Flutter-Constrained Optimization with the Linearized Frequency-Domain Approach

Kevin E. Jacobson,* Bret K. Stanford,†
NASA Langley Research Center, Hampton, VA, 23681

Due to the high computational cost associated with unsteady aeroelastic analysis, state-of-the-art aeroelastic optimizations based on computational fluid dynamics typically ignore critical constraints like flutter and aeroelastic gust response. The linearized frequency-domain method offers an approach for adding high-fidelity flutter constraints to multidisciplinary optimizations at relatively low cost compared to other unsteady computational fluid dynamics methods. In recent work, sensitivities have been implemented for the linearized frequency-domain method in FUN3D. In this work, the linearized frequency-domain method and associated sensitivities are applied to gradient-based aeroelastic optimization with flutter constraints based on computational fluid dynamics. An overview of the flutter constraint formulation and implementation is provided, and then two optimization problems are presented. The first optimization increases the flutter speed of a pitch and plunge airfoil at transonic conditions using the minimal amount of geometric changes. The second optimization minimizes the mass of the AGARD 445.6 wing subject to a flutter constraint.

I. Introduction

It is a known issue that high-fidelity aerostructural optimizations that only include static aeroelastic constraints will likely lead to infeasible designs once dynamic aeroelastic effects are considered [1]. Often the most critical dynamic aeroelastic behaviors occur in the transonic flow regime where linear aerodynamic models are insufficient, leading to a need for computational fluid dynamics (CFD)-based analysis. Development of a transonic flutter constraint efficient enough to apply in large multidisciplinary optimization problems remains an open problem [2]. The most general form of dynamic aeroelastic analysis is integration of the coupled equations in the time domain to observe the behavior of the vehicle. There are some examples of time-domain CFD with adjoint-based derivatives used for flutter identification [3] or flutter suppression [4–6], but these methods remain cost-prohibitive for more complete aerospace optimization problems.

The nature of many aeroelastic phenomena lends itself to making a periodicity assumption for the aerodynamic loading leading to frequency-domain methods. Several groups are studying nonlinear frequency-domain techniques [7–9] to compute flutter and post-flutter behavior, but as with the time-domain approaches, these efforts have been limited to relatively simple problems to date. Because flutter is a linear stability problem, the form of analysis can be further specialized to linearized methods. As with traditional linear aeroelastic tools, the output of the linearized frequency-domain (LFD) method [10] is the matrix of generalized aerodynamic forces (GAFs), which represents the aerodynamic loading on structural modes due to harmonic motion of structural modes at specified frequencies. Unlike the traditional linear methods, the LFD method is applicable in transonic flows because it is linearized about a nonlinear equilibrium state computed with static aeroelastic analysis based on CFD. Previous research [11, 12] has demonstrated that the LFD method reduces the cost of CFD-based flutter analysis by orders of magnitude compared to time-domain flutter analysis.

In prior work [13], an LFD solver was added to the streamline-upwind Petrov-Galerkin (SUPG) finite-element solver in FUN3D, called FUN3D/SFE [14, 15]. Linearizations for adjoint-based sensitivities were also added to the LFD solver in FUN3D/SFE [16] to allow efficient computation of sensitivities of cost functions computed from LFD-based flutter analysis. In this work, these adjoint-based sensitivities are utilized to perform flutter-constrained optimization problems. First, the flutter constraint formulation and implementation are described. Next, a transonic optimization of the Isogai [17] airfoil is discussed, and finally optimization of the AGARD 445.6 wing [18] is compared to a Doublet Lattice Method (DLM)-based analysis.

*Research Aerospace Engineer, Aeroelasticity Branch, MS 340, AIAA Member, kevin.e.jacobson@nasa.gov
†Research Aerospace Engineer, Aeroelasticity Branch, MS 340, AIAA Associate Fellow.

II. Methodology

This section summarizes the formulation and implementation of the flutter constraint. The analysis has been implemented with OpenMDAO [19] as the underlying framework. Coupled models are constructed in OpenMDAO by connecting inputs and outputs of analysis "components" and specifying which coupling algorithm to use for a collection of components. A key advantage of OpenMDAO is that the Python framework automates the sensitivity calculation for a coupled problem given that each component computes its own partial derivatives. The assembly of the flutter analysis processes in OpenMDAO is done with Mphys [20], an OpenMDAO library collaboratively developed to standardize high-fidelity multiphysics with OpenMDAO. With established coupling conventions and utilities provided for assembling multiphysics problems, Mphys has demonstrated the ability to readily swap physics solvers and even fidelities without modifying other aspects of the coupled optimization problem. The ability to swap fidelities is utilized in the AGARD 445.6 optimization in the results.

The overall process as assembled by Mphys for the flutter constraint is given in the extended design structure matrix (XDSM) diagram [21] in Fig. 1. In this diagram, each rectangle represents an OpenMDAO subsystem, which can be a single OpenMDAO component or a group of components associated with a discipline. An analysis starts with the optimizer setting the values of the structural sizing and shape design variables, $\mathbf{dv}_{sizing}$ and $\mathbf{dv}_{shape}$, respectively. From the shape design variables, the geometry component determines the coordinates of nodes in the structural mesh ($\mathbf{x}_{struct}$) and the surface nodes of the aerodynamic mesh, ($\mathbf{x}_{aero}$). Next, the structural solver performs a modal decomposition to compute the mode shapes ($\mathbf{\Phi}_{struct}$) and modal mass, damping, and stiffnesses ($\overline{\mathbf{M}}$, $\overline{\mathbf{C}}$, $\overline{\mathbf{K}}$), respectively.

Because the flutter analysis is a determination of the linear stability about the steady state condition, the next phase of the process is a coupling loop to find the static aeroelastic equilibrium point. For the results in this paper, the selected coupling algorithm is a nonlinear block Gauss-Seidel (NLBGS) with Aitken's underrelaxation [22] to improve stability. The gradient evaluation applies a linear block Gauss-Seidel (LNBGS) with Aitken's underrelaxation. In the static aeroelastic loop, the displacement transfer is given displacements of the structural mesh ($\mathbf{u}_{struct}$) and projects them onto the aerodynamic surface. From the new surface, the aerodynamic solver updates the flow state and computes the forces on the surface ($\mathbf{f}_{aero}$). The load transfer then projects those forces onto the structural model. New displacements are calculated by the structural solver, and the NLBGS loop continues until the convergence tolerance is met or the maximum number of cycles is performed. After the static aeroelastic problem converges, the flutter analysis is performed. The displacement transfer scheme projects the structural mode shapes onto the aerodynamic surface for the flutter subsystem, to form $\mathbf{\Phi}_{aero}$. The flutter subsystem in Fig. 1 is a set of components that computes linearized unsteady aerodynamics and then forms the flutter constraint function, $KS_{flutter}$, from the output of a p-k eigenanalysis. The details of the flutter group are discussed in the Section II.B.

The structural solver that performs the modal decomposition and the structural solver in the NLBGS loop is an in-house linear shell finite-element solver with hand-derived shape and sizing derivatives. For derivatives of eigenvalues and eigenvectors via the modal decomposition, Nelson's method [23] is utilized.

A. Static Aeroelastic Subsystems

For FUN3D/SFE, the aerodynamic solver subsystem in Fig. 1 contains three analysis components given in Fig. 2. Because FUN3D/SFE uses a body-fitted volume mesh and the Mphys convention is that the aerodynamic solver's input is the surface displacements, the first FUN3D/SFE component is a mesh deformation component, which updates the volume mesh coordinates, $\mathbf{x}_{G0}$, to the new surface, which includes geometric changes and aeroelastic displacements, $\mathbf{u}_{aero}$. The mesh deformation scheme in FUN3D is based on a linear elasticity analogy [24, 25]. The second aerodynamic component is the steady SUPG flow solver, which computes the flow state, $\mathbf{q}_0$, on the deformed mesh, and the third component determines the aerodynamic forces at the surface nodes, $\mathbf{f}_{aero}$. For calculating sensitivities, the SFE flow solver and force integration components perform automatic differentiation with operator overloading. The linearizations of the mesh deformation are implemented with hand-derived derivatives.

