Efficient Flutter Prediction Using Reduced-Order Modeling

Brandon M. Lowe * and David W. Zingg †
Institute for Aerospace Studies, University of Toronto,
4925 Dufferin St, Toronto, Ontario, M3H 5T6, Canada

Typical aerodynamic shape optimization and multidisciplinary optimization algorithms omit high-fidelity flutter predictions due to the associated computational costs. This paper presents a model order reduction framework as a step towards flutter-constrained aircraft optimization. The Euler equations linearized about a steady-state solution of the nonlinear Euler equations are used as the governing unsteady flow equations. Using a proper orthogonal decomposition approach, a reduced basis is constructed onto which the governing equations are projected. The result is a linear reduced-order model (ROM) with significantly fewer degrees of freedom capable of rapidly approximating aerodynamic forces. This ROM is coupled to a linear structural model to create a single monolithic aeroelastic system. The eigenvalues of the resulting system are analyzed for various flow conditions to determine the onset of flutter in the system. To ensure the stability of the ROM, the use of a stabilizing inner product is demonstrated. The flutter boundaries obtained for both a two degree of freedom airfoil structure and the AGARD 445.6 wing model show good agreement with the full-order model and with the literature.

Nomenclature

a	=	non-dimensional distance from elastic axis to midchord
$\mathbb{A}$	=	aeroelastic state matrix
$A,\ B,\ C,\ D,\ E$	=	linear time-invariant system matrices
$\mathcal{A}$	=	continuous inviscid flux Jacobian
b	=	airfoil semi-chord
C_l	=	lift coefficient
C_m	=	pitching moment coefficient
$\mathcal{E}$	=	vector of continuous inviscid fluxes
$\boldsymbol{f}$	=	force vector

*PhD Candidate, University of Toronto, brandon.lowe@mail.utoronto.ca
†University of Toronto Distinguished Professor of Computational Aerodynamics and Sustainable Aviation, Director, Centre for Research in Sustainable Aviation, dwz@oddjob.utias.utoronto.ca
This work is associated with a conference paper presented at AIAA SciTech 2020, January 10, Orlando FL. AIAA Paper 2020-1998.

$\mathcal{F}$	=	continuous initial conditions
$\mathcal{G}$	=	continuous boundary conditions
$\boldsymbol{g}_s$	=	structural force vector
J	=	diagonal matrix of metric Jacobian
$\mathcal{J}$	=	determinant of continuous coordinate mapping Jacobian
K_m	=	mesh movement matrix
K_s	=	structural stiffness matrix
L_ref	=	reference length
$\mathbb{M}$	=	aeroelastic mass matrix
M_s	=	structural mass matrix
M_∞	=	freestream Mach number
N	=	size of full-order model
n	=	size of reduced-order model
q_∞	=	freestream dynamic pressure
r_α	=	non-dimensional radius of gyration
$\boldsymbol{R}_\mathrm{a}$	=	semi-discrete Euler equation residual vector
$\boldsymbol{R}_\mathrm{ROM}$	=	Galerkin projection residual vector
s	=	thermodynamic entropy
$\bar{s}$	=	mathematical entropy
S	=	aerodynamic snapshot matrix
S_k	=	arc length between volume node and surface node
t	=	time
T	=	transfer matrix
$\boldsymbol{u}$	=	state vector
$\mathcal{U}$	=	vector of continuous flow states
U_W	=	cholesky decomposition of W
V_f	=	flutter speed index
V_∞	=	freestream flow velocity
W	=	inner product matrix
$\mathcal{W}$	=	Hessian of the mathematical entropy
$\boldsymbol{w}_\mathrm{s}$	=	structural generalized coordinate vector
$\boldsymbol{x}_\mathrm{a,surf}$	=	aerodynamic surface node coordinates

$\boldsymbol{x}_{\mathrm{a,vol}}$	=	aerodynamic volume node coordinates
x_α	=	non-dimensional distance from elastic axis to center of gravity
γ	=	continuity constant
η_{ROM}	=	POD snapshot reconstruction tolerance
λ	=	eigenvalue
Λ	=	diagonal matrix of eigenvalues
μ	=	two-dimensional mass ratio
ξ, η, ζ	=	curvilinear coordinates
$\xi_\mathrm{t}, \eta_\mathrm{t}, \zeta_\mathrm{t}$	=	aerodynamic time metrics
ω	=	structural natural frequency
ω_α	=	uncoupled torsional frequency
ω_h	=	uncoupled bending frequency
$\boldsymbol{\Omega}_\mathrm{s}$	=	diagonal matrix of squared natural frequencies
ρ_∞	=	freestream density
σ_f	=	force coupling factor
σ_t	=	time coupling factor
$\boldsymbol{\phi}$	=	reduced basis vector
$\boldsymbol{\Phi}$	=	matrix of reduced basis vectors
$\boldsymbol{\psi}$	=	structural mode shape vector
$\boldsymbol{\Psi}$	=	matrix of structural mode shape vectors

Symbols and Subscripts

$\sim$	=	reduced quantity
$\cdot$	=	time derivative
a	=	aerodynamic quantity
s	=	structural quantity
δ	=	fluctuating quantity
∞	=	freestream quantity
0	=	steady-state quantity

I. Introduction

In recent years, a strong emphasis has been placed on the need for increased aircraft fuel efficiency. Ambitious fuel burn reduction goals require a shift towards novel aircraft designs. Such efficient designs can be conceived with the aid of aerodynamic shape optimization and multidisciplinary optimization tools based on high-fidelity computational fluid dynamics (CFD) [1–4]. Historically, due to complexity and computational cost, flutter constraints have often been omitted from such tools. Resulting optimal aircraft geometries could thus be potentially subject to this dangerous aeroelastic phenomenon.

Recently, the inclusion of flutter constraints in both aerodynamic and structural optimization has gained traction. Mallik *et al*. [5] studied the effects of a flutter constraint on the multidisciplinary optimization (MDO) of a truss-braced-wing aircraft. Using a modified version of Theodorsen's linear aerodynamic function [6], they showed that inclusion of a flutter constraint lead to meaningful penalties on both the weight and fuel efficiency of the resulting aircraft. Jonsson *et al*. [7] have formulated a flutter constraint based on tracking the aeroelastic modes from the *pk* method and aggregated the constraints on all modes using the Kreisselmeier-Steinhauser function [8]. In their work, aerodynamic forces were approximated using a doublet lattice method. Lupp and Cesnik [9] have extended this method to include geometric nonlinearities, similarly relying on linear aerodynamic methods. A low-order aerodynamic model has been implemented into an aircraft design tool by Opgenoord *et al*. [10]. This low-order model is calibrated using high-fidelity CFD simulations and uses strip theory to extend its applications to three-dimensional geometries. Their aircraft optimization results demonstrate that a flutter constraint limits the aspect ratio of the wing, resulting in increased fuel burn. Jacobson *et al*. [11] have implemented a linearized frequency-domain approach for flutter prediction with the intent of formulating a flutter constraint. For further information on flutter-based constraints for aircraft optimization, see the excellent review by Jonsson *et al*. [12].

Linear aerodynamic methods (such as the doublet lattice method) have been widely used for flutter predictions in the past due to their low computational cost. However, these methods fail to model nonlinear flow features accurately in the transonic regime important for modern commercial aircraft [13]. This lack of accuracy has led to overly conservative aircraft designs [14], and hence losses in potential fuel savings. Alternatively, high-fidelity CFD presents a means to model aerodynamics accurately for dynamic aeroelastic analysis. In reality, however, due in part to the unsteady nature of the problem, the large number of degrees of freedom, and the number of flight conditions to consider, the use of CFD for flutter predictions in a design context remains intractable. This problem is exacerbated when introducing flutter predictions into an aerodynamic shape optimization framework, for which each new iteration of aircraft geometry requires a fresh flutter analysis.

Model order reduction of dynamical systems has reached significant maturity. These methods aim to generate reduced-order models (ROMs) which accurately capture the dynamic behavior of the full-order system in a fraction of the computational time. ROMs have previously been used for dynamic aeroelastic analysis [15–31]. Notably, Silva

et al. [15, 16] have constructed aerodynamic ROMs based on aerodynamic forces from full-order CFD simulations using the eigensystem realization algorithm [32]; the resulting ROM is coupled to the structural model for flutter prediction. Badcock *et al*. [17] use the eigenvectors of a full-order aeroelastic system to create ROMs capable of predicting the onset of limit cycle oscillations. Kim [18] has developed a system identification method capable of extracting the aerodynamic substates from aeroelastic time histories, which are used to create a ROM for aeroelastic predictions. Argaman and Raveh [19] have modeled an aeroelastic system as a multi-output autoregressive process with model parameters identified using aeroelastic responses from CFD solutions. Of particular interest to this paper are applications of projection-based ROMs with reduced bases constructed using proper orthogonal decomposition (POD). Notably, Hall *et al*. [20] and Thomas *et al*. [21] were among the first to use a POD-based ROM for aeroelastic modeling. Using the linearized Euler equations in the frequency domain, they constructed ROMs which provide flutter boundaries with good agreement to full-order methods. Beran *et al*. [22] used a POD-based ROM to predict the onset of limit-cycle oscillations for a nonlinear panel in a two-dimensional flow. A few years later, Lieu *et al*. [24, 25] modeled the dynamic aeroelastic behavior of a complete aircraft configuration using a POD-based ROM constructed from the frequency domain linearized Euler equations. More recently, Amsallem *et al*. [26] used the same governing flow equations to predict the flutter boundary of a wing-store configuration using a parameterized ROM. Through an interpolation framework, they were capable of quickly adapting the aerodynamic ROM for various Mach numbers and fuel fill levels. Bekemeyer and Timme [27] have constructed POD-based ROMs for the linearized Reynolds-Averaged Navier-Stokes (RANS) equations for gust load analysis.

This paper presents novel contributions towards a model order reduction approach for flutter prediction for commercial aircraft. The stability of a dynamic aeroelastic system at a given flight condition can be determined by analyzing the eigenvalues of the linearized aeroelastic equations. For this reason, the Euler equations linearized about a nonlinear steady state are used as the governing full-order aerodynamic equations for flutter predictions. The stability of the full-order aeroelastic system can be determined by obtaining the eigenvalues of the coupled linearized aerodynamic and structural models. However, due to the high dimension of the linearized Euler-based aerodynamic model, this eigenproblem is computationally intractable. In this work, a POD approach is used to construct an aerodynamic ROM which approximates the relevant flow dynamics of the full-order linearized Euler equations. The aerodynamic ROM is coupled to a linear structural model to form a single monolithic aeroelastic ROM. The aerodynamic ROM is several orders of magnitude smaller than the full-order aerodynamic model, and thus the aeroelastic eigenvalue problem becomes tractable.

Flow snapshots used for the POD approach are computed using a single unsteady flow solve for a given Mach number. As originally proposed by Silva [33], all structural mode shapes are excited using the Walsh functions. The resulting aerodynamic ROM provides accurate flutter predictions for the Mach number at which the snapshots were computed. Typically, all flow snapshots must be stored into memory simultaneously in order to perform the POD

approach. This paper demonstrates the use of an incremental POD approach which significantly reduces memory requirements while maintaining accuracy if large numbers of snapshots are required. One of the primary contributions of this paper is the application of a stabilizing inner product for the Galerkin projection in the construction of the aerodynamic ROM for flutter predictions. Barone *et al*. [34] demonstrated that a symmetry transformation leads to stable ROMs for the linearized Euler equations. In their work, they presented a symmetry transformation for the linearized Euler equations formulated in primitive variables. A theorem and proof for the stabilizing properties of this symmetry transformation was later provided by Kalashnikova *et al*. [35]. The current paper applies a symmetry transformation to create a stabilizing inner product in the construction of the aerodynamic ROM in order to preserve its stability for flutter predictions. The Hessian of the mathematical entropy is used as the symmetry transformation for the linearized Euler equations formulated in conservative variables. The original proof of stability by Kalashnikova *et al*. [35] is modified to account for a source term in the equations. In the case of flutter predictions, the source term originates from flow excitations due to structural motion.

This paper presents the aforementioned approach, and investigates both the reductions in computational expense and the ability of the method to predict the onset of flutter. Two example cases are presented: a two degree of freedom airfoil structure capable of pitching and plunging, and the well known AGARD 445.6 wing flutter test case. This article is organized as follows. In Section II, the governing flow equations and mesh movement strategy are presented. Section III presents the model order reduction methodology. Section IV presents the governing equations for the structural model. Section V presents a discussion of the strategy for coupling the aerodynamic ROM to the structural model, along with the methodology for flutter prediction. Both the ROM-based and full-order-based flutter prediction approaches are presented, the latter is used to validate the results of the former. In Section VI, the test cases are briefly discussed. Lastly, in Section VII, results for various aspects of this work are presented, including the aforementioned flutter test cases. These are followed by the conclusions in Section VIII. In order to keep the notation clear, quantities associated to the aerodynamic model are generally indicated with the subscript a, whereas quantities for the structural model are presented with a subscript s.

II. Flow Solver

This section presents an overview of the nonlinear flow solver developed at the University of Toronto used in this work to obtain steady-state flow solutions. Subsequently, the linearized discrete Euler equations are derived. Mesh deformation and time-marching methodologies are also presented.

A. Nonlinear Flow Solver

The steady-state flow solutions for this work are obtained using a multiblock structured finite-difference flow solver for the Euler equations developed by Hicken and Zingg [36]. Spatial discretization is achieved using second-order-accurate

summation-by-parts operators, while boundary conditions are weakly imposed using simultaneous approximation terms. The discrete flow equations are solved iteratively using Newton's method. At each Newton iteration, the linear system is solved inexactly using the Krylov-subspace algorithm GMRES with an approximate Schur preconditioner [37].

