navier–Stokes equations, respectively, for predicting flutter boundaries of three-dimensional wings. Both the indicial method and the coupled CFD–CSD method were used. Guruswamy[6] developed a Navier–Stokes code, ENSAERO, for aeroelastic simulations.

It is clear that time-accurate solutions with the coupled Euler/Navier–Stokes equations and the structural dynamic equations provide a powerful tool for simulating aerodynamic flutter phenomena. However, the computational time needed for such calculations still prevents such methods from being used for routine analysis and design purposes. In addition, the reliability and accuracy of such methods have not yet been well established.

An effective method to shorten the elapsed time for such computations is to use parallel processing. Alonso and Jameson[7] developed a two-dimensional parallel code for wing flutter calculations that couples the solution of the Euler equations with the solution of the structural equations. Byun and Guruswamy[8] also developed a parallel version of the ENSAERO code, which was recently used and extended by Goodwin et al.[9]

We present a new integrated CFD–CSD simulation code for flutter calculations based on a parallel, multiblock, multigrid flow solver for the Euler/Navier–Stokes equations. Structural modal dynamic equations are solved simultaneously in a strongly coupled fashion with the flow equations by a fully implicit time-marching method. A dual-time-stepping algorithm is used to achieve time accuracy and allow simultaneous integration of the flow and structural equations without any time delay. A novel moving mesh method is developed to dynamically move/deform the computational grid at each time step. A spline matrix method is used to provide the interpolation between the CFD and CSD grids. The complete integrated CFD–CSD computational method is implemented on a parallel computer by use of the message passing interface (MPI) standard. The code has been shown to offer good parallel speed up on clusters of personal computers networked with commodity 100-MB/s ethernet.

This computer program is capable of calculating conventional harmonic or indicial responses of an aeroelastic system, as well as performing direct CFD–CSD simulations. Computations are performed for a two-dimensional airfoil model and a well-established three-dimensional test case, the AGARD 445.6 wing,[10,11] to validate and establish the usefulness of the code. In the following section, we will describe each component of the integrated system. We will then present and analyze the computational results for the two-dimensional aeroelastic wing model and the three-dimensional AGARD 445.6 wing.

Received 25 April 2000; revision received 16 November 2000; accepted for publication 17 November 2000. Copyright © 2001 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Associate Professor, Department of Mechanical and Aerospace Engineering. Senior Member AIAA.

†Visiting Research Specialist, Department of Mechanical and Aerospace Engineering; currently Associate Professor, Department of Aerospace Engineering, Northwestern Polytechnical University, Xi'an, 710072 People's Republic of China.

‡Graduate Student, Department of Mechanical and Aerospace Engineering. Member AIAA.

§Senior Member of the Technical Staff, Dynamics Division, 20 Science Park Drive; currently Adjunct Associate Professor, Temasek Laboratories, National University of Singapore, Singapore 119260, Republic of Singapore. Member AIAA.

¶Member of Technical Staff, Dynamics Division, 20 Science Park Drive.

II. Unsteady Navier–Stokes Solver

Tsai et al.[12] developed a multiblock, multigrid Euler/Navier–Stokes code called NSAERO for steady flow calculations based on Jameson's finite volume and Runge–Kutta time-marching method.[13,14] The multiblock algorithm uses two layers of halo cells to exchange flow information across a common interface between two blocks. Multigrid is implemented in a horizontal mode as termed by Yadlin and Caughey,[15] in which the block loop is nested inside the loop for the multigrid cycle so that information is exchanged among blocks, and the flowfield is updated on all blocks within each time step at each multigrid level. Information is not exchanged among blocks for the internal multistages of the explicit Runge–Kutta time stepping within each time step. This code has been validated and extensively used for steady flow calculations of wing, wing–body combination, S ducts, and complete aircraft.[12]

In this research, the code is extended to perform time-accurate calculations. Time accuracy is achieved by using the dual-time-method proposed by Jameson.[16] The basic implementation of the dual-time-step method is the same as in Ref. 17. After being discretized in space by a finite volume method, the time-dependent Navier–Stokes equations can be written in the following semidiscrete form:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} + \boldsymbol{R}(\boldsymbol{w}) = 0 \tag{1}$$

where $\boldsymbol{w}$ is the vector of flow variables at each mesh point and $\boldsymbol{R}$ is the vector of the residuals, consisting of the spatially discretized flux balance of the Navier–Stokes equations. A second-order accurate fully implicit scheme is then used to integrate the preceding equation in time,

$$\frac{3\boldsymbol{w}^{n+1} - 4\boldsymbol{w}^{n} + \boldsymbol{w}^{n-1}}{2\Delta t} + \boldsymbol{R}(\boldsymbol{w}^{n+1}) = 0 \tag{2}$$

This implicit scheme is A stable. We can reformulate Eq.(2) into the following:

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t^*} + \boldsymbol{R}^*(\boldsymbol{w}) = 0 \tag{3}$$

where

$$\boldsymbol{R}^*(\boldsymbol{w}) = (3/2\Delta t)\boldsymbol{w} + \boldsymbol{R}(\boldsymbol{w}) - (2/\Delta t)\boldsymbol{w}^{n} + (1/2\Delta t)\boldsymbol{w}^{n-1} \tag{4}$$

where t^* is a pseudotime. The solution of the implicit equation (2) is now made equivalent to the steady-state solution of Eq. (3) with the pseudotime t^*. We can then apply all of the acceleration techniques, including the multigrid method, that are already implemented in the steady NSAERO in solving Eq. (3). Once the solution to Eq. (3) converges in pseudotime, we achieve the time-accurate solution to Eq. (2) for one time step. As was demonstrated in Ref. 17, the Courant–Friedrichs–Lewy (CFL) numbers of more than 4000 could be used for solving the unsteady equations with this technique. Each time step needs about 40 multigrid pseudotime cycles for the Euler equations to yield a reduction in residual of 3–4 decades and about 80 cycles for the Navier–Stokes equations. Note that in an unsteady calculation, there are time instances when the flow changes little from earlier time steps. When this happens, the initial residual is already low because the flowfield from the preceding time already provided a very good initial condition for the new time step. Further reduction of the residuals by 3 or 4 orders of magnitude is neither practical nor necessary. Consequently, a conservative 60 pseudotime steps in each real time step are enforced for the Euler calculations, and 100 pseudotime steps are enforced for the Navier–Stokes calculations, to ensure adequate convergence.

Time-accurate calculations for three-dimensional problems are still very time consuming with even the best algorithms available at the current time. To keep the computational time within realistic limits for large unsteady calculations, the code is also parallelized with domain decomposition and MPI to take advantage of parallel computers or networked clusters of workstations. The flowfield is partitioned into multiple blocks, which are distributed over a number of processors available on a parallel computer or networked workstations. The existing multiblock structure in the steady NSAERO code[12] provides the basis for the parallel implementation. The dual-time implicit–explicit solver is performed on each processor for the individual blocks assigned to that processor. Again, two layers of halo cells are used beyond the boundaries of each block to facilitate the implementation of boundary conditions and the communication between processors. Connectivity information of the blocks and processors are stored in preprocessed pointer arrays, and MPI is used to perform the communication between blocks that are on different CPUs.

