10.2514/6.2014-2726

AIAA AVIATION Forum
16-20 June 2014, Atlanta, GA
15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference

# Towards Gradient-Based Design Optimization of Flexible Transport Aircraft with Flutter Constraints

Graeme J. Kennedy[*]
*Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, Georgia*

Gaetan K. W. Kenway[†] and Joaquim R. R. A. Martins[‡]
*University of Michigan, Department of Aerospace Engineering, Ann Arbor, Michigan*

**The next generation of civil transport aircraft will have larger wing spans and higher aspect ratios that enable greater aerodynamic efficiency and reduce fuel consumption relative to previous generations of aircraft. These larger wing spans lead to greater wing flexibility, making these designs more susceptible to adverse aeroelastic phenomena. Within the design process, dynamic aeroelastic constraints are frequently omitted from the preliminary design stage, leaving the possibility of costly late-stage design modifications to meet aeroelastic requirements. This paper addresses this issue by developing a method to incorporate flutter constraints within a gradient-based aeroelastic design optimization framework. We present the governing equations for the coupled aeroelastic system which forms the basis of a technique to evaluate the damping of the most-critical modes. We derive and implement an adjoint-based derivative evaluation technique to compute the gradient of the flutter constraint with respect to the design variables. The results demonstrate that the proposed technique is an effective approach for aeroelastic problems with large-scale structural models. The technique also shows promise for high-fidelity aerodynamic models that utilize computational fluid dynamics.**

## I. Introduction

The next generation of civil transport aircraft will have greater aerodynamic efficiency and better fuel burn performance due, in part, to larger wing spans and higher aspect ratios than previous generations of aircraft. However, larger wing spans also lead to increased wing flexibility, which presents new analysis and design challenges for these aircraft. In particular, dynamic aeroelastic effects, such as flutter, are expected to become increasingly important design drivers for the next generation of transport aircraft. Furthermore, the use of advanced lightweight materials that enable slender, aerodynamically-favorable cross-sections without violating strength requirements, will also contribute additional flexibility by decreasing the second moment of area of the wing sections. To address the challenge of designing slender wings subject to dynamic aeroelastic requirements, we present a gradient-based aeroelastic design optimization method to incorporate flutter constraints at an early stage in the design process. This gradient-based design approach is integrated with the medium-fidelity aeroelastic framework presented by Kennedy and Martins [28].

The simultaneous consideration of structures and aerodynamics is essential when analyzing and designing very flexible aircraft. The coupling of both structural and aerodynamics is required to evaluate the performance of the aircraft in the deformed flying shape. If considered at an early design stage, aircraft flexibility can be used to improve overall aircraft performance [48]. Numerous studies have demonstrated the benefits of passive aeroelastic tailoring obtained by simultaneous structural and aerodynamic design [17, 15]. However, static aeroelastic design alone is not sufficient, since dynamic aeroelastic requirements are often critical design constraints [31]. In the aircraft industry, conventional design methods focus on static strength requirements early in the design process, and only include dynamic aeroelastic analysis at a later stage [6]. This can lead to either excessively conservative designs, or costly late-stage design modifications.

Several authors, recognizing the importance of aeroelastic design, have developed integrated aeroelastic analysis and design tools to study the dynamic response of flexible aircraft subject to a range of flight conditions. These tools integrate structural and aerodynamic response, and may also include rigid-body motion and control system synthesis [11, 19, 18, 47, 26, 28, 30]. In general, these aeroelastic design methods may be grouped into two broad categories: techniques that use simplified aerodynamic and structural models, but incorporate rigid-body motion and control synthesis [11, 19, 18, 47], and techniques that employ high-fidelity aerodynamic and structural design methods, but omit

[*] Assistant Professor, AIAA Member
[†] Postdoctoral Research Fellow, AIAA Member
[‡] Associate Professor, AIAA Associate Fellow

dynamic aeroelastic effects [37, 38, 27, 30]. The advantage of the techniques in the former category, is that the simple and computationally inexpensive aeroelastic models can be used to evaluate a range of flight conditions during a design study. However, these simplified physical models restrict the prediction capability, and produce inaccurate results when the underlying analysis assumptions are violated. Design methods that use high-fidelity analysis techniques can be utilized over a broader range of conditions, but require additional computational resources and are more challenging to implement efficiently.

In this paper, we present a technique for gradient-based aircraft design optimization with flutter-constraints designed to close the gap between design methods that use simple computational models, and high-fidelity design techniques that only consider static aeroelasticity. We develop a technique to impose a flutter constraint within an analysis that uses a full finite-element structural model, coupled to a three-dimensional panel method [28, 26]. As a result, the aeroelastic coupling more closely resembles volume-based CFD methods rather than traditional doublet-lattice methods [1, 43]. The rationale for this approach is twofold: first, we want to incorporate dynamic aeroelastic constraints with a full aeroelastic aircraft design optimization framework. Second, we want to understand the requirements for integrating a full flutter analysis within a high-fidelity aeroelastic design optimization problem.

The remainder of this paper is structured as follows: in Section II we review methods for optimization with flutter constraints presented in the literature. In Section III, we briefly describe the unsteady aeroelastic formulation that we use in this study. In Section IV, we present a general approach to stability analysis that we apply to flutter based on a linearization about the deformed equilibrium flying shape of the vehicle. In Section V, we present an adjoint-based derivative evaluation technique that is used to evaluate the gradient of the flutter constraints with respect to the design variables. In Section VI, we present a flutter analysis of a transport aircraft wing and demonstrate the accuracy of our proposed gradient-evaluation technique.

## II. Review of Optimization Methods with Flutter Constraints

Many authors have developed design optimization techniques to meet flutter requirements that use simple analysis models. In one of the earliest examples, Turner [55] developed a procedure to distribute mass within a structure to meet a flutter speed constraint. Bhatia and Rudisill [7] and Rudisill and Bhatia [44] developed first and second-order derivative evaluation methods for a flutter constraint and applied these methods to structural weight minimization subject to a minimum flutter speed requirement. Later, Gwin and Taylor [16] developed a numerical optimization method to obtain the member sizes of a wing based on the method of feasible directions. These early papers utilized flutter analysis that combined simple aerodynamic and structural models due to the limited computational resources of that time. More recent examples of optimization with flutter constraints focus on refinements of these early techniques that utilize doublet-lattice methods [41, 8]. These techniques often rely on a Schur-complement reduction to the structural degrees of freedom. While these Schur-complement methods are effective for aerodynamic meshes with a small number of panels, such methods do not scale well for large aerodynamic models and cannot be used with nonlinear computational fluid dynamics (CFD).

With the increasing availability of high-performance computing resources, more authors have developed methods to perform flutter analysis using sophisticated CFD models. The first flutter predictions to use high-fidelity CFD methods utilized full time-accurate simulations [10]. However, these methods require long computational times due to the costly nature of time-accurate simulations. To address these long computational times, several authors have sought alternatives that directly simulate periodic motion. These techniques include the harmonic balance method of Hall et al. [20], Thomas et al. [53], and the nonlinear frequency domain method McMullen et al. [39], and the time-spectral method [56]. In particular, the time-spectral method was used for gradient-based design optimization [33] with derivatives obtained using an automatic-differentiation adjoint method [34]. The issue with applying these approaches in the context of flutter prediction is that the frequency of the instability is not known *a priori*. Therefore, additional iteration is required to locate the instability.

As an alternative, a number of authors have utilized Hopf-bifurcation methods to locate and track the flutter point. The advantage of the Hopf-bifurcation approach is that it can be used to locate the point of flutter instability directly, avoiding costly transient analysis. Hui and Tobak [23] demonstrated the use of a direct Hopf-bifurcation approach in a study of the dynamic stability of an aircraft at a large mean angle of attack. The analysis model consisted of simplified aerodynamic and structural models and single degree-of-freedom pitching motion dynamics. Morton and Beran [40] applied the Hopf–bifurcation technique to determine the onset of flutter for a two-dimensional airfoil in transonic flight conditions. Morton and Beran modeled the airfoil using the Euler equations and demonstrated significant computational savings compared to a full transient analysis. More recently, Badcock et al. [3] developed a Newton method to solve the Hopf-bifurcation equations. They demonstrated the improved robustness and numerical performance of their approach and calculated the flutter point for a flexible AGARD 445.6 wing in a transonic flow modeled using CFD. Using a related approach, Badcock and Woodgate [2] performed an eigenvalue-based stability analysis of a large-scale aeroelastic wing model. The wing model consisted of a full computational fluid dynamics model coupled to a finite-element model of the wing structure. The paper utilized a Schur-complement technique to decouple components of the system Jacobian to compute the stability of the system robustly in parallel computing environment. Recently, Stanford and Beran [52] used the Hopf-bifurcation approach to optimize a flexible wing subject to both flutter constraints and a post-flutter limit-cycle-oscillation constraint.