B. Linearized Discrete Euler Equations

In this section, the Euler equations are linearized after discretization in space. Alternatively, the Euler equations can be linearized and then discretized [38]. The former option was selected for ease of implementation because the existing flow solver included a linearization of the discrete nonlinear Euler equations with respect to the flow state for use with the discrete adjoint method. Transforming the Euler equations to computational space and discretizing in space results in the following set of N_a ordinary differential equations:

$$\frac{d(J^{-1}\boldsymbol{u}_\mathrm{a})}{dt} + \boldsymbol{R}_\mathrm{a}(\boldsymbol{u}_\mathrm{a}, \boldsymbol{x}_\mathrm{a,vol}, \dot{\boldsymbol{x}}_\mathrm{a,vol}) = \boldsymbol{0}, \tag{1}$$

where $\boldsymbol{u}_\mathrm{a} \in \mathbb{R}^{N_\mathrm{a}}$ is the vector of conservative flow states at each node, $\boldsymbol{R}_\mathrm{a} \in \mathbb{R}^{N_\mathrm{a}}$ is the residual vector, and $J^{-1} \in \mathbb{R}^{N_\mathrm{a} \times N_\mathrm{a}}$ is a diagonal matrix of inverse metric Jacobians of the transformation to curvilinear coordinates. The equations are linearized with respect to the conservative flow state $\boldsymbol{u}_\mathrm{a}$, the volume grid node coordinates $\boldsymbol{x}_\mathrm{a,vol}$, and the volume grid node velocities $\dot{\boldsymbol{x}}_\mathrm{a,vol}$. Fluctuations of the latter two quantities will serve as inputs into the system of equations and will be obtained from the structural model. As the linearization is about a nonlinear steady-state, important nonlinear effects, such as small shock wave motion, are captured [38].

The time-derivative term in (1) is first put into non-conservative form,

$$\frac{d(J^{-1}\boldsymbol{u}_\mathrm{a})}{dt} = J^{-1}\dot{\boldsymbol{u}}_\mathrm{a} + \frac{dJ^{-1}}{dt}\boldsymbol{u}_\mathrm{a}, \tag{2}$$

where a dot above a quantity represents a derivative in time. Based on the geometric conservation law (GCL), first proposed by Thomas and Lombard [39], the term $\frac{dJ^{-1}}{dt}$ can be expressed as

$$\frac{dJ^{-1}}{dt} = -\left(\frac{d\hat{\xi}_t}{d\xi} + \frac{d\hat{\eta}_t}{d\eta} + \frac{d\hat{\zeta}_t}{d\zeta}\right), \tag{3}$$

where ξ, η, ζ are curvilinear coordinates, and $\hat{\xi}_t$, $\hat{\eta}_t$, $\hat{\zeta}_t$ are the time metric terms. The time metrics are dependent on both grid coordinates and velocities and can be expanded with respect to these quantities in a Taylor series about a steady-state. However, the steady-state the grid velocity is zero; therefore any first-order dependence on the grid coordinates disappears. Thus, expanding both terms in (2) in a Taylor series about the steady-state and omitting second-order terms gives

$$\frac{d(J^{-1}\boldsymbol{u}_\mathrm{a})}{dt} \approx J_0^{-1}\delta\dot{\boldsymbol{u}}_\mathrm{a} + \frac{\partial}{\partial\dot{\boldsymbol{x}}_\mathrm{a,vol}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{u}_\mathrm{a,0}\,\delta\dot{\boldsymbol{x}}_\mathrm{a,vol}, \tag{4}$$

where steady-state quantities are denoted with a subscript 0, and fluctuating quantities with a δ. Similarly, the residual in (1) is expanded in a Taylor series such that

$$\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{u}_{\mathrm{a}}, \boldsymbol{x}_{\mathrm{a,vol}}, \dot{\boldsymbol{x}}_{\mathrm{a,vol}}) \approx \boldsymbol{R}_{\mathrm{a}}(\boldsymbol{u}_{\mathrm{a},0}, \boldsymbol{x}_{\mathrm{a,vol},0}, \dot{\boldsymbol{x}}_{\mathrm{a,vol},0}) + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{u}_{\mathrm{a}}}\delta\boldsymbol{u}_{\mathrm{a}} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{x}_{\mathrm{a,vol}}}\delta\boldsymbol{x}_{\mathrm{a,vol}} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{x}}_{\mathrm{a,vol}}}\delta\dot{\boldsymbol{x}}_{\mathrm{a,vol}}, \tag{5}$$

where the residual of the steady problem, $\boldsymbol{R}_{\mathrm{a}}(\boldsymbol{u}_{\mathrm{a},0}, \boldsymbol{x}_{\mathrm{a,vol},0}, \dot{\boldsymbol{x}}_{\mathrm{a,vol},0})$, is zero. Combining equations (4) and (5) provides the linearized discrete Euler equations:

$$J_0^{-1}\delta\dot{\boldsymbol{u}}_{\mathrm{a}} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{u}_{\mathrm{a}}}\delta\boldsymbol{u}_{\mathrm{a}} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{x}_{\mathrm{a,vol}}}\delta\boldsymbol{x}_{\mathrm{a,vol}} + \left(\frac{\partial}{\partial \dot{\boldsymbol{x}}_{\mathrm{a,vol}}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{u}_{\mathrm{a},0} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{x}}_{\mathrm{a,vol}}}\right)\delta\dot{\boldsymbol{x}}_{\mathrm{a,vol}} = 0. \tag{6}$$

For aeroelastic considerations, the outputs of interest are the aerodynamic force fluctuations about a steady state. Forces are computed using the aerodynamic pressure, which is explicitly dependent on the flow state $\boldsymbol{u}_{\mathrm{a}}$, and the area-weighted surface normal, which is explicitly dependent on the grid node coordinates $\boldsymbol{x}_{\mathrm{a,vol}}$. Expanding the fluctuations of the force as a Taylor series, one obtains

$$\delta\boldsymbol{f}_{\mathrm{a}} \approx \frac{\partial \boldsymbol{f}_{\mathrm{a}}}{\partial \boldsymbol{u}_{\mathrm{a}}}\delta\boldsymbol{u}_{\mathrm{a}} + \frac{\partial \boldsymbol{f}_{\mathrm{a}}}{\partial \boldsymbol{x}_{\mathrm{a,vol}}}\delta\boldsymbol{x}_{\mathrm{a,vol}}. \tag{7}$$

Combining equations (6) and (7) provides the complete linear time-invariant (LTI) system describing the evolution and output of the aerodynamic model. Dropping the δ's to simplify the notation, the LTI system is

$$\begin{aligned} E_{\mathrm{a}}\dot{\boldsymbol{u}}_{\mathrm{a}} &= A_{\mathrm{a}}\boldsymbol{u}_{\mathrm{a}} + B_{\mathrm{a},1}\dot{\boldsymbol{x}}_{\mathrm{a,vol}} + B_{\mathrm{a},2}\boldsymbol{x}_{\mathrm{a,vol}} \\ \boldsymbol{f}_{\mathrm{a}} &= C_{\mathrm{a}}\boldsymbol{u}_{\mathrm{a}} + D_{\mathrm{a}}\boldsymbol{x}_{\mathrm{a,vol}}. \end{aligned} \tag{8}$$

From the linearization above, the system matrices are,

$$A_{\mathrm{a}} = \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{u}_{\mathrm{a}}}, \quad B_{\mathrm{a},1} = \left(\frac{\partial}{\partial \dot{\boldsymbol{x}}_{\mathrm{a,vol}}}\left(\frac{dJ^{-1}}{dt}\right)\boldsymbol{u}_{\mathrm{a},0} + \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \dot{\boldsymbol{x}}_{\mathrm{a,vol}}}\right), \quad B_{\mathrm{a},2} = \frac{\partial \boldsymbol{R}_{\mathrm{a}}}{\partial \boldsymbol{x}_{\mathrm{a,vol}}}, \quad C_{\mathrm{a}} = \frac{\partial \boldsymbol{f}_{\mathrm{a}}}{\partial \boldsymbol{u}_{\mathrm{a}}}, \quad D_{\mathrm{a}} = \frac{\partial \boldsymbol{f}_{\mathrm{a}}}{\partial \boldsymbol{x}_{\mathrm{a,vol}}}, \quad E_{\mathrm{a}} = J_0^{-1}.$$

C. Algebraic Mesh Deformation

In the formulation of equation (8), the vectors $\boldsymbol{x}_{\mathrm{a,vol}}$ and $\dot{\boldsymbol{x}}_{\mathrm{a,vol}}$ denote the fluctuating node coordinates and velocities of the entire computational grid, also known as the volume grid. A typical volume grid can easily contain $O(10^5)$ nodes, which leads to large input vectors. Such vectors require numerous operations for their respective matrix-vector products in equation (8); therefore reducing the size of these vectors would save on both memory and computational expense. Consider a linear mesh deformation algorithm which can be cast in the following form

$$\boldsymbol{x}_{\mathrm{a,vol}} = K_m\boldsymbol{x}_{\mathrm{a,surf}}, \tag{9}$$

where $\boldsymbol{x}_{\mathrm{a,vol}}$ is the vector of volume node displacements, $\boldsymbol{x}_{\mathrm{a,surf}}$ the vector of surface node displacements, and K_m the mesh deformation matrix. If K_m is time-invariant, then one can also relate the volume node velocity fluctuations to the surface node velocity fluctuations by

$$\dot{\boldsymbol{x}}_{\mathrm{a,vol}} = K_m \dot{\boldsymbol{x}}_{\mathrm{a,surf}}. \tag{10}$$

Inserting this into equation (8) gives:

$$\begin{aligned} E_{\mathrm{a}} \dot{\boldsymbol{u}}_{\mathrm{a}} &= A_{\mathrm{a}} \boldsymbol{u}_{\mathrm{a}} + B_{\mathrm{a},1} K_m \dot{\boldsymbol{x}}_{\mathrm{a,surf}} + B_{\mathrm{a},2} K_m \boldsymbol{x}_{\mathrm{a,surf}}, \\ \boldsymbol{f}_{\mathrm{a}} &= C_{\mathrm{a}} \boldsymbol{u}_{\mathrm{a}} + D_{\mathrm{a}} K_m \boldsymbol{x}_{\mathrm{a,surf}}. \end{aligned} \tag{11}$$

The mesh movement can be integrated into the linearized Euler equations by performing the matrix multiplications $\left(B_{\mathrm{a},1} K_m\right)$, $\left(B_{\mathrm{a},2} K_m\right)$, and $\left(D_{\mathrm{a}} K_m\right)$ as a precomputing step.

For this work, the matrix K_m is constructed based on an algebraic mesh deformation algorithm described by Leung and Zingg [40]. In this algorithm, the movement of the k^{th} volume node is given by the equation

$$x_{k,\mathrm{vol}} = \frac{x_{k,\mathrm{surf}}}{2} \left(1 + \cos(\pi S_k)\right), \tag{12}$$

where $x_{k,\mathrm{surf}}$ is the displacement of the surface node on the same grid line, and S_k is the normalized arc length from the surface node to the k^{th} volume node. Although algebraic mesh deformation has not been found to be robust for large surface movements and may lead to poor mesh quality as a result, the use of the linearized Euler equations is limited to small surface movements; thus algebraic mesh deformation is found to be sufficient.

D. Time-Marching Method

As will be discussed in the following section, the full-order linearized Euler equations (11) are marched in time in order to create a reduced basis onto which the equations are projected. The second-order backwards difference method is used to march the equations in time. At each time step, the linear system is solved using the modified Generalized Conjugate Residual with inner Orthogonal and outer Truncation (GCROT(m,k)) Krylov method [41] with an approximate Schur preconditioner [36].

III. Model Order Reduction

The LTI system (11) retains the same dimension as the original nonlinear discrete governing flow equations and thus remains impractical for fast flutter predictions. In order to reduce computational expense, a projection-based model order reduction approach is used for which the reduced basis is created using POD. Excellent literature exists on the fundamentals and applications of POD [42–45]. To keep this paper self-contained, the general concepts are reviewed

briefly in the first part of this section. Subsequently, a stability preserving inner product is presented, followed by the description of an incremental POD approach, and finally the procedure for snapshot collection.

A. POD-based Model Order Reduction

The key goal of POD is to represent a set of data with as few basis vectors as possible. To this end, consider a set of m state (or "snapshot") vectors $\boldsymbol{u}_{\mathrm{a},1}, \boldsymbol{u}_{\mathrm{a},2}, ..., \boldsymbol{u}_{\mathrm{a},m} \in \mathbb{R}^{N_\mathrm{a}}$ at different instants in time, that is $\boldsymbol{u}_{\mathrm{a},j} = \boldsymbol{u}_\mathrm{a}(t_j)$. A set of orthogonal basis vectors $\boldsymbol{\phi}_i \in \mathbb{R}^{N_\mathrm{a}}$ which minimizes the least squares error of the snapshot reconstruction is desired. This is done by solving the eigenproblem [43]:

$$WSS^T W\boldsymbol{\phi}_i = \lambda_i W\boldsymbol{\phi}_i, \tag{13}$$

where $S \in \mathbb{R}^{N_\mathrm{a}\times m}$ is the snapshot matrix containing the m snapshots stored column-wise, $W \in \mathbb{R}^{N_\mathrm{a}\times N_\mathrm{a}}$ is a symmetric positive-definite matrix which defines a weighted inner product in Euclidean space, and λ_i is the eigenvalue associated to the i^{th} eigenvector $\boldsymbol{\phi}_i$. The weighted inner product matrix W may serve multiple purposes, including as an approximation to an inner product in continuous space (for example an L^2 inner product) [43], and as an inner product which ensures the stability of the resulting ROM [35]. The reduced basis $\Phi \in \mathbb{R}^{N_\mathrm{a}\times n_\mathrm{a}}$ onto which the governing equations are projected, is formed such that $\Phi = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, ..., \boldsymbol{\phi}_{n_\mathrm{a}}]$ with the eigenvectors and values arranged as $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_m$ and $n_\mathrm{a} \leq m$.

The sum of the eigenvalues associated to the excluded eigenvectors quantifies the snapshot reconstruction error,

$$\sum_{j=1}^{m} \left\| \boldsymbol{u}_{\mathrm{a},j} - \sum_{i=1}^{n_\mathrm{a}} \left(\boldsymbol{u}_{\mathrm{a},j}^T W\boldsymbol{\phi}_i \right) \boldsymbol{\phi}_i \right\|_W^2 = \sum_{j=n_\mathrm{a}+1}^{m} \lambda_j, \tag{14}$$

where $||\boldsymbol{\psi}||_W = \sqrt{\boldsymbol{\psi}^T W\boldsymbol{\psi}}$ is the induced norm associated to W. This can be used as a metric to choose an appropriate reduced basis dimension n_a. With a tolerance η_ROM (often smaller than 10^{-1}), one can compute the smallest n_a such that

$$1 - \frac{\sum_{i=1}^{n_\mathrm{a}} \lambda_i}{\sum_{i=1}^{m} \lambda_i} \leq \eta_\mathrm{ROM}. \tag{15}$$

For further details, see [42, 43, 45].