The selected load and displacement transfer scheme is Matching-based Extrapolation for Loads and Displacements (MELD) [26]. This method computes the motion of each aerodynamic point from a weighted least-squares fit of the translation and rotation of the N closest structural nodes. Because this displacement transfer is nonlinear, the load transfer, derived by applying the principle of virtual work, is a function of the structural displacements in Fig. 1.

B. Flutter Constraint Computation

The FUN3D/SFE flutter subsystem in Fig. 1 consists of the four steps in Fig. 3. The modal mesh deformation uses the FUN3D linear elastic solver to compute the volume mesh coordinates corresponding to the aerodynamic surface

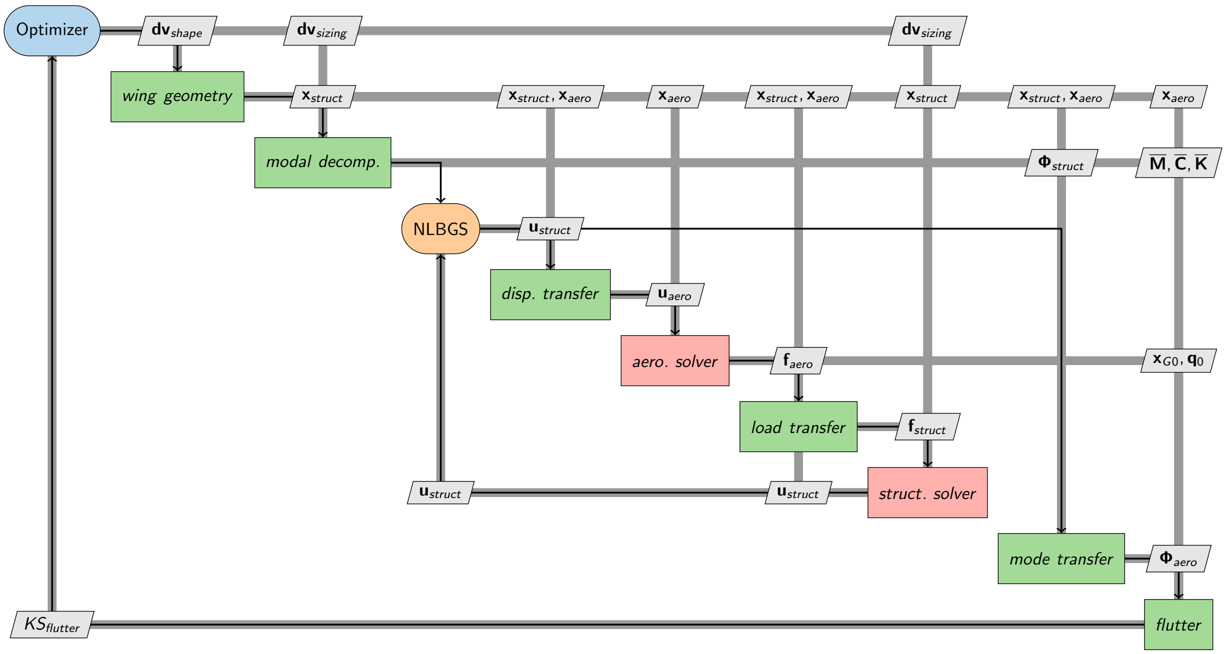


Fig. 1 General XDSM diagram of the flutter constraint assembled in OpenMDAO by Mphys.

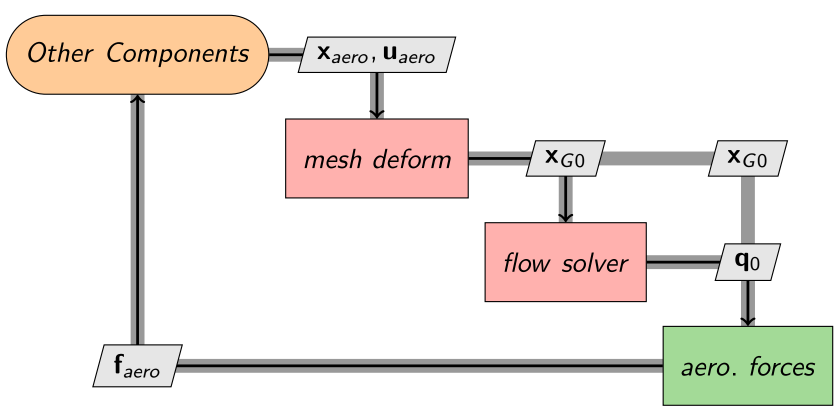


Fig. 2 The aerodynamic solver subsystem in Fig. 1 for FUN3D/SFE.

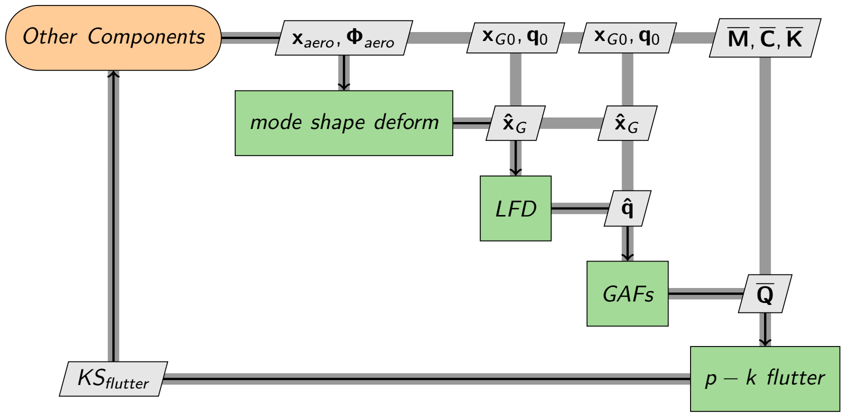


Fig. 3 The steps of the flutter subsystem in Fig. 1 for the LFD solver in FUN3D/SFE.

created by a unit displacement of each mode, $\mathbf{\Phi}_{aero}$. The LFD solver, described in Section II.B.2, then computes the linearized flow solutions, which are complex-valued states representing the linearized flow's response to harmonic perturbations of each mode at prescribed frequencies. In the third step, the GAFs are computed based on the LFD aerodynamic states. And finally the p-k eigenanalysis, described in Section II.B.1, is performed to determine the flutter response and compute the flutter constraint function.

The four steps of the flutter subsystem are implemented as two OpenMDAO components: the linearized aerodynamic subsystem and the p-k analysis. While the mesh deformation, flow solver, and force integration in Fig. 2 from the static aeroelastic loop are split into separate components for modularity, the modal mesh deformation, LFD solver, and GAFs computation are implemented as a single OpenMDAO component. This was decided because in its current form OpenMDAO will store a copy of the inputs and outputs of the components, which equates to a large amount of memory when considering there is a linearized flow solution for each mode and reduced frequency pair. By fusing the aerodynamic portions into a single component, the OpenMDAO memory footprint related to the LFD analysis is reduced to only the input surface mode shapes and the GAFs, which are of negligible size compared to the CFD solutions. While these vectors are not kept in memory, the linearized states and volume mesh motions are needed during the sensitivity computation and are saved to and loaded from files as needed. For the fused components, the relevant hand-derived adjoint equations from Ref. 16 are implemented in the component to compute the sensitivities of the GAFs, the fused component's output, with respect to the inputs: the mode shapes, steady flow state, and steady state deformed mesh.