In this paper, we present an approach to the analysis of the stability an aeroelastic system at a specified flight condition. This stability analysis closely mirrors the direct Hopf-bifurcation tracking method. The difference between the two approaches lies in the exchange of a single state variable: the dynamic pressure in the Hopf-bifurcation approach for the structural damping in the stability method. However, when the flutter constraint is active, the stability approach presented here and the Hopf-bifurcation technique are essentially equivalent. The main advantage of the proposed technique is that it does not require a search for the flutter point at each design iteration. This is advantageous because the flutter point may lie far outside the operational envelope, depending on the design point. In addition, to address the issue of mode switching, we formulate a constraint using the conservative Kreisselmeier–Steinhauser (KS) [60, 42] function. We form this conservative constraint by performing KS aggregation over the modes with the least damping. This approach smooths the gradient at locations in the design space where the lowest modes switch. Evaluating this constraint imposes little additional computational cost while smoothing the behavior of the constraint.

## III. Aeroelastic Equations of Motion

In this section, we briefly describe the coupled aeroelastic equations of motion for the aircraft. We derive the equations of motion in a descriptor form where the time-derivatives of the state variables appear as arguments to the governing equations. This descriptor representation is essential due to the presence of static constraints within both the finite-element equations and the aerodynamic panel method. Within the context of the finite-element method, these static constraints are used to define the drilling degrees of freedom that do not contribute to the kinetic energy of the structure [22, 14]. An additional benefit of the descriptor representation is that it simplifies the derivation of the adjoint equations.

The fully coupled equations of motion take the following form:

$$\mathbf{R}(\mathbf{x}, \ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) = 0, \tag{1}$$

where $\mathbf{x}$ are the design variables. The state variable vector $\mathbf{q}$ contains both the aerodynamic and structural state variables, and $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ represent the first and second time derivatives of $\mathbf{q}$, respectively. Within this work, the state vector consists of both aerodynamic state variables, $\mathbf{w}$, and the structural state variables, $\mathbf{u}$, as follows: $\mathbf{q}^T = \begin{bmatrix}\mathbf{u}^T \; \mathbf{w}^T\end{bmatrix}$.

The fully coupled aeroelastic system is based on the static aerostructural optimization framework presented by Kennedy and Martins [28]. This framework couples a medium-fidelity three-dimensional parallel panel method, called TriPan, to a sophisticated parallel finite-element analysis code called the Toolkit for the Analysis of Composite Structures (TACS) [27]. In the following section, we describe the essential details of the aeroelastic framework, focusing in particular on the unsteady dynamic analysis required for flutter prediction. We note that this framework was also used for fully transient gust-encounter simulations [26].

### A. Load and displacement transfer

The load and displacement transfer technique is based on the consistent and conservative displacement extrapolation technique of Brown [9], with an adaptive refinement for smooth load transfer presented by Kennedy and Martins [28]. The displacements from the structural model are transferred to the aerodynamic surface nodes as follows:

$$\mathbf{X}_A = \mathbf{X}_A^0 + \mathbf{T}_A \mathbf{u} \tag{2}$$

where $\mathbf{X}_A^0$ and $\mathbf{X}_A$ are the initial and deformed aerodynamic surface nodal locations and $\mathbf{T}_A\mathbf{u}$ represents the extrapolation of the structural deformation to the aerodynamic surface nodes through a series of rigid-links. Based on the method of virtual work, and the displacement extrapolation scheme (2), the nodal forces are obtained through an integration over the deformed aerodynamic surface to obtain a consistent structural force vector:

$$\mathbf{F}_A = \mathbf{F}_A(\mathbf{X}_S,\, \mathbf{T}_A\dot{\mathbf{u}},\, \mathbf{X}_A + \mathbf{T}_A\mathbf{u},\, \dot{\mathbf{w}},\, \mathbf{w}) \tag{3}$$

where $\mathbf{X}_S$ are the structural nodal locations. Note that in the context of a dynamic simulation, the consistent force vector also depends on the rate of change of the structural and aerodynamic state variables.

### B. Aerodynamic panel method

The aerodynamic analysis utilizes TriPan, a three-dimensional panel method with compressibility corrections based on the Prandtl–Glauert equation [13]. TriPan uses constant source and doublet singularity elements distributed over the surface of a body discretized with quadrilateral and triangular panels that form a closed, watertight surface [24, 21]. TriPan uses a drag model consisting of the induced drag, computed using a Trefftz plane integration scheme [50], and profile and wave drag corrections computed using an empirical model [59, 25].

The discretized governing equations for the panel method can be written as follows:

$$\mathbf{R}_A(\mathbf{x}_A,\, \dot{\mathbf{w}},\, \mathbf{w},\, \mathbf{T}_A\dot{\mathbf{u}},\, \mathbf{X}_A^0(\mathbf{x}_G) + \mathbf{T}_A\mathbf{u}) = \mathbf{B}\dot{\mathbf{w}} + \mathbf{A}\mathbf{w} - \mathbf{b} = 0, \tag{4}$$

where $\mathbf{x}_A$ and $\mathbf{x}_G$ are the aerodynamic and geometric design variables, respectively, and the state variables $\mathbf{w}$ represent a vector of the surface and wake doublet strengths. In addition, $\mathbf{B}$ is a matrix that contains the wake source convection terms, $\mathbf{A}$ is a matrix that contains the dense aerodynamic-influence coefficient matrix and the wake convection terms, and $\mathbf{b}$ is a vector of boundary conditions. We solve the linear equations (4) using the parallel linear algebra routines in PETSc [4, 5]. Details of the solution method and storage scheme can be found in Kennedy and Martins [28].

### C. Structural analysis

The structural analysis is performed using TACS, a parallel finite-element code designed specifically for the design optimization of stiffened, thin-walled composite structures using either linear or geometrically nonlinear strain relationships [27].

The residuals of the structural governing equations, can be written as follows:

$$\mathbf{R}_S(\mathbf{x}_M,\, \mathbf{X}_S(\mathbf{x}_G),\, \ddot{\mathbf{u}},\, \dot{\mathbf{u}},\, \mathbf{u},\, \dot{\mathbf{w}},\, \mathbf{w}) = \mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} - \mathbf{F}_A(\mathbf{X}_S,\, \mathbf{T}_A\dot{\mathbf{u}},\, \mathbf{X}_A + \mathbf{T}_A\mathbf{u},\, \dot{\mathbf{w}},\, \mathbf{w}) = 0 \tag{5}$$

where $\mathbf{u}$ is the finite-element state vector, $\mathbf{K}$ is the linear stiffness matrix, and $\mathbf{M}$ is the mass matrix. The structural design variables are split into two groups: material design variables, denoted $\mathbf{x}_M$, and geometric design variables, denoted $\mathbf{x}_G$. Note that the geometric variables are shared between the aerodynamics and the structures.

### D. Coupled aeroelastic system

The aeroelastic governing equations are the fully coupled system obtained by combining the aerodynamic governing equations (4), with the structural governing equations (5). This forms the following coupled system in a descriptor residual form:

$$\mathbf{R}(\mathbf{x},\, \ddot{\mathbf{q}},\, \dot{\mathbf{q}},\, \mathbf{q}) = \begin{bmatrix} \mathbf{R}_A(\mathbf{x}_A,\, \dot{\mathbf{w}},\, \mathbf{w},\, \mathbf{T}_A\dot{\mathbf{u}},\, \mathbf{X}_A^0(\mathbf{x}_G) + \mathbf{T}_A\mathbf{u}) \\ \mathbf{R}_S(\mathbf{x}_M,\, \mathbf{X}_S(\mathbf{x}_G),\, \ddot{\mathbf{u}},\, \dot{\mathbf{u}},\, \mathbf{u},\, \dot{\mathbf{w}},\, \mathbf{w}) \end{bmatrix} = 0 \tag{6}$$

Note that we combine aerodynamic, geometric and material design variables into a single design variable vector $\mathbf{x} = \begin{bmatrix} \mathbf{x}_A^T & \mathbf{x}_G^T & \mathbf{x}_M^T \end{bmatrix}^T$.