The accuracy of the unsteady flow solver using the preceding time-marching algorithm was validated and documented in Refs. 18 and 19. The flow over a pitching NACA 64A010 airfoil is used as a test case. Because the new unsteady parallel NSAERO code is fully three dimensional, we also tested the airfoil case by laying the pitching airfoil in the three different coordinate planes. Comparison of the computed results showed no difference between the calculations with different coordinate orientations, and the results agree with the experimental data found in Ref. 20.

III. Multiblock Moving Mesh Algorithm

A novel moving grid algorithm, AIM3D (Ref. 19), which remeshes the moving configuration adaptively in each block of grids, is also implemented in a parallel fashion and combined with the flow solver to handle flow problems with arbitrary motion of domain boundaries. The moving grid algorithm within each block is based on the method of arc-length-based transfinite interpolation, which is performed independently on local processors where the blocks reside. A spring network approach is used to determine the motion of the corner points of the blocks, which may be connected in an unstructured fashion in a general multiblock method. A smoothing operator is applied to the points of the block face boundaries and edges to maintain grid smoothness and grid angles. The details of this moving grid method is described in Ref. 19.

IV. CSD Model and Its Solution

Modal equations are used to calculate the structural deformation under an aerodynamic forcing. For each mode i, the modal dynamic equation is written in the following form:

$$\ddot{q}_i + 2\zeta_i\omega_i\dot{q}_i + \omega_i^2 q_i = Q_i \tag{5}$$

where q_i is the generalized normal mode displacement, ζ_i is the modal damping, ω_i is the modal frequency, and Q_i is the generalized aerodynamic force. The structural displacement vector can be written as a summation of N modal shapes extracted from a full finite element analysis of the structure:

$$\{u_s\} = \sum_{i=1}^{N} q_i\{h_i\} \tag{6}$$

where $\{h_i\}$ are the modal shapes.

Equation (5) is converted into a first-order system of equations for each i and integrated in time by a second-order fully implicit scheme. Following Alonso and Jameson,[7] we assume

$$x_{1i} = q_i, \qquad \dot{x}_{1i} = x_{2i} \tag{7}$$

for each of the modal equations. We can rewrite Eq. (5) in matrix form as

$$\{\dot{X}_i\} = [A_i]X_i + \{F_i\}, \qquad i = 1, N \tag{8}$$

where

$$\{X_i\} = \begin{Bmatrix} x_{1i} \\ x_{2i} \end{Bmatrix}, \qquad [A] = \begin{bmatrix} 0 & 1 \\ -\omega_i^2 & -2\omega_i\zeta_i \end{bmatrix}, \qquad \{F_i\} = \begin{Bmatrix} 0 \\ Q_i \end{Bmatrix}$$

Equation (8) can be decoupled to be

$$\frac{\mathrm{d}z_{1i}}{\mathrm{d}t} = \omega_i\left(-\zeta_i + \sqrt{\zeta_i^2 - 1}\right)z_{1i} + \frac{\left(-\zeta_i + \sqrt{\zeta_i^2 - 1}\right)}{2\sqrt{\zeta_i^2 - 1}}Q_i \tag{9}$$

$$\frac{\mathrm{d}z_{2i}}{\mathrm{d}t} = \omega_i\left(-\zeta_i - \sqrt{\zeta_i^2 - 1}\right)z_{2i} + \frac{\left(\zeta_i + \sqrt{\zeta_i^2 - 1}\right)}{2\sqrt{\zeta_i^2 - 1}}Q_i \tag{10}$$

where

$$\{Z_i\} = \begin{Bmatrix} z_{1i} \\ z_{2i} \end{Bmatrix} = P^{-1}\{X_i\}$$

and P_i is the diagonalization matrix:

$$P_i = \begin{bmatrix} \left(-\zeta_i - \sqrt{\zeta_i^2 - 1}\right)/\omega_i & \left(-\zeta_i + \sqrt{\zeta_i^2 - 1}\right)/\omega_i \\ 1 & 1 \end{bmatrix}$$

We use the same second-order-accurate fully implicit scheme as Eq. (2) to integrate the preceding equations in time:

$$\frac{3z_{1i}^{n+1} - 4z_{1i}^{n} + z_{1i}^{n-1}}{2\Delta t} = -R_{1i}\left(z_{1i}^{n+1}, z_{2i}^{n+1}, Q_i^{n+1}\right)$$

$$= \omega_i\left(-\zeta_i + \sqrt{\zeta_i^2 - 1}\right)z_{1i}^{n+1} + \frac{\left(-\zeta_i + \sqrt{\zeta_i^2 - 1}\right)}{2\sqrt{\zeta_i^2 - 1}}Q_i^{n+1} \tag{11}$$

$$\frac{3z_{2i}^{n+1} - 4z_{2i}^{n} + z_{2i}^{n-1}}{2\Delta t} = -R_{2i}\left(z_{1i}^{n+1}, z_{2i}^{n+1}, Q_i^{n+1}\right)$$

$$= \omega_i\left(-\zeta_i - \sqrt{\zeta_i^2 - 1}\right)z_{2i}^{n+1} + \frac{\left(\zeta_i + \sqrt{\zeta_i^2 - 1}\right)}{2\sqrt{\zeta_i^2 - 1}}Q_i^{n+1} \tag{12}$$

The variables z_{1i}, z_{2i}, and Q_i in the preceding equations are coupled through the flow equations. The deformation of the wing, represented by z_{1i} and z_{2i}, influences the flowfield and, thus, the aerodynamic force Q_i. Conversely, the aerodynamic force Q_i determines the deformation of the wing. Therefore, the preceding equations for the time marching of the structural equations must be solved simultaneously with Eq. (2) for the Navier–Stokes equations.

It is very convenient to reformulate Eqs. (11) and (12) into a pseudotime format identical to Eqs. (2) and (4), that is,

$$\frac{\mathrm{d}z_{1i}}{\mathrm{d}t^*} + R_{1i}^*(z_{1i}, Q_i) = 0 \tag{13}$$

$$\frac{\mathrm{d}z_{2i}}{\mathrm{d}t^*} + R_{2i}^*(z_{2i}, Q_i) = 0 \tag{14}$$

where

$$R_{1i}^*(z_{1i}, Q_i) = (3/2\Delta t)z_{1i} + R_{1i}\left(z_{1i}, z_{2i}, Q_i\right) - (2/\Delta t)z_{1i}^{n} + (1/2\Delta t)z_{1i}^{n-1} \tag{15}$$

$$R_{2i}^*(z_{1i}, Q_i) = (3/2\Delta t)z_{2i} + R_{2i}\left(z_{1i}, z_{2i}, Q_i\right) - (2/\Delta t)z_{2i}^{n} + (1/2\Delta t)z_{2i}^{n-1} \tag{16}$$

Equations (3), (15), and (16) can be regarded as one single system of time-dependent equations in the pseudotime t^*, which can be solved by existing efficient explicit time-marching methods until a steady state is reached. Once the computation reaches a steady state in the pseudotime t^*, the solutions to Eqs. (3), (13), and (14) become the time-accurate solution of the implicit fully coupled CFD–CSD equations (2), (11), and (12) in one physical time step without any time lag between the CFD and CSD equations.

In the current implementation, the same five-stage Runge–Kutta time-stepping scheme for the Navier–Stokes equations is used for the CSD equations. Computations show that the coupling of the CFD and CSD equations do not affect, either adversely or favorably, the maximum allowable CFL number in the pseudotime stepping. It is found, however, that the CSD pseudotime equations converge faster than their counterparts in the flow equations. Therefore, it is more efficient and, more robust to march the flow equations with more pseudotime steps than for the CSD equations. Note that the flow and structural equations are still stongly coupled at the same physical time level without any staggering. The final time accuracy in the physical time is preserved as long as a steady state is reached in the pseudotime.

