Huntley, S., Jones, D., & Gaitonde, A. (2017). Bifurcation Tracking for High Reynolds Number Flow Around an Airfoil. *International Journal of Bifurcation and Chaos*, *27*(4), [1750061]. https://doi.org/10.1142/S0218127417500614

International Journal of Bifurcation and Chaos

BIFURCATION TRACKING FOR HIGH REYNOLDS NUMBER FLOW AROUND AN AIRFOIL

S. HUNTLEY*, D. JONES and A. GAITONDE
Department of Aerospace Engineering, University of Bristol,
University Walk, Bristol, BS8 1TR, UK
**Samantha.Huntley@bristol.ac.uk*

Received (to be inserted by publisher)

High Reynolds number flows are typical for many applications including those found in aerospace. In these conditions non-linearities arise which can, under certain conditions, result in instabilities of the flow. The accurate prediction of these instabilities is vital to enhance understanding and aid in the design process. The stability boundary can be traced by following the path of a bifurcation as two parameters are varied using a direct bifurcation tracking method. Historically, these methods have been applied to small-scale systems and only more recently have been used for large systems as found in Computational Fluid Dynamics. However, these have all been concerned with flows that are inviscid. We show how direct bifurcation tracking methods are can be applied efficiently to high Reynolds number flows around an airfoil. This has been demonstrated through the presentation of a number of test cases using both flow and geometrical parameters.

Keywords: Hopf bifurcation,continuation,Navier-Stokes,bifurcation tracking

1. Introduction

When modeling flow that is inherently non-linear in nature it is possible that solutions occur that are unexpected and the reasons behind this must be understood. This is often a result of the non-linearities that arise when the system undergoes a bifurcation. A bifurcation is a phenomenon associated with a change in behavior of the system and can result in the loss of stability of the system. Identification of bifurcations, therefore, is necessary to understand when these instabilities occur and aid the design of the system of interest, in this case aircraft design.

A number of software packages aimed at performing bifurcation analysis of non-linear systems are currently available including AUTO [Doedel *et al.*, 1991b,a, 1986], MATCONT [Dhooge *et al.*, 2003] and PITCON [Rheinboldt & Burkardt, 1983]. These packages are designed to solve low-dimensional systems and not large-scale problems, largely due to the direct solvers they use. The solvers they use are developed for dense matrices and would be too computationally expensive for large scale problems [Wales, 2010]. There are, however, a limited number of bifurcation analysis tools that have been designed to solve large-scale problems. One of these packages is PDEcont [Lust *et al.*, 1998], which is concerned with time-discrete simulations and, as such, uses a Newton-Picard method. Another package is the LOCA library [Lehoucq &

*Author for correspondence.

Salinger, 2001; Salinger *et al.*, 2002]. They use an indirect solver known as Generalised Minimum Residual (GMRES). In order for these solution methods to be efficient they require a good preconditioner [Wales, 2010]. This is still an active area of research for the types of flow of interest in this work. Finally, the work of Bindel *et al.* [2014] implements the Continuation of Invariant Subspaces (CIS) method using a Matlab code to track simple bifurcations. This uses a projection method to reduce the size of the problem but requires a good approximation to the least stable subspace.

Bifurcation analysis in Computational Fluid Dynamics (CFD) has mainly been limited to continuation methods. These have predominately been aimed at low-dimensional, enclosed flows, such as Taylor-Couette flow, of which a review was conducted by Cliffe *et al.* [2001]. More recently continuation methods have been applied to larger fluid dynamics systems but with the main interest still being well-conditioned problems as the review by Dijkstra *et al.* [2014] highlights. The studies by Wales [Wales, 2010; Wales *et al.*, 2012b,a] have shown that continuation methods can be applied to high Reynolds number external flows, such as is of interest in this work. However these studies did not look into tracking the bifurcation as two parameters were changed but the results from this study suggests that bifurcation tracking methods should be applicable also.

The application of direct bifurcation tracking methods to transonic flows is limited but a few studies have been carried out [Badcock *et al.*, 2004, 2005; Woodgate & Badcock, 2007]. In Woodgate & Badcock [2007], the calculation of the stability limit of several aeroelastic systems using a direct bifurcation tracking method was investigated. They used an analytical Jacobian and performed simulations on systems that were smaller in size than those used in this work. They also used an analytical Jacobian in [Badcock *et al.*, 2004], which also only looked at symmetric cases, which simplifies the problem. This was also the case in the work of Badcock *et al.* [2005] where they demonstrated bifurcation tracking on aeroelastic systems using the Euler equations.

The aim of this study is to demonstrate the applicability of existing bifurcation computation and tracking methods to high Reynolds number external flows. Previous studies have not looked into tracking the bifurcation of large systems as two parameters change however the results presented here suggest that bifurcation computation and tracking methods are applicable to these systems.

2. CFD Model Formulation

This section describes the method used to model the fluid dynamics of the flow around an airfoil. The method is the same as used in the work of Wales *et al.* [2012a] where it has also been verified. The flow is modeled using the Reynolds Average Navier-Stokes (RANS) equations which also requires a model for the Reynolds stresses, achieved using the Spalart-Allmaras turbulence model here.

2.1. *Governing equations*

The two-dimensional Reynolds Averaged Navier-Stokes equations can be written in conservative form as

$$\frac{\partial \mathbf{Y}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{1}$$

where t is time, $\mathbf{Y}$ is the vector of conserved variables, $\mathbf{F}$ and $\mathbf{G}$ are flux vectors and x and y are the Cartesian coordinates.

The vector of conserved variables, $\mathbf{Y}$, is given by

$$\mathbf{Y} = (\rho, \rho u, \rho v, \rho E)^T . \tag{2}$$

Here ρ is the density, u and v are the velocity components in the x and y direction respectively and E is the total energy.

The flux vectors $\mathbf{F}$ and $\mathbf{G}$ consist of inviscid and viscid components so they may be written as

$$\mathbf{F} = \mathbf{F^I} - \mathbf{F^V}, \ \mathbf{G} = \mathbf{G^I} - \mathbf{G^V}. \tag{3}$$

The viscous flux vectors require the computation of a laminar viscosity and a turbulent viscosity. The laminar eddy viscosity is calculated using Sutherland's law and the turbulent eddy viscosity is given by the Spalart-Allmaras turbulence model described below.

2.2. *Spalart-Allmaras turbulence model*

In order to compute the turbulent eddy viscosity, the Spalart-Allmaras one-equation turbulence model was used. The Spalart-Allmaras model is robust with respect to the grid spacing and has shown to provide good results for separated flows [Rumsey & Ying, 2002] and drag predictions [Vassberg *et al.*, 2008]. The scalar transport equation describes the turbulent viscosity, solving for a variable, ν, which is related to the turbulent eddy viscosity. The transport equation is given as

$$\frac{\partial \nu}{\partial t} + U_i \cdot \nabla \nu - \frac{1}{\sigma Re}(\nabla \cdot ((\tilde{\nu} + \nu)\nabla \nu) + c_{b2} |\nabla \nu|^2) - c_{b1}\tilde{\omega}(\nu)\nu + c_{w1} f_w(\nu) \left(\frac{\nu}{d}\right)^2 = 0 \tag{4}$$

where $\tilde{\nu}$ is the molecular viscosity, and the variable ν is related to the turbulent eddy viscosity by $\nu_T = \nu f_{v1}(\nu/\tilde{\nu})$. Other relevant equations are:

$$\tilde{\omega}(\nu) = \omega + \frac{\nu}{\kappa^2 d^2} f_{v2}(\nu/\tilde{\nu}), \qquad f_{v2}(x) = 1 - \frac{x}{1 + x f_{v1}(x)}, \qquad f_{v1}(x) = \frac{x^3}{x^3 + c_{v1}^3}, \qquad x = \frac{\nu}{\tilde{\nu}},$$

$$\omega = ||\nabla \times U_i||^2, \qquad f_w(g) = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6}, \qquad g(r) = r + c_{w2}(r^6 - r), \qquad r(\nu) = \frac{\nu}{\tilde{\omega}\kappa^2 d^2}$$

The constants [Spalart & Allmaras, 1994] are given by:

$$c_{b1} = 0.1355,\ c_{b2} = 0.622,\ \sigma = 2/3,\ c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{\sigma},\ c_{w2} = 0.3,\ c_{w3} = 2,\ c_{v1} = 7.1 \tag{5}$$

2.3. *Solution method*

The RANS equations are augmented with the turbulence model resulting in a system that now includes the flow variables for the Navier-Stokes equations and the those for the turbulence model. This set of equations is solved using a finite volume formulation which gives a system of the form

$$\frac{d\mathbf{Y}}{dt} + \mathbf{R} = 0 \tag{6}$$

where $\mathbf{Y}$ now contains both the mean flow variables and the turbulence model variable and $\mathbf{R}$ is the residual. The system is solved as part of a numerical continuation method via Newton's method allowing both stable and unstable solutions to be found. This is vital for this work as we are interested in the bifurcation point.

