

Numerical continuation of high Reynolds number external flows

C. Wales^{1,*}, A. L. Gaitonde¹, D. P. Jones¹, D. Avitabile² and A. R. Champneys²

¹*Aerospace Engineering Department, University of Bristol, Queen's Building, University Walk, BS8 1TR, U.K.*

²*Department of Engineering Mathematics, University of Bristol, Queen's Building, University Walk, BS8 1TR, U.K.*

SUMMARY

A numerical continuation method for the compressible Reynolds-Averaged Navier–Stokes equation with the Spalart–Allmaras turbulence model is presented and applied to the flow around a 2D airfoil. Using continuation methods it is possible to study the steady flow states of a system as a parameter such as angle of attack is varied. This approach allows unstable solutions to be calculated, which are important for understanding the nonlinear dynamics of the system. Furthermore, this method can be used to find any multivalued solutions that exist at a single parameter value. The eigenvalues of the system are calculated using the Cayley transform to precondition the eigenvalue solver ARPACK. The eigenvalues are important as they show the stability of the solutions as well as accurately detect parameter values at which bifurcations take place. Copyright © 2010 John Wiley & Sons, Ltd.

Received 24 September 2009; Revised 22 July 2010; Accepted 24 October 2010

KEY WORDS: continuation; Newton; Navier–Stokes; bifurcation; eigenvalues; Spalart–Allmaras

1. INTRODUCTION

Understanding the nonlinearities that arise in aerodynamic flows as a function of various parameters, such as angle of attack, Mach number and Reynolds number, is important for safety and increasing performance by expanding the flight envelope. These nonlinearities can cause the change from steady to unsteady flow or give rise to static hysteresis, where the values of lift, drag and pitching moment depend on the history of motion. The resulting knowledge has applications in a number of areas. First, identification of multivalued aerodynamic behaviour is important as the resulting widely different values of lift, and lift to drag ratio, could affect recovery from stall and/or a spin. Second, parameter values beyond which stable steady solutions are unobtainable are important in aeroelastic calculations. Finally, attempts to produce reduced-order models of complex unsteady aerodynamic stall behaviour require the dynamics of that behaviour to be first identified, including the unstable portion of the steady flow branch.

There are two main numerical approaches to gaining insight into the possible solutions of a nonlinear system as the parameters on which it depends are varied. The first is to carry out time accurate simulations at a range of fixed parameters. This is the main method used in computational fluid dynamics. Mittal and Saxena [1] conducted a numerical study using a finite element method to predict the static hysteresis around an NACA 0012 airfoil, using an incompressible solver coupled with the Baldwin–Lomax turbulence model. At higher angles of attack, results were unsteady and hence the time average of these was plotted as the solution. Such an approach is usually only able to identify stable solution branches. The unstable solution that separates the two branches was not

*Correspondence to: C. Wales, Aerospace Engineering Department, University of Bristol, Queen's Building, University Walk, BS8 1TR, U.K.

†E-mail: chris.wales@bristol.ac.uk

calculated. The alternative approach is to discretize the problem to produce a system of nonlinear equations and then use methods from nonlinear analysis to compute solution paths as parameters vary, for example see Keller [2]. This method has the advantage that it can calculate the unstable solutions that are just as important as the stable solutions in understanding the overall dynamics of the system. This latter approach has been adopted in this study and used to investigate the behaviour of the steady Navier–Stokes equations.

There are several existing software packages available for carrying out the analysis of nonlinear equations, through path following, such as AUTO [3–5], PITCON [6] and MATCONT [7]. However, these packages tend to be designed for low-dimensional systems with dynamics dependent on multiple parameters. As a result they use direct solvers for dense matrices that make them unsuitable for analysing high-dimensional systems of equations that result from the discretization of the Navier–Stokes equations. With improvements in the computational power of modern processors and increased memory capacity, it has recently become possible to extend existing nonlinear analysis methods to analyse the Navier–Stokes equations. Previous investigations into viscous flows have mainly been on the Taylor–Couette-type flows [8, 9] with Reynolds numbers of order 4000. The software library LOCA has been used to study incompressible flow problems with larger number of unknowns, such as the flow in a chemical vapour deposition reactor [10] or the thermal convection in a 8:1 differentially heated cavity [11]. However, to date, there have been no studies of external flows at the high Reynolds numbers typical of aerospace applications. This paper describes an investigation into Navier–Stokes solutions over a 2D airfoil at high Reynolds numbers $Re > 10^6$ as the angle of attack is varied.

2. BASELINE CFD SOLVER

Outlined below is a 2D unsteady compressible Reynolds-averaged Navier–Stokes code for structured grids that form the basis of the continuation algorithm described later. The equations are solved using a cell-centred approach with scalar dissipation. The Spalart–Allmaras turbulence model is used [12]. The resulting equations are then solved using the Newton method to calculate the steady-state solutions. The key elements of the solver, the discretization of the governing equations, the calculation of the system Jacobian and the solution of the resulting set of equations are described in the following subsections.

2.1. Governing equations

The governing equations are the 2D Navier–Stokes equations that are written in an integral form as

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{Q} dV + \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} ds = 0, \quad (1)$$

where \mathbf{Q} is the vector of conserved variables, given by

$$\mathbf{Q} = (\rho, \rho u, \rho v, \rho E)^T \quad (2)$$

in which ρ is the density, u and v are the velocity components in the x and y directions, respectively, and the total energy

$$E = e + \frac{1}{2}(u^2 + v^2),$$

where e is the internal energy.

The flux tensor \mathbf{F} is composed of inviscid and viscous contributions in the x and y directions.

$$\mathbf{F} = \mathbf{F}^I - \mathbf{F}^V,$$

where the inviscid tensor $\mathbf{F}^I = \mathbf{f}^I \mathbf{i}_x + \mathbf{g}^I \mathbf{i}_y$ and the fluxes are given by

$$\mathbf{f}^I = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ (\rho E + p)u + p \end{bmatrix}, \quad \mathbf{g}^I = \begin{bmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (\rho E + p)v + p \end{bmatrix},$$

and $\mathbf{i}_x, \mathbf{i}_y$ are the cartesian unit vectors. The pressure, p , is related to the energy e by the equation of state for a perfect gas:

$$p = (\gamma - 1) \left(\rho E - \frac{\rho(u^2 + v^2)}{2} \right). \quad (3)$$

The viscous tensor is $\mathbf{F}^V = \mathbf{f}^V \mathbf{i}_x + \mathbf{g}^V \mathbf{i}_y$, where fluxes \mathbf{f}^V and \mathbf{g}^V are given by

$$\mathbf{f}^V = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}, \quad \mathbf{g}^V = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{bmatrix}.$$

The components of the stress tensor and the heat flux vector are given by

$$\begin{aligned} \tau_{xx} &= \frac{2}{3Re} (\mu + \mu_t) \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right), \\ \tau_{yy} &= \frac{2}{3Re} (\mu + \mu_t) \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right), \\ \tau_{xy} = \tau_{yx} &= \frac{1}{Re} (\mu + \mu_t) \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \\ q_x &= - \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_T} \right) \frac{1}{[\gamma - 1] M_\infty^2 Re} \frac{\partial T}{\partial x}, \\ q_y &= - \left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_T} \right) \frac{1}{[\gamma - 1] M_\infty^2 Re} \frac{\partial T}{\partial y}, \end{aligned}$$

where Re is the Reynolds number, M_∞ is the freestream Mach number, Pr, Pr_T are the laminar and turbulent Prantl numbers and μ, μ_t are the laminar and turbulent eddy viscosities. The laminar viscosity is given by Sutherland's law:

$$\mu = T^{\frac{3}{2}} \left(\frac{1+S}{T+S} \right) \quad \text{where } S = \frac{110}{T_\infty} \quad (4)$$

and the turbulent eddy viscosity, μ_t , is given by the Spalart–Allmaras turbulence model.

2.2. Finite volume

In order to solve the RANS equations, they are spatially discretized using a standard cell-centred finite volume formulation, with scalar dissipation due to Jameson [13]. For a cell i, j of fixed volume $V_{i,j}$, the RANS equations now become

$$V_{i,j} \frac{d\mathbf{Q}_{i,j}}{dt} = -\mathbf{R}_{i,j}, \quad (5)$$

where $\mathbf{Q}_{i,j}$ is the cell-averaged state vector given at each cell centre and the residual for the cell $\mathbf{R}_{i,j}$ is the sum of the fluxes through each cell face with the i, j th entry of the form

$$\mathbf{R}_{i,j} = \sum_{\text{faces}} \mathbf{F}(\mathbf{Q}_{i,j}) \cdot \mathbf{n} ds, \quad (6)$$

where \mathbf{n} is the normal vector at the face and ds is the length of the cell face.

Including dissipation equation (5) can be written as

$$\frac{d\mathbf{Q}_{i,j}}{dt} = -\mathbf{R}_{i,j}^*, \quad (7)$$

where

$$\mathbf{R}_{i,j}^* = \frac{\mathbf{R}_{i,j} - \mathbf{D}_{i,j}}{V_{i,j}}. \quad (8)$$

The no-slip condition is enforced at the surface. For the density and pressure an adiabatic boundary condition is used; hence, there is zero density and pressure gradient through the surface. The far-field boundary conditions are set using the locally 1D Riemann invariants.

2.3. Spalart–Allmaras turbulence model

The Spalart–Allmaras one-equation turbulence model was used [12], which solves a transport equation for a variable \bar{v}

$$\frac{D\bar{v}}{Dt} = \frac{1}{\sigma} [\nabla \cdot ((v + \bar{v}) \nabla v) + c_{b2} (\nabla \bar{v})^2] + c_{b1} [1 - f_{t2}] \bar{S} \bar{v} - \left[c_{w1} f_w - \frac{c_{b1}}{k^2} f_{t2} \right] \left[\frac{\bar{v}}{d} \right]^2. \quad (9)$$

This is related to the turbulent eddy viscosity by the following equation:

$$\nu_t = \bar{v} f_{v1}, \quad (10)$$

where f_{v1} is given by

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\bar{v}}{v}$$

and v is the laminar viscosity. The production term is given by

$$\bar{S} = S + \frac{\bar{v}}{k^2 d^2} f_{v2}, \quad (11)$$

where S is the magnitude of the vorticity, k is the von Karman constant, d is the distance to the wall,

$$S = \left| \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right| \quad (12)$$

and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}. \quad (13)$$

The destruction function is given by

$$f_w(r) = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, \quad (14)$$

where

$$g = r + c_{w2}(r^6 - r) \quad (15)$$

and

$$r = \frac{\bar{v}}{\bar{S}k^2d^2}. \quad (16)$$

In implementing large values of r are truncated to a value of 10 [12]. The term f_{t2} ,

$$f_{t2} = c_{t3} e^{(-c_{t4}\chi^2)}, \quad (17)$$

is designed to make $\bar{\mu}=0$ a stable solution.

