The Application of Sparse Matrix Techniques to the CFD based Aeroelastic Bifurcation Analysis of a Symmetric Wing

K.J.Badcock, M.A. Woodgate, B.E.Richards and G.N.Barakos
Department of Aerospace Engineering, University of Glasgow, Glasgow G12 8QQ, U.K.

Abstract

The application of a sparse matrix solver for the direct calculation of Hopf bifurcation points for the flexible AGARD wing in a transonic flow modelled using CFD 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 wing can be traced out in a time comparable to that required for a small number of time response calculations, yielding two orders of magnitude improvement when compared to the time marching approach.

1 Introduction

Time domain analysis is the main solution method used in computational aeroelasticity. However, the need to execute searches over multi-parameter space to identify stability behaviour leads to high computational cost due to 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 behaviour at previously identified problem conditions and there are several recent impressive demonstrations of this kind for complete aircraft configurations [1] [2].

A way of reducing the cost of parametric searches for stability behaviour was proposed by Morton and Beran from the US Air Force Laboratories. Their method uses dynamical systems theory to characterise 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 effectively calculates the value of the parameter where an eigenvalue of the system Jacobian matrix crosses the imaginary axis at the flutter point. A convection-diffusion problem was used to evaluate the approach [3]. The method was then applied to an aeroelastic system consisting of an aerofoil moving in pitch and plunge in reference [4]. 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 [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 [3] and [5].

The problems introduced by using a direct solver were resolved in [6] where a sparse matrix formulation was used to allow in principal the solution of the linear system for much larger grids. The Newton iteration was also modified to enhance the efficiency of the scheme following work on approximate Jacobian matrices for CFD only problems [7]. The method was shown to be effective for tracing out flutter boundaries for symmetric aerofoils moving in pitch and plunge, with reductions in the computational time required of two orders of magnitude 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, the use of a modal structural model and the resulting requirement to pass information between non-matching grids, and the larger problem size (eg for the linear system). The formulation is considered in the following section and then results are presented for the AGARD wing test case [8] to demonstrate the initial feasibility of the method for realistic problems.

2 Aerodynamic and Structural Simulations

2.1 Aerodynamics

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

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

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

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

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

In the above 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 $\dot{z}$ i.e.

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

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

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

The flow solution in the current work is obtained using the Glasgow University code PMB (parallel multi-block). A summary of the applications examined using the code can be found in reference [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

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

The term on the right hand side, called the residual, is the discretisation of the convective terms, given here by Osher's approximate Riemann solver [9], MUSCL interpolation [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 set 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. These are solved by an implicit method [11], the main features of which are an approximate linearisation 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 [12].

2.2 Structural Dynamics, Inter-grid Transformation and Mesh Movement

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

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

where ϕ_i are the mode shapes calculated from a full finite element model of the structure and α_i are the generalised coordinates. By projecting the finite element equations onto the mode shapes the scalar equations

$$\frac{d^2\alpha_i}{dt^2} + \omega_i^2\alpha_i = \mu\phi_i^T\mathbf{f}_s \tag{9}$$

are obtained where $\mathbf{f}_s$ is the vector of aerodynamic forces at the structural grid points and μ is a coefficient related to the fluid freestream dynamic pressure which redimensionalises the aerodynamic forces. These equations are rewritten as a system in the form

$$\frac{d\mathbf{w}_s}{dt} = \mathbf{R}_s \tag{10}$$

where $\mathbf{w}_s = (....., \alpha_i, \dot{\alpha}_i, ...)^T$ and $\mathbf{R}_s = (....., \dot{\alpha}_i, \mu\phi_i^T\mathbf{f}_s - \omega_i^2\alpha_i, ...)^T$.

The aerodynamic forces are calculated at cell centres on the aerodynamic surface grid. The problem of communicating these forces to the structural grid is complicated in the common situation where these grids not only don't match but also are not defined on the same surface. This problem, and the influence it can have on the aeroelastic response, was considered in [13] and [14], where a method was developed, called 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 $\mathbf{x}_a$ and $\mathbf{f}_a$, then

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

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

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

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

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

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

where the structural grid speeds are given by

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

The geometries of interest deform during the motion. This means, unlike the rigid aerofoil problem, that the aerodynamic mesh must be deformed rather than rigidly translated and rotated. This is achieved using transfinite interpolation of displacements (TFI) as described in reference [15]. The grid speeds are also interpolated from known boundary speeds. In this way the grid locations depend on α_i and the speeds on $\dot{\alpha}_i$.