1. P-k Eigenanalysis

The p-k flutter analysis is an eigenvalue problem for the stability of the dynamic aeroelastic system. Writing the system in a nondimensional state-space form leads to a generalized eigenproblem:

$$\left(\overline{\mathbf{A}}\, p - \overline{\mathbf{B}}\,(ik)\right)\overline{\mathbf{R}} = 0, \tag{1}$$

where p is the nondimensional eigenvalue, $p := bs/U_\infty = g + ik$, s is the Laplacian, b is the reference semichord of the wing, U_∞ is the freestream flow speed, and $\overline{\mathbf{R}}$ is the corresponding right eigenvector. The imaginary part of p, known as the reduced frequency, is a nondimensional form of frequency, $k = \omega b/U_\infty$. The real part of p corresponds to the damping with positive real parts representing an unstable system. The two matrices in the eigenproblem are defined as:

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

and

$$\overline{\mathbf{B}}\,(ik) := \begin{bmatrix} 0 & \mathbf{I} \\ -\overline{\mathbf{K}} + q_\infty\, \overline{\mathbf{Q}}(ik) & -\left(\frac{U_\infty}{b}\right)\overline{\mathbf{C}} \end{bmatrix}, \tag{3}$$

where q_∞ is the freestream dynamic pressure, $\overline{\mathbf{M}}$, $\overline{\mathbf{C}}$, and $\overline{\mathbf{K}}$ are the reduced structural mass, damping, and stiffness matrices, respectively, from the modal decomposition of the structural model. $\overline{\mathbf{Q}}(ik)$ is the GAF matrix. In this matrix $\overline{\mathbf{Q}}_{m,n}(ik)$ represents the Fourier coefficient of the aerodynamic load on mode m due to harmonic motion of mode n at reduced frequency k. The dependence of the GAFs on the imaginary part of the eigenvalue leads to the p-k equations being a nonlinear generalized eigenproblem. To solve this problem, a sweep over reduced frequency is performed for a given dynamic pressure, and valid solutions are found when the input k matches the imaginary part of the resulting eigenvalue. The GAFs are computed by the LFD solver at a set of reduced frequencies covering the possible range of flutter frequencies. When the p-k solver requires GAFs at a given frequency in its reduced frequency sweep, the GAFs are interpolated from those computed by the LFD analysis. The aeroelastic eigenproblem is solved for a range of dynamic pressures, and a transition from negative to positive real parts of the eigenvalues indicates a flutter instability or aeroelastic divergence in the case where the imaginary part is also zero.

The flutter constraint is formed from the eigenvalues of the p-k analysis. Unlike some aspects of aircraft design, which can be safely studied only at the extreme conditions like many loads and controls analyses, flutter can occur anywhere in the flight envelope. Hard flutter is a situation where the system is unstable at any point above the flutter condition; however, hump modes are instabilities that appear at particular dynamic pressures but then restabilize as the dynamic pressure continues to increase. To prevent unstable hump modes below the desired lower bound for the flutter point, the constraint defines a piecewise polynomial function to form a “keep out” zone for the real parts of the eigenvalues, written as a function of dynamic pressure as in Fig. 4 [27]. The boundary is defined as

$$G\left(q_\infty\right) := \begin{cases} g^*\left(3q^*q_\infty^2 - 2q_\infty^3\right)/(q^*)^3 & 0 \leq q_\infty < q^* \\ \beta\left(q_\infty - q^*\right)^2 + g^* & q_\infty \geq q^*, \end{cases} \tag{4}$$

where q^* is the minimum allowable flutter dynamic pressure, g^* is the minimum allowed damping at this condition, and β is a constant. The final constraint function is calculated as a Kreisselmeier-Steinhauser (KS) aggregation [28, 29] to approximate the maximum violation of the failure boundary (difference of the damping and the boundary) in a differentiable expression

$$KS_{flutter} := \frac{\ln\left[\sum_{k=1}^{M} e^{\rho\left((g_k - G(q_\infty))/g_{ref}\right)}\right]}{\rho}, \tag{5}$$

where g_{ref} is a reference damping value and ρ is a parameter to control the nonlinearity of the approximate maximum computed by the KS aggregation. Larger values of ρ lead to a better approximation of the maximum failure boundary violation but make the function more numerically challenging to apply in an optimization. Ultimately, the constraint enforced by the optimizer is $KS_{flutter} \leq 0$.

As discussed in Ref. 16, linearizing the flutter constraint formed by the p-k component requires special treatment. The KS output function is not complex differentiable because it does not satisfy the Cauchy-Riemann equations. Compared to the actual complex GAF matrix, the GAF input to the p-k OpenMDAO component is defined as a real-valued input by adding a size 2 dimension to represent the real and imaginary parts of the actual matrix, $\overline{\mathbf{Q}}(ik) = \mathbf{a} + i\mathbf{b}$. Due to these two real values representing the real and imaginary part of a complex number, the partial derivative computed for the imaginary term is multiplied by -1 in order to compute the correct final sensitivities. This creates equivalent adjoint equations to those derived in Ref. 16 where the right hand side of the adjoint equations is $\frac{\partial f}{\partial \mathbf{a}} - i\frac{\partial f}{\partial \mathbf{b}}$.

2. Computing Generalized Aerodynamic Forces with LFD

The unsteady governing equation of the SUPG formulation in FUN3D/SFE can be written as

$$\mathbf{A}(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{x}_G, \dot{\mathbf{x}}_G) := \mathbf{M}_A(\mathbf{q}, \mathbf{x}_G, \dot{\mathbf{x}}_G)\,\dot{\mathbf{q}} + \mathbf{R}(\mathbf{q}, \mathbf{x}_G, \dot{\mathbf{x}}_G) = 0, \tag{6}$$

where $\mathbf{q}$ is the flow state vector, $\mathbf{x}_G$ is the vector of volume mesh coordinates, $\mathbf{M}_A$ is the mass matrix, which is a function of the state due to use of the Petrov-Galerkin weighting function, $\mathbf{R}$ is the spatial residual, and Newton notation for time derivatives is used. To form the LFD equation, this governing equation is linearized about the static aeroelastic equilibrium solution, $\mathbf{x}_{G0}$ and $\mathbf{q}_0$, with assumed harmonic mesh perturbations at frequency ω_k:

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

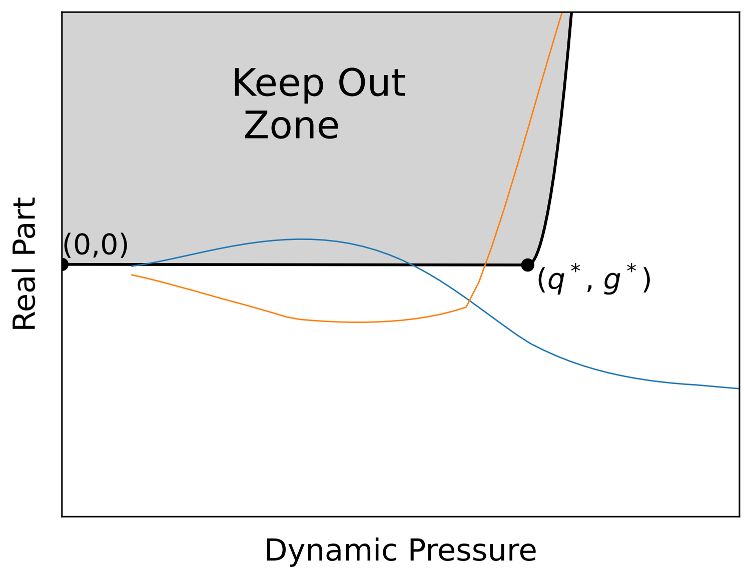


Fig. 4 Illustration of flutter constraint "keep out" zone for the real part of the p-k eigenvalues (the orange and blue lines) as a function of dynamic pressure.

