

Fixed- and Free-Mode Flutter Derivatives during Aeroelastic Optimization

Bret K. Stanford*, Andrew S. Thelen†, Kevin E. Jacobson‡

NASA Langley Research Center, Hampton, VA, 23681

Aeroelastic flutter design derivatives may be approximated by ignoring the sensitivity of the structural mode shapes. This “fixed-mode” derivative is less-expensive to compute than the exact “free-mode” derivative (which accommodates the mode shape dependency), but also may provide inaccurate sensitivities for optimization. This work formulates both fixed- and free-mode flutter derivatives, and demonstrates the conditions under which they are equivalent. Two different types of fixed-mode derivatives are also developed in this work. Using a cantilevered-plate demonstration problem with both shape and sizing design parameters, the accuracy of the free-mode derivative and the two fixed-mode derivatives, are all demonstrated via gradient-based optimization.

I. Introduction

Aeroelastic design optimization of wing structures must consider, among numerous other failure and performance metrics, a flutter constraint. The optimized wing must not flutter within the flight envelope, in addition to a prescribed empirical flutter margin. For design scenarios with large numbers of sizing and/or shape design variables, this is accomplished with gradient-based optimization, and requires the derivative of the flutter boundary with respect to each variable. A typical workflow for computing flutter (and its derivatives) involves first solving the structural dynamic eigenproblem for the structural mode shapes, and then mapping these mode shapes onto the aerodynamic mesh. The aerodynamic mode shapes are then used, along with linearized aerodynamic influence coefficients (AICs) computed in the frequency domain (typically with the doublet lattice method, DLM) to obtain the generalized aerodynamic forces (GAFs). These GAFs can finally be used to compute flutter via a nonlinear aeroelastic eigenvalue problem.

Flutter derivatives with respect to a large number of design variables requires reversing the above process, and a choice must be made as to whether, or not, to include the derivative of the mode shapes in this chain rule. In general, the mode shapes will be functions of structural and sizing design variables, and their sensitivities must be included in the overall chain rule to preserve an exact flutter derivative. However, a common assumption in the flutter optimization literature is to ignore this mode shape sensitivity, termed a “fixed-mode” derivative.¹ The computation of mode shape derivatives (typically done with Nelson’s method,² though alternative methods are reviewed in Refs. 3 and 4) has a relatively high computational cost, which motivates the fixed-mode approximation. Very few flutter optimization papers have included these modal derivatives (“free-mode” derivatives: also termed “updated-mode” derivatives, though the former phrase is used here); some examples are found in Refs. 5 - 9.

After the eigenproblem is solved for the first N mode shapes, Nelson’s method will require the solution of N separate linear systems of equations, as will be derived below. A fixed-mode approximation is cheaper, but because the modes are actually updated from one design iteration to the next, the derivatives are inexact, potentially slowing down or preventing the optimization convergence. Various papers^{1,9} have speculated that a fixed-mode approximation is particularly ill-suited for shape design variables, as opposed to sizing design variables, but the current authors are unaware of any research that has actually demonstrated this within the context of flutter optimization. Refs. 10 and 11 do explore some of these issues in the context of damped structural dynamics, however.

*Research Aerospace Engineer, Aeroelasticity Branch, bret.k.stanford@nasa.gov, AIAA Associate Fellow.

†Research Aerospace Engineer, Aeroelasticity Branch, AIAA Member.

‡Research Aerospace Engineer, Aeroelasticity Branch, AIAA Member.

A demonstration of the importance of free-mode derivatives for flutter-based optimization with shape and/or sizing design variables is the goal of the current work. Results in this paper are demonstrated via a numerical cantilevered-plate wing operating under subsonic flows, with planform shape variables, and thickness (sizing) variables attached to each plate finite element. Flutter computations and derivatives are computed with the assistance of OpenMDAO¹² and MPhys.^{13,14} MPhys has been developed as a collaborative effort seeking to standardize high-fidelity multiphysics problems (in this case, unsteady aeroelastic problems) in OpenMDAO. The MPhys library also provides utilities to help set up these multiphysics optimizations that can quickly grow in complexity.

II. Flutter Computations

A nonlinear flutter eigenvalue problem is written as:

$$(p \cdot \overline{\mathbf{A}} - \overline{\mathbf{B}}(i \cdot k)) \cdot \overline{\mathbf{R}} = \mathbf{0} \quad (1)$$

where p is the nondimensional eigenvalue, written as $p = b \cdot s / U_\infty$. The variable s is the Laplacian, b is a reference length, and U_∞ is the flow speed. The real and imaginary parts of p are expressed as $p = g + i \cdot k$, where k is the reduced frequency, and g is a damping parameter. $\overline{\mathbf{R}}$ in Eq. 1 is the corresponding right eigenvector ($\overline{\mathbf{L}}$ is the corresponding left eigenvector), and:

$$\overline{\mathbf{A}} := \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(\frac{U_\infty}{b}\right)^2 \cdot \Phi_s^T \cdot \mathbf{M} \cdot \Phi_s \end{bmatrix} \quad (2)$$

$$\overline{\mathbf{B}} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\Phi_s^T \cdot \mathbf{K} \cdot \Phi_s + q_\infty \cdot \overline{\mathbf{Q}}(i \cdot k) & -\left(\frac{U_\infty}{b}\right) \cdot \Phi_s^T \cdot \mathbf{C} \cdot \Phi_s \end{bmatrix} \quad (3)$$

$\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the mass, damping, and stiffness matrices, and Φ_s is the modal reduction matrix. $\overline{\mathbf{Q}}$ is a complex matrix of GAFs, a nonlinear function of the reduced frequency, and the dynamic pressure $q_\infty = \frac{1}{2} \cdot \rho_\infty \cdot U_\infty^2$.

The modal reduction matrix is computed via the eigenvalue problem:

$$\mathbf{R}_{\phi_n} := (\mathbf{K} - \omega_n^2 \cdot \mathbf{M}) \cdot \phi_n = \mathbf{0} \quad (4)$$

where ω_n is the n^{th} natural frequency, and ϕ_n is the corresponding mode shape. This eigenvalue problem is satisfied when the residual $\mathbf{R}_{\phi_n}$ is $\mathbf{0}$. The first N mode shapes may be grouped into the matrix Φ_s , where N is far smaller than the number of degrees of freedom in the finite element analysis. The mode shapes are normalized as:

$$R_{\omega_n} := \phi_n^T \cdot \mathbf{M} \cdot \phi_n - 1 = 0 \quad (5)$$

where again this normalization condition is satisfied when the residual $\mathbf{R}_{\omega_n}$ is 0.

The eigen-equation Eq. 1 is nonlinear because the reduced frequency, k , is both an input (to the $\overline{\mathbf{B}}$ matrix via the GAFs), and also an output (the imaginary part of p). A matched-point solution of Eq. 1 would therefore ensure that $\text{imag}(p) = k$. A flutter condition is encountered when the damping of a given eigenvalue becomes zero: $\text{real}(p) = g = 0$. These conditions are met when the following residual is driven to $\mathbf{0}$:

$$\mathbf{R}_u := \left\{ \begin{array}{cc} \text{real}(p) & k - \text{imag}(p) \end{array} \right\}^T = \mathbf{0} \quad (6)$$

The corresponding solution vector used to solve Eq. 6 is:

$$\mathbf{u} = \left\{ \begin{array}{cc} q_\infty & k \end{array} \right\}^T \quad (7)$$

III. Flutter Derivatives

It is desired to compute the derivative of some scalar output, J , with respect to a vector of design variables, $\mathbf{s}$. In this work J is the flutter dynamic pressure q_∞ , and so is only an explicit function of $\mathbf{u}$. Some form of aggregated aeroelastic damping¹⁵ is also commonly used in aeroelastic design constraints, but this is not considered here.