2.3 Time Domain Solver

For coupled CFD-CSD calculations the aerodynamic and structural solutions must be sequenced. For steady solutions, taking one step of the CFD solver followed by one step of the structural solver will result in the correct equilibrium. However, for time accurate calculations more care must be taken to avoid introducing additional errors. The exact formulation used to avoid this is discussed in reference [16].

3 Formulation of Hopf Analysis

The semi-discrete form of the coupled CFD-CSD system

$$\frac{d\mathbf{w}}{dt} = \mathbf{R}(\mathbf{w}, \mu) \tag{12}$$

where

$$\mathbf{w} = [\mathbf{w}_f, \mathbf{w}_s]^T \tag{13}$$

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

$$\mathbf{R} = [\mathbf{R}_f, \mathbf{R}_s]^T \tag{14}$$

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

Dynamical systems theory gives criteria for an equilibrium to be stable [17]. In particular, all eigenvalues of the Jacobian matrix of equation (12), given by $A = \partial \mathbf{R} / \partial \mathbf{w}$, 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 $A(\mathbf{w}_0, \mu)$ has one eigenvalue $i\omega$ which crosses the imaginary axis. Denoting the corresponding eigenvector by $\mathbf{P} = \mathbf{P}_1 + i\mathbf{P}_2$, a critical value of μ is one at which there is an eigenpair ω and $\mathbf{P}$ such that

$$A\mathbf{P} = i\omega\mathbf{P}. \tag{15}$$

This equation can be written in terms of real and imaginary parts as $A\mathbf{P}_1 + \omega\mathbf{P}_2 = \mathbf{0}$ and $A\mathbf{P}_2 - \omega\mathbf{P}_1 = \mathbf{0}$. A unique eigenvector is chosen by scaling against a constant real vector $\mathbf{q}$ to produce a

fixed complex value, taken to be $0 + 1i$. This yields two additional scalar equations $\mathbf{q}^T\mathbf{P}_1 = 0$ and $\mathbf{q}^T\mathbf{P}_2 - 1 = 0$.

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

$$\mathbf{R}_A(\mathbf{w}_A) = \mathbf{0} \tag{16}$$

where

$$\mathbf{R}_A = \begin{bmatrix} \mathbf{R} \\ A\mathbf{P}_1 + \omega\mathbf{P}_2 \\ A\mathbf{P}_2 - \omega\mathbf{P}_1 \\ \mathbf{q}^T\mathbf{P}_1 \\ \mathbf{q}^T\mathbf{P}_2 - 1 \end{bmatrix} \tag{17}$$

and $\mathbf{w}_A = [\mathbf{w}, \mathbf{P}_1, \mathbf{P}_2, \mu, \omega]^T$. If there are n components in $\mathbf{w}$ then $\mathbf{w}_A$ has $3n + 2$ components, as does $\mathbf{R}_A$. Hence equation (16) 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 $\mathbf{w}_A^n$ to a solution is generated by solving the linear system

$$\frac{\partial \mathbf{R}_A}{\partial \mathbf{w}_A}\Delta\mathbf{w}_A = -\mathbf{R}_A^n \tag{18}$$

where $\Delta\mathbf{w}_A = \mathbf{w}_A^{n+1} - \mathbf{w}_A^n$. The Jacobian matrix on the left hand side of equation (18) is given in expanded form as

$$\frac{\partial \mathbf{R}_A}{\partial \mathbf{w}_A} = \begin{bmatrix} A & 0 & 0 & \mathbf{R}_\mu & 0 \\ (A\mathbf{P}_1)_\mathbf{w} & A & I\omega & (A\mathbf{P}_1)_\mu & \mathbf{P}_2 \\ (A\mathbf{P}_2)_\mathbf{w} & -I\omega & A & (A\mathbf{P}_2)_\mu & -\mathbf{P}_1 \\ 0 & \mathbf{q}^T & 0 & 0 & 0 \\ 0 & 0 & \mathbf{q}^T & 0 & 0 \end{bmatrix}. \tag{19}$$