3. Numerical Continuation

In order to compute a starting solution for the bifurcation tracking algorithms, a continuation method was employed. This computed the system equilibria as a parameter, λ, was varied. At each equilibrium the corresponding stability information was also computed. This process was continued until a bifurcation had been detected. The method chosen to carry out this task was identical to that of the studies of Wales *et al.* [2012b,a] and more details can be found therein.

3.1. *Predictor-Corrector method*

In this work an arclength continuation method is applied using a predictor-corrector algorithm. The solution branch is found by calculating an initial solution, $(\mathbf{Y}^0, \lambda^0)$, then changing the value of the continuation parameter, λ, by a small amount and finding the solution at this new parameter value. Arclength continuation is used in order to allow the continuation to continue through a fold bifurcation. The solution curve is parameterized in terms of its arclength, which ensures that a singular point is not encountered. The arclength is given as the distance between two consecutive solution points along the curve.

The predictor-corrector algorithm works by computing an approximate solution in the predictor stage and then iterating on this value using Newton's method in the corrector stage to find the system equilibrium

point converged to acceptable accuracy. In this work a secant predictor is used. This requires the first two solutions on the branch, $(\mathbf{Y}^0, \lambda^0)$ and $(\mathbf{Y}^1, \lambda^1)$, to be generated. An approximate solution at the next parameter value is then generated using

$$\bar{\mathbf{X}}^{n+1} = \mathbf{X}^n + h\mathbf{t}^n \tag{7}$$

where $\mathbf{X} = (\mathbf{Y}^n, \lambda^n)$ and

$$\mathbf{t} = \frac{\mathbf{X}^n - \mathbf{X}^{n-1}}{\|\mathbf{X}^n - \mathbf{X}^{n-1}\|}. \tag{8}$$

This value is then used as the starting point for the Newton solver, which iterates on this value to find the actual solution at the current parameter value. If a turning point is encountered the standard parameter continuation method will fail and it is therefore necessary to allow both the solution vector, $\mathbf{Y}$, and the parameter itself, λ, to vary during the convergence process. This approach is known as arclength continuation. Allowing the parameter to vary introduces one extra unknown and an additional equation. This extra equation is the arclength equation used to parameterize the curve.

$$g(\mathbf{Y}, \lambda, \varsigma) = \left[\frac{\partial \mathbf{Y}}{\partial \varsigma}\right] \Delta\mathbf{Y} + \left[\frac{\partial \lambda}{\partial \varsigma}\right] \Delta\lambda - \Delta\varsigma = 0, \tag{9}$$

This process works by restricting the Newton updates to a hyperplane normal to the direction taken for the predictor. The complete augmented system which must be is solved is

$$\begin{bmatrix} \mathbf{J} & \frac{\partial \mathbf{R}}{\partial \lambda} \\ \frac{\partial g}{\partial \mathbf{Y}} & \frac{\partial g}{\partial \lambda} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{Y} \\ \Delta\lambda \end{bmatrix}^n = \begin{bmatrix} -\mathbf{R} \\ -g \end{bmatrix}^n, \tag{10}$$

where $\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{Y}}$ is the Jacobian of the flow variables. The system, Eq. 10, was solved using a bordered solver known as the block elimination mixed (BEM) [Govaerts, 1991] bordering algorithm. This requires only solves involving the sparse Jacobian, which were accomplished using Umfpack [Davis, 2004b,a; Davis & Duff, 1999, 1997], a direct solver designed for sparse systems.

Note also that the Jacobian was calculated numerically using first order differencing of the residual as in Wales [2010].

3.2. *Stability*

At each completed continuation step the eigenvalue problem

$$(\mathbf{A} - \Lambda\mathbf{I})\mathbf{v} = 0 \tag{11}$$

must be solved. Here $\mathbf{A} = \mathbf{J}$ is the Jacobian, Λ is the matrix of eigenvalues, $\mathbf{v}$ is the matrix of eigenvectors and $\mathbf{I}$ is the identity matrix [Wales, 2010].

In this work the Jacobian is a large matrix and computing all the eigenvalues would be impractical. Therefore only the most positive eigenvalues are computed as these are important for stability analysis. In order to do this, the eigenvalue problem package ARPACK [Lehoucq *et al.*, 1998] is used. This package uses an iterative solver known as Arnoldi's method to compute a subset of the system eigenvalues with the properties of interest, such as the rightmost eigenvalues. In order to ensure that no eigenvalues with large imaginary part that lie close to the imaginary axis are missed and the implementation is efficient, the iterative solver is combined with a preconditioner [Wales, 2010]. In this work the preconditioner used a combination of two rational transforms: the Cayley transform and Shift invert. These transforms are used as they allow better convergence to be achieved when iterative solvers are used to find well-separated eigenvalues [Voss, 2007].

The first transform used as a preconditioner to calculate the eigenvalues was the shift invert method. If the centre of the region of interest is denoted σ then the shift invert method transforms the eigenvalues

in the region of interest; those that are close to σ move far away from σ and those that are far away from σ move close to it. This means that the eigenvalues of interest now become well-separated after the transformation [Wales, 2010].

The Cayley transform was also used as a preconditioner. It transforms the eigenvalues in the region of interest, this time defined as those with real parts greater than $\frac{1}{2}(a_1 + a_2)$, to values outside the unit disk. It does this by mapping the eigenvalues with real parts close to a_2 to values far away from the unit disk. It also maps the eigenvalues close to a_1 to values with a small modulus [Wales, 2010].

In this work, these two methods are combined, the shift-invert is used first to produce a preliminary guess of the eigenvalues of interest and then the Cayley transform is employed to ensure that no eigenvalues have been missed [Wales, 2010].

4. Bifurcation Tracking

Continuation methods are useful because they allow the dependency of the solution on a particular parameter to be discovered. The main aspect of interest, however, is the bifurcation point. Continuation methods can detect when a bifurcation point has occurred but its exact location remains unknown, only the interval in which it lies is known. Therefore, once a bifurcation has been detected in the continuation, certain additional methods can be employed that allow it to be calculated more accurately. It is then possible to free a second parameter, γ, and track the bifurcation as it changes with both parameters. This allows a loci of bifurcation points in two parameter space to be mapped out which corresponds to the stability boundary of the problem being investigated. This is more efficient and accurate than performing continuation at multiple fixed values of the second parameter, just to determine the stability boundary.

The basic premise behind all bifurcation tracking methods is the same. They utilize the fact that a bifurcation results in a change in stability. The stability of the steady solutions to small perturbations can be determined using linear stability analysis. This gives the eigenvalue problem, which has the form $\mathbf{J}\mathbf{v} = \eta\mathbf{v}$. Here η is the eigenvalue and $\mathbf{v}$ is the corresponding eigenvector. The eigenvalue and eigenvector generally both have complex form and the eigenvalue can thus be written in terms of real values as $\eta = \nu \pm i\omega$. If any eigenvalue has a positive real part, ν, then any perturbations applied to the system grow exponentially with time and the system is considered unstable. Therefore, as a bifurcation point denotes a change in stability, a bifurcation occurs when the solution has at least one eigenvalue with real part equal to zero. In this work the type of bifurcation encountered was always of the Hopf type, therefore the bifurcation tracking methods employed were set up to locate and track a Hopf bifurcation, although for both methods an alternative formulation for tracking other simple bifurcations is possible. Hopf bifurcations occur when a complex pair of eigenvalues crosses the imaginary axis. This means that $\nu = 0$ and $\eta = \pm i\omega$. Two different methods were used to firstly locate and then to track a bifurcation. Both have a similar computational cost per iteration and should produce similar results.