A. Free-Mode Derivatives

A Lagrangian may be considered as:

$$\mathcal{L} := J + \boldsymbol{\lambda}_u^T \cdot \mathbf{R}_u + \sum_{n=1}^N \lambda_{\omega_n} \cdot R_{\omega_n} + \sum_{n=1}^N \boldsymbol{\lambda}_{\phi_n}^T \cdot \mathbf{R}_{\phi_n} \quad (8)$$

where $\boldsymbol{\lambda}_u$, λ_{ω_n} , and $\boldsymbol{\lambda}_{\phi_n}$ are adjoint vectors corresponding to the residuals defined above. Including every term in Eq. 8 provides a free-mode derivative, which will be demonstrated in this section. A fixed-mode derivative excludes the final term in Eq. 8, and this will be demonstrated in later sections.

The sought-after derivative of J with respect to $\mathbf{s}$ is:

$$\frac{dJ}{ds} = \frac{\partial \mathcal{L}}{\partial \mathbf{s}} = \boldsymbol{\lambda}_u^T \cdot \frac{\partial \mathbf{R}_u}{\partial \mathbf{s}} + \sum_{n=1}^N \lambda_{\omega_n} \cdot \frac{\partial R_{\omega_n}}{\partial \mathbf{s}} + \sum_{n=1}^N \boldsymbol{\lambda}_{\phi_n}^T \cdot \frac{\partial \mathbf{R}_{\phi_n}}{\partial \mathbf{s}} \quad (9)$$

The $\mathbf{R}_u$ derivative in Eq. 9 is computed via the derivative of Eq. 1:

$$\frac{\partial p}{\partial \mathbf{s}} = \left(\bar{\mathbf{L}}^T \cdot \left(\frac{\partial \bar{\mathbf{B}}}{\partial \mathbf{s}} - p \cdot \frac{\partial \bar{\mathbf{A}}}{\partial \mathbf{s}} \right) \cdot \bar{\mathbf{R}} \right) / \left(\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}} \right) \quad (10)$$

$$\frac{\partial \mathbf{R}_u}{\partial \mathbf{s}} = \left\{ \begin{array}{cc} \text{real} \left(\frac{\partial p}{\partial \mathbf{s}} \right) & -\text{imag} \left(\frac{\partial p}{\partial \mathbf{s}} \right) \end{array} \right\}^T \quad (11)$$

The modal reduction of Eqs. 4 and 5 is differentiated as:¹¹

$$\frac{\partial \mathbf{R}_{\phi_n}}{\partial \mathbf{s}} = \left(\frac{\partial \mathbf{K}}{\partial \mathbf{s}} - \omega_n^2 \cdot \frac{\partial \mathbf{M}}{\partial \mathbf{s}} \right) \cdot \boldsymbol{\phi}_n \quad (12)$$

$$\frac{\partial R_{\omega_n}}{\partial \mathbf{s}} = \boldsymbol{\phi}_n^T \cdot \frac{\partial \mathbf{M}}{\partial \mathbf{s}} \cdot \boldsymbol{\phi}_n \quad (13)$$

The adjoint vectors needed in Eq. 9 are computed by setting the partial derivatives of $\mathcal{L}$ with respect to $\mathbf{u}$, ω_n , and $\boldsymbol{\phi}_n$, all to 0. The adjoint term $\boldsymbol{\lambda}_u$ is computed via:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \mathbf{0} \quad \Rightarrow \quad \left(\frac{\partial \mathbf{R}_u}{\partial \mathbf{u}} \right)^T \cdot \boldsymbol{\lambda}_u = -\frac{\partial J}{\partial \mathbf{u}} \quad (14)$$

Similarly, the modal adjoint terms are computed as:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\phi}_n} = \mathbf{0} \quad \Rightarrow \quad (\mathbf{K} - \omega_n^2 \cdot \mathbf{M}) \cdot \boldsymbol{\lambda}_{\phi_n} + 2 \cdot \mathbf{M} \cdot \boldsymbol{\phi}_n \cdot \lambda_{\omega_n} = -\left(\frac{\partial \mathbf{R}_u}{\partial \boldsymbol{\phi}_n} \right)^T \cdot \boldsymbol{\lambda}_u \quad (15)$$

$$\frac{\partial \mathcal{L}}{\partial \omega_n} = 0 \quad \Rightarrow \quad 2 \cdot \omega_n \cdot \boldsymbol{\phi}_n^T \cdot \mathbf{M} \cdot \boldsymbol{\lambda}_{\phi_n} = \left(\frac{\partial \mathbf{R}_u}{\partial \omega_n} \right)^T \cdot \boldsymbol{\lambda}_u \quad (16)$$

Derivatives of $\mathbf{R}_u$ with respect to ω_n and $\boldsymbol{\phi}_n$ are computed with expressions similar to Eqs. 10 and 11. Pre-multiplying Eq. 15 by $\boldsymbol{\phi}_n^T$ yields an expression for λ_{ω_n} :

$$\lambda_{\omega_n} = -\boldsymbol{\phi}_n^T \cdot \left(\frac{\partial \mathbf{R}_u}{\partial \boldsymbol{\phi}_n} \right)^T \cdot \boldsymbol{\lambda}_u / 2 \quad (17)$$

Eq. 15 is not easily solved for $\boldsymbol{\lambda}_{\phi_n}$, since the matrix $\mathbf{K} - \omega_n^2 \cdot \mathbf{M}$ is singular. Nelson's method² can then be used, where the desired vector is assumed to have a form: $\boldsymbol{\lambda}_{\phi_n} = \mathbf{V} + c \cdot \boldsymbol{\phi}_n$. Eq. 15 is solved with $\mathbf{V}$ in

place of λ_{ϕ_n} , and with the k^{th} row and column removed from the system. The k^{th} member of $\mathbf{V}$ is set to 0. The scalar c is finally computed from Eq. 16:

$$2 \cdot \omega_n \cdot \phi_n^T \cdot \mathbf{M} \cdot (\mathbf{V} + c \cdot \phi_n) = \left(\frac{\partial \mathbf{R}_u}{\partial \omega_n} \right)^T \cdot \lambda_u \quad (18)$$

$$c = -\phi_n^T \cdot \mathbf{M} \cdot \mathbf{V} + \left(\frac{\partial \mathbf{R}_u}{\partial \omega_n} \right)^T \cdot \lambda_u / 2 / \omega_n \quad (19)$$

With these adjoint vectors, the final derivative of interest in Eq. 9 is written as:

$$\frac{dJ}{ds} = \lambda_u^T \cdot \frac{\partial \mathbf{R}_u}{\partial s} + \sum_{n=1}^N \lambda_{\phi_n}^T \cdot \left(\frac{\partial \mathbf{K}}{\partial s} - \omega_n^2 \cdot \frac{\partial \mathbf{M}}{\partial s} \right) \cdot \phi_n + \sum_{n=1}^N \lambda_{\omega_n} \cdot \phi_n^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \phi_n \quad (20)$$

The large computational cost of Nelson's method, and of the free-mode derivative, is via Eq. 15: a distinct linear system of equations that must be solved for each of the N mode shapes.

B. Fixed-Mode Derivatives

As noted above, a fixed-mode derivative removes the ϕ_n terms from the Lagrangian:

$$\mathcal{L} := J + \lambda_u^T \cdot \mathbf{R}_u + \sum_{n=1}^N \lambda_{\omega_n} \cdot R_{\omega_n} \quad (21)$$

The ω_n residual is necessarily redefined here as:

$$R_{\omega_n} := \phi_n^T \cdot (\mathbf{K} - \omega_n^2 \cdot \mathbf{M}) \cdot \phi_n = 0 \quad (22)$$

As above, the adjoint term λ_u is computed with Eq. 14. The ω_n adjoint term is computed as:

$$\frac{\partial \mathcal{L}}{\partial \omega_n} = 0 \quad \Rightarrow \quad \lambda_{\omega_n} = \left(\frac{\partial \mathbf{R}_u}{\partial \omega_n} \right)^T \cdot \lambda_u / 2 / \omega_n \quad (23)$$

The final derivative of J with respect to s is:

$$\frac{dJ}{ds} = \frac{\partial \mathcal{L}}{\partial s} = \lambda_u^T \cdot \frac{\partial \mathbf{R}_u}{\partial s} + \sum_{n=1}^N \lambda_{\omega_n} \cdot \frac{\partial R_{\omega_n}}{\partial s} \quad (24)$$

$$\frac{dJ}{ds} = \lambda_u^T \cdot \frac{\partial \mathbf{R}_u}{\partial s} + \sum_{n=1}^N \lambda_{\omega_n} \cdot \left(\phi_n^T \cdot \frac{\partial \mathbf{K}}{\partial s} \cdot \phi_n - \omega_n^2 \cdot \phi_n^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \phi_n \right) \quad (25)$$

Because Nelson's method is avoided here, the computational cost of Eq. 25 is much lower than Eq. 20.