## IV. Flutter and Stability Analysis

In this work we use a stability analysis about an equilibrium point to impose an effective constraint on flutter. The derivation of this technique is similar to the direct Hopf-bifurcation analysis, and we note the differences between the methods where appropriate. The stability of an equilibrium point can be determined based on the onset of an infinitesimal periodic motion about a steady-state solution. This motion can be expressed as follows:

$$\mathbf{q}(t) = \mathbf{q}_0 + \epsilon\, \mathbf{q}_t e^{\lambda t} \tag{7}$$

where $\mathbf{q}_0 \in \mathbb{R}^n$ is the steady-state solution, $\epsilon$ is the infinitesimal amplitude of the motion, $\lambda \in \mathbb{C}$, and $\mathbf{q}_t \in \mathbb{C}^n$ are the periodic components of the motion. To fix the magnitude of $\mathbf{q}_t$, we impose the condition:

$$(\mathbf{p},\, \mathbf{q}_t) = 1, \tag{8}$$

where $\mathbf{p} \in \mathbb{R}^n$ is an arbitrary vector.

To determine the stability of the descriptor system (1), we take a Taylor series expansion of the governing residuals about the steady state $\mathbf{q}_0$. Based on the equations of motion, this Taylor series expansion can be expressed as follows:

$$\mathbf{R}(\mathbf{x},\, \ddot{\mathbf{q}},\, \dot{\mathbf{q}},\, \mathbf{q}) = \mathbf{R}_0 + \epsilon \left( \frac{\partial \mathbf{R}}{\partial \mathbf{q}} + \lambda \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{q}}} + \lambda^2 \frac{\partial \mathbf{R}}{\partial \ddot{\mathbf{q}}} \right) \mathbf{q}_t e^{\lambda t} + \mathcal{O}(\epsilon^2), \tag{9}$$

where $\mathbf{R}_0 = \mathbf{R}(\mathbf{x},\, 0,\, 0,\, \mathbf{q}_0)$. To obtain a non-trivial infinitesimal motion, both terms on the right-hand-side must vanish independently. Furthermore, to reduce the equations to real variables for practical computations, we split the solution $\mathbf{q}_t \in \mathbb{C}^n$ into real and imaginary parts as follows:

$$\begin{aligned} \mathbf{q}_t &= \mathbf{q}_r + j\mathbf{q}_c, \\ \lambda &= \zeta + j\omega, \end{aligned}$$

where $\mathbf{q}_r$, $\mathbf{q}_c \in \mathbb{R}^n$ are the real and complex components of $\mathbf{q}_t$, respectively and $\zeta, \omega \in \mathbb{R}$ are the real and complex components of $\lambda$, respectively. Substituting this expression into Equation (9) and Equation (8) and taking the real and imaginary parts yields the following system of equations:

$$\mathbf{R}_S(\mathbf{x},\, \mathbf{Q}) = \begin{bmatrix} \mathbf{R}_0 \\ \left(\mathbf{J} + \zeta\mathbf{J}_{\dot{q}} + (\zeta^2 - \omega^2)\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_c + \left(\omega\mathbf{J}_{\dot{q}} + 2\zeta\omega\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_r \\ \left(\mathbf{J} + \zeta\mathbf{J}_{\dot{q}} + (\zeta^2 - \omega^2)\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_r - \left(\omega\mathbf{J}_{\dot{q}} + 2\zeta\omega\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_c \\ \mathbf{p}^T\mathbf{q}_r - 1 \\ \mathbf{p}^T\mathbf{q}_c \end{bmatrix} = 0, \qquad (10)$$

where we have used the following notation for the Jacobians to simplify the presentation:

$$\mathbf{J} = \frac{\partial \mathbf{R}}{\partial \mathbf{q}}, \qquad \mathbf{J}_{\dot{q}} = \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{q}}}, \qquad \mathbf{J}_{\ddot{q}} = \frac{\partial \mathbf{R}}{\partial \ddot{\mathbf{q}}}.$$

Note that the last two equations in the residuals (10), represent a constraint on the magnitude of the vectors. The vector $\mathbf{Q} \in \mathbb{R}^{3n+2}$ are the state variables for the equilibrium solution as well as the real and imaginary parts of the solution, $\mathbf{q}_t$, and the frequency and damping of the motion such that:

$$\mathbf{Q}^T = \begin{bmatrix} \mathbf{q}_0^T & \mathbf{q}_c^T & \mathbf{q}_r^T & \omega & \zeta \end{bmatrix}.$$

Note that the first $n$ equations are decoupled from the remaining $2n + 2$ equations since the residuals $\mathbf{R}_0$ only depend on $\mathbf{q}_0$. Therefore, when solving the system of equations, we first solve $\mathbf{R}_0 = 0$, and then solve the remaining $2n + 2$ equations.

The Hopf-bifurcation equations can be obtained from the stability equations (10) by setting the damping parameter $\zeta = 0$ and adding the dynamic pressure $q_\infty$ as an unknown. This yields the following system of equations:

$$\mathbf{R}_H(\mathbf{x},\, \mathbf{Q}_H) = \begin{bmatrix} \mathbf{R}_0 \\ \left(\mathbf{J} - \omega^2\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_c + \omega\mathbf{J}_{\dot{q}}\mathbf{q}_r \\ \left(\mathbf{J} - \omega^2\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_r - \omega\mathbf{J}_{\dot{q}}\mathbf{q}_c \\ \mathbf{p}^T\mathbf{q}_r - 1 \\ \mathbf{p}^T\mathbf{q}_c \end{bmatrix} = 0, \qquad (11)$$

The Hopf-bifurcation equations are again a system of $3n+2$ equations and $3n+2$ unknowns. The vector $\mathbf{Q}_H \in \mathbb{R}^{3n+2}$ is given as follows:

$$\mathbf{Q}_H^T = \begin{bmatrix} \mathbf{q}_0^T & \mathbf{q}_c^T & \mathbf{q}_r^T & \omega & q_\infty \end{bmatrix}.$$

Note in this case, however, the governing equilibrium equations are now coupled to the last $2n + 2$ equations through the dynamic pressure $q_\infty$. This a result of the dynamic pressure modifying both the equilibrium equations and the Jacobian of the final $2n + 2$ equations.

### A. A Jacobi–Davidson method for stability analysis

We solve Equation (10) governing the stability of the nonlinear aeroelastic equations using a two-stage procedure. First, we solve the governing equations $\mathbf{R}_0(\mathbf{q}_0) = \mathbf{R}(\mathbf{x}, 0, 0, \mathbf{q}_0)$, using the approximate Newton–Krylov scheme described in Kennedy and Martins [28]. The solution of these equations is decoupled from the solution of the remaining $2n + 2$ equations. After we have obtained an equilibrium solution, $\mathbf{q}_0$, we solve the eigenvalue problem for the eigenvector $\mathbf{q}_t \in \mathbb{C}^n$ and eigenvalue $\lambda \in \mathbb{C}$ using a Jacobi–Davidson method for generalized eigenvalue problems [49, 54].

The Jacobi-Davidson method can be formulated as a Newton-like method on the combined eigenpair $(\mathbf{q}_t,\, \lambda)$. After the $m^{\text{th}}$ iteration, the estimate of the eigenpair is $(\mathbf{u}_m, \theta_m)$. The generalized eigenvalue problem can be represented by the following nonlinear system of equations:

$$\begin{aligned} \left(\mathbf{J} + \theta_m \mathbf{J}_{\dot{q}} + \theta_m^2 \mathbf{J}_{\ddot{q}}\right) \mathbf{u}_m &= 0, \\ \mathbf{u}_m^H \mathbf{u}_m &= 1, \end{aligned}$$

where $\cdot^H$ denotes the complex-conjugate transpose of a vector or matrix. A Newton-like update for this nonlinear system takes the following form:

$$\begin{bmatrix} \mathbf{J} + \theta_m \mathbf{J}_{\dot{q}} + \theta_m^2 \mathbf{J}_{\ddot{q}} & \mathbf{w} \\ \mathbf{u}_m^H & 0 \end{bmatrix} \begin{bmatrix} \mathbf{t} \\ \mu \end{bmatrix} = \begin{bmatrix} -\mathbf{r} \\ 0 \end{bmatrix}, \tag{12}$$

where $\mathbf{t}$ and $\mu$ are the Newton updates for the eigenvector and eigenvalue Note that the residual and derivative vectors $\mathbf{r}$, $\mathbf{w} \in \mathbb{C}^n$ in Equation (12) are defined as follows:

$$\begin{aligned} \mathbf{r} &= (\mathbf{J} + \theta_m \mathbf{J}_{\dot{q}} + \lambda^2 \mathbf{J}_{\ddot{q}})\mathbf{u}_m, \\ \mathbf{w} &= (\mathbf{J}_{\dot{q}} + 2\theta_m \mathbf{J}_{\ddot{q}})\mathbf{u}_m. \end{aligned}$$

