JOURNAL OF AIRCRAFT
Vol. 42, No. 3, May–June 2005

Direct Aeroelastic Bifurcation Analysis of a Symmetric Wing Based on Euler Equations

K. J. Badcock,* M. A. Woodgate,† and B. E. Richards‡
University of Glasgow, Glasgow, Scotland G12 8QQ, United Kingdom

The application of a sparse matrix solver for the direct calculation of Hopf bifurcation points for the flexible AGARD 445.6 wing in a transonic flow modeled using computational fluid dynamics is considered. The iteration scheme for solving the Hopf equations is based on a modified Newton's method. Direct solution of the linear system for the updates has previously been restrictive for application of the method, and the sparse solver overcomes this limitation. Previous work has demonstrated the scheme for aerofoil calculations. The current paper gives the first three-dimensional results with the method, showing that an entire flutter boundary for the AGARD 445.6 wing can be traced out in a time comparable to that required for a small number of time-marching calculations, yielding two orders of magnitude improvement when compared to the time-marching approach.

Introduction

TIME-DOMAIN analysis is the main solution method used in computational aeroelasticity when the flow is modeled by the Euler and Reynolds-averaged Navier–Stokes (RANS) equations. However, the need to execute searches over multiparameter space to identify stability behavior can lead to high computational cost because of the need to do an unsteady coupled computational-fluid-dynamics (CFD) and computational-structural-dynamics (CSD) calculation for each combination of parameters. This cost is not prohibitive when the intention is to examine behavior at previously identified problem conditions, and there are several recent impressive demonstrations of this kind for complete aircraft configurations (e.g., see Farhat et al.[1] and Melville[2]).

A way of reducing the cost of parametric searches for stability behavior was proposed by Morton and Beran[3] from the U.S. Air Force Laboratories. Their method uses dynamical systems theory to characterize the nature of the aeroelastic instability, with this additional information concentrating the use of the CFD. In this way the problem of locating a one-parameter Hopf bifurcation was reduced from multiple time-marching calculations to a single steady-state calculation of a modified system. This modified system calculates the value of the parameter for which an eigenvalue of the system Jacobian matrix crosses the imaginary axis. A convection-diffusion problem was used to evaluate the approach in Beran and Carlson,[4] and the method was then applied to an aeroelastic system consisting of an aerofoil moving in pitch and plunge in Morton and Beran.[3] The linear system was solved using a direct method, and this motivated the use of an approximate Jacobian matrix to reduce the cost. Some robustness problems were encountered when applying the method, particularly at transonic Mach numbers. A complex variable formulation of the problem was introduced in Beran,[5] which resolved some of these problems. An approach considered to reduce the difficulties of applying a direct solver to large linear systems was to use domain decomposition to reduce the size of the system at the expense of an outer iteration over the domains. This was tested on the model problem in references in Beran and Carlson[4] and Beran.[5]

The problems introduced by using a direct solver were resolved in Badcock et al.,[6] where a sparse matrix formulation was used to make feasible the solution of the linear system for much larger grids. The Newton iteration was modified to enhance the efficiency of the scheme following work on approximate Jacobian matrices for CFD only problems reported in Badcock et al.[7] The method was shown to be effective for tracing out flutter boundaries for symmetric aerofoils moving in pitch and plunge, with reductions of two orders of magnitude in the computational time required when compared with time marching.

The current paper extends the method to calculate flutter boundaries for symmetric wings. The additional issues to be considered are the treatment of a moving grid around a deforming geometry (as opposed to rigid motions for the previous aerofoil cases), the use of a modal structural model (instead of the pitch-plunge equations), and the resulting requirement to pass information between nonmatching grids, and the larger problem size, and especially the impact of this on the solution of the linear system. The formulation is considered in the following two sections and then results are presented for the AGARD 445.6 wing test case of Yates[8] to demonstrate the feasibility of the method for three-dimensional problems.

Aerodynamic and Structural Simulations

Aerodynamics

The three-dimensional Euler equations can be written in conservative form and Cartesian coordinates as

$$\frac{\partial \boldsymbol{w}_f}{\partial t} + \frac{\partial \boldsymbol{F}^i}{\partial x} + \frac{\partial \boldsymbol{G}^i}{\partial y} + \frac{\partial \boldsymbol{H}^i}{\partial z} = 0 \tag{1}$$

where $\boldsymbol{w}_f = (\rho, \rho u, \rho v, \rho w, \rho E)^T$ denotes the vector of conserved variables. The flux vectors $\boldsymbol{F}^i$, $\boldsymbol{G}^i$ and $\boldsymbol{H}^i$ are,

$$\boldsymbol{F}^i = \begin{pmatrix} \rho U^* \\ \rho u U^* + p \\ \rho v U^* \\ \rho w U^* \\ U^*(\rho E + p) + \dot{x} \end{pmatrix} \tag{2}$$

$$\boldsymbol{G}^i = \begin{pmatrix} \rho V^* \\ \rho u V^* \\ \rho v V^* + p \\ \rho w V^* \\ V^*(\rho E + p) + \dot{y} \end{pmatrix}, \qquad \boldsymbol{H}^i = \begin{pmatrix} \rho W^* \\ \rho u W^* \\ \rho v W^* + p \\ \rho w W^* + p \\ W^*(\rho E + p) + \dot{z} \end{pmatrix} \tag{3}$$

Received 19 September 2003; revision received 20 February 2004; accepted for publication 27 February 2004.

*Reader, Computational Fluid Dynamics Laboratory, Department of Aerospace Engineering.

†Research Assistant, Computational Fluid Dynamics Laboratory, Department of Aerospace Engineering.

‡Mechan Professor, Computational Fluid Dynamics Laboratory, Department of Aerospace Engineering. Associate Fellow AIAA.