In projecting equation (11) onto the reduced basis, one approximates the evolution of the solution in the reduced space [42]:

$$\boldsymbol{u}_\mathrm{a}(t) \approx \Phi\tilde{\boldsymbol{u}}_\mathrm{a}(t), \tag{16}$$

where $\tilde{\boldsymbol{u}}_\mathrm{a} \in \mathbb{R}^{n_\mathrm{a}}$ is a vector of time-dependent coefficients. This approximation results in the residual:

$$\boldsymbol{R}_\mathrm{ROM} = E_\mathrm{a}\Phi\dot{\tilde{\boldsymbol{u}}}_\mathrm{a} - A_\mathrm{a}\Phi\tilde{\boldsymbol{u}}_\mathrm{a} - B_{\mathrm{a},1}K_m\dot{\boldsymbol{x}}_{\mathrm{a,surf}} - B_{\mathrm{a},2}K_m\boldsymbol{x}_{\mathrm{a,surf}}. \tag{17}$$

Enforcing the Galerkin condition, which states that the residual is orthogonal to the reduced space,

$$\Phi^T W \boldsymbol{R}_{\text{ROM}} = 0, \tag{18}$$

gives the ROM of interest,

$$\begin{aligned} \tilde{E}_{\text{a}} \dot{\tilde{\boldsymbol{u}}}_{\text{a}} &= \tilde{A}_{\text{a}} \tilde{\boldsymbol{u}}_{\text{a}} + \tilde{B}_{\text{a},1} K_m \dot{\boldsymbol{x}}_{\text{a,surf}} + \tilde{B}_{\text{a},2} K_m \boldsymbol{x}_{\text{a,surf}}, \\ \tilde{\boldsymbol{f}}_{\text{a}} &= \tilde{C}_{\text{a}} \tilde{\boldsymbol{u}}_{\text{a}} + \tilde{D}_{\text{a}} K_m \boldsymbol{x}_{\text{a,surf}}. \end{aligned} \tag{19}$$

This ROM contains n_{a} degrees of freedom, with $n_{\text{a}} \ll N_{\text{a}}$. The reduced matrices are,

$$\tilde{A}_{\text{a}} = \Phi^T W A_{\text{a}} \Phi, \quad \tilde{B}_{\text{a},1} = \Phi^T W B_{\text{a},1}, \quad \tilde{B}_{\text{a},2} = \Phi^T W B_{\text{a},2}, \quad \tilde{C}_{\text{a}} = C_{\text{a}} \Phi, \quad \tilde{D}_{\text{a}} = D_{\text{a}}, \quad \tilde{E}_{\text{a}} = \Phi^T W E_{\text{a}} \Phi.$$

B. Stabilizing Inner Product

The full-order aerodynamic equations are always stable prior to coupling with the structural model. Therefore, it is important that the aerodynamic ROM preserves this stability; however this is not always guaranteed. This section presents a theorem which provides conditions that ensure the stability of the aerodynamic ROM.

To ensure stability for ROMs created using the linearized Euler equations, Barone *et al*. [34] proposed the use of a weighted inner product which symmetrizes the flux Jacobian. Kalashnikova *et al*. [35] extended the work to several other linearized systems, and provided a theorem and proof for the stabilizing properties of a symmetrizing inner product. To demonstrate their theorem, the continuous form of the linearized Euler equations are first presented; these are differentiated from the discrete form (6) by using a calligraphic font. In a similar procedure as for the discrete equations, the continuous Euler equations are mapped from physical coordinates $x_{(1,2,3)}$ to curvilinear coordinates $\xi_{(1,2,3)}$, and each term is expanded in a Taylor series. The resulting continuous linearized Euler equations are:

$$\mathcal{J}_0^{-1} \frac{\partial \delta \mathcal{U}}{\partial t} + \frac{\partial \mathcal{A}_{(i)} \delta \mathcal{U}}{\partial \xi_{(i)}} = \mathcal{S}, \tag{20}$$

where Einstein notation is used such that repeated indices are summed. In this equation, $i \in [1, 3]$ refers to the coordinate index, $\delta \mathcal{U} \in \mathbb{R}^5$ is the vector of conservative flow variable fluctuations about a steady state, $\mathcal{J}_0 \in \mathbb{R}$ is the determinant of the coordinate mapping Jacobian at steady-state, $\mathcal{A}_{(i)} = \frac{\partial \hat{\mathcal{E}}_{(i)}}{\partial \mathcal{U}} \in \mathbb{R}^{5\times5}$ is the flux Jacobian in the i^{th} direction evaluated at steady-state, and $\hat{\mathcal{E}}_{(i)} \in \mathbb{R}^5$ denotes the flux vector in the i^{th} direction. The symbol $\hat{}$ on the flux vector indicates that it has been scaled by $\mathcal{J}_0^{-1}$. Additionally, $\xi_{(i)}$ is the i^{th} coordinate direction in curvilinear coordinates. The term $\mathcal{S} \in \mathbb{R}^5$ is a source term defining the linear influence of the position fluctuations $\delta x_{(1,2,3)}$ and velocities $\delta \dot{x}_{(1,2,3)}$ in the physical

domain on the system,

$$\mathcal{S} = -\frac{\partial}{\partial \xi_{(i)}}\left(\frac{\partial \hat{\mathcal{E}}_{(i)}}{\partial x_{(j)}}\delta x_{(j)}\right) - \frac{\partial}{\partial \xi_{(i)}}\left(\frac{\partial \hat{\mathcal{E}}_{(i)}}{\partial \dot{x}_{(j)}}\delta \dot{x}_{(j)}\right) - \mathcal{U}_0^T \frac{\partial}{\partial \dot{x}_{(j)}}\left(\frac{\partial \mathcal{J}^{-1}}{\partial t}\right)\delta \dot{x}_{(j)}. \tag{21}$$

The boundary and initial conditions are:

$$\begin{aligned} \mathcal{U}(x,t) &= \mathcal{G}(t), \qquad x \in \partial\Omega, \quad t \in [0,T], \\ \mathcal{U}(x,0) &= \mathcal{F}(x), \qquad x \in \Omega. \end{aligned} \tag{22}$$

The following theorem has been modified from Kalashnikova *et al*. [35] to generalize the applicable boundary conditions and to include contributions from the source term. The proof for Theorem III.1 is given in the Appendix.

Theorem III.1. *Begin with the linearized Euler equations* (20) *given on an open domain* Ω *with a smooth boundary* $\partial\Omega$ *and with initial and boundary conditions* (22)*. Suppose a ROM is constructed using the spatial basis functions* $\phi(x)$*. Define* $\tilde{\mathcal{U}}(t) = \left[\tilde{\mathcal{U}}_1(t), \tilde{\mathcal{U}}_2(t), ...\right]^T$ *as the vector of modal coefficients such that the solution is approximated as,*

$$\mathcal{U}(x,t) \approx \sum_i \phi_i(x)\tilde{\mathcal{U}}_i(t). \tag{23}$$

Assume that the initial and boundary conditions are bounded and exist in the reduced space. Moreover, assume that the source term $\mathcal{S}$ *is also bounded. If the ROM is constructed using a continuous Galerkin projection with a weighted* $L^2(\Omega)$ *inner product which symmetrizes the flux Jacobian* $\mathcal{A}_{(i)}$*, the ROM is energy stable with energy estimate:*

$$||\tilde{\mathcal{U}}(T)||^2_{\mathcal{W}} \leq k\left(||\mathcal{F}||^2_{\mathcal{W}} + ||\mathcal{G}||^2_{\mathcal{G}}\right), \tag{24}$$

where T *is a given time,* $||\cdot||_{\mathcal{W}}$ *is the induced norm from the weighted* $L^2(\Omega)$ *inner product, and* $||\cdot||_{\mathcal{G}}$ *is the induced norm from the* $L^2(\partial\Omega \times (0,T))$ *inner product.*

Theorem III.1 ensures that the resulting aerodynamic ROM is stable for any reduced basis used for its construction. With this stability property, any instability seen in the ROM-based aeroelastic system must arise from the coupling between the structural model and the aerodynamic ROM. For the Euler equations, it has been shown that a number of mathematical entropy functions exist for which the Hessian symmetrizes the flux Jacobian [46]. In this work, the following mathematical entropy is used:

$$\bar{s} = -\rho s, \tag{25}$$

where s is the thermodynamic entropy, and ρ is the density. The Hessian of the mathematical entropy $\mathcal{W} = \frac{\partial^2 \bar{s}}{\partial \mathcal{U}^2} \in \mathbb{R}^{5\times 5}$ is symmetric positive definite. This allows it to be used as a weight for the L^2 inner product used in the Galerkin

projection to provide a stable ROM.

For a semi-discrete system, a matrix W is formed which approximates the continuous weighted L^2 inner product. This is accomplished by storing the values of $\mathcal{W}$ at each node in the mesh in blocks along the diagonal of W and using a trapezoidal quadrature to approximate the integration over the domain.

C. Incremental Basis Creation

The POD approach presented in Section III.A requires that all snapshots first be computed and stored in order to construct the reduced basis. However, each snapshot contains N_a elements, where N_a is the number of nodes in the aerodynamic mesh multiplied by five. Thus, storing the required number of snapshots simultaneously may be impractical or even impossible. An incremental basis creation method allows one to circumvent this by updating an existing reduced basis with new batches of snapshots, eliminating the need to store all snapshots concurrently.

The algorithm implemented for this incremental approach is a modified version of Levey and Lindenbaum's sequential Karhunen–Loeve algorithm [47]. This algorithm relies on the singular value decomposition of the snapshot matrix S. In order to use a weighted inner product, the Cholesky decomposition of the inner product matrix $W = U_W^T U_W$ is used. To obtain a singular value decomposition equivalent to the POD eigenvalue problem from equation (13), this Cholesky decomposition is first introduced into the eigenvalue equation,

$$U_W^T U_W S S^T U_W^T U_W \Phi = U_W^T U_W \Phi \Lambda, \tag{26}$$

where Λ is a diagonal matrix containing the eigenvalues, and Φ is the matrix containing the right eigenvectors. By defining $\bar{S} = U_W S$ and $\bar{\Phi} = U_W \Phi$, an equivalent eigenproblem is obtained,

$$\bar{S}\bar{S}^T \bar{\Phi} = \bar{\Phi}\Lambda. \tag{27}$$

This can then be transformed into a singular value decomposition by introducing the decomposition $\Lambda = \Sigma\Sigma$ and the identity $V^T V = I$,

$$\bar{S}\bar{S}^T = \bar{\Phi}\Sigma V^T V \Sigma \bar{\Phi}^T = \left(\bar{\Phi}\Sigma V^T\right)\left(\bar{\Phi}\Sigma V^T\right)^T \tag{28}$$

The resulting singular value decomposition for weighted POD is $\bar{S} = \bar{\Phi}\Sigma V^T$. The modified sequential Karhunen–Loeve procedure used to incrementally create the reduced basis is found in Algorithm 1. For parallel computation, except for snapshot creation, the only multi-processor operation required in this algorithm is the QR factorization. For efficient implementation of this step, ScaLAPACK [48] is used.

Algorithm 1: Incremental weighted proper-orthogonal decomposition, modified from [47].

Input: Initial POD basis Φ_0, associated singular values stored in the diagonal matrix Σ_0, weighted inner product matrix W, desired number of snapshots ;

Compute the Cholesky decomposition: $U_W^T U_W = W$;
Compute the weighted reduced basis $\bar{\Phi}_0 = U_W \Phi_0$;
$i \leftarrow 0$;
while *number of snapshots* $\leq$ *desired number of snapshots* **do**
  $i \leftarrow i + 1$;
  Compute n_i snapshots and store them in S_i ;
  Weight the snapshot matrix: $\bar{S}_i = U_w S_i$;
  Compute the matrix: $\Psi_{i-1} = \bar{\Phi}_{i-1}\Sigma_{i-1}$;
  QR factorization: $QR = \left[\Psi_{i-1}|\bar{S}_i\right]$;
  Singular value decomposition: $UDV^T = R$;
  Let Σ_i be diagonal matrix containing the r largest singular values from D, and $\tilde{U}$ be the matrix containing the corresponding left singular vectors from U ;
  Update the weighted basis: $\bar{\Phi}_i = Q\tilde{U}$;
end
Remove the weights from reduced basis: $\Phi = U_W^{-1}\bar{\Phi}_i$;

Output: Updated reduced basis Φ ;

D. Snapshot Collection

Snapshots are collected using a simultaneous input excitation method, similar to the approaches detailed in [30, 33]. As the name suggests, all inputs for the aerodynamic model are excited simultaneously during a single flow solve, and the resulting flowfield at each time instance is stored as a snapshot and used later to create the POD-based reduced basis. The use of simultaneous excitations for snapshot collection has been found to be more economical in terms of computational time when generating the ROM compared to exciting each system input in separate flow simulations [30]. One key aspect is that each input signal must be relatively uncorrelated to generate a rich database of flowfield reactions.

Previously, Silva [33] studied the use of three different types of inputs for multi-excitation: the lagged-step, the block-pulse, and the Walsh functions. Of these, it was found that ROMs constructed using snapshots from the Walsh functions demonstrated the best results. The Walsh functions are impulsive in nature due to their step-like behavior. This allows for a broad bandwidth of frequencies to be excited in the flow. Moreover, they are orthogonal in time, which helps provide a rich data set of flow reactions. For this reason, the Walsh functions are used to define the structural excitations when collecting flow snapshots. The amplitude of the structural excitations are limited to 0.01 semi-chord lengths to ensure that all snapshots remain physical, i.e. no negative pressures. The first four Walsh function excitations used in this work are given in Figure 1.

