INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:1361–1402
Published online 10 May 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1232

Study of linear response identification techniques and reduced-order model generation for a 2D CFD scheme

Ann L. Gaitonde*,† and D. P. Jones

Department of Aerospace Engineering, University of Bristol, University Walk, Bristol BS8 1TR, U.K.

SUMMARY

Reduced-Order Models (ROMs) have been the focus of research in various engineering situations, but it is only relatively recently that such techniques have begun to be introduced into the CFD field. The purpose of generating such models is to capture the dominant dynamics of the full set of CFD equations, but at much lower cost. One method that has been successfully implemented in the field of fluid flows is based on the calculation of the linear pulse responses of the CFD scheme coupled with an Eigensystem Realization algorithm (ERA), resulting in a compact aerodynamic model. The key to the models is the identification of the linear responses of the non-linear CFD code. Two different methods have been developed and reported in literature for linear response identification; the first method linearizes the CFD code and the second method uses Volterra theory and the non-linear code. As these methods were developed independently they have not previously been brought together and compared. This paper first explains the subtle, but fundamental differences between the two methods. In addition, a series of test cases are shown to demonstrate the performance and drawbacks of the ROMs derived from the different linear responses. The conclusions of this study provide useful guidance for the implementation of either of the two approaches to obtain the linear responses of an existing CFD code. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: linear pulse responses; reduced-order model; CFD

1. INTRODUCTION

Accurate modelling of the fluid motion about aerofoils and wings is computationally expensive for unsteady problems, particularly in the fields of aeroelastics and aeroservoelastics. This is because the full non-linear Euler or Navier–Stokes equations have many degrees of freedom for typical configurations (see for example References [1–5]) and are therefore inpractical when many parameter variations need to be investigated, such as during flutter boundary prediction. More recently, research has been directed towards the application of

*Correspondence to: Ann L. Gaitonde, Department of Aerospace Engineering, University of Bristol, University Walk, Bristol BS8 1TR, U.K.
†E-mail: ann.gaitonde@bris.ac.uk

Received 4 October 2005
Revised 13 February 2006
Accepted 15 February 2006

system reduction methods, developed in other fields of engineering, to fluid modelling. The objective of a reduced-order model (ROM) is to identify from the full system of equations a lower cost model that still captures the dominant behaviour. While many reduction methods can be directly applied to the matrices of the large time-continuous linear system of equations (e.g. eigensystem realization algorithm (ERA) (modified Ho algorithm) [6, 7], POD [8], Krylov-subspace [9], Arnoldi [10, 11]), in fluid dynamics, it has been more common to create ROMs in the discrete frequency or discrete time domain using methods that do not require explicit construction of the system matrices. This is because in general for many computational fluid dynamics (CFD) codes, neither the continuous nor the discrete system matrices are ever explicitly constructed. Instead, the CFD code solves an approximation to the continuous time set of equations (obtained after spatial discretization) in either the discrete frequency domain [12–15] or the discrete time domain [16–18], and thus discrete frequency or discrete time ROMs are initially obtained. For an extensive review of fluid motion modelling from classical methods through to recent developments, see Reference [19].

This paper focuses on the construction of a ROM of a CFD code that is solved in the discrete time domain. A ROM can be constructed by assuming the flow can be represented as the sum of a statically non-linear base flow and a dynamically linear perturbation. Depending on the choice of steady base flow model, it may be possible to include many physical phenomena for example shock waves, 3D vorticity and separated flows. However due to the assumption of a small dynamic perturbation, any dynamic changes to these features must be small. The assumption of dynamic linearity is a common approach used in the development of ROMs and it is a good approximation in many situations e.g. flutter calculations. Making this assumption the full non-linear system is replaced by a linear time-dependent system that can be written in state-space form. If the dynamically linear pulse responses of the linear system can be calculated, then ERA, a standard system identification algorithm developed in the field of system dynamics, can be used to find a lower-order linear state-space model of the time-dependent system.

In the literature, two methods to obtain the required linear responses of the CFD code have been independently described. There are subtle, but fundamental differences between these two methods. In the first approach the linear responses can be found directly by time-linearizing the CFD code [20, 21] and performing a series of calculations for pulse inputs [22]. The system of equations solved is the actual dynamically linear version of the full CFD code. In the second approach, each linear pulse response is approximated as the linear portion of the non-linear response of the full non-linear code; this approach has been used by Silva [17]. In this case the non-linearity of the flow is assumed to have a particular form. This yields a formula for the linear portion of the non-linear response and requires twice as many pulse response calculations to be performed; the need to linearize the CFD code is, however, avoided. The application of both approaches to the same fluid problem has not previously been reported, and is considered in detail here.

This paper considers the issues involved in the implementation of both methods for linear response identification and explores the subsequent impact on the behaviour of the solutions of the ROMs produced. To enable comparisons between the methods, the individual steps in the formation of the ROM are considered separately so areas of difference can be highlighted. Although a specific flow solver has been used the conclusions should be general. The CFD code used, as the basis of this work, is a modified Jameson cell-centre CFD scheme that solves the full unsteady non-linear Euler equations on a moving mesh. Spatial discretization yields

a set of continuous time non-linear differential equations. The code solves these equations in the discrete time domain using a dual-time approach. The test case chosen is sinusoidal pitching for a range of frequencies; two Mach numbers are selected to illustrate flows with and without shockwaves. This test case was chosen as it will show the frequency range that has been captured by the ROMs.

2. 2D EULER EQUATIONS ON A MOVING GRID

The 2D unsteady Euler equations in non-dimensional integral form on a moving mesh are

$$\frac{\mathrm{d}}{\mathrm{d}t}\iint_{\Omega}\mathbf{W}\,\mathrm{d}x\,\mathrm{d}y+\int_{\partial\Omega}(\mathbf{F}\,\mathrm{d}y-\mathbf{G}\,\mathrm{d}x)=0 \tag{1}$$

where

$$\begin{aligned}\mathbf{W}&=[\rho,\rho u,\rho v,E_t]^{\mathrm{T}}\\ \mathbf{F}&=[\rho(u-x_t),\rho u(u-x_t)+p,\rho v(u-x_t),E_t(u-x_t)+pu]\\ \mathbf{G}&=[\rho(v-y_t),\rho u(v-y_t),\rho v(v-y_t)+p,E_t(v-y_t)+pv]\end{aligned} \tag{2}$$

where $\mathbf{X}_t=[x_t,y_t]^{\mathrm{T}}$ is the mesh speed and $E_t=\rho e+\rho(v^2+u^2)$ is the total energy. In this study, the non-linear unsteady equation (1) is solved using a version of Jameson's cell-centred approach that is extended to be time accurate and account for mesh motion.

Equation (1) is applied to the ijth cell giving the following non-linear system:

$$\frac{\mathrm{d}(\mathbf{W}_{ij}V_{ij})}{\mathrm{d}t}+\sum_{k=1}^{4}\{\mathbf{F}_k\Delta y_k-\mathbf{G}_k\Delta x_k\}_{ij}-\mathbf{D}_{ij}=0 \tag{3}$$

where $\mathbf{W}_{ij}$ is the cell average value of $\tilde{\mathbf{W}}$, V_{ij} is the cell volume and $\mathbf{D}_{ij}$ is numerical dissipation. These non-linear time-continuous equations are actually solved in discrete-time form via a dual-time scheme [4, 5], together with a geometric conservation law (GCL) to account for the moving mesh [23, 24].

3. IDENTIFYING LINEAR RESPONSES FOR ROMs

3.1. Linear state-space models

The above CFD equations (3) together with an output equation can be represented in the following multi-input multi-output (MIMO) system form

$$\begin{aligned}\dot{\mathbf{x}}(t)&=f(\mathbf{x}(t),\mathbf{u}(t))\\ \mathbf{y}(t)&=g(\mathbf{x}(t),\mathbf{u}(t))\end{aligned} \tag{4}$$

where $\mathbf{u}$ is the input vector and $\mathbf{x}$ is the state-vector. The vector $\mathbf{y}$ is the output and is chosen to produce the required information about the system.

Consider as an example the application of these equations to calculations of the flow about aerofoils with trailing edge flaps that can undergo linearized pitch motions [21], then

$$\mathbf{u} = [\alpha, \delta, \dot{\alpha}, \dot{\delta}]^{\mathrm{T}} \tag{5}$$

where α is the pitch angle and δ is the flap angle.

An example of a suitable $\mathbf{y}$ for coupling with a simple structural model for this motion is

$$\mathbf{y} = [\hat{C}_l, \hat{C}_m, \hat{C}_h]^{\mathrm{T}} \tag{6}$$

where $\hat{C}_l$, $\hat{C}_m$ and $\hat{C}_h$ are the changes in the lift, pitching moment and hinge moment coefficients from their mean values.

If the dynamic behaviour of the system is assumed to be approximately linear, then the non-linear system (4) can be replaced by a linear time-continuous state-space system

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{7}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ are the system matrices.

It is assumed that the method used to put this linear system into discrete-time form is the same as for the non-linear equations. Thus approximating equation (7) at time $k\Delta t$

$$\begin{aligned} \dot{\mathbf{x}}(k\Delta t) &= \mathbf{A}\mathbf{x}(k\Delta t) + \mathbf{B}\mathbf{u}(k\Delta t) \\ \mathbf{y}(k\Delta t) &= \mathbf{C}\mathbf{x}(k\Delta t) + \mathbf{D}\mathbf{u}(k\Delta t) \end{aligned} \tag{8}$$

and using a backward difference for the time derivative leads to a discrete-time MIMO state-space model of the following form:

$$\begin{aligned} \tilde{\mathbf{x}}_k &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k-1} + \tilde{\mathbf{B}}\tilde{\mathbf{u}}_k \\ \tilde{\mathbf{y}}_k &= \tilde{\mathbf{C}}\tilde{\mathbf{x}}_k + \tilde{\mathbf{D}}\tilde{\mathbf{u}}_k \end{aligned} \tag{9}$$

where subscript k indicates the time $k\Delta t$ and the discrete system matrices are given by

$$\begin{aligned} \tilde{\mathbf{A}} &= (\mathbf{I} - \mathbf{A}\Delta t)^{-1} \\ \tilde{\mathbf{B}} &= (\mathbf{I} - \mathbf{A}\Delta t)^{-1}\mathbf{B}\Delta t \\ \tilde{\mathbf{C}} &= \mathbf{C} \\ \tilde{\mathbf{D}} &= \mathbf{D} \end{aligned} \tag{10}$$

The identification and reduction of the discrete linear system matrices will lead to a discrete-time ROM. A continuous-time ROM of the CFD equations can then be obtained via an inverse transformation [22, 25]. The process starts by finding the linear pulse responses of the CFD scheme. Two methods are considered to extract these responses: the first method linearizes the CFD code, which allows the direct identification of the system matrices and the linear responses; the second method approximates the linear responses as the linear portion of the non-linear responses of the CFD code.

3.2. Method 1—dynamically linear Euler equations

The Euler equations (3) are non-linear and unsteady. In many cases, the unsteadiness in the flow is small. It is then possible to approximate the flow as the sum of a mean or steady flow component and a small-disturbance unsteady component. So if A is an unsteady quantity then it is approximated by

$$A(x, y, t) = A(x, y) + \hat{A}(x, y, t) \tag{11}$$

Substitutions of this form are made for ρ, u, v, p, x, y, V and $\mathbf{D}$. It is assumed that all perturbations and grid speeds can be considered small and the steady solution satisfies Equation (3). Thus only first-order small terms need to be retained and a dynamically linear approximation to the non-linear CFD equations is obtained (see Reference [20] for more details of the derivation). If $\mathbf{q} = [\rho, u, v, p]^{\mathrm{T}}$ then the resulting time-continuous equations can be written as

$$\frac{\mathrm{d}\hat{\mathbf{q}}_{ij}}{\mathrm{d}t} + \frac{1}{\overline{V}_{ij}}(\mathbf{B}^1)_{ij}^{-1}\mathbf{R}_{ij}^1 + \frac{1}{\overline{V}_{ij}}(\mathbf{B}^1)_{ij}^{-1}\mathbf{S}_{ij}^1 = 0 \tag{12}$$

where $(\mathbf{B}^1)^{-1} = \partial\overline{\mathbf{q}}/\partial\overline{\mathbf{W}}$ and

$$\begin{aligned}
\mathbf{R}_{ij}^1 &= \sum_{k=1}^{4}\{\mathbf{B}_k^{2S}\Delta\overline{y}_k - \mathbf{B}_k^{3S}\Delta\overline{x}_k\}\hat{\mathbf{q}}_k - \hat{\mathbf{D}}_{ij} \\
\mathbf{S}_{ij}^1 &= \overline{\mathbf{W}}_{ij}\frac{\mathrm{d}\hat{V}_{ij}}{\mathrm{d}t} + \sum_{k=1}^{4}\{\overline{\mathbf{F}}_k\Delta\hat{y}_k - \overline{\mathbf{G}}_k\Delta\hat{x}_k\} - \sum_{k=1}^{4}\overline{\mathbf{W}}_k\{x_{t_k}\Delta\overline{y}_k - y_{t_k}\Delta\overline{x}_k\}
\end{aligned} \tag{13}$$

Matrices $\mathbf{B}^{2S}$ and $\mathbf{B}^{3S}$ are given by

$$\mathbf{B}^{2S} = \begin{bmatrix} \overline{u} & \overline{\rho} & 0 & 0 \\ \overline{u}^2 & 2\overline{\rho u} & 0 & 1 \\ \overline{vu} & \overline{\rho v} & \overline{\rho u} & 0 \\ \frac{1}{2}(\overline{u}^2 + \overline{v}^2)\overline{u} & (\overline{\rho e} + \overline{p}) + \overline{\rho u}^2 & \overline{\rho vu} & \dfrac{\gamma\overline{u}}{\gamma - 1} \end{bmatrix} \tag{14}$$

$$\mathbf{B}^{3S} = \begin{bmatrix} \overline{v} & 0 & \overline{\rho} & 0 \\ \overline{uv} & \overline{\rho v} & \overline{\rho u} & 0 \\ \overline{v}^2 & 0 & 2\overline{\rho v} & 1 \\ \frac{1}{2}(\overline{u}^2 + \overline{v}^2)\overline{v} & \overline{\rho uv} & (\overline{\rho e} + \overline{p}) + \overline{\rho v}^2 & \dfrac{\gamma\overline{v}}{\gamma - 1} \end{bmatrix} \tag{15}$$

Note that the dissipation $\mathbf{D}_{ij}$ is of the standard Jameson form [26]. This contains a switch that turns on second-order dissipation near shocks. This cannot be linearized. In order to obtain a truely linear dissipation, a switch based on the mean switch value is used. The correct dissipation will not be obtained if the mesh or shock move too far. However, for small perturbations this assumption is reasonable, see Reference [20] for further discussion.

Equation (12) is in linear form and if the output equation is also linearized, the time-continuous linear system (7) approximating the full non-linear system is available in analytic

form. Note however that in the current implementation, it is not necessary to explicitly construct the system matrices. The ERA scheme does not require this information to produce ROMs. A dual-time scheme identical to that used in the non-linear CFD code is implemented, together with a linearized version of the GCL, to solve the system in discrete-time space. To complete the process, all the boundary conditions applied within the non-linear CFD code are linearized [20]. The linear responses of the CFD scheme in discrete-time form are then directly available as the outputs of the dynamically linear CFD code for pulses on each of the system inputs in turn. These can be used to find a discrete-time ROM and also a corresponding continuous-time ROM [25].

3.3. Method 2—extraction of the linear portion of the non-linear response of the CFD code

An alternative approach based on Volterra theory is available that allows the linear responses of the system to be approximately identified directly from the non-linear CFD code. The basic premise of Volterra theory (which is discussed extensively by Silva [17]) is that any non-linear time-continuous system such as the CFD equations (3) can be modelled as an infinite sum of multi-dimensional convolution integrals, of increasing order, involving system kernels. If the equations are put into discrete-time form, then the integrals are replaced by summations. In both continuous and discrete-time space, only the first two terms of the series involving the zeroth order kernel, h_0 (steady term), and the first-order kernel, h_1 (the linear pulse response), are non-zero for linear systems. For 'weakly non-linear' systems, the solution can be represented by a truncated Volterra series for small inputs. If the system may be approximated by a second-order Volterra series, then the first-order kernel of the system, h_1 can be found from

$$h_1 = 2y_1 - \tfrac{1}{2}y_{11} \tag{16}$$

where the notation used is that of Silva [17]. y_1 represents the response of the non-linear system to a pulse input and y_{11} represents the response to a pulse of twice the amplitude. For a non-linear system, h_1 captures some level of the amplitude dependence [17] and is therefore in general different from the purely linear pulse response. However if the non-linear system exhibits approximately linear behaviour to small inputs, then it can be assumed that the non-linear first-order kernel is that of an approximating linear system (7). Thus the linear responses can be found from two responses of the non-linear code.

4. SYSTEM REDUCTION

To simplify the process of obtaining a ROM of the discrete-time system (9), the system output is modified. The matrices $\mathbf{D}$ and thus $\tilde{\mathbf{D}}$ will be known (as the output equation is user specified), so both the continuous system and the discrete approximation define a modified output by subtracting the term $\mathbf{Du}(t)$ from the continuous output and the term $\tilde{\mathbf{D}}\tilde{\mathbf{u}}$ from the discrete output equation. The modified discrete system is then

$$\begin{aligned} \tilde{\mathbf{x}}_k &= \tilde{\mathbf{A}}\tilde{\mathbf{x}}_{k-1} + \tilde{\mathbf{B}}\tilde{\mathbf{u}}_k \\ \tilde{\mathbf{y}}_k^m &= \tilde{\mathbf{y}}_k - \tilde{\mathbf{D}}\tilde{\mathbf{u}}_k = \tilde{\mathbf{C}}\tilde{\mathbf{x}}_k \end{aligned} \tag{17}$$

Then if $\tilde{\mathbf{x}}_{-1} = 0$, it follows from (17) that

$$\tilde{\mathbf{y}}_l^m = [\mathbf{H}_l, \mathbf{H}_{l-1}, \ldots, \mathbf{H}_2, \mathbf{H}_1, \mathbf{H}_0] \begin{bmatrix} \tilde{\mathbf{u}}_0 \\ \tilde{\mathbf{u}}_1 \\ \vdots \\ \tilde{\mathbf{u}}_{l-2} \\ \tilde{\mathbf{u}}_{l-1} \\ \tilde{\mathbf{u}}_l \end{bmatrix} \tag{18}$$

where the sequence $\mathbf{H}_k, k = 0, \infty$ is given by

$$\{\mathbf{H}_0, \mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_k, \ldots\} = \{\tilde{\mathbf{C}}\tilde{\mathbf{B}}, \tilde{\mathbf{C}}\tilde{\mathbf{A}}\tilde{\mathbf{B}}, \ldots, \tilde{\mathbf{C}}\tilde{\mathbf{A}}^k\tilde{\mathbf{B}}, \ldots\} \tag{19}$$

and is called the Markov sequence, the weighting sequence or the impulse-response sequence of the discrete system. The forced response of the discrete system is uniquely determined by its Markov sequence and the input. Thus any two systems with identical Markov sequences have identical forced responses for the same input. Note that for a system of rank n the sequence $\mathbf{H}_k, k = 0, 2n$ defines the forced response exactly.