V. Interfacing Between the CSD and CFD Grids

The proceeding sections lay out the mathematical formation of the coupling of the CFD and CSD equations. However, a missing link remains in actual computation, that is, the relation between the CFD computational grid and the CSD computational grid. Although Bendiksen and Hwang[21] developed a finite element algorithm for both the flow equations and the structural equations so that the same surface grid can be used, most CFD codes use different algorithms and different computational grids for the flow and the structures. Consequently, interpolation of computational grids and aerodynamic loads must be performed between the two systems. A suitable method that handles this need in a general fashion is still in the making. The current engineering practice, as exemplified in MSC/NASTRAN[22] and used in the current paper, uses a combination of methods. The infinite plate spline by Harder and Desmarais[23] and thin plate spline by Duchon[24] is more suitable for winglike components where the structure is usually modeled by plate and shell elements. The beam spline method by Done[25] is used for a bodylike component as the structure is modeled by beam elements. The structural grid of a complex structure that uses a combination of different elements can be related to the aerodynamic grid by using different spline methods. Details of the approach are to be found in Ref. 22. Once a spline method is applied, the displacement vector defined on the structural grid $\{u_s\}$ can be related to the displacement vector on the aerodynamic grid $\{u_a\}$ via a spline matrix $[G]$,

$$\{u_a\} = [G]\{u_s\} \tag{17}$$

Once the structure equations are solved, the displacements on the structural grid $\{u_s\}$ is transformed to the displacement on the aerodynamic grid $\{u_a\}$ by the preceding equation, from which the deforming grid code AIM3D discussed in Sec. III can then be used to regenerate the volume grid in the flowfield for CFD computations. Once the flow equations are solved, the aerodynamic loading on the CFD grid, $\{F_a\}$, can then be transformed to the aerodynamic loading on the structural grid, $\{F_s\}$, via the following equation, based on the principle of virtual work:

$$\{F_s\} = [G]^T\{u_a\} \tag{18}$$

where $[G]^T$ is the transpose of $[G]$. Equation (18) guarantees the conservation of energy between the flow and the structural systems because it is derived by the principle of virtual work.

The size of the vector $\{u_a\}$ is three times m, where m is the number of aerodynamic grid points on the wing surface. The size of the structural displacement vector $\{u_s\}$ is three times n, where n is the number of finite element grid point on the wing surface. The size of the $[G]$ matrix is then $(3m) \times (3n)$. The $[G]$ matrix is pregenerated and stored in the code.

VI. Integrated Aerostructure System

Figure 1 shows the integration strategy for our aeroelastic simulation system. At the start of the program, the specifications of the structures are input into NASTRAN, which extracts the modal shapes and parameters needed by the CSD solver in the coupled simulation code. Inputs are also given to the flow solver NSAERO and its preprocessor PREPNS. The flow solver will first compute the static loading on the structure. The aerodynamic forcing $\{F_a\}$ is then transformed to the form suitable for the CSD modal equations using the SPLINE matrix G. The CSD solver then computes the structural displacement $\{u_s\}$, which is subsequently transformed to the grid points for the flow solver. With the new displacement $\{u_a\}$, the dynamic moving mesh program REMESH (AIM3D) is implemented to regenerate the mesh for the flow solver. The flow solver and the CSD solver are coupled within each time step through the preceding process, as shown by the arrows in Fig. 1. Because both the flow equations and the structural equations are solved by a pseudotime iteration algorithm, the aforementioned coupling between the CFD and CSD solver are easily absorbed in the pseudotime

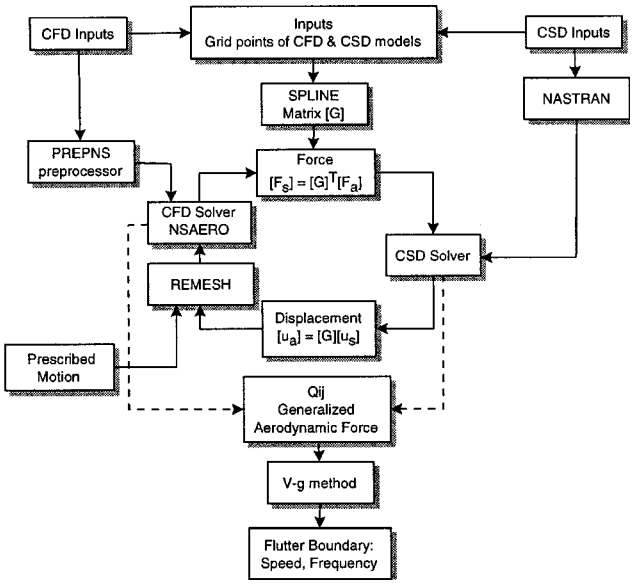

Fig. 1 Integrated CFD–CSD method for flutter calculations.

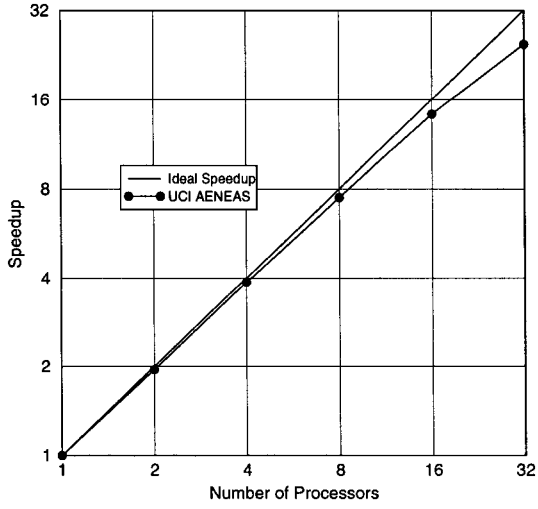

Fig. 2 Parallel speedup for unsteady calculations on a 32-node personal computer cluster with 100-MB switched ethernet, 71,001 grid points.

iteration process until a fully converged solution is achieved at the new time level as discussed in Sec. IV.

Figure 1 also shows a decoupled approach for flutter prediction, marked by the dashed lines. In such an approach, prescribed structural motion is provided to the flow solver without being coupled with the CSD model. The generalized aerodynamic force Q_{ij} on the frequency domain is extracted from the CFD calculations by use of either the harmonic method or the indicial response method. Q_{ij} will then be fed into the conventional V–g method[26] to obtain flutter boundaries of the aeroelastic system. This method provides a fast tool for flutter predictions, but it is restricted to linear systems. The fully coupled method provides a compliment to the linearized approach.

In a typical run of the coupled option in the Euler mode, the grid generation part takes less than 1% of the total CPU time, which indicates that the moving grid method is very efficient. Significantly more time is needed to perform the spline interpolation by using the G matrix method to exchange surface deformation and aerodynamic forcing data between the flow solver and the structural solver. The time needed for the interpolation and grid generation altogether amounts to 7% of the total CPU time.