There are some specific situations where the fixed-mode derivative (Eq. 25) is equal to the free-mode derivative (Eq. 20), namely when $\frac{\partial \mathbf{K}}{\partial s}$ is proportional to $\mathbf{K}$, and $\frac{\partial \mathbf{M}}{\partial s}$ is proportional to $\mathbf{M}$. This can occur when a design variable imparts a uniform stiffness and/or inertial change to a structure: for example computing derivatives with respect to material properties (as long as the structure is composed of a single material). A shape design variable will not, in general, meet these proportionality conditions. Additional discussion of this idea is provided in Appendix A.

Furthermore, different fixed-mode derivatives can be obtained depending on how the matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are defined. They may be defined as written in Eq. 2 and Eq. 3, and analytical expressions for various derivatives with respect to ρ_∞ , U_∞ , k , and Φ can be derived as needed. Alternatively, they may be defined as:

$$\bar{\mathbf{A}} := \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \left(\frac{U_\infty}{b} \right)^2 \cdot \mathbf{I} \end{bmatrix} \quad (26)$$

$$\bar{\mathbf{B}} := \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\Omega + q_\infty \cdot \bar{\mathbf{Q}}(i \cdot k) & -\left(\frac{U_\infty}{b} \right) \cdot \Sigma \end{bmatrix} \quad (27)$$

The diagonal matrix $\mathbf{\Omega}$ contains ω_n^2 along the diagonal, and $\mathbf{\Sigma}$ is some diagonal modal damping matrix.

The matrices in Eq. 2 and Eq. 3 are identical to those in Eq. 26 and Eq. 27, and either expression will provide the same flutter solution. Either expression will provide the same free-mode derivative also, but for fixed-mode derivatives, they will lead to different answers. Here, a fixed-mode derivative with matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ defined in Eq. 2 and Eq. 3 is termed fixed-mode approximation 1, and matrices from Eq. 26 and Eq. 27 is termed fixed-mode approximation 2. Additional discussion of their differences is provided in Appendix B.

If the proportionality requirements discussed above are met, then fixed-mode approximation 1 will return the same result as the free-mode derivative; fixed-mode approximation 2 will not.

IV. MPhys Implementation

As noted above, the flutter computation and derivative process is handled with OpenMDAO and MPhys. The MPhys-generated aeroelastic workflow can be seen in the extended design structure matrix (XDSM) diagram¹⁶ of Fig. 1:

1. An optimizer dictates shape $\mathbf{s}_{shape}$ and sizing $\mathbf{s}_{sizing}$ design variables.
2. A geometry component uses the shape variables to determine coordinates of the aerodynamic ($\mathbf{x}_{aero}$) and structural ($\mathbf{x}_{struct}$) nodes on the wing.
3. A structural modal analysis computes the matrix of structural mode shapes $\mathbf{\Phi}_s$, as in Eqs. 4 and 5, and the modally-reduced mass, damping, and stiffness matrices ($\bar{\mathbf{M}} = \mathbf{\Phi}_s^T \cdot \mathbf{M} \cdot \mathbf{\Phi}_s$, etc.)
4. The mode shapes are interpolated onto the aerodynamic surface to form $\mathbf{\Phi}_a$.
5. An unsteady oscillatory solver (typically the DLM) computes the matrix of GAFs $\bar{\mathbf{Q}}$.
6. The flutter metric of interest, J , is computed. The $p - k$ method is used to find an initial guess of the flutter solution in Eq. 7, and then the flutter point is found precisely by driving the residual of Eq. 6 to zero.

OpenMDAO can automatically handle the forward connections between each component in Fig. 1 (e.g., $\mathbf{\Phi}_s$, $\mathbf{\Phi}_a$, $\bar{\mathbf{Q}}$, and finally J), as well as the reverse connections of the adjoint vectors (e.g., λ_u , λ_{ϕ_n} , λ_{ω_n}) for the computation of the final flutter derivative.

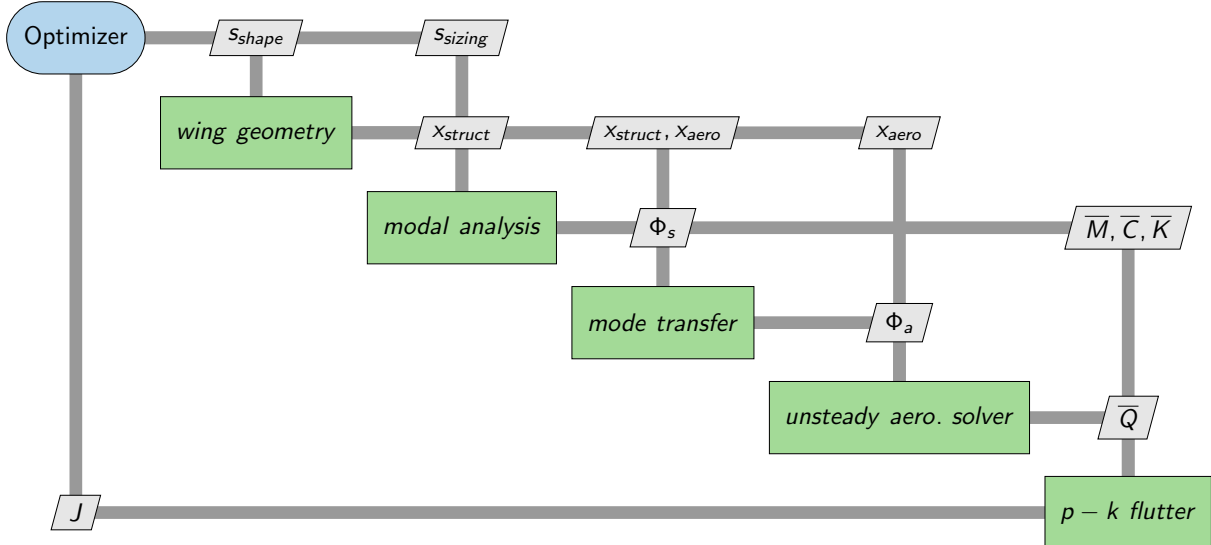


Figure 1. XDSM¹⁶ diagram of the aeroelastic flutter optimization process.

The example problem considered here uses the exact workflow shown in Fig. 1. However, a key feature of the MPhys library is swappable fidelity, and higher-fidelity aerodynamics (based on computational fluid dynamics, CFD, for example^{6,17}) may be utilized also to compute GAFs. In this case, a steady CFD-based

flow solver must be used to compute a steady flow vector, and this steady flow vector is then fed to the unsteady aerodynamic solver for the GAF linearization. For lifting flows with a static aeroelastic component, a substantially more complex OpenMDAO model is required (see Ref. 18).

V. Results

The wing demonstration case used here is shown in Fig. 2. There are 4 planform shape design variables, attached to the root chord, the tip chord, the tip sweep (defined here as the streamwise distance between the root leading edge and the tip leading edge), and the span. Various planform shapes are considered in this paper, but the wing in Fig. 2 has a root chord of 0.5 m, a tip chord of 0.35 m, a tip sweep of 0.8 m, and a span of 0.75 m. Each additive design variable ranges from -0.15 to 0.15, with a baseline value of 0; the wingspan in Fig. 2 can therefore range between 0.6 to 0.9 m, for example. The structural discretization is also shown in the figure: a thickness (sizing) design variable is attached to each shell finite element, and can range between 0.003 and 0.01 m.