There are three key issues for the application of equation (18). First, a good initial guess is required or the iterations will diverge. Secondly, the Jacobian matrix $\partial\mathbf{R}_A/\partial\mathbf{w}_A$ is required. Thirdly, the large sparse linear system given in equation (18) must be solved. These points were considered for the aerofoil problem in [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 previously for the aerofoil case in [6] and are only summarised 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 used, again using the sparse matrix formulation, to calculate the eigenvector corresponding to the critical eigenvalue. This information is then used as the starting solution for the Hopf calculation at the first Mach number and then at 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 [19]. Whilst not optimised 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 factorisation family [18] can be very effective at approximating the inverse of the coefficient matrix with a

small number of terms. For CFD calculations, block ILU factorisations with no fill in have proved very successful [7]. Here no fill in means that the factorisation has the same sparsity pattern as the coefficient matrix. This method has been used in the current work.

One simplification arises if we are dealing with a symmetric problem eg a symmetrical aerofoil at zero incidence [4]. In this case the equilibrium solution $\mathbf{w}_0$ is independent of μ and hence can be calculated from equation (12) independently of the other Hopf conditions in equation (16). Then, a smaller system can be solved for the bifurcation parameter. This simplification is exploited in the current work, although is not inherent in the approach used.

4 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

$$A = \begin{bmatrix} \frac{\partial \mathbf{R}_f}{\partial \mathbf{w}_f} & \frac{\partial \mathbf{R}_f}{\partial \mathbf{w}_s} \\ \frac{\partial \mathbf{R}_s}{\partial \mathbf{w}_f} & \frac{\partial \mathbf{R}_s}{\partial \mathbf{w}_s} \end{bmatrix} = \begin{bmatrix} A_{ff} & A_{fs} \\ A_{sf} & A_{ss} \end{bmatrix}. \tag{20}$$

The block A_{ff} describes the influence of the fluid unknowns on the fluid residual and has by far the largest number of non zeros for a modal structural model. The treatment of this term is crucial to the efficiency of the scheme and is discussed in [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 only calculations [7].

The only issue for the Newton iteration matrix is that the scheme converges to the correct answer, which is determined by the calculation of the residual on the right hand side of the Newton iteration. Hence, the products $A\mathbf{P}_1$ and $A\mathbf{P}_2$ must be computed exactly. This can be done using a matrix free formulation as

$$A\mathbf{x} \approx \frac{\mathbf{R}(\mathbf{w} + h\mathbf{x}) - \mathbf{R}(\mathbf{w} - h\mathbf{x})}{2h} \tag{21}$$

where $\mathbf{x}$ 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 a very accurate approximation to the required product without having to evaluate and store A. The matrix A is required for the left hand side coefficient matrix but the modified order approximation is used for this purpose which reduces the storage. Hence, 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 generalised coordinates and velocities, transferring these to the fluid surface grid using the linearised CVT transformation and then applying TFI to transfer these boundary values to the volume grid. By using second order finite differences, the terms for A_{fs} can be calculated in $2n_s$ evaluations of $\mathbf{R}_a$ if there are n_s structural unknowns.

The term A_{sf} involves calculating the dependence of the generalised fluid forces on the fluid unknowns. The surface forces on the aerodynamic grid are calculated and then transferred to the

structural grid using the CVT transformation. The inner product is then formed using the forces on the structural grid and the modal coefficients. The Jacobian matrix for the force 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. However, it is important to remember that the generalised force will change with the structural unknowns also since the surface normals to the wing will change as the wing moves. A finite difference calculation is used here to include this effect.

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

$$A_\mu = \begin{bmatrix} 0 & 0 \\ 0 & \frac{\partial^2 \mathbf{R}_s}{\partial \mu \partial \mathbf{w}_s} \end{bmatrix} \tag{22}$$

Due to the simple algebraic expression for $\partial \mathbf{R}_s / \partial \mathbf{w}_s$ it is straightforward to calculate the required term analytically.

5 Results for Symmetric Problem

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 hence more expensive than rigid model tests. A complete set of measurements is available for the AGARD 445.6 wing and results have been included for most simulation papers on flutter, giving a wide range of data with which to evaluate the current method. However, the disadvantage of the current test case is that it does not feature significant nonlinear effects since 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 in reference [16].