The constants in the equations are:

$$\begin{aligned} c_{b1} &= 0.1355, & c_{b2} &= 0.622, & \sigma &= \frac{2}{3}, & k &= 0.41, \\ c_{w1} &= \frac{c_{b1}}{k^2} + \frac{1+c_{b2}}{\sigma}, & c_{w2} &= 0.3, & c_{w3} &= 2, & c_{v1} &= 7.1, \\ c_{t1} &= 1.0, & c_{t2} &= 2.0, & c_{t3} &= 1.2, & c_{t4} &= 0.5. \end{aligned}$$

In the current implementation, the Spalart–Allmaras model is run fully turbulent. This means that the trip term f_{t1} is excluded. At the wall, the eddy viscosity is set to zero. At the outflow far-field boundary, first-order extrapolation is used and at the inflow boundary $3\mu \leq \bar{\mu} \leq 5\mu$ as recommended in [14] for fully turbulent simulations. This prevents the model from artificially transitioning at low Reynolds numbers as well as helping solution convergence at the leading edge stagnation point. The wall distance is calculated as the closest point on the surface to each cell centre. For steady calculations, the wall distances only need to be calculated once.

The $(\nabla\bar{v})^2$ term can cause numerical problems for implicit solvers [15]; hence, the diffusion term is rearranged using the following identity:

$$\nabla \cdot (\bar{v}\nabla\bar{v}) = (\nabla\bar{v})^2 + \bar{v}(\nabla^2\bar{v})$$

and after further simplification [15], leads to

$$\frac{D\bar{v}}{Dt} = \frac{1}{\sigma} [\nabla \cdot ((v + (1 + c_{b2})\bar{v})\nabla\bar{v}) - c_{b2}\bar{v}\nabla^2\bar{v}] + c_{b1} [1 - f_{t2}] \bar{S}\bar{v} - \left[c_{w1}f_w - \frac{c_{b1}}{k^2}f_{t2} \right] \left(\frac{\bar{v}}{d} \right)^2. \quad (18)$$

Multiplying the equation ρ and making use of the continuity equation, the SA model can be written in a conservative form using $\bar{\mu} = \rho\bar{v}$.

$$\frac{D\bar{\mu}}{Dt} = \frac{1}{\sigma} \left[\nabla \cdot ((\mu + (1 + c_{b2})\bar{\mu})\nabla\bar{v}) - c_{b2}\bar{\mu}\nabla^2\bar{v} \right] + c_{b1} [1 - f_{t2}] \bar{S}\bar{\mu} - \rho \left[c_{w1}f_w - \frac{c_{b1}}{k^2}f_{t2} \right] \left(\frac{\bar{v}}{d} \right)^2. \quad (19)$$

2.4. Spalart–Allmaras model with Edwards modification

The standard Spalart–Allmaras model can experience convergence issues close to the leading edge stagnation point. Edwards and Chandra [16] used a modified source term that improves the numerical behaviour of the model near the wall, giving better convergence characteristics. In the modified model, f_{t2} is set to zero and the following terms are redefined as

$$\begin{aligned} \bar{S} &= S^{\frac{1}{2}} \left[\frac{1}{\chi} + f_{v1} \right] \\ r &= \frac{\tanh \left[\frac{\bar{v}}{\bar{S}k^2d^2} \right]}{\tanh(1.0)} \end{aligned}$$

3. CONTINUATION ALGORITHM

Continuation methods can be used to numerically analyse the solution of the unsteady dynamical system equivalent of

$$\frac{d\mathbf{Q}}{dt} = -\mathbf{R}^*(\mathbf{Q}, \lambda), \quad (20)$$

which depends on one or more parameters λ and where \mathbf{R}^* is a smooth function. Continuation methods calculate a sequence of points that approximate a curve of the dependence of steady-state solutions to Equation (20) on a single parameter α . The analysis in this study is limited to the steady-state equilibrium, which is the solution to

$$\mathbf{R}^*(\mathbf{Q}, \lambda) = 0. \quad (21)$$

In this study the parameter investigated will be the angle of attack α , which is applied through the far-field boundary conditions as well as the continuation algorithm outlined below a standard Runge–Kutta time stepping scheme [17] is used to obtain starting solutions.

One approach to numerical continuation is the predictor corrector algorithm. This consists of two steps as shown in Figure 1. The first step generates an approximate solution $\bar{\mathbf{X}}^{n+1}$, where $\mathbf{X} = (\mathbf{Q}, \lambda)$, to the solution at a new equilibrium point. In this work angle of attack is used as the continuation parameter, $\lambda = \alpha$. This is called the predictor step. The second step starts from the predicted solution $\bar{\mathbf{X}}^{n+1}$ and uses an iterative solver, typically the Newton or a quasi-Newton method, to calculate the new equilibrium point, \mathbf{X}^{n+1} , to sufficient accuracy. This is called the corrector step.

3.1. Predictor

At least one equilibrium point at a fixed parameter $\alpha = \alpha_0$ is required before the continuation can commence. Once a solution has been found, \mathbf{X}^0 , with $\alpha = \alpha^0$, an approximate solution $\bar{\mathbf{X}}^1$ at a new parameter value is produced.

$$\bar{\mathbf{X}}^1 = (\mathbf{Q}^0, \alpha^1). \quad (22)$$

For subsequent points with solutions at \mathbf{X}^n and \mathbf{X}^{n-1} , an approximate solution $\bar{\mathbf{X}}^{n+1}$ at a new parameter value is produced as shown in Figure 1. Here a first-order polynomial is used to estimate the new solution

$$\bar{\mathbf{X}}^{n+1} = \mathbf{X}^n + h \frac{\mathbf{X}^n - \mathbf{X}^{n-1}}{\|\mathbf{X}^n - \mathbf{X}^{n-1}\|}, \quad (23)$$

where h is the step length. This is called a secant predictor.

The step length is an important factor in determining the overall efficiency and robustness of the continuation method and will be discussed later.

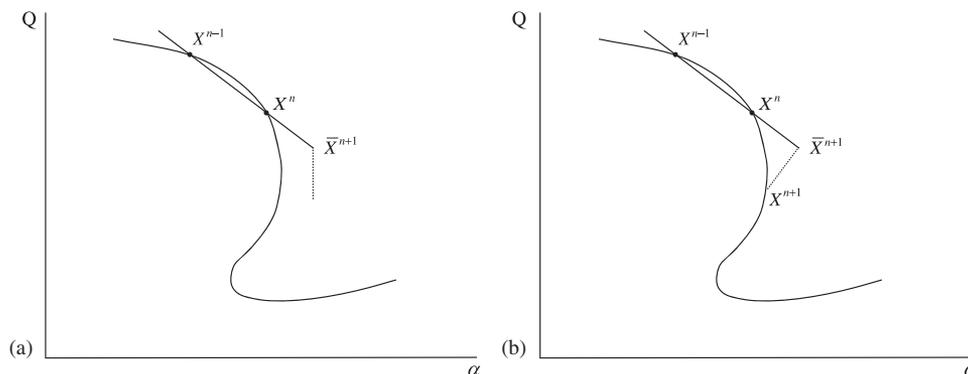


Figure 1. Comparison of parameterized and unparameterized Newton: (a) Newton and (b) pseudo-arclength.

3.2. Corrector

Once a predicted point $\bar{\mathbf{X}}^{n+1}$ has been found, the correct solution \mathbf{X}^{n+1} needs to be found to a specified accuracy. This can be done using the Newton method, to converge to a solution at the new parameter α^{n+1} . However in the presence of a fold, or turning points, in the solution curve as shown in Figure 1, where there is no solution close to $\bar{\mathbf{X}}^{n+1}$ at α^{n+1} , the Newton solver will fail to converge. The most robust method for overcoming this difficulty is the so-called pseudo-arclength continuation due to Keller [2]. Here both X and α are allowed to vary during the corrector step. Introducing an extra unknown means an extra equation must be added, the simplest being that the correction is perpendicular to the secant; see Figure 1. The solution curve is parameterized, introducing an extra parameter, γ , which allows each point along the curve to be identified. This means that the solution, \mathbf{X} , becomes a function of γ

$$\mathbf{X} = (\mathbf{Q}(\gamma), \alpha(\gamma)). \quad (24)$$

The parameterization does not need to have physical meaning and can be geometrically based. The parameter, γ , corresponds to a linearized arclength, between two solutions and results in the equation

$$g(\mathbf{Q}, \alpha, \gamma) = \left[\frac{\partial \mathbf{Q}}{\partial \gamma} \right] \Delta \mathbf{Q} + \left[\frac{\partial \alpha}{\partial \gamma} \right] \Delta \alpha - \Delta \gamma = 0, \quad (25)$$

where $\partial \mathbf{Q} / \partial \gamma$ and $\partial \alpha / \partial \gamma$ are approximated numerically using the previous two solutions. This is called the pseudo-arclength parameterization and constrains the Newton updates to a hyperplane normal to the direction taken for the predictor, as shown in Figure 1.

This leads to the following augmented system of equations, which are solved using the Newton method:

$$\begin{bmatrix} \frac{\partial \mathbf{R}^*}{\partial \mathbf{Q}} & \frac{\partial \mathbf{R}^*}{\partial \alpha} \\ \frac{\partial g}{\partial \mathbf{Q}} & \frac{\partial g}{\partial \alpha} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{Q} \\ \Delta \alpha \end{bmatrix} = \begin{bmatrix} -\mathbf{R}^* \\ -g \end{bmatrix}. \quad (26)$$

The partial derivatives $\partial \mathbf{R}^* / \partial \alpha$ are calculated numerically using the same finite difference method as outlined in Section 3.3.