The system realization method described by Juang and Pappa [6] relies on being able to construct the generalized Hankel matrix whose entries are terms of the form

$$\tilde{\mathbf{C}}\tilde{\mathbf{A}}^k\tilde{\mathbf{B}} \tag{20}$$

for $k \geqslant 0$. Now these terms are equal to the terms of the Markov sequence $\mathbf{H}_k$. Each matrix $\mathbf{H}_k$ is composed of columns, which are the outputs for a unit sample pulse input on each input separately, i.e. the ith column is the output vector at time k for a unit sample input in the ith component of $\tilde{\mathbf{u}}$, with all other entries of $\tilde{\mathbf{u}}$ set to zero, see Reference [27]. This amounts to setting $\tilde{\mathbf{u}}_0 = I$ and all the other input matrices $\tilde{\mathbf{u}}_k$ to null matrices.

The Hankel matrix in this case is the $r \times s$ block matrix given by

$$\mathbf{H}_{rs}(k) = \begin{bmatrix} \mathbf{H}_k & \mathbf{H}_{k+1} & \mathbf{H}_{k+2} & \ldots & \mathbf{H}_{k+s-1} \\ \mathbf{H}_{k+1} & \mathbf{H}_{k+2} & \mathbf{H}_{k+3} & \ldots & \mathbf{H}_{k+s} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathbf{H}_{k+r-1} & \mathbf{H}_{k+r} & \mathbf{H}_{k+r+1} & \ldots & \mathbf{H}_{k+s+r-2} \end{bmatrix} \tag{21}$$

This matrix can be used to realize and reduce the system matrices via ERA [6].

If there are p outputs and m inputs to the system then each of the Markov parameters is of size $p \times m$. Thus the size of the Hankel matrix is $rp \times sm$. Then the singular-value decomposition (SVD) for $k = 0$ is given by

$$\mathbf{H}_{rs}(0) = \mathbf{U}\mathbf{W}\mathbf{V}^{\mathrm{T}} \tag{22}$$

where $\mathbf{W}$ is an $sm \times sm$ diagonal matrix whose diagonal entries are called singular values which are either positive or zero, $\mathbf{U}$ is $rp \times sm$ and $\mathbf{V}$ is a $sm \times sm$ matrix. The elements of

$\mathbf{W}$ are arranged in size order i.e. $[w(1,1) > w(2,2) > w(3,3) \cdots]$. The rank of the ROM of the system is then determined by the number of elements of $\mathbf{W}$ which are larger than some desired accuracy or by taking into account only the n largest singular values in $\mathbf{W}$. Then matrix $\mathbf{H}_{rs}(0)$ can be approximated as

$$\mathbf{H}_{rs}(0) = \mathbf{P}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}} \tag{23}$$

where the matrices $\mathbf{U}, \mathbf{W}, \mathbf{V}$ have been reduced in size by deleting unnecessary columns and rows as appropriate. The reduced matrix from $\mathbf{U}$ is $\mathbf{P} : rp \times n$, the reduced matrix from $\mathbf{W}$ is $\mathbf{\Lambda} : n \times n$ and the reduced matrix from $\mathbf{V}$ is $\mathbf{Q} : sm \times n$.

It is then possible to show [6] that a realization is

$$\begin{aligned} \tilde{\mathbf{A}} &= \mathbf{\Lambda}^{-1/2}\mathbf{P}^{\mathrm{T}} H_{rs}(1)\mathbf{Q}\mathbf{\Lambda}^{-1/2} \\ \tilde{\mathbf{B}} &= \mathbf{\Lambda}^{1/2}\mathbf{Q}^{\mathrm{T}}\mathbf{E}_m \\ \tilde{\mathbf{C}} &= \mathbf{E}_p^{\mathrm{T}}\mathbf{P}\mathbf{\Lambda}^{1/2} \end{aligned} \tag{24}$$

where $\mathbf{E}_p^{\mathrm{T}} = [\mathbf{I}_p, \mathbf{0}_p, \mathbf{0}_p, \ldots, \mathbf{0}_p]$ has size $p \times rp$ and $\mathbf{E}_m^{\mathrm{T}} = [\mathbf{I}_m, \mathbf{0}_m, \mathbf{0}_m, \ldots, \mathbf{0}_m]$ has size $m \times sm$. Note that the triple $[\tilde{\mathbf{A}}, \tilde{\mathbf{B}}, \tilde{\mathbf{C}}]$ is not unique and that for any non-singular matrix $\mathbf{T}$ the triple $[\mathbf{T}\tilde{\mathbf{A}}\mathbf{T}^{-1}, \mathbf{T}\tilde{\mathbf{B}}, \tilde{\mathbf{C}}\mathbf{T}^{-1}]$ is also a realization.

The above scheme is used to get a discrete-time ROM of the CFD code. However in discrete form, the model is restricted to problems with a fixed time step, which means that the discrete ROM cannot be accurately applied to structural models with discrete non-linearities (such as freeplay in a control surface) since any aerodynamic model must capture the 'switching' points between discrete regions [28] or unphysical limit cycle behaviour may be introduced into the solution [29]. A time continuous ROM does not have this restriction.

The simplest method to obtain a continuous-time ROM is to invert the transformation used to get from the continuous to the discrete system (10) before system reduction. This is applied to the reduced system matrices and leads to a ROM of the continuous-time system [25].

5. TEST CASE INPUT CHANNELS

The methods described above will be applied to the identification of a continuous ROM for the flow about an aerofoil with trailing edge flaps that can undergo linearized pitch motions. For this motion, the position of a point on the aerofoil surface at any time is

$$\begin{aligned} x(t) &= \bar{x} + \Delta x(t) \\ y(t) &= \bar{y} + \Delta y(t) \end{aligned} \tag{25}$$

where $\bar{x}, \bar{y}$ are the mean steady position of the point on the aerofoil surface and $\Delta x, \Delta y$ are the displacements of the point at time t. The displacements are given by

$$\begin{aligned} \Delta x &= (\bar{y} - b_c)\alpha_{\mathrm{amp}}(t) + S(h_x)(\bar{y} - h_y)\delta(t) \\ \Delta y &= -(\bar{x} - a_c)\alpha_{\mathrm{amp}}(t) - S(h_x)(\bar{x} - h_x)\delta(t) \end{aligned} \tag{26}$$

Table I. Channel inputs to the linearized Euler equations for linearized pitch and linearized flap motions.

	Channel 1 α $t=k\Delta t$	Channel 2 δ $t=k\Delta t$	Channel 3 $\dot{\alpha}$ $t=k\Delta t$	Channel 4 $\dot{\delta}$ $t=k\Delta t$	All channels $\alpha,\delta,\dot{\alpha},\dot{\delta}$ $t\neq k\Delta t$
Δx	$(\overline{y}-b_c)c_1$	$(\overline{y}-h_y)c_2$	0.	0.	0.
Δy	$-(\overline{x}-a_c)c_1$	$-(\overline{x}-h_x)c_2$	0.	0.	0.
x_t	0.	0.	$(\overline{y}-b_c)c_3$	$(\overline{y}-h_y)c_4$	0.
y_t	0.	0.	$-(\overline{x}-a_c)c_3$	$-(\overline{x}-h_x)c_4$	0.

$\Delta x, \Delta y$ are displacements and x_t, y_t are grid speeds of a point on the aerofoil.

and the grid speed of the point on the aerofoil surface x_t, y_t are given by

$$\begin{aligned}\dot{x} &= (\overline{y}-b_c)\dot{\alpha}_{\text{amp}}(t) + S(h_x)(\overline{y}-h_y)\dot{\delta(t)}\\ \dot{y} &= -(\overline{x}-a_c)\dot{\alpha}_{\text{amp}}(t) - S(h_x)(\overline{x}-h_x)\dot{\delta(t)}\end{aligned} \tag{27}$$

where (h_x, h_y) is the hinge location and

$$S(h_x)=0 \quad \overline{x}<h_x$$
$$S(h_x)=1 \quad \overline{x}\geqslant h_x$$

There are four associated input channels for this flow. The pulse inputs are given in Table I where $c_1,\ldots,c_4$ are scaling constants used to avoid divergence of the numerical scheme.

6. RESULTS

The test cases shown here are for a NACA 0012 aerofoil. The grid used for the calculations is of size 191×36, this means that the number of unknowns is 26 600 (ncells $\times$ 4 solution quantities). A steady calculation is performed for the aerofoil at incidence $\alpha=0°$. The steady flow pressure distributions at freestream Mach numbers $M=0.7$ and 0.80 are shown in Figure 1. These Mach numbers have been selected to illustrate flows without and with shockwaves.

ROMs are generated using ERA for each Mach number. The pulse responses are calculated using the two methods described above. The aim of this paper is to compare these methods for linear response identification and the subsequent impact on the ROMs produced by ERA. To enable this comparison, the individual steps in the formation of the ROM are considered separately, so that areas of difference can be highlighted; these steps are the linear response calculation, the SVD of the Hankel matrix and the eigenvalues of the ROMs. Finally, forced sinusoidal linear pitching motions are compared to show the overall properties of the ROMs for a range of time steps.

6.1. Linear responses

The procedure to generate continuous ROMs starts with the calculation of the linear responses of the CFD scheme for each input. The non-dimensional time step used in the calculations is Δt_1.

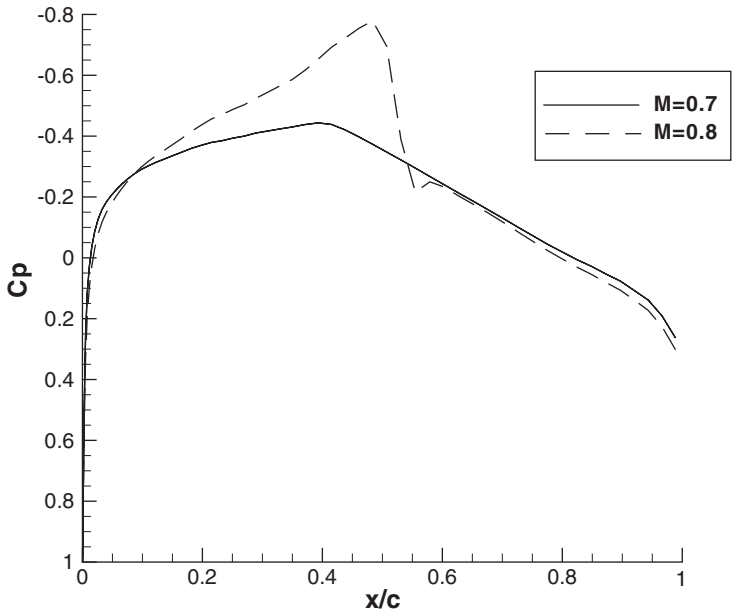

Figure 1. Steady pressure distributions, $M = 0.7, 0.8$.

Table II. Channel inputs to the linearized Euler equations for linearized pitch and linearized flap motions.

M	Δt_1	c_1	c_2	c_3	c_4
0.7	0.2608	$\pi/180$	$\pi/180$	$0.335\pi/180$	$0.335\pi/180$
0.8	0.2282	$\pi/180$	$\pi/180$	$0.382\pi/180$	$0.382\pi/180$

For the dynamically linear CFD code the time step and the input scaling constants for the two Mach numbers under consideration are shown in Table II. The pulse responses are calculated using 150 subiterations per real time step.

For the non-linear code, two sets of pulses are needed to extract an approximation to the linear response. The extracted linear response will depend on the amplitude of the non-linear pulses used to calculate it. Therefore non-linear pulse responses corresponding to amplitudes $2c_1$, c_1, $c_1/2$, $c_1/4$ and $c_1/8$ (where c_1 is as used for the linear responses) are calculated. This means linear pulse responses corresponding to c_1, $c_1/2$, $c_1/4$ and $c_1/8$ can be extracted. These can then be rescaled for comparison to the amplitude of the responses of the dynamically linear code.

To compare the linear responses constructed from the non-linear responses to those from the dynamically linear code, difference terms are used. For example, for the lift perturbation, the difference is given by

$$\text{Diff}_{C_l} = \frac{(\hat{C}_{l,2} - \hat{C}_{l,1})}{\max|\hat{C}_{l,1}|} \tag{28}$$

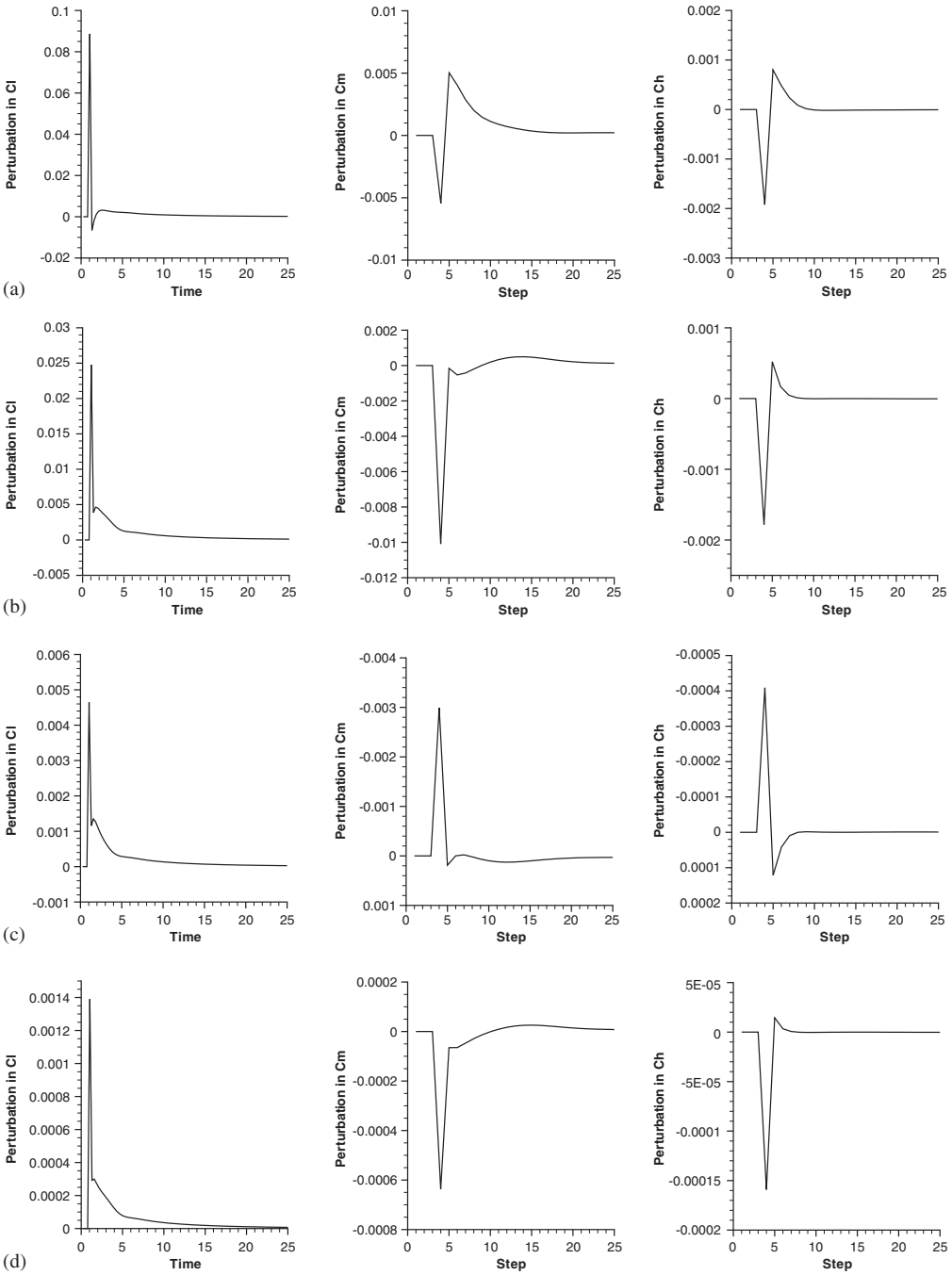

Figure 2. Pulse responses from linearized code, $M = 0.7$: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

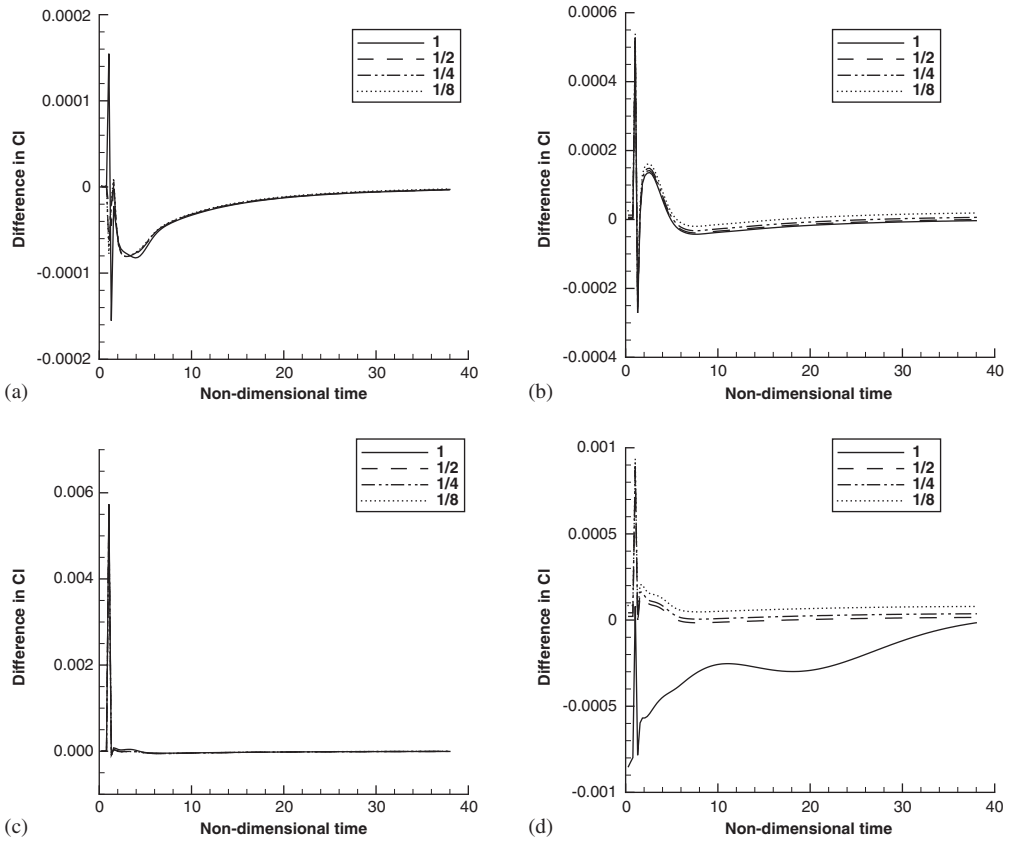

Figure 3. Differences in lift perturbation at $M = 0.7$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

where the subscript 1 corresponds to the dynamically linear method and subscript 2 corresponds to the pulses constructed using non-linear responses. The maximum value in the dynamically linear response history is used to normalize the values obtained.