Flutter dynamic pressures (q_∞) are computed at Mach 0.5, using linear DLM aerodynamics, and a fully-clamped boundary condition at the wing root.

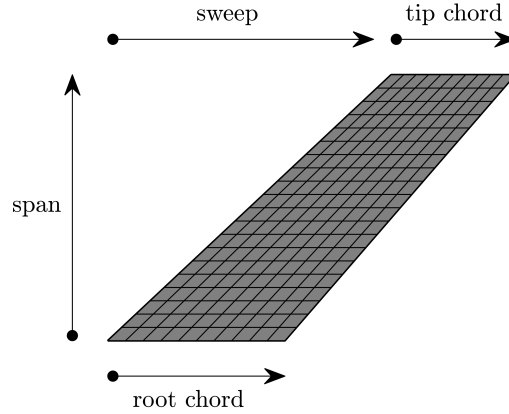


Figure 2. Shape and sizing design variables for the cantilevered wing.

Analytical flutter derivatives are compared against finite difference estimates in Tables 1 (root and tip chord), 2 (span and sweep), and 3 (thickness). Derivative verification is typically done with complex step comparisons, but this is impossible here due to the fact that the flutter computation itself utilizes complex-valued operations. Analytical free-mode derivatives are computed with the MPhys setup described above, whereas the analytical fixed-mode derivatives (both approximation 1 and 2) are found by omitting the relevant terms from the derivations discussed above. In order to properly verify the implementation of the fixed-mode derivative via finite differences, the finite perturbation must not include a perturbed mode shape. Finite differences are then found by utilizing a set of structural mode shapes (Φ_s) read in externally, rather than computing new modes corresponding to the perturbed design.

For the four shape design variables, the key finding is that fixed-mode approximation 1 is relatively accurate compared to the free-mode result, but fixed-mode approximation 2 is not. For the sweep design variable, approximation 1 is in error by 5.5% (the largest error in Tables 1 and 2), but approximation 2 is in error by 168%. The free- and fixed-mode (approximation 2) finite differences are generally well-converged for step sizes 10^{-5} - 10^{-6} , but fixed-mode approximation 1 converges much slower (particularly for sweep and the two chord variables), indicating a large amount of curvature in those design spaces.

The thickness design variable is a vector (the thickness of every finite element, as seen in Fig. 2), and so the sum of the derivatives (a scalar) is shown in Table 3. Conclusions from Table 3 are largely the same as the previous tables, with the exception that the free-mode and fixed-mode (approximation 1) derivatives are identical. This equivalence exists because the proportionality conditions discussed in Appendix A are met. For a flat plate with a uniform thickness, the mass matrix may be written as $\mathbf{M} = (t_0 \cdot (1 + t_{dv})) \cdot \mathbf{M}_0$,

Table 1. Free- and fixed-mode flutter derivatives with respect to chord design variables.

	$d_{root\ chord}$			$d_{tip\ chord}$		
	free	fixed 1	fixed 2	free	fixed 1	fixed 2
$\delta = 10^{-3}$	5777.07593	16975.99037	10734.32032	-21928.07095	38954.69235	-25124.01137
$\delta = 10^{-4}$	5777.69080	6926.90966	10749.86978	-21972.48642	-15920.04515	-25146.32561
$\delta = 10^{-5}$	5777.75598	5921.36566	10751.41256	-21976.93877	-21428.35021	-25148.56503
$\delta = 10^{-6}$	5777.75075	5820.81950	10751.46159	-21977.41501	-21979.40629	-25148.90190
$\delta = 10^{-7}$	5777.73832	5810.84701	10750.34899	-21977.58262	-22034.60198	-25150.02899
$\delta = 10^{-8}$	5778.15008	5810.71999	10751.45009	-21975.41998	-22038.95001	-25147.51995
analytical	5777.76130	5809.67596	10751.59805	-21977.43147	-22040.84586	-25148.80007

Table 2. Free- and fixed-mode flutter derivatives with respect to span and sweep design variables.

	d_{span}			d_{sweep}		
	free	fixed 1	fixed 2	free	fixed 1	fixed 2
$\delta = 10^{-3}$	-29941.12644	-30066.72834	-51211.93484	-4072.69147	4597.23506	-10952.38776
$\delta = 10^{-4}$	-29999.28182	-30192.96437	-51352.97089	-4076.74061	-3007.07980	-10950.42355
$\delta = 10^{-5}$	-30005.10808	-30206.01130	-51367.11898	-4077.14698	-3768.66254	-10950.23845
$\delta = 10^{-6}$	-30005.66747	-30207.31579	-51368.64350	-4077.16926	-3844.82419	-10950.33939
$\delta = 10^{-7}$	-30005.80606	-30207.39999	-51369.88800	-4077.38447	-3852.58499	-10951.48600
$\delta = 10^{-8}$	-30004.25986	-30206.52984	-51368.42992	-4076.25993	-3853.01991	-10950.43990
analytical	-30005.75579	-30207.72258	-51368.67879	-4077.19119	-3853.33185	-10950.20335

where $\mathbf{M}_0$ is a thickness-independent mass matrix. Similarly, $\mathbf{K} = (t_0 \cdot (1 + t_{dv}))^3 \cdot \mathbf{K}_0$. The proportionality constants are then $m = 1$ and $k = 3$ (for the baseline value of $t_{dv} = 0$), and so $\frac{\partial \omega_n}{\partial s} = \omega_n$, $\frac{\partial \phi_n}{\partial s} = -\phi_n/2$, and the fixed-mode (approximation 1) is exact. Approximation 2 in Table 3 is still erroneous, as the conditions required for approximations 1 and 2 to be equal (Eqs. 38 and 39) are nearly impossible to attain.

Table 3. Free- and fixed-mode flutter derivatives with respect to the lumped thickness design variable.

	$d_{thickness}$		
	free	fixed 1	fixed 2
$\delta = 10^{-3}$	30631.72203	30631.79502	20539.35124
$\delta = 10^{-4}$	30603.15911	30602.88638	20529.27372
$\delta = 10^{-5}$	30600.30492	30600.04636	20528.25466
$\delta = 10^{-6}$	30600.02016	30599.76740	20528.03920
$\delta = 10^{-7}$	30599.90209	30599.88001	20526.94298
$\delta = 10^{-8}$	30600.82999	30599.96015	20528.54015
analytical	30599.98681	30599.98681	20528.15550

The summed-thickness sensitivity scalar is considered in Table 3, but the actual vector of thickness sensitivities is shown graphically across the wing in Fig. 3. The vectorized fixed-mode derivative (approximation 1) is very close to the free-mode derivative, but there are small differences (quantified via a log-scale in the bottom row of the figure), because the proportionality conditions are no longer exactly met. Approximation 2 is, again, far less accurate, though the qualitative trends down the wing span are reasonable.