Each corrector iteration requires the solution of the linear system

$$\begin{bmatrix} A & b \\ c^T & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (27)$$

where A is a sparse matrix, and b , c and d are typically dense. Bordered solvers allow the matrix in (27) to be decomposed in such a way that the structure of A is unchanged and existing solvers that exploit the sparsity of A can be used. Here the block elimination mixed (BEM) [18] bordering algorithm can be used as it produces accurate solutions even if A is nearly singular. This is an advantage because at (fold or other steady) bifurcation points, $A = \partial \mathbf{R}^* / \partial \mathbf{Q}$ becomes singular. The BEM algorithm is based on two different block LU factorizations, the Doolittle factorization and the Crout factorization. The BEM algorithm uses the Doolittle factorization to compute y , then

an iterative refinement step based on the Crout factorization is applied with the computed value y and $x=0$ as an initial guess. This leads to the following algorithm [18]:

$$\begin{aligned}
 &\text{Solve} && A^T w = c \\
 &\text{Compute} && \delta^* = d - w^T b \\
 &\text{Solve} && A v = b \\
 &\text{Compute} && \delta = d - c^T v \\
 &\text{Compute} && y_1 = \frac{g - w^T f}{\delta^*} \\
 &\text{Compute} && f_1 = f - b y_1 \\
 &\text{Compute} && g_1 = g - d y_1 \\
 &\text{Solve} && A \xi = f_1 \\
 &\text{Compute} && y_2 = \frac{g_1 - c^T \xi}{\delta} \\
 &\text{Compute} && x = \xi - v y_2 \\
 &\text{Compute} && y = y_1 + y_2
 \end{aligned}$$

As can be seen from the above, the BEM algorithm requires one solve with A^T and two with A . For details of the linear solver used, see Section 3.4.

3.3. Jacobian calculation

The Newton method requires an accurate Jacobian to obtain the desired quadratic performance. The Jacobian can be calculated analytically or numerically. Analytical Jacobians have the advantage that the derivatives are exactly calculated. However as the spatial discretization becomes more complex, deriving the exact analytical derivatives becomes harder and more prone to mistakes, both in derivation and coding. There are computer packages available that can carry out symbolic differentiation and produce code for calculating the analytical Jacobian. The resulting code can be large and difficult to optimize due to the failure of symbolic mathematics packages to simplify expressions. Numerical Jacobians on the other hand are simple to implement, even for complicated discretization schemes and can reuse existing code for the evaluation of the residual (8). However unlike an analytical Jacobian they are not exact and can take longer to compute, but with a careful choice of step size a numerical Jacobian can obtain the same convergence rate of the Newton scheme as an analytical Jacobian [19].

The l th column, the Jacobian of the discretized system of equations can be numerically approximated using the first term in a Taylor series

$$\frac{\partial \mathbf{R}^*}{\partial Q_l} = \frac{\mathbf{R}^*(\mathbf{Q} + \mathbf{e}_l h_l) - \mathbf{R}^*(\mathbf{Q})}{h_l}, \quad (28)$$

where h_l is a small perturbation in Q_l , see Equation (29), Q_l is the l th entry of \mathbf{Q} and \mathbf{e}_l is a vector of zeros with a 1 in the l th location. The error in the Jacobian is made up of two parts. The first is due to the truncation of the Taylor series and the second source of error comes from machine rounding when carrying out the finite difference calculation. These two components of the error have competing requirements on the size of the perturbation h_l . If h_l is too large, then the error due to the truncation of the Taylor series dominates and if h_l is too small then the error due to machine rounding dominates. The perturbation size, h_l , that we adopt is [20],

$$h_l = \sqrt{\eta} \max\{|Q_l|, \text{typ}\mathbf{Q}\} \text{sign}(Q_l), \quad (29)$$

where η is machine precision and $\text{typ}\mathbf{Q}$ is a typical value for the magnitude of \mathbf{Q} , over the entire flow field, given by

$$\text{typ}\mathbf{Q} = \frac{1}{2}(|\min(\mathbf{Q})| + |\max(\mathbf{Q})|).$$

This keeps the perturbation size small while preventing it becoming too small if Q_l is close to zero.

The residual only depends on the values in a few cells. This means that several values can be perturbed at the same time and multiple derivatives can be obtained for a single evaluation of the residual vector $\mathbf{R}^*(\mathbf{Q})$. The Jacobian calculation then becomes

$$\mathbf{w} = \mathbf{R}^* \left(\mathbf{Q} + \sum_{l \in \Gamma} h_l \mathbf{e}_l \right)$$

for each nonzero row k of a column $l \in \Gamma$, where Γ is a set of columns that can be calculated independently, due to the fact that Q_l is only in the stencil for a few of the equations \mathbf{R}_k . The Jacobian is approximated by

$$\frac{\partial \mathbf{R}_k^*}{\partial Q_l}(\mathbf{Q}) = \frac{w_k - \mathbf{R}_k^*(\mathbf{Q})}{h_l}.$$

The number of evaluations of $\mathbf{R}^*(\mathbf{Q})$ required to approximate the Jacobian is just the number of independent sets Γ that span all the columns. The number of sets Γ depends only on the size of the mesh and the stencil of the spatial discretization used. Calculating which variables are independent of each other is a graph colouring problem [21]. Using a simple graph colouring algorithm in Matlab as a preprocessor to calculate the independent columns, the Jacobian for a system of 71 040 unknowns can be produced in just 175 evaluations of the residual vector $\mathbf{R}^*(\mathbf{Q})$ with minimal or no modification to the existing code.

3.4. Linear solver

There are several different methods available for solving the linear system of equations that result from applying the Newton method to obtain the nonlinear solution of Equation (7). They can be split into direct solvers and iterative methods. Direct methods are computationally expensive in terms of run time and memory used. Offsetting this high cost is the fact that once the factorization has been done, resolving a system with a different right-hand side can be achieved with minimal extra computation. Iterative methods generally converge to the required solution in less computational time than direct methods. Matrix-free iterative methods also exist where the Jacobian vector product is calculated in a single step without the need to form the Jacobian, saving time and memory. However, in order to achieve rapid convergence a preconditioner is often required. Without a good preconditioner, the time taken to converge to the correct solution can approach the time taken using a direct method. Several preconditioning strategies for Jacobian-free iterative solvers are mentioned in [22]. These often require the formation of an approximate Jacobian. Although indirect solvers have been used to solve a wide range of flows, a direct solver has been used in this work as it is a more robust approach and the same factorization can be used to solve several systems, which is an advantage for calculating the eigenvalues of the system as discussed in Section 4. The chosen solver is a sparse unsymmetric multifrontal method called Umfpack [23–26].

3.5. Step length control

Step length control is one of the most important parts of the continuation algorithm. Too small a step size and computational effort is wasted by solving too many points; however, larger steps mean more Newton iterations are needed for the corrector step and if the step length is too large the predicted solution can be outside the radius of convergence and the corrector step will fail.

refer to asymptotically stable. It is important to note that here stability refers to the local stability. A system can be stable for a small perturbation, but unstable for larger perturbations. The local stability can be analysed using linearized stability, which provides an insight into the behaviour of the system close to the equilibrium.

The system of ODEs resulting from the spatial discretization of the Navier–Stokes equations can be written as

$$\frac{d\mathbf{Q}}{dt} = -\mathbf{R}^*(\mathbf{Q}) \quad (31)$$

taking the Taylor series expansion about the stationary solution $\mathbf{Q} = \mathbf{Q}^s$

$$\mathbf{R}^*(\mathbf{Q}^s + \tilde{q}) = \mathbf{R}^*(\mathbf{Q}^s) + \frac{\partial \mathbf{R}^*(\mathbf{Q}^s)}{\partial \mathbf{Q}} \tilde{q} + O(\tilde{q}^2), \quad (32)$$

where \tilde{q} is a small perturbation in \mathbf{Q} as defined in Equation (2). Assuming that the second-order and higher terms are negligible, the linearized stability equation becomes

$$\frac{d\tilde{q}}{dt} = -\frac{\partial \mathbf{R}^*(\mathbf{Q}^s)}{\partial \mathbf{Q}} \tilde{q} = A\tilde{q}. \quad (33)$$

Using the assumption that the solution takes the form

$$\tilde{q}(t) = e^{\Lambda t} \mathbf{v} \quad (34)$$

and substituting it into Equation (31) leads to the eigenvalue problem

$$(A - \mu I)\mathbf{v} = 0 \quad (35)$$

with eigenvalue Λ , eigenvector v and I is the identity matrix. The local linear stability depends on the sign of the real part of the eigenvalue Λ . If the eigenvalues all have negative real parts, then the solution is stable. If one or more of the eigenvalues has a positive real part, then the solution is unstable.

For systems where A is large, such as those arising from the discretization of the Navier–Stokes equations, the cost of calculating all the eigenvalues using the QR method is prohibitive. As the stability of the system is determined by the sign of the rightmost (largest real part) eigenvalues, a more efficient approach is to use an iterative eigenvalue solver such as the Arnoldi method or subspace iteration. These iterative methods are designed to find n eigenvalues in specified regions of the eigenvalue spectrum, where $n \ll N$. Iterative eigenvalue solvers converge quickest to well-separated extreme eigenvalues of a system. Generally, the rightmost eigenvalues for systems resulting from the spatial discretization of PDEs are not well separated. This leads to slow convergence and even possible convergence to undesired eigenvalues. To overcome this problem, the system A is transformed into a new system T , where the rightmost eigenvalues of A are well-spaced extreme eigenvalues of T . The iterative methods can then be used to calculate the eigenvalues of T and then the eigenvalues can be transformed back to obtain the rightmost eigenvalues of A . There are a variety of different preconditioners available to transform the system A , such as Chebyshev polynomials, rational transforms, such as the shift-invert and Cayley transforms. Rational transforms are very good for separating eigenvalues making them a very suitable preconditioner for the Arnoldi method. Two different rational transforms were tried as preconditioners for ARPACK.

