

Adjoint-based Sensitivities of Flutter Predictions based on the Linearized Frequency-domain Approach

Kevin E. Jacobson,^{*} Bret K. Stanford,[†] Stephen L. Wood,[‡] and W. Kyle Anderson[§]
NASA Langley Research Center, Hampton, VA, 23681

Flutter is a critical factor in designing and certifying aircraft. The linearized frequency-domain method offers a lower cost alternative to time-marching computational fluid dynamics for high-fidelity flutter analysis. In this work, adjoint-based sensitivities are added to a flutter analysis based on the linearized frequency-domain method to efficiently compute derivatives of flutter cost functions with respect to design variables or uncertain parameters. The derivation of the adjoint equations, which involve complications such as derivatives of a nonlinear generalized eigenvalue problem with complex-valued inputs and derivatives of the linearized Navier-Stokes equations, is provided. The implemented adjoint terms and derivatives are verified before demonstrating the approach for derivatives of flutter dynamic pressure with respect to Mach number for the AGARD 445.6 wing.

I. Introduction

Establishing the flutter boundaries of a vehicle is a difficult and time consuming task. Aeroelastic behaviors critical to aircraft design and analysis are often sensitive to changes in the vehicle properties or flight conditions. As observed in Ref. [1], even minute changes in leading edge geometry can affect transonic flutter predictions in models based on Computational Fluid Dynamics (CFD). Aircraft manufacturing processes introduce uncertainty that can be significantly larger than the geometry perturbations in Ref. [1]. Many aircraft have critical flutter behavior in the transonic regime where traditional tools based on linear aerodynamics are not reliable. Alternatives like time-domain CFD-based flutter models, wind tunnel experiments, and flight tests are much more expensive; therefore, many aircraft are still primarily designed and analyzed with flutter models based on linear aerodynamics. Insufficient understanding of the transonic behavior and parametric uncertainty leads aircraft manufacturers to incorporate large flutter safety factors to ensure safe vehicles. In the aerospace industry, there is strong interest in certification by analysis as a path toward reducing the cost of aircraft certification and modification. For aeroelastic certification based on computational models, the analysis needs to be accurate enough for the flight conditions, including transonic flows, and efficient enough to allow parametric uncertainty quantification.

As discussed in detail in Ref. [2], development of an efficient transonic flutter constraint remains an open problem in the area of multidisciplinary optimization. Time-domain CFD analyses and adjoint-based derivatives for flutter constraints have been demonstrated for some cases [3, 4] but remain cost-prohibitive for optimization of more complex problems. Several groups are exploring flutter constraints with nonlinear frequency-domain techniques [5–7], but these efforts have been limited to relatively simple problems as well. To perform Reynolds-averaged Navier-Stokes (RANS)-based flutter analysis at a lower computational cost, the linearized frequency-domain (LFD) method [8, 9] has been implemented in the Stabilized Finite-Element (SFE) solver of FUN3D [10]. The LFD method replaces nonlinear time-marching simulations that require thousands of time steps to predict flutter with tens or hundreds of complex-valued linear problems resulting in a more efficient flutter analysis [11].

To further reduce the computational cost of optimization and uncertainty quantification for flutter, this work implements adjoint-based sensitivities of flutter cost functions with respect to input parameters to the LFD process. The adjoint method efficiently computes derivatives of a cost function with respect to many input parameters [12, 13] and reduces the number of evaluation points to sufficiently populate a parameter space for uncertainty quantification. Adjoint solutions can also be used as a basis for estimating discretization errors [14, 15] and output-based adaptation [16, 17]. The adjoint method is frequently employed in gradient-based optimization for aeroelastic models [18, 19]. The LFD flutter analysis has some features that complicate the sensitivity analysis. The p-k flutter problem requires derivatives of a

^{*}Research Aerospace Engineer, Aeroelasticity Branch, MS 340, AIAA Member, kevin.e.jacobson@nasa.gov

[†]Research Aerospace Engineer, Aeroelasticity Branch, MS 340, AIAA Associate Fellow.

[‡]Research Scientist, Computational Aerosciences Branch, MS 128, AIAA Member.

[§]Senior Research Scientist, Computational Aerosciences Branch, MS 128, AIAA Associate Fellow.

generalized eigenvalue problem that does not satisfy the Cauchy-Riemann equations for complex differentiability, and the LFD residual contains linearizations of the flow governing equations, so the adjoint method requires second derivatives of the RANS equations. In this paper, the adjoint equations are derived, and the corresponding implementation in FUN3D/SFE is verified. The flutter sensitivity analysis is then demonstrated for the AGARD 445.6 wing [20].

II. Methodology

This section summarizes the LFD method that is described in more detail in Ref. [10].

A. Stabilized Finite Elements in FUN3D

FUN3D is a NASA Langley Research Center CFD code that has been applied to a wide range of aerospace analysis and design problems [13, 21–23]. Recently [24, 25], a streamline-upwind Petrov-Galerkin finite-element solver called FUN3D/SFE has been added. As discussed in more detail in Ref. [24], the reasons for adding the alternate discretization to FUN3D are 1) the finite elements offer a viable path to higher-order CFD, 2) it is easier to maintain linearizations for adjoint-based design and strong solvers due to the smaller stencil, and 3) it provides improved accuracy on tetrahedral meshes. For the frequency-domain analysis presented in this paper, the underlying governing equations are the RANS equations with the negative Spalart-Allmaras turbulence model [26]. To account for mesh movement, these equations are solved using an arbitrary Lagrangian-Eulerian formulation [25]:

$$\int_{\Omega(t)} (N + P) [\dot{\mathbf{q}} + \nabla \cdot (\mathbf{F} - \mathbf{q} \dot{\mathbf{x}}_G) + \mathbf{q} \nabla \cdot \dot{\mathbf{x}}_G - \mathbf{S}] d\Omega = 0, \quad (1)$$

where N is the Galerkin weighting function, P is the Petrov-Galerkin weighting function described in Ref. [24], $\mathbf{q}$ is the flow state vector, $\mathbf{x}_G$ is the vector of volume mesh coordinates, $\mathbf{F}$ is the flux vector, $\mathbf{S}$ is the source term, and Newton notation for time derivatives is used. For aeroelastic analysis, dynamic grids in FUN3D/SFE are driven by the same mesh motion infrastructure as the finite-volume flow solver in FUN3D; therefore, the finite-element solver can be coupled to FUN3D's internal modal structural solver.