where $\hat{\mathbf{x}}_{G,n}$ is the CFD volume mesh motion due to the unit displacement of structural mode n (computed via $\mathbf{\Phi}_{aero}$), and $\hat{\mathbf{q}}_{n,k}$ is the linearized flow response to motion of structural mode n at frequency ω_k. To solve this equation for $\hat{\mathbf{q}}_{n,k}$, the grid motion terms are moved to the right hand side

$$\left(i\omega_k \left. \mathbf{M}_A \right|_{\mathbf{x}_{G0},\mathbf{q}_0} + \left. \frac{\partial \mathbf{R}}{\partial \mathbf{q}} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \right) \hat{\mathbf{q}}_{n,k} = - \left. \frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} - i\omega_k \left. \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}_G} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n}. \tag{8}$$

To form the left and right hand sides of this complex-variabled linear system, exact linearizations of the flow residual are obtained with automatic differentiation using operator overloading. The volume mesh motion due to a unit displacement of structural mode $\hat{\mathbf{x}}_{G,n}$ is computed with the linear elasticity mesh deformation solver in FUN3D with the surfaces defined by the mode shape transfer component in Fig. 1. The linear system in Eq. 8 is solved using a preconditioned Generalized Minimal Residual (GMRES) method [30]. GAFs are computed from the linearized mesh motion and flow states by linearizing the surface force integration in FUN3D/SFE and taking an inner product of the linearized forces with the mode shapes

$$\overline{Q}_{m,n}(i\omega_k) := \mathbf{\Phi}^T_{aero,m} \hat{\mathbf{f}}_{n,k}, \tag{9}$$

where $\overline{Q}_{m,n}(i\omega_k)$ is the Fourier coefficient representing the GAF on mode m due to harmonic motion of mode n at frequency ω_k. $\hat{\mathbf{f}}_{n,k}$ is the linearized force distribution due to motion of mode n computed as

$$\hat{\mathbf{f}}_{n,k} := \left. \frac{\partial \mathbf{f}}{\partial \mathbf{q}} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{q}}_{n,k} + \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_G} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n} + i\omega_k \left. \frac{\partial \mathbf{f}}{\partial \dot{\mathbf{x}}_G} \right|_{\mathbf{x}_{G0},\mathbf{q}_0} \hat{\mathbf{x}}_{G,n}, \tag{10}$$

where $\mathbf{f}$ is the function used to calculate the aerodynamic force distribution.

Because the LFD residual in Eq. 7 and GAF computation in Eq. 9 have linearizations of the flow residual and force computation, matrix-vector products with the second derivatives are needed for the sensitivity analysis. As discussed in more detail in Ref. 16, these second derivatives are computed with complex-step directional derivatives of the operator overloaded first derivatives. An example of one of these terms when linearizing with respect to the steady flow state and

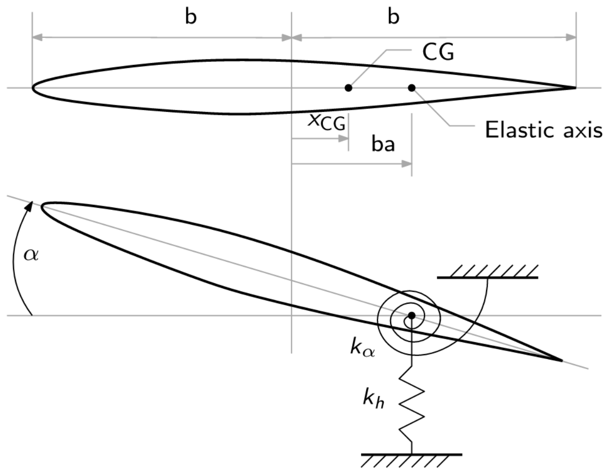


Fig. 5 Pitch-plunge airfoil model of Isogai [17].

taking the matrix-vector product with the vector $\mathbf{v}$ is

$$
\begin{aligned}
\left[\frac{\partial}{\partial \mathbf{q}}\left(\left.\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right|_{\mathbf{x}_{G0},\mathbf{q}_0}\hat{\mathbf{q}}_{n,k}\right)\right]^T \mathbf{v} &= \left(\frac{\partial}{\partial \mathbf{q}}\left(\left[\left.\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right|^T_{\mathbf{x}_{G0},\mathbf{q}_0}\mathbf{v}\right)\hat{\mathbf{q}}_{n,k}\right)\right. \\
&= \frac{\Im\left(\left.\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right|^T_{\mathbf{x}_{G0},\mathbf{q}=\mathbf{q}_0+ih\,\Re\left(\hat{\mathbf{q}}_{n,k}\right)}\Re\left(\mathbf{v}\right)\right)}{h} \\
&\quad - \frac{\Im\left(\left.\frac{\partial \mathbf{R}}{\partial \mathbf{q}}\right|^T_{\mathbf{x}_{G0},\mathbf{q}=\mathbf{q}_0+ih\,\Im\left(\hat{\mathbf{q}}_{n,k}\right)}\Im\left(\mathbf{v}\right)\right)}{h}.
\end{aligned}
\tag{11}
$$

where $\Re()$ and $\Im()$ are the operators to take the real or imaginary parts of the arguments. The complex-step method [31, 32] allows very small finite-difference pertubation sizes ($10^{-30} - 10^{-50}$), and therefore, the resulting derivatives are accurate to machine precision.

III. Transonic Airfoil Shape Optimization

The NACA64A-010 airfoil problem from Isogai [17] is a classic flutter problem noted for the inviscid S-shaped flutter boundary created by a hump mode in the transonic regime. The two-dimensional problem has a pitch-plunge structural model given in Fig. 5 with properties defined in Table 1. The elastic axis is located in front of the leading edge of the airfoil to represent an outboard section of a swept-back wing. For this problem the aerodynamic mesh contains 15,854 nodes, and the inviscid flow is simulated at Mach 0.875. The structural model only requires two degrees of freedom, and there are 14 LFD input frequencies making a total of 28 LFD linear problems per function evaluation.

The optimization problem developed for the Isogai airfoil is to make the minimum amount of geometry changes in order to satisfy the flutter constraint, given that the initial design does not satisfy the flutter constraint as shown in Fig. 6a. The geometry changes are done with a free-form deformation (FFD) method [33]. FFD is a geometry manipulation technique borrowed from computer graphics where the airfoil geometry moves as if it is embedded in a flexible rubber-like material in the space defined by a set of control points. The FFD control points are in a 3x8 rectangular grid around the airfoil in Fig. 7a; movement of these control points deforms the airfoil. With two degrees of freedom at each FFD control point, there are a total of 48 design variables. The optimization is then

$$
\begin{aligned}
\min_{\Delta\mathbf{x},\Delta\mathbf{y}} \quad & \sum_{i=1}^{24}\sqrt{\Delta x_i^2 + \Delta y_i^2} \\
\text{subject to:} \quad & KS_{flutter} \leq 0,
\end{aligned}
\tag{12}
$$

Table 1 Structural parameters of the NACA64A-010 airfoil [17].