The present parallel code runs efficiently on clustered personal computers. Figure 2 shows the parallel speedup for the pitching NACA 64A012 airfoil case running on the AENEAS computer consisting of 32 300-MHz Pentium II processors connected by a 100-MB switched ethernet network. It can be seen that the parallel efficiency remains high for the maximum available processors for this case, with 71,001 total grid points. A speedup factor of 25 is achieved with 32 processors. A three-dimensional Euler run with 32 processors and 176,601 grid points needs 12 h elapsed time on the aforementioned Pentium cluster and has a speedup factor of 24 compared to a single CPU run of the same case. The three-dimensional speedup is essentially the same as in the two-dimensional case. Although the flow solver with the structures coupled together has been implemented and tested for Navier–Stokes calculations, only Euler results are presented in the following section due to the long computational time needed for Navier–Stokes calculations.

VII. Computational Results

A. Two-Dimensional Wing Model

As a first test case, we apply the coupled CFD–CSD method to the two-dimensional Isogai wing model,[27,28] case A. This model simulates the bending and torsional motion of a wing cross section in the outboard portion of a swept wing. It consists of two DOF, plunging and pitching, for a NACA 64A010 airfoil. We compute this case with the Euler equations and compare the results given by Alonso and Jameson.[7] The details of the structural model can be found in Ref. 7 as well as in Refs. 27 and 28.

Figures 3–7 show the flutter computational results for the Isogai wing model[27,28] at a flight Mach number of 0.825. Plotted in Figs. 3–7 are the time history of the pitching and plunging amplitude computed by the integrated CFD–CSD code with the fully

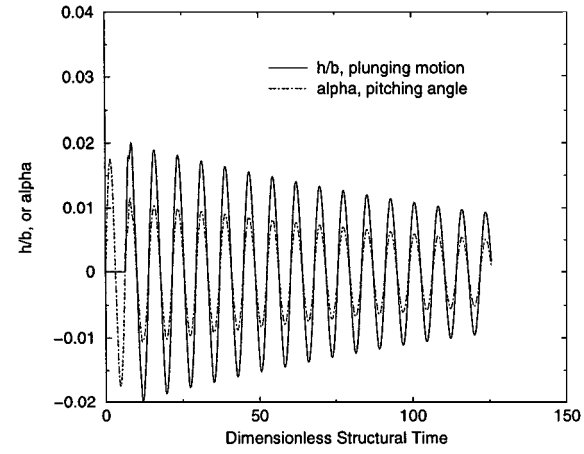

Fig. 3 Time history of pitching and plunging motion for the Isogai wing model[27,28] for $M_\infty = 0.825$ and $V_f = 0.530$.

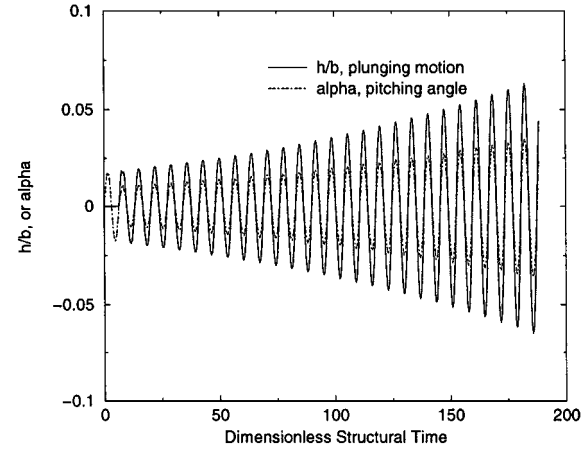

Fig. 4 Time history of pitching and plunging motion for the Isogai wing model[27,28] for $M_\infty = 0.825$ and $V_f = 0.725$.

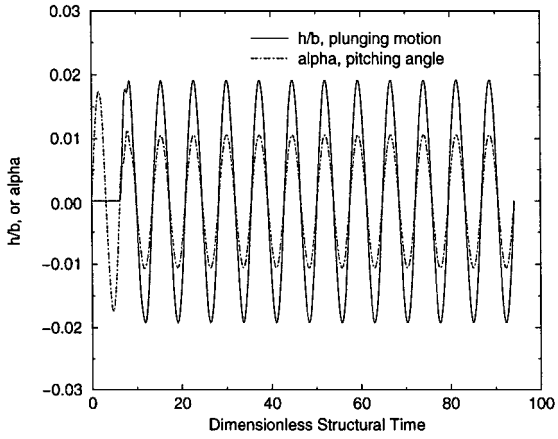

Fig. 5 Time history of pitching and plunging motion for the Isogai wing model[27,28] for $M_\infty = 0.825$ and $V_f = 0.630$.

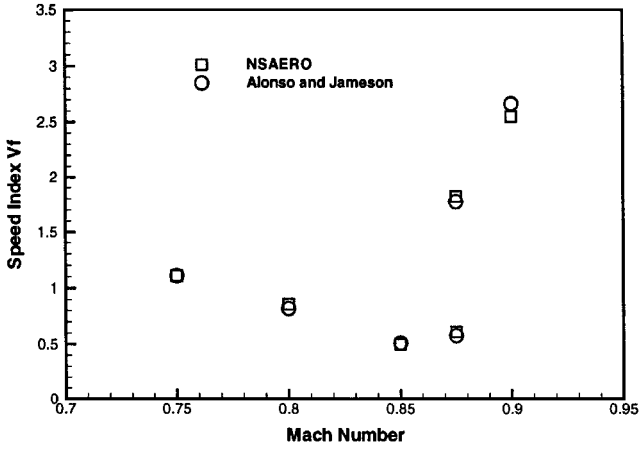

Fig. 6 Flutter boundary predicted by NSAERO compared to that by Alonso and Jameson.[7]

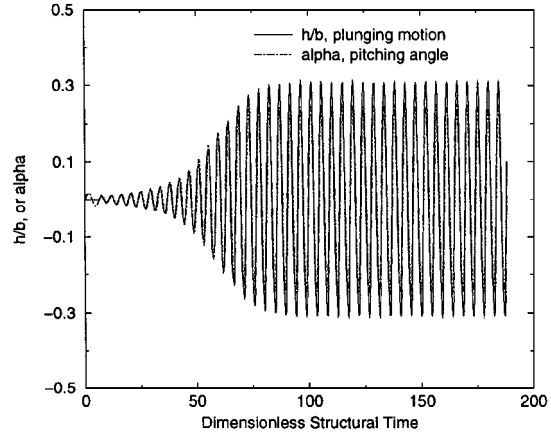

Fig. 7 Time history of pitching and plunging motion for the Isogai wing model[27,28] for $M_\infty = 0.75$ and $V_f = 1.33$.

coupled CFD and CSD approach. Figure 3 is a case with a low-speed index $V_f = 0.530$. V_f is defined as

$$V_f = U_\infty / b\omega\sqrt{\mu}$$

where U_∞ is the freestream velocity, b is the half-chord, ω is the structural frequency, and μ is the mass ratio. For this low V_f, Fig. 3 shows that both the pitching and plunging amplitude decay with time, indicating that the aeroelastic system is stable for this particular condition. At a higher V_f, the system may become less and less stable until one or both of the pitching and plunging motions diverge as shown in Fig. 4, when $V_f = 0.725$. In between these two V_f conditions, there is a particular point where the system is neutrally stable. This is shown in Fig. 5, when $V_f = 0.630$.

A converging point and a diverging point like those just mentioned are first identified, and from them we can interpolate the V_f in between to obtain an estimate of the neutral point. We then perform a computation with the new V_f to see if it is above or below the stability limit, or perhaps right at the neutral point. It may take several runs for a given freestream Mach number before the V_f corresponding to the neutral stability point can be accurately located by this bisection method. However, most of the runs do not need computations with many time periods because we can easily identify whether the system is oscillating with a diverging or converging amplitude by looking at only the first few periods. With this method, computations for a number of freestream Mach numbers for the Isogai wing model[27,28] are performed. The flutter boundary predicted by our code is plotted and compared with that obtained by Alonso and Jameson[7] in Fig. 6. Clearly, the results given by the two codes agree very closely for this case.