For the LFD method, the governing equations of the flow, Eq. 1, are rewritten as:

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_G, \dot{\mathbf{x}}_G) := \mathbf{M}_A(\mathbf{q}, \mathbf{x}_G, \dot{\mathbf{x}}_G) \dot{\mathbf{q}} + \mathbf{R}(\mathbf{q}, \mathbf{x}_G, \dot{\mathbf{x}}_G) = 0, \quad (2)$$

where $\mathbf{M}_A$ is the mass matrix, which is a function of the state due to use of the Petrov-Galerkin weighting function, and $\mathbf{R}$ is the spatial residual. The steady flow residual is a simplification with the time derivatives set to zero:

$$\mathbf{A}_0(\mathbf{q}_0, \mathbf{x}_{G0}) := \mathbf{R}(\mathbf{q}_0, \mathbf{x}_{G0}) = 0. \quad (3)$$

B. Mesh Deformation

For aeroelastic bodies, FUN3D uses a mesh deformation approach based on a linear elasticity model to propagate surface motion into the volume mesh [13, 22]. The discretized linear elasticity equation is:

$$\mathbf{G}_0(\mathbf{x}_{G0}, \mathbf{x}_{a0}) := \mathbf{K}_G(\mathbf{x}_{G0} - \bar{\mathbf{x}}_G) - (\mathbf{x}_{a0} - \bar{\mathbf{x}}_a) = 0, \quad (4)$$

where $\mathbf{K}_G$ is the mesh deformation stiffness matrix, $\mathbf{x}_{a0}$ is the deformed aerodynamic surface coordinates, and the $\bar{\mathbf{x}}_G$ and $\bar{\mathbf{x}}_a$ are reference volume and surface coordinates, respectively. The stiffness matrix for the pseudo-elasticity problem can be formed based on the wall distance or the cell volume in FUN3D.

C. Linearized Frequency-domain Method

For the LFD method, the time-dependent mesh and solution vectors in Eq. 2 are assumed to be the superposition of an equilibrium state and small, harmonic perturbations due to motion of structural modes, n , at frequency ω_k :

$$\mathbf{x}_G(t) = \mathbf{x}_{G0} + \sum_k \sum_n \alpha_k \hat{\mathbf{x}}_{G,n} e^{i\omega_k t}, \quad (5)$$

$$\mathbf{q}(t) = \mathbf{q}_0 + \sum_k \sum_n \alpha_k \hat{\mathbf{q}}_{n,k} e^{i\omega_k t}, \quad (6)$$

where $\hat{\mathbf{x}}_{G,n}$ is a spatially varying displacement field corresponding to unit displacements of the structural mode n , α_k is an amplitude of motion, and $\hat{\mathbf{q}}_{n,k}$ is the Fourier coefficient for the flow response to motion of mode n at frequency ω_k with a unit amplitude. The governing equations of the flow are linearized about the equilibrium solution and simplified to:

$$\mathbf{A}_{LFD,n,k}(\mathbf{x}_{G0}, \mathbf{q}_0, \hat{\mathbf{x}}_{G,n}, \hat{\mathbf{q}}_{n,k}) := \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0}, \mathbf{q}_0} + \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \right) \hat{\mathbf{q}}_{n,k} + \frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{x}}_{G,n} + i\omega_k \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{x}}_{G,n} = 0. \quad (7)$$

This linear system is solved once for each combination of frequency of interest (ω_k) and mesh motion input ($\hat{\mathbf{x}}_{G,n}$), which is computed by propagating the mode shapes of the structure through the CFD volume mesh motion process. The elasticity-based volume mesh deformation process from Eq. 4 is reused here:

$$\mathbf{G}_{LFD,n}(\hat{\mathbf{x}}_{G,n}, \Phi_{a,n}) := \mathbf{K}_G \hat{\mathbf{x}}_{G,n} - \Phi_{a,n} = 0, \quad (8)$$

where $\Phi_{a,n}$ is the n^{th} -mode shape projected onto the aerodynamic surface. The linear system in Eq. 7 is solved with a generalized minimum residual method [27] implemented in the Sparse Linear Algebra Toolkit (SLAT) [28]. From the resulting Fourier coefficients $\hat{\mathbf{q}}_{n,k}$ and $\hat{\mathbf{x}}_{G,n}$, Fourier coefficients of surface force perturbations are computed as:

$$\hat{\mathbf{f}}_{n,k} = \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{q}}_{n,k} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{x}}_{G,n} + i\omega_k \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{x}}_{G,n}, \quad (9)$$

where $\mathbf{f}$ is the function to calculate the aerodynamic force distribution. The dependence on grid velocity is necessary for the weak boundary condition for viscous surfaces in FUN3D/SFE [24].

The generalized aerodynamic forces (GAFs) can then be calculated by projecting the surface force perturbations onto the structural mode shapes using the load transfer scheme. Alternatively, they can be computed using the mode shapes that have been projected onto the aerodynamic surface, $\Phi_{a,m}$:

$$Q_{m,n}(i\omega_k) = \Phi_{a,m}^T \hat{\mathbf{f}}_{n,k}, \quad (10)$$

where $Q_{m,n}(i\omega_k)$ is the Fourier coefficient representing the GAF on mode m due to harmonic motion of mode n at frequency ω_k . The linearized force integration residual is defined as:

$$\mathbf{F}_{m,n,k}(Q_{m,n}(i\omega_k), \mathbf{x}_{G0}, \mathbf{q}_0, \hat{\mathbf{x}}_{G,n}, \hat{\mathbf{q}}_{n,k}) = \Phi_{a,m}^T \hat{\mathbf{f}}_{n,k} - Q_{m,n}(i\omega_k) = 0. \quad (11)$$

The matrix of GAFs is typically written as $\bar{\mathbf{Q}}(ik)$ where $k = \omega b/U_\infty$ is the reduced frequency, b is the reference semichord, and U_∞ is the freestream velocity. Equations 7-10 are solved for a specified list of frequencies of interest to form a full set of GAFs. When GAFs are required at intermediate frequencies in the p-k flutter analysis, they are interpolated from the frequencies at which the LFD problems were solved.

D. P-k Flutter Analysis Process

GAFs generated by the LFD method can be utilized in various analyses. In this section, their use in a p-k flutter method is discussed. See Jacobson et al. [10] for more details. First, the static aeroelastic problem is solved with FUN3D/SFE coupled to the structural solver. However, the problem is assumed symmetric, i.e., no static aeroelastic deformation, while focusing on the LFD sensitivity terms in this paper; the extension to static aeroelastic problems with nonzero deflections is discussed in Section V. Next, the frequency-domain problem is solved and linearized about the static aeroelastic state, $q_\infty, \mathbf{x}_{G,0}, \mathbf{q}_0$ where q_∞ represents the freestream dynamic pressure. Eq. 7 is solved at selected frequencies, ω , that cover the possible range of flutter frequencies, typically 10-15 total frequencies. After solving Eq. 7, the GAFs are found from Eqs. 9 and 10. The GAFs are then passed to a p-k flutter solver. In the p-k method, a flutter point is computed by solving the eigenvalue problem associated with the aeroelastic system. Writing the system in a nondimensional state-space form leads to a nonlinear generalized eigenproblem:

$$(\bar{\mathbf{A}} p - \bar{\mathbf{B}}(ik)) \bar{\mathbf{R}} = 0, \quad (12)$$

where p is the nondimensional eigenvalue, $p := bs/U_\infty = g + ik$, s is the Laplacian, and $\bar{\mathbf{R}}$ is the corresponding right eigenvector. The two matrices in the eigenproblem are defined as:

$$\bar{\mathbf{A}} := \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \left(\frac{U_\infty}{b}\right)^2 \bar{\mathbf{M}} \end{bmatrix}, \quad (13)$$

and

$$\bar{\mathbf{B}}(ik) := \begin{bmatrix} 0 & \mathbf{I} \\ -\bar{\mathbf{K}} + q_\infty \bar{\mathbf{Q}}(ik) & -\left(\frac{U_\infty}{b}\right) \bar{\mathbf{C}} \end{bmatrix}, \quad (14)$$

where $\bar{\mathbf{M}}$, $\bar{\mathbf{C}}$, and $\bar{\mathbf{K}}$ are the modally-reduced structural mass, damping, and stiffness matrices, respectively.