Parameter	Value
Mass ratio (μ)	60.0
Elastic axis location (a)	-2.0
Normalized static unbalance	1.8
Radius of gyration (r_α^2)	3.48
Plunge natural frequency [rad/s] (ω_h)	100.0
Pitch natural frequency [rad/s] (ω_α)	100.0

where the objective is the sum of the displacements of the 24 FFD control points. For the flutter constraint, the minimum allowable flutter speed index and damping are $V_f^* = 2.75$ and $g^* = -0.25$, where the speed index is a nondimensional form of the freestream speed, $V_f := U_\infty / b\omega_\alpha\sqrt{\mu}$ where b is the reference semichord. For the optimization, the structural properties remain fixed. Although the problem is initially symmetric and therefore not requiring the static aeroelastic portion of the process, the analysis still includes the full process in Fig. 1 to allow nonsymmetric optimization solutions.

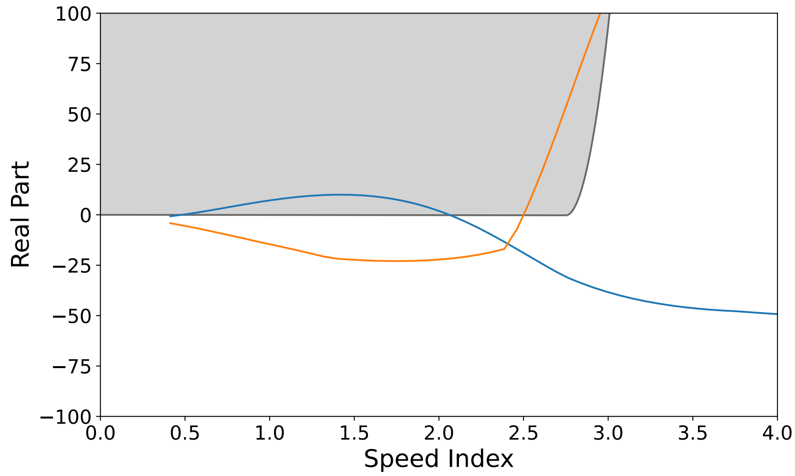
The optimization was performed with the sequential least-squares quadratic programming (SLSQP) optimizer [34]. It converged after 11 function evaluations and 9 gradient evaluations in Fig. 8. In the optimized aeroelastic response in Fig. 6b, the optimizer has suppressed the hump mode and pushed the hard flutter point beyond the minimum allowable speed index. It has achieved this by making the shape changes in Fig. 7b. The primary changes of reducing the thickness near the quarter-chord and increasing it near the midchord have the effect of shifting the shocks toward the trailing edge in Fig. 7d. Additionally, there is a small amount of asymmetry in the final airfoil design, which leads to a nonzero lift and static aeroelastic displacement. More analysis would need to be done to determine if the introduction of an asymmetrc airfoil is due to the initial asymmetry of the unstructured CFD mesh or if this is due to something physical like the staggering of the shocks or a nonzero lift creating a more stable condition.

The total optimization time was 129.26 minutes on a dual-socket Intel Skylake node with 40 CPU cores. An average of 9.1 iterations were needed in the NLBGS of the function evaluation and a maximum of 10 iterations were needed for the LNBGS in the gradient evaluation. As expected, Table 2 shows that essentially all the time is spent in the CFD solver doing mesh deformation, the steady flow solution, and the LFD analysis. Because the design variables do not affect the structural model, the modal decomposition component is not called by OpenMDAO when computing the gradients. There are several ways this computational time can be reduced in future work. Currently the CFD solver operates with complex variables throughout the process, but the components in the static aeroelastic loop do not require complex values. Switching to a real-valued version of the FUN3D/SFE solver in the OpenMDAO components in the static aeroelastic loop is estimated to save about a factor of 3 for those components, but there will be some overhead to reinstantiate the solver when switching variable types.

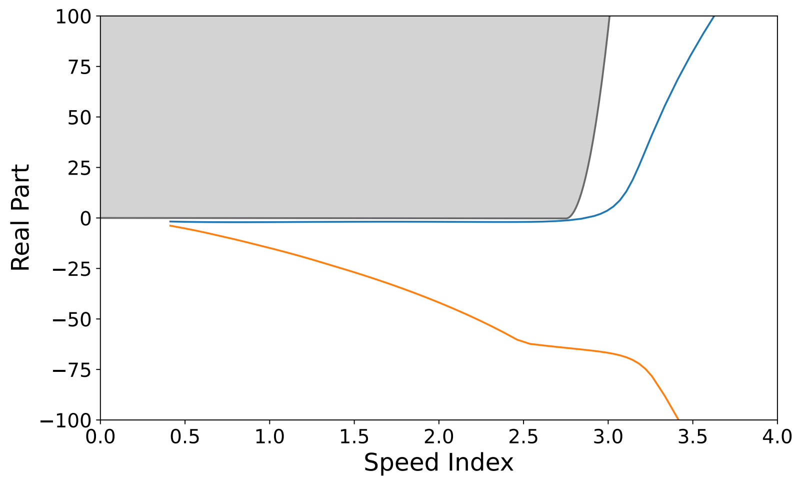
Another cause of unnecessary cost is the LFD sensitivity calculation being called to compute linearizations three times per gradient evaluation with the same input data. OpenMDAO is doing this because it needs derivatives from LFD with respect to the steady flow state, volume mesh, and aerodynamic mode shapes at different points in the gradient evaluation process; all of these recompute the LFD component's internal flow and modal mesh deformation adjoint equations because of the multiple analysis steps in the single component. At the cost of some additional data kept in memory, rewriting the LFD component to store the outputs on the first call and return the precomputed values on the next two would reduce the LFD cost in the gradient evaluation by a factor of 3. Since this work is an initial attempt to investigate whether this process will work, these types of code optimizations have not yet been performed, but they will be critical in order to make this approach tractable for bigger models like cases that will require turbulent CFD.

IV. AGARD 445.6 Optimization

The second optimization is performed with the AGARD 445.6 wing [18] using both DLM-based and LFD-based flutter constraints. One of the goals of performing this optimization is to compare the DLM-optimized design to that from the LFD version at a condition where both models are applicable; therefore, the optimization is performed at Mach 0.5 and 0.0° angle of attack. The CFD model for FUN3D/SFE is an inviscid analysis with a mesh containing 145,118 nodes. The DLM mesh is 256 quadrilateral panels with 8 panels in the chordwise direction and 16 in the spanwise direction. The structural model is a shell element model with the initial thickness distribution representing weakened



(a) Initial design.



(b) Optimized design.

Fig. 6 Predicted Isogai aeroelastic eigenvalues.

Table 2 Average time spent in each component per evaluation for the Isogai airfoil. Starred values occur within the static aeroelastic loop.

Component	Funciton Evaluation [s]	Gradient Evaluation [s]
Modal decomp.	0.002	N/A
Displacement transfer*	0.012	0.088
Mesh deform*	11.525	7.962
Flow solve*	133.363	70.716
Aero. forces*	0.384	1.240
Load transfer*	0.033	0.032
Struct. solve*	0.001	0.002
Mode transfer	0.001	0.002
LFD	157.863	466.800
p-k flutter	2.238	0.204