In the preceding equations ρ, u, v, w, p, and E denote the density, the three Cartesian components of the velocity, the pressure, and the specific total energy, respectively, and U^*, V^*, W^* the three Cartesian components of the velocity relative to the moving coordinate system, which has local velocity components $\dot{x}$, $\dot{y}$, and $\dot{z}$, that is,

$$U^* = u - \dot{x} \tag{4}$$

$$V^* = v - \dot{y} \tag{5}$$

$$W^* = w - \dot{z} \tag{6}$$

The variables here have been nondimensionalized with respect to the wing root chord c for x, y, and z; the freestream velocity U_∞ for u, v, and w; the freestream density ρ_∞ for ρ, U_∞/c for t and $\rho_\infty U_\infty^2$ for p.

The time-marching results in the current work are obtained using the Glasgow University code PMB (which stands for parallel multiblock). A summary of some applications examined using the code can be found in Badcock et al.[7] A fully implicit steady solution of the Euler equations is obtained by advancing the solution forward in time by solving the discrete nonlinear system of equations

$$\frac{\boldsymbol{w}_f^{n+1} - \boldsymbol{w}_f^n}{\Delta t} = \boldsymbol{R}_f\left(\boldsymbol{w}_f^{n+1}\right) \tag{7}$$

The term on the right-hand side, called the residual, the discretization of the convective terms, given here by the approximate Riemann solver of Osher and Chakravarthy,[9] MUSCL interpolation of Van Leer,[10] and Van Albada's limiter. The sign of the definition of the residual is opposite to convention in CFD, but this is to provide a system of ordinary differential equations, which follows the convention of dynamical systems theory, as will be discussed in the next section. Equation (7) is a nonlinear system of algebraic equations, which is solved by an implicit method described in Cantariti et al.,[11] the main features of which are an approximate linearization to reduce the size and condition number of the linear system and the use of a preconditioned Krylov subspace method to calculate the updates.

The steady-state solver is applied to unsteady problems within a pseudo-time-stepping iteration as formulated by Jameson.[12]

Structural Dynamics, Intergrid Transformation, and Mesh Movement

The wing deflections $\delta\boldsymbol{x}_s$ are defined at a set of points $\boldsymbol{x}_s$ by

$$\delta\boldsymbol{x}_s = \Sigma\alpha_i\phi_i \tag{8}$$

where ϕ_i are the mode shapes calculated from a full finite element model of the structure and α_i are the generalized coordinates. By projecting the finite element equations onto the mode shapes and assuming that the mode shapes have been scaled to give dimensional generalized masses $m_i = 1$, the scalar equations

$$\frac{\mathrm{d}^2\alpha_i}{\mathrm{d}t^2} + D_i\frac{\mathrm{d}\alpha_i}{\mathrm{d}t} + \omega_i^2\alpha_i = \mu\phi_i^T\boldsymbol{f}_s \tag{9}$$

are obtained where $\boldsymbol{f}_s$ is the vector of aerodynamic forces at the structural grid points and D_i is the coefficient of structural damping. Here a nondimensionalization consistent with the flow solver has been used and the factor $\mu = \rho_\infty/\rho_w$ is a density ratio, where ρ_w is the density of the wing structure. These equations are rewritten as a system in the form

$$\frac{\mathrm{d}\boldsymbol{w}_s}{\mathrm{d}t} = \boldsymbol{R}_s \tag{10}$$

where $\boldsymbol{w}_s = (\ldots, \alpha_i, \dot{\alpha}_i, \ldots)^T$ and $\boldsymbol{R}_s = (\ldots, \dot{\alpha}_i, \mu\phi_i^T\boldsymbol{f}_s - \omega_i^2\alpha_i - D_i\dot{\alpha}_i, \ldots)^T$. This equation can be solved by a two-stage Runge–Kutta method, which requires a knowledge of $\boldsymbol{f}_s^n$ and $\boldsymbol{f}_s^{n+1}$. To avoid introducing sequencing errors by approximating the value of $\boldsymbol{f}_s^{n+1}$, the Runge–Kutta solution is iterated in pseudotime along with the CFD solver, with the latest pseudoiterate being used to give a value for $\boldsymbol{f}_s^{n+1}$. At convergence the fluid and structural solvers are properly sequenced, at very little extra computational cost beyond what is required for the aerodynamic solution.

The aerodynamic forces are calculated at face centers on the aerodynamic surface grid. The problem of communicating these forces to the structural grid is complicated in the common situation that these grids not only do not match, but are also not even defined on the same surface. This problem, and the influence it can have on the aeroelastic response, was considered in Goura[13] and Goura et al.,[14] where a method was developed, called the constant-volume-tetrahedron (CVT) transformation. This method uses a combination of projection of fluid points onto the structural grid, transformation of the projected point, and recovery of the out-of-plane component to obtain a cheap but effective relation between deformations on the structural grid and those on the fluid grid. Denoting the fluid grid locations and aerodynamic forces as $\boldsymbol{x}_a$ and $\boldsymbol{f}_a$, then

$$\delta\boldsymbol{x}_a = \mathcal{S}(\boldsymbol{x}_a, \boldsymbol{x}_s, \delta\boldsymbol{x}_s)$$

where $\mathcal{S}$ denotes the relationship defined by CVT. In practice this equation is linearized to give

$$\delta\boldsymbol{x}_a = S(\boldsymbol{x}_a, \boldsymbol{x}_s)\delta\boldsymbol{x}_s$$

and then by the principle of virtual work, $\boldsymbol{f}_s = S^T\boldsymbol{f}_a$.