Figure 7 shows a situation where the system initially diverges but then reaches a steady oscillatory mode with finite amplitude, the LCO. The LCO for this case was first discovered by Kousen and Bendiksen[3] and was also shown by Alonso and Jameson.[7] This test case demonstrates the capability of the integrated CFD–CSD program to predict LCO, although the computation may have to be performed for a long time before LCO can be identified in the time integration.

B. AGARD 445.6 Wing

Although the results in the preceding section are calculated by using the full three-dimensional code with eight cells in the spanwise direction, the flow is only two dimensional. In this section, the integrated aeroelastic system is used to predict the flutter boundary for the three-dimensional AGARD 445.6 wing[10,11] by both the indicial method and the coupled CFD–CSD method. This wing is a semispan model made of the NACA 65A004 airfoil that has a quarter-chord sweep angle of 45 deg, a panel aspect ratio of 1.65, and a taper ratio of 0.66. We consider the weakened wing model as listed in Ref. 10. It was tested in the Transonic Dynamics Tunnel at NASA Langley Research Center. This is a well-defined test case proposed as an AGARD standard aeroelastic configuration for flutter calculations.[11] The wing is modeled structurally by the first four natural vibrational modes shown in Fig. 8 as taken from Ref. 11. Those are identified as the first bending, first torsion, second bending, and second torsion modes, respectively, by a finite element analysis. The natural frequencies of these modes are also shown in Fig. 8. A grid of 176,601 is used for this case. With 32 networked Pentium II 300 MHz personal computers, one run of three periods in the coupled mode takes 12 h for the AGARD 445.6 wing with 64 time steps per period.

In a harmonic method to calculate the generalize aerodynamic force Q_{ij} for the ith vibrational mode, the wing is assumed to undergo a sinusoidal motion of the ith mode at a given reduced frequency κ. The unsteady flow induced by this sinusoidal motion is then calculated, from which the generalized aerodynamic force on each of the jth modes, $j = 1, 4$, can be deduced.

To use a classical V-g method[26] to determine the flutter boundary, Q_{ij} must be calculated over a range of frequencies of interest. Consequently, computation of the flowfield must be performed for a number of reduced frequencies and for each vibrational mode of importance in a harmonic method. This demands a large amount of computational time. An alternative is to use the indicial method originally proposed by Tobak,[29] and also by Ballhaus and Goorjian,[1] in which a step function input excitation is fed into the aerodynamic system for each structural mode. The response of the aerodynamics system is called the indicial response. A Fourier analysis on this indicial response is enough to deduce the system response $Q(\kappa)$ for the complete range of the reduced frequency κ. In this way, only one time integration of the Euler/Navier–Stokes equations is needed for each mode of the structural system to obtain the complete generalized aerodynamic force (GAF) matrix $Q_{ij}(\kappa)$. Seidel et al.[30]

introduced an impulse function method to avoid the discontinous nature of the step function used by Ballhaus and Goorjian.[1] The impulse method is used in this work.

Figure 9 shows the comparison of the generalized aerodynamic forces Q_{11} and Q_{21} calculated by the harmonic method and the impulse method for the 445.6 wing at a freestream Mach number $M_\infty = 0.901$. Note that except near the high reduced frequency end, the indicial method based on the impulse input agrees well with the harmonic method. The indicial method, however, significantly reduces the computational time.

Figure 10 shows the computed generalized aerodynamic forces Q_{11}, Q_{12}, Q_{21}, and Q_{22}, respectively, by the indicial method for the first two vibrational modes. The first four modes are calculated, although only two of them are shown here. Those Q_{ij} are then used in a classical V–g method code to determine the flutter boundary of the wing.

In the indical method approach, as many runs as there are structural modes must be performed to determine the generalized aerodynamic forces. Typically, for a wing, four modes are needed. Each run must be performed with adequate time resolution to accurately calculate the aerodynamic response for high frequencies. In addition, from three to five periods of the lowest frequency mode are also needed for the disturbances due to the initial impulse to sufficiently decay.

Unlike in an indicial or harmonic method, where a prescribed motion of the wing is used, the motion of the wing is not predetermined in a direct CFD–CSD coupled simulation. It is computed based entirely on the flow and structural dynamic equations and their interaction. The wing is started with either an initial displacement and zero initial vibrational velocity or zero displacement but nonzero vibrational velocity in each of the four modes. A steady-state solution is first obtained for a given initial position of the wing, and then the wing is let go. The wing will then start oscillating in time resulting in either damped, diverging, or neutral vibrations.

Figures 11–15 show the time histories of the generalize coordinates q_i of the first four modes in the coupled CFD–CSD unsteady computations for some different flight Mach numbers and different speed indices V_f. Although the time history may be different when different initial conditions are used, it was found that the general behavior of the wing is the same as far as stability is concerned. If a mode is unstable (diverging in time) for the zero velocity but nonzero displacement initial condition, it will also be unstable for the zero displacement but nonzero velocity initial condition. In principle, it is not impossible that stability depends on initial conditions for a nonlinear system. However, no special effort is made to identify this in the current work.

Consider the supersonic flight condition $M_\infty = 1.141$ tested in the experiment.[10] Figure 11 shows the time history of the four generalized coordinates for the 445.6 wing for a velocity index $V_f = 0.58$. For this case, the flow is started with zero initial displacement but a nonzero initial velocity in all of the four modes of the wing. Clearly, Fig. 11 shows that this is a damped case. The amplitudes of all modes decrease in time. Therefore, the system is stable for this condition. At a little higher speed index, $V_f = 0.652$, the system exhibits high vibrational amplitude and slower damping, as shown in Fig. 12. However, it is still a damped case, although it is getting close to the neutral condition. At an even higher speed index, $V_f = 0.70$, as shown in Fig. 13, we can see that the amplitudes of the first three modes grow very fast, indicating that we have passed the neutral stability point. At this point, the flutter boundary for this flight Mach number can be roughly estimated to be in between $V_f = 0.652$ and 0.700. This process can be iterated to obtain better estimates of the flutter boundary. However, for this case, a mean value from $V_f = 0.652$ and 0.700 will suffice given the large difference between the experimental data and the computed data to be discussed later.

The flutter mode can be determined by looking at Figs. 11–13 and identifying which mode first starts to be unstable as we increase V_f. Once this mode is identified, a Fourier analysis of this mode can be performed in its last few periods of vibration, near the neutral point, to identify its frequency. A quick estimate can be obtained by simply measuring the time between the two zero points in the end of the vibration. Determining the flutter mode and flutter frequency may get somewhat ambiguous in some cases. Note, however, from Figs. 11–13, that the vibrational frequencies of the four modes

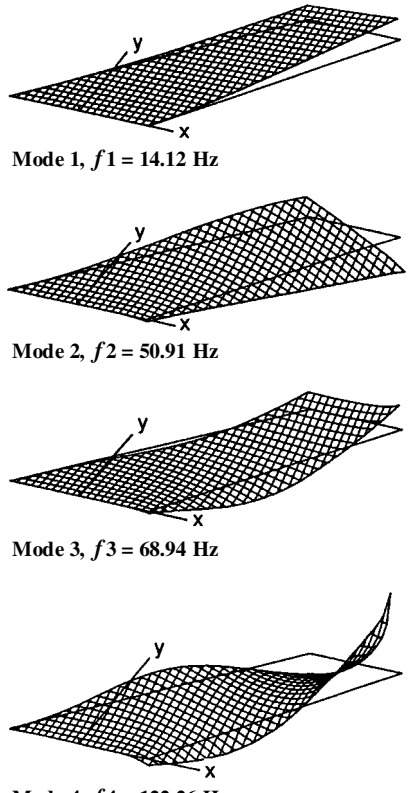

Fig. 8 Modal deflections for the 445.6 wing from Ref. 11.