It should be noted that the internal calculations of the non-linear code work on the actual values of lift, moment and hinge moment. The required outputs **y** (see Equation (6)) are calculated by subtraction of the mean values. This means that any rounding error effects will be amplified as the pulse size decreases.

6.1.1. $M = 0.7$. The linear pulse responses calculated using the dynamically linear CFD code are shown in Figure 2. Figure 3 shows the differences in lift perturbation for linear responses extracted from non-linear responses at different amplitudes. Note that the switch in dissipation takes its usual variable form and 150 subiterations per real time step are used. It can be seen that for pulses in $\alpha, \dot{\alpha}$ and δ the initial difference dies away quickly. For $\dot{\delta}$, the difference for the linear response calculated using pulses of amplitude c_1 and $2c_1$ exhibits large differences for a significant part of the time history. This is a different behaviour to that shown for linear responses extracted from smaller amplitude non-linear responses.

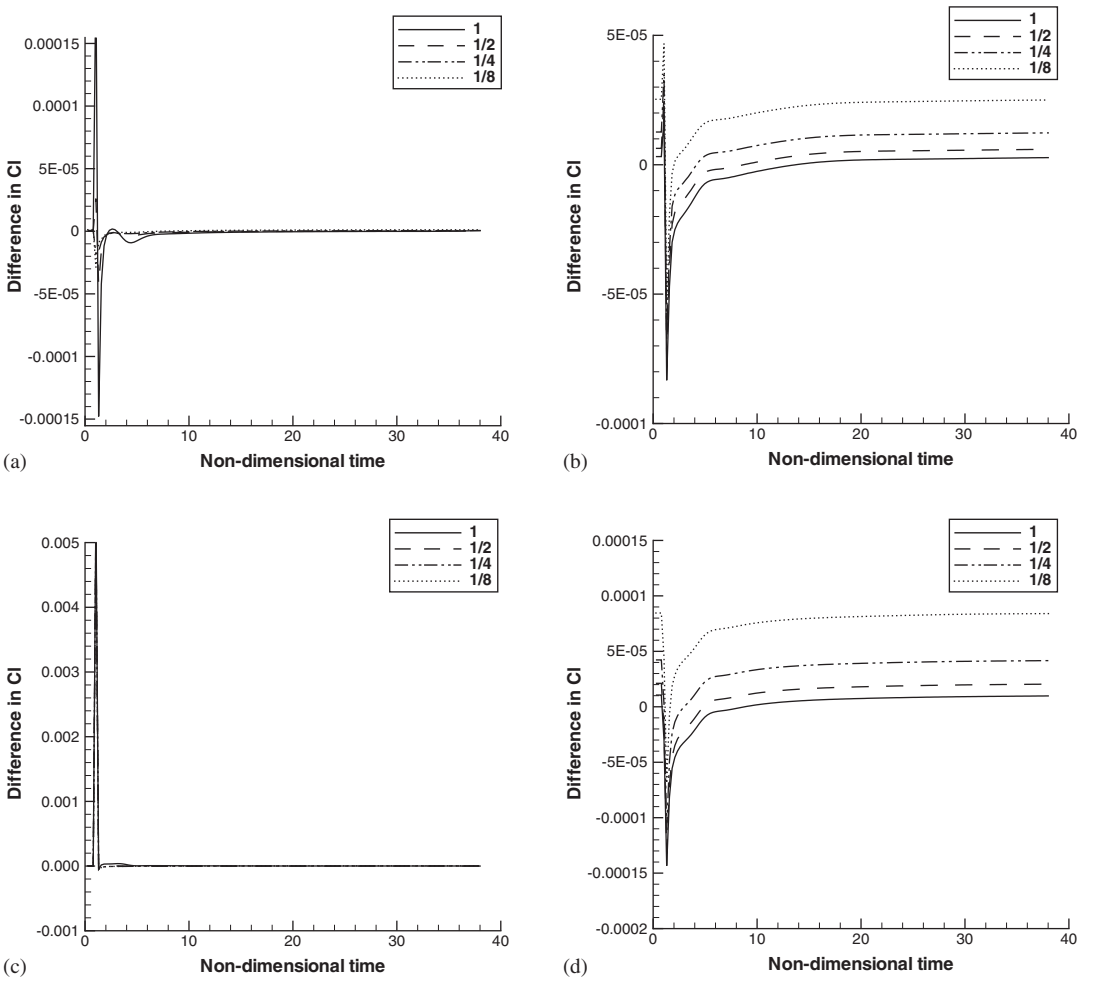

Figure 4. Differences in lift perturbation at $M = 0.7$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

The dissipation switch was then fixed at its mean value (like in the dynamically linear responses) to see how this would effect the responses. Figure 4 shows the differences in lift perturbation in this case. The differences in α and δ show only a very slight change compared to the variable switch results. For $\dot{\alpha}$ and $\dot{\delta}$, the reduction in the magnitude of the differences is much higher. Also in the case of $\dot{\delta}$, the behaviour of the difference for the linear response constructed from the c_1 and $2c_1$ non-linear pulses is similar to that for those from smaller pulses. This suggests the non-linear pulse corresponding to $2c_1$ is exhibiting significant non-linear behaviour due to the variable switch as the shock moves. When the non-linear effects of the variable switch are eliminated from the $2c_1$ non-linear pulse, the differences in the linear responses for δ and $\dot{\delta}$ are larger for those calculated from the smallest non-linear pulses than those using the largest non-linear responses. The magnitude of the forces for the variations in flap speed are very small and there is a precision issue, as discussed in Section 6.1. In this

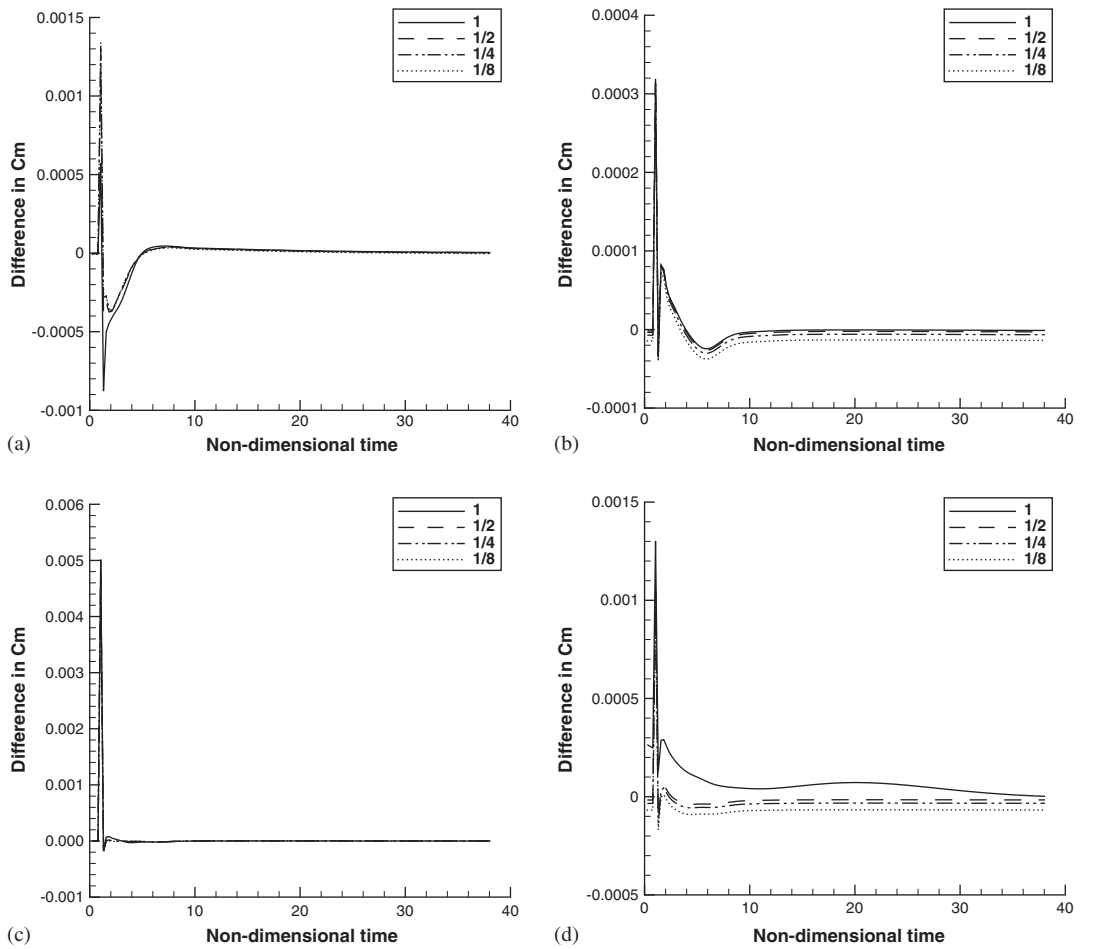

Figure 5. Differences in pitch moment perturbation at $M = 0.7$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

figure, the long-term differences are proportional to 1/amplitude of the original linear response calculated. This suggests the rescaling, used to produce amplitude c_1 responses, is magnifying a fixed precision error.

Differences in $\hat{C}_m$ and $\hat{C}_h$ are shown in Figures 5–8. The maximum differences are small in all cases. The number of subiterations was doubled to check convergence. This had a negligible impact on the difference levels, so further results are not included here.

6.1.2. $M = 0.8$. The linear pulse responses calculated using the dynamically linear CFD code are shown in Figure 9. Figures 10 and 11 show the differences in lift perturbation for linear responses extracted from the non-linear responses with variable and fixed switches, respectively. The differences take longer to die away in most cases compared to the $M = 0.7$ results, where the absence of the shock makes the non-linear effects less significant.

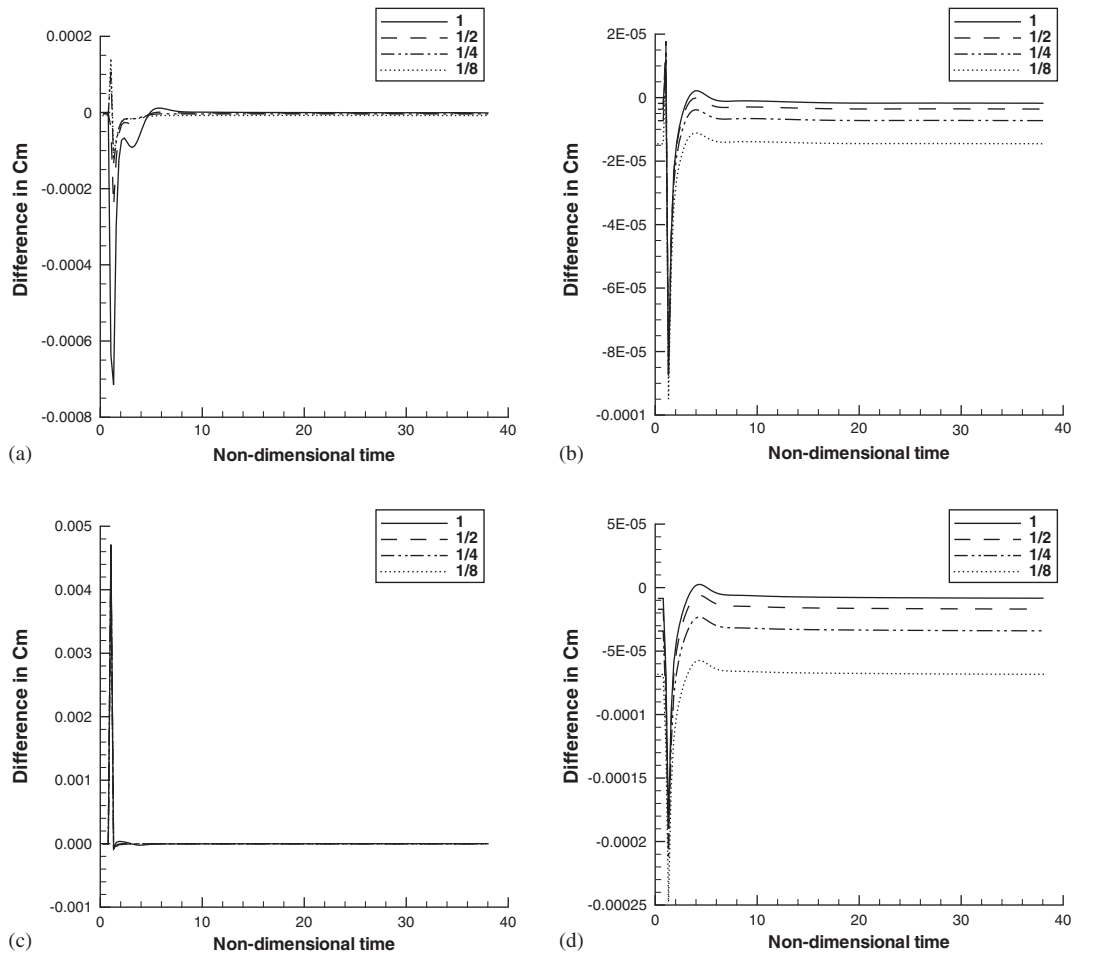

Figure 6. Differences in pitch moment perturbation at $M = 0.7$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

Differences in $\hat{C}_m$ and $\hat{C}_h$ are shown in Figures 12–15. The number of subiterations was doubled to check convergence. Again this had a negligible impact on the difference levels and so further results are not included here.

6.2. *Singular values of the Hankel matrix*

The first step in the use of ERA to find a ROM is the construction and SVD of the Hankel matrix. This yields the singular values of the matrix. As the size of the Hankel matrix is increased, more singular values can be identified and the values for the larger singular values converge. This is illustrated for the dynamically linear responses at $M = 0.7$ and 0.8 in Figures 16 and 17, respectively, where singular values for Hankel matrices constructed using 20, 40 and 80 Markov parameters are shown.

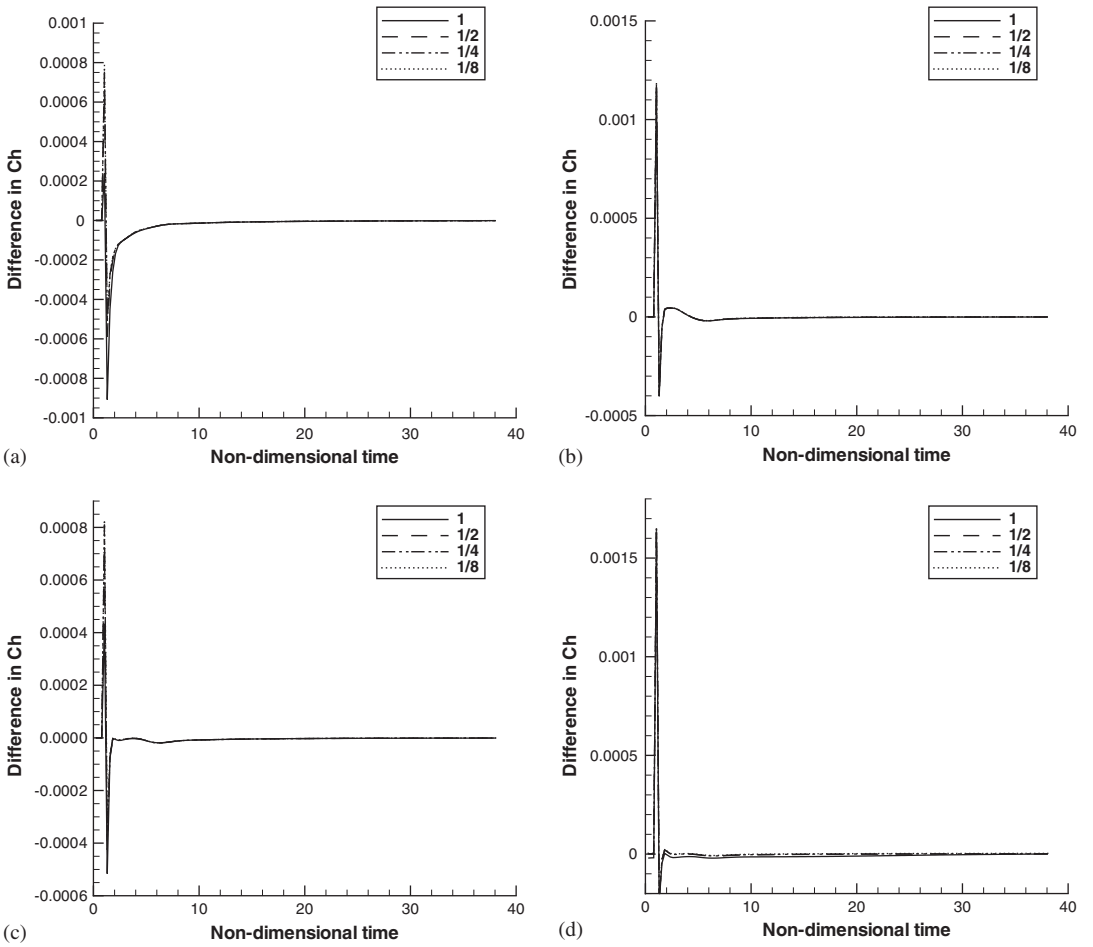

Figure 7. Differences in hinge moment perturbation at $M = 0.7$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

The differences between the singular values for the dynamically linear responses and those extracted from non-linear pulses of various amplitudes for a particular size of Hankel matrix are shown in Figures 18 and 19 for $M = 0.7$ and 0.8, respectively. For $M = 0.7$, the largest singular values for Hankel matrices constructed using linear responses extracted from the non-linear CFD code are close to those constructed using the dynamically linear code. The smaller singular values of the Hankel matrices constructed using different amplitude pulses from the non-linear code are all similar, but there is a small difference from those based on the dynamically linear code pulses. For $M = 0.8$, the singular values show similar behaviour when a fixed switch is used, but greater differences are exhibited when a variable switch is used. The non-linear behaviour in this case is greater because of the shock.