A flutter dynamic pressure is found as a q_∞ value that has a zero real part in the resulting eigenvalue: $\Re(p) = g = 0$. These conditions are met when the following residual is driven to 0:

$$\mathbf{R}_f := \begin{Bmatrix} \Re(p) \\ k - \Im(p) \end{Bmatrix} = 0. \quad (15)$$

The corresponding solution vector used to solve Eq. 15 is:

$$\mathbf{u}_f := \begin{Bmatrix} q_\infty \\ k \end{Bmatrix}. \quad (16)$$

III. Adjoint Sensitivities with the Flutter Cost Functions

Flutter cost functions are real-valued outputs like flutter dynamic pressure or flutter frequency, but the GAF inputs to the p-k solver are complex. The cost functions do not satisfy the Cauchy-Riemann equations for complex differentiation. Therefore, approaches used to derive real-valued adjoint equations need to be modified to account for the fact that the cost function sensitivity with respect to the complex-valued state is not defined. To simplify the presentation, a residual encapsulating the entire process up to the GAF computation is defined:

$$\mathbf{W}^T := \left[\mathbf{A}_0^T, \mathbf{G}_0^T, \mathbf{G}_{LFD,1}^T, \dots, \mathbf{G}_{LFD,n_m}^T, \mathbf{A}_{LFD,1,1}^T, \dots, \mathbf{A}_{LFD,n_m,n_f}^T, \mathbf{F}_{1,1,1}^T, \dots, \mathbf{F}_{n_m,n_m,n_f}^T \right]^T = 0, \quad (17)$$

and a state vector $\mathbf{y}$,

$$\mathbf{y}^T := \left[\mathbf{q}_0^T, \mathbf{x}_{G0}^T, \hat{\mathbf{x}}_{G,1}^T, \dots, \hat{\mathbf{x}}_{G,n_m}^T, \hat{\mathbf{q}}_{1,1}^T, \dots, \hat{\mathbf{q}}_{n_m,n_f}^T, Q_{1,1}(i\omega_1), \dots, Q_{n_m,n_m}(i\omega_{n_f}) \right]^T. \quad (18)$$

Following a similar approach to Ref. [29], the complex-valued state vector can be treated as two real-valued variables, $\mathbf{a}$ and $\mathbf{b}$ where $\mathbf{y} = \mathbf{a} + i\mathbf{b}$. With this form of the state vector, the total derivative of the cost function with respect to a design variable is:

$$\begin{aligned} \frac{df}{dx} &= \frac{\partial f}{\partial x} + \frac{\partial f}{\partial \mathbf{a}} \frac{d\mathbf{a}}{dx} + \frac{\partial f}{\partial \mathbf{b}} \frac{d\mathbf{b}}{dx} \\ &= \frac{\partial f}{\partial x} + \Re \left[\left(\frac{\partial f}{\partial \mathbf{a}} - i \frac{\partial f}{\partial \mathbf{b}} \right) \left(\frac{d\mathbf{a}}{dx} + i \frac{d\mathbf{b}}{dx} \right) \right] \\ &= \frac{\partial f}{\partial x} + \Re \left[\left(\frac{\partial f}{\partial \mathbf{a}} - i \frac{\partial f}{\partial \mathbf{b}} \right) \frac{d\mathbf{y}}{dx} \right]. \end{aligned} \quad (19)$$

Differentiating the residual with respect to the design variable:

$$\frac{d\mathbf{W}}{dx} = \frac{\partial \mathbf{W}}{\partial x} + \frac{\partial \mathbf{W}}{\partial \mathbf{y}} \frac{d\mathbf{y}}{dx} = 0. \quad (20)$$

Rearranging,

$$\frac{d\mathbf{y}}{dx} = - \left[\frac{\partial \mathbf{W}}{\partial \mathbf{y}} \right]^{-1} \frac{\partial \mathbf{W}}{\partial x}. \quad (21)$$

Substituting into Eq. 19:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \Re \left[- \left(\frac{\partial f}{\partial \mathbf{a}} - i \frac{\partial f}{\partial \mathbf{b}} \right) \left[\frac{\partial \mathbf{W}}{\partial \mathbf{y}} \right]^{-1} \frac{\partial \mathbf{W}}{\partial x} \right]. \quad (22)$$

The adjoint vector is defined as:

$$\psi^T = - \left(\frac{\partial f}{\partial \mathbf{a}} - i \frac{\partial f}{\partial \mathbf{b}} \right) \left[\frac{\partial \mathbf{W}}{\partial \mathbf{y}} \right]^{-1}, \quad (23)$$

which results in the adjoint equation:

$$\left[\frac{\partial \mathbf{W}}{\partial \mathbf{y}} \right]^T \psi = - \left(\frac{\partial f}{\partial \mathbf{a}} - i \frac{\partial f}{\partial \mathbf{b}} \right)^T, \quad (24)$$

and a final expression for the design variable derivatives:

$$\frac{df}{dx} = \frac{\partial f}{\partial x} + \Re \left[\psi^T \frac{\partial \mathbf{W}}{\partial x} \right]. \quad (25)$$

This adjoint equation and design variable derivative resemble the real-valued versions except for the cost function term in Eq. 24 and the real part operator in Eq. 25.

A. Discipline Level Adjoint Equations

Defining the general adjoint vector as a column of adjoint vectors associated with the various governing residuals:

$$\psi^T := \left[\psi_{A0}^T, \psi_{G0}^T, \psi_{GLFD,1}^T, \dots, \psi_{GLFD,n_m}^T, \psi_{ALFD,1,1}^T, \dots, \psi_{ALFD,n_m,n_f}^T, \psi_{F,1,1,1}^T, \dots, \psi_{F,n_m,n_m,n_f}^T \right]^T, \quad (26)$$

the rows of Eq. 24 can be written as a set of adjoint equations where each equation is associated with a discipline in $\mathbf{W}$:

$$\left[\frac{\partial \mathbf{F}_{m,n,k}}{\partial Q_{m,n}} \right]^T \psi_{F_{m,n,k}} = - \left[\frac{\partial f}{\partial (\Re(Q_{m,n}))} - i \frac{\partial f}{\partial (\Im(Q_{m,n}))} \right]^T \quad \text{for } m, n = 1, \dots, n_m, \quad \text{and } k = 1, \dots, n_f, \quad (27)$$