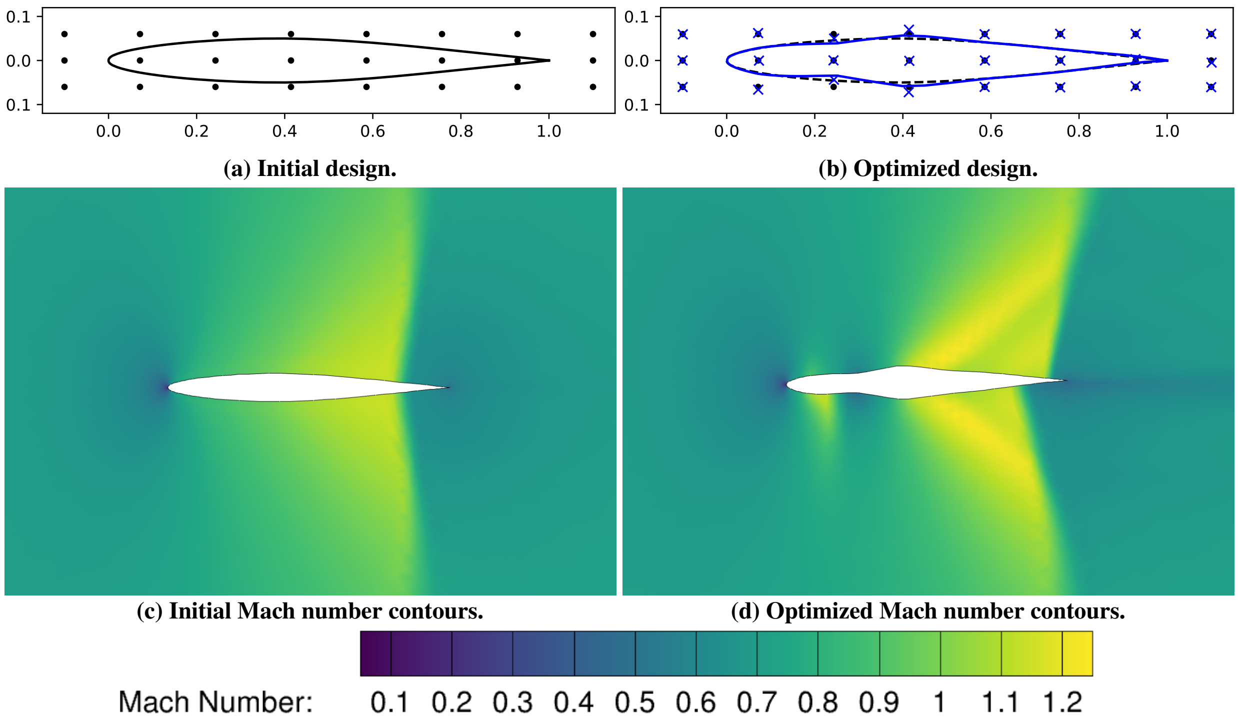


Fig. 7 Airfoil shape and Mach number contours for the Isogai airfoil optimization problem.

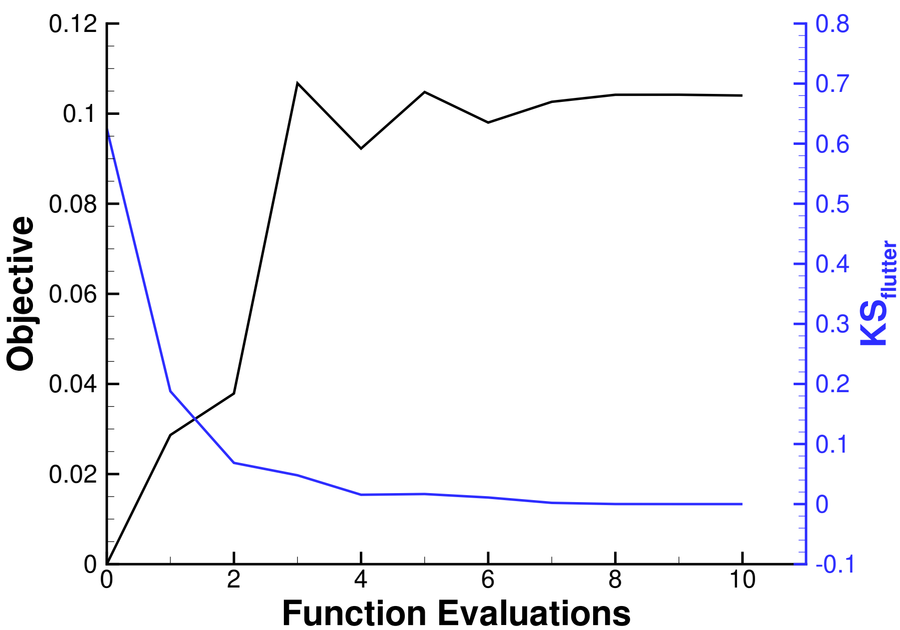


Fig. 8 Isogai airfoil optimization history.

Table 3 Design variables for the AGARD 445.6 Optimization.

Variable Type	Quantity
Structural Thickness	100
Wing Sweep	1
Wing Span	1
Root Chord Length	1
Tip Chord Length	1

model 3 from Ref. 18. For the modal decomposition of the structure, six modes are kept, and each mode is assigned mass-proportional and stiffness-proportional damping of 2.0 and 0.0005, respectively. There are 11 input frequencies in the LFD analysis making 66 LFD linear problems per function evaluation.

The objective of the optimization is to minimize the mass of the wing while maintaining the minimum allowable flutter speed and a fixed planform area, A

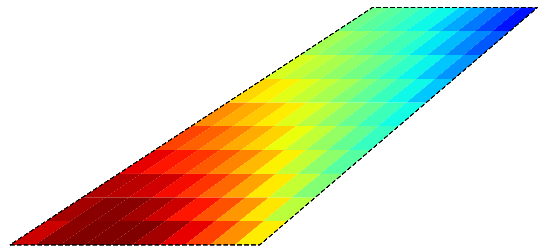
$$\begin{aligned} \text{min} \quad & \quad mass \\ \text{subject to:} \quad & KS_{flutter} \leq 0 \\ & A = A_0. \end{aligned} \tag{13}$$

The minimum allowable flutter dynamic pressure was selected to be $q^* = 2.0$ psi and $g^* = -0.25$ rad/s. There are 100 structural thickness variables and 4 geometric design variables affecting the planform in Table 3. The geometric design variables affect both the structural and aerodynamic model. An additional constraint for the smoothness of the variation of the thickness variables is enforced as part of the analysis process as a form of manufacturing consideration; an OpenMDAO component applies a radius-based smoothing to the thickness design variables before passing them to the structural solver components. The thickness distributions in the figures for this section show the distribution after the radial smoothing is applied. The initial design is shown in Fig. 9a and the resulting DLM flutter response is given in Fig. 10a. The initial design has a hard flutter point around $q = 0.66$ psi, which is well within the keep out zone.

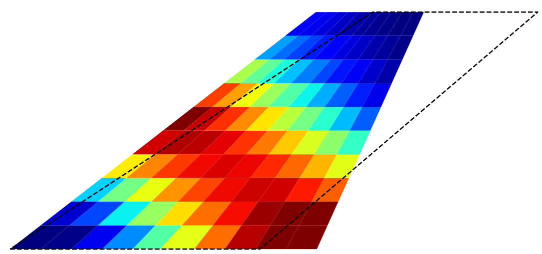
Since the AGARD 445.6 wing is initially symmetric and the geometric design variables cannot make it asymmetric, a symmetry assumption can be maintained throughout the optimization. Therefore, the optimizations are performed with the static aeroelastic loop in Fig. 1 replaced by a single mesh deformation to account for geometric changes and a single steady flow solver call to save computational cost.

The optimization histories of the objective and the flutter constraint for both the LFD and DLM optimizations are given in Fig. 11. The DLM optimization reached the optimizer's convergence tolerance of 10^{-4} after 37 total function evaluations and 34 gradient evaluations. The LFD optimization converged in fewer steps, requiring 24 function evaluations and 19 gradient evaluations. As expected for this flight condition where both aerodynamic models are applicable, the final designs for the DLM in Fig. 9b and LFD in Fig. 9c are very similar. Both versions have optimized wings that push the root chord and wing sweep to the design variable upper bounds, and both have thickness distributions that create a stiff section from the root trailing edge that shifts upstream as it moves outboard. The LFD design has reduced the span of the wing slightly while the DLM version has opted to reduce the tip chord to the design variable lower bound requiring the same span to maintain the planform area. The LFD design also has slightly higher thickness in the stiff section leading to the higher final mass by 0.007 lb in Fig. 11. These minor differences are due to the differences in aerodynamic fidelity. The real parts of the aeroelastic eigenvalues are shown in Fig. 10. Both the LFD and DLM optimizations have a softer flutter instability (lower slope at the zero crossing point) that occurs in such a way that the eigenvalue trace nearly touches the boundary of the keep out zone around the minimum allowable flutter dynamic pressure. There is a small gap between the failure boundary because of the conservatism of the KS aggregation function. The LFD result also has an aeroelastic divergence (the purple curve in 10c) that occurs at $q = 2.29$ psi because both the real and imaginary part (not shown) are zero, but this is not critical for the constraint because it is sufficiently above the chosen minimum allowable flutter dynamic pressure.