Additional derivative accuracy information is given in Figs. 4, 5, and 6, which shows free-mode and fixed-mode flutter derivatives (thickness and sweep sensitivities) across changes in wing planform. The change

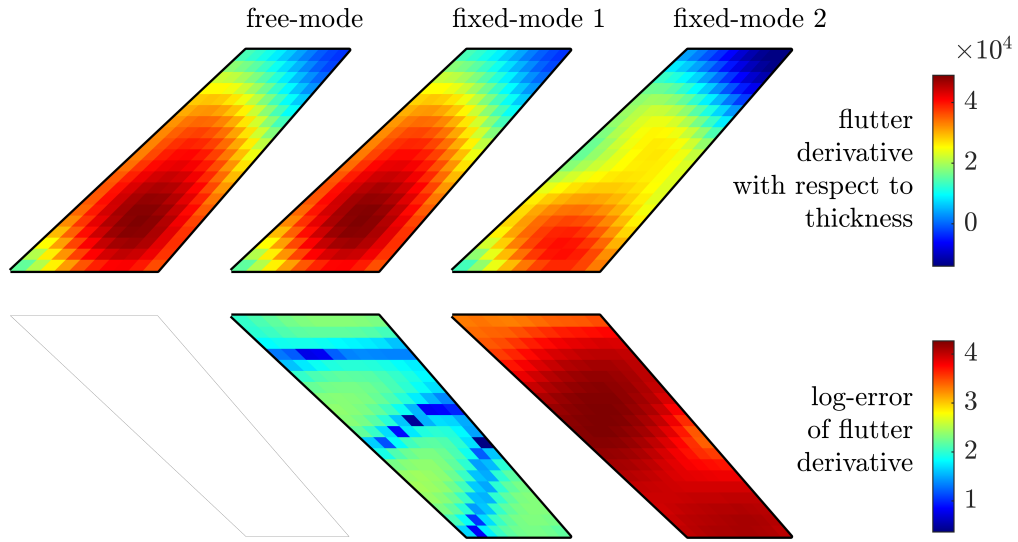


Figure 3. Contour map of sizing derivatives across the wing (top row), and log differences between fixed- and free-mode derivatives (bottom row).

in flutter q_∞ is given in each plot also, for reference. Figure 4 shows a change in span from 0.75 to 2.5 m, which leads to a monotonic drop in the flutter boundary, and a drop in the magnitude of both derivatives. Figures 5 and 6 show a change in sweep (from -0.875 to 0.8) for a low- and a high-aspect ratio wing, respectively. Transitioning from an aft-swept wing to forward-swept wing sees a “switch” in the instability mechanism, from unsteady flutter to quasi-steady divergence. The high-aspect ratio wing (Fig. 6) further sees two separate flutter mechanisms in the aft-swept regime.

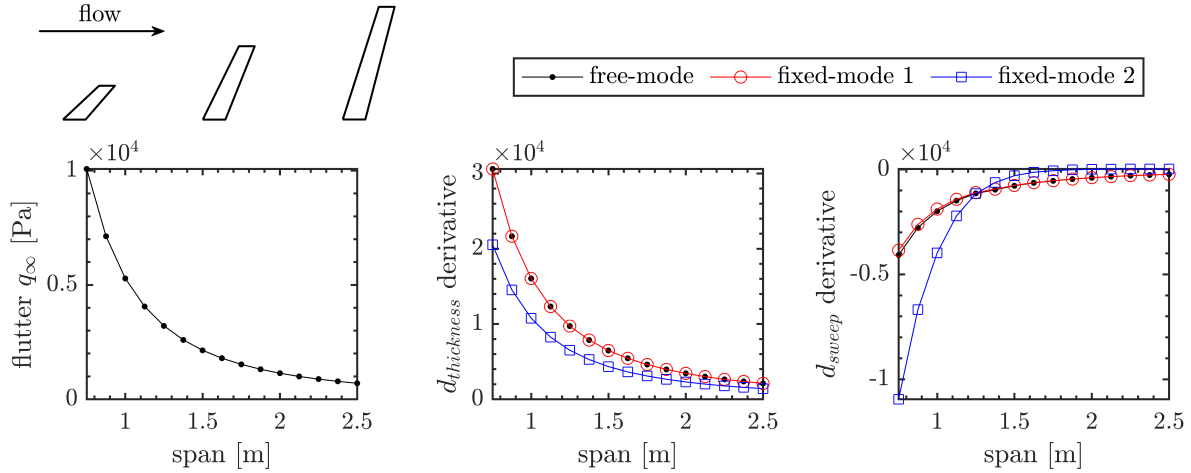


Figure 4. Flutter q_∞ (left), lumped thickness derivative (center), and sweep derivative (right), plotted as a function of the wing span.

For nearly all of these cases, fixed-mode approximation 1 is quite accurate relative to the free-mode derivative (and for thickness derivatives, it is exactly accurate given the stiffness and mass proportionality of this derivative). Fixed-mode approximation 2 computes accurate derivatives for a forward-swept low-aspect ratio wing, but otherwise there are persistent gaps in the derivatives between approximation 1 and 2. These gaps become smaller as the wing is swept forward, and as the wing span is increased, but the % error (which is not shown here) remains on the order of 100%. Similarly, the difference between the two approximations, normalized by the flutter q_∞ (also not shown here) remains on the order of 1.0; this normalization is likely important for optimization-centric applications.

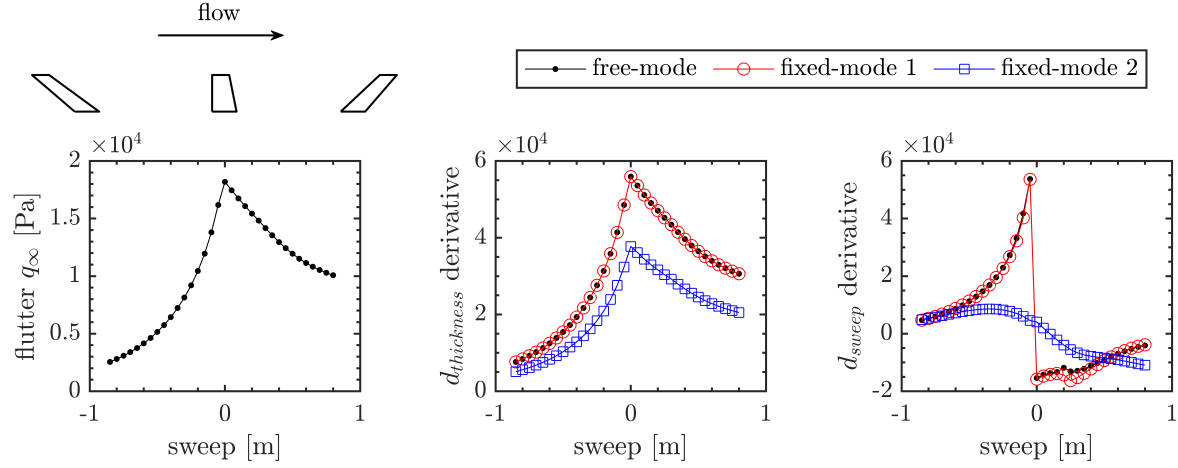


Figure 5. Flutter q_∞ (left), lumped thickness derivative (center), and sweep derivative (right), plotted as a function of the wing sweep (low-aspect ratio wing).

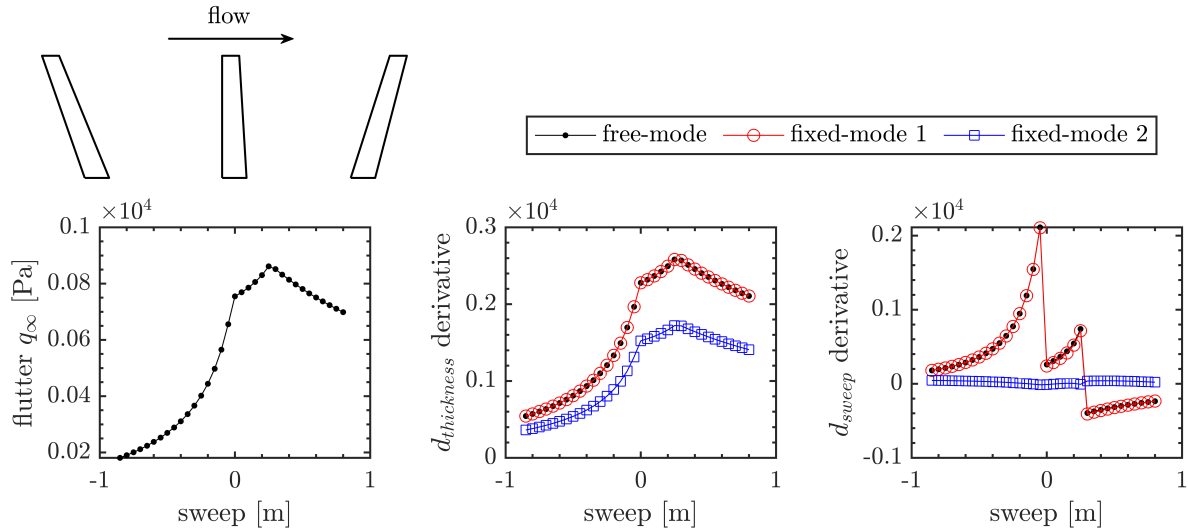


Figure 6. Flutter q_∞ (left), lumped thickness derivative (center), and sweep derivative (right), plotted as a function of the wing sweep (high-aspect ratio wing).