$$\left[\frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \hat{\mathbf{q}}_{n,k}} \right]^T \psi_{A_{LFD,n,k}} = - \sum_{m=1}^{n_m} \left[\frac{\partial \mathbf{F}_{m,n,k}}{\partial \hat{\mathbf{q}}_{n,k}} \right]^T \psi_{F_{m,n,k}} \quad \text{for } n = 1, \dots, n_m, \quad \text{and } k = 1, \dots, n_f, \quad (28)$$

$$\left[\frac{\partial \mathbf{G}_{LFD,n}}{\partial \hat{\mathbf{x}}_{G,n}} \right]^T \psi_{G_{LFD,n}} = - \sum_{m=1}^{n_m} \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{F}_{m,n,k}}{\partial \hat{\mathbf{x}}_{G,n}} \right]^T \psi_{F_{m,n,k}} - \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \hat{\mathbf{x}}_{G,n}} \right]^T \psi_{A_{LFD,n,k}} \quad \text{for } n = 1, \dots, n_m, \quad (29)$$

$$\left[\frac{\partial \mathbf{A}_0}{\partial \mathbf{q}_0} \right]^T \psi_{A0} = - \sum_{m=1}^{n_m} \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{F}_{m,n,k}}{\partial \mathbf{q}_0} \right]^T \psi_{F_{m,n,k}} - \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \mathbf{q}_0} \right]^T \psi_{A_{LFD,n,k}}, \quad (30)$$

$$\left[\frac{\partial \mathbf{G}_0}{\partial \mathbf{x}_{G0}} \right]^T \psi_{G0} = - \left[\frac{\partial \mathbf{A}_0}{\partial \mathbf{x}_{G0}} \right]^T \psi_{A0} - \sum_{m=1}^{n_m} \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{F}_{m,n,k}}{\partial \mathbf{x}_{G0}} \right]^T \psi_{F_{m,n,k}} - \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \left[\frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \mathbf{x}_{G0}} \right]^T \psi_{A_{LFD,n,k}}. \quad (31)$$

The parenthetical $(i\omega_k)$ has been dropped for the GAFs in Eq. 27 for ease of presentation. For the LFD adjoint equation, Eq. 28, the left hand side is the transpose of the left hand side of the forward linear problem. In Eq. 29, the left hand side is the transpose of the mesh deformation stiffness matrix $\mathbf{K}_G$, and the term on the right hand side related to the LFD residual is:

$$\left[\frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \hat{\mathbf{x}}_{G,n}} \right]^T \psi_{A_{LFD,n,k}} = \frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \bigg|_{\mathbf{x}_{G0}, \mathbf{q}_0}^T \psi_{A_{LFD,n,k}} + i\omega_k \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}_G} \bigg|_{\mathbf{x}_{G0}, \mathbf{q}_0}^T \psi_{A_{LFD,n,k}}. \quad (32)$$

These terms are the transpose of the right hand side of the original LFD linear problem. In FUN3D/SFE, these terms are computed as matrix-free transposed products and are accumulated over of the number of LFD frequencies to form the full term in Eq. 29. The LFD term in Eq. 30 is more complicated. Each of the linearizations in the LFD residual are

a function of $\mathbf{q}_0$:

$$\begin{aligned}
\left[\frac{\partial \mathbf{A}_{LFD}}{\partial \mathbf{q}_0} \right]^T \psi_{A_{LFD},n,k} &= \left[\frac{\partial}{\partial \mathbf{q}} \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\
&+ \left[\frac{\partial}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\
&+ \left[\frac{\partial}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k} \\
&+ \left[\frac{\partial}{\partial \mathbf{q}} \left(i\omega_k \frac{\partial \mathbf{R}}{\partial \hat{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k}.
\end{aligned} \tag{33}$$

Using $\mathbf{M}_A = \frac{\partial \mathbf{A}}{\partial \dot{\mathbf{q}}}$, the first term can be rewritten and the order of the partial derivatives can be reversed:

$$\begin{aligned}
\left[\frac{\partial}{\partial \mathbf{q}} \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} &= i\omega_k \left[\frac{\partial}{\partial \mathbf{q}} \left(\frac{\partial \mathbf{A}}{\partial \dot{\mathbf{q}}} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \right) \hat{\mathbf{q}}_{n,k} \right]^T \psi_{A_{LFD},n,k} \\
&= i\omega_k \frac{\partial}{\partial \dot{\mathbf{q}}} \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0} \psi_{A_{LFD},n,k} \right) \hat{\mathbf{q}}_{n,k}.
\end{aligned} \tag{34}$$

The expression is the form of a $\dot{\mathbf{q}}$ directional derivative of the product in parentheses in the direction of $\hat{\mathbf{q}}_{n,k}$. After splitting the transposed Jacobian vector product into a sum of four real-valued matrix-vector products by splitting the real and imaginary parts of $\hat{\mathbf{q}}_{n,k}$ and $\psi_{A_{LFD},n,k}$, complex-step directional derivatives are used to achieve machine precision accuracy of this term and avoid questions about subtractive cancellation error due to the choice of step size:

$$\begin{aligned}
i\omega_k \frac{\partial}{\partial \dot{\mathbf{q}}} \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0} \psi_{A_{LFD},n,k} \right) \hat{\mathbf{q}}_{n,k} &= i\omega_k \left[\frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0, \dot{\mathbf{q}}=ih \Re(\hat{\mathbf{q}}_{n,k})} \Re(\psi_{A_{LFD},n,k}) \right)}{h} \right. \\
&- \frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0, \dot{\mathbf{q}}=ih \Im(\hat{\mathbf{q}}_{n,k})} \Im(\psi_{A_{LFD},n,k}) \right)}{h} \\
&+ i \frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0, \dot{\mathbf{q}}=ih \Re(\hat{\mathbf{q}}_{n,k})} \Re(\psi_{A_{LFD},n,k}) \right)}{h} \\
&\left. + i \frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0, \dot{\mathbf{q}}=ih \Im(\hat{\mathbf{q}}_{n,k})} \Im(\psi_{A_{LFD},n,k}) \right)}{h} \right].
\end{aligned} \tag{35}$$

Since the steady flow problem is real-valued and Eq. 25 will take the real part, the actual implementation does not require the first two terms in square brackets. Therefore, only the final two directional derivatives are needed, and the cost is the same as a standard finite difference directional derivative yet accurate to machine precision while avoiding analytically computing the second derivative. The other terms in Eq. 33 are handled the same way. For example, the second term in Eq. 33 is computed as:

$$\begin{aligned}
\Re \left(\frac{\partial}{\partial \mathbf{q}} \left(\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}_0} \psi_{A_{LFD},n,k} \right) \hat{\mathbf{q}}_{n,k} \right) &= \frac{\Im \left(\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}=\mathbf{q}_0+ih \Re(\hat{\mathbf{q}}_{n,k})} \Re(\psi_{A_{LFD},n,k}) \right)}{h} \\
&- \frac{\Im \left(\left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right]^T_{\mathbf{x}_{G0},\mathbf{q}=\mathbf{q}_0+ih \Im(\hat{\mathbf{q}}_{n,k})} \Im(\psi_{A_{LFD},n,k}) \right)}{h}.
\end{aligned} \tag{36}$$