The AGARD 445.6 wing is made of mahogany and has a 45^o quarter chord sweep, a root chord of 22.96 inches and a constant NACA64A004 symmetric profile [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 whilst being structurally weaker. Published experimental data includes 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 [13]. Four modes are retained with the first to bending modes having frequencies 9.7 and 50.3, and the two torsional modes at 36.9 and 90.0.

A multiblock grid with a CH topology was generated containing 100 thousand points (referred to as the fine grid). From this grid a second was extracted by taking every second point in each coordinate direction to give a coarse grid containing 12 thousand points.

Two sets of results were generated for comparison. First, the bifurcation method was used to trace out the flutter boundary. Secondly, the time marching method was used to check the consistency of the results. The computed boundaries are shown in figure 1, in terms of reduced air speed, on the two grids. As expected, grid refinement shifts the curves down (i.e. makes the system flutter at lower values of dynamic pressure). From the results a finer grid level would be required to obtain

grid independent results closer to experimental measurements. However, as discussed below, it is not currently possible to use the bifurcation method on finer grids and so this was not done.

More importantly for demonstrating the direct method, the results are consistent with those from time marching solutions. Care has been taken to test the time marching results for independence from time step and pseudo tolerance. A time step of 0.2 and a pseudo residual of 0.01 was used. This means around 10 pseudo iterations per real time step. The cost of each unsteady calculation is around 50 times that of a steady state calculation.

The calculations were run on a 2.5 GHz PC. The fine grid bifurcation calculation traced out the boundary between Mach numbers of 0.67 and 1.14 in steps of 0.05. The bifurcation calculation on the fine grid took 1.56 times the required steady state calculation at the same Mach number. Thirty Newton iterations were used at each Mach number. A major concern is the performance of the linear solver for the larger problems encountered in 3d. However, on average 38 Krylov steps are needed at each Newton step which is similar to the costs of solving the linear systems in the aerofoil case [6].

To evaluate the bifurcation method, the cost of tracing out the boundary using 9 Mach numbers is equivalent to 23 steady state calculations. Using time marching, and making a conservative estimate that 3 time marching calculations will be required at each Mach number, the cost will be equivalent to over one thousand steady state calculations. Therefore, the bifurcation method turns out to be around two orders of magnitude more efficient than brute force time marching. This is consistent with the conclusions from the aerofoil cases [6].

Current efforts are being directed at an efficient parallel implementation to allow large scale problems to be computed.

6 Conclusions

The method proposed in [6] for calculating Hopf Bifurcations of two dimensional systems has been successfully applied to the three dimensional AGARD 445.6 wing case. The performance of the Krylov based sparse linear solver and the approximate Newton iteration scheme has been shown to be consistent with the previous two dimensional results.

7 Acknowledgements

This work was supported by EPSRC, MoD, DERA and BAE SYSTEMS.

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, to appear, 2002.

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

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

[4] Morton, S.A. and Beran, P.S., *Hopf-Bifurcation Analysis of Airfoil Flutter at Transonic Speeds*, J Aircraft, 36, pp 421-429, 1999.

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

[6] Badcock, K.J., M.A. Woodgate, M.A. and Richards, B.E., The Application of Sparse Matrix Techniques for the CFD based Aeroelastic Bifurcation Analysis of a Symmetric Aerofoil, submitted to AIAA J, 2002.

[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 Report 765, 1988.

[9] Osher,S. and Chakravarthy, S.R., *Upwind Schemes and Boundary Conditions with Applications to Euler equations in General Coordinates* , Journal Computational Physics, vol. 50 , 1983, p 447-481.

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

[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*, University of Glasgow, Aerospace Engineering report 9704, 1997.

[12] Jameson, A., *Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings*, Technical report, AIAA 91-1596, 1991.

[13] Goura, G.S.L., *Time Marching Analysis of Flutter using Computational Fluid Dynamics*, PhD thesis, University of Glasgow, 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 J, March, 2003.

[15] Gordon, W.J. and Hall, C.A., *Construction of curvilinear coordinate systems and applications to mesh generation*, Int J Num Meth Engr, 7, 1973, 461-477.