This design optimization is considered next, with and without the fixed-mode assumption. The objective function is to minimize structural mass, with a constraint that the planform area remain equal to the baseline value, and also a constraint that the flutter q_∞ not be less than the baseline value. The four shape design variables (root chord, tip chord, sweep, span) and 100 sizing variables (thickness values attached to each finite element) are included here, and the optimization convergence is shown in Fig. 7. Following previous results, the optimization driven by free-mode and fixed-mode approximation 1 derivatives are nearly indistinguishable, though the fixed-mode results are unable to satisfy the flutter constraint as thoroughly as the free-mode results. It should be noted here that the fixed-mode optimality shown in Fig. 7 use the incorrect fixed-mode derivatives; it would be possible to post-process each design in the figure for the exact free-mode derivative, and compute optimality based on those derivatives, but this has not been attempted.

The process in Fig. 7 using fixed-mode approximation 2 cannot converge: the final iterations of this process involve lengthy line-searches that are unable to improve the optimization (though only the outer iterations at the end of the line search are shown in the figure). A flutter optimization design space can in some cases possess discontinuities, owing to a potential switching of the critical flutter mechanism¹⁵ (seen in the sweeps of Figs. 5 and 6, for example), but this is not the case in Fig. 7: there is a single

flutter mechanism (coupling of modes 1 and 2) for every design in the figure. Therefore, the inability of the fixed-mode approximation 2 to converge is entirely due to inaccurate derivatives.

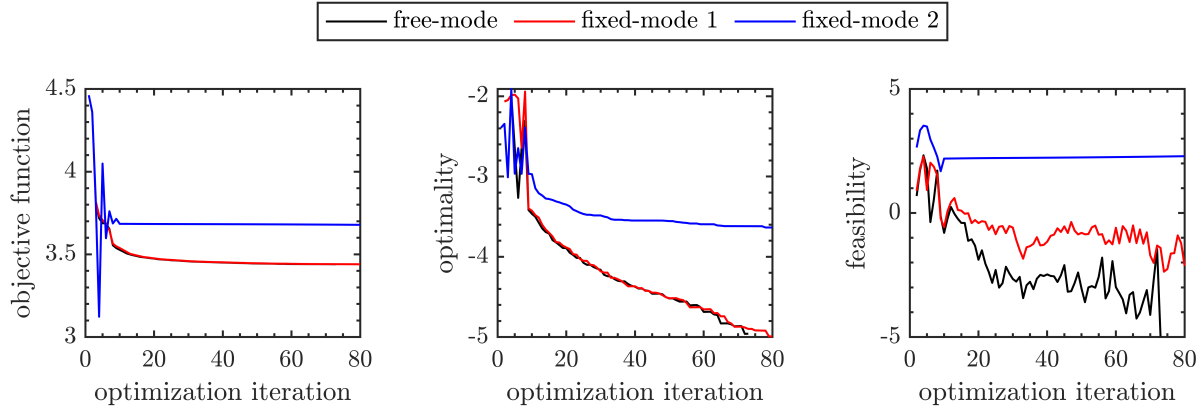


Figure 7. Optimization convergence history when both shape and sizing design variables are utilized.

The process is repeated with frozen shape design variables in Fig. 8, and only uses the 100 sizing design variables. The optimal objective function for this case is 4.095 kg (structural mass), which is inferior to the objective function in Fig. 7 (3.422 kg), owing to the reduced design space. Here, fixed-mode approximation 2 is better, as the flutter constraint is largely satisfied, though the optimization process converges to an inferior value (4.195 kg), relative to the case driven by exact free-mode derivatives. The baseline design, and the two optimal designs (as located with free-mode derivatives), are summarized in Fig. 9 in terms of the planform shape and structural thickness variables, as well as the the eigenvalue migration ($real(p)$ vs. q_∞). As dictated by the optimization process, all 3 designs have the same flutter value of 10071 Pa.

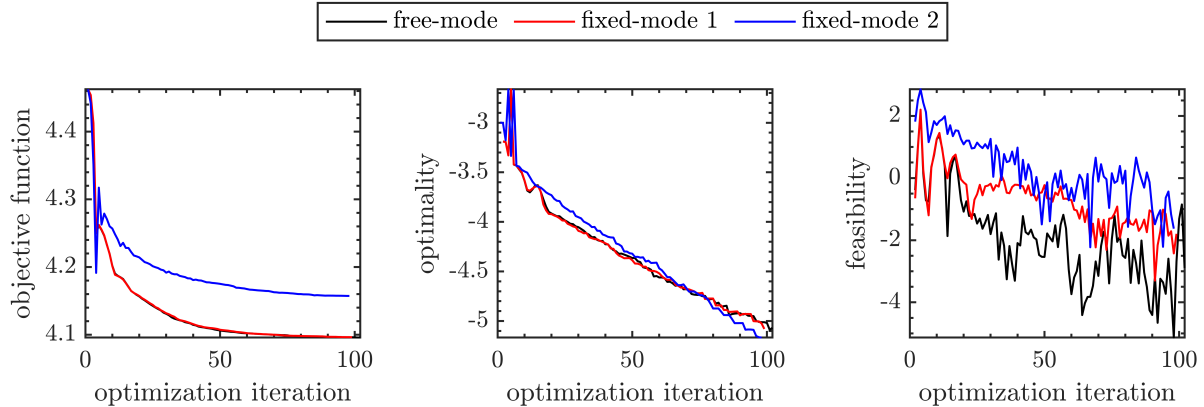


Figure 8. Optimization convergence history when only sizing design variables are utilized.

VI. Conclusions

Past literature (Ref. 1, for example) has speculated that a fixed-mode approximation for flutter-based optimization (where the derivatives of mode shapes are ignored in the gradient computation process) may be inadequate for shape design variables: i.e., the resulting inaccurate gradients may impede convergence of the gradient-based optimizer. A second hypothesis states that fixed-mode derivatives may be inaccurate for low-aspect ratio wings, but become more accurate for higher-aspect ratios, where the mode shapes become more beam-like and less dependent on design variables. Neither of these hypotheses are validated in the current work, where fixed-mode derivatives only show a small error relative to exact free-mode derivatives, for a range of design variable types and wing aspect ratios. Optimization convergence with free- and fixed-

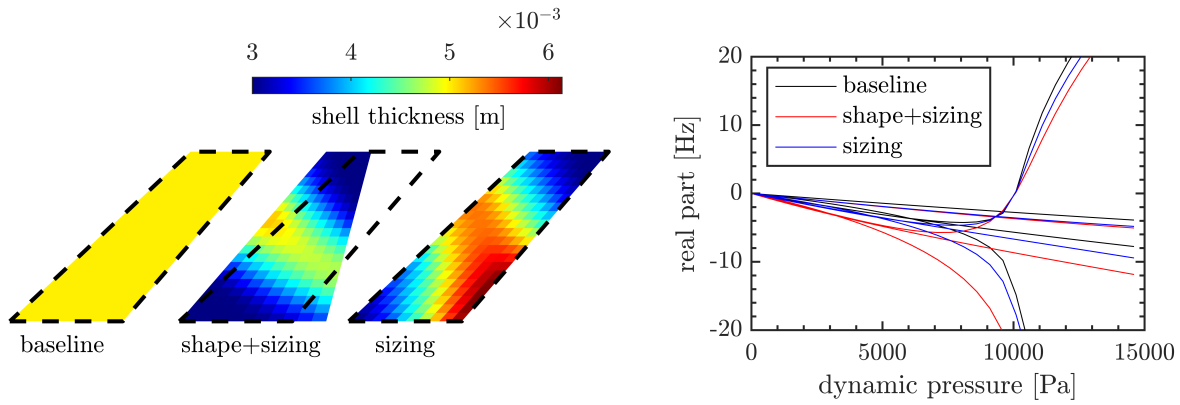


Figure 9. Optimal designs (left) and eigenvalue migration plots (right).

mode derivatives are very similar. This conclusion may be biased by the aeroelastic problem considered here: a linear plate-like wing structure in linear subsonic flow. Different aeroelastic configurations with different design parameterizations driven by different aeroelastic physics, may show a degraded fixed-mode accuracy.