The LFD term in the grid adjoint equation, Eq. 31, and the linearizations of $\mathbf{F}_{m,n,k}$ require similar second order linearizations:

$$\begin{aligned} \left[\frac{\partial \mathbf{A}_{LFD}}{\partial \mathbf{x}_{G0}} \right]^T \psi_{A_{LFD},n,k} &= \left[\frac{\partial}{\partial \mathbf{x}_G} \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}_G} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}_G} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}_G} \left(i\omega_k \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k}, \end{aligned} \quad (37)$$

and

$$\begin{aligned} \left[\frac{\partial \mathbf{F}}{\partial \mathbf{q}_0} \right]^T \psi_{F_{m,n,k}} &= \left[\frac{\partial}{\partial \mathbf{q}} \left(\Phi_{a,m}^T \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{F_{m,n,k}} \\ &+ \left[\frac{\partial}{\partial \mathbf{q}} \left(\Phi_{a,m}^T \frac{\partial \mathbf{f}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{F_{m,n,k}} \\ &+ \left[\frac{\partial}{\partial \mathbf{q}} \left(i\omega_k \Phi_{a,m}^T \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{F_{m,n,k}}, \end{aligned} \quad (38)$$

The manipulation to move the transposed linearization to the inside is applied to allow complex-step directional derivatives for these terms as well.

B. Design Variable Derivatives

Like the adjoint equation, Eq. 25 can be written in terms of the disciplines' adjoint variables and residual partial derivatives:

$$\begin{aligned} \frac{df}{d\mathbf{x}} &= \frac{\partial f}{\partial \mathbf{x}} + \Re \left[\psi_{G0}^T \frac{\partial \mathbf{G}_0}{\partial \mathbf{x}} + \psi_{A0}^T \frac{\partial \mathbf{A}_0}{\partial \mathbf{x}} + \sum_{n=1}^{n_m} \psi_{G_{LFD},n}^T \frac{\partial \mathbf{G}_{LFD,n}}{\partial \mathbf{x}} \right. \\ &\quad \left. + \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \psi_{A_{LFD},n,k}^T \frac{\partial \mathbf{A}_{LFD,n,k}}{\partial \mathbf{x}} + \sum_{m=1}^{n_m} \sum_{n=1}^{n_m} \sum_{k=1}^{n_f} \psi_{F_{m,n,k}}^T \frac{\partial \mathbf{F}_{m,n,k}}{\partial \mathbf{x}} \right]. \end{aligned} \quad (39)$$

The steady grid and flow terms are the same terms from the steady aerodynamic adjoint problem. The $G_{LFD,n}$ residual is not an explicit function of any design variables. Writing the LFD term as the transpose to match the form of the LFD terms in the steady grid and flow adjoint equations:

$$\begin{aligned} \left\{ \psi_{A_{LFD},n,k}^T \left[\frac{\partial \mathbf{A}_{LFD}}{\partial \mathbf{x}} \right] \right\}^T &= \left[\frac{\partial \mathbf{A}_{LFD}}{\partial \mathbf{x}} \right]^T \psi_{A_{LFD},n,k} \\ &= \left[\frac{\partial}{\partial \mathbf{x}} \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{q}} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}} \left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k} \\ &+ \left[\frac{\partial}{\partial \mathbf{x}} \left(i\omega_k \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}_G} \Big|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} \right) \right]^T \psi_{A_{LFD},n,k}. \end{aligned} \quad (40)$$

the same complex step technique can be utilized. For example, the mass matrix contribution is computed as:

$$\begin{aligned} \Re \left(\left[\frac{\partial}{\partial \mathbf{x}} \left(i\omega_k \mathbf{M}_A|_{\mathbf{x}_{G0}, \mathbf{q}_0} \hat{\mathbf{q}}_{n,k} \right) \right]^T \psi_{ALFD, n, k} \right) &= \Re \left(i\omega_k \frac{\partial}{\partial \hat{\mathbf{q}}} \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}} \right]_{\mathbf{x}_{G0}, \mathbf{q}_0}^T \psi_{ALFD, n, k} \right) \hat{\mathbf{q}}_{n,k} \right) \\ &= i\omega_k \left[i \frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}} \right]_{\mathbf{x}_{G0}, \mathbf{q}_0, \hat{\mathbf{q}}=ih}^T \Im(\hat{\mathbf{q}}_{n,k}) \Re(\psi_{ALFD, n, k}) \right)}{h} \right. \\ &\quad \left. + i \frac{\Im \left(\left[\frac{\partial \mathbf{A}}{\partial \mathbf{x}} \right]_{\mathbf{x}_{G0}, \mathbf{q}_0, \hat{\mathbf{q}}=ih}^T \Re(\hat{\mathbf{q}}_{n,k}) \Im(\psi_{ALFD, n, k}) \right)}{h} \right]. \end{aligned} \quad (41)$$

As with the steady adjoint equations, the imaginary part does not need to be computed because of the real part operator in Eq. 40. Since nothing has been introduced by the complex step directional derivative approach that scales with the number of design variables, this method maintains the adjoint technique's good scalability with respect to the number of design variables.

C. Linearization of the p-k Flutter Cost Function

The derivative of the cost function with respect to the real and imaginary parts of the GAFs must be found for the right hand side of Eq. 27. Having solved Eqs. 15 and 16, the cost function in this work is the resulting flutter dynamic pressure:

$$f = \mathbf{a}^T \mathbf{u}_f, \quad (42)$$

where $\mathbf{a}^T = [1, 0]$. The required semitotal derivative for the real part of the GAF can be written as:

$$\frac{df}{\partial (\Re(Q_{m,n}))} = \frac{\partial f}{\partial (\Re(Q_{m,n}))} + \frac{\partial f}{\partial \mathbf{u}_f} \frac{\partial \mathbf{u}_f}{\partial (\Re(Q_{m,n}))}, \quad (43)$$

where the partial derivative of the dynamic pressure with respect to the real part of the GAF is zero. The flutter residual definition can be used for the second term of this expression:

$$\frac{d\mathbf{R}_f}{\partial (\Re(Q_{m,n}))} = \frac{\partial \mathbf{R}_f}{\partial (\Re(Q_{m,n}))} + \frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} \frac{\partial \mathbf{u}_f}{\partial \Re(Q_{m,n})} = 0, \quad (44)$$

$$\frac{\partial \mathbf{u}_f}{\partial \Re(Q_{m,n})} = - \left[\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} \right]^{-1} \frac{\partial \mathbf{R}_f}{\partial (\Re(Q_{m,n}))}. \quad (45)$$

Substituting into Eq. 43,

$$\frac{df}{\partial (\Re(Q_{m,n}))} = - \frac{\partial f}{\partial \mathbf{u}_f} \left[\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} \right]^{-1} \frac{\partial \mathbf{R}_f}{\partial (\Re(Q_{m,n}))}, \quad (46)$$

similar to the primary adjoint, an adjoint problem can be formed. Defining the adjoint vector:

$$\psi_f^T = - \frac{\partial f}{\partial \mathbf{u}_f} \left[\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} \right]^{-1}, \quad (47)$$

the adjoint equation is then:

$$\left[\frac{\partial \mathbf{R}_f}{\partial \mathbf{u}_f} \right]^T \psi_f = - \frac{\partial f}{\partial \mathbf{u}_f}^T, \quad (48)$$

and the GAF derivative expression is:

$$\frac{df}{\partial (\Re(Q_{m,n}))} = \psi_f^T \frac{\partial \mathbf{R}_f}{\partial (\Re(Q_{m,n}))}. \quad (49)$$

Table 1 Verification of derivatives of flutter dynamic pressure with respect to GAFs, for the AGARD 445.6 wing.