As $(\mathbf{u}_m,\, \theta_m) \rightarrow (\mathbf{q}_t,\, \lambda)$, the matrix $\mathbf{J} + \theta_m \mathbf{J}_{\dot{q}} + \theta_m^2 \mathbf{J}_{\ddot{q}}$ becomes more poorly conditioned and approaches singularity. The Jacobi-Davidson method overcomes this difficulty by removing the singularity through a projection onto the space perpendicular to the eigenvector estimate, $\mathbf{u}_m$. This projection technique yields the following linear system which ensures that $\mathbf{t} \in \{\mathbf{u}_m\}^{\perp}$:

$$(\mathbf{I} - \tilde{\mathbf{w}}\mathbf{u}_m^H)(\mathbf{J} + \theta \mathbf{J}_{\dot{q}} + \theta^2 \mathbf{J}_{\ddot{q}})(\mathbf{I} - \mathbf{u}_m \mathbf{u}_m^H)\mathbf{t} = -\mathbf{r} \tag{13}$$

where $\tilde{\mathbf{w}} = \mathbf{w}/\mathbf{u}_m^H \mathbf{w}$. We solve this equation inexactly using FGMRES(5) [46], where we utilize the aerostructural preconditioner described in Kennedy and Martins [28]. We form this preconditioner at the initial iteration using only the terms in $\mathbf{J}$.

Instead of applying the update $\mathbf{t}$ directly to $\mathbf{u}_m$, we maintain an orthonormal basis $\mathbf{V}_m \in \mathbb{C}^{n \times m}$, such that $\mathbf{V}_m^H \mathbf{V}_m = \mathbf{I}_m$. After $\mathbf{t}$ is computed approximately, we expand the orthonormal subspace by performing a modified Gram–Schmidt orthogonalization, such that $\mathbf{V}_{m+1} = \text{MGS}(\mathbf{V}_m, \mathbf{t})$ [45]. We then compute a new estimate of $\theta_m$ by computing a solution to the reduced eigenproblem:

$$\mathbf{V}_{m+1}^H (\mathbf{J} + \theta \mathbf{J}_{\dot{q}} + \theta^2 \mathbf{J}_{\ddot{q}}) \mathbf{V}_{m+1} \mathbf{y}_{m+1} = 0 \tag{14}$$

where select $\theta_{m+1}$ as the eigenvalue of the reduced $(m+1) \times (m+1)$ eigenproblem that is closest to the desired spectrum. In this case, we use the eigenvalue with the smallest frequency that has positive real part, or if no eigenvalues have positive real parts, then the eigenvalue with the smallest imaginary part. We then update the eigenvector estimate using $\mathbf{u}_{m+1} = \mathbf{V}_{m+1}\mathbf{y}_{m+1}$ and repeat until the residual $\mathbf{r}$ has been reduced by a relative tolerance of $10^{-8}$.

We note here that in the implementation of the Jacobi–Davidson method, we separate the real and complex components of the vectors. Therefore, despite the fact that the method utilizes complex arithmetic, the storage is entirely in real numbers. This scheme facilitates the use of the complex-step derivative approximation for gradient-verification [51, 36].

### B. An approximate Newton–Krylov method the Hopf-bifurcation equations

We use a three-stage solution procedure to solve the Hopf-bifurcation equations (11). The first two stages represent a start-up strategy to obtain a good initial estimate of the equilibrium configuration, and the lowest natural frequency of motion at the initial dynamic pressure. Based on this initial estimate, we then enter an approximate Newton–Krylov phase in which an approximate linearized system is solved at each iteration to a loose tolerance until convergence is achieved.

The start-up method proceeds as follows: First, given an initial free-stream dynamic pressure $q_\infty$, we solve the static governing equations to a loose tolerance to obtain the approximate equilibrium configuration $\mathbf{q}_0$. The details of this solution procedure are outlined in Kennedy and Martins [28]. Next, given $q_\infty$ and $\mathbf{q}_0$, we compute initial estimates for $\mathbf{q}_c$, $\mathbf{q}_r \in \mathbb{R}^n$ and $\omega \in \mathbb{R}$ by finding an approximate solution to the following generalized eigenvalue problem:

$$\begin{bmatrix} \mathbf{J}\mathbf{q}_c - \omega^2 \mathbf{J}_{\ddot{q}} \mathbf{q}_c + \omega \mathbf{J}_{\dot{q}} \mathbf{q}_r \\ \mathbf{J}\mathbf{q}_r - \omega^2 \mathbf{J}_{\ddot{q}} \mathbf{q}_r - \omega \mathbf{J}_{\dot{q}} \mathbf{q}_c \end{bmatrix} = 0,$$

using the Jacobi–Davidson method described above. Since $\omega$ is real, these equations cannot be satisfied exactly at an arbitrary point.

In the last stage of the solution procedure, we use an approximate Newton–Krylov method to rapidly solve the nonlinear coupled system of equations (11). At each iteration of the approximate Newton method, we solve the following linear system:

$$\mathbf{J}_H(\mathbf{Q}^n)\Delta\mathbf{Q}^n = -\mathbf{R}_H(\mathbf{Q}^n), \tag{15}$$

for the update $\Delta\mathbf{Q}^n$. Since the linearization is approximate, we do not solve the equations to a tight tolerance. After we obtain the update, we compute the next iterate as follows $\mathbf{Q}^{n+1} = \mathbf{Q}^n + \Delta\mathbf{Q}^n$.

The matrix $\mathbf{J}_H$ is a computationally convenient approximation of the true Jacobian of the Hopf-bifurcation equations (11). Within this work, we utilize the following approximation:

$$\mathbf{J}_H = \begin{bmatrix} \mathbf{J} & 0 & 0 & 0 & \mathbf{R}_{,q_\infty} \\ * & \mathbf{J} - \omega^2\mathbf{J}_{\ddot{q}} & \omega\mathbf{J}_{\dot{q}} & \mathbf{J}_{\dot{q}}\mathbf{q}_r - 2\omega\mathbf{J}_{\ddot{q}}\mathbf{q}_c & * \\ * & -\omega\mathbf{J}_{\dot{q}} & \mathbf{J} - \omega^2\mathbf{J}_{\ddot{q}} & -\mathbf{J}_{\dot{q}}\mathbf{q}_c - 2\omega\mathbf{J}_{\ddot{q}}\mathbf{q}_r & * \\ 0 & \mathbf{p}^T & 0 & 0 & 0 \\ 0 & 0 & \mathbf{p}^T & 0 & 0 \end{bmatrix}, \tag{16}$$

where each $*$ indicates the location of a non-zero term which has been neglected.

One of the main difficulties in solving the approximate Newton system (11), is the constraints in the final two rows. These constraints represent a separation of the real and complex components of the periodic solution that must be satisfied precisely. As in the Jacobi–Davidson method, we use a projection technique to ensure that these equations are satisfied exactly.

## V. Adjoint-Based Derivative Methods for Flutter

In this section, we formulate a flutter constraint based on imposing a minimum specified damping in the most-critical dynamic mode. We then derive an adjoint-based derivative evaluation technique for this constraint based on the stability equations (10). Note that while the linear combination of the Jacobians at the eigenvalue is a singular matrix, the presence of the normalization constraint, $(\mathbf{p}, \mathbf{q}_t) = 1$, removes this singularity as long $\mathbf{p}$ is not parallel to $\mathbf{q}_t$. Since the linearization of the stability equations (10) is nonsingular at the solution, the requirements of the implicit function theorem are satisfied and there is no theoretical issue with the application of the adjoint method.

While more sophisticated flutter constraints are possible [52], in this work we constrain the damping in the most-critical mode to be below some specified value:

$$f(\mathbf{x}, \mathbf{Q}) = -\beta - \zeta \geq 0, \tag{17}$$

where $\beta > 0$ is a minimum damping parameter, and this constraint ensures $\zeta \leq -\beta$. This constraint is imposed at a specified flight condition, i.e. $q_\infty$, and imposes a constraint on the damping that ensures that, within the assumptions of the analysis, the aircraft will not flutter.