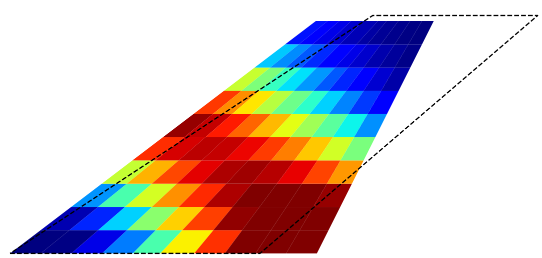
The total time to complete the DLM optimization was 43.5 minutes on a single Skylake CPU core. The LFD version completed in 33.1 hours with 2 dual socket Skylake nodes (80 cores total). For the DLM version, the primary cost of the forward analysis in Table 4 is the p-k solver as it sweeps over dynamic pressures and inputs reduced frequencies in search of valid solutions to Eq. 1. The p-k solver performs mode tracking, which is a non-negligible amount of time as well. For the gradient evaluation, the p-k solver is not as expensive because it doesn't have to track modes or sweep over



(a) Initial design.



(b) Final DLM design.



(c) Final LFD design.

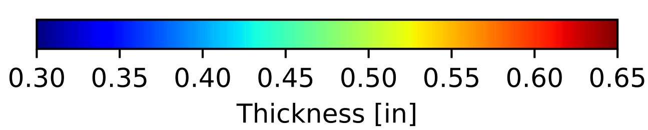
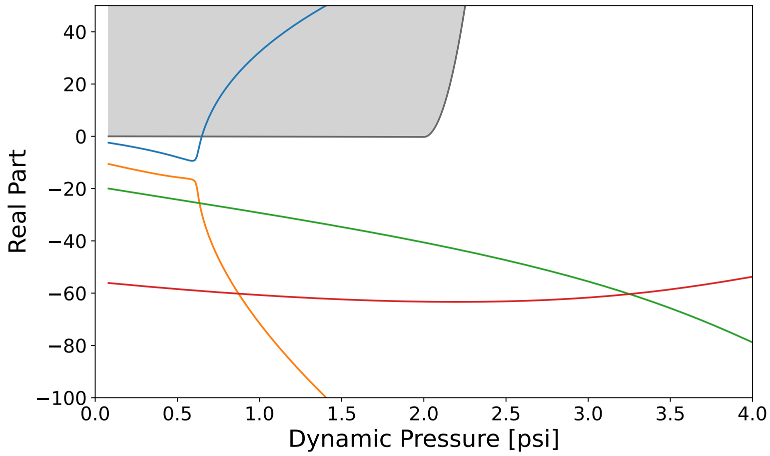
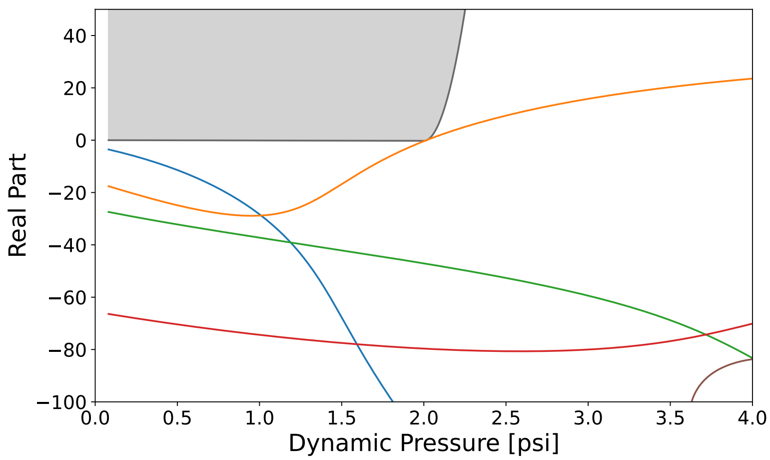


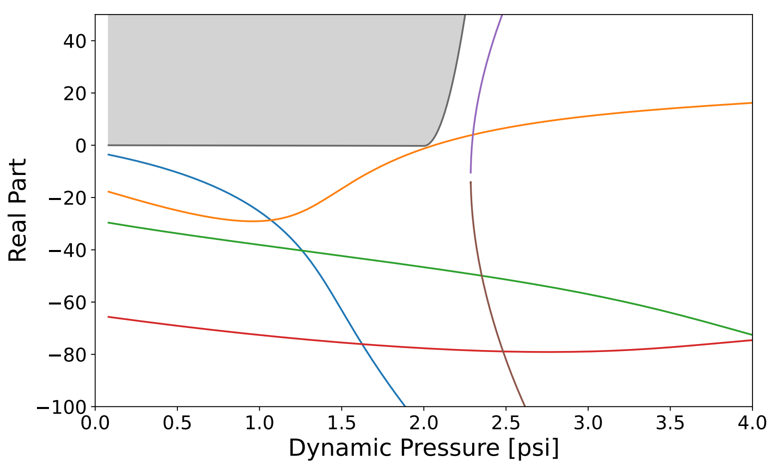
Fig. 9 AGARD 445.6 planform and thickness distributions. The flow direction is left to right.



(a) Initial design.



(b) Final DLM design.



(c) Final LFD design.

Fig. 10 Predicted AGARD 445.6 aeroelastic eigenvalues.

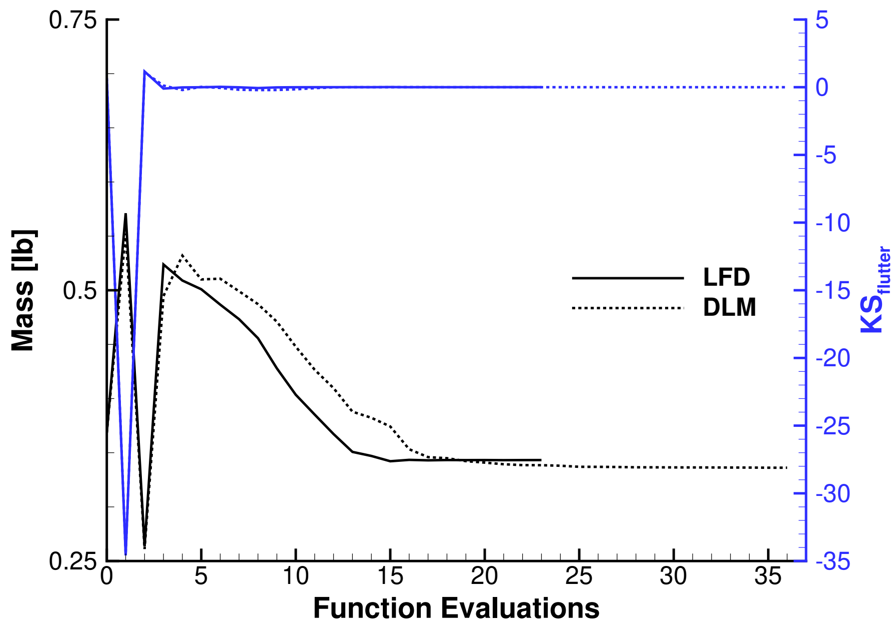


Fig. 11 AGARD 445.6 optimization histories.