	Real part of $Q_{1,2}$	Imaginary part of $Q_{1,2}$
Adjoint	-4.77637417162	-18.42548436723
Finite Difference (10^{-3})	-4.77492740836	-18.40590905454
Finite Difference (10^{-4})	-4.77622649100	-18.42352702795
Finite Difference (10^{-5})	-4.77633969638	-18.42533264380
Finite Difference (10^{-6})	-4.77622052131	-18.42533973218

An expression is formed for the imaginary part of the GAF as well:

$$\frac{df}{\partial (\Im (Q_{m,n}))} = \psi_f^T \frac{\partial \mathbf{R}_f}{\partial (\Im (Q_{m,n}))}. \quad (50)$$

For the partial derivatives of the flutter residual, Eq. 15, the derivatives of the eigenvalue are computed from the generic equation:

$$\frac{\partial p}{\partial z} = \left[\bar{\mathbf{L}}^T \left(\frac{\partial \bar{\mathbf{B}}}{\partial z} - p \frac{\partial \bar{\mathbf{A}}}{\partial z} \right) \bar{\mathbf{R}} \right] / (\bar{\mathbf{L}}^T \bar{\mathbf{A}} \mathbf{R}), \quad (51)$$

where z is the real or imaginary part of a variable in $\bar{\mathbf{A}}$ or $\bar{\mathbf{B}}$, e.g., the real or imaginary parts of the GAFs or the dynamic pressure. As noted above, GAFs are computed at a range of frequencies, with linear interpolation to cover frequencies in between. The flutter metric f will typically be a function of two of these GAF sets: the one computed at a frequency directly below the flutter reduced frequency k , and the one directly above. A finite difference verification of these derivatives can be seen in Table 1, for the AGARD 445.6 wing.

D. Implementation and Verification

The full set of adjoint equations, Eqs. 27-31, are solved in the order presented. A Python wrapper orchestrates the process by calling framework-independent solver interfaces that the SFE and grid deformation solver implement [30]. For each adjoint equation, FUN3D/SFE provides interface functions to compute the left hand side and solve the linear system given a right hand side vector. SFE and the grid deformation solver use SLAT to solve the linear systems. FUN3D/SFE also provides interfaces to perform transposed-matrix vector products of the residual and force distribution linearizations with respect to the flow state, grid coordinates, grid velocities. To form the right hand side terms in the adjoint equations, the Python wrapper calls these transposed matrix vector product functions with the appropriate complex step perturbation vectors to perform the directional derivatives. The LFD summations in Eqs. 29, 30, and 31 are accumulated during the nested loops solving the LFD flow adjoint equation, Eq. 28. The linearized flow states, $\hat{\mathbf{q}}_{j,k}$, are saved to disk during the forward problem to avoid having to recompute or store them in memory.

All terms in the adjoint equations have been verified individually. Even though the LFD process involves complex variables, most of the linearizations of the residuals only involve real-valued states and equations. Therefore, these terms can be verified with the complex step method. When possible, terms were verified with complex step derivatives; otherwise terms like Eq. 32 were verified as best as possible with real-valued finite difference checking (the residual partial derivatives in Eq. 32 with respect the grid coordinates and velocities had previously been complex-step verified in Ref. [10]).

The verification of the linearization required for Eq. 36 is shown in Table 2. The dot product test compares $\mathbf{v}_1^T J^T \mathbf{v}_2$ where $\mathbf{v}_1$ and $\mathbf{v}_2$ are randomly generated vectors. In the table, the complex-step version is the forward mode operation ($\mathbf{v}_2^T J \mathbf{v}_1$) where J is computed as a complex-step derivative of $\frac{\partial \mathbf{R}}{\partial \mathbf{q}}$ in the direction of $\mathbf{v}_1$. The forward and adjoint operations agree to machine precision verifying that the implementation for transposed second-order linearizations of the flow residual is correct.

Once all of the individual terms had been verified using the dot product test, a real-valued finite difference comparison were performed for the sensitivities of the GAFs with respect to two input parameters, Mach number and angle of attack. The adjoint sensitivities of the real and imaginary parts of a GAF for an NACA 64A010 airfoil are shown in Table 3. The adjoint derivatives agree best with the finite difference step sizes of 10^{-5} and 10^{-6} before subtractive cancellation reduces the accuracy of the finite difference derivative.

Table 2 Dot product test verification of Eq. 36.

Derivative Type	Dot Product
Adjoint	0.42633332843857663
Complex step	0.42633332843857674

Table 3 Verification of derivatives of GAFs with respect to design variables.

Derivative Type	Real part of GAF		Imaginary part of GAF	
	Angle of Attack	Mach number	Angle of Attack	Mach number
Adjoint	-0.00183288	5.28476421	0.00208708	-5.86158014
Finite Difference (10^{-4})	-0.00183304	5.28435621	0.00208862	-5.86071691
Finite Difference (10^{-5})	-0.00183322	5.28472296	0.00208715	-5.86149420
Finite Difference (10^{-6})	-0.00183523	5.28475593	0.00208768	-5.86157535
Finite Difference (10^{-7})	-0.00185616	5.28472113	0.00209200	-5.86161732

IV. Flutter Boundary of the AGARD 445.6 Wing

With the LFD adjoint and p-k linearizations verified, these pieces were combined to model the flutter boundary of the AGARD 445.6 wing [20]. The wing that was tested at the NASA Langley Transonic Dynamics Tunnel (TDT) has an aspect ratio of 1.65, a 45° sweep angle, and 0.66 taper ratio. The airfoil is symmetric, and the wing has no twist meaning this case can be studied without accounting for the static aeroelastic effects at an angle of attack of zero. The wing is simulated in FUN3D/SFE as an inviscid flow with a mixed-element mesh consisting of 292,638 nodes for a Mach number range from 0.5 to 1.141. The structure is represented by four modes, and ten reduced frequencies in the range of $k = 0.01 - 2.0$ are selected for the LFD analysis. The computed flutter boundary is compared to the experimental boundary and data from FUN3D finite volume (FV) simulations [31] in Fig. 1. For the three lower Mach numbers there is agreement between the different computations and the experiment. As shocks appear in the flow above $M_\infty = 0.901$, LFD predicts lower flutter dynamic pressures than the time-domain solver. The first mode is unstable for all dynamic pressures greater than zero at $M_\infty = 0.96$. It is not known how mesh-independent either Euler flutter boundary is. Differences in the resolution of the shocks and applied shock smoothing techniques may explain the higher stability of the time-domain results. Another possibility is numerical damping in the time-domain results from the time step size. The differences between the time-domain FV and frequency-domain SFE results warrant further investigation but are not the focus of this paper.