The adjoint method for the nonlinear system (10) can be derived using a conventional approach [35, 38], by introducing a vector of adjoint variables $\boldsymbol{\psi}$ that satisfies the following linearized system:

$$\left[\frac{\partial \mathbf{R}_S}{\partial \mathbf{Q}}\right]^T \boldsymbol{\psi} = -\frac{\partial f}{\partial \mathbf{Q}}^T, \tag{18}$$

where $f(\mathbf{x}, \mathbf{Q})$ is a function of interest that depends on the design variables, $\mathbf{x}$, and the stability variables, $\mathbf{Q}$. Note that, in this case, no simplifications of the Jacobian $\partial\mathbf{R}_S/\partial\mathbf{Q}$ can be made without sacrificing the accuracy of the derivatives. Once the adjoint variables $\boldsymbol{\psi}$ are evaluated, the total derivative can be computed as follows:

$$\nabla_{\mathbf{x}} f = \frac{\partial f}{\partial \mathbf{x}} + \boldsymbol{\psi}^T \frac{\partial \mathbf{R}_S}{\partial \mathbf{x}}, \tag{19}$$

where $\nabla_{\mathbf{x}} f$ is the total derivative of constraint $f(\mathbf{x}, \mathbf{Q})$.

A number of simplifications of the adjoint method are possible due to the simplicity of the constraint (17). The first simplification is that the derivative of $f(\mathbf{x}, \mathbf{Q})$ with respect to the state variables has only a single entry at the index corresponding to the damping parameter. Second, since the constraint (17) does not have any direct design-variable dependence, i.e. the partial derivative with respect to the design variables is zero. These observations yield the following simplification of the adjoint equation:

$$\left[\frac{\partial \mathbf{R}_S}{\partial \mathbf{Q}}\right]^T \boldsymbol{\psi} = \begin{bmatrix}0 & 0 & 0 & 0 & 1\end{bmatrix}^T \tag{20}$$

and the following expression for the total derivative:

$$\nabla_{\mathbf{x}} f = \boldsymbol{\psi}^T \frac{\partial \mathbf{R}_S}{\partial \mathbf{x}}. \tag{21}$$

These simplifications reduce the computational cost of computing the derivative of the flutter constraint.

## A. Solving the adjoint equation

The primary challenge in computing the derivative of the flutter constraint, is accurately solving the adjoint equations (20). This is challenging due to the coupled nature of the initial equilibrium analysis and eigenvalue computation. Unfortunately, some of the required in the adjoint equations involve derivatives of matrix-vector products due to the eigenvalue problem embedded in the stability equations (20). While these terms can be computed exactly using either hand-coded routines, or automatic differentiation techniques [34], here we use finite-difference matrix-vector product techniques. We delay a complete discussion of these terms until Section B.

The adjoint equation (20), with the exact Jacobian transpose, can be written in the following form:

$$\begin{bmatrix} \mathbf{J}^T & \mathbf{J}_{21}^T & \mathbf{J}_{31}^T & 0 & 0 \\ 0 & \mathbf{J}_{22}^T & -\mathbf{J}_{23}^T & \mathbf{p} & 0 \\ 0 & \mathbf{J}_{23}^T & \mathbf{J}_{22}^T & 0 & \mathbf{p} \\ 0 & \mathbf{J}_{24}^T & \mathbf{J}_{34}^T & 0 & 0 \\ 0 & \mathbf{J}_{25}^T & \mathbf{J}_{35}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\psi}_0 \\ \boldsymbol{\psi}_c \\ \boldsymbol{\psi}_r \\ \psi_\omega \\ \psi_\zeta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{22}$$

where the Jacobian terms are defined as follows:

$$\begin{aligned} \mathbf{J}_{21} &= \frac{\partial \mathbf{R}_c}{\partial \mathbf{q}_0}, & \mathbf{J}_{31} &= \frac{\partial \mathbf{R}_r}{\partial \mathbf{q}_0}, \\ \mathbf{J}_{22} &= \mathbf{J} + \zeta \mathbf{J}_{\dot{q}} + (\zeta^2 - \omega^2)\mathbf{J}_{\ddot{q}}, & \mathbf{J}_{23} &= \omega \mathbf{J}_{\dot{q}} + 2\zeta\omega \mathbf{J}_{\ddot{q}}, \\ \mathbf{J}_{24} &= \mathbf{J}_{\dot{q}}\mathbf{q}_r + 2\zeta \mathbf{J}_{\ddot{q}}\mathbf{q}_r - 2\omega \mathbf{J}_{\ddot{q}}\mathbf{q}_c, & \mathbf{J}_{34} &= -\mathbf{J}_{\dot{q}}\mathbf{q}_c + 2\zeta \mathbf{J}_{\ddot{q}}\mathbf{q}_c - 2\omega \mathbf{J}_{\ddot{q}}\mathbf{q}_r, \\ \mathbf{J}_{25} &= \mathbf{J}_{\dot{q}}\mathbf{q}_c + 2\zeta \mathbf{J}_{\ddot{q}}\mathbf{q}_c + 2\omega \mathbf{J}_{\ddot{q}}\mathbf{q}_c, & \mathbf{J}_{35} &= \mathbf{J}_{\dot{q}}\mathbf{q}_r + 2\zeta \mathbf{J}_{\ddot{q}}\mathbf{q}_r - 2\omega \mathbf{J}_{\ddot{q}}\mathbf{q}_c. \end{aligned}$$

Here, we have defined $\mathbf{R}_c(\mathbf{x}, \mathbf{q}_0, \mathbf{q}_c, \mathbf{q}_r)$ and $\mathbf{R}_r(\mathbf{x}, \mathbf{q}_0, \mathbf{q}_c, \mathbf{q}_r)$ as follows:

$$\begin{aligned} \mathbf{R}_c(\mathbf{x}, \mathbf{q}_0, \mathbf{q}_c, \mathbf{q}_r) &= \left(\mathbf{J} + \zeta \mathbf{J}_{\dot{q}} + (\zeta^2 - \omega^2)\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_c + \left(\omega \mathbf{J}_{\dot{q}} + 2\zeta\omega \mathbf{J}_{\ddot{q}}\right)\mathbf{q}_r, \\ \mathbf{R}_r(\mathbf{x}, \mathbf{q}_0, \mathbf{q}_c, \mathbf{q}_r) &= \left(\mathbf{J} + \zeta \mathbf{J}_{\dot{q}} + (\zeta^2 - \omega^2)\mathbf{J}_{\ddot{q}}\right)\mathbf{q}_r - \left(\omega \mathbf{J}_{\dot{q}} + 2\zeta\omega \mathbf{J}_{\ddot{q}}\right)\mathbf{q}_c. \end{aligned}$$

These expressions can be simplified by considering the complex-conjugate transpose of the eigenproblem (12), which yields the following system of equations, that is equivalent to the adjoint equation (22):

$$\begin{bmatrix} \mathbf{J}^H + \bar{\lambda}\mathbf{J}_{\dot{q}}^H + \bar{\lambda}^2 \mathbf{J}_{\ddot{q}}^H & \mathbf{u} \\ \mathbf{w}^H & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\psi}} \\ \psi_\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{23}$$

where $\mathbf{u}$ is the approximate eigenvector from the Jacobi–Davidson method and $\tilde{\boldsymbol{\psi}} = \boldsymbol{\psi}_r + j\boldsymbol{\psi}_c \in \mathbb{C}^n$.

Note that premultiplying the first row of Equation (23), by $\mathbf{u}^H$ yields the result $\psi_\lambda = 0$, since the eigenvector $\mathbf{u}$ satisfies $(\mathbf{J} + \mathbf{J}_{\dot{q}} + \lambda \mathbf{J}_{\ddot{q}})\mathbf{u} = 0$. Therefore, the solution to this system of equations is $\tilde{\psi} = \alpha \mathbf{v}$, where $\mathbf{v}$ is a left-eigenvector:

$$\mathbf{v}^H(\mathbf{J} + \lambda \mathbf{J}_{\dot{q}} + \lambda^2 \mathbf{J}_{\ddot{q}}) = 0,$$

and the scalar $\alpha$ is determined as follows:

$$\alpha = \frac{1}{\mathbf{w}^H \mathbf{v}}.$$

After the real and complex components of the scaled left-eigenvector $\tilde{\boldsymbol{\psi}} = \alpha \mathbf{v}$ have been computed, we compute the adjoint variables for the static equilibrium equations as follows:

$$\mathbf{J}^T \boldsymbol{\psi}_0 = -\mathbf{J}_{21}^T \boldsymbol{\psi}_c - \mathbf{J}_{31}^T \boldsymbol{\psi}_r, \tag{24}$$

where the two terms on the right-hand-side require the derivative of a transpose matrix-vector product with the adjoint vectors $\boldsymbol{\psi}_c$ and $\boldsymbol{\psi}_r$, respectively.