IV. Structural Model

It is common to model a structure's motion using a modal method [49]. Given a structural model with N_s degrees of freedom, the spatial dependence is described via n_s mode shapes, where $n_s < N_s$. These mode shapes typically

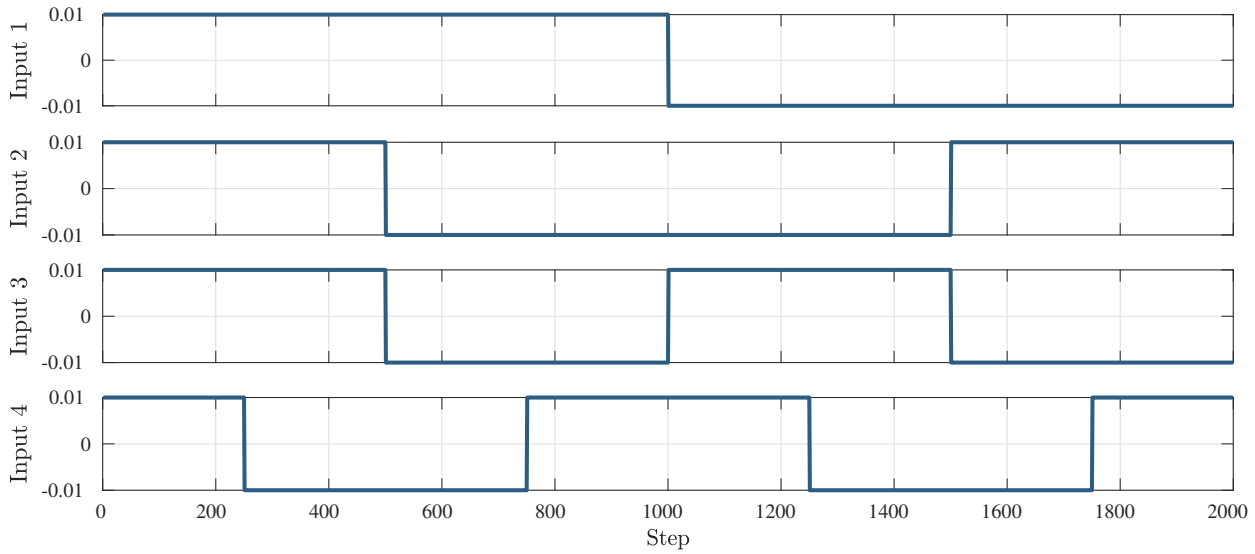


Fig. 1 Example of Walsh function based input excitations for snapshot collection.

correspond to the free vibration modes of the structure, given by the generalized eigenvalue problem

$$K_s \boldsymbol{\psi}_i = \omega_i^2 M_s \boldsymbol{\psi}_i, \tag{29}$$

where $K_s \in \mathbb{R}^{N_s \times N_s}$ is the stiffness matrix, $M_s \in \mathbb{R}^{N_s \times N_s}$ is the mass matrix, and $\omega_i \in \mathbb{R}$ and $\boldsymbol{\psi}_i \in \mathbb{R}^{N_s}$ are the modal frequency and mode shape associated to the i^{th} free vibration mode. These mode shapes are orthogonal to one another and can be scaled to produce unit modal mass. Using the first n_s modes to model the system and neglecting damping terms gives the following uncoupled equations of motion:

$$\ddot{\boldsymbol{w}}_s + \Omega_s \boldsymbol{w}_s = \Psi^T \boldsymbol{g}_s. \tag{30}$$

Here, $\boldsymbol{w}_s \in \mathbb{R}^{n_s}$ is the vector of generalized displacements for each mode, $\boldsymbol{g}_s \in \mathbb{R}^{N_s}$ the vector of applied forces, $\Omega_s \in \mathbb{R}^{n_s \times n_s}$ the diagonal matrix of squared modal frequencies ω_i^2, and $\Psi \in \mathbb{R}^{N_s \times n_s}$ the matrix containing the structural mode shapes $\boldsymbol{\psi}_i$ stored column-wise.

In order to couple this system with the aerodynamic model, it is convenient to rewrite equation (30) as a system of first-order ordinary differential equations. Introducing the new state vector $\boldsymbol{u}_s = \left[\dot{\boldsymbol{w}}_s^T, \boldsymbol{w}_s^T\right]^T$ leads to the following formulation:

$$\dot{\boldsymbol{u}}_s = A_s \boldsymbol{u}_s + \boldsymbol{f}_s, \tag{31}$$

where,

$$A_\mathrm{s} = \begin{bmatrix} 0 & -\Omega_\mathrm{s} \\ I & 0 \end{bmatrix}, \qquad \boldsymbol{f}_\mathrm{s} = \begin{bmatrix} \Psi^T \boldsymbol{g}_\mathrm{s} \\ 0 \end{bmatrix}.$$

V. Flutter Predictions

This section presents the approach used for flutter predictions. First, the displacement and force transfer between the aerodynamic and the structural model is described. Subsequently, the ROM-based flutter prediction approach is described. This is followed by a description of the time-dependent full-order model-based flutter prediction approach used to validate the results obtained by the ROM.

A. Model Coupling

For aeroelastic analysis, the structural model (31) must be coupled to the aerodynamic model. In this section, the term aerodynamic model is kept general, and may refer to either the full-order nonlinear Euler equations (1), the full-order linearized Euler equations (11), or the aerodynamic ROM (19). For this coupling, two challenges arise. The first comes from the fact that quantities exchanged between models must have common dimensions. The second arises from the non-coinciding surface grids between the two models, resulting in the need for force and displacement transfer between grids. In this section, the variable representing time is t_a for the aerodynamic model and t_s for the structural model. Additionally, for clarity, time derivatives in this section will be denoted not by a dot, but rather by $\frac{d}{dt_\mathrm{a}}$ or $\frac{d}{dt_\mathrm{s}}$.

1. Dimensional Consistency

To couple the aerodynamic and structural models, quantities which are exchanged must be dimensionally consistent. To this end, the following time and force scaling factors are introduced:

$$t_\mathrm{s} = \sigma_\mathrm{t} t_\mathrm{a}, \qquad f_\mathrm{s} = \sigma_\mathrm{f} f_\mathrm{a}. \tag{32}$$

Note that f_s and f_a represent only scalar force quantities for the structural and aerodynamic model, respectively. They are not denoted as vectors because the surface grids do not coincide between models; this is addressed in the next subsection.

From the aerodynamic model, dimensional time $\hat{t}$ and force $\hat{f}$ are obtained with the following equations:

$$\hat{t} = \frac{L_\mathrm{ref} M_\infty}{V_\infty} t_\mathrm{a}, \qquad \hat{f} = L_\mathrm{ref}^2 q_\infty f_\mathrm{a}, \tag{33}$$

where L_ref is a reference length, M_∞ is the freestream Mach number, V_∞ is the freestream velocity, and q_∞ is the

freestream dynamic pressure. The structural models presented in this paper each possess different dimensions. For the AGARD 445.6 wing, the structural model is kept in dimensional form, thus $t_s = \hat{t}$ and $f_s = \hat{f}$. Additionally, the aerodynamic grid was constructed around a dimensional AGARD wing geometry, therefore $L_{ref} = 1$ in. This leads to the following scaling factors for the AGARD wing test case:

$$\sigma_t = \frac{L_{ref} M_\infty}{\sqrt{2 q_\infty \rho_\infty}}, \qquad \sigma_f = L_{ref}^2 q_\infty, \tag{34}$$

where ρ_∞ is the freestream density. For the two degree of freedom structural model, time is nondimensionalized by the torsional natural frequency $t_s = \omega_\alpha \hat{t}$, and applied forces are nondimensionalized as $f_s = \hat{f}/(mb\omega_\alpha^2)$, where m is the airfoil mass, and b the semi-chord. The aerodynamic grid was constructed about an airfoil geometry nondimensionalized by the chord length, therefore the length scaling is $L_{ref} = 2b$. Using the definition of the two-dimensional mass ratio $\mu = m/(\rho_\infty b^2 \pi)$, the scaling factors for the two degree of freedom structural model is obtained:

$$\sigma_t = \frac{2M_\infty}{V\sqrt{\mu}}, \qquad \sigma_f = \frac{1}{\pi} V^2, \tag{35}$$

where V is the nondimensional speed index defined as

$$V = \frac{V_\infty}{b\omega_\alpha \sqrt{\mu}}. \tag{36}$$

The value of V at which flutter occurs is named the flutter speed index, and will be denoted by V_f.

2. Displacement and Force Transfer

As the fluid grid surface nodes do not coincide with the structural grid surface nodes due to the different nature of model discretizations, quantities which must be exchanged between the two models must be transferred between grids. Specifically, the generalized coordinates $\boldsymbol{w}_s$ and velocities $\frac{d\boldsymbol{w}_s}{dt_s}$ of the structure must be transformed into aerodynamic surface mesh deformations $\boldsymbol{x}_{a,surf}$ and velocities $\frac{d\boldsymbol{x}_{a,surf}}{dt_a}$, respectively. Additionally, the force output from the aerodynamic model $\boldsymbol{f}_a$ must be transformed into $\boldsymbol{f}_s$, the force applied to the structural model. For this, the transfer matrices T_s and T_a are introduced, and the scaling factors in (32) are applied,

$$\boldsymbol{x}_{a,surf} = T_s \boldsymbol{w}_s, \qquad \frac{d\boldsymbol{x}_{a,surf}}{dt_a} = \sigma_t T_s \frac{d\boldsymbol{w}_s}{dt_s}, \qquad \boldsymbol{f}_s = \sigma_f T_a \boldsymbol{f}_a, \tag{37}$$

In the two degree of freedom structural system introduced later in this paper, the matrix T_s is formulated analytically using a small angle approximation, while T_a is formulated based on the equations for C_l and C_m, the lift and pitching moment coefficients. For more complex geometries, a number of options exist for constructing T_s and T_a. In the present

work, Rendall and Allen's radial basis function interpolation [50] is used to construct the transfer matrices for the AGARD 445.6 wing case. The radial basis function approach ensures that energy, and total force and moment are conserved between both models. Moreover, it allows for the exact recovery of translation and rotation. This method is chosen for the aforementioned properties, and for its ease of implementation: it is a point cloud based method and hence no grid connectivity information is required. For further information see [50].

B. Reduced-Order Model Based Flutter Predictions

Recall that the inputs for the aerodynamic model (19) are: the aerodynamic surface velocity fluctuations $\frac{d\boldsymbol{x}_{\mathrm{a,surf}}}{dt_{\mathrm{a}}}$, and the aerodynamic surface displacement fluctuations $\boldsymbol{x}_{\mathrm{a,surf}}$. From equation (37) above, these inputs can be redefined as

$$\begin{bmatrix} \frac{d\boldsymbol{x}_{\mathrm{a,surf}}}{dt_{\mathrm{a}}} \\ \boldsymbol{x}_{\mathrm{a,surf}} \end{bmatrix} = \begin{bmatrix} \sigma_{\mathrm{t}} T_{\mathrm{s}} \frac{d\boldsymbol{w}_{\mathrm{s}}}{dt_{\mathrm{s}}} \\ T_{\mathrm{s}} \boldsymbol{w}_{\mathrm{s}} \end{bmatrix} = \begin{bmatrix} \sigma_{\mathrm{t}} T_{\mathrm{s}} & 0 \\ 0 & T_{\mathrm{s}} \end{bmatrix} \boldsymbol{u}_{\mathrm{s}}, \tag{38}$$

where $\boldsymbol{u}_{\mathrm{s}}$ was previously introduced as the augmented structural state vector in equation (31). Inserting equation (38) into the aerodynamic ROM (19), and using the dimensionalization factors from (32) gives

$$\begin{aligned} \sigma_{\mathrm{t}} \tilde{E}_{\mathrm{a}} \frac{d\tilde{\boldsymbol{u}}_{\mathrm{a}}}{dt_{\mathrm{s}}} &= \tilde{A}_{\mathrm{a}} \tilde{\boldsymbol{u}}_{\mathrm{a}} + \bar{B}_{\mathrm{a}} \boldsymbol{u}_{\mathrm{s}}, \\ \tilde{\boldsymbol{f}}_{\mathrm{a}} &= \tilde{C}_{\mathrm{a}} \tilde{\boldsymbol{u}}_{\mathrm{a}} + \bar{D}_{\mathrm{a}} \boldsymbol{u}_{\mathrm{s}}, \end{aligned} \tag{39}$$

where

$$\bar{B}_{\mathrm{a}} = \begin{bmatrix} \sigma_{\mathrm{t}} \tilde{B}_{\mathrm{a},1} K_m T_{\mathrm{s}} & \tilde{B}_{\mathrm{a},2} K_m T_{\mathrm{s}} \end{bmatrix}, \qquad \bar{D}_{\mathrm{a}} = \begin{bmatrix} 0 & \tilde{D}_{\mathrm{a}} K_m T_{\mathrm{s}} \end{bmatrix}.$$

Next, the aerodynamic forces are applied to the structure. Rewriting the forces on the right-hand side of the structural model (31) using force transformation matrix in equation (37) and the aerodynamic force approximation from (39) gives

$$\frac{d\boldsymbol{u}_{\mathrm{s}}}{dt_{\mathrm{s}}} = \left(A_{\mathrm{s}} + \sigma_{\mathrm{f}} T_{\mathrm{a}} \bar{D}_{\mathrm{a}} \right) \boldsymbol{u}_{\mathrm{s}} + \sigma_{\mathrm{f}} T_{\mathrm{a}} \tilde{C}_{\mathrm{a}} \tilde{\boldsymbol{u}}_{\mathrm{a}} \tag{40}$$

Finally, there remains only the task of creating the monolithic aeroelastic ROM from equations (39) and (40),

$$\mathbb{M} \frac{d\boldsymbol{u}}{dt_{\mathrm{s}}} = \mathbb{A} \boldsymbol{u}, \tag{41}$$

where

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{s}} \\ \tilde{\boldsymbol{u}}_{\mathrm{a}} \end{bmatrix}, \qquad \mathbb{M} = \begin{bmatrix} I & 0 \\ 0 & \sigma_{\mathrm{t}} \tilde{E}_{\mathrm{a}} \end{bmatrix}, \qquad \mathbb{A} = \begin{bmatrix} A_{\mathrm{s}} + \sigma_{\mathrm{f}} T_{\mathrm{a}} \bar{D}_{\mathrm{a}} & \sigma_{\mathrm{f}} T_{\mathrm{a}} \tilde{C}_{\mathrm{a}} \\ \bar{B}_{\mathrm{a}} & \tilde{A}_{\mathrm{a}} \end{bmatrix}, \tag{42}$$

where $I \in \mathbb{R}^{2n_s \times 2n_s}$ is the identity matrix. Note that for symmetric airfoils at 0° angle of attack, this system is only dependent on the dynamic pressure q_∞ (or the speed index V) through the scaling factors σ_t and σ_f. The generalized eigenvalue problem for the aeroelastic system is: find an eigenvalue λ and eigenvector $\boldsymbol{v}$ such that,

$$\mathbb{A}\boldsymbol{v} = \lambda \mathbb{M}\boldsymbol{v}. \tag{43}$$

Through the application of model order reduction, all the eigenvalues of this system can be obtained rapidly. By performing a sweep across a range of dynamic pressures, the stability of the system can be determined. The point at which any of the eigenvalues in the system cross the imaginary axis (i.e. gain a positive real value) indicates the onset of flutter.