The grid speeds on the wing surface are also needed, and these are approximated directly from the linearized transformation as

$$\delta\dot{\boldsymbol{x}}_a = S(\boldsymbol{x}_a, \boldsymbol{x}_s)\delta\dot{\boldsymbol{x}}_s$$

where the structural grid speeds are given by

$$\delta\dot{\boldsymbol{x}}_s = \Sigma\dot{\alpha}_i\phi_i. \tag{11}$$

The geometries of interest deform during the motion. This means that, unlike the rigid aerofoil problem, the aerodynamic mesh must be deformed rather than rigidly translated and rotated. This is achieved using transfinite interpolation of displacements (TFI) within the blocks containing the wing. More elaborate treatments that move blocks to maintain grid orthogonality are possible (see Tsai et al.[15]) but are not necessary here because only small wing deflections are encountered and the blocks in the mesh can be extended well away from the wing. The wing surface deflections are interpolated to the volume grid points $\boldsymbol{x}_{ijk}$ as

$$\delta\boldsymbol{x}_{ijk} = \psi_j^0\delta\boldsymbol{x}_{a,ik} \tag{12}$$

where ψ_j^0 are values of a blending function (see Gordon and Hall[16]), which varies between one at the wing surface (here $j = 1$) and zero at the block face opposite. The surface deflections $\boldsymbol{x}_{a,ik}$ are obtained from the transformation of the deflections on the structural grid and so ultimately depend on the values of α_i. The grid speeds can be obtained by differentiating Eq. (12) to obtain their explicit dependence on the values of $\dot{\alpha}_i$.

Formulation of Hopf Analysis

The semidiscrete form of the coupled CFD–CSD system is

$$\frac{\mathrm{d}\boldsymbol{w}}{\mathrm{d}t} = \boldsymbol{R}(\boldsymbol{w}, \mu) \tag{13}$$

where

$$\boldsymbol{w} = [\boldsymbol{w}_f, \boldsymbol{w}_s]^T \tag{14}$$

is a vector containing the fluid unknowns $\boldsymbol{w}_f$ and the structural unknowns $\boldsymbol{w}_s$ and

$$\boldsymbol{R} = [\boldsymbol{R}_f, \boldsymbol{R}_s]^T \tag{15}$$

is a vector containing the fluid residual $\boldsymbol{R}_f$ and the structural residual $\boldsymbol{R}_s$. The residual also depends on a parameter μ, which is independent of $\boldsymbol{w}$. An equilibrium of this system $\boldsymbol{w}_0(\mu)$ satisfies $\boldsymbol{R}(\boldsymbol{w}_0, \mu) = \boldsymbol{0}$.

Dynamical systems theory gives criteria for an equilibrium to be stable (e.g., see Seydel[17]). In particular, all eigenvalues of the Jacobian matrix of Eq. (13), given by $A = \partial \boldsymbol{R}/\partial \boldsymbol{w}$, must have negative real part. A Hopf bifurcation with respect to the parameter μ occurs in the stability of the equilibrium at values of μ such that $A(\boldsymbol{w}_0, \mu)$ has one eigenvalue $i\omega$ that crosses the imaginary axis. Denoting the corresponding eigenvector by $\boldsymbol{P} = \boldsymbol{P}_1 + i\boldsymbol{P}_2$, a critical value of μ is one at which there is an eigenpair ω and $\boldsymbol{P}$ such that

$$A\boldsymbol{P} = i\omega\boldsymbol{P}. \tag{16}$$

This equation can be written in terms of real and imaginary parts as $A\boldsymbol{P}_1 + \omega\boldsymbol{P}_2 = \boldsymbol{0}$ and $A\boldsymbol{P}_2 - \omega\boldsymbol{P}_1 = \boldsymbol{0}$. A unique eigenvector is chosen by scaling against a constant real vector $\boldsymbol{q}$ to produce a fixed complex value, taken to be $0 + 1i$. This yields two additional scalar equations $\boldsymbol{q}^T\boldsymbol{P}_1 = 0$ and $\boldsymbol{q}^T\boldsymbol{P}_2 - 1 = 0$.

A bifurcation point can be calculated directly by solving the system of equations

$$\boldsymbol{R}_A(\boldsymbol{w}_A) = \boldsymbol{0} \tag{17}$$

where

$$\boldsymbol{R}_A = \begin{bmatrix} \boldsymbol{R} \\ A\boldsymbol{P}_1 + \omega\boldsymbol{P}_2 \\ A\boldsymbol{P}_2 - \omega\boldsymbol{P}_1 \\ \boldsymbol{q}^T\boldsymbol{P}_1 \\ \boldsymbol{q}^T\boldsymbol{P}_2 - 1 \end{bmatrix} \tag{18}$$

and $\boldsymbol{w}_A = [\boldsymbol{w}, \boldsymbol{P}_1, \boldsymbol{P}_2, \mu, \omega]^T$. If there are n components in $\boldsymbol{w}$, then $\boldsymbol{w}_A$ has $3n + 2$ components, as does $\boldsymbol{R}_A$, and hence Eq. (17) is closed. The catch is that this is a large sparse system of nonlinear equations.

Newton's method can be used to solve this type of problem. A sequence of approximations $\boldsymbol{w}_A^n$ to a solution is generated by solving the linear system

$$\frac{\partial \boldsymbol{R}_A}{\partial \boldsymbol{w}_A}\Delta\boldsymbol{w}_A = -\boldsymbol{R}_A^n \tag{19}$$

where $\Delta\boldsymbol{w}_A = \boldsymbol{w}_A^{n+1} - \boldsymbol{w}_A^n$. The Jacobian matrix on the left-hand side of Eq. (19) is given in expanded form as

$$\frac{\partial \boldsymbol{R}_A}{\partial \boldsymbol{w}_A} = \begin{bmatrix} A & 0 & 0 & \boldsymbol{R}_\mu & 0 \\ (A\boldsymbol{P}_1)_{\boldsymbol{w}} & A & I\omega & (A\boldsymbol{P}_1)_\mu & \boldsymbol{P}_2 \\ (A\boldsymbol{P}_2)_{\boldsymbol{w}} & -I\omega & A & (A\boldsymbol{P}_2)_\mu & -\boldsymbol{P}_1 \\ 0 & \boldsymbol{q}^T & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{q}^T & 0 & 0 \end{bmatrix} \tag{20}$$

There are three key issues for the application of Eq. (19). First, a good initial guess is required, or the iterations will diverge. Secondly, the Jacobian matrix $\partial \boldsymbol{R}_A/\partial \boldsymbol{w}_A$ is required. Thirdly, the large sparse linear system given in Eq. (19) must be solved. These points were considered for the aerofoil problem in Badcock et al.[6] For the three-dimensional problem with a modal structural model, the Jacobian calculation is the aspect which is different from the aerofoil case. This is therefore considered in the next section. The details of the first and third points are as described earlier for the aerofoil case in Badcock et al.[6] and are only summarized here.