4.1. Shift invert

The shift-invert transform matrix is given by

$$T_{SI} = (A - \sigma B)^{-1} B. \quad (36)$$

The shift-invert transform maps the eigenvalues, λ of A , lying close to σ far away from σ and those eigenvalues far away close to σ , as shown in Figure 3. This means that the eigenvalues of interest, close to σ , are mapped to the extremities of the eigenvalue spectrum where they are now

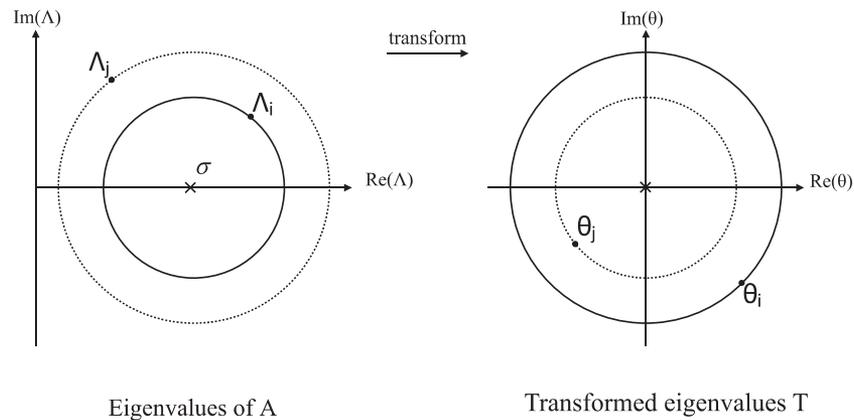


Figure 3. Mapping of the shift-invert transform.

well separated and can be easily calculated using Arnoldi's method. The transformed eigenvalues θ , where

$$\theta = (\Lambda - \sigma)^{-1},$$

are calculated and then transformed back to the eigenvalues of the original problem by

$$\Lambda = \frac{1}{\theta} + \sigma.$$

The transform requires the inverse of a matrix potentially making it expensive for large systems, but if a direct method can be used the factorization is only required once and the inverse matrix multiplications required for Arnoldi's method can be achieved relatively cheaply as back substitutions. The shift invert requires the user to have a good idea of the value of the required eigenvalue to set an appropriate value of σ and if the eigenvalue of interest has a large imaginary part, a complex shift would be needed.

4.2. Cayley transform

The Cayley transform of a matrix A is given by

$$T_C = (A - a_1 B)^{-1} (A - a_2 B),$$

where $a_1 > a_2$. This transforms the eigenvalues, λ , of A to the eigenvalues, θ , of T_C :

$$\theta = \frac{\Lambda - a_2}{\Lambda - a_1}.$$

The Cayley transform maps the eigenvalues Λ close to a_2 to eigenvalues θ far away from the unit circle and maps eigenvalues close to a_1 to eigenvalues θ with small modulus. As can be seen from Figure 4, this means that eigenvalues with real parts greater than $\frac{1}{2}(a_1 + a_2)$ will be mapped outside the unit circle. These means that a_1 and a_2 could be chosen in a manner that the rightmost eigenvalues can be found.

4.3. Eigenvalue search strategy

The shift-invert method converges quickly to eigenvalues close to σ . If $\sigma = 0$ is used, it will allow any real eigenvalues crossing the imaginary axis to be detected. However for flows that undergo a Hopf bifurcation, change from steady to unsteady flow, the shift-invert method is a poor choice for a preconditioner. This is because the Hopf bifurcation corresponds to a complex pair of eigenvalues crossing the imaginary axis and as can be seen from Figure 5 there are a large number of eigenvalues close to zero. This means that if the complex pair have large magnitude imaginary

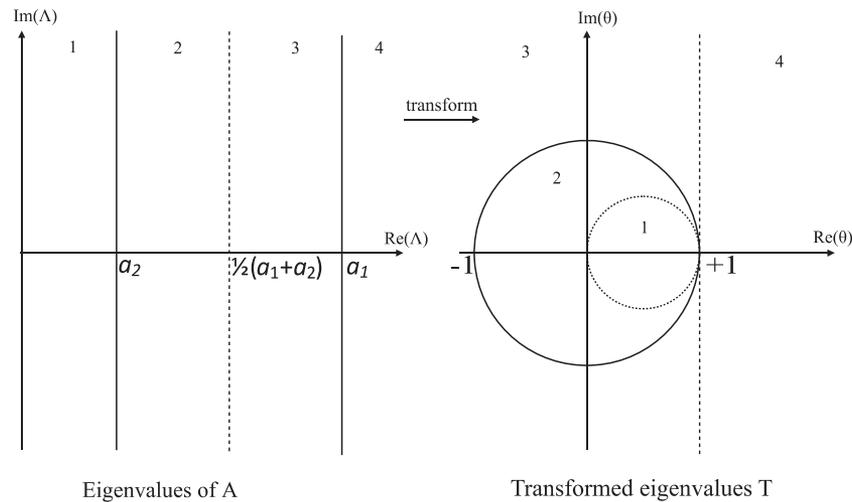


Figure 4. Mapping of the Cayley transform.

parts, a large number of eigenvalues will have to be found before a complex pair of eigenvalues of interest is found. Instead σ could be set to the magnitude of the complex eigenvalue of interest, but this is not always known *a priori*. Instead the shift invert, with $\sigma=0$, is used to produce an initial estimate of the rightmost eigenvalue, then the Cayley transform is used to check if there are any eigenvalues to the right of the ones found. This is done by setting

$$\frac{1}{2}(a_1 + a_2) = \Lambda_1^{\text{real}},$$

where Λ_1^{real} is the real part of the rightmost eigenvalue found using the shift-invert transform. This places any eigenvalues to the right of Λ_1^{real} outside the unit circle for the transformed matrix, see Figure 4; hence, the eigenvalue solver will now converge to any eigenvalues to the right of Λ_1^{real} first. A parameter τ , which is set as the distance between Λ_1^{real} and the eigenvalue, with the largest imaginary part, found using the shift invert, is used to set a_1 and a_2 ,

$$a_1 = \Lambda_1^{\text{real}} - \tau, \quad a_2 = \Lambda_1^{\text{real}} + \tau.$$

5. RESULTS

5.1. Validation of baseline CFD solver

Two of the viscous transonic workshop cases for the NACA0012 aerofoil [27] are presented in order to demonstrate that the underlying CFD solver can predict separated flow with reasonable accuracy. The first case is at $M_\infty=0.799$, $Re=9 \times 10^6$ and angle of attack $\alpha=2.26^\circ$. The pressure coefficients are shown in Figure 6 and the skin friction in Figure 7. The results show good agreement with the shock location. In this test case, a fairly strong shock tends to cause the boundary layer to separate over the rear of the aerofoil. This is predicted by the solver that shows separated flow from the shock to the trailing edge. The lower surface does not show good agreement with experimental results; however, this is consistent with both other implementations of the Spalart–Allmaras models and calculation with other turbulence models for this test case [28, 29].

The second case is at $M_\infty=0.55$, $\alpha=8.34$ and $Re=9 \times 10^6$. The pressure coefficients and the skin friction coefficients are shown in Figures 8 and 9, respectively. The calculations for this test case show a small region of separation downstream of the shock; the flow then reattaches before separating again very close to the trailing edge. This is consistent with the solutions in [30].

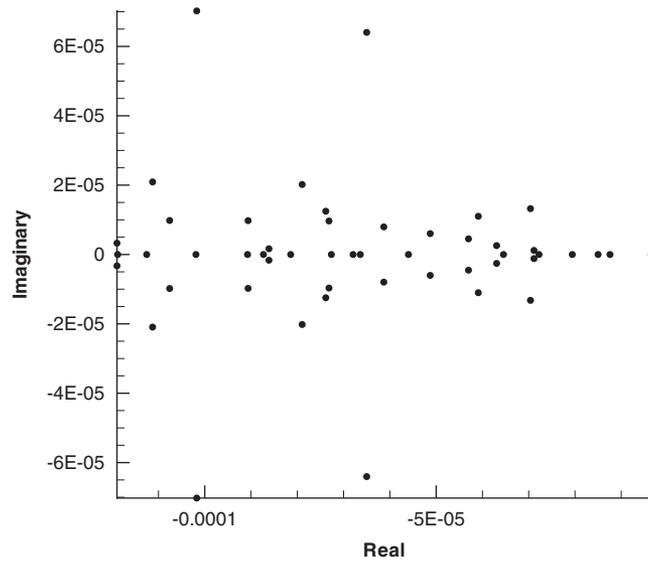


Figure 5. Plot showing a cluster of eigenvalues close to zero.

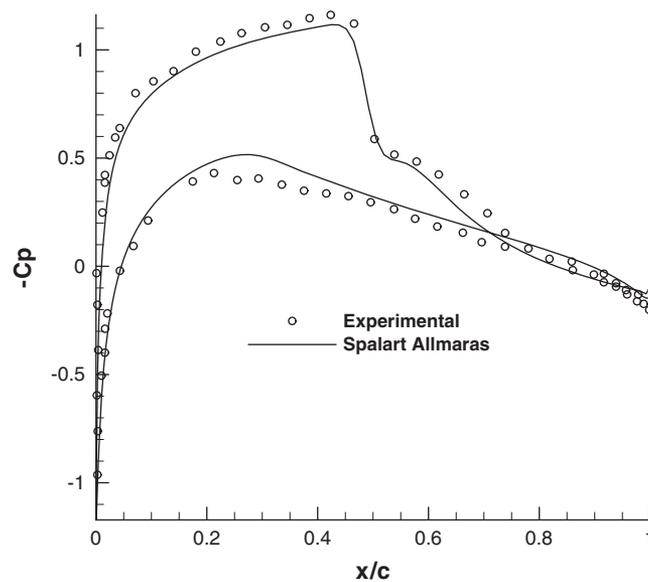


Figure 6. Pressure coefficient for an NACA 0012 at $M_\infty=0.799$, $Re=9 \times 10^6$ and $\alpha=2.26^\circ$.