C. Full-Order Model Based Flutter Predictions

For validation purposes, the results of the ROM-based flutter predictions are compared to the full-order model predictions, including both the full-order nonlinear Euler model and the full-order linearized Euler model. Due to the high-dimension of the full-order aerodynamic model, the use of an eigenvalue analysis to determine the stability of the aeroelastic system is intractable. Hence, this work relies on a time marching approach to determine the flutter boundary of the full-order aeroelastic system. This methodology is based on the one described by Edwards *et al*. [51]. In this approach, the structural model (31) is marched in time with the following method,

$$\boldsymbol{u}_s^{(n+1)} = \exp(A_s \Delta t)\boldsymbol{u}_s^{(n)} + \left[\int_0^{\Delta t} \exp(A_s(\Delta t - \tau)d\tau\right] \boldsymbol{f}_s^{(n)}, \tag{44}$$

where Δt is the time step size, and n is the step index. For each time step, the aerodynamic forces are first computed using either the full-order nonlinear or the linearized Euler equations marched in time. These forces are transferred to the structural model, which then determines the structural deformation. Deformations are transferred to the aerodynamic mesh, and computations are repeated until the aerodynamic forces have converged to a relative tolerance of $O(10^{-6})$ between iterations. Typically, four iterations are required to complete one step in time.

In order to determine the flutter boundary using the full-order nonlinear and linearized Euler equations, at least two time-dependent simulations are required. With two or more simulations, the flutter point (the point of zero aeroelastic damping) can be interpolated or extrapolated based on the damping values obtained at the simulated conditions. For accurate results, the simulated conditions must be sufficiently close to the flutter point. Time histories of the aeroelastic transients are fit to a damped sinusoidal wave using the methodology of Bennett and Desmarais [52]. This allows for a damping factor to be obtained for each dynamic pressure of interest.

It is important to note that the full-order time-marched aeroelastic simulations were not optimized to reduce

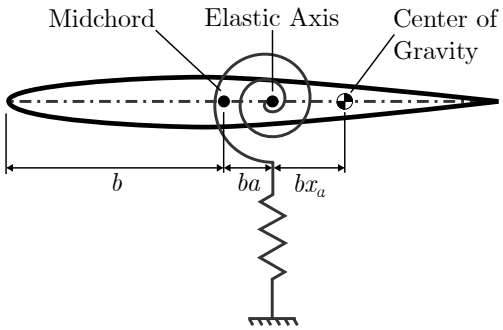


Fig. 2 Two degree of freedom airfoil structure capable of pitching and plunging.

computational expense. However, the total time reported in the results for the full-order approach assumes the best case scenario for which only two simulations are performed at flight conditions near the flutter boundary. With aircraft for which flutter behavior is not known *a priori*, increased computational costs can be anticipated because the aeroelastic transients may be required for more than two dynamic pressures.

VI. Test Case Description

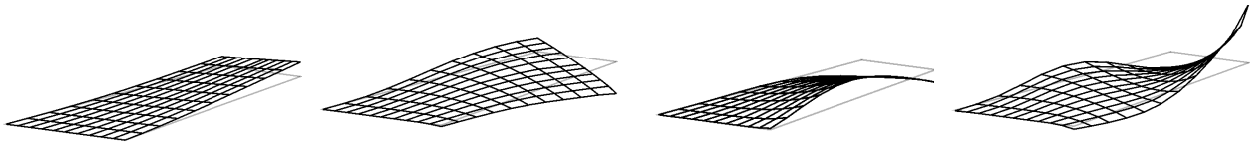
This section presents the two structural models used as validation cases: a two degree of freedom structural model, and the AGARD 445.6 wing structural model.

A. Two Degree of Freedom Aeroelastic Test Case

To test the accuracy and efficiency of the proposed approach for fast flutter predictions, the two degree of freedom airfoil structure shown in Figure 2 has been implemented. The structure is supported by a torsional and a linear spring and is capable of both pitching and plunging. For the equations of motion presented in (29) and (30), the mass and stiffness matrices and the applied force vector are:

$$M_{\mathrm{s}} = \begin{bmatrix} 1 & x_\alpha \\ x_\alpha & r_\alpha^2 \end{bmatrix}, \quad K_{\mathrm{s}} = \begin{bmatrix} (\omega_h/\omega_\alpha)^2 & 0 \\ 0 & r_\alpha^2 \end{bmatrix}, \quad \boldsymbol{g}_{\mathrm{s}} = \begin{bmatrix} -C_{\mathrm{l}} \\ 2C_{\mathrm{m}} \end{bmatrix},$$

where x_α is the distance from the elastic axis to the center of gravity (divided by b), r_α is the radius of gyration (divided by b), and ω_h and ω_α are the uncoupled plunging and pitching natural frequencies, respectively. C_{l} and C_{m} are the lift and pitching moment coefficients, the latter taken about the elastic axis location. Additionally, a is defined as the location of the elastic axis (divided by b) relative to the midchord. Aeroelastic results are presented for Case A of Isogai [53]. Parameter values used are: $a = -2.0$, $x_\alpha = 1.8$, $r_\alpha = 1.865$, $\omega_h = 100$ rad/s, $\omega_\alpha = 100$ rad/s, and $\mu = 60$. Note that this value of a corresponds to an elastic axis location ahead of the leading edge; this replicates the behaviors of a swept back wing. Flutter predictions are performed about a steady-state angle of attack of 0°.



(a) Mode 1 **(b) Mode 2** **(c) Mode 3** **(d) Mode 4**

Fig. 3 First four mode shapes of the AGARD 445.6 wing structure.

All simulations pertaining to this model are performed on the NACA 64A010 airfoil with a structured mesh. The mesh consists of a two-dimensional mesh with 38,000 nodes around the airfoil extruded to create 11 span-wise sections, resulting in a total of 418,000 nodes subdivided into 32 blocks. All results and snapshots are obtained using 32 Intel "Skylake" processors at 2.4 GHz.

B. AGARD Wing Aeroelastic Test Case

As a three-dimensional aeroelastic test case, the weakened AGARD 445.6 wing model was chosen, for which wind tunnel results were obtained by Yates [54]. The structural model provided by Yates [54] is composed of four natural frequencies and associated mode shapes, the latter shown in Figure 3. Simulations are performed on the AGARD 445.6 wing geometry using a structured mesh with 1,987,392 nodes subdivided into 160 blocks. All results and snapshots are obtained using 160 Intel "Skylake" processors at 2.4 GHz.

VII. Results

This section presents results obtained for the various components of the flutter prediction algorithm, followed by complete flutter predictions. Results are first presented for the linearized Euler equations compared to the nonlinear equations for a transonic pitching airfoil test case. Subsequently, numerical results are given demonstrating the stabilizing inner product and the incremental POD approaches. Lastly, flutter predictions are presented for the two degree of freedom aeroelastic test case and the AGARD 445.6 wing case. An aeroelastic transient time history comparison between the ROM and the full-order linearized Euler equations is presented for the AGARD wing.

A. Results for the Linearized Discrete Euler Equations

Results are presented to assess the accuracy of the linearized discrete Euler equations for unsteady aerodynamic force approximations. A popular test case for unsteady transonic flow is the pitching NACA 64A010 airfoil at a freestream Mach number of 0.796, for which experimental data is provided by Davis [55]. The airfoil pitches at a reduced frequency ($\omega b/V_\infty$) of 0.202, with a mean incidence of 0°, and a pitching amplitude of ±1.01°. The case was run using both the nonlinear and linearized Euler equations.

Simulations were run using the implicit Euler time-marching method with a time step size of 0.02 nondimensional time units, equivalent to 977 steps per pitching cycle. This small time step size was chosen to ensure that time-marching

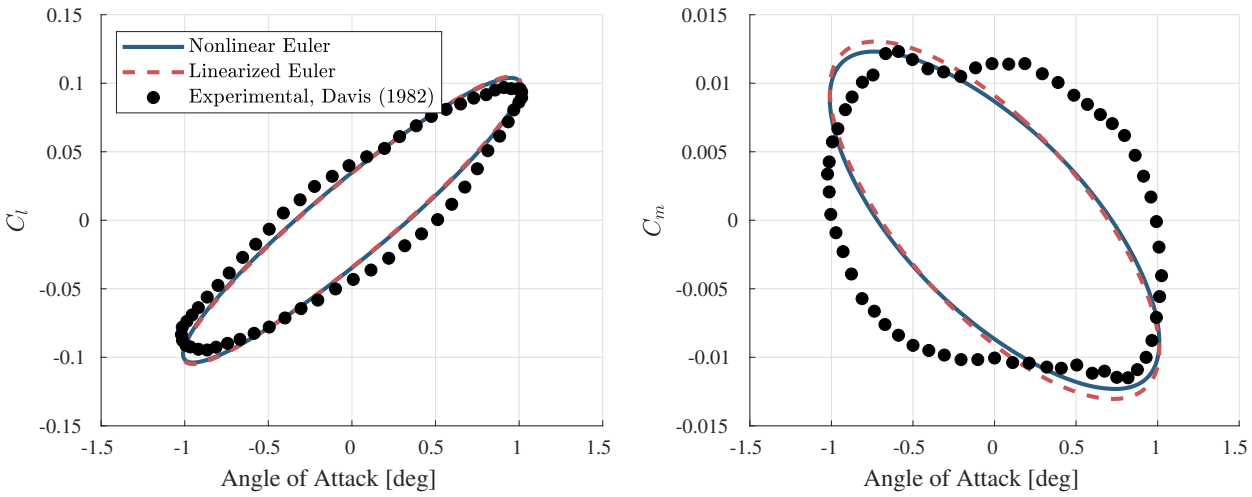


Fig. 4 C_l (left) and C_m (right) versus angle of attack for the nonlinear and linearized Euler equations, and experimental results obtained by Davis [55].

errors were negligible. Figure 4 shows the evolution of C_l and C_m, the lift and pitching moment coefficients, as a function of the angle of attack α for the sixth cycle of motion. One can see that, though some error is present, in general the linearized Euler equations are capable of capturing similar behavior as their nonlinear counterpart for this case. Quantitatively, the maximum difference in C_l and C_m between the linearized and nonlinear equations for the sixth pitching cycle are found to be 1.26% and 6.10%, respectively. From both the computational and experimental results, a shock wave is observed which oscillates about the mid-chord. This indicates that the linearized Euler equations are able to capture some shock wave motion. Discrepancies between the experimental and computational results are attributed, in part, to the absence of shock-boundary-layer interactions due to the use of the Euler equations, as discussed by Chyu *et al*. [56].

When predicting the onset of flutter, only infinitely small oscillations need be considered [6]. Indeed, the two degree change in angle of attack used for this analysis is much larger than is required for flutter prediction. Thus, it is expected that the differences between the nonlinear and linearized Euler equations will become negligible for flutter analysis. However, as will be observed in the results for the AGARD 445.6 wing test case, the exclusion of viscosity can lead to errors in the flutter boundary at higher Mach numbers.

B. Results for Stabilizing Inner Product

The use of the weighted inner product presented in Section III.B is demonstrated. The stability of the ROM is determined by the real part of the eigenvalues of the matrix pencil $(\tilde{A}_\mathrm{a}, \tilde{E}_\mathrm{a})$, where $\tilde{A}_\mathrm{a}$ is the reduced flux Jacobian and $\tilde{E}_\mathrm{a}$ the reduced diagonal matrix of inverse metric Jacobians, first shown in equation (19). If any eigenvalue possesses a positive real part, then the aerodynamic ROM is unstable. One may see parallels between this stability analysis and the flutter predictions outlined above. It is important to note that for flutter predictions, the instability comes from the

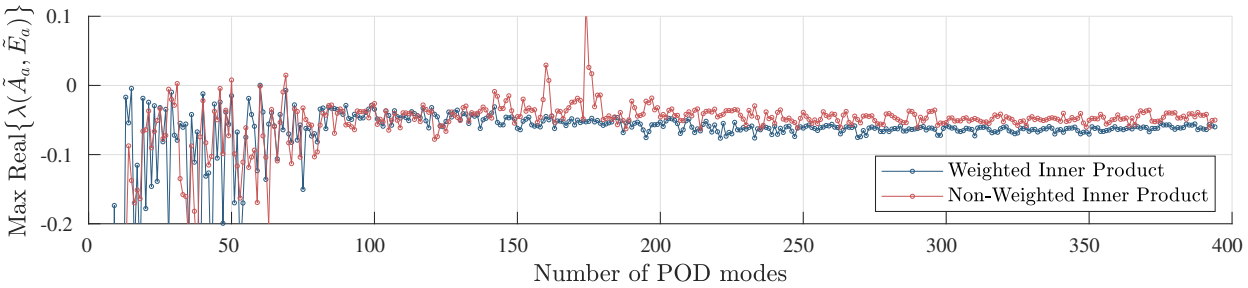


Fig. 5 Comparison between the stability of ROMs with weighted and non-weighted L^2 inner products for various numbers of POD basis vectors.

coupling between the aerodynamic model and the structural model. If the ROM used for aerodynamic predictions is inherently unstable, the flutter boundary cannot be predicted.

To test the stabilizing properties, the structural model used is the two degree of freedom model presented in Section VI.A. Four thousand snapshots were collected using the Walsh functions to excite the structural states, with a time step size of 0.004 nondimensional time units. The standard POD approach is used to obtain a reduced basis from these snapshots.

Figure 5 shows a comparison between the weighted and non-weighted L^2 inner products for the Galerkin projection. The largest real parts of all the eigenvalues of the matrix pencil $(\tilde{A}_\mathrm{a}, \tilde{E}_\mathrm{a})$ are plotted as a function of the number of POD modes used for the Galerkin projection. It is immediately apparent that for various numbers of POD modes the non-weighted L^2 inner product results in an unstable ROM. One observes several numbers of POD modes for which the largest positive real part of the eigenvalues is above zero. The weighted inner product demonstrates stability for almost all numbers of POD modes, save for at 60 modes where the largest real part of the eigenvalues has a value of approximately 2×10^{-4}. This instability is believed to be caused by the use of a discrete approximation of the continuous inner product.

C. Results for Incremental POD

The performance of the incremental POD approach presented in Section III.C is demonstrated. As for the test of the stabilizing inner product, the two degree of freedom structural model test case was used. Two thousand snapshots were collected using the Walsh functions to excite the structural states, with a time step size of 0.1 nondimensional time units. Both the standard POD and incremental POD methods are used to create the reduced basis from these snapshots. The standard POD approach required all 2000 snapshots to be computed and stored before the basis is created. For the incremental approach, the first 400 snapshots are used to create the initial basis, and each subsequent batch of 200 snapshots is used to update the existing basis. The largest memory requirements for the incremental approach is thus 600 vectors, a 70 % reduction in memory. For the current test case, taking into account all other memory requirements including the system matrices and preconditioner, the incremental approach showed a reduction of 40 % in total memory

usage.