These conclusions are true for the first fixed-mode approximation developed here: a second fixed-mode approximation (driven by a different formulation for the modally-reduced mass and stiffness matrix) does compute highly inaccurate derivatives. The computational cost of the two approximations is similar, and the only reason one may use approximation 2, is via a coding oversight: the different formulations for the modally-reduced matrices produce the exact same flutter mechanism, and the exact same free-mode derivative. Differences only arise for fixed-mode derivatives, and the existence of this gap is a subtle one that may not be obvious to inexperienced practitioners.

Finally, it is also noted that the computational cost of Nelson’s method for the problem considered here is roughly one-third of the total computational cost. Utilizing a fixed-mode derivative can then decrease cost considerably with only a small sacrifice in accuracy, as long as approximation 1 is used: though again, this conclusion may be biased by the aeroelastic scenario considered here. For higher-fidelity CFD-based aerodynamics, where more of the total cost is concentrated on the aerodynamics, structures-centric cost considerations become less important. Similarly, the existence of additional design constraints (such as a static aeroelastic constraint) can dilute the dependence of the optimal design upon the flutter constraint, again rendering fixed- and free-mode accuracy issues less important.

Acknowledgments

This work is supported by the Transformational Tools and Technologies (TTT) project of the NASA Transformative Aeronautics Concepts Program (TACP). Computational resources for this work are provided by the NASA Langley K cluster.

References

- ¹Jonsson, E., Riso, C., Lupp, C., Cesnik, C., Martins, J., Epureanu, B., “Flutter and Post-Flutter Constraints in Aircraft Design Optimization,” *Progress in Aerospace Sciences*, Vol. 109, 100537, 2019.
- ²Nelson, R., “Simplified Calculation of Eigenvector Derivatives,” *AIAA Journal*, Vol. 14, pp. 1201-1205, 1976.
- ³Sutter, T., Camarda, C., Walsh, J., Adelman, H., “Comparison of Several Methods for Calculating Vibration Mode Shape Derivatives,” *AIAA Journal*, Vol. 26, pp. 1506-1511, 1988.
- ⁴Lin, R., Mottershead, J., Ng, T., “A State-of-the-Art Review on Theory and Engineering Applications of Eigenvalue and Eigenvector Derivatives,” *Mechanical Systems and Signal Processing*, Vol. 138, 106536, 2020.
- ⁵Haftka, R., Yates, C., “Repetitive Flutter Calculations in Structural Design,” *Journal of Aircraft*, Vol. 13, No. 7, pp. 454-461, 1976.
- ⁶Chen, P., Zhang, Z., Livne, E., “Design-Oriented Computational Fluid Dynamics-Based Unsteady Aerodynamics for Flight-Vehicle Aeroelastic Shape Optimization,” *AIAA Journal*, Vol. 53, No. 12, pp. 3603-3619, 2017.
- ⁷Zhang, Z., Chen, P., Yang, S., Wang, Z., Wang, Q., “CFD-based Aeroelastic Adjoint Sensitivities for Flight Vehicle Weight Minimization using Sizing and/or Shape Design Variables,” *AIAA Structures, Structural Dynamics, and Materials Conference*, Grapevine, TX, January 2017.

- ⁸Jonsson, E., Mader, C., Martins, J., Kennedy, G., “Computational Modeling of Flutter Constraint for High-Fidelity Aerostructural Optimization,” *AIAA SciTech Forum*, San Diego, CA, January 2019.
- ⁹He, S., Jonsson, E., Mader, C., Martins, J., “Aerodynamic Shape Optimization with Time Spectral Flutter Adjoint,” *AIAA SciTech Forum*, San Diego, CA, January 2019.
- ¹⁰Sandridge, C., Haftka, R., “Accuracy of Derivatives of Control Performance using a Reduced Structural Model,” *AIAA Structures, Structural Dynamics, and Materials Conference*, Monterey, CA, April 1987.
- ¹¹Greene, W., Haftka, R., “Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis,” *Structural Optimization*, Vol. 3, pp. 176-201, 1991.
- ¹²Gray, J., Hwang, J., Martins, J., Moore, K., Naylor, B., “OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, pp. 1075-1104, 2019.
- ¹³Yildirim, A., Jacobson, K., Stanford, B., Gray, J., Mader, C., Martins, J., Kennedy, G., “A Modular Multiphysics Simulation Framework Using OpenMDAO,” *AIAA Aviation Forum*, virtual event, June 15-19, 2020.
- ¹⁴Yildirim, A., Jacobson, K., Anibal, J., Stanford, B., Gray, J., Mader, C., Martins, J., Kennedy, G., “MPhys: A Modular Multiphysics Simulation Library Using the OpenMDAO Framework,” in preparation.
- ¹⁵Ringertz, U., “On Structural Optimization with Aeroelasticity Constraints,” *Structural Optimization*, Vol. 8, pp. 16-23, 1994.
- ¹⁶Lambe, A., Martins, J., “Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes” *Structural and Multidisciplinary Optimization*, Vol. 46, pp. 273-284, 2012.
- ¹⁷Jacobson, K., Stanford, B., Wood, S., Anderson, W., “Frequency-Domain Flutter Analysis and Sensitivities with Stabilized Finite Elements,” *AIAA SciTech Forum*, Orlando, FL, January 6-10, 2020.
- ¹⁸Jacobson, K., Stanford, B., “Flutter-Constrained Optimization with the Linearized Frequency-Domain Approach” *AIAA Scitech Forum*, San Diego, CA, January 3-7, 2022.

A. Appendix: The Equivalence of Free- and Fixed-Mode Derivatives

If $\frac{\partial \mathbf{K}}{\partial \mathbf{s}} = k \cdot \mathbf{K}$, and $\frac{\partial \mathbf{M}}{\partial \mathbf{s}} = m \cdot \mathbf{M}$ (where k and m are scalar proportionality constants), then the fixed-mode derivative will give the same result as the free-mode derivative: the ϕ_n terms from Eq. 20 disappear. This idea is best-demonstrated with forward-mode direct derivatives, as opposed to the adjoint ideas used above. The eigenvalue derivatives are computed by differentiating all terms in Eq. 22 with respect to $\mathbf{s}$:

$$\frac{\partial \omega_n}{\partial \mathbf{s}} = \left(\phi_n^T \cdot \frac{\partial \mathbf{K}}{\partial \mathbf{s}} \cdot \phi_n - \omega_n^2 \cdot \phi_n^T \cdot \frac{\partial \mathbf{M}}{\partial \mathbf{s}} \cdot \phi_n \right) / 2\omega_n \quad (28)$$

If $\phi_n^T \cdot \mathbf{K} \cdot \phi_n = \omega_n^2$ and $\phi_n^T \cdot \mathbf{M} \cdot \phi_n = 1$, then $\frac{\partial \omega_n}{\partial \mathbf{s}} = \omega_n \cdot (k - m) / 2$. Using Nelson’s method to solve for the eigenvector derivative yields the simple result $\frac{\partial \phi_n}{\partial \mathbf{s}} = -\phi_n \cdot m / 2$. The key result here is that $\frac{\partial \phi_n}{\partial \mathbf{s}}$ is proportional to ϕ_n .