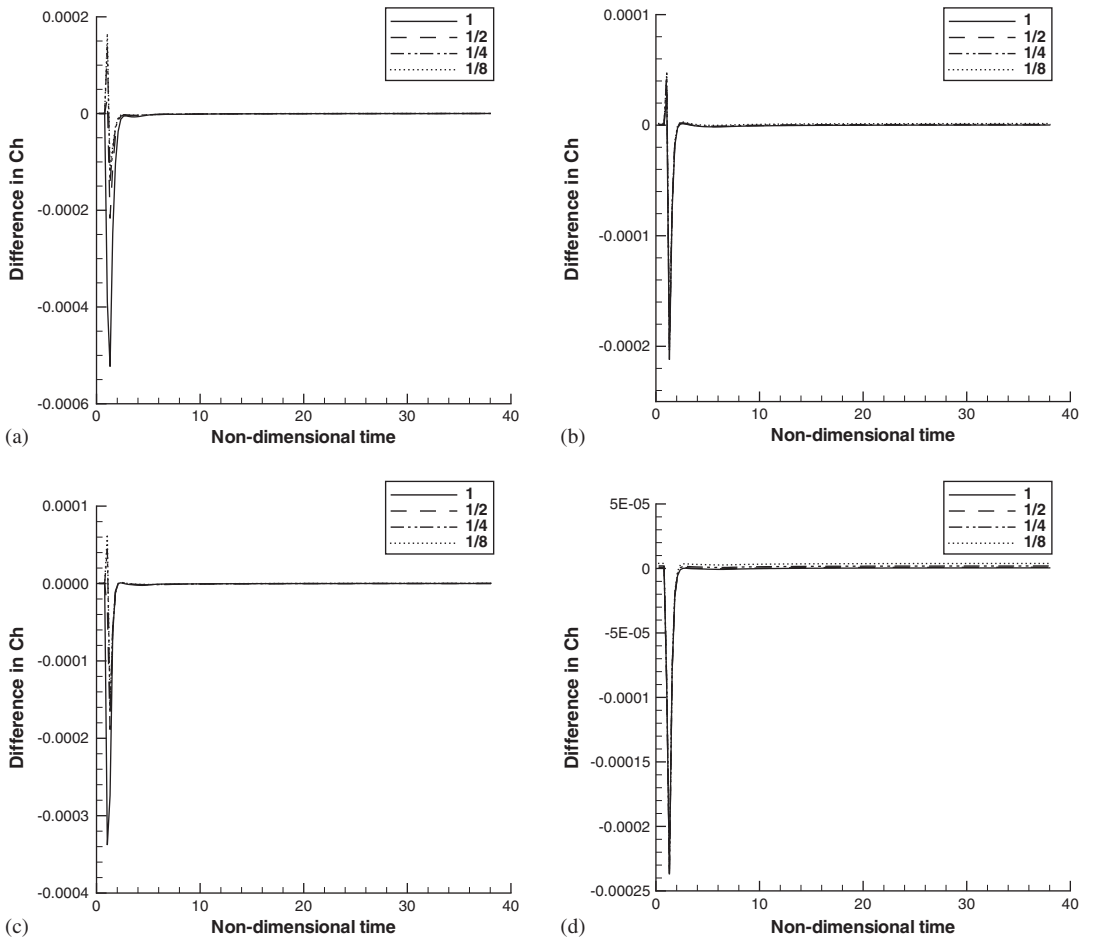

Figure 8. Differences in hinge moment perturbation at $M = 0.7$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

6.3. *ROM generation and eigenvalues*

Once the SVD of the Hankel matrix is known, ERA can generate a discrete time ROM using Equation (24). The discrete time ROM is stable when each eigenvalue λ^{dsc} of the reduced-order system matrix $\tilde{\mathbf{A}}$ satisfies

$$|\lambda^{\mathrm{dsc}}| < 1 \tag{29}$$

This model is a discrete time ROM of the CFD code for a fixed time-step of Δt_1. If a continuous time ROM is required, so that the time-step may be altered, then the inverse of the transformation given in Equation (10) is used. Note that not all stable discrete models map to stable continuous models. All the eigenvalues of the continuous system matrix $\mathbf{A}$ must have negative real part for stability. The corresponding eigenvalues of the discrete scheme

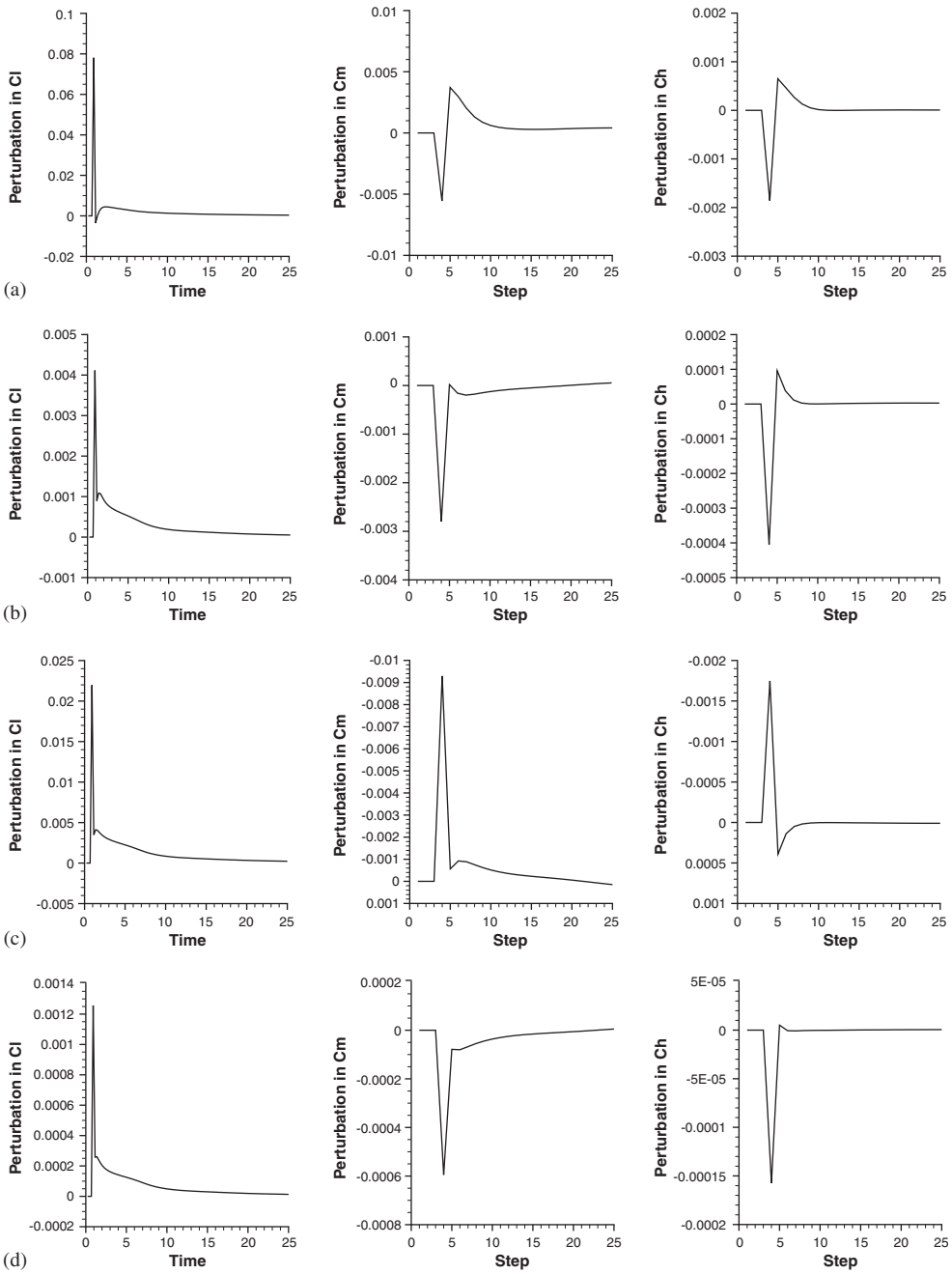

Figure 9. Pulse responses, $M = 0.8$: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

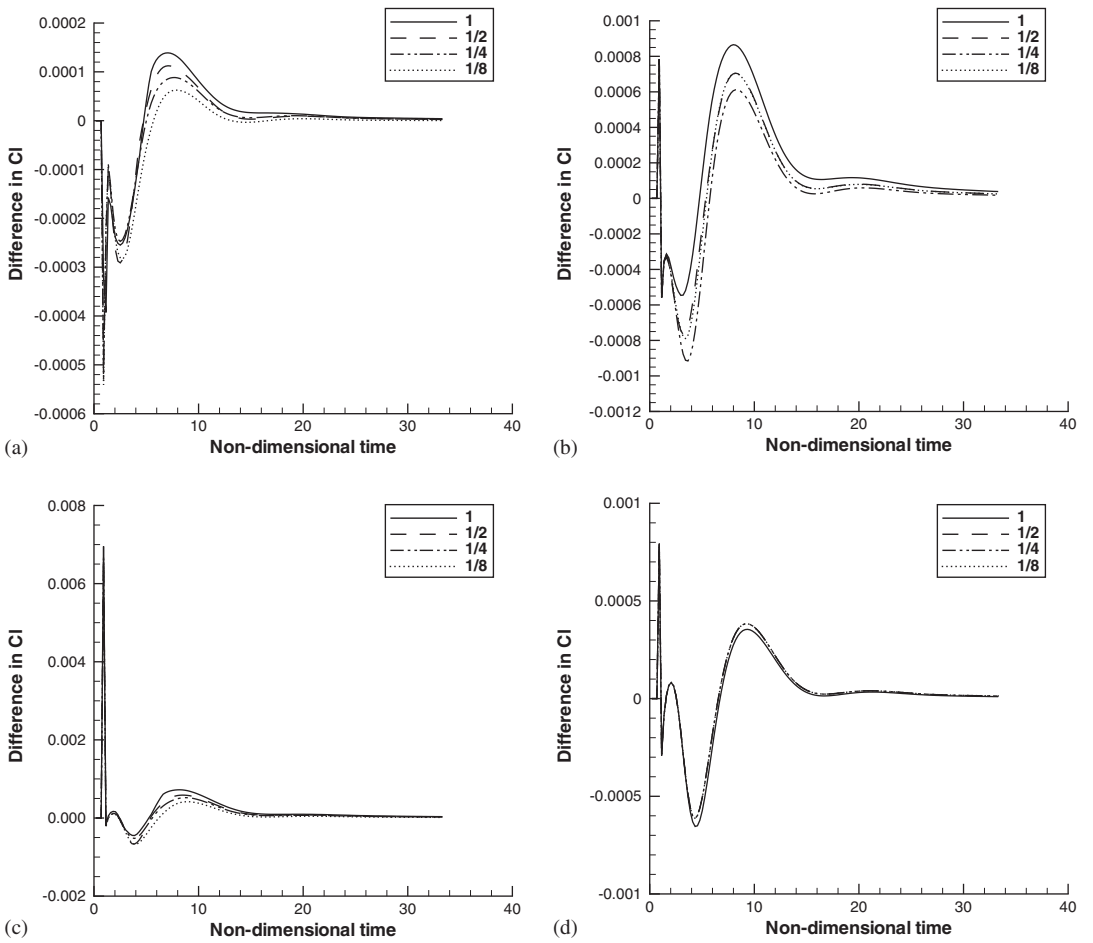

Figure 10. Differences in lift perturbation at $M = 0.8$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

matrix $\tilde{\mathbf{A}}$ for a stable continuous scheme lie within the smaller circle shown in Figure 20. The differences in the values calculated from the linear responses derived from the smallest non-linear pulses is a result of the small magnitude of the outputs being more affected by rounding error. These effects are magnified by the post-processing used to derive the reduced system matrices.

All the Hankel matrices were constructed using 80 Markov parameters. For $M = 0.7$, ROMs of rank 18 were constructed. Figures 21 and 22 show the discrete eigenvalues for the discrete ROMs constructed using pulses of varying amplitude from the non-linear code with variable and fixed switches, respectively. Also included in the plots are the eigenvalues of the discrete ROM constructed from the dynamically linear code (labelled Lin). In this case, there is little difference between the fixed and variable switch plots, which is to be expected given the closeness of the singular values. The eigenvalues for models constructed using different amplitude responses of the non-linear code exhibit small differences. For all

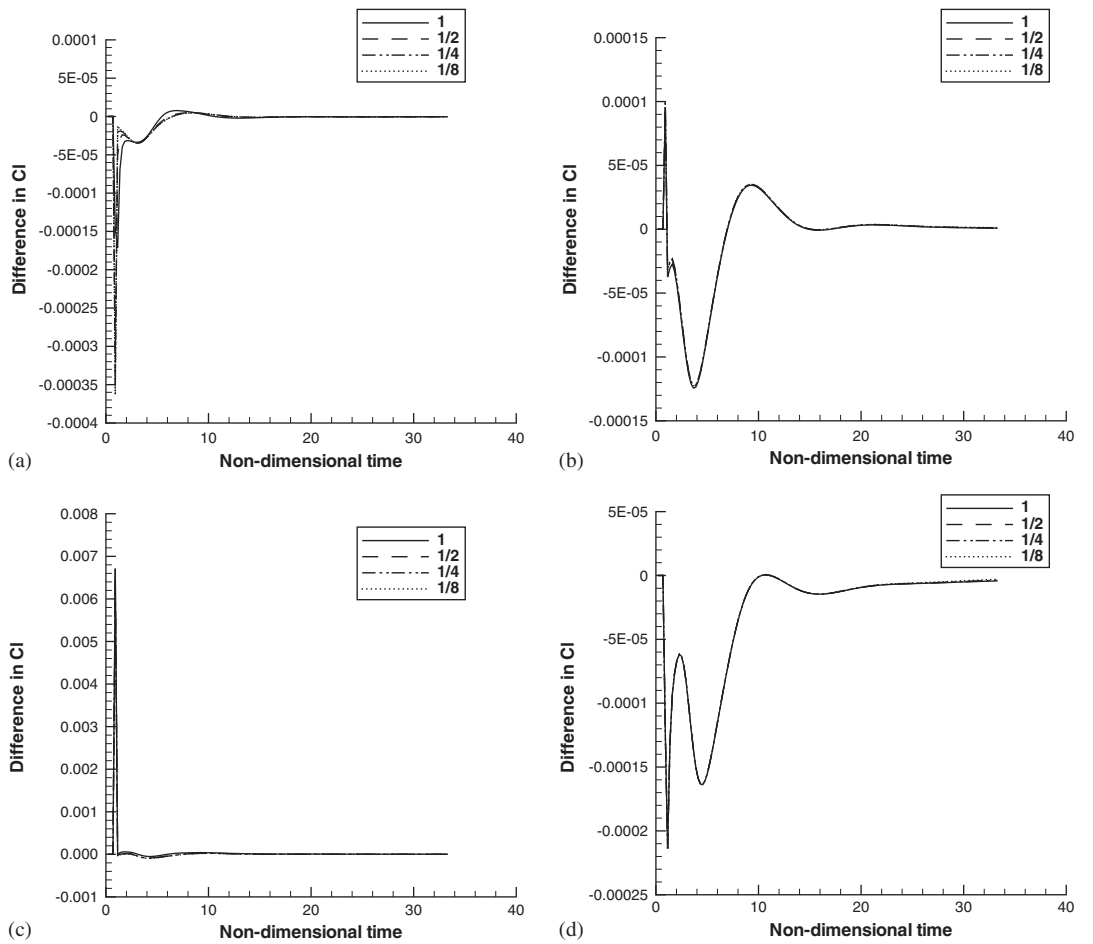

Figure 11. Differences in lift perturbation at $M = 0.8$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

the models, all the eigenvalues are within the circle required to ensure the discrete ROM maps to a stable continuous ROM. For $M = 0.8$, ROMs of rank 20 were constructed. Figures 23 and 24 show the discrete eigenvalues for the discrete ROMs constructed using pulses of varying amplitude from the non-linear code with variable and fixed switches, respectively. In this case, there are large differences between the variable and fixed switch plots. The fixed switch eigenvalues show little variation with pulse amplitude suggesting essentially the same model has been identified by all the non-linear pulses and the dynamically linear responses. In contrast, the variable switch eigenvalues vary significantly with amplitude. In this case, the non-linear effects of the shock motion on the switch are significant and different models are being found depending on the pulse amplitudes. For the fixed switch cases, all the discrete ROMs produced map to stable continuous ROMs. However for the variable switch cases, the discrete ROM produced using the largest amplitude pulses from the non-linear code is not stable. It was not possible to find a discrete ROM of any rank for

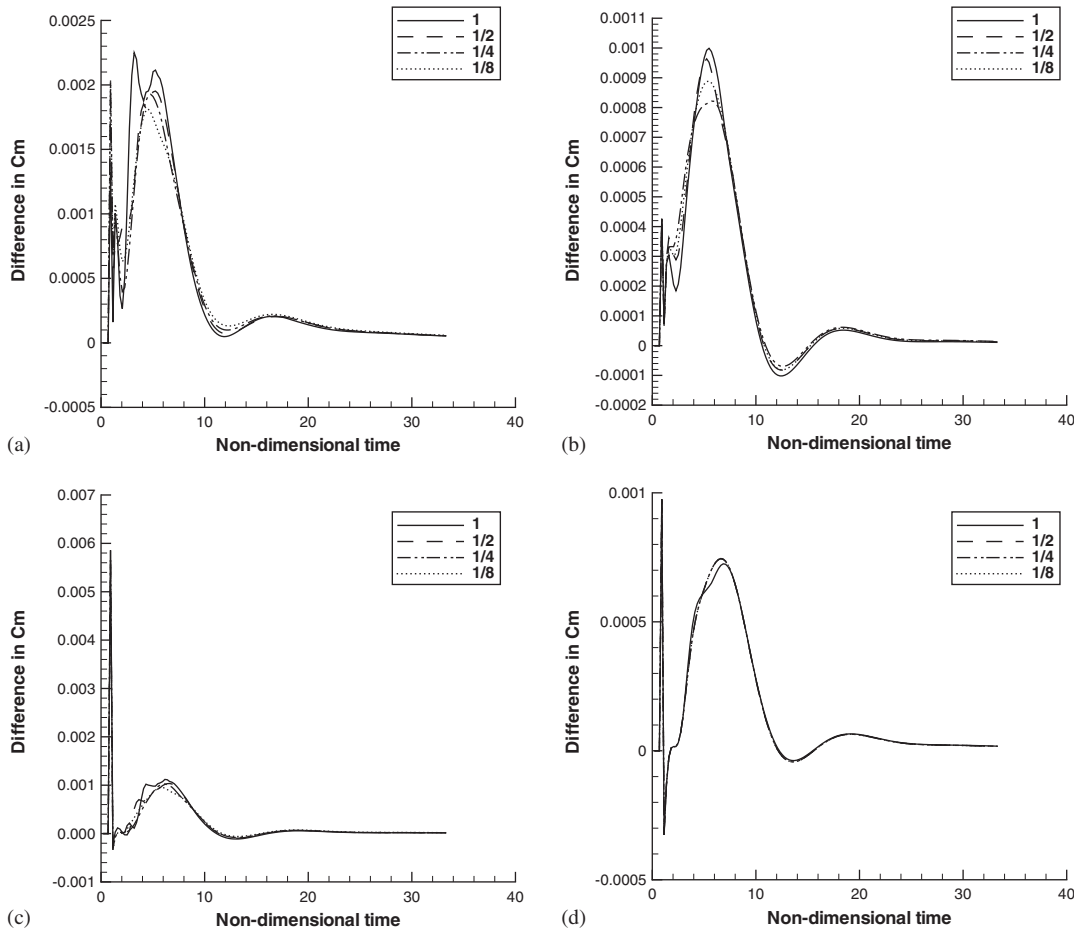

Figure 12. Differences in pitch moment perturbation at $M = 0.8$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

this case that would map to a stable continuous ROM, although stable discrete ROMs can be found.