First, for the application of the scheme it is assumed that a good estimate of the flutter point and frequency is available from some other source, for example, linear theory, at the first Mach number of interest. The inverse power method is then applied, again using the sparse matrix formulation, to calculate the eigenvector corresponding to the critical eigenvalue when the fluid Jacobian from the first-order spatial scheme is used. This information is then used as the starting solution for the Hopf calculation at the first Mach number, which is initially converged using the first-order fluid Jacobian before switching to the Jacobian of the full second-order spatial scheme. For subsequent Mach numbers, the converged solution from the previous one provides an adequate starting solution. In this way the flutter boundary is traced for a range of Mach numbers.

The linear system at each Newton step is solved using the sparse matrix package Aztec (see Tuminaro et al.[18]). Although not optimized for the current problem, the generality of the package has allowed various experiments to be carried out. This package has three main solvers available, namely, GMRES, CGS, and TFQMR, although the differences in performance for the current problem were found to be small. The key issue for iterative linear solvers is usually the preconditioner. The incomplete LU factorization family as described in Axelsson[19] can be very effective at approximating the inverse of the coefficient matrix with a small number of terms. For CFD calculations, block incomplete upper lower factorizations with no fill in have proved very successful, as illustrated in Badcock et al.[7]

One simplification arises if we are dealing with a symmetric problem, in which case the equilibrium solution $\boldsymbol{w}_0$ is independent of μ and hence can be calculated from Eq. (13) independently of the other Hopf conditions in Eq. (17). Then, a smaller system can be solved for the bifurcation parameter. This algorithm has been implemented in a code, which will be referred to as the BIFOR code. The formulation of the coupled CFD–CSD problem follows that of the PMB code already given.

Calculation of the Jacobian Matrix

The difficult terms to form in the Jacobian matrix of the augmented system are A and A_μ. The calculation of A is most conveniently done by partitioning the matrix as

$$A = \begin{bmatrix} \dfrac{\partial \boldsymbol{R}_f}{\partial \boldsymbol{w}_f} & \dfrac{\partial \boldsymbol{R}_f}{\partial \boldsymbol{w}_s} \\ \dfrac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{w}_f} & \dfrac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{w}_s} \end{bmatrix} = \begin{bmatrix} A_{\mathrm{ff}} & A_{\mathrm{fs}} \\ A_{\mathrm{sf}} & A_{\mathrm{ss}} \end{bmatrix} \tag{21}$$

The block A_{ff} describes the influence of the fluid unknowns on the fluid residual and has by far the largest number of nonzeros when a modal structural model is used. The treatment of this term is crucial to the efficiency of the scheme and is discussed in Badcock et al.[6] To drive the Newton iteration to convergence, the analytical Jacobian corresponding to the first-order spatial scheme is used. This approach has proved successful for CFD-alone calculations shown in Badcock et al.[7]

The only consideration when choosing the Newton iteration matrix is that the scheme converges to the correct answer, the latter being determined by the calculation of the residual on the right-hand side of the Newton iteration. Hence, the products $A\boldsymbol{P}_1$ and $A\boldsymbol{P}_2$ must be computed exactly. This can be done using a matrix-free formulation of the form

$$A\boldsymbol{x} \approx \frac{\boldsymbol{R}(\boldsymbol{w} + h\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{w} - h\boldsymbol{x})}{2h} \tag{22}$$

where $\boldsymbol{x}$ denotes the real or imaginary part of the critical eigenvalue and h is the increment applied. Computing this expression is not costly as it requires only two residual evaluations. This gives an accurate approximation to the required product without having to evaluate and store the exact A. Using the matrix-free evaluation of the augmented residual reduces the memory requirements for the scheme overall and simplifies the code considerably.

The dependence of the fluid residual on the structural unknowns α_i and $\dot{\alpha}_i$ is partially hidden by the notation used. The fluid residual depends not only on the fluid cell values but also on the location of the grid points themselves and the cell volumes. The fluid and structural unknowns are independent variables, and hence to calculate the term A_{fs} the fluid unknowns are kept fixed. The influence of the structural unknowns is felt through the moving grid. Using the modal structural model, the updated grid locations and speeds are calculated by moving the structural grid according to the values of the generalized coordinates and velocities, transferring these to the fluid surface grid using the transformation and then applying TFI to transfer these boundary values to the volume grid. By using second-order finite differences, the terms in A_{fs} can be calculated in $2n_s$ evaluations of the aerodynamic residual if there are n_s structural unknowns.

The term A_{sf} involves calculating the dependence of the generalized fluid forces on the fluid unknowns. The surface forces on the aerodynamic grid are calculated and then transferred to the structural grid using the transformation. The inner product is then formed using the forces on the structural grid and the modal coefficients. The Jacobian matrix for the forces on the structural grid with respect to the fluid unknowns can be calculated first analytically. Then the required terms for A_{sf} can be calculated through matrix-vector multiplication.