In addition to the flutter points, the lines in Fig. 1 represent the derivatives of the flutter dynamic pressure with respect to Mach number as computed by the adjoint method. The derivatives have a shallow negative slope at the lower Mach numbers and steep positive slopes at the supersonic speeds, which is consistent with the trends of the computed flutter points. The computed sensitivities at two Mach numbers in Fig. 1 are compared to real-valued finite differences in Table 4. The adjoint and finite difference derivatives agree to five digits for $h = 10^{-5}$ at the subsonic Mach number and to six digits for $h = 10^{-7}$ at the supersonic Mach number.

The computational cost of the LFD analysis and adjoint-based derivative is summarized in Table 5. The derivative required one adjoint solution per flow condition with the cost function directly being the flutter dynamic pressure. Despite the need for multiple complex step derivatives of linearization of the residual like Eq. 36, the LFD adjoint solution is less expensive than the forward linear problems. Because the cost function in the p-k solver is based on linear interpolation of GAFs between frequencies at the flutter point, the LFD adjoint equation solutions and right hand side terms for the steady flow adjoint equations are only required for two of the ten frequencies from the forward analysis. Therefore, not only is the adjoint better because it is essentially independent of the number of design variables, but it is also lower cost than a real-valued finite difference for studying the sensitivity of flutter with respect to a single variable. It is also more accurate since the finite difference issue of subtractive cancellation is not present. Overall, the LFD method and corresponding adjoint allow the calculation of the flutter point and its derivatives for about 5.3 times the cost of the steady forward analysis for this case.

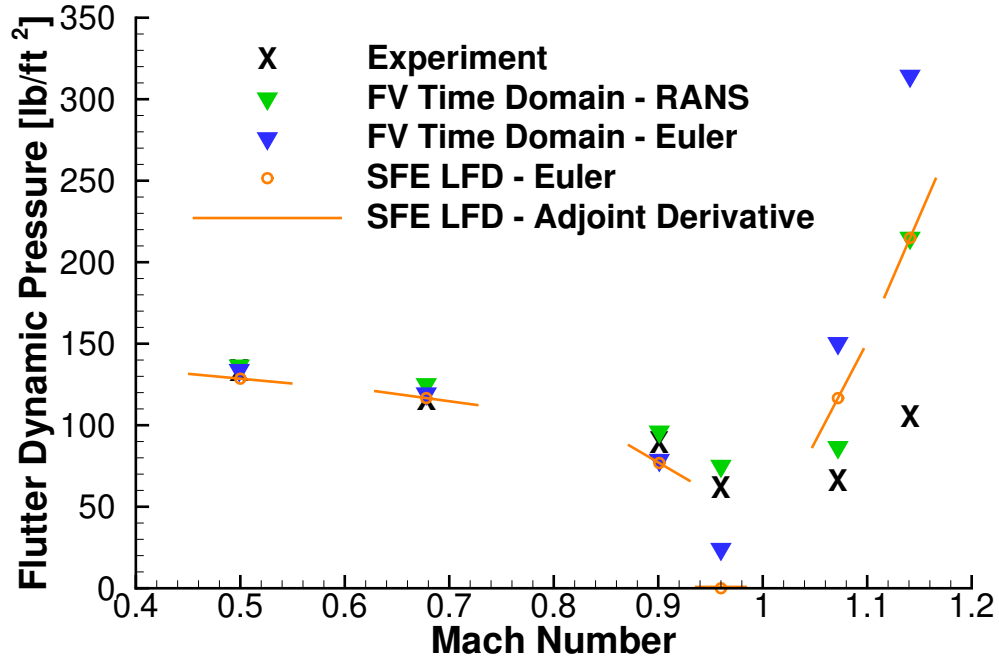


Fig. 1 Flutter boundary of the AGARD 445.6 wing. FUN3D Finite volume (FV) solutions from Ref. [31].

Table 4 Sensitivities of the flutter dynamic pressure with respect to Mach number for the AGARD 445.6 using LFD.

	$M_\infty = 0.5$	$M_\infty = 1.072$
Adjoint	-59.6929941	1225.945250
Finite Difference (10^{-4})	-59.7222326	1248.034599
Finite Difference (10^{-5})	-59.6930351	1245.346810
Finite Difference (10^{-6})	-59.6926180	1226.022198
Finite Difference (10^{-7})	-59.6874500	1225.946330
Finite Difference (10^{-8})	-59.6376992	1225.944700

Table 5 Computational time for the stages of the AGARD 445.6 analysis at $M_\infty = 0.5$ with one dual socket Skylake node (40 cores). The solver was run in complex variable mode for all stages.

Solution phase	Walltime (min)	Percentage of total time
Steady flow solution	87.9	18.7
Volume mode shapes	4.7	1.0
LFD linear problems	226.2	48.2
LFD adjoint solver and terms	143.5	30.6
Steady flow adjoint	7.0	1.5
Total	469.3	100.0

V. Future Work

The next steps in this work are to integrate the developed LFD adjoint and p-k flutter cost functions with static aeroelastic analysis to study cases beyond symmetric wings like the AGARD 445.6. To connect the LFD and p-k solvers into a hand derived set of adjoint equations would only require adding the LFD summations and cost function terms in Eqs. 30 and 31 to the existing right hand sides of the grid and flow adjoint equations in an implementation like FUNtoFEM [32]. Much of FUNtoFEM has been refactored into MPhys [33], which builds on OpenMDAO [34] as the underlying framework, to automate coupled sensitivity calculations and problem set up for high-fidelity optimization problems. The LFD solver, the p-k solver, and their respective linearizations will be encapsulated into OpenMDAO components and connected to static aeroelastic analysis through MPhys.

VI. Conclusions

This work has presented the adjoint equations corresponding to the LFD solver in FUN3D/SFE and the sensitivities of flutter cost functions computed by a p-k flutter problem. The problem requires an adjoint formulation accounting for flutter cost functions that do not satisfy the Cauchy-Riemann equations. Directional complex step derivatives are applied to compute sensitivities of the linearized flow equations. The adjoint implementation was verified and demonstrated on the AGARD 445.6 wing. For the AGARD 445.6 wing, the adjoint-based derivatives of the flutter dynamic pressure with respect to Mach number was shown to agree with finite difference derivatives to 5-6 digits. The LFD solver and the adjoint is scalable with respect to the number of design variables and significantly more efficient than time-domain analysis. This makes it suitable for adding flutter constraints to multidisciplinary optimization with gradient-based optimizers and uncertainty quantification.