6.4. *Sinusoidal linear pitch oscillations*

The ROMs were then applied to a sinusoidal linear pitch test case. The linear pitch motion is defined in dimensional variables by

$$\alpha_{\text{amp}}(t) = \alpha_{\max} \sin(\omega t) \tag{30}$$

where ω is related to the non-dimensional reduced frequency, k_{red} via

$$k_{\text{red}} = \frac{\omega c}{2U_\infty} \tag{31}$$

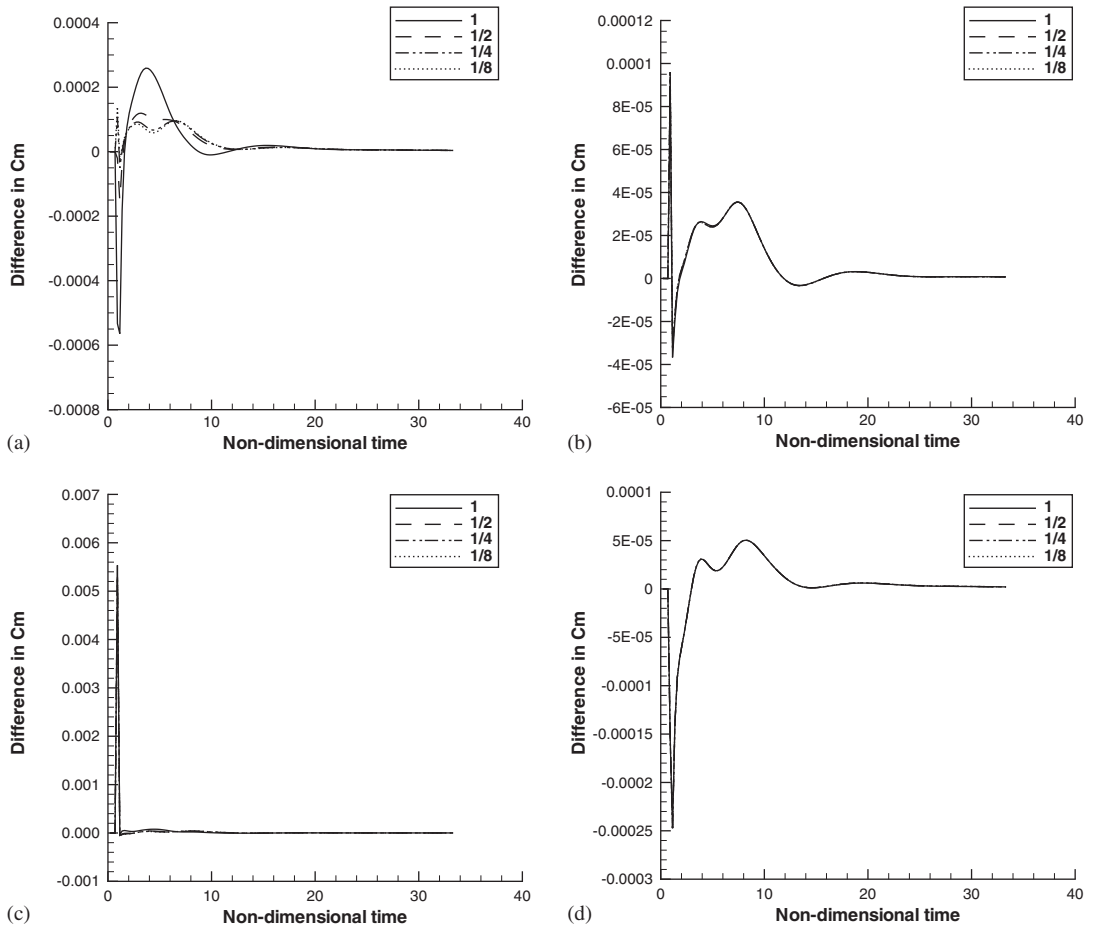

Figure 13. Differences in pitch moment perturbation at $M = 0.8$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

Calculations were performed with $\alpha_{\max} = 0.25^\circ$ at $M = 0.7, 0.8$ for reduced frequencies $k_{\text{red}} = 0.0404, 0.202, 4.04, 10.1$ and 20.2. Each calculation was performed with 72 steps per period. At these frequencies, the non-dimensional time steps are $5\Delta t_1$, Δt_1, $\Delta t_1/20$, $\Delta t_1/50$ and $\Delta t_1/100$, respectively, where the continuous ROM was created using a pulse width of Δt_1.

Solutions for the lift perturbation at $M = 0.7$ are shown in Figures 25 and 26. The plots compare ROM solutions to full non-linear solutions with a variable switch and full linear solutions. The results for ROMs produced from the non-linear codes with variable and fixed switches are very close. These results are also similar, for all frequencies, to those from the dynamically linear code. It can be seen that the long term sinusoidal behaviour is captured by all models up to $k = 10.1$ which corresponds to a time step of $\Delta t_1/50$. However, the initial transient shown by the full-order non-linear and full-order linearized Euler solutions is not captured by the ROMs as frequency increases. The pitch moment and hinge moment

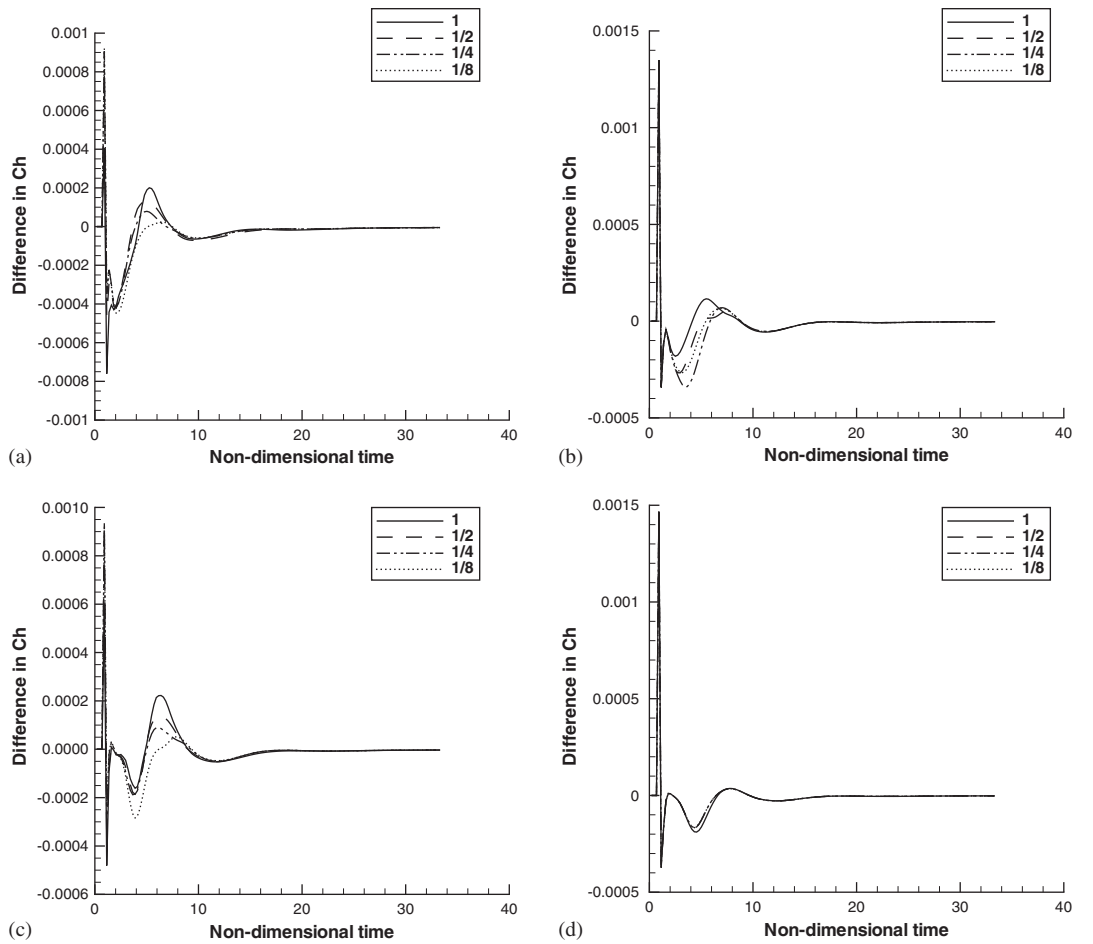

Figure 14. Differences in hinge moment perturbation at $M = 0.8$, variable switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

perturbation results for ROMs from the variable switch non-linear code are shown in Figures 27 and 28, respectively. Like the lift perturbation results, the plots for ROMs for the fixed switch code exhibit excellent agreement like the lift perturbation results and so are not shown. The reason that the methods breakdown for the highest frequency is that the current approach identifies the dominant eigenvalues of $(I - A\Delta t)^{-1}$, see Equation (10). This biases the ROM to model a low-frequency range. Thus as the frequency is increased, a point is reached where the eigenvalues driving the response have not been captured in the ROM. Even at lower input frequencies, the initial transients are not well predicted, because they include high-frequency components that again are not modelled by the ROMs.

Solutions at $M = 0.8$ are shown in Figures 29–34. The results for ROMs produced from the non-linear codes with variable and fixed switches are very close except for the ROM constructed from the largest amplitude pulses of the variable switch code, which maps to an

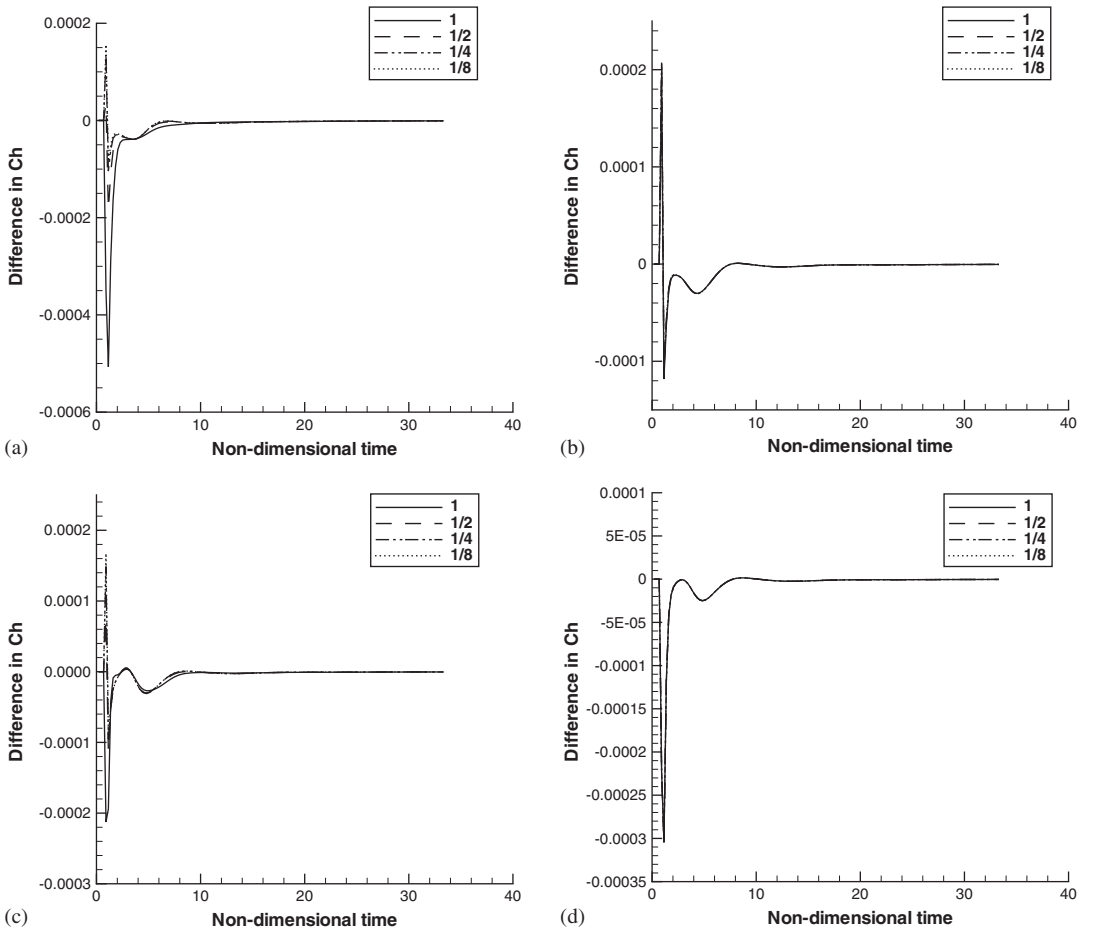

Figure 15. Differences in hinge moment perturbation at $M = 0.8$, fixed switch: (a) α; (b) δ; (c) $\dot{\alpha}$; and (d) $\dot{\delta}$.

unstable continuous ROM. In this case, at the higher frequencies when a time step smaller than that used to produce the ROM is used, the continuous ROM maps to an unstable discrete ROM and the solutions become unstable; at the lowest frequency, with a time step larger than that used to generate the ROM, the solution is stable and in good agreement with the other solutions. It can be seen that the long-term sinusoidal behaviour is again captured by all the stable models up to $k = 10.1$ and that the initial transient shown by the full-order non-linear and full-order linearized Euler solutions is not captured by the ROMs. Note also that the results from ROMs constructed using the variable switch non-linear pulses show a small amplitude dependence at the highest frequencies. This is because the non-linear effects in the code make the assumption of weak non-linearity invalid, even for relatively small motions when a shock is present. This effect was not observed at $M = 0.7$ when no shock was present.

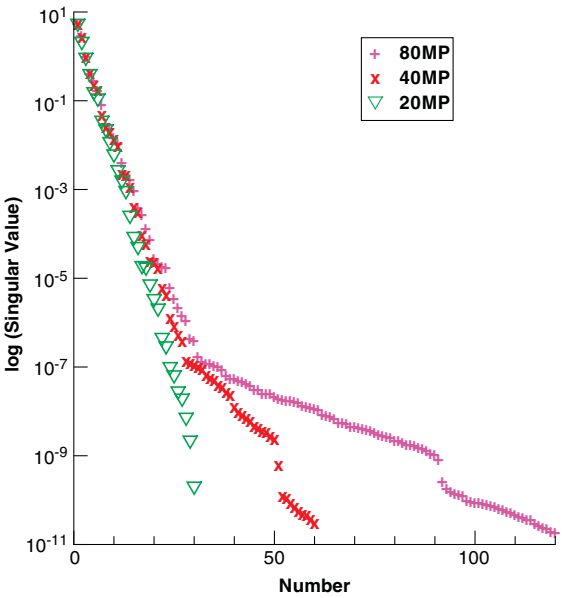

Figure 16. Singular values for Hankel matrix constructed with 20, 40 or 80 Markov parameters from the dynamically linear code, $M = 0.7$.

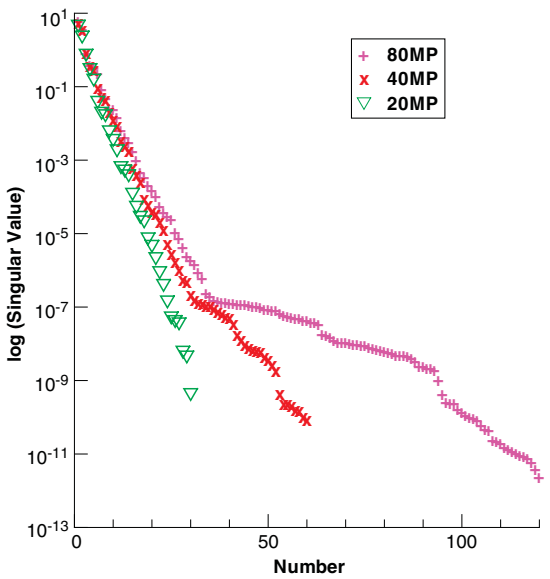

Figure 17. Singular values for Hankel matrix constructed with 20, 40 or 80 Markov parameters from the dynamically linear code, $M = 0.8$.