Once the full adjoint vector $\boldsymbol{\psi}$ has been evaluated, we compute the total derivative of the flutter equations as follows:

$$\nabla_{\mathbf{x}} f = \boldsymbol{\psi}_0^T \frac{\partial \mathbf{R}_0}{\partial \mathbf{x}} + \boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}_c}{\partial \mathbf{x}} + \boldsymbol{\psi}_s^T \frac{\partial \mathbf{R}_s}{\partial \mathbf{x}}. \tag{25}$$

Note that the last two terms in this expression involve design variable derivatives of matrix-vector products. This completes the computation of the derivatives required for the stability constraint (17).

### B. Evaluating the derivative of matrix-vector products

Two groups of terms in the adjoint method described above require the derivative of an expression that is itself a matrix-vector product. In principle it is possible to evaluate these terms exactly through either hand-coded derivative routines or automatic differentiation methods [34]. However, these expressions do not appear in the adjoint method for static or even dynamic analysis, and they are therefore unlikely to be implemented within existing code for design optimization.

To circumvent this issue, in this work we approximate these terms using selective finite-difference matrix-vector products. This technique compromises the accuracy of constraint derivatives, since finite-difference methods are susceptible to subtractive cancellation. However, the approach can be used in combination with existing design optimization routines that are required for adjoint-based gradient evaluation. This reduces the difficulty of implementing these necessary terms.

Note that the adjoint variables for the eigenvalue problem, $\boldsymbol{\psi}_c$, and $\boldsymbol{\psi}_r$, can be determined without approximation. The difficulty arises when using Equation (24) to obtain the adjoint variables for the static equilibrium equations, $\boldsymbol{\psi}_0$. We derive an approximation for the right-hand-side of Equation (24) by first computing an approximate finite-difference matrix-vector product for the residual $\mathbf{R}_c$ as follows:

$$\begin{aligned}\mathbf{R}_c \approx \mathbf{r} =& \frac{1}{\epsilon}\left[\mathbf{R}\left(\mathbf{x},\, \epsilon(\zeta^2 - \omega^2)\mathbf{q}_c,\, \epsilon\zeta\mathbf{q}_c,\, \epsilon\mathbf{q}_c + \mathbf{q}_0\right) - \mathbf{R}(\mathbf{x},\, 0,\, 0,\, \mathbf{q}_0)\right] + \\ & \frac{1}{\epsilon}\left[\mathbf{R}\left(\mathbf{x},\, 2\epsilon\omega\zeta\mathbf{q}_r,\, \epsilon\omega\mathbf{q}_r,\, \mathbf{q}_0\right) - \mathbf{R}(\mathbf{x},\, 0,\, 0,\, \mathbf{q}_0)\right].\end{aligned}$$

Note that the terms in the first and second time-derivatives form approximate finite-difference matrix-vector products. Based on this approximate $\mathbf{r} \approx \mathbf{R}_c$, we can compute:

$$\mathbf{J}_{21}^T \boldsymbol{\psi}_c = \frac{\partial \mathbf{R}_c}{\partial \mathbf{q}_0}^T \boldsymbol{\psi}_c = \frac{\partial \mathbf{r}}{\partial \mathbf{q}_0}^T \boldsymbol{\psi}_c + \mathcal{O}(\epsilon),$$

where $\partial \mathbf{r}/\partial \mathbf{q}_0$ only involves computing the Jacobian with respect to the static state variables.

Many design optimization codes implement a capability to compute the product of an adjoint vector with the derivative of the residuals with respect to the design variables. This code can be used to compute the terms required for the evaluation of the total derivative (25) as follows:

$$\begin{aligned}\boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}_c}{\partial \mathbf{x}} =& \frac{1}{\epsilon}\left[\boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}\left(\mathbf{x},\, \epsilon(\zeta^2 - \omega^2)\mathbf{q}_c,\, \epsilon\zeta\mathbf{q}_c,\, \epsilon\mathbf{q}_c + \mathbf{q}_0\right) - \boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}(\mathbf{x},\, 0,\, 0,\, \mathbf{q}_0)\right] + \\ & \frac{1}{\epsilon}\left[\boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}\left(\mathbf{x},\, 2\epsilon\omega\zeta\mathbf{q}_r,\, \epsilon\omega\mathbf{q}_r,\, \mathbf{q}_0\right) - \boldsymbol{\psi}_c^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}}(\mathbf{x},\, 0,\, 0,\, \mathbf{q}_0)\right] + \mathcal{O}(\epsilon).\end{aligned}$$

Note that the time-derivatives of the state variables have been perturbed to produce an effective matrix-vector product with the different matrix components.

## VI. Results

In this section, we present results from the stability analysis technique presented above. While these results are preliminary, they demonstrate the potential of the methods described above.

Within this work, we use the NASA Common Research Model (CRM) wing as our aerostructural test problem. The CRM model is widely used for aerodynamic verification and validation and was originally developed for the AIAA Drag Prediction workshop [57, 58, 32]. However, since the CRM is intended primarily for aerodynamic analysis, the original CRM geometry is deformed in the 1 g cruise condition. In this work, we use a modified form of the CRM geometry where the jig shape is inferred based on fitting algorithm. Details of this procedure are outlined in Kenway et al. [29].

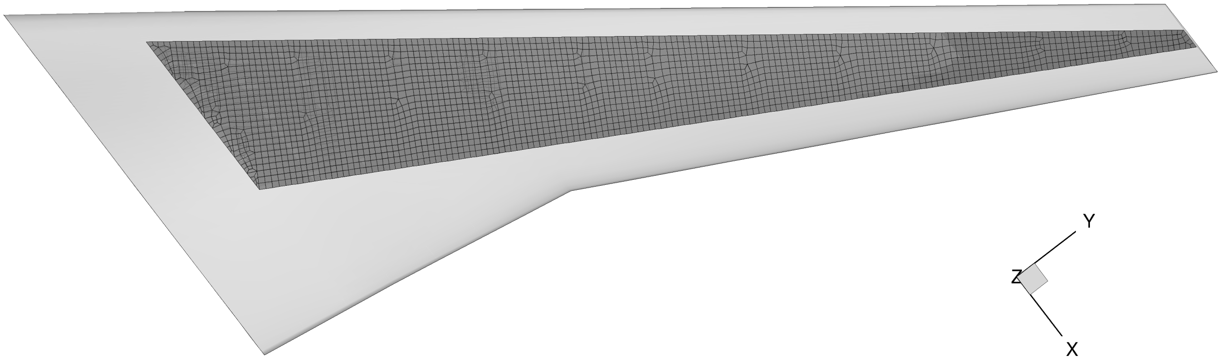

Figure 1: The Common Research Model (CRM) wing geometry and structural mesh.

In this case, we use a finite-element model of the CRM geometry consisting of 10 127 MITC4 shell elements [12], with 9342 nodes and just over 56 000 degrees of freedom. The three-dimensional aerodynamic mesh consists of 800 panels distributed over the wing surface. Figure 1 shows the finite-element mesh and the aerodynamic surface. Note that the finite-element structural model ends at the wing-body intersection, while the aerodynamic surface extends to the symmetry plane.

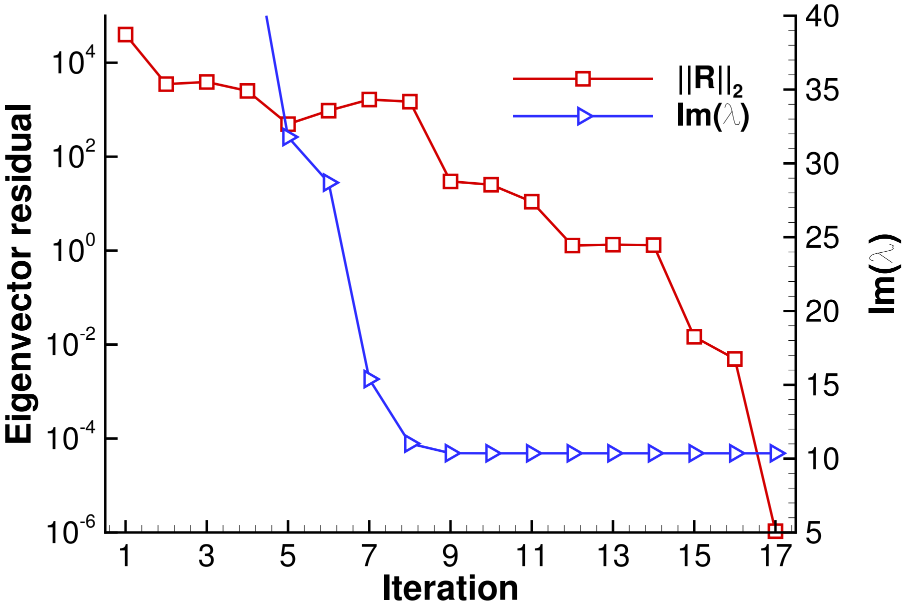

Figure 2: Convergence history of the Jacobi-Davidson method on the Common Research Model test case.