Next, Eq. 6 is expanded to accommodate a direct derivative:

$$\frac{d\mathbf{R}_u}{d\mathbf{s}} = \frac{\partial \mathbf{R}_u}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}} + \sum_{n=1}^N \frac{\partial \mathbf{R}_u}{\partial \omega_n} \cdot \frac{\partial \omega_n}{\partial \mathbf{s}} + \sum_{n=1}^N \frac{\partial \mathbf{R}_u}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial \mathbf{s}} + \frac{\partial \mathbf{R}_u}{\partial \mathbf{s}} = \mathbf{0} \quad (29)$$

$$\frac{dJ}{d\mathbf{s}} = \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{s}} \quad (30)$$

$$\frac{dJ}{d\mathbf{s}} = -\frac{\partial J}{\partial \mathbf{u}} \cdot \left(\frac{\partial \mathbf{R}_u}{\partial \mathbf{u}} \right)^{-1} \cdot \left(\frac{\partial \mathbf{R}_u}{\partial \mathbf{s}} + \sum_{n=1}^N \frac{\partial \mathbf{R}_u}{\partial \omega_n} \cdot \frac{\partial \omega_n}{\partial \mathbf{s}} + \sum_{n=1}^N \frac{\partial \mathbf{R}_u}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial \mathbf{s}} \right) \quad (31)$$

The residual vector, $\mathbf{R}_u$, defined in Eq. 6, is not explicitly dependent on the eigenvalues ω_n , and so $\frac{\partial \mathbf{R}_u}{\partial \omega_n}$ terms in Eq. 29 are $\mathbf{0}$. If $\frac{\partial \mathbf{R}_u}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial \mathbf{s}}$ can also be shown to be $\mathbf{0}$, then the dependence on ϕ_n disappears, and fixed-mode derivatives are equal to free-mode derivatives.

$\mathbf{R}_u$ in Eq. 6 depends on the real and imaginary parts of the eigenvalue p , and so $\frac{\partial p}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial \mathbf{s}}$ may be considered in place of $\frac{\partial \mathbf{R}_u}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial \mathbf{s}}$. Similar to Eq. 10:

$$\frac{\partial p}{\partial \phi_n} = \left(\bar{\mathbf{L}}^T \cdot \left(\frac{\partial \bar{\mathbf{B}}}{\partial \phi_n} - p \cdot \frac{\partial \bar{\mathbf{A}}}{\partial \phi_n} \right) \cdot \bar{\mathbf{R}} \right) / \left(\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}} \right) \quad (32)$$

Referencing the original aeroelastic eigensystem Eq. 1, the scalar f may be defined as:

$$f = \left(\bar{\mathbf{L}}^T \cdot (\bar{\mathbf{B}} - p \cdot \bar{\mathbf{A}}) \cdot \bar{\mathbf{R}} \right) / \left(\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}} \right) = 0 \quad (33)$$

At the flutter condition, f is always equal to 0. The final relationship of interest is then:

$$\frac{\partial p}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial s} = \frac{\partial f}{\partial \phi_n} \cdot \frac{\partial \phi_n}{\partial s} = -\frac{\partial f}{\partial \phi_n} \cdot \phi_n \cdot m/2 \quad (34)$$

Numerical experiments have shown that $\frac{\partial f}{\partial \phi_n}$ is always orthogonal to ϕ_n , and so the term in Eq. 34 is 0, and any dependence of Eq. 29 on ϕ_n disappears. It is again emphasized that this only occurs because $\frac{\partial \phi_n}{\partial s}$ is proportional to ϕ_n , which in turn is dependent on mass- and stiffness-proportionality outlined above.

B. Appendix: The Equivalence of the First and Second Fixed-Mode Approximations

For either version of the fixed-mode derivative, the final sensitivity of interest is given by Eq. 25. If $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are computed via Eq. 2 and Eq. 3 (approximation 1), then the residual vector, $\mathbf{R}_u$, defined in Eq. 6, is not explicitly dependent on the eigenvalues ω_n , and so $\frac{\partial \mathbf{R}_u}{\partial \omega_n}$ terms in Eq. 25 are $\mathbf{0}$. The complex eigenvectors may be expanded as $\bar{\mathbf{L}} = [\bar{\mathbf{l}}^T \ i \cdot k \cdot \bar{\mathbf{l}}^T]^T$ and $\bar{\mathbf{R}} = [\bar{\mathbf{r}}^T \ i \cdot k \cdot \bar{\mathbf{r}}^T]^T$. The eigenvalue derivative of Eq. 10 is then expanded as:

$$\frac{\partial p}{\partial s} = \left(\bar{\mathbf{l}}^T \cdot \left(-\left(\frac{k \cdot U_\infty}{b} \right)^2 \cdot \Phi_s^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \Phi_s + \Phi_s^T \cdot \frac{\partial \mathbf{K}}{\partial s} \cdot \Phi_s - q_\infty \cdot \frac{\partial \bar{\mathbf{Q}}(i \cdot k)}{\partial s} \right) \cdot \bar{\mathbf{r}} \right) / (\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}}) \quad (35)$$

Where the structural damping terms have been dropped for simplicity. Next, the residual derivative is computed via Eq. 11, and the final derivative of interest is computed as $\frac{dJ}{ds} = \boldsymbol{\lambda}_u^T \cdot \frac{\partial \mathbf{R}_u}{\partial s}$.

Alternatively, if $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ are computed via Eq. 26 and Eq. 27 (approximation 2), the residual vector now does become an explicit function of ω_n via the diagonal matrix $\boldsymbol{\Omega}$. The eigenvalue derivative of Eq. 10 with respect to frequency is:

$$\frac{\partial p}{\partial \omega_n} = -(\bar{\mathbf{l}}^T \cdot \bar{\mathbf{r}}) \cdot 2 \cdot \omega_n / (\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}}) \quad (36)$$

The eigenvalue derivative with respect to s now only depends on the GAFs:

$$\frac{\partial p}{\partial s} = \left(\bar{\mathbf{l}}^T \cdot \left(q_\infty \cdot \frac{\partial \bar{\mathbf{Q}}(i \cdot k)}{\partial s} \right) \cdot \bar{\mathbf{r}} \right) / (\bar{\mathbf{L}}^T \cdot \bar{\mathbf{A}} \cdot \bar{\mathbf{R}}) \quad (37)$$

Assembling these terms into Eq. 25, and comparing with derivatives computed via approximation 1, it can be seen that the GAF-based derivatives are identical between the two forms. The pertinent stiffness-based derivatives are:

$$\underbrace{\bar{\mathbf{l}}^T \cdot \left(\Phi_s^T \cdot \frac{\partial \mathbf{K}}{\partial s} \cdot \Phi_s \right) \cdot \bar{\mathbf{r}}}_{\text{approximation 1}} \iff \underbrace{(\bar{\mathbf{l}}^T \cdot \bar{\mathbf{r}}) \cdot \sum_{n=1}^N \left(\phi_n^T \cdot \frac{\partial \mathbf{K}}{\partial s} \cdot \phi_n \right)}_{\text{approximation 2}} \quad (38)$$

These two forms are equal only if $\Phi_s^T \cdot \frac{\partial \mathbf{K}}{\partial s} \cdot \Phi_s$ is a diagonal matrix. As noted above, this can occur for certain types of structural sizing variables, but not, in general, for shape variables. The pertinent mass-based derivatives are:

$$\underbrace{\bar{\mathbf{l}}^T \cdot \left(\left(\frac{k \cdot U_\infty}{b} \right)^2 \cdot \Phi_s^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \Phi_s \right) \cdot \bar{\mathbf{r}}}_{\text{approximation 1}} \iff \underbrace{(\bar{\mathbf{l}}^T \cdot \bar{\mathbf{r}}) \cdot \sum_{n=1}^N \left(\omega_n^2 \cdot \phi_n^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \phi_n \right)}_{\text{approximation 2}} \quad (39)$$

Similarly, these two forms are equal if $\Phi_s^T \cdot \frac{\partial \mathbf{M}}{\partial s} \cdot \Phi_s$ is a diagonal matrix, but a second requirement is that every natural frequency (ω_n) be equal to the flutter frequency ($k \cdot U_\infty / b$). This second requirement is practically impossible to meet for any realistic aeroelastic problem, and therefore fixed-mode approximations 1 and 2 will always compute distinct derivatives.