Finally, the exact Jacobian matrix for the dependence of the structural equations on the structural unknowns is easy to calculate analytically. However, it is important to remember that the generalized force will change with the structural unknowns because the surface normals to the wing will change as the wing moves. A finite difference calculation is used to calculate this effect.

The bifurcation parameter (μ in this case) only appears in the structural equations. Therefore, for this case,

$$A_\mu = \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial^2 \boldsymbol{R}_s}{\partial \mu \partial \boldsymbol{w}_s} \end{bmatrix} \tag{23}$$

Because of the simple algebraic expression for $\partial \boldsymbol{R}_s / \partial \boldsymbol{w}_s$, it is straightforward to calculate the required term analytically.

Results for Symmetric Problem

Test Case

An important problem with the development of aeroelastic simulation tools is the lack of experimental data available for assessment. The experiments are intrinsically destructive and require careful model construction to ensure proper scaling, and hence the expense is much higher than rigid model tests. A complete set of measurements is available for the AGARD 445.6 wing, and results have been included in most papers on computational aeroelasticity, giving a wide range of data with which to evaluate the current method. However, a disadvantage of this test case is that it does not feature significant nonlinear effects because the wing is thin. Despite this, it is commonly the first test case used to test time-marching codes and is suitable for the current work because it is symmetric. Previous time-marching results are reviewed by Goura et al.[20]

The AGARD 445.6 wing is made of mahogany and has a 45-deg quarter-chord sweep, a root chord of 22.96 in. (0.583 m), and a constant NACA64A004 symmetric profile (see Yates[8]). A series of flutter tests, which were carried out at the NASA Langley Transonic Dynamics Tunnel to determine stability characteristics, was reported in 1963. Various wing models were tested (and broken). The case for which most published results have appeared is the weakened wing (wing 3) in air. This wing had holes drilled out, which were filled with plastic to maintain the aerodynamic shape while being structurally weaker. Published experimental data include the dynamic conditions at which the wing was viewed to be unstable for Mach numbers in the range 0.338 to 1.141. The structural characteristics of the wing were provided in the form of measured natural frequencies and mode shapes derived from a finite element model. Full details of the structural model used are given in Goura.[13] Four modes are retained with the first two bending modes having frequencies 9.7 and 50.3 and the two torsional modes at 36.9 and 90.0.

A problem with the published results for the AGARD 445.6 wing is that most are of a demonstration nature in the sense that verification is hardly ever shown. There is a significant spread of the results obtained when using solutions of the Euler equations. The results that cluster around the measured data in the region of the flutter dip tend to be on coarser grids, with finer CFD grids generally giving lower flutter speeds. The only published attempt at a systematic grid refinement study was shown by Melville and Gordnier.[21] In this commendable study the fine- and medium-grid results were further apart than those on the medium and coarse grids, and hence grid independence was not achieved, casting doubt on other published results on coarser grids. The main obstacle to a rigorous study is of course the cost of the calculations. A second question mark against

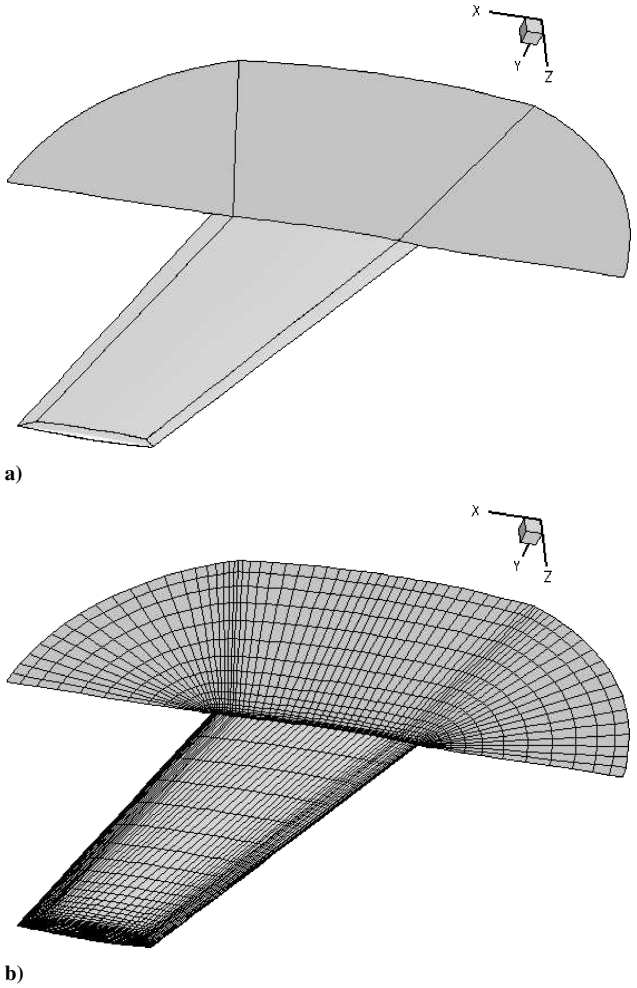

Fig. 1 Only the inner blocks above the wing are shown on the symmetry plane: a) grid topology and b) medium surface mesh.

the published results is that in the majority of cases no structural damping was used. In the description of the experiment, a value of 2% is suggested by Yates,[8] although it is not clear how certain this value is.

Time-Marching Solutions

An attempt was made to do a detailed grid-convergence study within the limits of the computers available. All calculations reported in this section were done with the PMB code. To optimize the grid used, two requirements were identified. First, because the calculations are inviscid, and hence no wake needs to be preserved, an O grid was used, which helps to maximize the number of grid points on the surface of the wing. A genuine multiblock topology was used to allow a good quality mesh to be preserved in the tip leading- and trailing-edge regions as shown in Fig. 1. Second, the important quantities that must be well predicted for the flutter calculations are the generalized modal forces. The pressure difference between the upper and lower surface therefore must be predicted accurately. The flutter response is dominated by the first bending mode, which features some twist near the tip but is essentially a plunging motion near the root. The significant pressure difference, and following from this the main contribution to the generalized force, comes from the region towards the wing tip. The grid points were therefore concentrated in this region. Most structured grids shown in the literature have been of the C-H topology and are reasonably uniform in the spanwise direction.