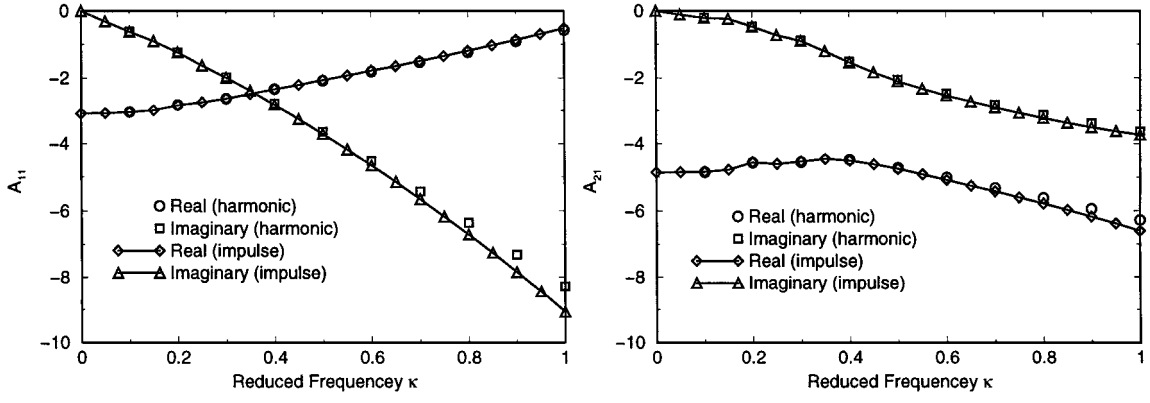

Fig. 9 Comparison of generalized aerodynamic forces calculated by the harmonic and indicial methods Q_{11} and Q_{21} for $M_\infty = 0.901$.

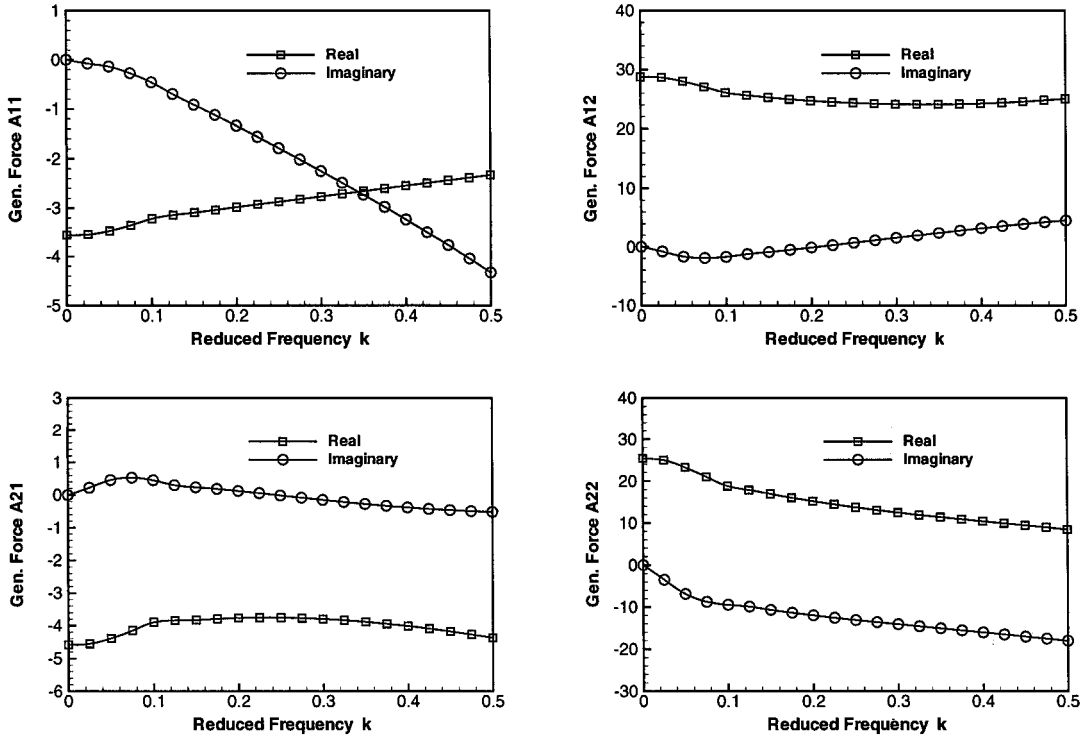

Fig. 10 Generalized aerodynamic force Q_{11}, Q_{12}, Q_{21}, and Q_{22} for $M_\infty = 0.960$.

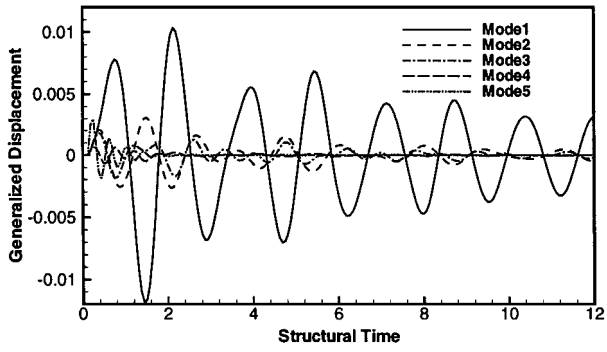

Fig. 11 Time history of the generalized coordinates for the AGARD 445.6 wing for $M_\infty = 1.141$ and $V_f = 0.58$.

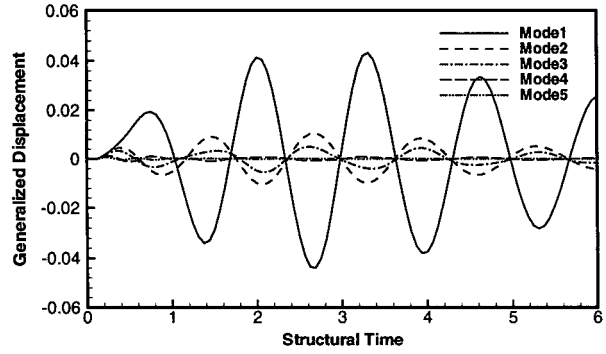

Fig. 12 Time history of the generalized coordinates for the AGARD 445.6 wing for $M_\infty = 1.141$ and $V_f = 0.652$.

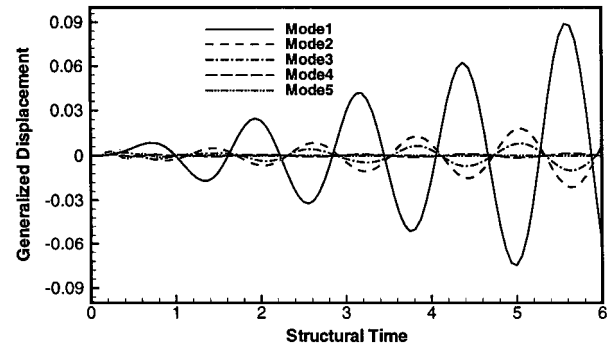

Fig. 13 Time history of the generalized coordinates for the AGARD 445.6 wing for $M_\infty = 1.141$ and V_f =0.70.