Similar behaviors in the decay of the singular values is observed between the standard and incremental POD approaches until they reach values of approximately 10^{-10}, where larger discrepancies are observed. Quantitatively, the first 200 singular values, which together represent a POD reconstruction error of $\eta_{\text{ROM}} = 7.245 \times 10^{-12}$ from equation (15), have an average relative error of 1.147 %. Comparing only the first 10 singular values, which represent a value of $\eta_{\text{ROM}} = 5.073 \times 10^{-4}$, an average relative error of 0.5268 % is observed. The ROMs constructed using each approach produce almost identical outputs.

D. Two Degree of Freedom Flutter Predictions

A comparison of the flutter boundaries obtained for the two degree of freedom test case from Section VI.A using the full-order equations and the ROM methodology is presented. The aerodynamic ROMs used for this test case were constructed using the procedure outlined in Section III. Incremental POD was used to create the reduced basis from snapshots collected using the Walsh function excitations. Through experimentation, it was found that 2000 snapshots for each Mach number with a time step size of 0.2 nondimensional time units resulted in ROMs capable of accurate flutter predictions. Using the POD-based approach, the ROMs were constructed for five Mach numbers ranging from 0.75 to 0.875, each with between 150 and 188 modes obtained using a tolerance $\eta_{\text{ROM}} = 10^{-10}$ from equation (15). This reduction in dimension is significant compared to the 2.09×10^6 degrees of freedom in the original system. This value of η_{ROM} may seem excessively small, but note that the time required to determine the flutter boundary using the ROMs, even with hundreds of POD modes, is insignificant compared to the cost of collecting the snapshots.

Results for the flutter boundary obtained by the nonlinear and linearized Euler equations, the ROM, and those found in the literature are shown in Figure 6. Results from Alonso and Jameson [57] and Sanchez *et al*. [58] were obtained using the Euler equations. All methods used in the present work have good agreement with the results from the literature for all Mach numbers. The transonic dip around Mach 0.85 is observed in the results. This is the result of nonlinear behavior in the flow field, specifically the oscillation of the shock wave on the upper surface [53]. This indicates that the ROM is capable of capturing this nonlinear flow behavior, which would not be captured by a linear aerodynamic method. Additionally, one observes two points at Mach 0.875: an onset of flutter at $V_{\text{f}} \approx 0.6$, and the return of stability at $V_{\text{f}} \approx 1.8$. Of these two points, the ROM predicts the former well, but is incapable of predicting the latter. This is not a vital concern as the primary goal of the ROM is to find the initial onset of flutter.

A comparison of the total time (measured in equivalent number of steady flow solves) required to obtain the flutter boundary for all Mach numbers is presented in Table 1. As expected, the nonlinear and linearized Euler equations require significantly more time than the ROM. However, as mentioned above, the methodology used to obtain the flutter boundary for the full-order models was not optimized for computational time, and further reduction in these times may be expected. Despite these caveats, the use of the ROM for flutter prediction performed considerably faster. Notably,

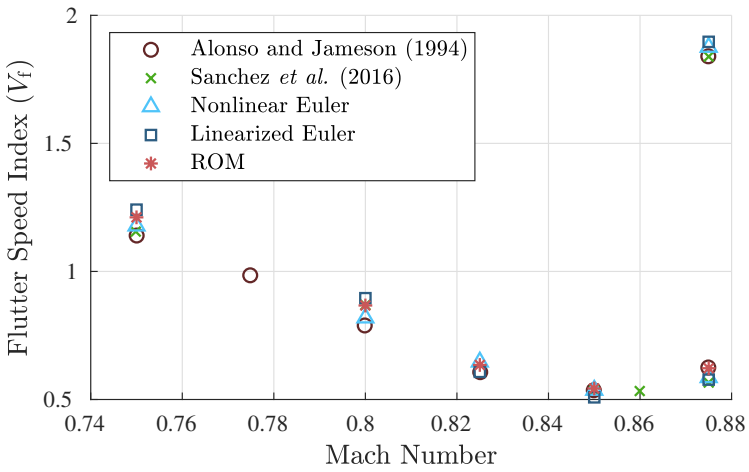


Fig. 6 Flutter boundary obtained by Alonso and Jameson [57] and Sanchez *et al.* [58], by the full-order nonlinear and linearized Euler equations, and by the ROM-based approach.

Table 1 Total time required to obtain the flutter boundary for all Mach numbers of interest, measured in equivalent number of steady flow solves.

	Nonlinear Euler	Linearized Euler	ROM
Total Time [equivalent number of steady solves]	1751	783	46

most of the time for the ROM was in collecting the snapshots for basis construction. On average, 92.1 % of the time was required to collect the snapshots and construct the reduced basis, 7.77 % was required for the steady flow solve, while only 0.134 % was required to perform the eigenvalue analyses of the ROM-based aeroelastic system at all values of V_f.

E. AGARD Wing Aeroelastic Test Case

A comparison of the aeroelastic transients and the flutter boundary obtained for the AGARD 445.6 wing test case from Section VI.B is presented. The aerodynamic ROMs used for the AGARD 445.6 wing were constructed using the procedure outlined in Section III. Snapshots were collected using the Walsh function excitations, and incremental POD was used to create the reduced basis. It was determined through experimentation that 3000 snapshots for each Mach number with a time step size of 2.0 nondimensional time units produced accurate ROMs. The ROMs were constructed for six Mach numbers ranging from 0.499 to 1.141, each with between 258 and 320 modes obtained using $\eta_{\text{ROM}} = 10^{-10}$.

Figure 7 shows the time history of the generalized coordinates of the AGARD wing structural model coupled to the aerodynamic ROM. The time history of the same structural model coupled to the full-order linearized Euler equations is also presented. All aeroelastic transients are initialized using a perturbation in the velocity of the structure. From this figure, it is observed that the aerodynamic ROM is capable of producing similar aeroelastic transients as those obtained using the full-order linearized Euler equations. The damping factor for each of the displacements is well captured, indicating that instabilities in the aeroelastic system are reproducible with the ROM. Further comparison

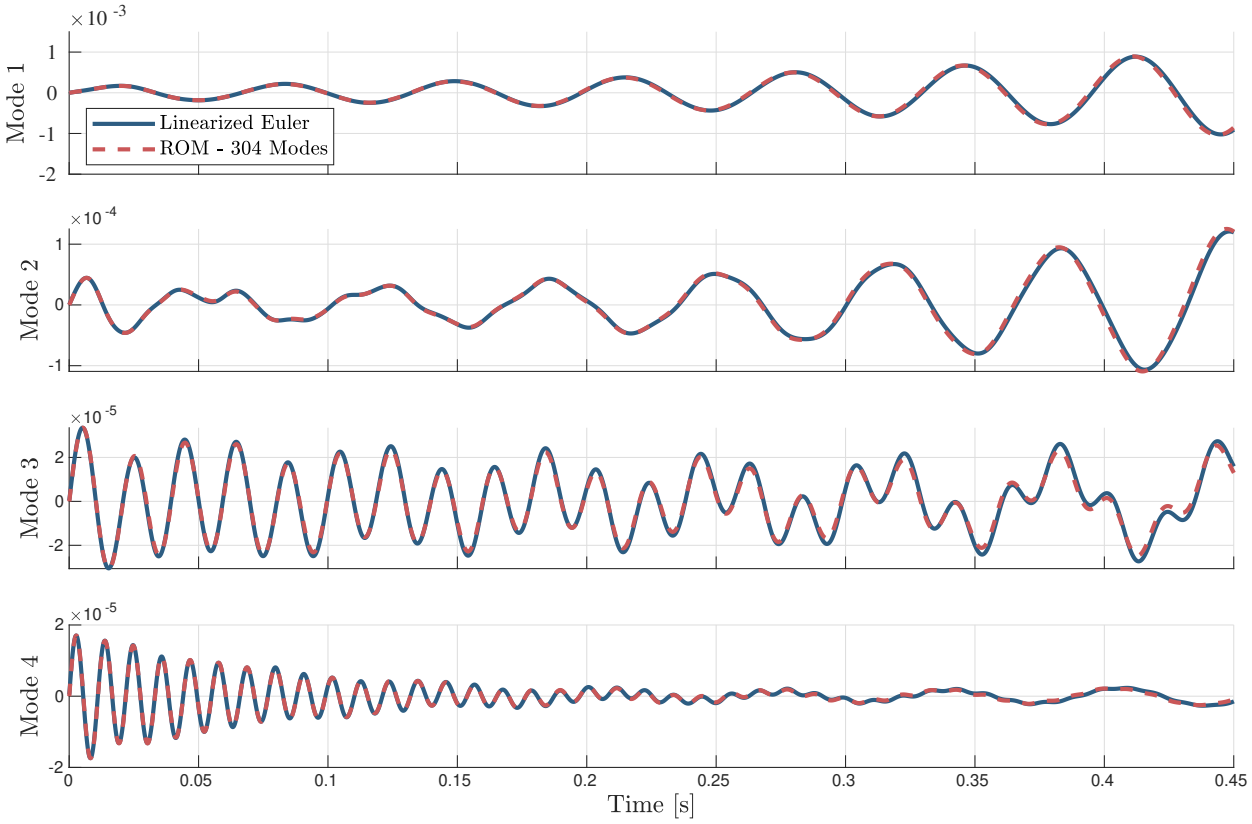


Fig. 7 Generalized structural coordinate displacements obtained by both the linearized Euler equations and the ROM for the AGARD 445.6 wing at a Mach number of 0.960 and dynamic pressure of 0.57 psi.

between the ROM and the full-order linearized Euler results is provided by looking at the frequency spectrum shown in Figure 8, obtained by performing a fast-Fourier transform of the time histories from Figure 7. It is observed that the frequency peaks in the solution are generally well captured by the ROM. Additionally, frequencies further away from the dominant peaks are also well captured by the ROM, with the largest discrepancies at the lowest frequencies in the solution. This indicates, at least for these flow conditions, that the use of the Walsh functions as structural excitations when collecting flow snapshots provides a sufficiently rich set of flow reactions to capture the relevant frequencies for dynamic aeroelastic analysis.

Results for the flutter boundary obtained for the AGARD wing are shown in Figure 9, from which several observations can be made. First, as observed by Lee-Rausch and Batina [59], the inclusion of viscosity in the solution is important for predicting accurate flutter behavior at Mach numbers close to unity. Indeed, the current ROM-based approach is not limited to inviscid equations and can be extended to the linearized discrete RANS equations. Second, the flutter boundary obtained by the linearized Euler equations compares well with the nonlinear inviscid results obtained by Silva *et al*. [15]. Finally, it is observed that the ROM-based approach demonstrates similar accuracy as the linearized Euler equations for all Mach numbers of interest except above Mach 1. Thus, in general the ROM shows an ability to closely

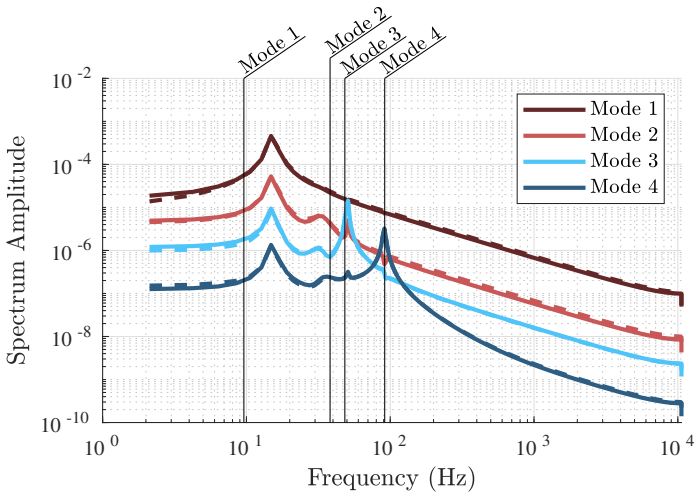


Fig. 8 Frequency spectrum of the time history response for the AGARD wing shown in Figure 7; solid lines are the full-order model, dashed lines are the ROM, black vertical lines indicate the natural frequency for each mode.

mimic the governing equations from which it is derived for the lower range of the Mach numbers presented. For this work, this lower range is of primary interest as this methodology is aimed at flutter prediction for subsonic commercial aircraft.

The behavior of an aerodynamic ROM for the prediction of flutter for the AGARD wing at Mach 1.141 has been previously analyzed by Silva *et al.* [15]. With the use of the root locus plot of the aeroelastic system, Silva *et al.* concluded that for the ROM constructed based on the Euler equations at this Mach number, the third structural mode demonstrated instabilities at all dynamic pressures. In the present work, instabilities were observed at dynamic pressures above 0.455 psi, well below those predicted by the full-order linearized Euler equations. This instability is difficult to capture using the time-marched aeroelastic transient approach used for the full-order equations due to the high frequency content of the third mode's response, which makes it difficult to fit with a damped sinusoidal wave. At lower Mach numbers, the first mode demonstrates the aeroelastic instability; it is thus assumed that at Mach 1.141, this mode will also be responsible for flutter. The third mode's transients are stabilized with the inclusion of viscosity, i.e. with the use of the RANS equations [15]. Ultimately, it can be concluded that the use of the Euler equations is inappropriate for determining the flutter boundary of the AGARD wing at Mach numbers above unity.

Table 2 shows the total time required to obtain the flutter boundary for all Mach numbers of interest, measured in terms of the equivalent number of steady flow solves. The use of the ROM for flutter prediction is considerably faster than solving the full-order linearized discrete Euler equations.

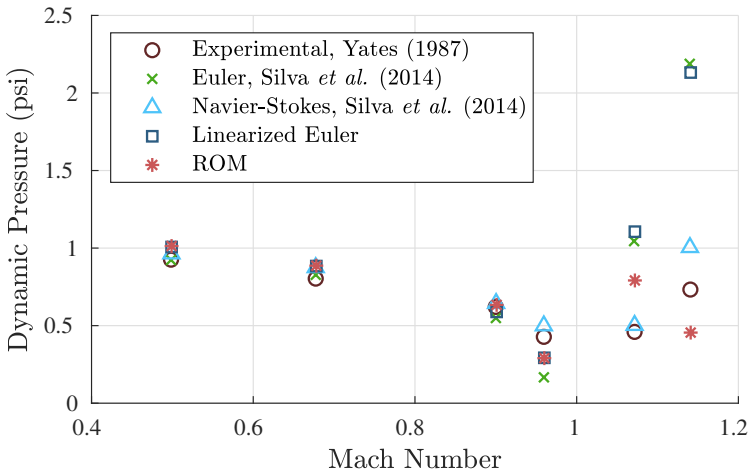


Fig. 9 Flutter boundary for the AGARD 445.6 wing obtained experimentally by Yates [54], computationally by Silva *et al*. [15], by the linearized discrete Euler equations, and by the current ROM approach.