The finest grid in the current work has 1.43 million points and 17,700 on the wing surface. Medium and coarse grids that have 190,000 and 27,000 points, respectively, with 4453 and 1131 points on the wing surface were extracted from this. The number of volume

points used in the refinement study of Gordnier and Melville,[21] where the fine, medium and coarse grids have 2.1 million, 901,000, and 274,000 points, respectively, are comparable to the current grids, but significantly the number of surface points are less (9231, 5343, and 2366, respectively). It is stressed however that the topology in the current study would not be ideal for RANS calculations, which were the main focus of Gordnier and Melville.[21] Comparison is also made with the structured[22] and unstructured[23] studies by Batina and co-workers. The structured grid has 517,000 points with 5289 on the wing surface. The unstructured grid has 129,000 tetrahedra, and, although no information is given about the number of points on the wing surface, the pictures shown in the paper suggest that the grid points are more strongly clustered in the wing region than is possible for structured grids.

A number of tests using the medium grid were made on the temporal parameters (time step and convergence level) at a Mach number of 0.96, a freestream velocity of 308 m/s, a density of 0.08 kg/m^3, and structural damping of 0.5%. First, the responses when using 10 or 20 pseudosteps per real time step were identical, indicating that 10 steps was more than adequate. Secondly, using a reduced time step of 0.01 and 0.02 also gave an identical response, and hence the larger time step is adequate.

The influence of grid resolution is harder to test because of the calculation cost on the density of grids that are required. The three grid levels were used to locate the flutter point for a range of Mach numbers. Two calculations were run at different values of freestream density for each Mach number and the growth of the response calculated using the approach of Gordnier and Melville,[21] where the ratio of consecutive peaks was taken to define an amplification factor. Linear interpolation of the amplification was then used to estimate the value of density at which a neutrally stable response would be obtained. The medium grid calculations took about 5–6 h on a 2.5-GHz PC to calculate five periods of the flutter response.

The comparison at Mach 0.96, which is close to the bottom of the flutter dip, between the predicted flutter speeds on the three grid levels with other predictions is shown in Table 1. The results of Gordnier and Melville[21] show a downward (and accelerating) trend in the flutter speed, with bigger differences between the fine and medium than between the medium and coarse. The value from Rausch et al.[23] is lower still. The current results suggest that the medium grid provides good spatial resolution, with the three grids showing behavior consistent with spatial convergence. The grid-converged value using no structural damping is much lower than experiment. Using a value of structural damping of 2% shifts the flutter speed index above the experimental value.

The trends from these results suggest that the grid-converged solution without structural damping is significantly below the experimental result. Adding structural damping brings the flutter speed back up into the range of the measurements, as shown in Fig. 2. The solution obtained using 0.5% structural damping is in good agreement with the experimental values.

Bifurcation Results

The bifurcation solver was applied on the coarse and medium grids using the BIFOR code. Currently no parallel version of the bifurcation code has been written, and the main consequence of this is that the medium grid is the largest problem that can be tackled on the computers that were available. This case requires 1.5 Gb of memory. The PMB and BIFOR codes are based on similar formulations for the steady-state solutions. However, the PMB code has been under development for 10 years and so has reached a level of optimization well in advance of the newer BIFOR code. To quantify this, an equivalent implicit step with the BIFOR code requires 2.35 times the CPU time when compared with the PMB code. This will be accounted for in the comparisons between the bifurcation and time-marching calculation costs by expressing these as multiples of the CPU time required for a steady-state calculation with the same code. The timings are likely to be conservative when assessing the performance of the bifurcation method because additional gains are likely from writing a dedicated linear solver (i.e., one that is not configured to handle general-sized matrix blocks).

Guided by the time-marching results, a value of structural damping of 0.5% was used. The comparison on the medium grid between the measured, time-marching, and bifurcation results is shown for the dip region in Fig. 3. The bifurcation boundary was computed first for eight Mach numbers between 1.07 and 0.67, with a Mach number interval of 0.05, and subsequently for 12 Mach numbers in the dip region, with an interval of 0.01. The frequency from the time-marching calculations was used with the inverse power method at the largest Mach number to initiate the calculations. Good agreement between the predictions of the two codes is observed as required.

Table 1 Grid-refinement influence on flutter speed index at Mach 0.96

Reference	Grid volume	Grid surface	Damping	Flutter speed index
Current	Coarse	1,131	0	0.227
Current	Medium	4,453	0	0.192
Current	Fine	17,700	0	0.175
Current	Coarse	1,131	2%	0.401
Current	Medium	4,453	2%	0.381
Current	Fine	17,700	2%	0.375
21	Coarse	2,366	0	0.314
21	Medium	5,340	0	0.304
21	Fine	9,231	0	0.285
23	Unstructured	N/A	0	0.230
22	Structured	5,289	0	0.294

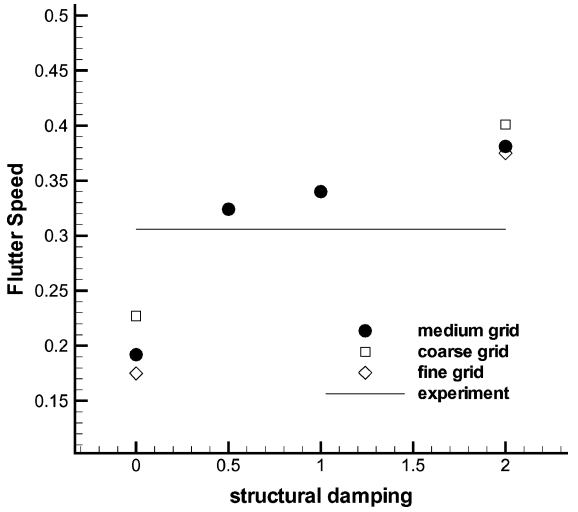

Fig. 2 Variation of flutter speed index with the structural damping applied at Mach 0.96: ——, measurement.