4.1. *Augmented system*

The first bifurcation tracking method employed was a method known as an augmented system. This type of method involves solving a defining system that corresponds to the type of bifurcation that has been detected. This enlarges the standard system by augmenting equations that define the bifurcation. The solution of this larger system is a bifurcation point.

The set of equations that define the Hopf bifurcation are [Salinger *et al.*, 2005]

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{J}\mathbf{v}_1 + \omega\mathbf{v}_2 \\ \mathbf{J}\mathbf{v}_2 - \omega\mathbf{v}_1 \\ \phi \cdot \mathbf{v}_1 - 1 \\ \phi \cdot \mathbf{v}_2 \end{bmatrix} = 0. \tag{12}$$

This system solves for the five unknowns: the flow solution vector, $\mathbf{Y}$, the real and imaginary parts of the complex eigenvector, $\mathbf{v}_1$ and $\mathbf{v}_2$, the eigenvalue, ω, and the first parameter, λ. This involves expanding the system to size $3N + 2$. The first equation in system (12) specifies that the solution must be on the flow

solution branch, the second and third ensure that the point has a purely imaginary eigenvalue and the final two fix the phase and amplitude of the eigenvectors. Different formulations of the last two equations have been used in other papers [Griewank & Reddien, 1983; Poliashenko & Aidun, 1995; Wriggers & Simo, 1990]. Applying Newton's method to these equations gives

$$\begin{bmatrix} \mathbf{J} & \mathbf{0} & \mathbf{0} & 0 & \frac{\partial \mathbf{R}}{\partial \lambda} \\ \frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \mathbf{Y}} & \mathbf{J} & \omega\mathbf{I} & \mathbf{v}_2 & \frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \lambda} \\ \frac{\partial \mathbf{J}\mathbf{v}_2}{\partial \mathbf{Y}} & -\omega\mathbf{I} & \mathbf{J} & -\mathbf{v}_1 & \frac{\partial \mathbf{J}\mathbf{v}_2}{\partial \lambda} \\ \mathbf{0} & \phi^T & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \phi^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{Y} \\ \Delta\mathbf{v}_1 \\ \Delta\mathbf{v}_2 \\ \Delta\omega \\ \Delta\lambda \end{bmatrix} = - \begin{bmatrix} \mathbf{R} \\ \mathbf{J}\mathbf{v}_1 + \omega\mathbf{v}_2 \\ \mathbf{J}\mathbf{v}_2 - \omega\mathbf{v}_1 \\ \phi \cdot \mathbf{v}_1 - 1 \\ \phi \cdot \mathbf{v}_2 \end{bmatrix} \tag{13}$$

When dealing with large systems, such as in this case, increasing the size of the system from N to $3N + 2$ can be prohibitive. Some of the computational burden can be removed by using a bordering method as employed by LOCA [Salinger *et al.*, 2005]. The solution algorithm for one iteration is then given by:

$$\mathbf{J}\mathbf{a} = -\mathbf{R}, \tag{14}$$

$$\mathbf{J}\mathbf{b} = -\frac{\partial \mathbf{R}}{\partial \lambda} \tag{15}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{I} \\ -\omega\mathbf{I} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{v}_2 \\ -\mathbf{v}_1 \end{bmatrix} \tag{16}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{I} \\ -\omega\mathbf{I} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \mathbf{Y}}\mathbf{a} \\ -\frac{\partial \mathbf{J}\mathbf{v}_2}{\partial \mathbf{Y}}\mathbf{a} \end{bmatrix} \tag{17}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{I} \\ -\omega\mathbf{I} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} -\frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \mathbf{Y}}\mathbf{b} - \frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \lambda} \\ -\frac{\partial \mathbf{J}\mathbf{v}_2}{\partial \mathbf{Y}}\mathbf{b} - \frac{\partial \mathbf{J}\mathbf{v}_2}{\partial \lambda} \end{bmatrix} \tag{18}$$

$$\Delta\lambda = \frac{(\phi \cdot \mathbf{c})(\phi \cdot \mathbf{f}) - (\phi \cdot \mathbf{e})(\phi \cdot \mathbf{d}) + (\phi \cdot \mathbf{d})}{(\phi \cdot \mathbf{d})(\phi \cdot \mathbf{g}) - (\phi \cdot \mathbf{c})(\phi \cdot \mathbf{h})} \tag{19}$$

$$\Delta\omega = \frac{(\phi \cdot \mathbf{h})\Delta\lambda + (\phi \cdot \mathbf{f})}{\phi \cdot \mathbf{d}} \tag{20}$$

$$\Delta\mathbf{Y} = \mathbf{a} + \Delta\lambda\mathbf{b} \tag{21}$$

$$\Delta\mathbf{v}_1 = \mathbf{e} + \Delta\lambda\mathbf{g} - \Delta\omega\mathbf{c} - \mathbf{v}_1 \tag{22}$$

$$\Delta\mathbf{v}_2 = \mathbf{f} + \Delta\lambda\mathbf{h} - \Delta\omega\mathbf{d} - \mathbf{v}_2 \tag{23}$$

This solution algorithm requires two solves involving $\mathbf{J}$, an $N \times N$ matrix, and three solves involving $\begin{bmatrix} \mathbf{J} & \omega\mathbf{I} \\ -\omega\mathbf{I} & \mathbf{J} \end{bmatrix}$ a $2N \times 2N$ matrix. These two matrices are sparse so it is possible to employ the same solver as used for the continuation method. The vector ϕ can be any reasonable vector, in this work ϕ was chosen to be a vector of ones. The derivatives that appear on the right hand side of Eq. (17) and Eq. (18)

are calculated using second order differences. Equations (24)-(26) show the derivatives involving $\mathbf{v}_1$. The derivatives involving $\mathbf{v}_2$ are calculated in a similar manner.

$$\frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \mathbf{Y}}\mathbf{a} = \frac{\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1+\epsilon_1\mathbf{a},\lambda)-\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1-\epsilon_1\mathbf{a},\lambda)-\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1+\epsilon_1\mathbf{a},\lambda)+\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1-\epsilon_1\mathbf{a},\lambda)}{4\sigma_1\epsilon_1} \tag{24}$$

$$\frac{\partial \mathbf{J}\mathbf{v}_1}{\partial \mathbf{Y}}\mathbf{b} = \frac{\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1+\epsilon_2\mathbf{b},\lambda)-\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1-\epsilon_2\mathbf{b},\lambda)-\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1+\epsilon_2\mathbf{b},\lambda)+\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1-\epsilon_2\mathbf{b},\lambda)}{4\sigma_1\epsilon_2} \tag{25}$$

$$\frac{\partial \mathbf{J}}{\partial \lambda}\mathbf{v}_1 = \frac{\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1,\lambda+\epsilon_3)-\mathbf{R}(\mathbf{Y}+\sigma_1\mathbf{v}_1,\lambda-\epsilon_3)-\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1,\lambda+\epsilon_3)+\mathbf{R}(\mathbf{Y}-\sigma_1\mathbf{v}_1,\lambda-\epsilon_3)}{4\sigma_1\epsilon_3} \tag{26}$$

The perturbations, ϵ_1, ϵ_2, ϵ_3, and σ_1 have an impact on the efficiency and robustness of the algorithm. In this work they are calculated using the following equations as they were found to perform well in the work of Salinger *et al.* [2005]

$$\epsilon_1 = \delta\left(\frac{\|\mathbf{Y}\|}{\|\mathbf{a}\|}+\delta\right) \tag{27}$$

$$\epsilon_2 = \delta\left(\frac{\|\mathbf{Y}\|}{\|\mathbf{b}\|}+\delta\right) \tag{28}$$

$$\epsilon_3 = \delta\left(|\lambda|+\delta\right) \tag{29}$$

$$\sigma_1 = \delta\left(\frac{\|\mathbf{Y}\|}{\|\mathbf{v}_1\|}+\delta\right) \tag{30}$$

The value of δ can be chosen by the user to be any small number. A value of 1×10^{-6} proved to work well for these types of problems [Huntley, 2015]. This method requires good initial approximations for all the unknowns. These are provided by the continuation method, which is employed until a bifurcation has been detected and the eigenvalues and eigenvectors are calculated using the eigenvalue calculation method described in Sec. 3.