Table 2 Total time required to obtain the flutter boundary for the AGARD wing for all Mach numbers, measured in equivalent number of steady flow solves.

	Linearized Euler	ROM
Total Time [equivalent number of steady solves]	1101	182

VIII. Conclusions

This paper presents a methodology for flutter predictions using a POD-based model order reduction approach. Snapshots for the ROMs are created with a simultaneous excitation method, using the Walsh functions as input functions. The use of a stabilizing inner product which symmetrizes the flux Jacobian has been shown to be effective in ensuring the stability of the ROMs.

The resulting aerodynamic ROMs are of sufficiently low order that the determination of a flutter boundary can be cast as a tractable eigenvalue problem. The concepts presented herein have been shown to be effective for predicting the flutter boundary of a two degree of freedom airfoil structure and the three-dimensional AGARD 445.6 wing for most Mach numbers. Significant time reductions are obtained relative to the full-order flutter prediction methodology. The efficiency of the methodology presented could enable implementation of a flutter constraint in aerodynamic shape optimization. Moreover, the discrepancy seen for Mach numbers above unity due to the inviscid flow approximation is not important for application to subsonic commercial aircraft. However, if applications to supersonic aircraft are of interest, the present methodology can be extended to the RANS equations.

Appendix: Proof of Theorem III.1

The proof for Theorem III.1 is presented, which has been modified from Kalashnikova *et al*. [35] to generalize the boundary condition treatment and to include a source term contribution.

Proof. The proof of the stability of the system follows. Note that the δ's denoting fluctuating quantities are omitted from the continuous linearized Euler equations (20) for clarity. First, equation (20) is premultiplied by the determinant of the metric Jacobian $\mathcal{J}_0$ and a matrix $\mathcal{W}$ which symmetrizes the flux Jacobian $\mathcal{A}_{(i)}$. Using the product rule to expand the spatial derivative gives:

$$\begin{aligned} \mathcal{W}\frac{\partial \mathcal{U}}{\partial t} + \mathcal{J}_0\mathcal{W}\mathcal{A}_{(i)}\frac{\partial \mathcal{U}}{\partial \xi_{(i)}} + \mathcal{J}_0\mathcal{W}\frac{\partial \mathcal{A}_{(i)}}{\partial \xi_{(i)}}\mathcal{U} = \mathcal{J}_0\mathcal{W}\mathcal{S}, \\ \mathcal{W}\frac{\partial \mathcal{U}}{\partial t} + \mathcal{B}_{(i)}\frac{\partial \mathcal{U}}{\partial \xi_{(i)}} + \mathcal{C}\mathcal{U} = \mathcal{J}_0\mathcal{W}\mathcal{S}, \end{aligned} \tag{45}$$

where $\mathcal{B}_{(i)} = \mathcal{J}_0\mathcal{W}\mathcal{A}_{(i)}$, and $\mathcal{C} = \mathcal{J}_0\mathcal{W}\frac{\partial \mathcal{A}_{(i)}}{\partial \xi_{(i)}}$. To perform an energy stability analysis on this form, the equation is premultiplied with $\mathcal{U}^T$ and integrate over the domain,

$$\begin{aligned} \int_\Omega \mathcal{U}^T\mathcal{W}\frac{\partial \mathcal{U}}{\partial t}d\Omega + \int_\Omega \mathcal{U}^T\mathcal{B}_{(i)}\frac{\partial \mathcal{U}}{\partial \xi_{(i)}}d\Omega + \int_\Omega \mathcal{U}^T\mathcal{C}\mathcal{U}d\Omega = \int_\Omega \mathcal{U}^T\mathcal{J}_0\mathcal{W}\mathcal{S}d\Omega, \\ \frac{1}{2}\frac{d}{dt}||\mathcal{U}||^2_{\mathcal{W}} + \int_\Omega \mathcal{U}^T\mathcal{B}_{(i)}\frac{\partial \mathcal{U}}{\partial \xi_{(i)}}d\Omega + \frac{1}{2}\int_\Omega \mathcal{U}^T(\mathcal{C}+\mathcal{C}^T)\mathcal{U}d\Omega = \int_\Omega \mathcal{U}^T\mathcal{J}_0\mathcal{W}\mathcal{S}d\Omega. \end{aligned} \tag{46}$$

Note that the matrix $\mathcal{B}_{(i)}$ is symmetric. This allows the use of integration by parts and the product rule to show that

$$\int_\Omega \mathcal{U}^T\mathcal{B}_{(i)}\frac{\partial \mathcal{U}}{\partial \xi_{(i)}}d\Omega = \frac{1}{2}\int_{\partial\Omega}\mathcal{U}^T\mathcal{B}_{(i)}n_{(i)}\mathcal{U}d\Gamma - \frac{1}{2}\int_\Omega \mathcal{U}^T\frac{\partial \mathcal{B}_{(i)}}{\partial \xi_{(i)}}\mathcal{U}d\Omega. \tag{47}$$

Inserting this into equation (46), and removing the common $\frac{1}{2}$ factors gives,

$$\frac{d}{dt}||\mathcal{U}||^2_{\mathcal{W}} = -\int_{\partial\Omega}\mathcal{U}^T\mathcal{B}_{(i)}n_{(i)}\mathcal{U}d\Gamma + \int_\Omega \mathcal{U}^T\left(\frac{\partial \mathcal{B}_{(i)}}{\partial \xi_{(i)}} - (\mathcal{C}+\mathcal{C}^T)\right)\mathcal{U}d\Omega + 2\int_\Omega \mathcal{U}^T\mathcal{J}_0\mathcal{W}\mathcal{S}d\Omega. \tag{48}$$

Defining the new matrix $\mathcal{D} = \frac{\partial \mathcal{B}_{(i)}}{\partial \xi_{(i)}} - (\mathcal{C}+\mathcal{C}^T)$, and inserting the known quantities at the boundaries $\mathcal{U} = \mathcal{G}(t)$ gives,

$$\frac{d}{dt}||\mathcal{U}||^2_{\mathcal{W}} = -\int_{\partial\Omega}\mathcal{G}^T\mathcal{B}_{(i)}n_{(i)}\mathcal{G}d\Gamma + \int_\Omega \mathcal{U}^T\mathcal{D}\mathcal{U}d\Omega + 2\int_\Omega \mathcal{U}^T\mathcal{J}_0\mathcal{W}\mathcal{S}d\Omega. \tag{49}$$

Defining $\gamma_{\mathcal{D}}$ as the continuity constant of the linear operator $\mathcal{D}$, and using the equivalence between the $L^2(\Omega)$ and $\mathcal{W}$ norms gives,

$$\int_\Omega \mathcal{U}^T\mathcal{D}\mathcal{U}d\Omega \leq \gamma_{\mathcal{D}}||\mathcal{U}||^2_{L^2(\Omega)} \leq \tilde{\gamma}_{\mathcal{D}}||\mathcal{U}||^2_{\mathcal{W}}. \tag{50}$$

Similarly, assuming the sesquilinear form of the source term to be appropriately bounded, with continuity constant $\gamma_{\mathcal{S}}$ gives,

$$\int_\Omega \mathcal{U}^T\mathcal{J}_0\mathcal{W}\mathcal{S}d\Omega \leq \gamma_{\mathcal{S}}||\mathcal{U}||^2_{L^2(\Omega)} \leq \tilde{\gamma}_{\mathcal{S}}||\mathcal{U}||^2_{\mathcal{W}}. \tag{51}$$

Next, assuming the quantities at the boundary are bounded gives,

$$-\int_{\partial\Omega} \mathcal{G}^T \mathcal{B}_{(i)} n_{(i)} \mathcal{G} d\Gamma \leq k||\mathcal{G}||^2_{L^2(\partial\Omega)}. \tag{52}$$

Substituting (50), (51), and (52) into (49), and rearranging leads to

$$\frac{d}{dt}||\mathcal{U}||^2_{\mathcal{W}} - (\tilde{\gamma}_{\mathcal{D}} + 2\tilde{\gamma}_{\mathcal{S}})\,||\mathcal{U}||^2_{\mathcal{W}} \leq k||\mathcal{G}||^2_{L^2(\partial\Omega)}. \tag{53}$$

Denote $\tilde{\gamma} = \tilde{\gamma}_{\mathcal{D}} + 2\tilde{\gamma}_{\mathcal{S}}$, and premultiply this equation with the integration factor $e^{-\tilde{\gamma}t}$ to obtain

$$e^{-\tilde{\gamma}t}\frac{d}{dt}||\mathcal{U}||^2_{\mathcal{W}} - e^{-\tilde{\gamma}t}\tilde{\gamma}||\mathcal{U}||^2_{\mathcal{W}} = \frac{d}{dt}\left(e^{-\tilde{\gamma}t}||\mathcal{U}||^2_{\mathcal{W}}\right) \leq e^{-\tilde{\gamma}t}k||\mathcal{G}||^2_{L^2(\partial\Omega)}. \tag{54}$$

Integrating (54) in time from 0 to T gives

$$\begin{aligned} e^{-\tilde{\gamma}T}||\mathcal{U}(\cdot,T)||^2_{\mathcal{W}} - ||\mathcal{U}(\cdot,0)||^2_{\mathcal{W}} &\leq k\int_0^T e^{-\tilde{\gamma}t}||\mathcal{G}||^2_{L^2(\partial\Omega)}dt, \\ ||\mathcal{U}(\cdot,T)||^2_{\mathcal{W}} &\leq k\int_0^T e^{\tilde{\gamma}(T-t)}||\mathcal{G}||^2_{L^2(\partial\Omega)}dt + e^{\tilde{\gamma}T}||\mathcal{U}(\cdot,0)||^2_{\mathcal{W}}. \end{aligned} \tag{55}$$

Focusing on the first term of the right-hand side leads to,

$$k\int_0^T e^{\tilde{\gamma}(T-t)}||\mathcal{G}||^2_{L^2(\partial\Omega)}dt \leq ke^{\tilde{\gamma}T}\int_0^T ||\mathcal{G}||^2_{L^2(\partial\Omega)}dt = \tilde{k}||\mathcal{G}||^2_{\mathcal{G}}. \tag{56}$$

Introducing (56) into (55) and substituting for the initial conditions gives,

$$||\mathcal{U}(\cdot,T)||^2_{\mathcal{W}} \leq \tilde{k}||\mathcal{G}||^2_{\mathcal{G}} + e^{\tilde{\gamma}T}||\mathcal{F}||^2_{\mathcal{W}}. \tag{57}$$

Assuming the state vector evolves in a finite number of spatial modes, $\mathcal{U}(x,t) = \sum_i \phi_i(x)\tilde{\mathcal{U}}_i(t)$ gives,

$$||\mathcal{U}(\cdot,T)||^2_{\mathcal{W}} = ||\sum_i \phi_i\tilde{\mathcal{U}}_i(T)||^2_{\mathcal{W}} = \sum_i\sum_j \tilde{\mathcal{U}}_i(T)\tilde{\mathcal{U}}_j(T)\int_\Omega \phi_i\mathcal{W}\phi_j d\Omega = ||\tilde{\mathcal{U}}(T)||^2_{\mathcal{W}}. \tag{58}$$

Introducing (58) into (57) gives

$$||\tilde{\mathcal{U}}(T)||^2_{\mathcal{W}} \leq \tilde{k}||\mathcal{G}||^2_{\mathcal{G}} + e^{\tilde{\gamma}T}||\mathcal{F}||^2_{\mathcal{W}}. \tag{59}$$

Assuming that both the boundary and initial conditions exist in the reduced basis, and observing that $k = \max(\tilde{k}, e^{\tilde{\gamma}T})$ completes the proof. □

Acknowledgements

This research was enabled in part by the financial support provided by Bombardier, the Government of Ontario, the Natural Sciences and Research Council of Canada, and the University of Toronto. Computational resources were provided by SciNet High Performance Computing Consortium, a part of Compute Canada. The authors are extremely grateful for the input and support provided by Masayuki Yano.

References

[1] Hicken, J. E., and Zingg, D. W., "Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement," *AIAA Journal*, Vol. 48, No. 2, 2010, pp. 400–413. https://doi.org/10.2514/1.44033.

[2] Osusky, L., Buckley, H., Reist, T., and Zingg, D. W., "Drag Minimization Based on the Navier–Stokes Equations Using a Newton–Krylov Approach," *AIAA Journal*, Vol. 53, No. 6, 2015, pp. 1555–1577. https://doi.org/10.2514/1.J053457.

[3] Reist, T. A., and Zingg, D. W., "High-Fidelity Aerodynamic Shape Optimization of a Lifting-Fuselage Concept for Regional Aircraft," *Journal of Aircraft*, Vol. 54, No. 3, 2016, pp. 1085–1097. https://doi.org/10.2514/1.C033798.

[4] Zhang, Z. J., and Zingg, D. W., "Efficient Monolithic Solution Algorithm for High-Fidelity Aerostructural Analysis and Optimization," *AIAA Journal*, Vol. 56, No. 3, 2018, pp. 1251–1265. https://doi.org/10.2514/1.J056163.

[5] Mallik, W., Kapania, R. K., and Schetz, J. A., "Effect of Flutter on the Multidisciplinary Design Optimization of Truss-Braced-Wing Aircraft," *Journal of Aircraft*, Vol. 52, No. 6, 2015, pp. 1858–1872. https://doi.org/10.2514/1.C033096.

[6] Theodorsen, T., "General Theory of Aerodynamic Instability and the Mechanism of Flutter," Tech. Rep. 496, National Advisory Committee for Aeronautics, 1949.

[7] Jonsson, E., Mader, C. A., Kennedy, G., and Martins, J., "Computational Modeling of Flutter Constraint for High-Fidelity Aerostructural Optimization," *AIAA SciTech 2019 Forum*, San Diego, California, 2019. https://doi.org/10.2514/6.2019-2354.

[8] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *Computer Aided Design of Control Systems*, edited by M. A. Cuenod, Pergamon, 1980, pp. 113–117. https://doi.org/10.1016/S1474-6670(17)65584-8.