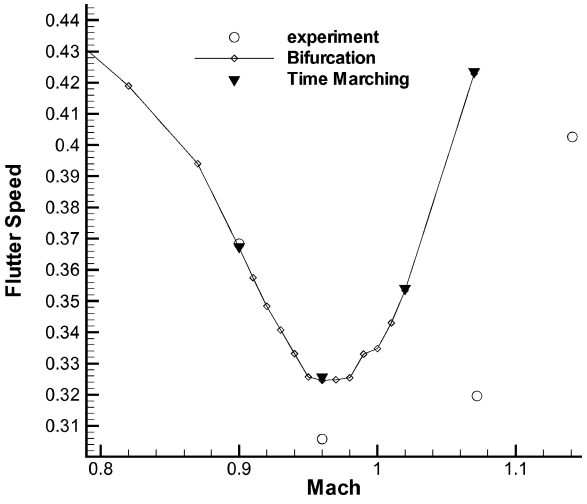

Fig. 3 Comparison with measurements of the stability boundaries calculated on the medium grid using time-marching and the bifurcation solver.

Table 2 Average calculation cost using the PMB code for the first two rows and the BIFOR code for the bottom two rows in the table (The relative costs have been scaled by the time for a steady-state calculation with the appropriate code.)

Calculation type	CPU, s	CPU (relative)
Steady state	787	1
Unsteady solution	19,810	45
Steady calculation	1,767	1
Bifurcation calculation	3,304	1.87

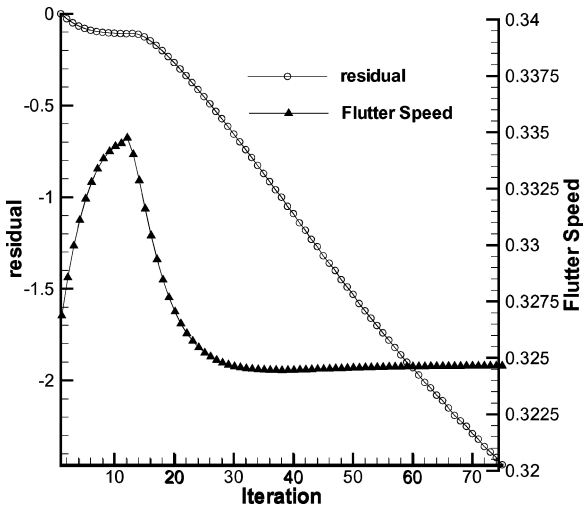

Fig. 4 Convergence of the bifurcation parameter with the bifurcation solver iteration number at Mach 0.96. The flutter speed index is shown against the right-hand axes and the augmented residual against the left-hand axes.

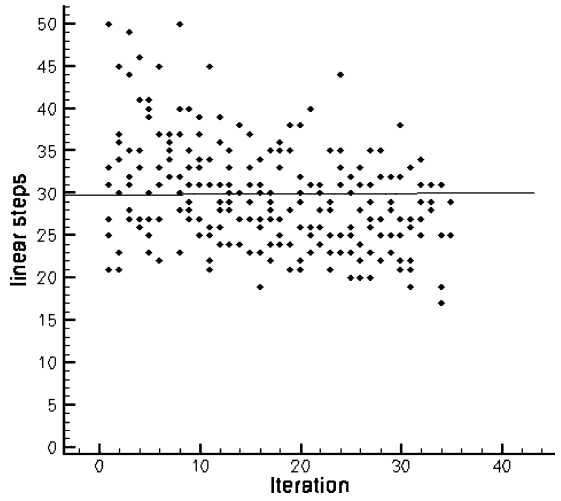

Fig. 5 Number of linear solver steps per bifurcation solver iteration: ——, average number of steps for an aerofoil calculation.[6]

An assessment of the relative cost of the time-marching and bifurcation calculations can be made from the times in Table 2. Here the average CPU time of an unsteady calculation and the average bifurcation cost for each Mach number have been expressed in multiples of the cost of a steady-state calculation using the respective codes. The steady-state calculation in each case has been converged five orders of magnitude. The bifurcation solver has been converged to at least three significant figures for the flutter speed, as indicated in Fig. 4. The stopping criterion is based on reducing the magnitude of the augmented residual, defined by Eq. (18), by one order from the starting value. The time for the unsteady calculations is based on 750 time steps resolving five cycles.

Similar conclusions to the previous aerofoil study can be drawn. The time required to trace out the flutter boundary for eight Mach numbers is less than half the cost of a single time-marching calculation. Considering the number of time-marching calculations required to trace out a flutter boundary, the calculation cost can be reduced by two orders of magnitude by using the bifurcation method.

One concern with the previous work on aerofoil problems was that the performance of the linear solver would deteriorate for larger problems. On average for the aerofoil 30 iterations were required to achieve a reduction of two orders in the residual. The number of linear solver iterations at each bifurcation iteration is shown in the scatter plot in Fig. 5 along with the average number of iterations required for the previous aerofoil calculations. The fact that the number of linear solver iterations is spread about the average two-dimensional cost indicates that the performance of the Krylov solver has been maintained for the larger three-dimensional problems.

Conclusions

The performance of a bifurcation method for calculating flutter boundaries has been evaluated for the AGARD 445.6 wing test case. This is believed to be the first time that such a method has been used to calculate a three-dimensional aeroelastic instability problem. The good performance of the method previously observed for aerofoil problems has been preserved. In particular the cost of the iterative linear solver in terms of the number of iterations required has not increased as the size of the matrix has increased. It is estimated that the cost of tracing out a flutter boundary over 10 Mach numbers has been reduced by two orders of magnitude compared with the time-marching method.