4.2. *Test function*

An alternative way of locating a bifurcation is by using a test function. Test functions are continuous around the bifurcation point and have the condition that at a bifurcation point the solution is zero. There are a wide range of test functions available and although they solve the same system the choice of test function is dependent upon the problem size and implementation [Govaerts, 1991]. The most common test function for Hopf bifurcations works by noting that at a Hopf bifurcation the matrix $\mathbf{J}^2+\omega^2\mathbf{I}$ has rank defect 2. This method was considered initially, however although the matrix $\mathbf{J}^2+\omega^2\mathbf{I}$ has the same dimensions as $\mathbf{J}$ it is a lot more dense, which slows down the computation considerably. Therefore an alternative method was implemented that works on the basis that at a Hopf bifurcation point the matrix $\mathbf{J}-i\omega\mathbf{I}$ is singular. This matrix is then appended with elements to make it non-singular. This gives a bordered matrix of

$$\mathbf{M} = \begin{bmatrix} \mathbf{J}-i\omega\mathbf{I} & \mathbf{W} \\ \mathbf{V}^T & 0 \end{bmatrix} \tag{31}$$

Here $\mathbf{V}$ is the right complex eigenvector, where $\mathbf{V}=\mathbf{v}_1+i\mathbf{v}_2$ and $\mathbf{W}$ is the left complex eigenvector, $\mathbf{W}=\mathbf{w}_1+i\mathbf{w}_2$. This system can be split into real and imaginary components. This increases the size of

the system from $N+1$ to $2N+2$ but allows a real-valued solver to be applied to the system. The bordered matrix written in terms of real-valued variables is

$$\mathbf{M} = \begin{bmatrix} \mathbf{J} & \omega\mathbf{I} & \mathbf{w}_1 & -\mathbf{w}_2 \\ -\omega\mathbf{I} & \mathbf{J} & \mathbf{w}_2 & \mathbf{w}_1 \\ \mathbf{v}_1^T & -\mathbf{v}_2^T & 0 & 0 \\ \mathbf{v}_2^T & \mathbf{v}_1^T & 0 & 0 \end{bmatrix} \tag{32}$$

This matrix is then used to define the test function, which in this case has two components (h_1 and h_2), defined by

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{I} & \mathbf{w}_1 & -\mathbf{w}_2 \\ -\omega\mathbf{I} & \mathbf{J} & \mathbf{w}_2 & \mathbf{w}_1 \\ \mathbf{v}_1^T & -\mathbf{v}_2^T & 0 & 0 \\ \mathbf{v}_2^T & \mathbf{v}_1^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ 1 \\ 0 \end{bmatrix} \tag{33}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are ... In order for Eq. (32) to have rank defect 1 it must be true that $h_1 = h_2 = 0$. This means the defining system $\mathbf{s} = (\mathbf{Y}, \lambda, \omega)$ is then given by

$$\mathbf{R} = 0, \tag{34}$$

$$h_1 = 0, \tag{35}$$

$$h_2 = 0. \tag{36}$$

A single Newton iteration then solves the system

$$\begin{bmatrix} \mathbf{J} & \frac{\partial \mathbf{R}}{\partial \lambda} & \mathbf{0} \\ \frac{\partial h_1}{\partial \mathbf{Y}} & \frac{\partial h_1}{\partial \lambda} & \frac{\partial h_1}{\partial \omega} \\ \frac{\partial h_2}{\partial \mathbf{Y}} & \frac{\partial h_2}{\partial \lambda} & \frac{\partial h_2}{\partial \omega} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{Y} \\ \Delta\lambda \\ \Delta\omega \end{bmatrix} = - \begin{bmatrix} \mathbf{R} \\ h_1 \\ h_2 \end{bmatrix} \tag{37}$$

The partial derivatives of h_1 and h_2 with respect to $\mathbf{s}$ can be computed efficiently by considering the solution of the transpose of the system

$$\begin{bmatrix} \mathbf{q}_1^T & \mathbf{q}_2^T & g_1 & g_2 \\ \mathbf{q}_3^T & \mathbf{q}_4^T & g_3 & g_4 \end{bmatrix} \mathbf{M} = \begin{bmatrix} \mathbf{0}^T & \mathbf{0}^T & 1 & 0 \\ \mathbf{0}^T & \mathbf{0}^T & 0 & 1 \end{bmatrix} \tag{38}$$

By differentiating Eq. (33) and substituting Eq. (38) into the result, the derivatives with respect to $\mathbf{Y}$, λ, and ω are shown to be:

$$\frac{\partial h_1}{\partial \mathbf{Y}} = -\mathbf{q}_1^T \frac{\partial \mathbf{J}}{\partial \mathbf{Y}} \mathbf{p}_1 - \mathbf{q}_2^T \frac{\partial \mathbf{J}}{\partial \mathbf{Y}} \mathbf{p}_2 \tag{39}$$

$$\frac{\partial h_2}{\partial \mathbf{Y}} = -\mathbf{q}_3^T \frac{\partial \mathbf{J}}{\partial \mathbf{Y}} \mathbf{p}_1 - \mathbf{q}_4^T \frac{\partial \mathbf{J}}{\partial \mathbf{Y}} \mathbf{p}_2 \tag{40}$$

$$\frac{\partial h_1}{\partial \lambda} = -\mathbf{q}_1^T \frac{\partial \mathbf{J}}{\partial \lambda} \mathbf{p}_1 - \mathbf{q}_2^T \frac{\partial \mathbf{J}}{\partial \lambda} \mathbf{p}_2 \tag{41}$$

$$\frac{\partial h_2}{\partial \lambda} = -\mathbf{q}_3^T \frac{\partial \mathbf{J}}{\partial \lambda} \mathbf{p}_1 - \mathbf{q}_4^T \frac{\partial \mathbf{J}}{\partial \lambda} \mathbf{p}_2 \tag{42}$$

$$\frac{\partial h_1}{\partial \omega} = -\mathbf{q}_1^T \mathbf{p}_2 + \mathbf{q}_2^T \mathbf{p}_1 \tag{43}$$

$$\frac{\partial h_2}{\partial \omega} = -\mathbf{q}_3^T \mathbf{p}_2 + \mathbf{q}_4^T \mathbf{p}_1 \tag{44}$$

where $\frac{\partial \mathbf{J}}{\partial \mathbf{Y}}$ is the Hessian matrix and $\frac{\partial \mathbf{J}}{\partial \lambda}$ is the derivative of the Jacobian matrix, $\mathbf{J}$, with respect to the parameter, λ. For a two-dimensional system the Hessian matrix is a large three-dimensional system and its computation is expensive. However, in the current implementation the Hessian is only required when post-multiplied by two vectors, as in Eqs. (47) and (48) which allows it to be computed using a finite difference formulation such as that given in Eq. (24). The system was solved by a block elimination algorithm that requires only two solves of the matrix $\mathbf{J}$ as shown below.