[16] 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, volume 105, number 1046, April, 2001.

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

[18] Axelsson, O., *Iterative Solution Methods*, Cambridge University Press, 1994.

[19] Tuminaro, R.S., Heroux, M., Hutchinson, S.A. and Shahid, J.N., *Official Aztec User's Guide Version 2.1*, SAND99-8801J, Sandia Laboratory, 1999.

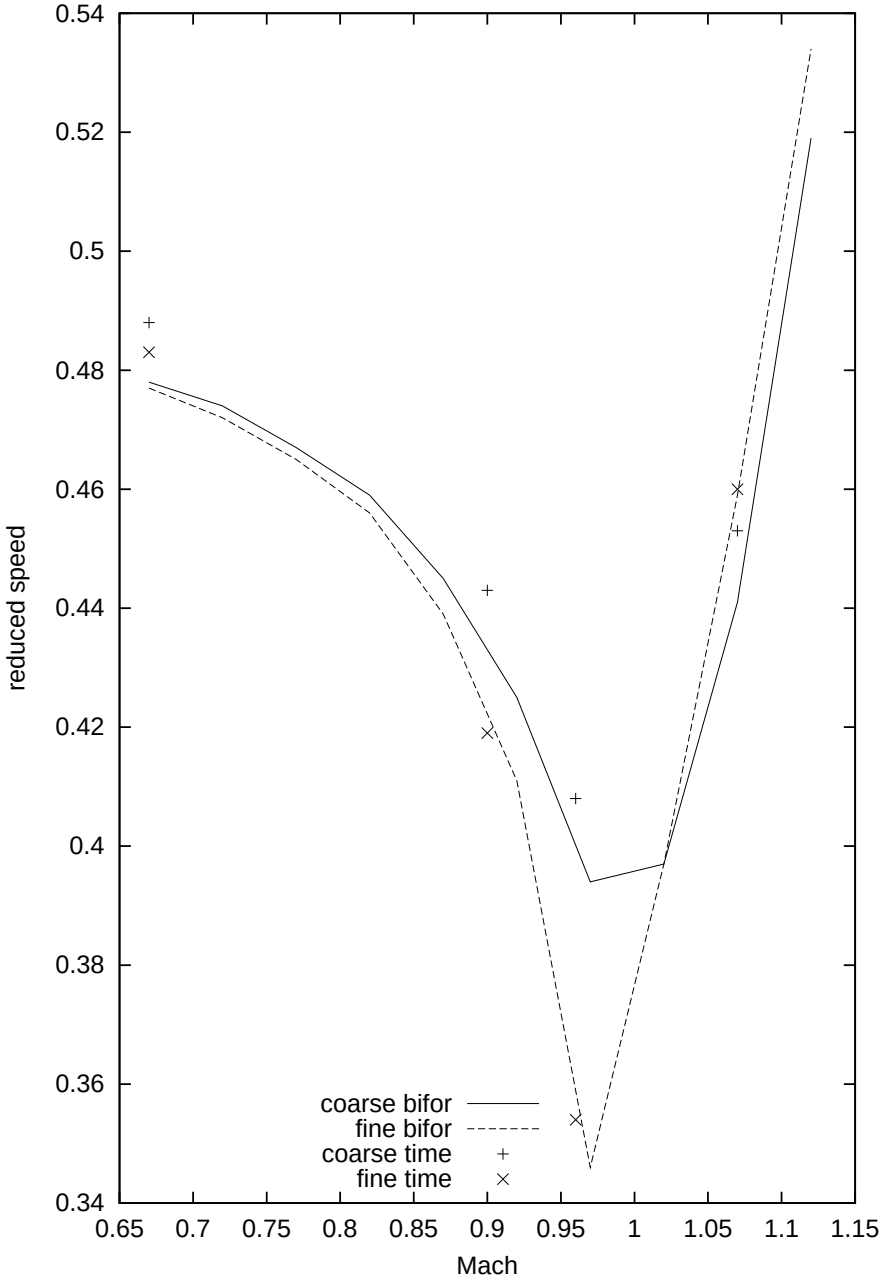

Figure 1: Flutter boundary computed using time marching and bifurcation methods on the coarse and fine grids.