The method presented relied on the system being symmetric, which means that the equilibrium solution is independent of the bifurcation parameter. In the asymmetric case the equilibrium solution has to be recalculated as the bifurcation parameter is updated during the Newton iterations. Two approaches are being contemplated to realize this. First, the problem could be solved in a completely coupled way, with the Jacobian matrix driving the Newton convergence given by Eq. (20). This has the likely advantage of optimizing the convergence rate at a cost of having to store and use a matrix that is around 50% larger. The alternative is to decouple the equilibrium and bifurcation calculations, which in effect ignores the terms in the first row, fourth and first columns, and third and fourth rows of the matrix in Eq. (20). This approach will probably degrade the convergence but has the benefit of reducing the storage requirements as the equilibrium and bifurcation calculations can be sequenced. Further work is required to evaluate the merits of these approaches.

The next stage of this work is to implement a parallel version of the code with the linear solver tailored for the particular block sizes encountered for this problem. This will allow the goal of assessing the method for full aircraft flutter calculations to be realized.

Acknowledgments

This work was supported by Engineering and Physical Sciences Research Council, Ministry of Defence, and BAE Systems and forms part of the work programme of the Partnership for Unsteady Methods Defence and Aerospace Research Partnership (PUMA DARP).

References

[1]Farhat, C., Geuzaine, P., and Brown, G., "Application of a Three-Field Nonlinear Fluid-Structure Formulation to the Prediction of the Aeroelastic Parameters of an F-16 Fighter," *Computers and Fluids*, Vol. 32, No. 3, 2002, pp. 3–29.

[2]Melville, R., "Nonlinear Simulation of F-16 Aeroelastic Instability," AIAA Paper 2001-0570, Jan. 2001.

[3]Morton, S. A., and Beran, P. S., "Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds," *Journal of Aircraft*, Vol. 36, 1999, pp. 421–429.

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

[5]Beran, P. S., "A Domain-Decomposition Method for Airfoil Flutter Analysis," AIAA Paper 98-0098, 1998.

[6]Badcock, K. J., Woodgate, M. A., and Richards, B. E., "Hopf Bifurcation Calculations for a Symmetric Airfoil in Transonic Flow," *AIAA Journal*, Vol. 42, No. 5, 2004, pp. 883–892.

[7]Badcock, K. J., Richards, B. E., and Woodgate, M. A., "Elements of Computational Fluid Dynamics on Block Structured Grids Using Implicit Solvers," *Progress in Aerospace Sciences*, Vol. 36, 2000, pp. 351–392.

[8]Yates, E. C., "AGARD Standard Aeroelastic Configurations for Dynamic Response 1: Wing 445,6," AGARD, Rept. 765, 1988.

[9]Osher, S., and Chakravarthy, S. R., "Upwind Schemes and Boundary Conditions with Applications to Euler Equations in General Coordinates," *Journal of Computational Physics*, Vol. 50, 1983, pp. 447–481.

[10]Van Leer, B., "Towards the Ultimate Conservative Conservative Difference Scheme II: Monotonicity and Conservation Combined in a Second Order Scheme," *Journal of Computational Physics*, Vol. 14, 1974, pp. 361–374.

[11]Cantariti, F., Dubuc, L., Gribben, B., Woodgate, M., Badcock, K., and Richards, B., "Approximate Jacobians for the Solution of the Euler and Navier–Stokes Equations," Univ. of Glasgow, Aerospace Engineering Rept. 9704, Scotland, U.K., 1997.

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

[13]Goura, G. S. L., "Time Marching Analysis of Flutter Using Computational Fluid Dynamics," Ph.D. Dissertation, Dept. of Aerospace Engineering, Univ. of Glasgow, Scotland, U.K., Nov. 2001.

[14]Goura, G. S. L., Badcock, K. J., Woodgate, M. A., and Richards, B. E., "Extrapolation Effects on Coupled CFD-CSD Simulations," *AIAA Journal*, Vol. 41, No. 2, 2003, pp. 312–314.

[15]Tsai, H. M., Wong, A. S. F., Cai, J., Zhu, Y., and Liu, F., "Unsteady Flow Calculations with a Parallel Multiblock Moving Mesh Algorithm," *AIAA Journal*, Vol. 39, No. 6, 2001, pp. 1021–1029.

[16]Gordon, W. J., and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," *International Journal for Numerical Methods in Engineering*, Vol. 7, 1973, pp. 461–477.

[17]Seydel, R., *Practical Bifurcation Analysis and Stability Analysis*, 2nd ed., Springer-Verlag, Berlin, 1994.

[18]Tuminaro, R. S., Heroux, M., Hutchinson, S. A., and Shahid, J. N., "Official Aztec User's Guide," Ver. 2.1, Sandia Lab., SAND99-8801J, Albuquerque, NM, 1999.

[19]Axelsson, O., *Iterative Solution Methods*, Cambridge Univ. Press, Cambridge, UK, 1994.

[20]Goura, G. S. L., Badcock, K. J., Woodgate, M. A., and Richards, B. E., "Implicit Method for the Time Marching Analysis of Flutter," *Aeronautical Journal*, Vol. 105, No. 1046, April 2001, pp. 192–214.

[21]Gordnier, R. E., and Melville, R. B., "Transonic Flutter Simulations Using an Implicit Aeroelastic Solver," *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 872–879.

[22]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.

[23]Rausch, R. D., Batina, J. T., and Yang, H. T. Y., "Three-Dimensional Time Marching Aeroelastic Analyses Using an Unstructured-Grid Euler Method," *AIAA Journal*, Vol. 31, No. 9, 1993, pp. 1626–1633.