5.2. Validation of continuation results

To check that the continuation extension to the baseline CFD solver does not change the Navier–Stokes results, a continuation is run with angle of attack as the parameter for an NACA 0012 airfoil. For this case, $M_\infty=0.5$ and $Re=2.9 \times 10^6$, and the results at four points are compared with experiment. All solutions are converged to a residual value less than 10^{-9} and the target number of corrector iterations was set at 6, for step length control. At this moment, the code is limited to single-parameter continuation on a fixed mesh. This makes it a challenge to create a single mesh for a large range of flow conditions. With sufficient refinement in the boundary layer, a mesh should be good enough to predict the onset of separation and flow unsteadiness. A future extension to this work will be to include mesh refinement, which would allow meshes better adapted to each flow as the parameter value changes. All the meshes used here have a $y^+ \approx 1$ across all parameter

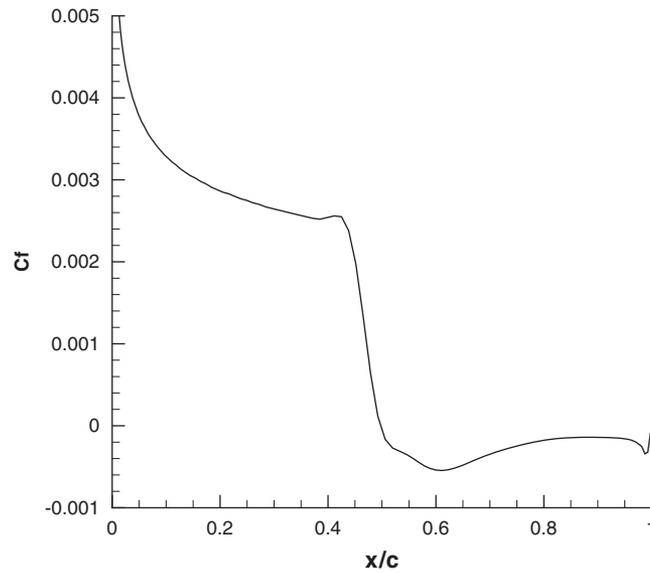


Figure 7. Skin friction coefficient for an NACA 0012 at $M_\infty=0.799$, $Re=9\times 10^6$ and $\alpha=2.26^\circ$.

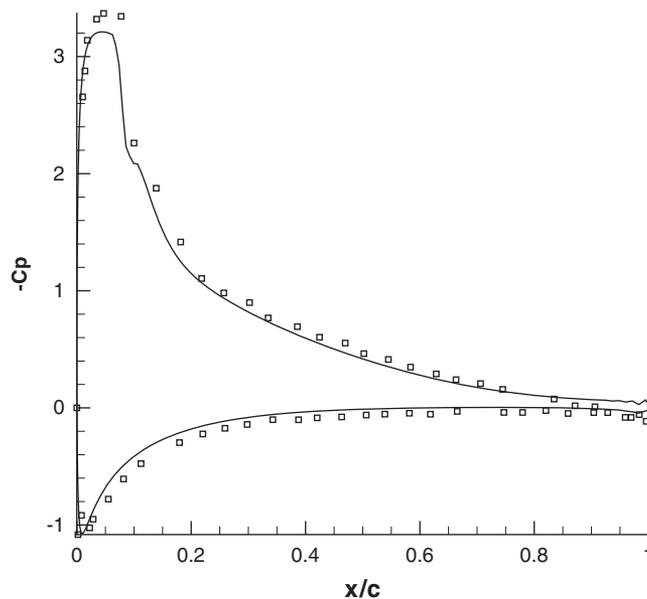


Figure 8. Pressure coefficient for an NACA 0012 at $M_\infty=0.55$, $Re=9\times 10^6$ and $\alpha=8.34^\circ$.

values. Figure 10 shows the variation of, the coefficient of lift, for the steady- state solutions of the 2D Navier–Stokes equations and the stability of the corresponding solutions. The surface pressures are compared with the experimental results [31] at four points along the stable solution branch, in Figure 11. As α is the continuation parameter, the actual values it takes is controlled by the step length algorithm that is embedded in the process. Thus, calculated solutions cannot be forced to fall exactly on the experimental angles of attack. The solutions that are closest to the experimental values are therefore taken for comparison. It should be noted that all the calculated solutions used in the comparisons have values of α within 0.07177° of the experimental values. The pressures match closely allowing for the slight differences in α . Figure 12 shows surface pressures for the unstable steady flow solutions found above 9.15° .

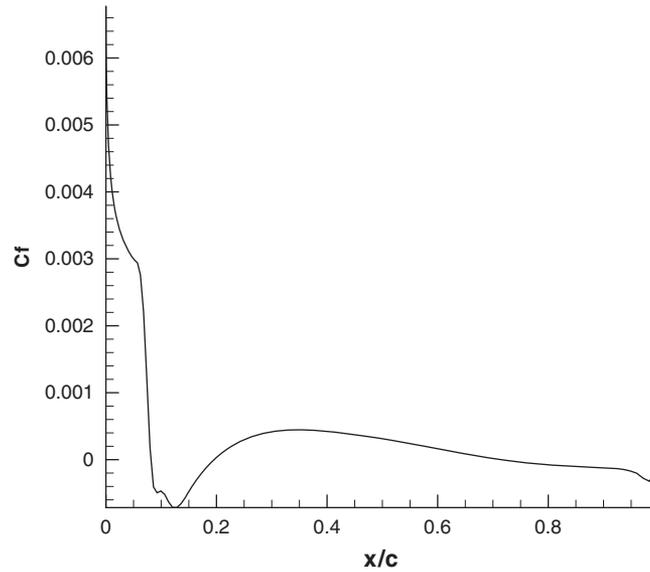


Figure 9. Skin friction coefficient for an NACA 0012 at $M_\infty=0.55$, $Re=9 \times 10^6$ and $\alpha=8.34^\circ$.

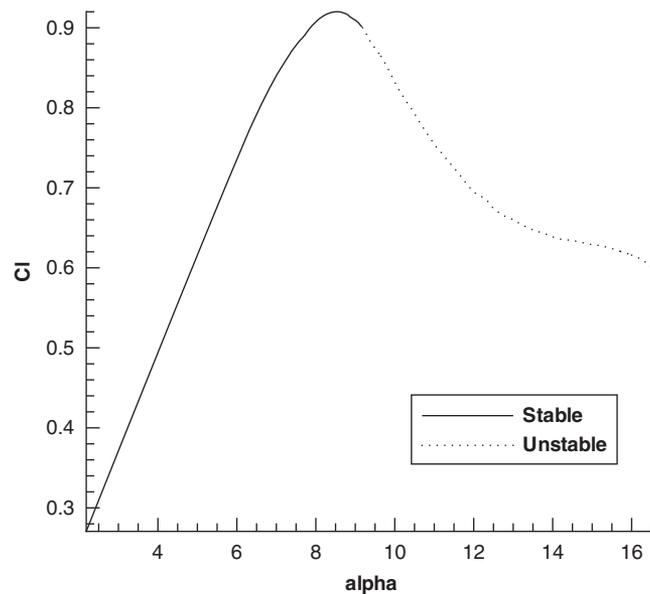


Figure 10. Lift coefficient versus angle of attack, NACA 0012, $M=0.5$, $Re=2.9 \times 10^6$.

The solutions at the highest angles of attack correspond to steady solutions of the Navier–Stokes equations that are unstable, that is if a solution is subject to a small perturbation then it would not return to the original solution. As real flows are not free of perturbations, the unstable branches cannot be found directly experimentally. These unstable branches are important as they can be boundaries between the domain of attraction for two stable solutions. The change in stability corresponds to a complex pair of eigenvalues crossing the imaginary axis, corresponding to a Hopf bifurcation, as shown in Figure 13. This means that if a small perturbation is applied to the solution, it will grow eventually forming a periodic flow. In order to check the changes in stability, a time accurate dual-time [32] version of the baseline CFD code was used to produce unsteady solutions of the nonlinear equations. This change in stability is confirmed by running time accurate unsteady

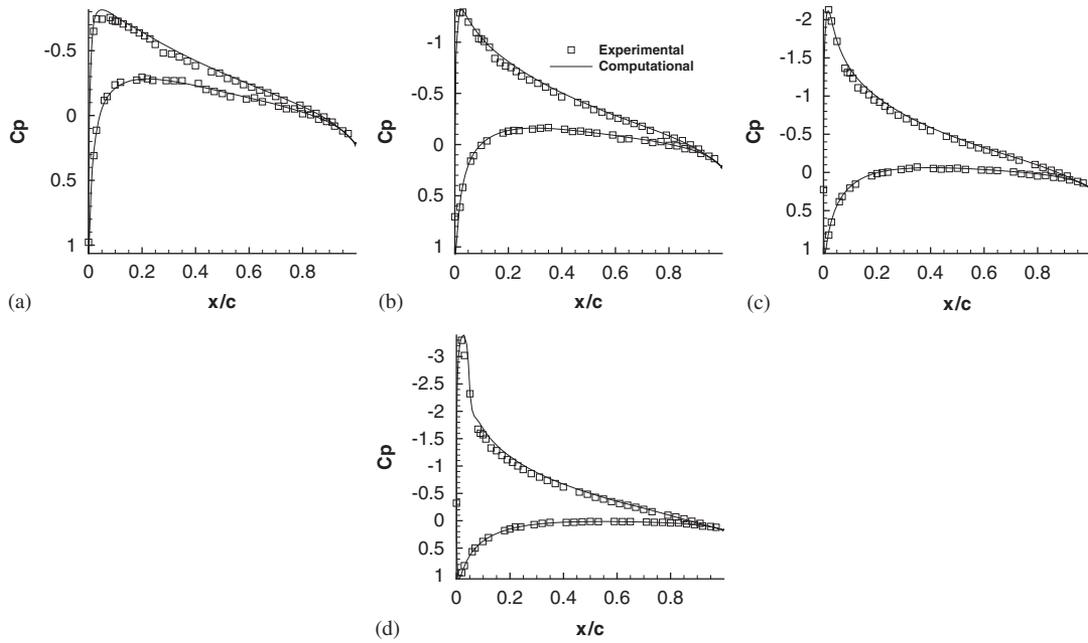


Figure 11. Surface C_p for an NACA 0012, $M_\infty=0.5$, $Re=2.9 \times 10^6$, at different angles of attack: (a) 1.80° ; (b) 3.35° ; (c) 5.02° ; and (d) 6.98° .