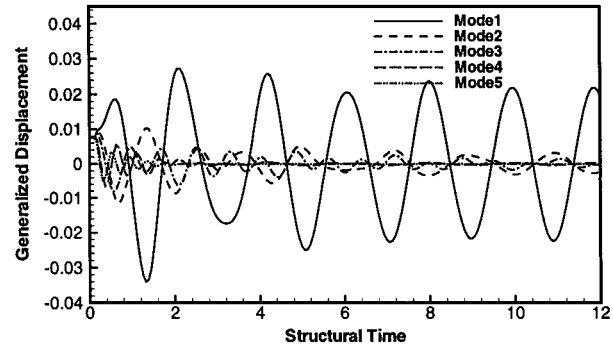

Fig. 14 Time history of the generalized coordinates for the AGARD 445.6 wing for $M_\infty = 1.072$ and $V_f = 0.46$.

initially are rather different and closer to their natural frequencies for the low V_f case. As the speed index increases, the low-frequency modes shift to higher frequencies, whereas the high modes tend to shift toward the low-frequency end. Near the neutral point, the first two modes tend to oscillate at the same frequency.

Figure 14 shows the calculation for the condition $M_\infty = 1.072$ and $V_f = 0.46$. Clearly, this shows that the V_f is at or very close to the flutter boundary and that the first or the second mode is the flutter mode. These modes are almost neutral and oscillate at almost equal frequency, whereas the higher modes damp out in time.

Figure 15 shows a similar calculation for the subsonic condition $M_\infty = 0.901$ and $V_f = 0.34$. Again, this shows that the chosen V_f is at or very close to the flutter boundary and that the first mode is the flutter mode because all other modes are being damped. Note that the frequency of the higher modes remains high in this case.

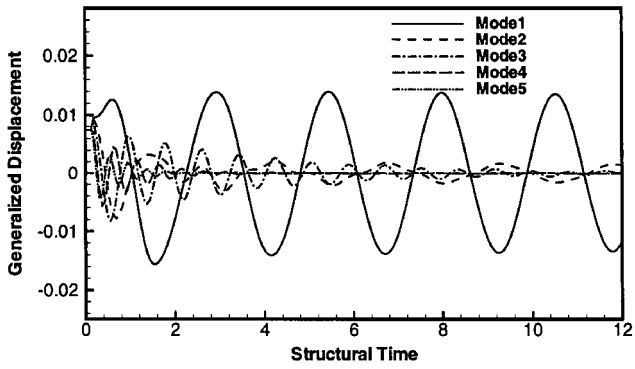

Fig. 15 Time history of the generalized coordinates for the AGARD 445.6 wing for $M_\infty = 0.901$ and $V_f = 0.34$.

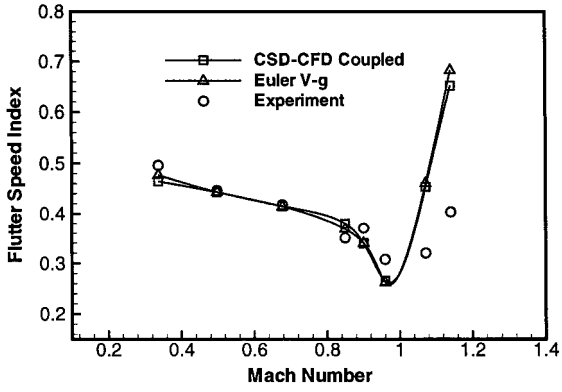

Fig. 16 Flutter speed for the AGARD 445.6 wing.

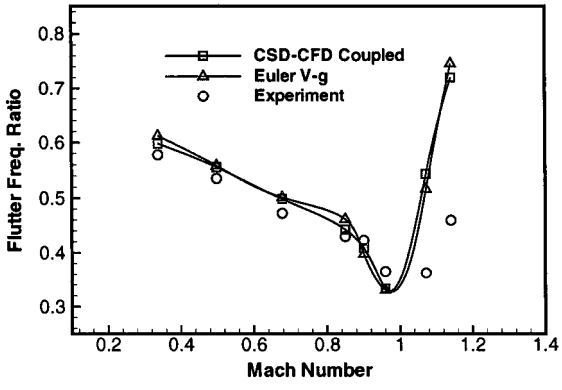

Fig. 17 Flutter frequency for the AGARD 445.6 wing.

Using the aforementioned approach, we determined the flutter boundary and frequency of the weakened wing over the flight Mach number range $M_\infty = 0.338$–1.141 studied experimentally in Ref. 10. The results are shown in Figs. 16 and 17, together with the experimental data and those calculated by the V–g methods based on the calculated indicial responses. Note that the results computed by the coupled CFD–CSD method agree very well with the experimental data in the subsonic and transonic range.

In the supersonic range, however, both the coupled method and the indicial method yield much higher flutter velocities than those determined in the experiment. Also note that Ref. 4 also overpredicted the V_f for supersonic flight conditions by almost an equal amount compared to our result. Although Refs. 5 and 9 showed some improvement by performing full Navier–Stokes calculations, the amount of improvement was small compared to the still large differences between their computational results and the experiment. It is not clear why this is the case. However, subtle differences between the subsonic and supersonic cases can be observed from the time histories shown in Figs. 11–15. In the subsonic case shown in Fig. 15, the damped modes show monotonically decreasing amplitudes similar to those found for the two-dimensional airfoil case shown in Fig. 3. For the supersonic cases, however, the first mode shows a clear growth before it starts to decay (Figs. 11 and 12) or remains neutral (Fig. 14). This behavior is found to be independent of the initial conditions used in the calculations. It might be conjectured that this nonlinear behavior can be responsible for the early detection of flutter in the model wing tested in Ref. 10. Further investigation is necessary.

It has been suggested that LCO may be indicated in the V–g analysis if it exists. The coupled CFD–CSD approach may then be used to confirm the existence of LCO. However, no LCO phenomenon for the 445.6 wing has been found in this research or by any other researchers.

VIII. Summary and Conclusions

A parallel integrated CFD–CSD simulation program has been developed for the simulation and prediction of flutter of an aeroelastic system. This program consists of a three-dimensional, parallel, multiblock, multigrid, unsteady Navier–Stokes solver, a parallel dynamic grid deformation code, a CSD solver strongly coupled with the flow solver using dual time stepping, and a spline matrix method for interfacing the CFD and CSD grids and aerodynamic loading variables. This program has two options. Option 1 allows it to obtain the harmonic or indicial responses of an aeroelastic system by prescribing the motion of the structure. Option 2 uses the coupled CFD–CSD method to simulate directly the unsteady behavior of an aeroelastic system in the time domain. The program is efficient and scales well on parallel computers. Coupled three-dimensional unsteady CFD–CSD simulations can be performed overnight on a cluster of 32 networked ordinary Pentium personal computers. The code has been used to calculate transonic flutter of a two-dimensional airfoil and the three-dimensional AGARD 445.6 wing by both options. Some useful conclusions are summarized here:

1) As many runs as there are structural modes must be performed in the indical approach to determine the generalized aerodynamic forces for each flight condition. Three to five time periods of calculation are needed in each run for sufficient accuracy of the indicial response.