$$\mathbf{J}\mathbf{a} = -\mathbf{R} \tag{45}$$

$$\mathbf{J}\mathbf{b} = \frac{\partial \mathbf{R}}{\partial \lambda} \tag{46}$$

$$\Delta\lambda = \frac{\left(\frac{\frac{\partial h_1}{\partial \omega}}{\frac{\partial h_2}{\partial \omega}} h_2 - h_1 - h_{1\mathbf{Y}}\mathbf{a} + \frac{\frac{\partial h_1}{\partial \omega}}{\frac{\partial h_2}{\partial \omega}} \frac{\partial h_2}{\partial \mathbf{Y}}\mathbf{a} \right)}{\left(\frac{\partial h_1}{\partial \mathbf{Y}}\mathbf{b} + \frac{\partial h_1}{\partial \lambda} - \frac{\frac{\partial h_1}{\partial \omega}}{\frac{\partial h_2}{\partial \omega}} \left(\frac{\partial h_2}{\partial \mathbf{Y}}\mathbf{b} + \frac{\partial h_2}{\partial \lambda} \right) \right)} \tag{47}$$

$$\Delta\omega = \frac{-h_2 - \frac{\partial h_2}{\partial \mathbf{Y}}\mathbf{a} - \left(\frac{\partial h_2}{\partial \mathbf{Y}}\mathbf{b} + \frac{\partial h_2}{\partial \lambda} \right) \Delta\lambda}{\frac{\partial h_2}{\partial \omega}} \tag{48}$$

$$\Delta\mathbf{Y} = \mathbf{a} + \mathbf{b}\Delta\lambda \tag{49}$$

5. Results

The purpose of this work is to demonstrate the feasibility of bifurcation tracking methods for large systems such as those that are created when CFD is used to model high Reynolds number flows around an airfoil. Both the underlying CFD and the base continuation method have been verified for these types of problems (see [Wales *et al.*, 2012b,a]). The bifurcation tracking methods should be able to trace the path of a Hopf bifurcation as two parameters are changed. The aim of this study is to show the applications of this approach rather than compute any specific solutions. For this reason a typical geometry and mesh have been used as the starting point for all the calculations. This was a NACA 0012 airfoil with a structured 225×65 C-mesh shown in Fig. 1 . The parameters used to track the bifurcation included the flow parameters: angle of attack, Reynolds number and Mach number and the geometrical parameters: camber and thickness. When the bifurcation tracking process involved a geometrical parameter, this was used as the second continuation parameter. This meant that it could be held constant whilst the bifurcation tracking algorithm was being converged and the mesh only needed to be adapted between bifurcation solution points. The mesh adaptation was performed using a simple interpolation method. All the examples used the Spalart-Allmaras turbulence model detailed above.

5.1. *Bifurcation tracking of shock induced oscillations*

The onset of shock induced oscillations is associated with a Hopf bifurcation and, as such, continuation methods can be used to identify the parameter values where this unsteady flow regime begins. The original continuation method can be used to calculate rough values where this bifurcation occurs and the accuracy of the bifurcation point will depend on the step length used as it is unlikely that the step will give the exact parameter value where the bifurcation occurs. Furthermore, computing the stability boundary for a range of second parameters would require a continuation run to be performed for a selection of second parameter values. This method was used in [Wales, 2010]. Before this, the previous computational evaluation of stability used standard time integration and inspection of time histories to decide on the stability. Through the use of bifurcation tracking methods the stability boundary can be computed directly, giving the exact bifurcation point without the need for multiple continuation runs. This first test case examines the ability of the direct bifurcation tracking methods to compute this stability boundary.

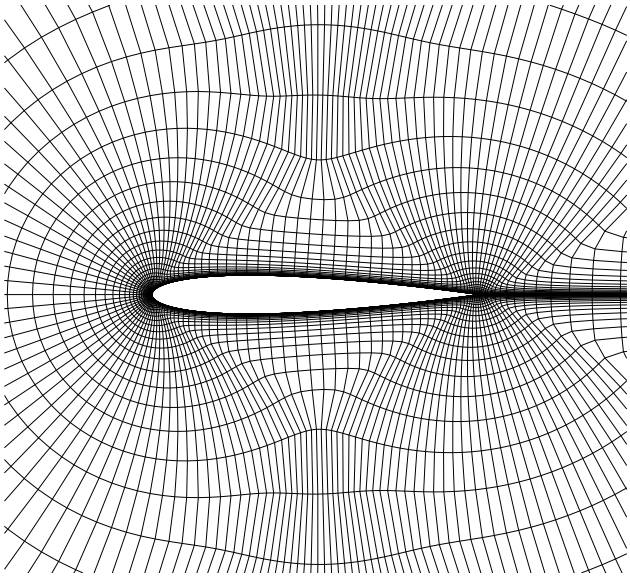

Fig. 1. Close up of mesh used for NACA 0012 computations.

The angle of attack was used as the first continuation parameter, $\lambda = \alpha$, and Mach number was used as the second continuation parameter, $\gamma = M$, with a constant Reynolds number of 1×10^7. Fig. 2 shows the predicted boundary of the flow stability in terms of angle of attack and Mach number. The required converged solution tolerance was set at 10^{-8}. This test case required more work as the step size for the Mach number had to be constantly monitored and adapted to enable both solution methods to converge.

Both methods show fairly good agreement but also fail to find any solutions in certain regions of the solution curve. The two methods differ most at the higher Mach numbers. The augmented system method looks likely to be correct whilst the solutions found using the test function are not entirely as expected. As the test function still managed to compute converged solutions but diverged from the expected stability boundary, this suggests that the test function may have located a different Hopf bifurcation and started to track that. There can be multiple Hopf bifurcations, both physical and numerical, in a given system and this number is also dependent on the CFD method used and how the problem is discretized. Furthermore, ARPACK struggles to compute any eigenvalues at the higher Mach numbers indicating that the problem is ill-conditioned. A mesh dependency study was not carried out for this particular test case although it has previously been reported that it is more important to have a finer grid than one that extends further into the free-stream [Iovnovich & Raveh, 2012] and the mesh used was chosen with this in mind. The CPU time taken for the methods for this test case are given in Table 5.1. This shows that there is not much difference in cost between the augmented system and test function. This is expected as both methods have a similar computational cost per iteration, took a similar number of iterations to converge onto each solution and computed a similar number of solutions.

5.2. *Bifurcation tracking in Reynolds number and angle of attack*

This test case investigates the performance of the bifurcation tracking methods at mapping the stability boundary in a parameter space defined by both angle of attack and Reynolds number. The first continuation parameter used was the angle of attack, $\lambda = \alpha$, whilst the second continuation parameter was Reynolds number, $\gamma = Re$. The tracking was performed for a range of Reynolds numbers between 5×10^6 and

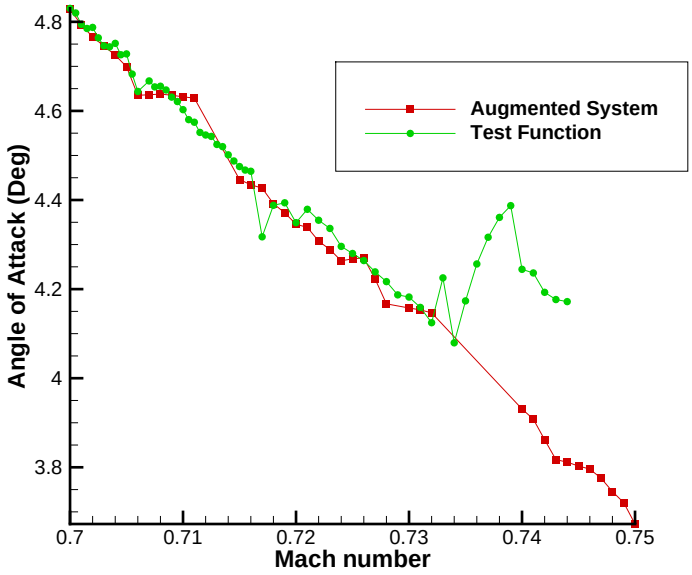

Fig. 2. Shock induced oscillations for a NACA 0012 airfoil $Re = 1 \times 10^7$.

Table 1. Time taken for the bifurcation tracking methods for shock induced oscillations.