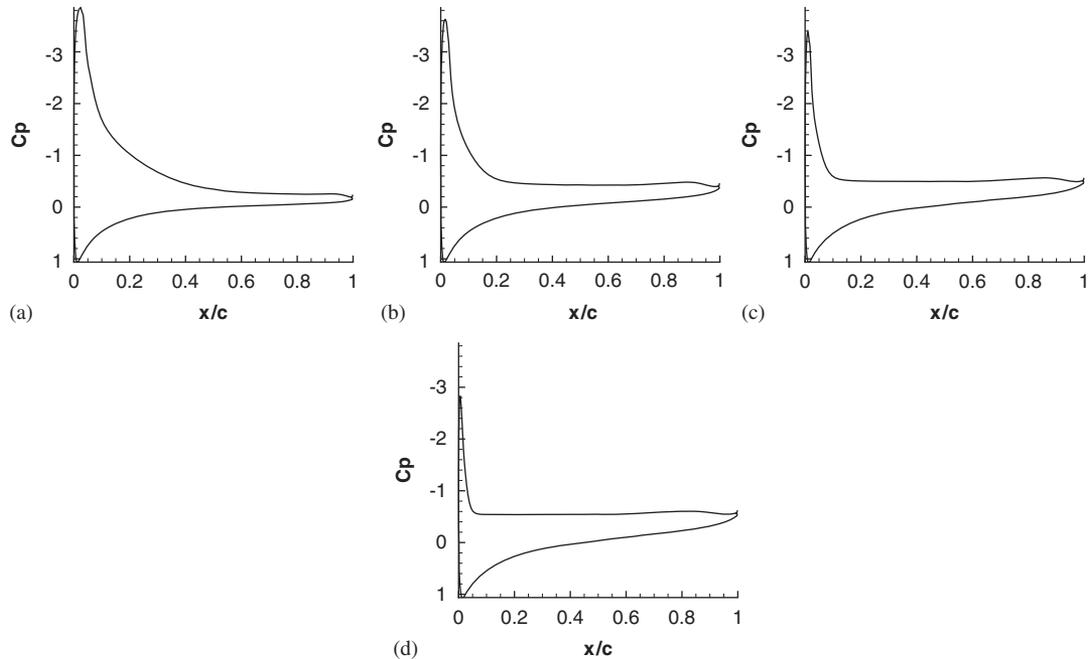


Figure 12. Surface C_p for unstable steady-state solutions of an NACA 0012, $M_\infty=0.5$, $Re=2.9 \times 10^6$, at different angles of attack: (a) 10.0° ; (b) 12.0° ; (c) 14.0° ; and (d) 16.0° .

simulations at 9.0° and 9.5° . Each solution is started from the solution generated by the continuation algorithm with a small perturbation applied. The time histories (lift coefficient against Time Step Number—NDTS) are shown in Figure 14. At 9.0° , the perturbation results in a oscillation that decays to a steady solution indicating that the solution is stable. However at 9.5° , the perturbation

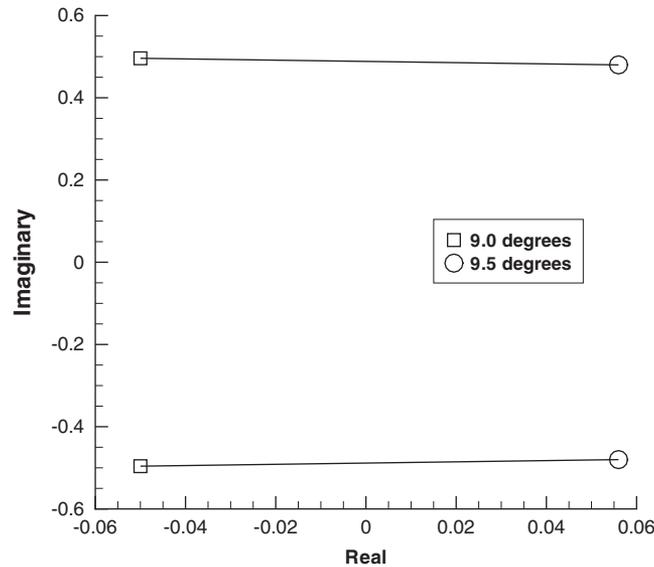


Figure 13. Eigenvalues crossing the imaginary axis, showing a change in stability.

grows indicating that the solution is unstable. This shows that the change in stability predicted by the crossing of the eigenvalues is correct.

5.3. Accuracy of the Jacobian

The main parameter that can be used to control the accuracy of the Jacobian is the finite difference step size. In [33], the effect of the finite difference step size on the accuracy of the Jacobian for the flow around a 15° ramp at $M_\infty=2$ was investigated. In that study, the effects of high flow-gradients, which will effect the truncation error in numerical Jacobians, could be examined. It was found that there was an optimum step size of around 4×10^{-8} for double precision calculations. A study was carried out to determine the sensitivity of the eigenvalues to the finite difference step size. A series of Jacobians were calculated from a converged solution using different perturbation sizes and what effect it had on the real part of the rightmost eigenvalue found. The perturbation sizes were set using different values of $\sqrt{\eta}$ for Equation (29). Once $\sqrt{\eta}$ gets above 10^{-6} , the real part of the rightmost eigenvalue starts to change rapidly and even changes sign which would lead to the incorrect local stability being identified, as seen in Figure 15. For all results, a value of $\sqrt{\eta}=10^{-8}$ has been used.

Whereas an approximate Jacobian can be sufficient to drive the residual to zero using the Newton method, an accurate Jacobian is required to calculate the correct eigenvalues. The Jacobian is a linear approximation to the nonlinear system about the point the Jacobian is calculated. Hence to check that the Jacobian has the correct local dynamical behaviour the time response to a small perturbation, about a converged solution, of the Jacobian was compared with a time accurate simulation of the full nonlinear CFD code. Both the linear and nonlinear responses were produced using first-order time stepping. The time histories match for the first few time steps, as shown in Figure 16, which confirms that close to the solution, the linearization about the solution produces the correct flow dynamics. As time increases, the two solutions start to diverge due to the nonlinear effects which stop the oscillations growing and eventually form a limit cycle, whereas the linear solution continues growing.

5.4. Parametric studies

Numerical continuation methods can be used to analyse the effect that changes in one parameter have on the nonlinear dynamics of the system. This parameter can be based on the flow conditions;

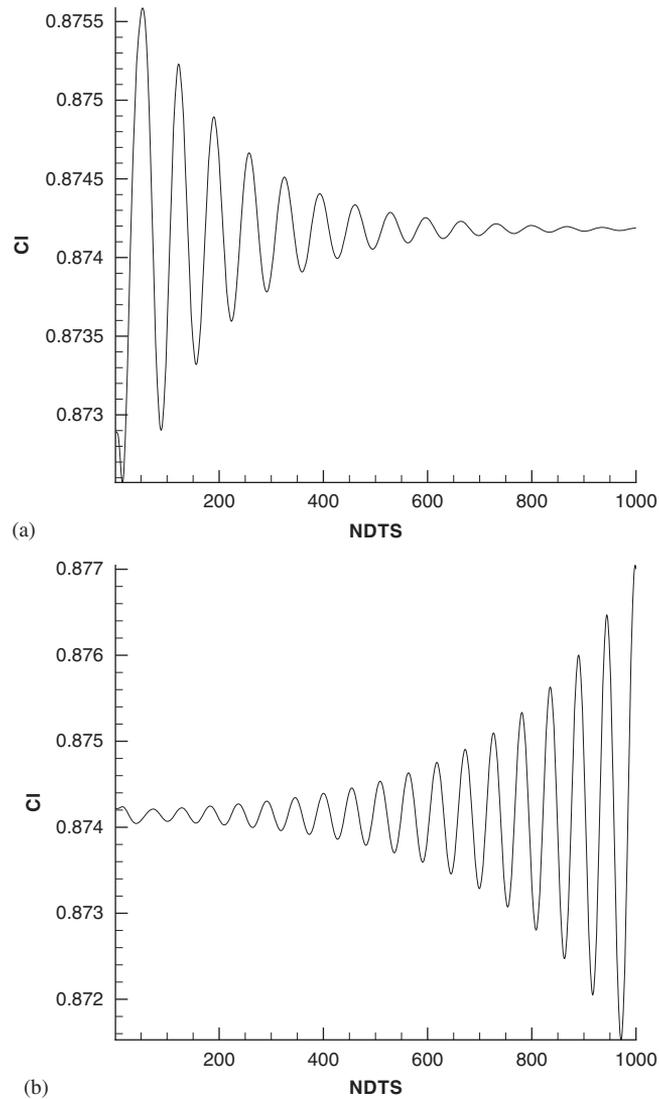


Figure 14. Time histories of lift coefficient past an NACA 0012 airfoil at $M_\infty = 0.5$, $Re = 2.9 \times 10^6$: (a) 9.0° and (b) 9.5° .

such as Mach number, Reynolds number, angle of attack or onset turbulence intensity. The parameter investigated could also be a coefficient in the turbulence model, a numerical parameter such as the level of dissipation applied or the change in a geometric parameter. The effect of two or more different parameters can be investigated using standard continuation in one parameter for a series of fixed values of the other parameters. In this way, the changes in nonlinear dynamics with respect to the parameters can be built up. In the current study the effect of changing the freestream turbulence intensity and the thickness for a symmetric airfoil, on the steady-state solution as the angle of attack is varied, was investigated.