Acknowledgments

The authors would like to thank Boris Diskin from the National Institute of Aerospace for thoroughly reviewing the formal presentation of the equations. 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 were provided by the NASA Langley K cluster.

References

- [1] Housman, J. A., and Kiris, C. C., "Overset Grid Simulations for the Second AIAA Aeroelastic Prediction Workshop," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Grapevine, Texas, 2017.
- [2] Jonsson, E., Riso, C., Lupp, C. A., Cesnik, C. E., Martins, J. R., and Epureanu, B. I., "Flutter and post-flutter constraints in aircraft design optimization," *Progress in Aerospace Sciences*, Vol. 109, 2019, p. 100537.
- [3] Zhang, Z., Chen, P. C., Yang, S., Wang, Z., and Wang, Q., "Unsteady Aerostructure Coupled Adjoint Method for Flutter Suppression," *AIAA Journal*, Vol. 53, No. 8, 2015, pp. 2121–2129.
- [4] Jacobson, K. E., Kiviaho, J. F., Kennedy, G. J., and Smith, M. J., "Evaluation of Time-domain Damping Identification Methods for Flutter-constrained Optimization," *Journal of Fluids and Structures*, Vol. 87, 2019, pp. 174 – 188.
- [5] Thomas, J., and Dowell, E. H., "Discrete Adjoint Constrained Design Optimization Approach for Unsteady Transonic Aeroelasticity and Buffet," *AIAA Aviation Forum*, Virtual, 2020.
- [6] He, S., Jonsson, E., Mader, C. A., and Martins, J. R. R. A., "A Coupled Newton-Krylov Time Spectral Solver for Wing Flutter and LCO Prediction," *AIAA Aviation Forum*, Dallas, Texas, 2019.
- [7] Prasad, R., and Choi, S., "Aerodynamic Shape Optimization for Flutter/LCO based design using Coupled Adjoint," *58th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2020.
- [8] Widhalm, M., Dwight, R. P., Thormann, R., and Hübner, A., "Efficient Computation of Dynamic Stability Data with a Linearized Frequency Domain Solver," *Proceedings of the European Conference on Computational Fluid Dynamics*, Lisbon, Portugal, 2010.
- [9] Widhalm, M., and Thormann, R., "Efficient Evaluation of Dynamic Response Data with a Linearized Frequency Domain Solver at Transonic Separated Flow Condition," *35th AIAA Applied Aerodynamics Conference*, 2017.

- [10] Jacobson, K. E., Stanford, B. K., Wood, S. L., and Anderson, W. K., "Flutter Analysis with Stabilized Finite Elements based on the Linearized Frequency-domain Approach," *58th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2020.
- [11] Stanford, B. K., Jacobson, K. E., and Massey, S. J., "Transonic Aeroelastic Modeling of the NACA 0012 Benchmark Wing," *AIAA Aviation Forum*, Virtual, 2020.
- [12] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi:10.1007/BF01061285.
- [13] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, 2002, pp. 1155–1163.
- [14] Giles, M. B., and Süli, E., "Adjoint Methods for PDEs: A Posteriori Error Analysis and Postprocessing by Duality," *Acta numerica*, Vol. 11, 2002, pp. 145–236.
- [15] Pierce, N. A., and Giles, M. B., "Adjoint and Defect Error Bounding and Correction for Functional Estimates," *Journal of Computational Physics*, Vol. 200, No. 2, 2004, pp. 769–794.
- [16] Venditti, D., and Darmofal, D., "A Grid Adaptive Methodology for Functional Outputs of Compressible Flow Simulations," *15th AIAA Computational Fluid Dynamics Conference*, 2001, p. 2659.
- [17] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46.
- [18] Kenway, G., Kennedy, G., and Martins, J., "Aerostructural Optimization of the Common Research Model Configuration," *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 2014.
- [19] Jacobson, K. E., Kiviaho, J. F., Kennedy, G. J., and Smith, M. J., "Evaluation of Time-domain Damping Identification Methods for Flutter-constrained Optimization," *Journal of Fluids and Structures*, Vol. 87, 2019, pp. 174 – 188.
- [20] Yates, C. E., "AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I - Wing 445.6," Tech. Rep. TM-100492, NASA, August 1 1987.
- [21] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers & Fluids*, Vol. 23, No. 1, 1994, pp. 1–21.
- [22] Biedron, R. T., and Thomas, J. L., "Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications," *AIAA 47th Aerospace Sciences Meeting*, Orlando, Florida, 2009.
- [23] Wang, L., Diskin, B., Biedron, R., Nielsen, E. J., and Bauchau, O., "Sensitivity Analysis of Multidisciplinary Rotorcraft Simulations," *55th AIAA Aerospace Sciences Meeting*, 2017, p. 1670.
- [24] Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *55th AIAA Aerospace Sciences Meeting*, 2017.
- [25] Anderson, W. K., and Newman, J., "High-Order Stabilized Finite Elements on Dynamic Meshes," *2018 AIAA Aerospace Sciences Meeting*, 2018.
- [26] Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model," 2012.
- [27] Saad, Y., and Schultz, M., "GMRES: a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *Siam Journal on Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.
- [28] Wood, S. L., Jacobson, K. E., Jones, W. T., and Anderson, W. K., "Sparse Linear Algebra Toolkit for Computational Aerodynamics," *58th AIAA Aerospace Sciences Meeting*, Orlando, Florida (submitted for publication), 2020.
- [29] Igarashi, H., and Watanabe, K., "Complex Adjoint Variable Method for Finite-Element Analysis of Eddy Current Problems," *IEEE Transactions on Magnetics*, Vol. 46, No. 8, 2010, pp. 2739–2742. doi:10.1109/TMAG.2010.2043936.
- [30] Jones, W. T., Wood, S. L., Jacobson, K. E., and Anderson, W. K., "Interoperable Application Programming Interfaces for Computer Aided Engineering Applications," *59th AIAA Aerospace Sciences Meeting*, Virtual, 2021.

- [31] Silva, W. A., Chwalowski, P., and Perry, B., “Evaluation of Linear, Inviscid, Viscous, and Reduced-order Modelling Aeroelastic Solutions of the AGARD 445.6 Wing using Root Locus Analysis,” *International Journal of Computational Fluid Dynamics*, Vol. 28, No. 3-4, 2014, pp. 122–139.
- [32] Kiviaho, J. F., Jacobson, K. E., Smith, M. J., and Kennedy, G. J., “A Robust and Flexible Coupling Framework for Aeroelastic Analysis and Optimization,” *Proceedings of the 19th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Denver, Colorado, 2017.
- [33] Yildirim, A., Jacobson, K. E., Stanford, B. K., Gray, J. S., Mader, C. A., Martins, J. R. R. A., and Kennedy, G. J., “MPhys: A Modular Multiphysics Simulation Package using the OpenMDAO Framework,” *AIAA Aviation Forum*, Presentation Only, Virtual, 2020.
- [34] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, 2019, pp. 1075–1104. doi:10.1007/s00158-019-02211-z.