2) The coupled CFD–CSD option provides the complete time history of an aerostructure system given an initial disturbance of the structures. Signal analysis of the calculated modal displacements is performed to determine the flutter mode, speed, and frequency. Stability can be readily noted within two to three periods of calculations for most situations for each run. Three to five runs are usually adequate to give a good estimate of the neutral stability point for each flight condition. The stability boundary thus determined does not depend on the initial conditions. The coupled CFD–CSD method can estimate the flutter point more quickly than the indicial method if one has a rough idea of where flutter might occur.

3) Flutter speed and flutter frequency predictions of the AGARD 445.6 wing by the coupled approach agree very well with experimental data for subsonic and transonic speeds. The transonic dip phenomena is well captured. For subsonic cases, the coupled approach predicts either monotonically decaying modes or monotonically growing modes for velocity index values below or above the numerically determined neutral point.

4) Both the indicial method and the coupled method overpredict the flutter speed and flutter frequency compared to experimental data for the AGARD 445.6 wing at supersonic speeds. The coupled approach, however, predicts consistent initial growths of the structural motions before they decay for the supersonic cases when the velocity index value is above the experimental neutral point and below the numerically determined neutral point. Monotonic growth is observed when the velocity index is above the numerically determined neutral point.

5) The coupled CFD–CSD option may be used to predict LCO, although long computational time is needed for this purpose. This is demonstrated in the two-dimensional airfoil case, but no LCO mode was found for the three-dimensional AGARD 445.6 wing case.

Acknowledgment

Computations of this work has been performed on the AENEAS parallel computer system at the University of California, Irvine.

References

[1]Ballhaus, W. F., and Goorjian, P. M., "Computation of Unsteady Transonic Flows by the Indical Method," *AIAA Journal*, Vol. 16, No. 2, 1978, pp. 117–124.

[2]Bendiksen, O. O., and Kousen, K. A., "Transonic Flutter Analysis Using the Euler Equation," AIAA Paper 87-0911, April 1987.

[3]Kousen, K. A., and Bendiksen, O. O., "Limit Cycle Phenomena in Computational Transonic Aeroelasticity," *Journal of Aircraft*, Vol. 31, No. 6, 1994, pp. 1257–1263.

[4]Lee-Rausch, E. M., and Batina, J. T., "Wing Flutter Boundary Prediction Using Unsteady Euler Aerodynamic Method," *Journal of Aircraft*, Vol. 32, No. 2, 1995, pp. 416–422.

[5]Lee-Rausch, E. M., and Batina, J. T., "Wing Flutter Computations Using an Aerodynamic Model Based on the Navier–Stokes Equations," *Journal of Aircraft*, Vol. 33, No. 6, 1996, pp. 1139–1147.

[6]Guruswamy, G. P., "Navier–Stokes Computations on Swept-Tapered Wings, Including Flexibility," AIAA Paper 90-1152, April 1990.

[7]Alonso, J. J., and Jameson, A. , "Fully Implicit Time-Marching Aeroelastic Solutions," AIAA Paper 94-0056, Jan. 1994.

[8]Byun, C., and Guruswamy, G. P., "Aeroelastic Computations on Wing–Body–Control Configurations on Parallel Computers," AIAA Paper 96-1389, April 1996.

[9]Goodwin, S. A., Weed, R. A., Sankar, L. N., and Raj, P., "Toward Cost-Effective Aeroelastic Analysis on Advanced Parallel Computing Systems," *Journal of Aircraft*, Vol. 36, No. 4, 1999, pp. 710–715.

[10]Yates, E. C., Jr., Land, N. S., and Foughner, J. T., Jr., "Measured and Calculated Subsonic and Transonic Flutter Characteristics of a 45° Sweptback Wing Planform in Air and in Freon-12 in the Langley Transonic Dynamics Tunnel," NASA TN D-1616, March 1963.

[11]Yates, E. C., Jr., "AGARD Standard Aeroelastic Configurations for Dynamic Response I—Wing 445.6," NASA TM 100492, Aug. 1987.

[12]Tsai, H. M., Dong, B., and Lee, K. M., "Development and Validation of a Three-Dimensional Multiblock, Multigrid Flow Solver for External and Internal Flows," AIAA Paper 96-0171, Jan. 1996.

[13]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[14]Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," Rept. 1743, Dept. of Mechanical and Aerospace Engineering, Princeton Univ., Princeton, NJ, Oct. 1985.

[15]Yadlin, Y., and Caughey, D. A., "Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow," AIAA Paper 90-0106, Jan. 1990.

[16]Jameson, A., "Time-Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA Paper 91-1596, June 1991.

[17]Liu, F., and Ji, S., "Unsteady Flow Calculations with a Multigrid Navier-Stokes Method," *AIAA Journal*, Vol. 34, No. 10, 1996, pp. 2047–20**XX**.

[18]Zhu, Y., "Time-Accurate Calculations of Unsteady Flows with Aeroelastic Applications," M.S. Thesis, Dept. of Mechanical and Aerospace Engineering, Univ. of California, Irvine, Irvine, CA, Dec. 1998.

[19]Wong, A., Tsai, H., Cai, J., Zhu, Y., and Liu, F., "Unsteady Flow Calculations with a Multiblock Moving Mesh Algorithm," AIAA Paper 2000-1002, Jan. 2000.

[20]Davis, S. S., "NACA64A010 Oscillatory Pitching," *Compendium of Unsteady Aerodynamics Measurements*, AGARD-R-702, 1982.

[21]Bendiksen, O. O., and Hwang, G., "Nonlinear Flutter Calculations for Transonic Wings," *Proceedings of the International Forum on Aeroelasticity and Structure Dynamics*, 1997, Confederal of Europe Aerospace Societies Rome, pp. 105–114.

[22]Rodden, W. P., and Johnson, E. H., *MSC/NASTRAN Aeroelastic Analysis User's Guide*, Ver. 68, Macneal–Schwendler Corp., Los Angels, 1994.

[23]Harder, R. L., and Desmarais, R. N., "Interpolation Using Surface Splines," *Journal of Aircraft*, Vol. 9, No. 2, 1972, pp. 189–191.

[24]Duchon, J. P., "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces," edited by W. Schempp and K. Zeller, *Constructive Theory of Functions of Several Variables*, Springer-Verlag, Berlin, 1977, pp. 85–100.

[25]Done, G. T. S., "Interpolation of Mode Shapes: A Matrix Scheme Using Two-Way Spline Curves," *Aeronautical Quarterly*, Vol. 16, Nov. 1965, pp. 333–349.

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

[27]Isogai, K., "Transonic-Dip Mechanism of Flutter of a Sweptback Wing," *AIAA Journal*, Vol. 17, No. 7, 1979, pp. 793–795.

[28]Isogai, K., "Transonic-Dip Mechanism of Flutter of a Sweptback Wing: Part II," *AIAA Journal*, Vol. 19, No. 7, 1981, pp. 1240–1242.

[29]Tobak, M., "On the Use of Indicial Function Concept in the Analysis of Unsteady Motions of Wings and Wing–Tail Combinations," NACA Rept. 1188, 1954.

[30]Seidel, D. A., and Bennet, R. M., and Ricketts, R. H., "Some Recent Applications of XTRAN3S," AIAA Paper 83-1811, July 1983.