Figure 2 shows the convergence history of the Jacobi–Davidson iteration on the CRM test case described above. The method converges to a tolerance of $10^{-10}$ within 17 iterations. This illustrates the typical characteristic of the Jacobi–Davidson method, which exhibits second-order convergence close to the solution of the eigenvalue problem [49]. Figure 3 shows the least-damped mode for the CRM wing found using the Jacobi–Davidson technique.

To illustrate the potential of the gradient evaluation method presented above, we compute the derivative of the

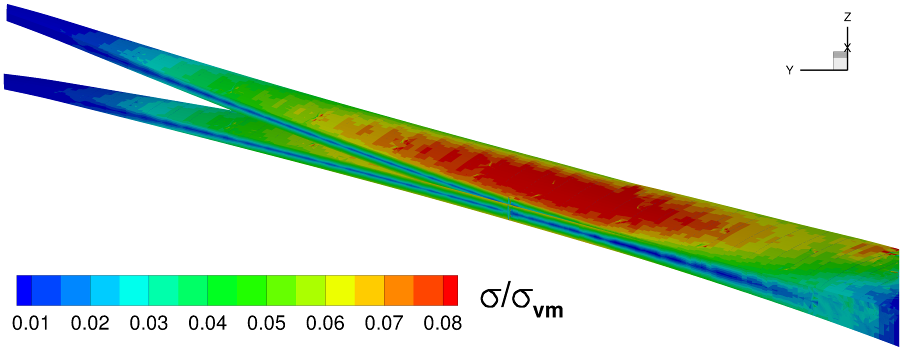

Figure 3: Illustration of the least-damped mode for the CRM geometry at a fixed flight condition.

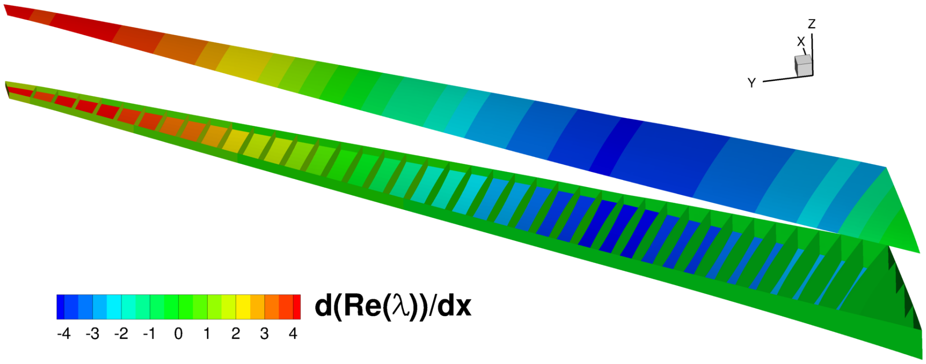

Figure 4: Illustration of the derivative of the real component of the eigenvalue with respect to the thickness variables.

real part of the least-damped eigenvalue with respect to structural thickness variables. We add thickness variables for each structural component formed by the intersection of the wing spars and ribs, as well as each rib and spar segment, resulting in a total of 266 structural thickness variables in the finite-element model. Figure 4 shows the derivative of the real part of the eigenvalue with respect to the structural thickness variables. Note that this derivative is for the same mode illustrated in Figure 3. The derivative is close to zero in the spars and ribs, and takes its highest and lowest values in the wing skins. This is unsurprising given the bending-dominated nature of the eigenmode. Furthermore, we note that a step along the positive gradient directly will increase the real part of the eigenvalue, leading to a possible instability. Therefore, to reduce susceptibility to flutter, it would be necessary to thicken the wing structure in areas where there is a negative gradient. In this case, material would be removed from the wing tip skins and placed near the wing crank region.

## VII. Conclusions

As wing aspect ratios and spans increase to meet fuel efficiency demands, it will become increasingly important to consider wing flexibility at an early stage in the design process. Current design practice is to omit dynamic aeroelastic requirements until later stages in the design process, leaving the possibility that dynamic aeroelastic constraints may be design-critical. This paper addresses this issue by developing a method to incorporate flutter constraints within a gradient-based aeroelastic design optimization framework. We presented a general approach to impose a constraint based on the damping in the flutter modes. This method does not rely on a reduction to the aerodynamic degrees of freedom and is compatible with high-fidelity CFD techniques. Next, we presented an adjoint-based derivative evaluation technique to compute the gradient of the proposed stability constraint with respect to the design variables. The results demonstrate that the proposed technique is an effective approach for aeroelastic problems with large-scale structural models. The technique also shows promise for high-fidelity aerodynamic models that utilize computational fluid dynamics.

# References

[1] E. Albano and W. P. Rodden. A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. *AIAA Journal*, 7:279–285, 1969. doi:10.2514/3.5086.

[2] K. J. Badcock and M. A. Woodgate. Bifurcation prediction of large-order aeroelastic models. *AIAA Journal*, 48:1037–1046, 2010. doi:10.2514/1.40961.

[3] K. J. Badcock, M. A. Woodgate, and B. E. Richards. Direct aeroelastic bifurcation analysis of a symmetric wing based on the euler equations. *Journal of Aircraft*, 42:731–737, 2005. doi:10.2514/1.5323.

[4] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[5] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[6] K. G. Bhatia. Airplane aeroelasticity: Practice and potential. *Journal of Aircraft*, 40:1010–1018, 2003. doi:10.2514/2.7210.

[7] K. G. Bhatia and C. S. Rudisill. Optimization of complex structures to satisfy flutter requirements. *AIAA Journal*, 9:1487–1491, 1971. doi:10.2514/3.6389.

[8] M. Bhatia, R. K. Kapania, and R. T. Haftka. Structural and aeroelastic characteristics of truss-braced wings: A parametric study. *Journal of Aircraft*, 49:302–310, 2012. doi:10.2514/1.C031556.

[9] S. A. Brown. Displacement extrapolation for CFD+CSM aeroelastic analysis. AIAA Paper 97-1090, 1997.

[10] M. J. de C. Henshaw, K. J. Badcock, G. Vio, C. Allen, J. Chamberlain, I. Kaynes, G. Dimitriadis, J. Cooper, M. Woodgate, A. Rampurawala, D. Jones, C. Fenwick, A. Gaitonde, N. Taylor, D. Amor, T. Eccles, and C. Denley. Non-linear aeroelastic prediction for aircraft applications. *Progress in Aerospace Sciences*, 43(46):65 – 137, 2007. ISSN 0376-0421. doi:10.1016/j.paerosci.2007.05.002.

[11] M. Drela. Integrated simulation model for preliminary aerodynamic, structural and control-law design of aircraft. In *Proceedings of the 40th AIAA Structures Dynamics and Materials Conference*, St. Louis, MO, 1999. AIAA 99-1394.

[12] E. N. Dvorkin and K.-J. Bathe. A continuum mechanics based four-node shell element for general nonlinear analysis. *Engineering Computations*, 1:77–88, 1984.

[13] L. L. Erickson. Panel methods: An introduction. Technical Report NASA TP-2995, NASA Ames Research Center, Moffett Field, California, 1990.

[14] D. Fox and J. Simo. A drill rotation formulation for geometrically exact shells. *Computer Methods in Applied Mechanics and Engineering*, 98(3):329 – 343, 1992. ISSN 0045-7825. doi:10.1016/0045-7825(92)90002-2.

[15] B. Grossman, Z. Gürdal, R. T. Haftka, G. J. Strauch, and W. M. Eppard. Integrated aerodynamic/structural design of a sailplane wing. *Journal of Aircraft*, 25(9):855–860, 2013/09/28 1988. doi:10.2514/3.45980.