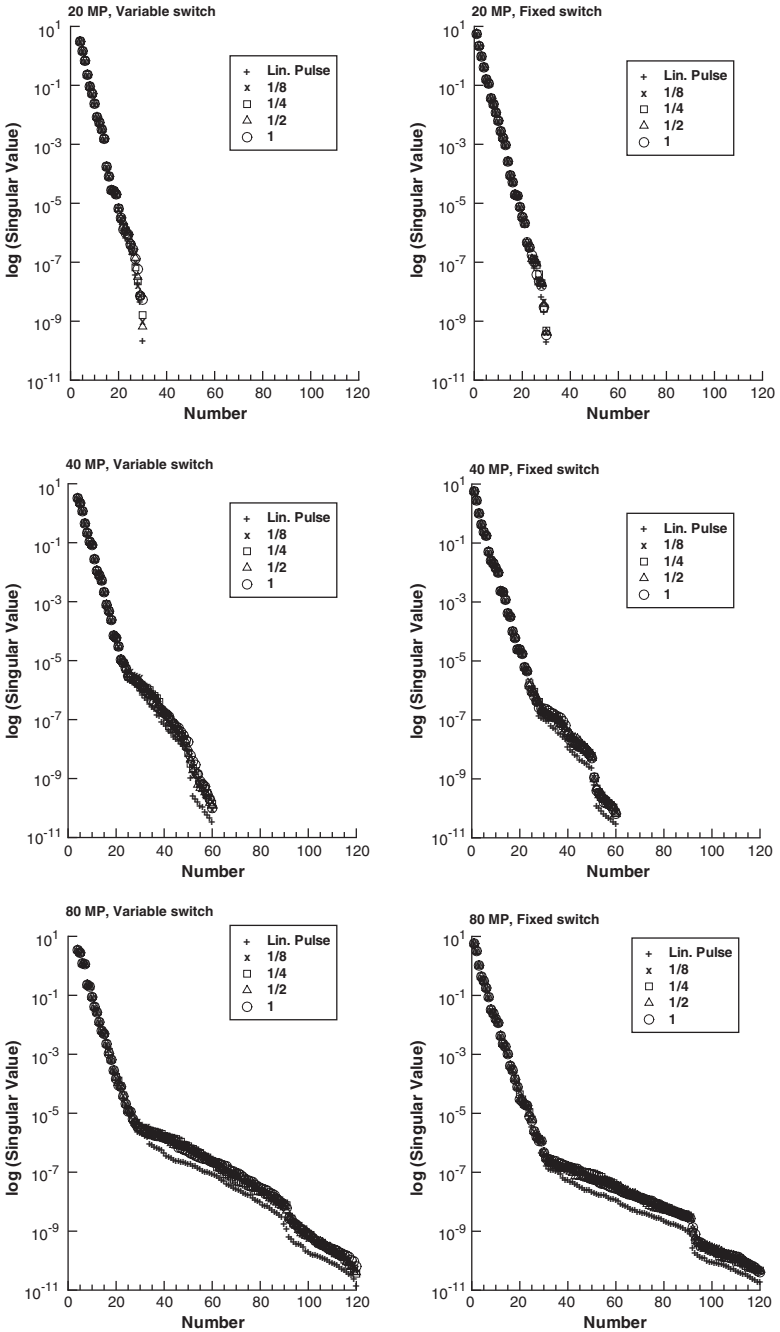

Figure 18. Singular values for Hankel matrices constructed with pulses from the dynamically linear code and non-linear code, $M = 0.7$.

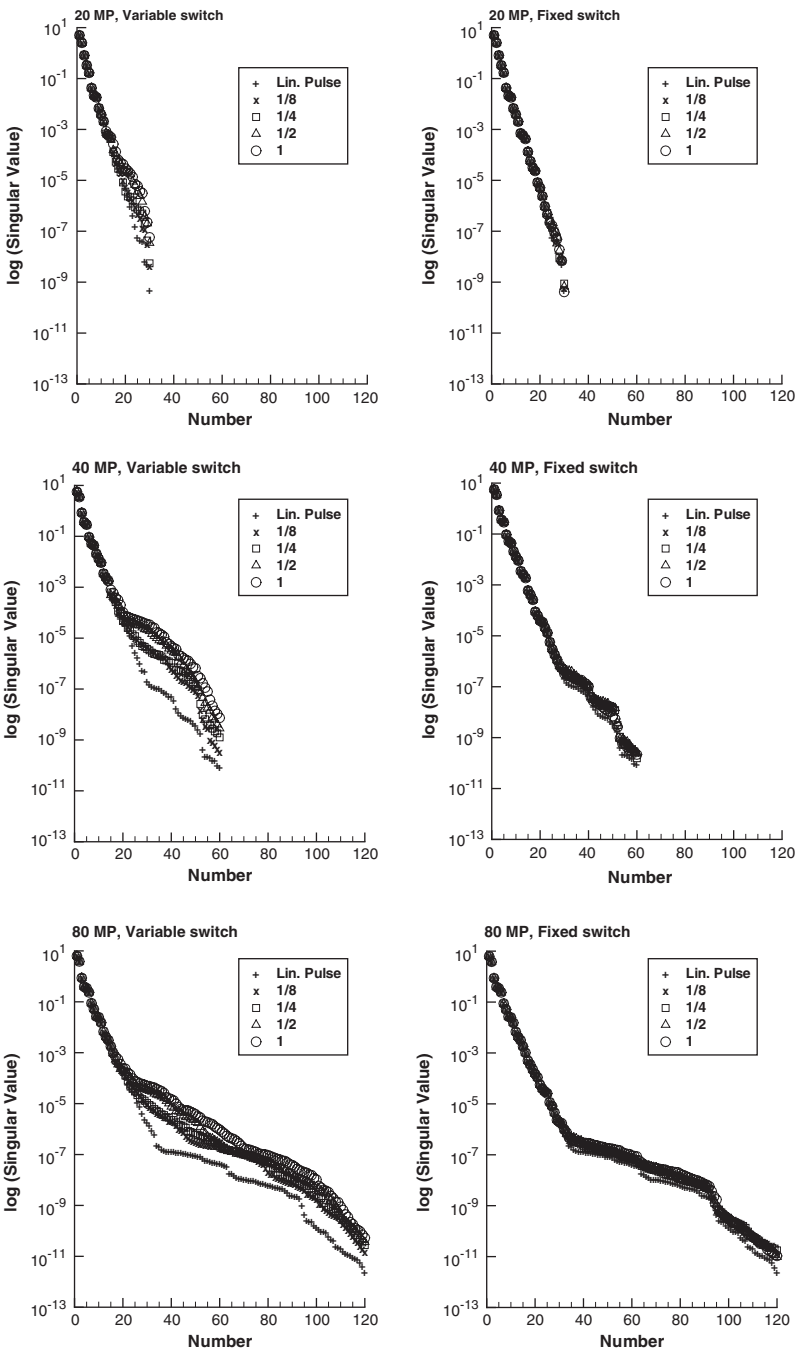

Figure 19. Singular values for Hankel matrices constructed with pulses from the dynamically linear code and non-linear code, $M = 0.8$.

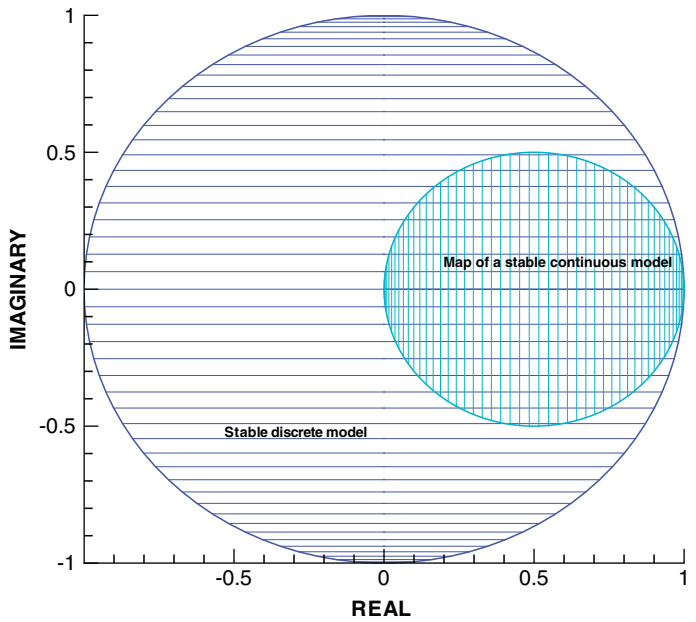

Figure 20. Polar diagram of discrete eigenvalue space.

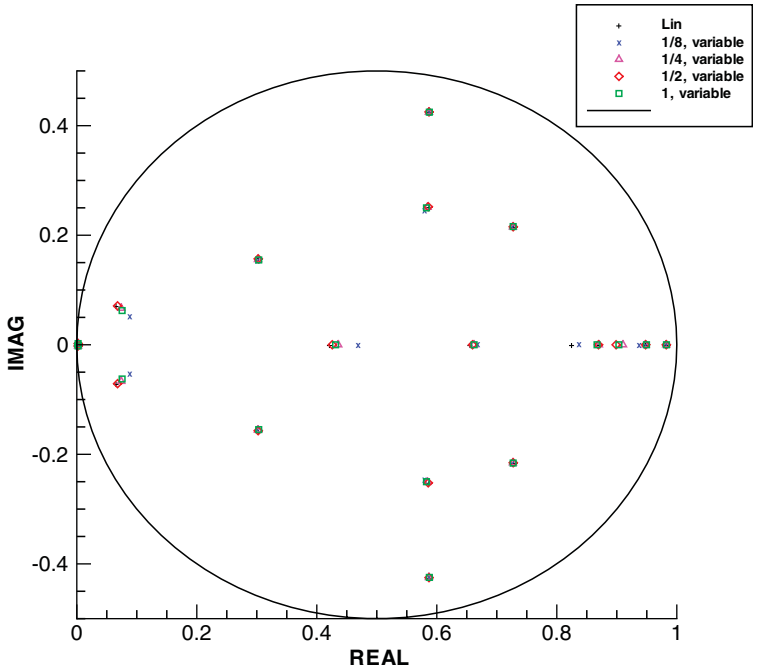

Figure 21. Polar diagram of discrete eigenvalue space showing eigenvalues of rank 18 ROM, $M = 0.7$—variable switch.

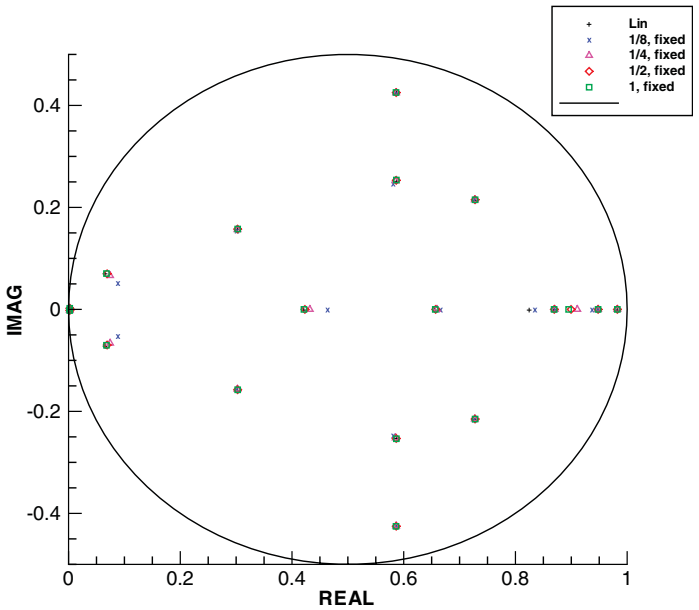

Figure 22. Polar diagram of discrete eigenvalue space showing eigenvalues of rank 18 ROM, $M = 0.7$—fixed switch.

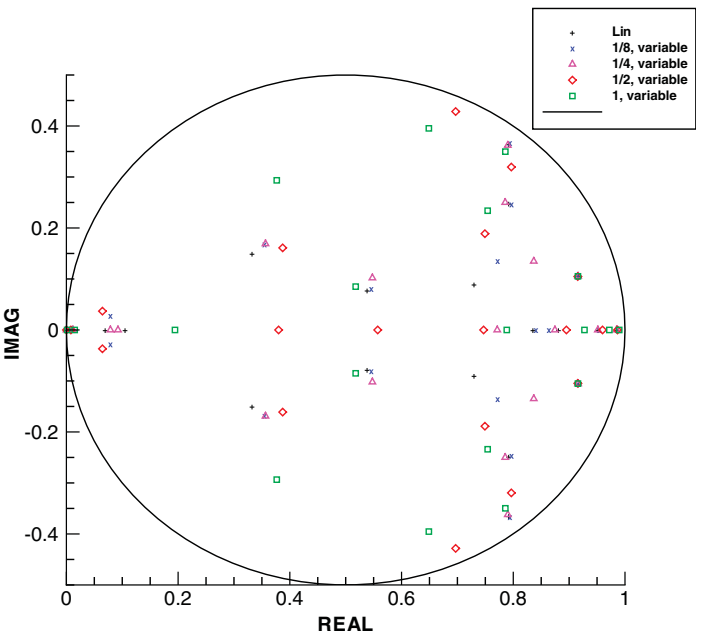

Figure 23. Polar diagram of discrete eigenvalue space showing eigenvalues of rank 20 ROM, $M = 0.8$—variable switch.

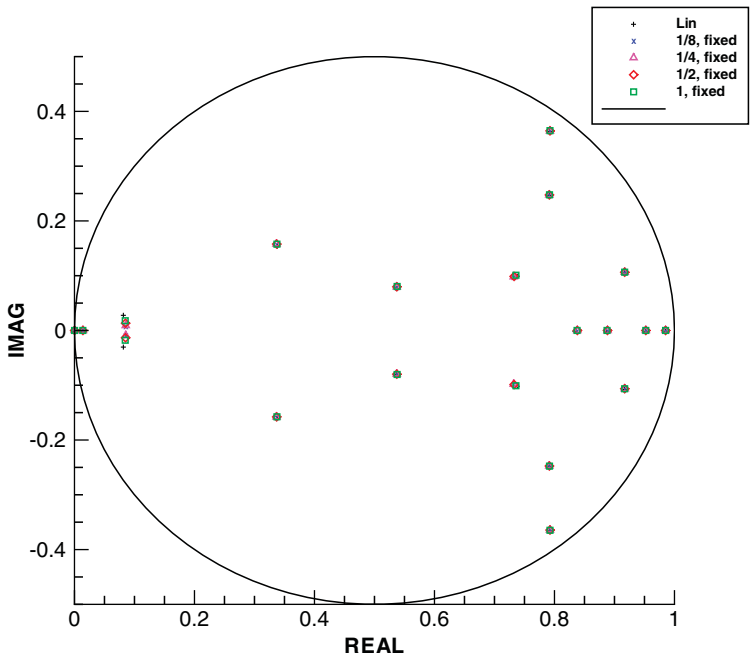

Figure 24. Polar diagram of discrete eigenvalue space showing eigenvalues of rank 20 ROM, $M = 0.8$—fixed switch.

7. CONCLUSIONS

This study has compared, in detail, two methods for extracting the linear responses of a CFD code for the construction of ROMs via ERA, and has provided extremely valuable insight into the characteristics of each approach. This study has shown that the linear responses produced by both methods lead to ROMs that give very similar results even when variable/fixed switch issues are present. Either method can be used to derive ROMs, which give acceptable performance over a particular frequency range. ROMs produced by either method provide greater computational efficiency compared to full-order models for aeroelastic investigations.

The following points should, however, be considered when selecting and implementing either approach. The first approach involving linearizing the CFD code requires fewer pulse response calculations to be performed and is robust in terms of amplitude selection. The second approach, that approximates the linear pulse response as the linear portion of the non-linear response of the full non-linear code, requires twice as many pulse response calculations; however the need to linearize the code is avoided. In this latter case, it has been shown that care must be taken in selecting the pulse amplitudes used to extract the linear response. If large pulses are used the non-linear effects in the code can become so big that the assumptions leading to the formula for the linear response are no longer valid. This situation may arise even for relatively small motions when a shock is present. If small pulses are used then the small size of the responses may lead to problems with rounding error.

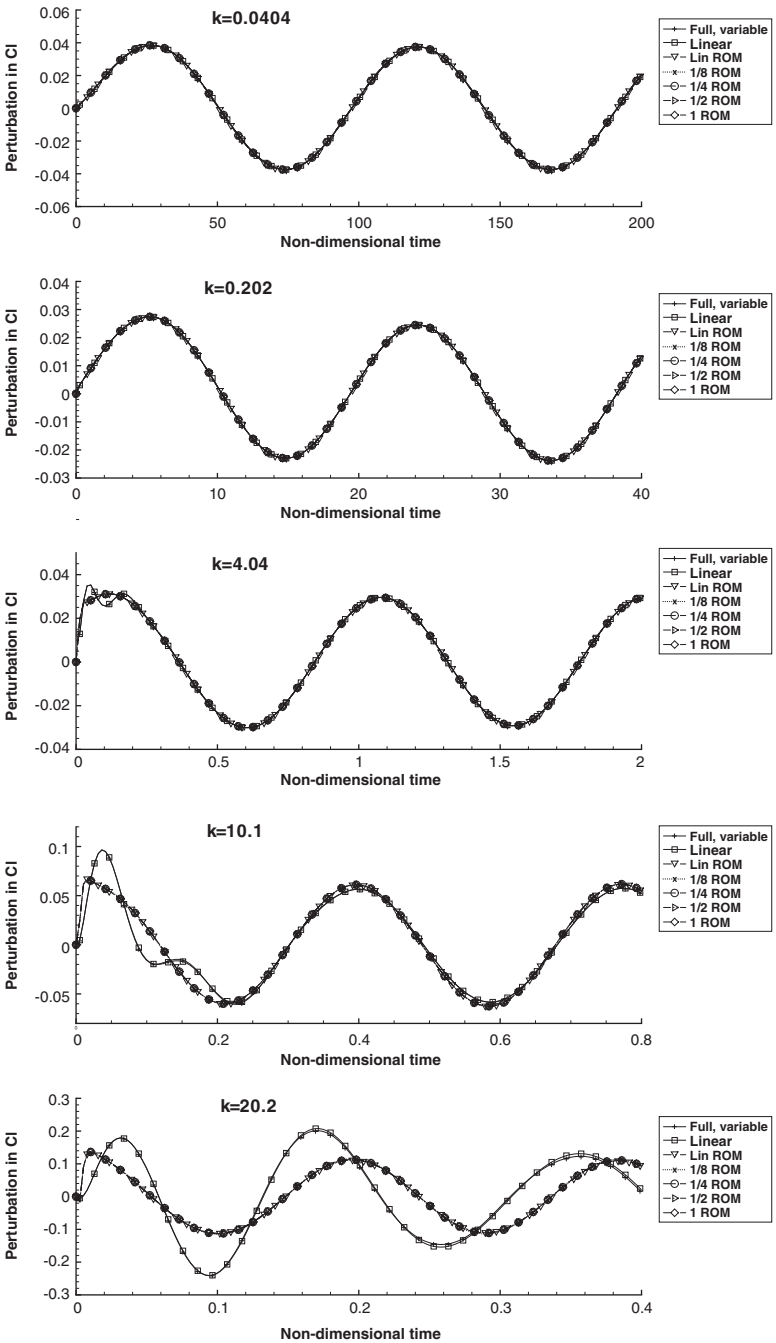

Figure 25. Lift perturbation for linear pitch oscillations, $M = 0.7$—ROMs from non-linear code using variable switch.