[9] Lupp, C. A., and Cesnik, C. E., "A Gradient-Based Flutter Constraint Including Geometrically Nonlinear Deformations," *AIAA SciTech 2019 Forum*, San Diego, California, 2019. https://doi.org/10.2514/6.2019-1212.

[10] Opgenoord, M. M. J., Drela, M., and Willcox, K. E., "Influence of Transonic Flutter on the Conceptual Design of Next-Generation Transport Aircraft," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 1973–1987. https://doi.org/10.2514/1.J057302.

[11] Jacobson, K., Stanford, B., Wood, S., and Anderson, W. K., "Flutter Analysis with Stabilized Finite Elements based on the Linearized Frequency-domain Approach," *AIAA SciTech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-0403.

[12] Jonsson, E., Riso, C., Lupp, C. A., Cesnik, C. E., Martins, J. R., and Epureanu, B. I., "Flutter and Post-Flutter Constraints in Aircraft Design Optimization," *Progress in Aerospace Sciences*, Vol. 109, 2019, p. 100537. https://doi.org/10.1016/j.paerosci.2019.04.001.

[13] Bendiksen, O. O., "Review of Unsteady Transonic Aerodynamics: Theory and Applications," *Progress in Aerospace Sciences*, Vol. 47, No. 2, 2011, pp. 135–167. https://doi.org/10.1016/j.paerosci.2010.07.001.

[14] Henshaw, M., Badcock, K., Vio, G., Allen, C., Chamberlain, J., Kaynes, I., Dimitriadis, G., Cooper, J., Woodgate, M., Rampurawala, A., Jones, D., Fenwick, C., Gaitonde, A., Taylor, N., Amor, D., Eccles, T., and Denley, C., "Non-Linear Aeroelastic Prediction for Aircraft Applications," *Progress in Aerospace Sciences*, Vol. 43, No. 4, 2007, pp. 65–137. https://doi.org/10.1016/j.paerosci.2007.05.002.

[15] Silva, W. A., Chwalowski, P., and Perry III, B., "Evaluation of Linear, Inviscid, Viscous, and Reduced-order Modelling Aeroelastic Solutions of the AGARD 445.6 Wing using Root Locus Analysis," *International Journal of Computational Fluid Dynamics*, Vol. 28, No. 3-4, 2014, pp. 122–139. https://doi.org/10.1080/10618562.2014.922179.

[16] Silva, W., "AEROM: NASA's Unsteady Aerodynamic and Aeroelastic Reduced-Order Modeling Software," *Aerospace*, Vol. 5, No. 2, 2018, pp. 41–58. https://doi.org/10.3390/aerospace5020041.

[17] Badcock, K. J., Timme, S., Marques, S., Khodaparast, H., Prandina, M., Mottershead, J., Swift, A., Da Ronch, A., and Woodgate, M., "Transonic Aeroelastic Simulation for Instability Searches and Uncertainty Analysis," *Progress in Aerospace Sciences*, Vol. 47, No. 5, 2011, pp. 392–423. https://doi.org/10.1016/j.paerosci.2011.05.002.

[18] Kim, T., "System Identification for Coupled Fluid-Structure: Aerodynamics is Aeroelasticity Minus Structure," *AIAA Journal*, Vol. 49, No. 3, 2011, pp. 503–512. https://doi.org/10.2514/1.J050245.

[19] Argaman, M., and Raveh, D. E., "Multioutput Autoregressive Aeroelastic System Identification and Flutter Prediction," *Journal of Aircraft*, Vol. 56, No. 1, 2019, pp. 30–42. https://doi.org/10.2514/1.C034789.

[20] Hall, K. C., Thomas, J. P., and Dowell, E. H., "Proper Orthogonal Decomposition Technique for Transonic Unsteady Aerodynamic Flows," *AIAA Journal*, Vol. 38, No. 10, 2000, pp. 1853–1862. https://doi.org/10.2514/2.867.

[21] Thomas, J. P., Dowell, E. H., and Hall, K. C., "Three-Dimensional Transonic Aeroelasticity Using Proper Orthogonal Decomposition-Based Reduced-Order Models," *Journal of Aircraft*, Vol. 40, No. 3, 2003, pp. 544–551. https://doi.org/10.2514/2.3128.

[22] Beran, P. S., Lucia, D. J., and Pettit, C. L., "Reduced-Order Modelling of Limit-Cycle Oscillation for Aeroelastic Systems," *Journal of Fluids and Structures*, Vol. 19, No. 5, 2004, pp. 575–590. https://doi.org/10.1016/j.jfluidstructs.2004.04.002.

[23] Anttonen, J. S., King, P. I., and Beran, P. S., "Applications of Multi-POD to a Pitching and Plunging Airfoil," *Mathematical and Computer Modelling*, Vol. 42, No. 3, 2005, pp. 245–259. https://doi.org/10.1016/j.mcm.2005.06.003.

[24] Lieu, T., Farhat, C., and Lesoinne, M., "Reduced-Order Fluid/Structure Modeling of a Complete Aircraft Configuration," *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 41, 2006, pp. 5730–5742. https://doi.org/10.1016/j.cma.2005.08.026.

[25] Lieu, T., and Farhat, C., "Adaptation of Aeroelastic Reduced-Order Models and Application to An F-16 Configuration," *AIAA Journal*, Vol. 45, No. 6, 2007, pp. 1244–1257. https://doi.org/10.2514/1.24512.

[26] Amsallem, D., Tezaur, R., and Farhat, C., "Real-Time Solution of Linear Computational Problems Using Databases of Parametric Reduced-Order Models with Arbitrary Underlying Meshes," *Journal of Computational Physics*, Vol. 326, 2016, pp. 373–397. https://doi.org/10.1016/j.jcp.2016.08.025.

[27] Bekemeyer, P., and Timme, S., "Flexible Aircraft Gust Encounter Simulation Using Subspace Projection Model Reduction," *Aerospace Science and Technology*, Vol. 86, 2019, pp. 805–817. https://doi.org/10.1016/j.ast.2019.02.011.

[28] Kou, J., and Zhang, W., "A Hybrid Reduced-Order Framework for Complex Aeroelastic Simulations," *Aerospace Science and Technology*, Vol. 84, 2019, pp. 880 – 894. https://doi.org/10.1016/j.ast.2018.11.014.

[29] Waite, J., Stanford, B., Bartels, R. E., and Silva, W. A., "Reduced Order Modeling for Transonic Aeroservoelastic Control Law Development," *AIAA SciTech 2019 Forum*, San Diego, California, 2019. https://doi.org/10.2514/6.2019-1022.

[30] Kim, T., Hong, M., Bhatia, K. G., and SenGupta, G., "Aeroelastic Model Reduction for Affordable Computational Fluid Dynamics-based Flutter Analysis," *AIAA Journal*, Vol. 43, No. 12, 2005, pp. 2487–2495. https://doi.org/10.2514/1.11246.

[31] Zhang, W., Wang, B., Ye, Z., and Quan, J., "Efficient Method for Limit Cycle Flutter Analysis Based on Nonlinear Aerodynamic Reduced-Order Models," *AIAA Journal*, Vol. 50, No. 5, 2012, pp. 1019–1028. https://doi.org/10.2514/1.J050581.

[32] Juang, J.-N., and Pappa, R. S., "An Eigensystem Realization Algorithm for Modal Parameter Identification and Model Reduction," *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 5, 1985, pp. 620–627. https://doi.org/10.2514/3.20031.

[33] Silva, W. A., "Simultaneous Excitation of Multiple-Input/Multiple-Output CFD-Based Unsteady Aerodynamic Systems," *Journal of Aircraft*, Vol. 45, No. 4, 2008, pp. 1267–1274. https://doi.org/10.2514/1.34328.

[34] Barone, M. F., Kalashnikova, I., Segalman, D. J., and Thornquist, H. K., "Stable Galerkin Reduced Order Models for Linearized Compressible Flow," *Journal of Computational Physics*, Vol. 228, No. 6, 2009, pp. 1932–1946. https://doi.org/10.1016/j.jcp.2008.11.015.

[35] Kalashnikova, I., Barone, M. F., Arunajatesan, S., and van Bloemen Waanders, B. G., "Construction of Energy-stable Projection-based Reduced Order Models," *Applied Mathematics and Computation*, Vol. 249, 2014, pp. 569–596. https://doi.org/10.1016/j.amc.2014.10.073.

[36] Hicken, J. E., and Zingg, D. W., "Parallel Newton-Krylov Solver for the Euler Equations Discretized Using Simultaneous Approximation Terms," *AIAA Journal*, Vol. 46, No. 11, 2008, pp. 2773–2786. https://doi.org/10.2514/1.34810.

[37] Hicken, J. E., Osusky, M., and Zingg, D. W., "Comparison of Parallel Preconditioners for a Newton-Krylov Flow Solver," *Computational Fluid Dynamics 2010*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 457–463. https://doi.org/10.1007/978-3-642-17884-9_57.

[38] Kreiselmaier, E., and Laschka, B., "Small Disturbance Euler Equations: Efficient and Accurate Tool for Unsteady Load Prediction," *Journal of Aircraft*, Vol. 37, No. 5, 2000, pp. 770–778. https://doi.org/10.2514/2.2699.

[39] Thomas, P., and Lombard, C., "Geometric Conservation Law and its Application to Flow Computations on Moving Grids," *AIAA Journal*, Vol. 17, No. 10, 1979, pp. 1030–1037. https://doi.org/10.2514/3.61273.

[40] Leung, T. M., and Zingg, D. W., "Aerodynamic Shape Optimization of Wings Using a Parallel Newton-Krylov Approach," *AIAA Journal*, Vol. 50, No. 3, 2012, pp. 540–550. https://doi.org/10.2514/1.J051192.

[41] Hicken, J. E., and Zingg, D. W., "A Simplified and Flexible Variant of GCROT for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific Computing*, Vol. 32, No. 3, 2010, pp. 1672–1694. https://doi.org/10.1137/090754674.

[42] Benner, P., Gugercin, S., and Willcox, K., "A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems," *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531. https://doi.org/10.1137/130932715.

[43] Gubisch, M., and Volkwein, S., "Proper Orthogonal Decomposition for Linear-Quadratic Optimal Control," *Model Reduction and Approximation: Theory and Algorithms*, SIAM, Philadelphia, 2017, Chap. 1, pp. 3–63. https://doi.org/10.1137/1.9781611974829.

[44] Holmes, P., Lumley, J. L., Berkooz, G., and Rowley, C. W., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge,UK, 1996. https://doi.org/10.1017/CBO9780511622700.

[45] Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer Science+Business Media, New York, USA, 2006.

[46] Harten, A., "On the Symmetric Form of Systems of Conservation Laws with Entropy," *Journal of Computational Physics*, Vol. 49, 1983, pp. 151–164. https://doi.org/10.1016/0021-9991(83)90118-3.

[47] Levey, A., and Lindenbaum, M., "Sequential Karhunen-Loeve Basis Extraction and its Application to Images," *IEEE Transactions on Image Processing*, Vol. 9, No. 8, 2000, pp. 1371–1374. https://doi.org/10.1109/83.855432.

[48] Choi, J., Dongarra, J. J., Pozo, R., and Walker, D. W., "ScaLAPACK: A Scalable Linear Algebra Library for Distributed Memory Concurrent Computers," *The Fourth Symposium on the Frontiers of Massively Parallel Computation*, IEEE, 1992, pp. 120–127. https://doi.org/10.1109/FMPC.1992.234898.

[49] Dowell, E. H., Peters, D. A., Clark, R., Scanlan, R., Cox, D., Simiu, E., Curtiss, H. J., Sisto, F., Edwards, J. W., and Hall, K. C., *A Modern Course in Aeroelasticity*, 4^{th} ed., Kluwer Academic Publishers, New York, USA, 2004. https://doi.org/10.1007/978-3-319-09453-3.

[50] Rendall, T., and Allen, C., "Unified Fluid–Structure Interpolation and Mesh Motion Using Radial Basis Functions," *International Journal for Numerical Methods in Engineering*, Vol. 74, No. 10, 2008, pp. 1519–1559. https://doi.org/10.1002/nme.2219.

[51] Edwards, J. W., Bennett, R. M., Whitlow, W., and Seidel, D. A., "Time-Marching Transonic Flutter Solutions Including Angle-Of-Attack Effects," *Journal of Aircraft*, Vol. 20, No. 11, 1983, pp. 899–906. https://doi.org/10.2514/3.48190.

[52] Bennett, R. M., and Desmarais, R. N., "Curve Fitting of Aeroelastic Transient Response Data with Exponential Functions," *Flutter Testing Techniques*, NASA SP-415, 1976, pp. 43–58.

[53] Isogai, K., "Numerical Study of Transonic Flutter of a Two-Dimensional Airfoil," Tech. Rep. TR-617T, National Aerospace Laboratory, July 1980.

[54] Yates Jr, E. C., "AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I.-Wing 445.6," Tech. Rep. N88-11202, National Aeronautics and Space Administration, 1987.

[55] Davis, S. S., "NACA 64A010 (NASA Ames model) oscillatory pitching," *Compendium of Unsteady Aerodynamic Measurements*, AGARD R-702, 1982, Chap. 1, pp. 2–1 – 2–22.

[56] Chyu, W., Davis, S., and Chang, K. S., "Calculation of Unsteady Transonic Flow over an Airfoil," *AIAA Journal*, Vol. 19, No. 6, 1981, pp. 684–690. https://doi.org/10.2514/3.50991.

[57] Alonso, J., and Jameson, A., "Fully-Implicit Time-Marching Aeroelastic Solutions," *32nd Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 1994. https://doi.org/10.2514/6.1994-56.

[58] Sanchez, R., Kline, H., Thomas, D., Variyar, A., Righi, M., Economon, T. D., Alonso, J. J., Palacios, R., Dimitriadis, G., and Terrapon, V., "Assessment of the Fluid-structure Interaction Capabilities for Aeronautical Applications of the Open-source Solver SU2," *Proceedings of the VII European Congress on Computational Methods in Applied Science and Engineering*, ECCOMAS, Crete Island, Greece, 2016, pp. 1498–1529. https://doi.org/10.7712/100016.1903.6597.

[59] Lee-Rausch, E., and Batina, J., "Calculation of AGARD Wing 445.6 Flutter Using Navier-Stokes Aerodynamics," *11th Applied Aerodynamics Conference*, Monterey, California, 1993. https://doi.org/10.2514/6.1993-3476.