Method	CPU Time (s)
Augmented System	3.06×10^5
Test Function	3.19×10^5

3.3×10^7 with an initial step size of 0.5×10^6. The Mach number was fixed at 0.3. This problem was set up to replicate that of the Reynolds number continuation test case in Wales [2010], which mapped the stability boundary by performing continuation at fixed Reynolds numbers and angles of attack.

Both methods were converged to a tolerance of 10^{-8}. Both tracking methods were initialized by performing continuation in angle of attack at a fixed Reynolds number of 5×10^6, using the arclength continuation method. The continuation was performed until the eigenvalue solver had detected that a bifurcation had occurred. Both methods generally show good agreement throughout the range of parameter values, as shown in Fig. 3. Some bifurcation points are computed at the same parameter values, whilst others have a small discrepancy, most noticeably at the lower Reynolds numbers. This could again be due to multiple Hopf bifurcations, numerical or physical, being present in this range of parameters.

The CPU time taken for the methods for this test case are given in Table 5.2. Again, the cost of the augmented system and test function method are similar, as would be expected given the size of the system is the same.

The following results will present bifurcation tracking with a shape parameter. This reflects the idea of how a change in design could move the unwanted bifurcation outside the operating range.

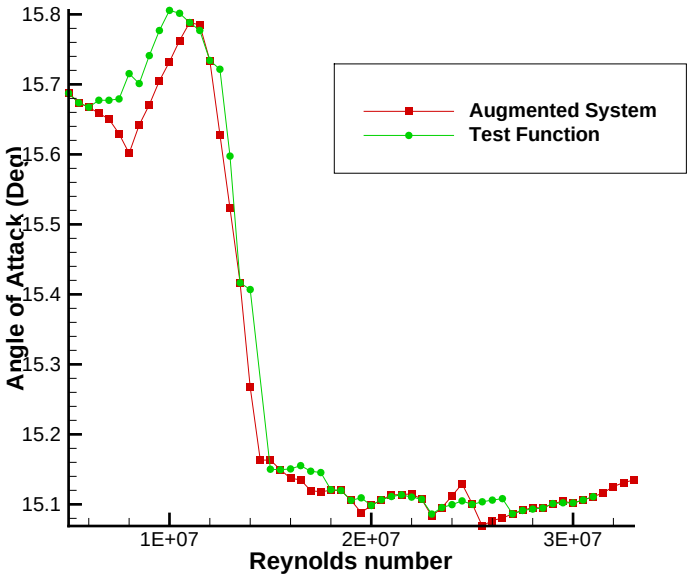

Fig. 3. Stability limit of Reynolds number and angle of attack for a NACA 0012 airfoil $M = 0.3$.

Table 2. Time taken for the bifurcation tracking methods for Reynolds number and angle of attack.

Method	CPU Time (s)
Augmented System	3.18×10^5
Test Function	3.22×10^5

5.3. *Bifurcation tracking in camber and angle of attack*

This test case compares the performance of the bifurcation tracking methods at mapping the stability boundary in a parameter space defined by both angle of attack and camber. The results were carried out for the flow past a NACA 4-digit airfoil with a constant thickness of 12% chord, with the initial airfoil being a NACA 0012. The first continuation parameter used was the angle of attack whilst the second continuation parameter was camber. The tracking was performed for a range of cambers between 0% chord and 0.4% chord with the maximum camber location being kept constant at 40% chord. The Mach number was fixed at 0.7 and the Reynolds number was fixed at 1×10^7. The required converged solution tolerance was set at 10^{-8} for both methods.

The results, shown in Fig. 4, show a fairly good agreement between the test function and augmented system method. Tracking a bifurcation in camber proved difficult for all the methods with neither method being able to track the bifurcation beyond a camber of 0.4% chord. The test function method also required a much smaller step size in camber in order to be able to continue the solution curve. The step size for the augmented system was 0.04% chord whilst the test function needed the step size to be reduced to 0.02% chord. Furthermore, the test function only managed to compute solutions up to 0.2% chord, which is half the range that the augmented system was able to achieve. Part of the reason for the lesser performance of

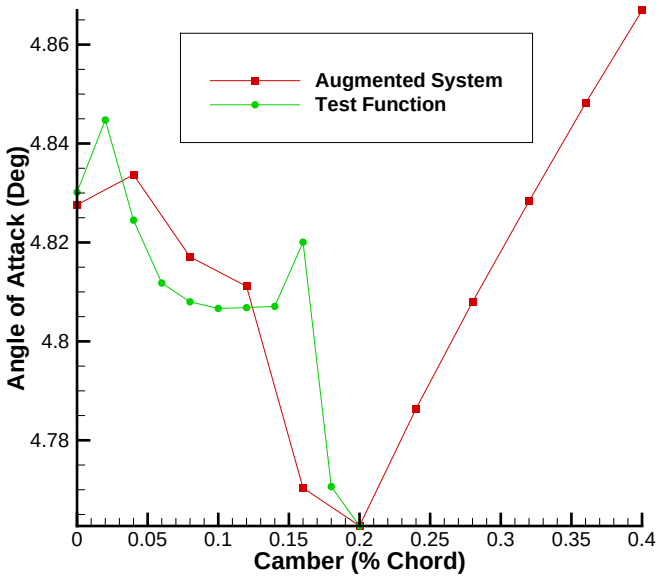

Fig. 4. Stability limit of camber and angle of attack for a NACA 4-digit airfoil with a thickness of 12% of the chord at $M = 0.7$ and $Re = 1 \times 10^7$.

both methods in this test case could be due to the simple implementation of mesh adaptation.

5.4. *Bifurcation tracking in thickness and angle of attack*

This test case looks at the performance of the bifurcation tracking methods at mapping the stability boundary in a parameter space defined by both angle of attack and thickness. The results were carried out for the flow past a symmetric NACA 4-digit airfoil, again starting with the NACA 0012 airfoil.

The first continuation parameter used was angle of attack whilst the second continuation parameter was thickness. The tracking was performed for a range of thicknesses between 12% chord and 15% chord with a constant step size of 0.1% chord. The Mach number was fixed at 0.7 and the Reynolds number was fixed at 1×10^7. As with all the previous test cases, the required converged solution tolerance was set at 10^{-8}. Both tracking methods were initialized by performing continuation in angle of attack at a thickness of 12% chord, giving the NACA 0012 airfoil, using the arclength continuation method. The continuation was performed until the eigenvalue solver had detected that a bifurcation had occurred. Again, the results, shown in Fig. 5, show a good agreement between the two methods. It shows that a thicker airfoil reduces the angle at which the flow solution becomes unstable, although this occurs at a much higher rate initially and then levels off towards the thickest airfoils computed.

5.5. *Bifurcation tracking in thickness and Mach number*

This test case compares the performance of the test function and augmented system bifurcation tracking methods at mapping the stability boundary in a parameter space defined by both Mach number and thickness. The first continuation parameter used was the Mach number whilst the second continuation parameter was thickness. The tracking was performed for a range of thicknesses between 12% chord and 15% chord with a constant step size of 0.25% chord. The angle of attack was fixed at 4.8° and the Reynolds number was fixed at 1×10^7. The required converged solution tolerance was set at 10^{-8}. Both tracking methods were initialized by performing continuation in Mach number at a thickness of 12% chord, giving the NACA 0012 airfoil, using the arclength continuation method. The tracking methods were implemented once the continuation method had detected that a bifurcation had occurred, which was achieved by computing

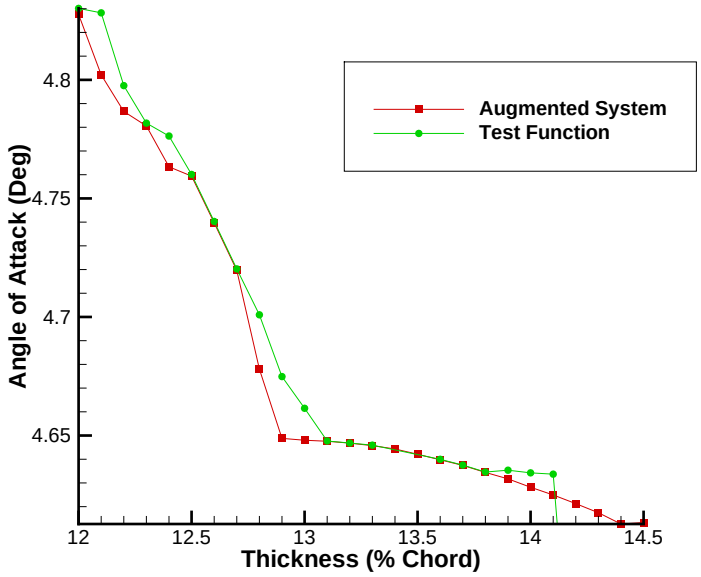

Fig. 5. Stability limit of thickness and angle of attack for a NACA 4-digit symmetric airfoil at $M = 0.7$ and $Re = 1 \times 10^7$.