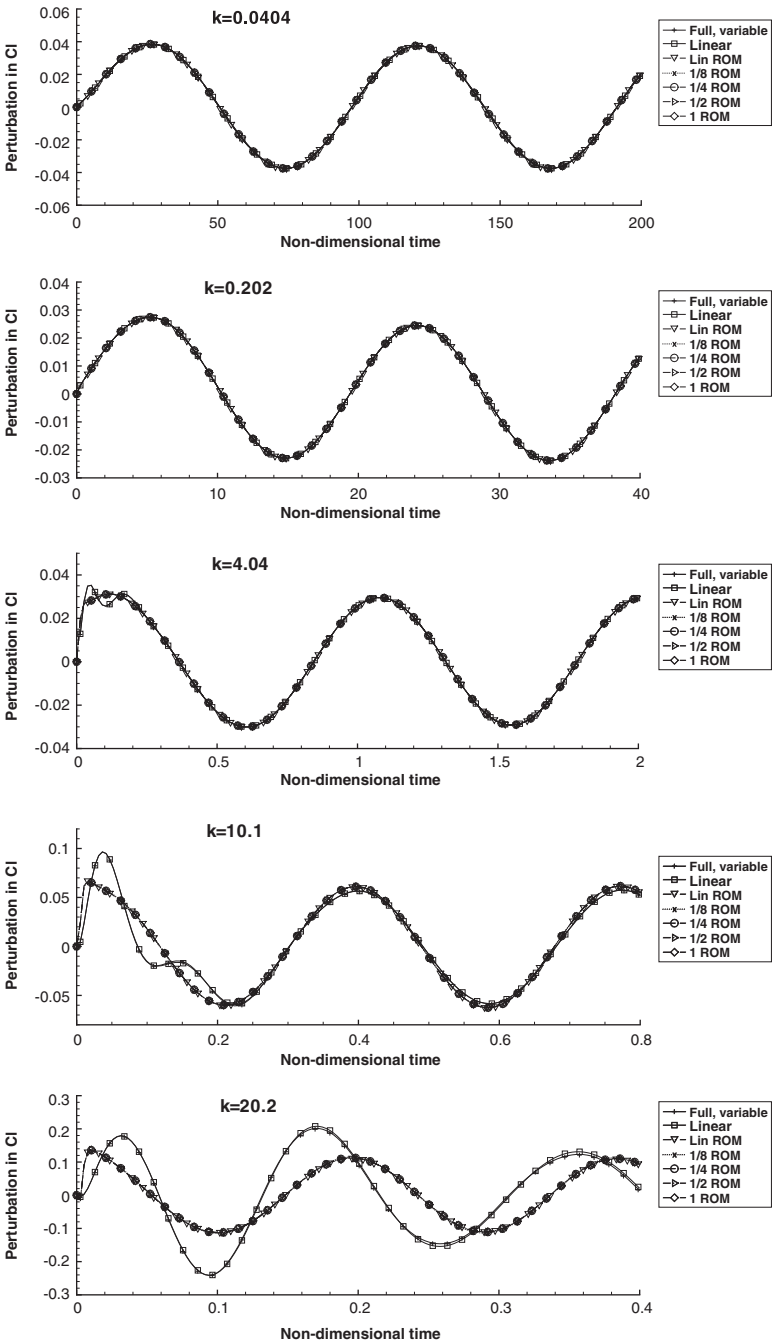

Figure 26. Lift perturbation for linear pitch oscillations, $M = 0.7$—ROMs from non-linear code using fixed switch.

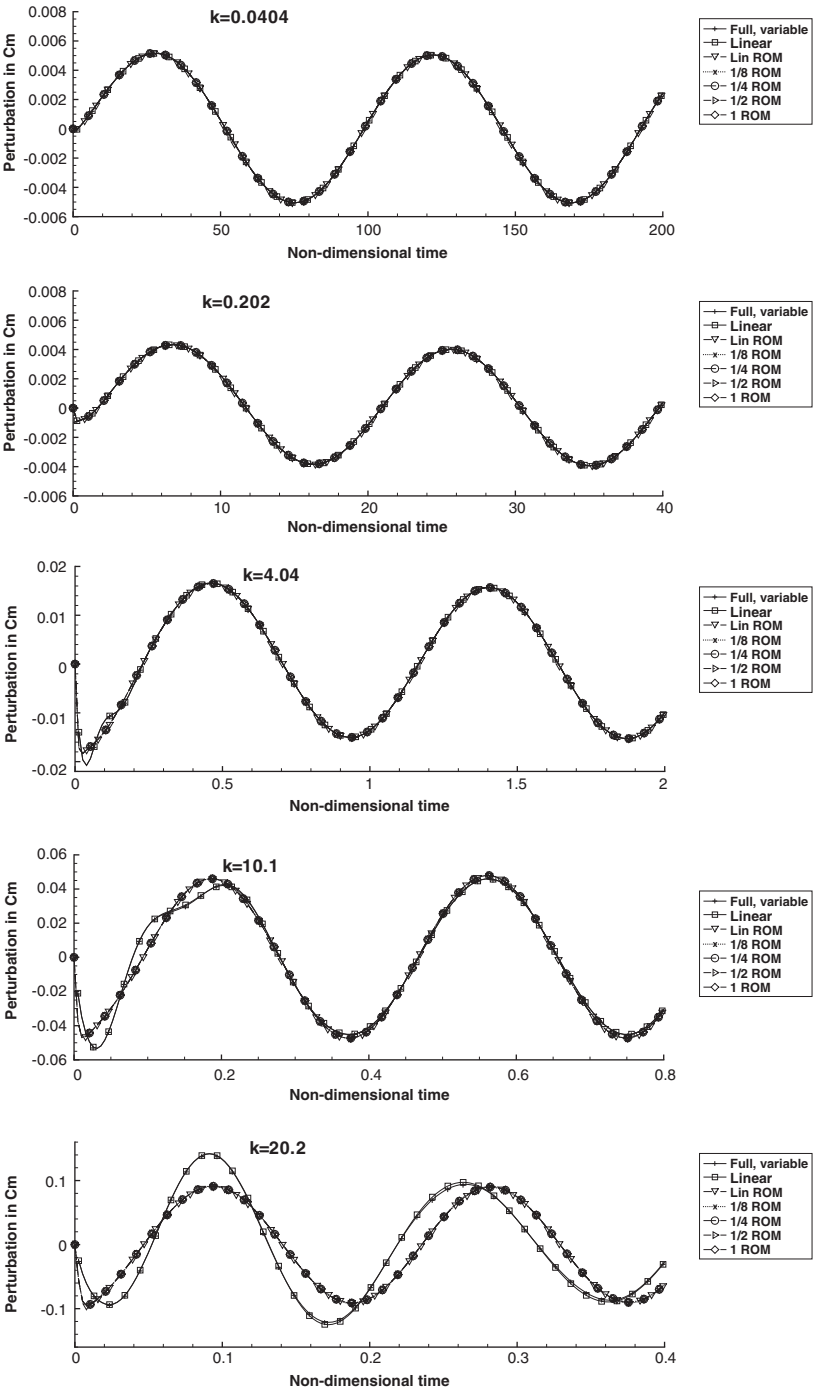

Figure 27. Pitch moment perturbation for linear pitch oscillations, $M = 0.7$—ROMs from non-linear code using variable switch.

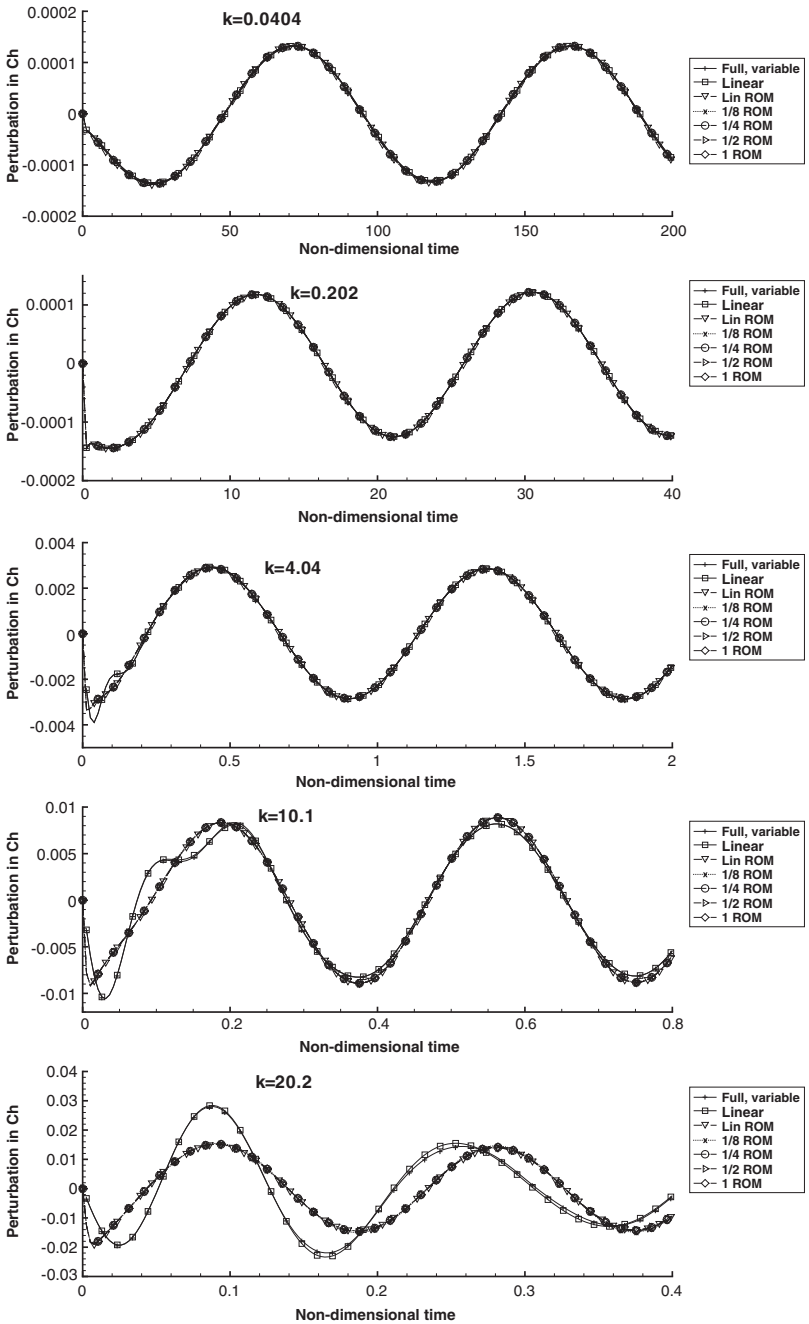

Figure 28. Hinge moment perturbation for linear pitch oscillations, $M = 0.7$—ROMs from non-linear code using variable switch.

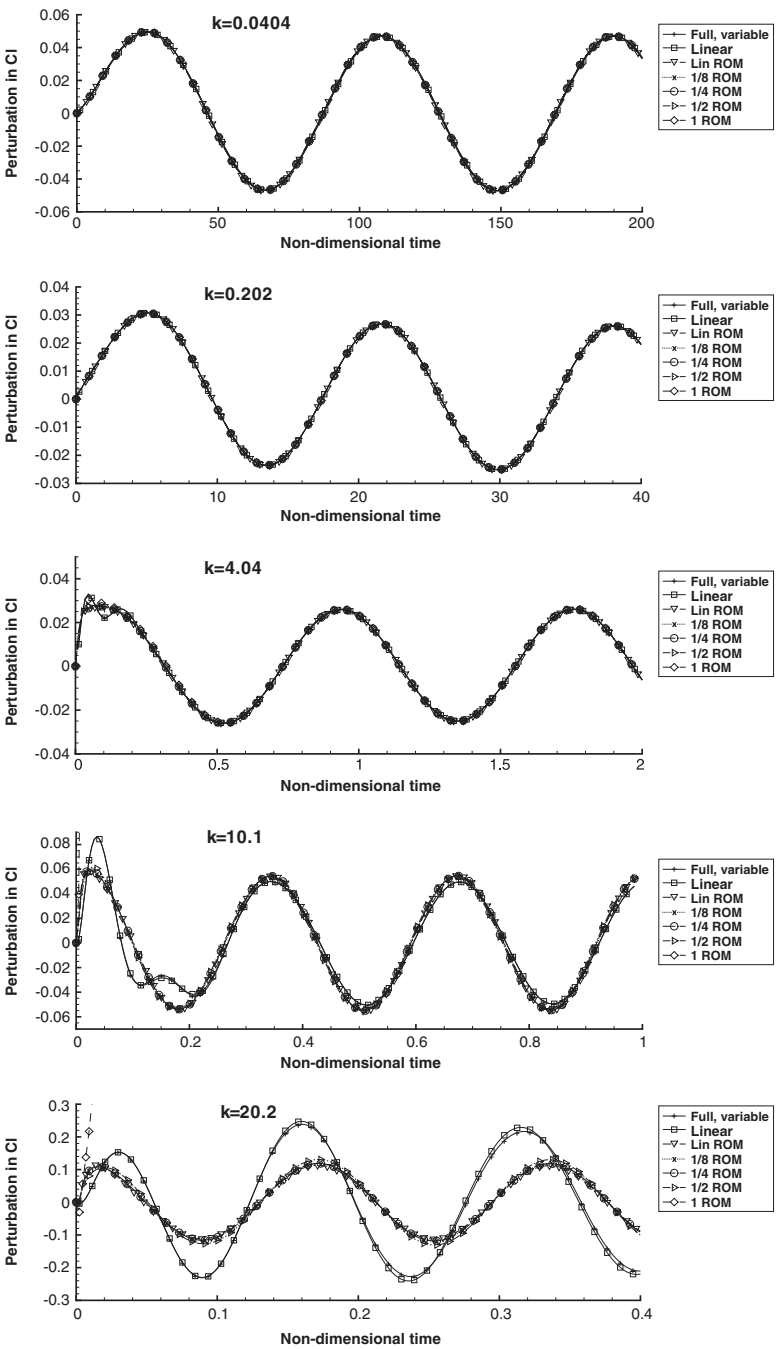

Figure 29. Lift perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using variable switch.

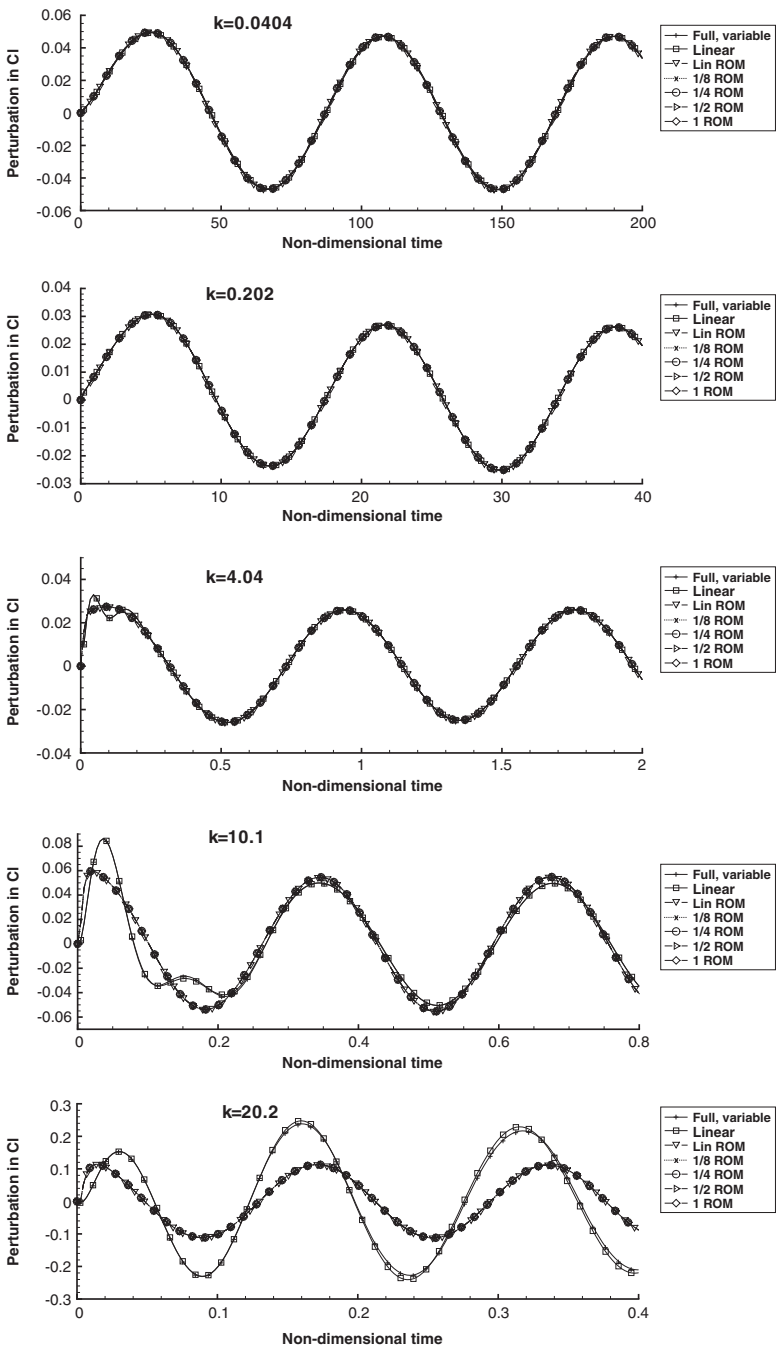

Figure 30. Lift perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using fixed switch.