5.4.1. Comparison of nonlinear behaviour of different turbulence models. The level of freestream turbulence intensity is known to effect the nonlinear behaviour of an aerofoil [34]; hence, the Edwards modification of the Spalart–Allmaras turbulence model, see Section 2.4, was also implemented. The Edwards model has better numerical behaviour close to the wall that allows convergence at lower freestream $\bar{\mu}_{\text{freestream}}$ values than the original model. The level of $\bar{\mu}_{\text{freestream}}$ is a model parameter that effects the solution in a similar way to increasing freestream

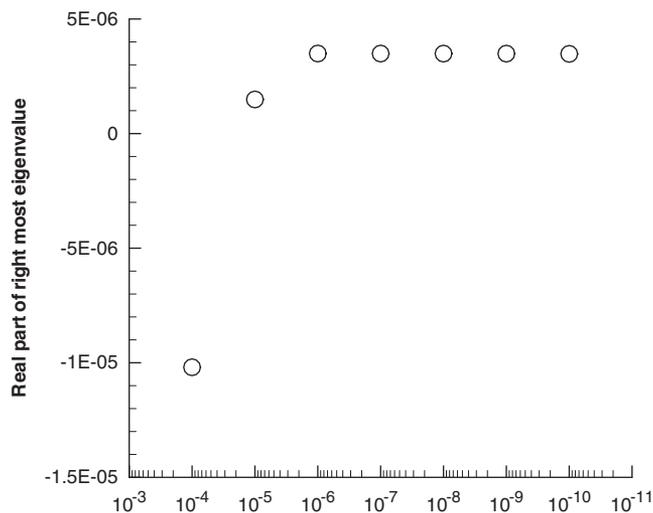


Figure 15. The variation in the real part of the rightmost eigenvalue with $\sqrt{\eta}$.

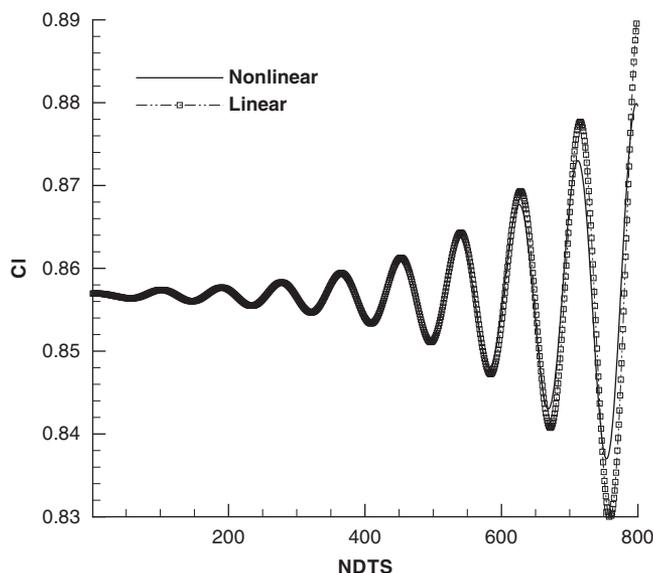


Figure 16. Comparison of the linear system and the nonlinear system responses.

turbulence intensity. A comparison of the two turbulence models at two different freestream conditions in Figure 17 shows that their post stall behaviour is quite different. This is particularly evident at the lower freestream boundary condition, where the modified turbulence model shows a small region of hysteresis between 16° and 17° ; within this region there are multiple steady solutions, all of which are unstable. For both turbulence models, the behaviour through stall depends on the freestream conditions. However solutions before and after stall are the same, although the Edwards modification produces a lower C_L for the unstable steady solution post stall.

This effect of freestream boundary conditions on the solution behaviour through the stall region was investigated for a wider range of freestream conditions for the Spalart–Allmaras model with the Edwards modification. Between the $\bar{\mu}_{\text{freestream}} = 0.1\mu$ and $\bar{\mu}_{\text{freestream}} = 1.0\mu$, there is very little change in solutions but as $\bar{\mu}_{\text{freestream}}$ increases the hysteresis becomes smaller until it is eventually unfolded and the solution curve no longer has regions where there are multiple solutions. The

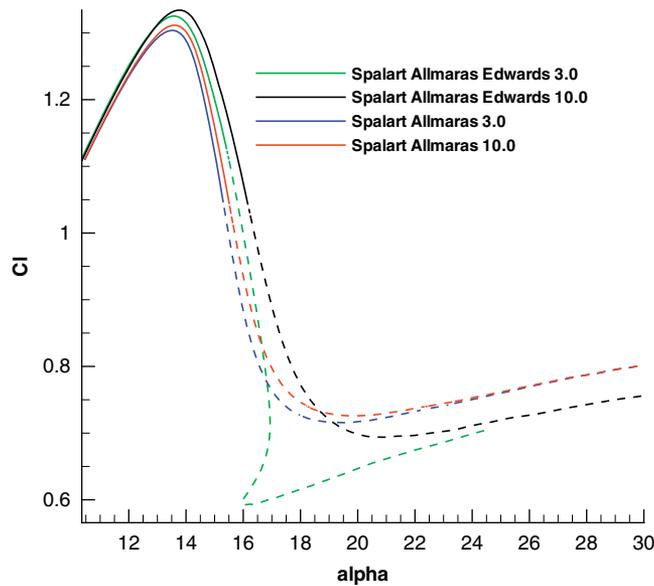


Figure 17. Comparison of turbulence models with different inflow boundary conditions $\bar{\mu}_{\text{freestream}}$, for the flow around an NACA 0012 airfoil at $M_{\infty}=0.3$, $Re=1.85 \times 10^6$.

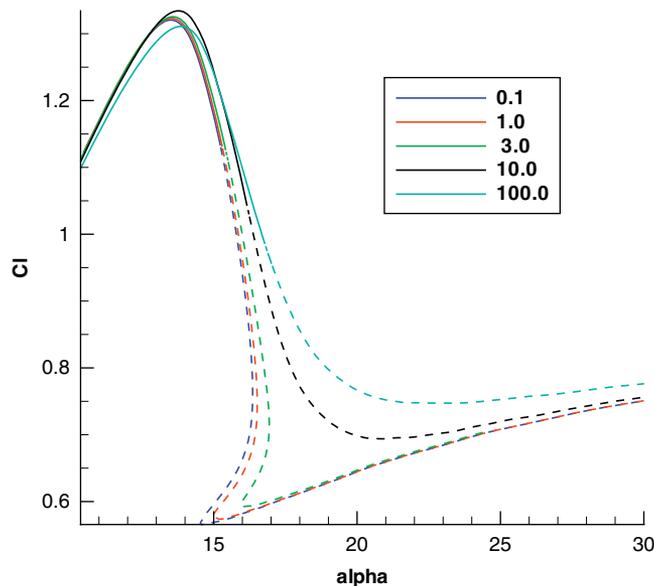


Figure 18. Effect of turbulent inflow boundary condition, $\bar{\mu}_{\text{freestream}}$, on the flow solution for the Spalart-Allmaras turbulence model, for the flow around an NACA 0012 airfoil at $M_{\infty}=0.3$, $Re=1.85 \times 10^6$.

second effect that can be seen in Figure 18 is that as $\bar{\mu}_{\text{freestream}}$ increases, the α at which the solution becomes unstable is also delayed. It should be noted that the onset of instability was by the Hopf bifurcation in all cases. Figures 19 and 20 show the pressure and density contour maps around the airfoil at $\alpha=16.25$ for $\bar{\mu}_{\text{freestream}}=\mu_{\infty}$, where there are multiple solutions. Figure 19 shows that the suction peak at the leading edge decreases and there is a corresponding loss in lift.

5.4.2. Effect of geometry changes. To investigate the effect of airfoil thickness on stall behaviour, a series of symmetric NACA four series airfoils were run using the continuation algorithm described earlier with angle of attack as the continuation parameter. As can be seen from Figure 21, increasing

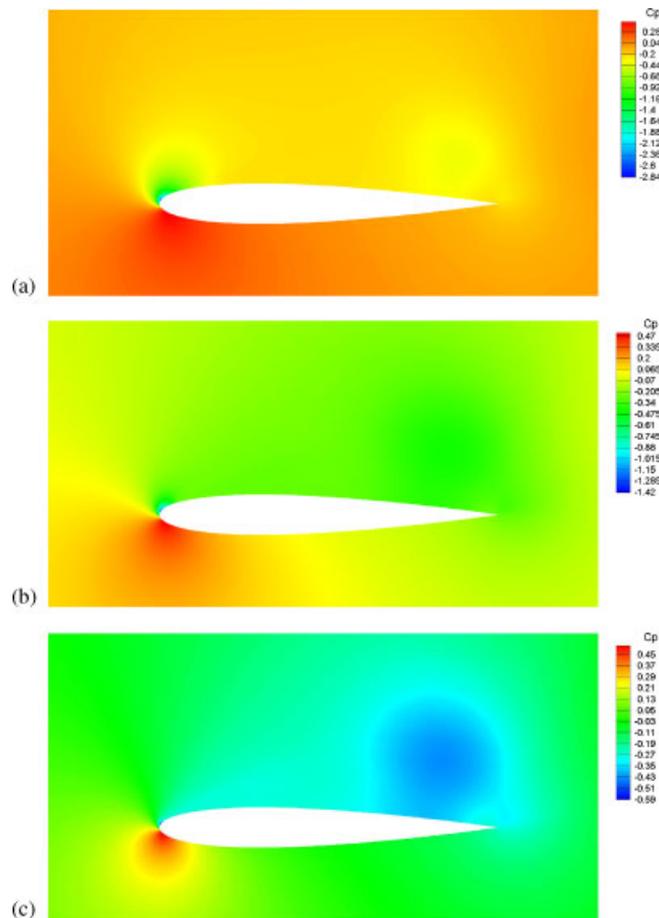


Figure 19. Pressure contours for multiple solutions at $\alpha = 16.25$ and $\bar{\mu}_{\text{freestream}} = \mu_{\infty}$ for the flow around an NACA 0012 airfoil at $M_{\infty} = 0.3$, $Re = 1.85 \times 10^6$, using the Edwards modified Spalart–Allmaras: (a) top branch; (b) middle branch; and (c) bottom branch.

the airfoil thickness increases the $C_{l_{\max}}$, and that the rate of increase in $C_{l_{\max}}$ decreases as the thickness increases. A similar trend is observed with the loss of steady-state stability. As the thickness increases, the angle at which the flow becomes unsteady also increases. In all cases, this change in stability corresponds to a complex pair of eigenvalues crossing the imaginary axis, indicating that a Hopf bifurcation has occurred which would lead to the flow becoming periodic. For the NACA 0006, the flow separates from the leading edge first and for all the other thicker airfoils the flow separates from the trailing edge. Note that at present this change in behaviour is not detectable by that analysis.