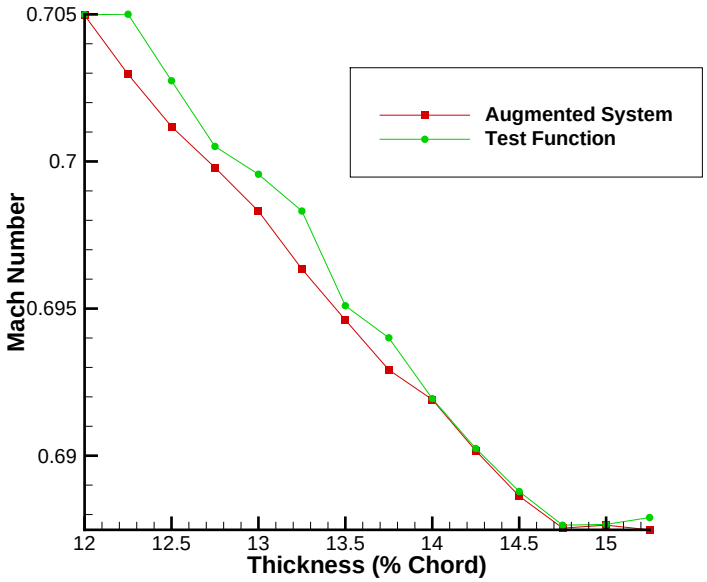

Fig. 6. Stability limit of thickness and Mach number for a NACA 4-digit symmetric airfoil at an angle of attack of 4.8° and $Re = 1 \times 10^7$.

the eigenvalues. The results, shown in Fig. 6, show a good agreement between the two methods especially towards the end of the tracking. This figure shows that as the thickness of the airfoil is increased the Mach number at which the flow solution becomes unstable is reduced.

Combining the results from bifurcation tracking in angle of attack and thickness at Mach 0.7 and with addition of further solutions at different fixed Mach numbers allows stability boundary surfaces to

be generated. The boundary shown in Fig. 7 was generated using the augmented system method, as this method was found to be the most robust and produce the most consistent results. This shows a range of angles, Mach numbers and thicknesses, for a constant Reynolds number of 1×10^7 where there is a change in stability due to a pair of complex eigenvalues crossing the imaginary axis. Anything above this surface as it is displayed gives an unstable solution, whilst anything below it is a stable flow solution. The three Figs. 7, 8 and 9 show the corresponding coefficients for lift, drag and moment as contours on the surface, respectively.

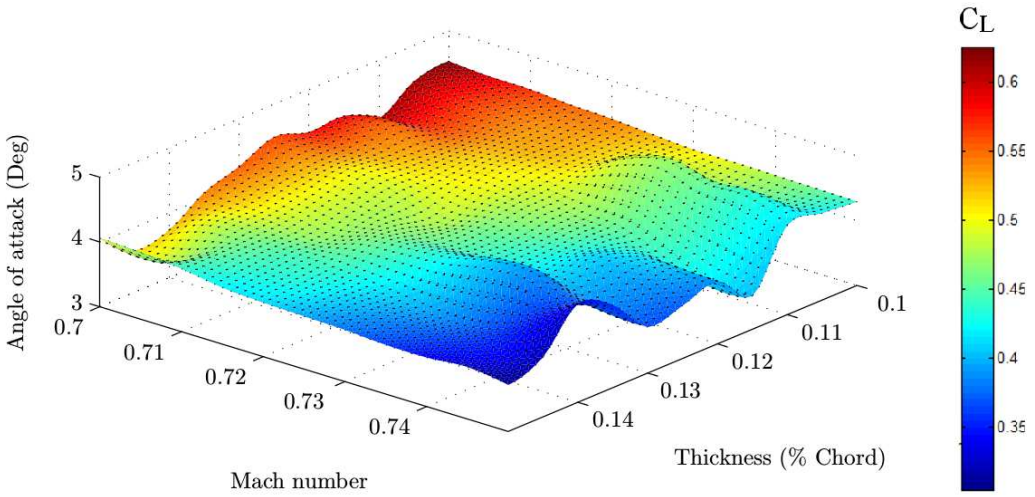

Fig. 7. Lift coefficient on the stability limit of thickness, Mach number and angle of attack for a NACA 4-digit airfoil with $Re = 1 \times 10^7$.

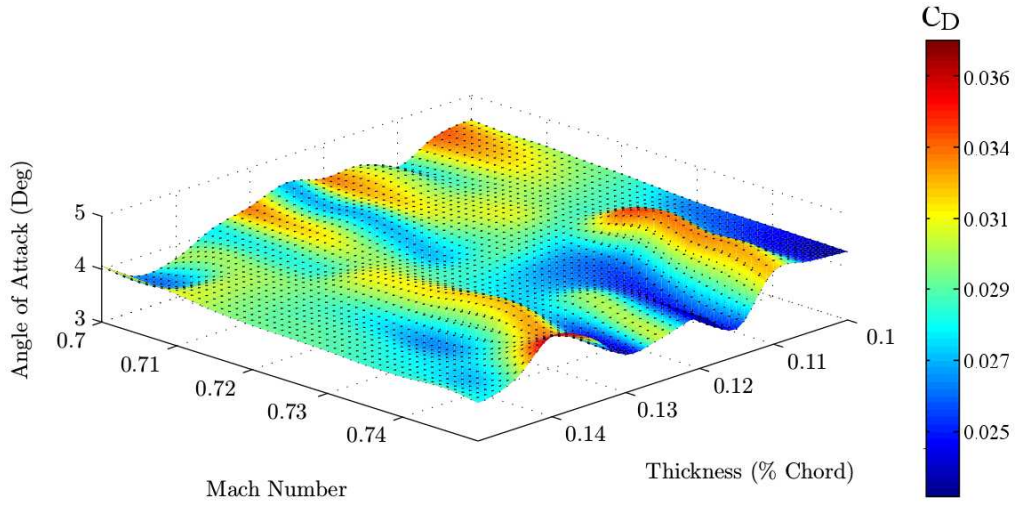

Fig. 8. Drag coefficient on the stability limit of thickness, Mach number and angle of attack for a NACA 4-digit airfoil with $Re = 1 \times 10^7$.

The last three test cases demonstrate that if applied in three dimensions these methods could be used to aid the design of the aircraft if applied in a optimisation type calculation.

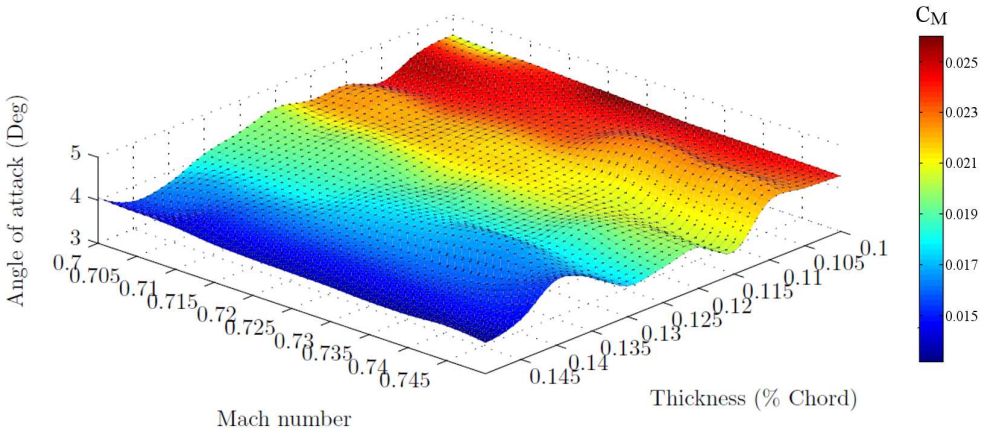

Fig. 9. Moment coefficient on the stability limit of thickness, Mach number and angle of attack for a NACA 4-digit airfoil with $Re = 1 \times 10^7$.