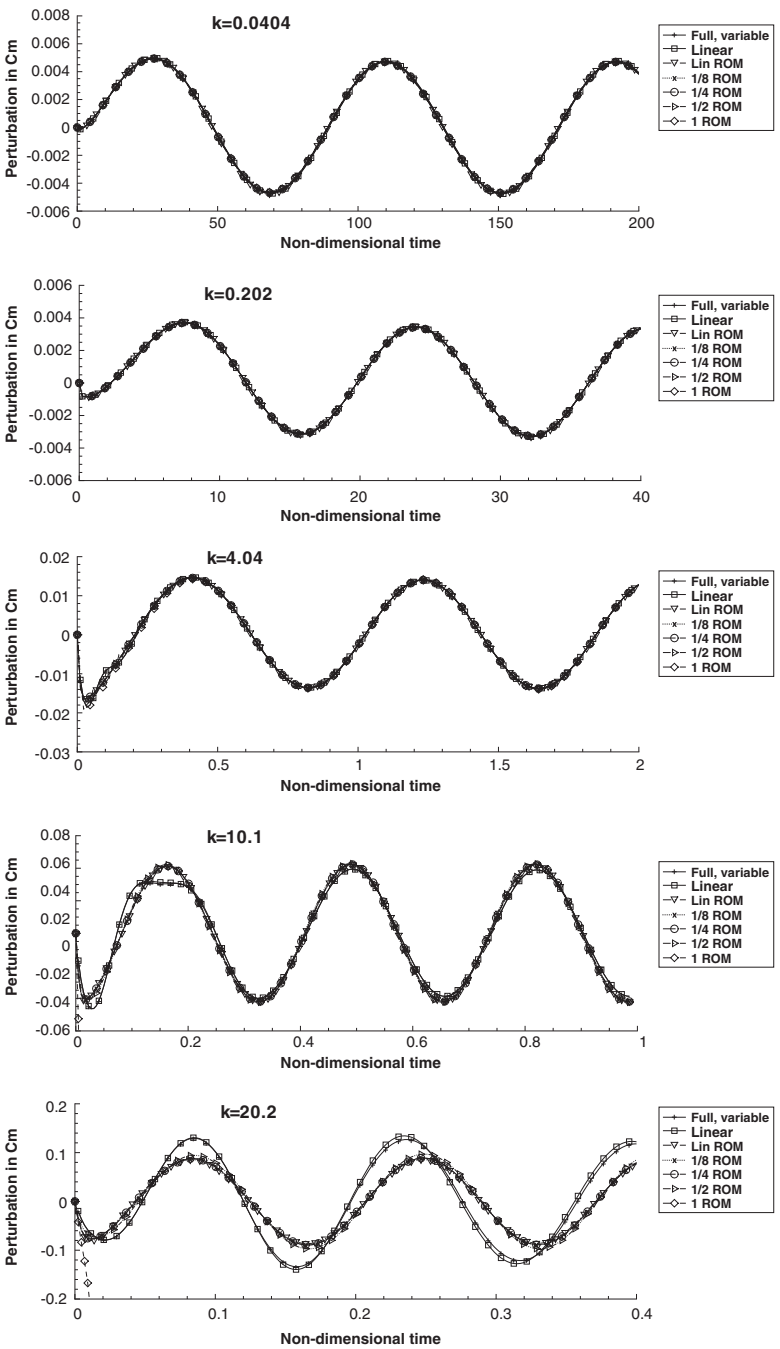

Figure 31. Pitch moment perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using variable switch.

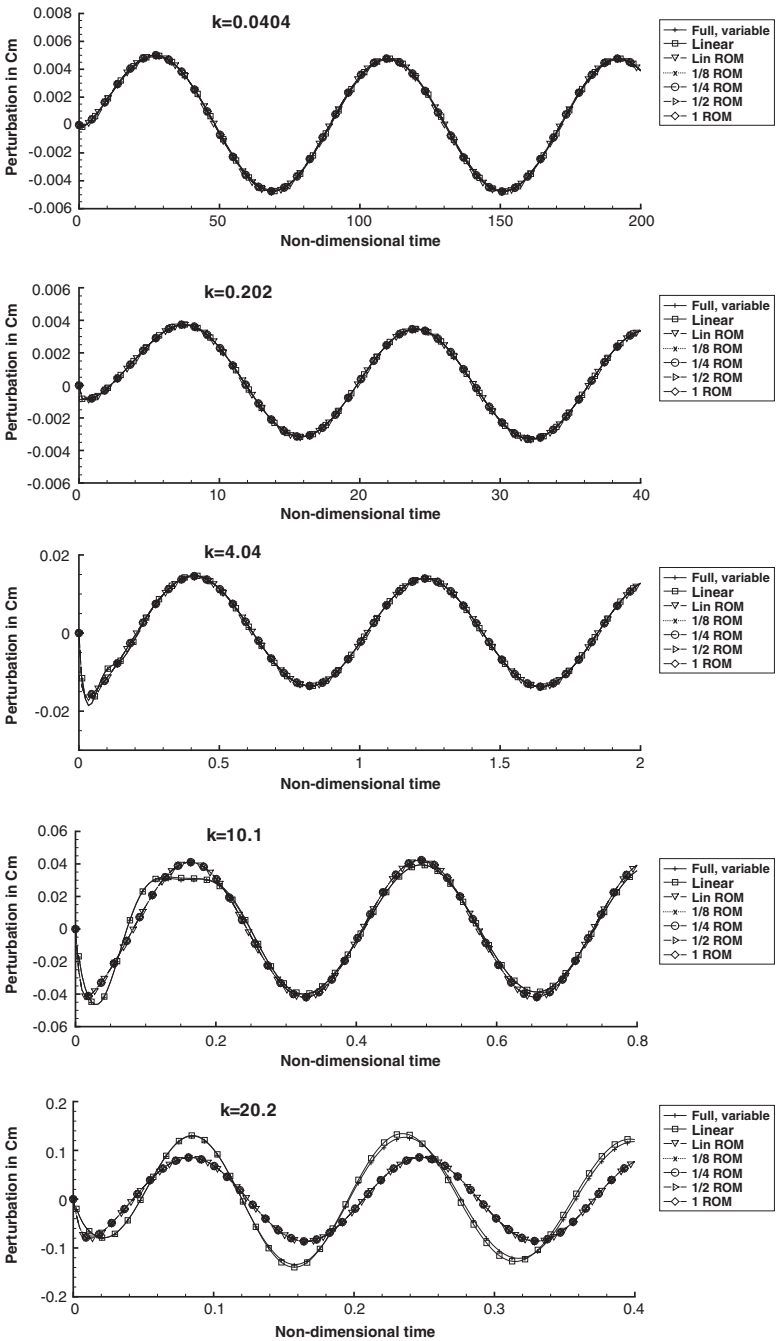

Figure 32. Pitch moment perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using fixed switch.

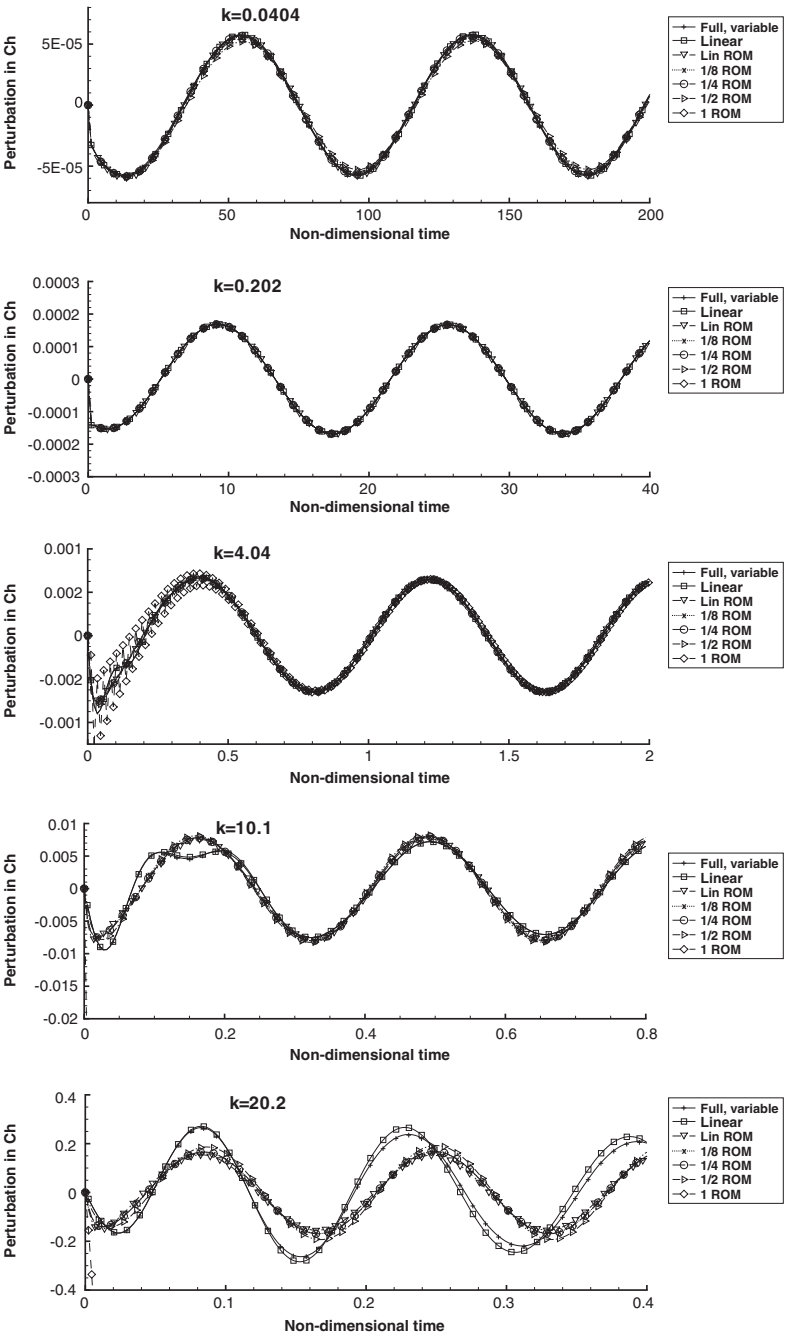

Figure 33. Hinge moment perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using variable switch.

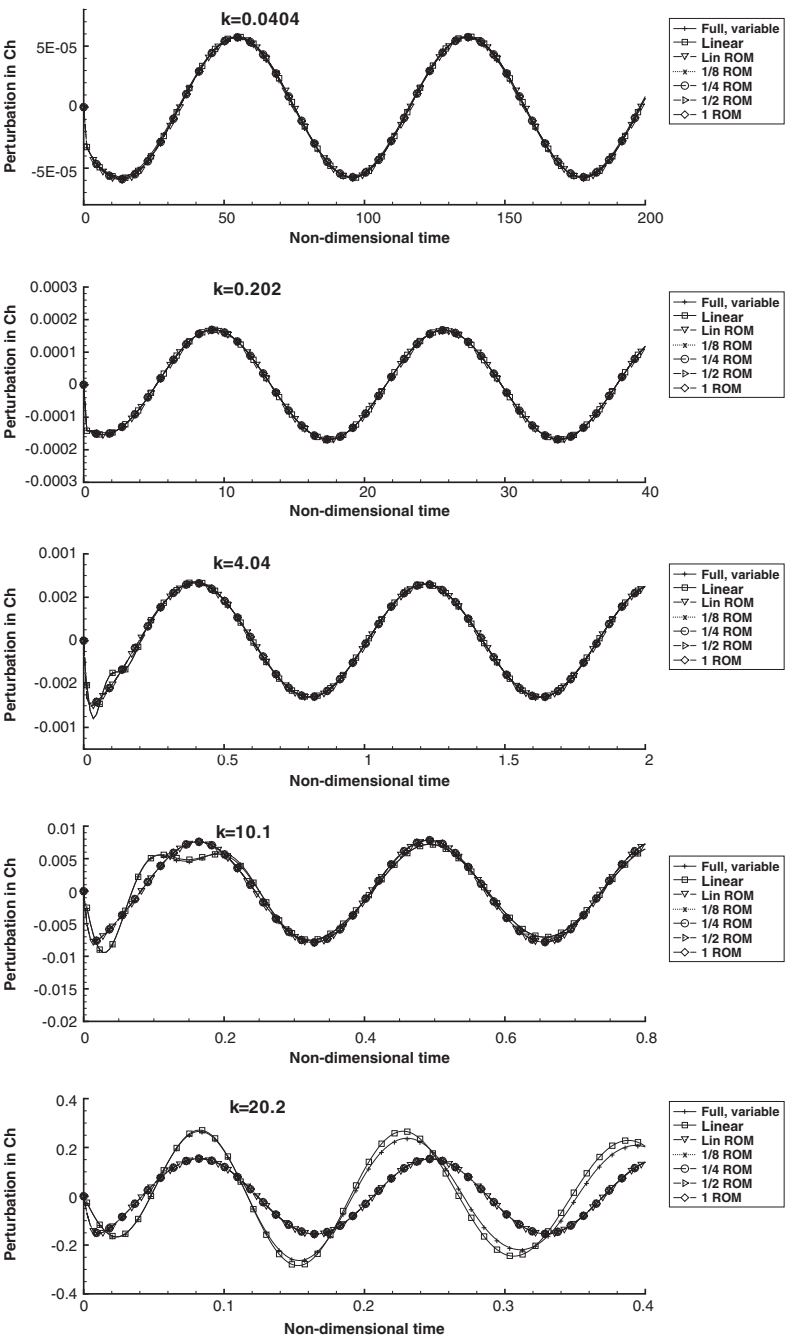

Figure 34. Hinge moment perturbation for linear pitch oscillations, $M = 0.8$—ROMs from non-linear code using fixed switch.

Thus for any code some experimentation with pulse size will be necessary to ensure the best amplitude is selected. For the CFD code used here, it has been demonstrated that fixing the switch in the non-linear code, at its mean value, leads to greater robustness in ROM generation.

REFERENCES

1. Venkatakrishnan V, Jameson A. Computation of unsteady transonic flows by the solution of the Euler equations. *AIAA Journal* 1998; **26**:974–981.
2. Brenneis A, Erbele A. Evaluation of an unsteady implicit Euler code against two and three-dimensional standard configurations. *AGARD CP-507*, 1991, Paper 10.
3. Guruswamy GP. Unsteady aerodynamic and aeroelastic calculations for wings using Euler equations. *AIAA Journal* 1990; **28**:461–469.
4. Jameson AJ. Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings. *AIAA Paper 91-1596*, 1991.
5. Gaitonde AL. A dual time method for the solution of the unsteady Euler equations. *Aeronautical Journal* 1994; **98**(978):283–291.
6. Juang J-N, Pappa RS. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control and Dynamics* 1985; **8**(5):620–627.
7. Ho BL, Kalman RE. Effective construction of linear state-variable models from input/output functions. *Regelungstechnik* 1966; **12**(14):545–548.
8. A survey of model reduction methods for large systems. *Contemporary Mathematics* 2001; **280**:193–219; Technique for unsteady transonic aerodynamic flows. *AIAA Journal* 2000; **38**(10):1853–1862.
9. Grimme EJ. Krylov projection methods for model reduction. *Ph.D. Thesis*, University of Illinois Urbana-Champagne, 1997.
10. Willcox KE. Reduced-order aerodynamic models for aeroelastic control of turbomachines. *MIT Thesis*, 2000.
11. Chen J, Kang S, Zou J, Liu C, Schutt-Aine JE. Reduced-order modeling of weakly non-linear MEMS devices with Taylor-series expansion and Arnoldi approach. *Journal of Microelectromechanical Systems* 2004: **13**(3):441–451.
12. Romanowski MC. Reduced order unsteady aerodynamic and aeroelastic models using Karhunen-Loeve eigenmodes. *Proceedings of the Sixth AIAA Symposium on Multidisciplinary Analysis and Optimization*, 1996. AIAA paper 96-3981.
13. Hall KC, Florea R, Lanzkron PJ. A reduced order model of unsteady flows in turbomachinery. *Journal of Turbomachinery* 1995; **117**(3):375–383.
14. Epureanu BI, Hall KC, Dowell EH. Reduced order in turbomachinery using inviscid-viscous coupling. *Journal of Fluids and Structures* 2001; **15**(2):255–273.
15. Hall KC, Thomas JP, Dowell EH. Proper orthogonal decomposition technique for transonic unsteady aerodynamic flows. *AIAA Journal* 2000; **38**(10):1853–1862.
16. Raveh DE. Reduced-order models for nonlinear unsteady aerodynamics. *AIAA Journal* 2001; **39**(8):1417–1429.
17. Silva WA. Discrete-time linear and nonlinear aerodynamic impulse responses for efficient CFD analyses. *Ph.D. Thesis*, Department of Applied Science, College of William and Mary, Virginia, U.S.A., 1997.
18. Gaitonde AL, Jones DP. System identification and reduction from the pulse responses of a linearised Euler scheme. *CEAS Aerospace Aerodynamics Research Conference*, Cambridge, 2002.
19. Dowell EH, Hall KC. Modelling of fluid-structure interation. *Annual Review of Fluid Mechanics* 2001; **33**: 445–490.
20. Gaitonde AL, Jones DP. A two-dimensional linearized unsteady Euler scheme for pulse response calculations. *Proceedings of the Institute of Mechanical Engineering*; *Journal of Aerospace Engineering* 2002; **216**(Part G):89–104.
21. Gaitonde AL, Jones DP. The use of pulse responses for 2D unsteady flows and system reduction. *Aeronautical Journal* 2002; 483–492.
22. Gaitonde AL, Jones DP. Reduced order state-space models from the pulse responses of a linearised CFD scheme. *International Journal of Numerical Methods in Fluids* 2003; **42**:581–606.
23. Thomas PD, Lombard CK. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal* 1979; **17**:1030–1037.
24. Obayashi S. Freestream capturing for moving coordinates in three dimensions. *AIAA Journal* 1992; **30**: 1125–1128.
25. Gaitonde AL, Jones DP. Study of techniques for obtaining continuous models from 2D discrete reduced order state-space CFD models. *International Journal of Numerical Methods in Fluids* 2006, accepted.
26. Jameson A, Schmidt W, Turkel E. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. *AIAA-Paper 81-1259*, 1981.

27. Aplevich JD. *The Essentials of Linear State-Space Systems*. Wiley: New York, 2000.
28. Conner MD, Virgin LN, Dowell EH. Accurate numerical integration of state-space models for aeroelastic systems with freeplay. *AIAA Journal* 1996; **34**(10):2202–2205.
29. Lin WB, Cheng WH. Nonlinear flutter of loaded lifting surfaces (I) & (II). *Journal of the Chinese Society of Mechanical Engineering* 1993; **14**(5):446–466.