6. CONCLUSIONS

The feasibility of analysing the nonlinear steady-state dynamics for the 2D compressible Reynolds-averaged Navier–Stokes equations with two variations of the Spalart–Allmaras turbulence model using a numerical continuation method has been demonstrated in this study. Numerical continuation can be used to analyse the effect of changes in one parameter, on the nonlinear dynamics of the system over a second parameter space. Continuation methods can be used to find nonunique solutions if multiple solutions to the investigated system exist: in addition the unstable steady-state solutions can be found and traced, which is not possible with standard CFD solvers. An important part of this method is the calculation of the local stability of the solutions. It has been shown

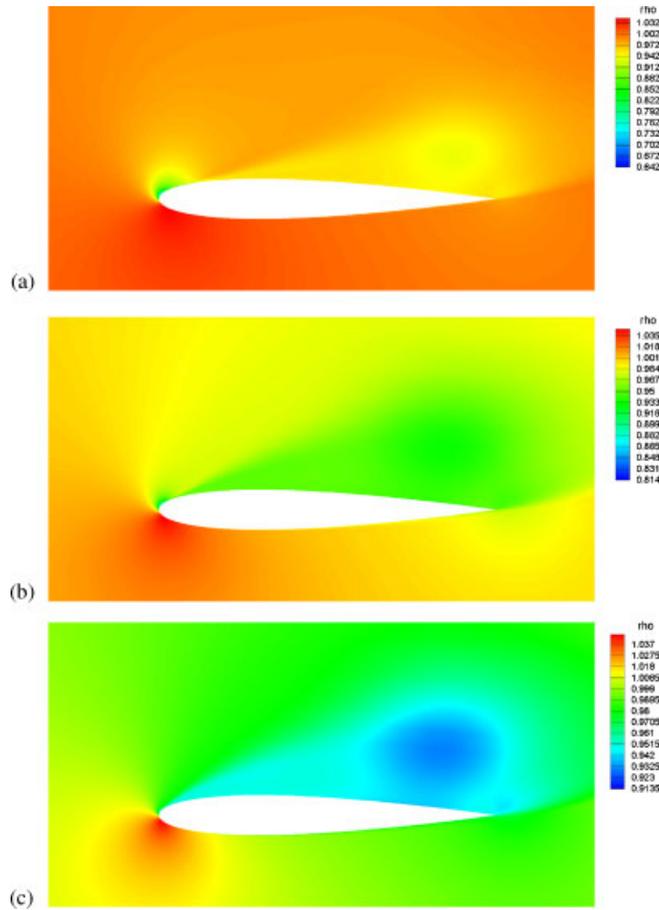


Figure 20. Density contours for multiple solutions at $\alpha=16.25$ and $\bar{\mu}_{\text{freestream}}=\mu_{\infty}$ for the flow around an NACA 0012 airfoil at $M_{\infty}=0.3$, $Re=1.85 \times 10^6$, using the Edwards modified Spalart–Allmaras: (a) top branch; (b) middle branch; and (c) bottom branch.

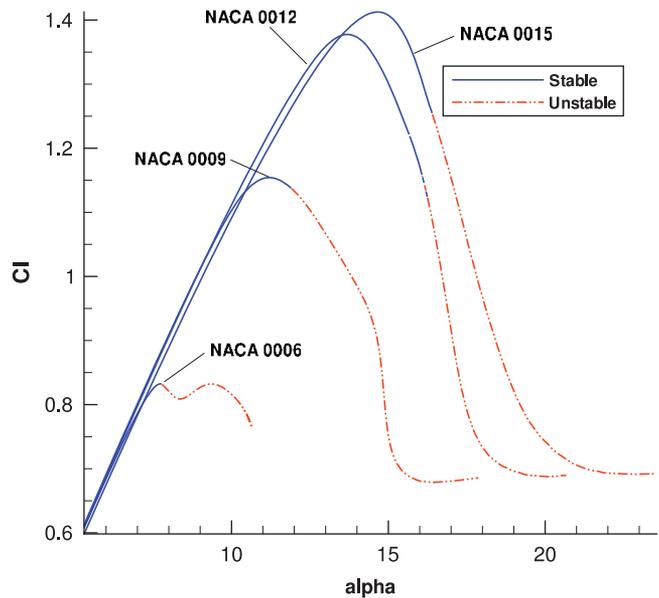


Figure 21. Effect of changing airfoil thickness for symmetric NACA four series airfoil at $M_{\infty}=0.3$, $Re=9.0 \times 10^6$.

that by using the ARPACK routines, along with a preconditioner based on the Cayley transform, it is possible to find the rightmost eigenvalues of the system and accurately determine the local stability, so that the critical parameter values where this change in stability occurs can be found.

REFERENCES

1. Mittal S, Saxena P. Prediction of hysteresis associated with static stall of an airfoil. *AIAA Journal* 2000; **38**:933–935.
2. Keller HB. Numerical solution of bifurcation and nonlinear eigenvalue problems. *Applications of Bifurcation Theory*. Academic Press: New York, 1977; 359–384.
3. Doedel EJ, Keller HB, Kernevez JP. Numerical analysis and control of bifurcation problems: (I) Bifurcation in finite dimensions. *International Journal of Bifurcation and Chaos* 1991; **1**(3):493–520.
4. Doedel EJ, Keller HB, Kernevez JP. Numerical analysis and control of bifurcation problems, part II. *International Journal of Bifurcation and Chaos* 1991; **3**(4):493–520.
5. Doedel EJ, Champneys AR, Fairgrieve TF, Kuznetsov YA, Sandstede B, Wang X. AUTO97: continuation and bifurcation software for ordinary differential equations (with HomCont). Computer Science, Concordia University, Montreal, Canada, 1986.
6. Rheinboldt WC, Burkardt JV. Algorithm 596: a program for a locally parameterized. *ACM Transactions on Mathematical Software* 1983; **9**(2):236–241.
7. Dhooze A, Govaerts W, Kuznetsov YuA. Matcont: a matlab package for numerical bifurcation analysis of ODES. *ACM Transactions on Mathematical Software* 2003; **29**(2):141–164.
8. Sanchez J, Marques F, Lopez JM. A continuation and bifurcation technique for Navier–Stokes flows. *Journal of Computational Physics* 2002; **180**(1):78–98.
9. Cliffe KA, Spence A, Tavener SJ. The numerical analysis of bifurcation problems with application to fluid mechanics. *Acta Numerica* 2009; **9**(1):39–131.
10. Lehoucq RB, Salinger AG. Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. *International Journal for Numerical Methods in Fluids* 2001; **36**(3):309–327.
11. Salinger AG, Lehoucq RB, Pawlowski RP, Shadid JN. Computational bifurcation and stability studies of the 8:1 thermal cavity problem. *International Journal for Numerical Methods in Fluids* 2002; **40**(8):1059–1073.
12. Spalart P, Allmaras S. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 92-0439*, 1992.
13. Schmidt W, Jameson AJ, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes. *AIAA Paper 81-1259*, 1981.
14. Spalart PR, Rumsey CL. Effective inflow conditions for turbulence models in aerodynamic calculations. *AIAA Journal* 2007; **45**(10):2544–2553.
15. Seror S, Rubin T, Peigin S, Epstein B. Implementation and validation of the Spalart–Allmaras turbulence model in parallel environment. *Journal of Aircraft* 2005; **42**(1):179–188. *AIAA 21st Applied Aerodynamics Conference*, Orlando, FL, 23 June 2003.
16. Edwards JR, Chandra S. Comparison of eddy viscosity-transport turbulence models for three-dimensional, shock-separated flowfields. *AIAA Journal* 1996; **34**(4):756–763.
17. Kroll N, Jain R. Solution of two-dimensional Euler equations: experience with a finite volume code. *Forschungsbericht- Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt*, 1987.
18. Govaerts W. Stable solvers and block elimination for bordered systems. *SIAM Journal on Matrix Analysis and Applications* 1991; **12**:459–483.
19. Vanden KJ, Orkwis PD. Comparison of numerical and analytical Jacobians. *AIAA Journal* 1996; **34**(6):1125–1129.
20. Dennis JE, Schnabel RB. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial Mathematics: Philadelphia, PA, 1996.
21. Coleman TF, More JJ. Estimation of sparse Jacobian matrices and graph-coloring problems. *SIAM Journal on Numerical Analysis* 1998; **20**(1):187–209.
22. Knoll DA, Keyes DE. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**(2):357–397.
23. Davis TA. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 2004; **30**(2):165–195.
24. Davis TA. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software* 2004; **30**(2):196–199.
25. Davis TA, Duff IS. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Transactions on Mathematical Software* 1999; **25**(1):1–20.
26. Davis TA, Duff IS. An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM Journal on Matrix Analysis and Applications* 1997; **18**(1):140–158.
27. Holst T. Viscous transonic airfoil workshop compendium of results. *Journal of Aircraft* 1988; **25**(12):1073–1087.
28. Elkhoury M. Assessment and modification of one-equation models of turbulence for wall-bounded flows. *Journal of Fluids Engineering* 2007; **129**:921.
29. Doerffer P, Szulc O. Shock wave smearing by wall perforation. *Archives of Mechanics* 2006; **58**(6):543.

30. Rumsey C, Anderson W. Parametric study of grid size, time step and turbulence modeling on Navier–Stokes computations over airfoils. *AGARD, Validation of Computational Fluid Dynamics*, Lisbon, Portugal, vol. 1, 2–5 May 1988.
31. Thibert JJ, Grandjacques M, Ohman LH. Experimental data base for computer program assessment. *AGARD AR-138*, 1979.
32. Gaitonde AL. A dual-time method for the solution of the 2D unsteady Navier–Stokes equations. *Proceedings of the 13th AIAA Applied Aerodynamics Conference*, San Diego, CA, 1995; 95–1877.
33. Onur O, Eyi S. Effects of the Jacobian evaluation on Newton’s solution of the Euler equations. *International Journal for Numerical Methods in Fluids* 2005; **49**(2):211–231.
34. Hoffmann J. Effects of free stream turbulence on the performance characteristics of an airfoil. *AIAA Journal* 1991; **29**:1353–1354.