Table 4 Average time spent in each component per evaluation for the AGARD 445.6 with DLM.

Component	Function Evaluation [s]	Gradient Evaluation [s]
Modal decomp.	0.36	43.98
Mode transfer	0.01	0.18
DLM	2.22	4.04
p-k flutter	22.36	0.05

parameters to compute sensitivities. The modal decomposition sensitivities are an expensive aspect due to the need to compute the eigenvector (mode shape) derivatives via Nelson's method. For the LFD optimization, the CFD is again by far the largest cost in Table 5. Compared to the Isogai case, the relative cost of linearized aerodynamics per the function evaluation is higher due to there being more structural modes and therefore more LFD linear problems to solve; also, the symmetry assumption resulting in an ommitted static aeroelastic subproblem has reduced the relative cost of the mesh deformation and steady flow solver per function evaluation. The cost of LFD analysis in the gradient evaluation is less than that in the function evaluation because the lack of a coupling loop in the model means that OpenMDAO only queries the LFD component for linearizations once per gradient evaluation as opposed to the three times discussed for the Isogai.

V. Conclusions

This work is an initial attempt to apply the linearized frequency-domain CFD solver in FUN3D/SFE to flutter-constrained optimization. The LFD solver and flutter constraint and related sensitivities have been implemented as OpenMDAO components to improve modularity and automate the computation of coupled sensitivities. The flutter constraint was successfully applied to two optimization problems. The first optimization was performed with the Isogai airfoil at transonic conditions. This optimization minimized the geometric changes of the airfoil in order to increase the minimum flutter point of the airfoil to a target speed index. The second optimization found the minimum mass of the AGARD 445.6 wing subject to a similar flutter constraint. The AGARD 445.6 optimization demonstrated the modularity afforded by the OpenMDAO integration; the optimization was performed with both LFD- and DLM-based flutter analysis with the only changes to the optimization problem being the aerodynamic inputs. This is useful for multifidelity optimization activities in the future as well as testing the optimization with the less expensive DLM models

Table 5 Average time spent in each component per evaluation for the AGARD 445.6 with LFD.

Component	Function Evaluation [s]	Gradient Evaluation [s]
Modal decomp.	0.61	47.19
Mesh deform	2.83	1.87
Flow Solver	223.98	41.96
Mode transfer	0.02	0.08
LFD	3140.85	1711.76
p-k flutter	23.77	0.01

before attempting the LFD version in this work. While the two aerodynamic models produced similar optimized designs for the AGARD 445.6 with the DLM having a much lower cost, the Isogai optimization demonstrated that LFD can be applied to optimize at transonic conditions where DLM is insufficient. With the process successfully demonstrated on these small cases, future work will look to improve the computational efficiency in order to apply LFD-based flutter constraints to large meshes and viscous problems.

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 is provided by the NASA Langley K cluster.

References

[1] 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.

[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] 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.

[4] Mani, K., and Mavriplis, D. J., "Adjoint-Based Sensitivity Formulation for Fully Coupled Unsteady Aeroelasticity Problems," *AIAA Journal*, Vol. 47, No. 8, 2009, pp. 1902–1915.

[5] 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.

[6] Zhang, Z., Chen, P.-C., Wang, Q., Zhou, Z., Yang, S., and Wang, Z., *Adjoint-based Structure and Shape Optimization with Flutter Constraints*, American Institute of Aeronautics and Astronautics, 2016. doi:doi:10.2514/6.2016-1176.

[7] Thomas, J., and Dowell, E. H., "Discrete Adjoint Constrained Design Optimization Approach for Unsteady Transonic Aeroelasticity and Buffet," *AIAA Aviation Forum*, Virtual, 2020.

[8] 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.

[9] Prasad, R., and Choi, S., "Aerodynamic Shape Optimization for Flutter/LCO based design using Coupled Adjoint," *58th AIAA Aerospace Sciences Meeting*, Orlando, Florida, 2020.

[10] 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.

[11] Clark, W. S., and Hall, K. C., "A Time-Linearized Navier–Stokes Analysis of Stall Flutter," *Journal of Turbomachinery*, Vol. 122, No. 3, 1999, pp. 467–476.

[12] 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.

[13] 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.

[14] Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *55th AIAA Aerospace Sciences Meeting*, 2017.

[15] Anderson, W. K., and Newman, J., "High-Order Stabilized Finite Elements on Dynamic Meshes," *2018 AIAA Aerospace Sciences Meeting*, 2018.

[16] Jacobson, K. E., Stanford, B. K., Wood, S. L., and Anderson, W. K., "Adjoint-based Sensitivities of Flutter Predictions based on the Linearized Frequency-domain Approach," *59th AIAA Aerospace Sciences Meeting*, Virtual Event, 2021.

[17] Isogai, K., "On the Transonic-Dip Mechanism of Flutter of a Sweptback Wing," *AIAA Journal*, Vol. 17, No. 7, 1979, pp. 793–795.

[18] Yates, C. E., "AGARD Standard Aeroelastic Configurations for Dynamic Response. Candidate Configuration I - Wing 445.6," Tech. Rep. TM-100492, NASA, August 1 1987.

[19] 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.

[20] 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.

[21] Lambe, A. B., and Martins, J. R. R. A., "Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes," *Structural and Multidisciplinary Optimization*, Vol. 46, 2012, pp. 273–284. doi:10.1007/s00158-012-0763-y.

[22] Irons, B. M., and Tuck, R. C., "A Version of the Aitken Accelerator for Computer Iteration," *International Journal for Numerical Methods in Engineering*, Vol. 1, No. 3, 1969, pp. 275–277.

[23] Nelson, R. B., "Simplified Calculation of Eigenvector Derivatives," *AIAA Journal*, Vol. 14, No. 9, 1976, pp. 1201–1205.

[24] Nielsen, E. J., and Anderson, W. K., "Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, 2002, pp. 1155–1163.

[25] 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.

[26] Kiviaho, J. F., and Kennedy, G. J., "Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares," *AIAA Journal*, Vol. 57, No. 5, 2019, pp. 2237–2243.

[27] Stanford, B. K., "Role of Unsteady Aerodynamics During Aeroelastic Optimization," *AIAA Journal*, Vol. 53, No. 12, 2015, pp. 3826–3831.

[28] Kreisselmeier, G., and Steinhauser, R., "Systematic Control Design by Optimizing a Vector Performance Index," *International Federation of Active Controls Symposium on Computer-Aided Design of Control Systems*, Zurich, Switzerland, 1979.

[29] Kennedy, G. J., and Hicken, J. E., "Improved Constraint-Aggregation Methods," *Computer Methods in Applied Mechanics and Engineering*, Vol. 289, 2015, pp. 332–354.

[30] 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.

[31] Anderson, W. K., Newman, J. C., Whitfield, D. L., and Nielsen, E. J., "Sensitivity Analysis for Navier-Stokes Equations on Unstructured Meshes Using Complex Variables," *AIAA Journal*, Vol. 39, No. 1, 2001, pp. 56–63. doi:10.2514/2.1270, URL `https://doi.org/10.2514/2.1270`.

[32] Martins, J., Sturdza, P., and Alonso, J., "The Complex-step Derivative Approximation," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 29, No. 3, 2003, pp. 245–262.

[33] Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *SIGGRAPH Computer Graphics*, Vol. 20, 1986, pp. 151–160.

[34] Kraft, D., "A Software package for Sequential Quadratic Programming," Tech. Rep. DFVLR-FB 88-28, Köln, Germany, 1988.