[16] L. B. Gwin and R. F. Taylor. A general method for flutter optimization. *AIAA Journal*, 11:1613–1617, 1973. doi:10.2514/3.50657.

[17] R. T. Haftka. Optimization of flexible wing structures subject to strength and induced drag constraints. *AIAA Journal*, 15(8): 1101–1106, August 1977. doi:10.2514/3.7400.

[18] S. Haghighat, H. H. T. Liu, and J. R. R. A. Martins. A model predictive gust load alleviation controller for a highly flexible aircraft. *Journal of Guidance, Control and Dynamics*, 36:1751–1766, 2012. doi:10.2514/1.57013.

[19] S. Haghighat, J. R. R. A. Martins, and H. H. T. Liu. Aeroservoelastic design optimization of a flexible wing. *Journal of Aircraft*, 49(2):432–443, 2012. doi:10.2514/1.C031344.

[20] K. C. Hall, J. P. Thomas, and W. S. Clark. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. *AIAA Journal*, 40:879–886, 2002. doi:10.2514/2.1754.

[21] J. Hess and A. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aerospace Sciences*, 8:1–138, 1967.

[22] T. J. R. Hughes and F. Brezzi. On drilling degrees of freedom. *Computer Methods in Applied Mechanics and Engineering*, 72:105–121, 1989.

[23] W. H. Hui and M. Tobak. Bifurcation analysis of aircraft pitching motions about large mean angles of attack. *Journal of Guidance, Control, and Dynamics*, 7:113–122, 1984. doi:10.2514/3.8553.

[24] J. Katz and A. Plotkin. *Low–Speed Aerodynamics*. McGraw–Hill Inc., 1991.

[25] G. J. Kennedy and J. R. R. A. Martins. A comparison of metallic and composite aircraft wings using aerostructural design optimization. In *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, September 2012.

[26] G. J. Kennedy and J. R. R. A. Martins. An adjoint-based derivative evaluation method for time-dependent aeroelastic optimization of flexible aircraft. In *Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Boston, MA, April 2013.

[27] G. J. Kennedy and J. R. R. A. Martins. A parallel finite element framework for large-scale gradient-based design optimization of aerospace structures. *Finite Elements in Analysis and Design*, 2014. doi:10.1016/j.finel.2014.04.011. Accepted.

[28] G. J. Kennedy and J. R. R. A. Martins. A parallel aerostructural optimization framework for aircraft design studies. *Structural and Multidisciplinary Optimization*, 2014. doi:10.1007/s00158-014-1108-9. Accepted.

[29] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Aerostructural optimization of the common research model configuration. In *Aviation 2014*, Atlanta, Georgia, June 2014.

[30] G. K. W. Kenway, G. J. Kennedy, and J. R. R. A. Martins. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA Journal*, 52:935–951, 2014. doi:10.2514/1.J052255.

[31] E. Livne. Future of airplane aeroelasticity. *Journal of Aircraft*, 40:1066–1092, 2003. doi:10.2514/2.7218.

[32] Z. Lyu, G. K. Kenway, and J. R. R. A. Martins. Aerodynamic shape optimization studies on the common research model wing benchmark. *AIAA Journal*, 2014. (In press).

[33] C. A. Mader and J. R. R. A. Martins. Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. *AIAA Journal*, 50:2809–2819, 2012. doi:10.2514/1.J051658.

[34] C. A. Mader, J. R. R. A. Martins, J. J. Alonso, and E. van der Weide. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA Journal*, 46(4):863–873, April 2008. doi:10.2514/1.29123.

[35] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, November 2013. doi:10.2514/1.J052184.

[36] J. R. R. A. Martins, P. Sturdza, and J. J. Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software*, 29(3):245–262, Sept. 2003. doi:10.1145/838250.838251.

[37] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. High-fidelity aerostructural design optimization of a supersonic business jet. *Journal of Aircraft*, 41(3):523–530, 2004. doi:10.2514/1.11478.

[38] J. R. R. A. Martins, J. J. Alonso, and J. J. Reuther. A coupled–adjoint sensitivity analysis method for high–fidelity aero–structural design. *Optimization and Engineering*, 6:33–62, 2005. doi:10.1023/B:OPTE.0000048536.47956.62.

[39] M. McMullen, A. Jameson, and J. Alonso. Demonstration of nonlinear frequency domain methods. *AIAA Journal*, 44: 1428–1435, 2006. doi:10.2514/1.15127.

[40] S. A. Morton and P. S. Beran. Hopf-bifurcation analysis of airfoil flutter at transonic speeds. *Journal of Aircraft*, 36:421–429, 1999. doi:10.2514/2.2447.

[41] D. J. Neill, E. H. Johnson, and R. Canfield. ASTROS - a multidisciplinary automated structural design tool. *Journal of Aircraft*, 27:1021–1027, 1990. doi:10.2514/3.45976.

[42] C. G. Raspanti, J. A. Bandoni, and L. T. Biegler. New strategies for flexibility analysis and design under uncertainty. *Computers and Chemical Engineering*, 24:2193–2209, 2000.

[43] W. P. Rodden, P. F. Taylor, and S. C. McIntosh. Further refinement of the subsonic doublet-lattice method. *Journal of Aircraft*, 35:720–727, 1998. doi:10.2514/2.2382.

[44] C. S. Rudisill and K. G. Bhatia. Second derivatives of the flutter velocity and the optimization of aircraft structures. *AIAA Journal*, 10:1569–1572, 1972. doi:10.2514/3.6690.

[45] Y. Saad. *Numerical methods for large eigenvalue problems*. Algorithms and architectures for advanced scientific computing. Manchester University Press, 1992. ISBN 9780719033865.

[46] Y. Saad. A fexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993. doi:10.1137/0914028.

[47] C. M. Shearer and C. E. Cesnik. Nonlinear flight dynamics of very flexible aircraft. *Journal of Aircraft*, 44:1528 –1545, 2007. doi:10.2514/1.27606.

[48] M. H. Shirk, T. J. Hertz, and T. A. Weisshaar. Aeroelastic tailoring - Theory, practice, and promise. *Journal of Aircraft*, 23: 6–18, 1986. doi:10.2514/3.45260.

[49] G. Sleijpen and H. Van der Vorst. A Jacobi–Davidson iteration method for linear eigenvalue problems. *SIAM Review*, 42(2): 267–293, 2000. doi:10.1137/S0036144599363084.

[50] S. Smith. A computational and experimental study of nonlinear aspects of induced drag. Tech. Rep. NASA TP 3598, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, 94035-1000, 1996.

[51] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, 1998. doi:10.1137/S003614459631241X.

[52] B. Stanford and P. Beran. Direct flutter and limit cycle computations of highly flexible wings for efficient analysis and optimization. *Journal of Fluids and Structures*, 36(0):111 – 123, 2013. ISSN 0889-9746. doi:10.1016/j.jfluidstructs.2012.08.008.

[53] J. P. Thomas, E. H. Dowell, and K. C. Hall. Modeling viscous transonic limit cycle oscillation behavior using a harmonic balance approach. *Journal of Aircraft*, 41:1266–1274, 2004. doi:10.2514/1.9839.

[54] F. Tisseur and K. Meerbergen. The quadratic eigenvalue problem. *SIAM Review*, 43(2):235–286, 2001. doi:10.1137/S0036144500381988.

[55] M. J. Turner. Optimization of structures to satisfy flutter requirements. *AIAA Journal*, 7:945–951, 1969. doi:10.2514/3.5248.

[56] E. van der Weide, A. Gopinath, and A. Jameson. Turbomachinery applications with the time spectral method. In *35th AIAA Fluid Dynamics Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2005. doi:10.2514/6.2005-4905.

[57] J. Vassberg, M. Dehaan, M. Rivers, and R. Wahls. Development of a common research model for applied cfd validation studies. In *26th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, August 2008. doi:10.2514/6.2008-6919.

[58] J. C. Vassberg. A unified baseline grid about the common research model wing-body for the fifth aiaa cfd drag prediction workshop. In *29th Applied Aerodynamics Conference*, Honolulu, Hawaii, June 2011. doi:10.2514/6.2011-3508. AIAA 2011-3508.

[59] S. Wakayama and I. Kroo. Subsonic wing planform design using multidisciplinary optimization. 32(4):746–753, 1995. doi:10.2514/3.46786.

[60] G. Wrenn. An indirect method for numerical optimization using the Kreisselmeier-Steinhauser function. NASA Technical Report CR-4220, 1989.