6. Conclusions

This paper shows that direct bifurcation tracking methods are a viable means to map stability boundaries of high Reynolds number flow around an airfoil as system parameters are varied. The parameters demonstrated in this paper include flow parameters: angle of attack, Mach number and Reynolds number and geometrical parameters: camber and thickness.

Although the bifurcation tracking methods have successfully been applied to a large system to predict the stability limit, they are still currently limited by the size of the problem they can handle. This could be overcome by solving the matrix system using an indirect method as opposed to the direct method currently implemented. This would lead to a larger range of problems that could be analysed but would require the development of a preconditioner suitable for this type of flow, which must have the properties of accelerating the indirect solver and being robust over a wide range of parameters.

This paper has used only the Spalart-Allmaras turbulence model, this has a direct bearing on the accuracy of the results but not on the effectiveness of the bifurcation tracking methods. Other turbulence models could be implemented. This may be especially beneficial for the cases where the two methods do not agree, to investigate whether these bifurcations are physical or numerical.

References

Badcock, K., Woodgate, M. & Richards, B. [2004] "Hopf bifurcation calculations for a symmetric airfoil in transonic flow," *AIAA Journal* **42**, 883–892.

Badcock, K., Woodgate, M. & Richards, B. [2005] "Direct aeroelastic bifurcation analysis of a symmetric wing based on euler equations," *Journal of Aircraft* **42**, 731–737.

Bindel, D., Friedman, M., Govaerts, W., Hughes, J. & Kuznetsov, Y. [2014] "Numerical computation of bifurcations in large equilibrium systems in matlab," *Journal of Computational and Applied Mathematics* **261**, 232–248.

Cliffe, K., Spence, A. & Tavener, S. [2001] "The numerical analysis of bifurcation problems with application to fluid mechanics," *Acta Numerica* **9**, 39–131.

Davis, T. [2004a] "Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.* **30**, 196–199.

Davis, T. [2004b] "A column pre-ordering strategy for the unsymmetric-pattern multifrontal method," *ACM Trans. Math. Softw.* **30**, 165–195.

Davis, T. & Duff, I. [1997] "An unsymmetric-pattern multifrontal method for sparse lu factorization," *SIAM Journal on Matrix Analysis and Applications* **18**, 140–158.

Davis, T. & Duff, I. [1999] "A combined unifrontal/multifrontal method for unsymmetric sparse matrices,"

ACM Trans. Math. Softw. **25**, 1–20.

Dhooge, A., Govaerts, W. & Kuznetsov, Y. A. [2003] "Matcont: A matlab package for numerical bifurcation analysis of odes," *ACM Trans. Math. Softw.* **29**, 141–164, doi:http://doi.acm.org/10.1145/779359.779362.

Dijkstra, H. A., Wubs, F. W., Cliffe, A. K., Doedel, E., Dragomirescu, I. F., Eckhardt, B., Gelfgat, A. Y., Hazel, A. L., Lucarini, V., Salinger11, A. G. *et al.* [2014] "Numerical bifurcation methods and their application to fluid dynamics: Analysis beyond simulation," *Commun. Comput. Phys.* , 1–45.

Doedel, E., Champneys, A., Fairgrieve, T., Kuznetsov, Y., Sandstede, B. & Wang, X. [1986] "AUTO97: Continuation and bifurcation software for ordinary differential equations (with HomCont)," *Computer Science, Concordia University, Montreal, Canada* .

Doedel, E., Keller, H. & Kernevez, J. [1991a] "Numerical analysis and control of bifurcation problems, part II," *Int. J. of Bif. and Chaos* **3**, 493–520.

Doedel, E., Keller, H. & Kernevez, J. [1991b] "Numerical analysis and control of bifurcation problems:(I) Bifurcation in finite dimensions," *Int. J. Bifurcation and Chaos* **1**, 493–520.

Govaerts, W. [1991] "Stable solvers and block elimination for bordered systems." *SIAM J. Matrix Anal. Appl.* **12**, 459,483.

Griewank, A. & Reddien, G. [1983] "The calculation of hopf points by a direct method," *IMA J. Numerical Analysis* **3**, 295–303.

Huntley, S. [2015] "Bifurcation tracking and continuation methods for high reynolds number compressible flows," PhD thesis, University of Bristol.

Iovnovich, M. & Raveh, D. [2012] "Reynolds-averaged navierstokes study of the shock-buffet instability mechanism," *AIAA Journal* **50**, 880–890.

Lehoucq, R. & Salinger, A. [2001] "Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers," *International Journal for Numerical Methods in Fluids* **36**, 309–327.

Lehoucq, R., Sorensen, D. & Yang, C. [1998] *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods* (Siam).

Lust, K., Roose, D., Spence, A. & Champneys, A. [1998] "An adaptive newton-picard algorithm with subspace iteration for computing periodic solutions," *SIAM J. Sci. Comput.* **19**, 1188–1209.

Poliashenko, M. & Aidun, C. K. [1995] "A direct method for computation of simple bifurcations," *J. Comput. Physics* , 246–260.

Rheinboldt, W. & Burkardt, J. [1983] "Algorithm 596: a program for a locally parameterized," *ACM Trans. Math. Softw.* **9**, 236–241, doi:http://doi.acm.org/10.1145/357456.357461.

Rumsey, C. & Ying, S. [2002] "Prediction of high lift: review of present CFD capability," *Progress in Aerospace Sciences* **38**, 145–180.

Salinger, A., Burroughs, E., Pawlowski, R., Phipps, E. & Romero, L. [2005] "Bifurcation tracking algorithms and software for large scale applications," *Int. J. Bifurcation and Chaos* **15**, 1015–1032.

Salinger, A., Lehoucq, R., Pawlowski, R. & Shadid, J. [2002] "Computational bifurcation and stability studies of the 8: 1 thermal cavity problem," *International Journal for Numerical Methods in Fluids* **40**, 1059–1073.

Spalart, P. & Allmaras, S. [1994] "A one equation turbulence model for aerodynamic flows," *Recherche Aerospatiale French edition* , 5–5.

Vassberg, J., Tinoco, E., Mani, M., Brodersen, O., Eisfeld, B., Wahls, R., Morrison, J., Zickuhr, T., Laflin, K. & Mavriplis, D. [2008] "Abridged summary of the third AIAA computational fluid dynamics drag prediction workshop," *Journal of aircraft* **45**, 781.

Voss, H. [2007] "A new justification of the jacobidavidson method for large eigenproblems," *Linear Algebra and its Applications* **424**, 448–455.

Wales, C. [2010] "Continuation methods for high reynolds number compressible flows," PhD thesis, University Of Bristol.

Wales, C., Gaitonde, A. & Jones, D. [2012a] "An initial study of the flow around an aerofoil at high reynolds nnumber using continuation," *International Journal of Bifurcation and Chaos* **22**.

Wales, C., Gaitonde, A. L., Jones, D. P., Avitabile, D. & Champneys, A. R. [2012b] "Numerical continuation of high reynolds number external flows," *International Journal for Numerical Methods in Fluids* **68**,

135–159.

Woodgate, M. & Badcock, K. [2007] "Fast prediction of transonic aeroelastic stability and limit cycles," *AIAA Journal* **45**, 1370–1381.

Wriggers, P. & Simo, J. [1990] "A general procedure for the direct computation of turning and bifurcation points," *Int. J. for Numerical Methods in Engineering* **30**, 155–176.