GRADIENT-BASED OPTIMIZATION OF THE COMMON RESEARCH MODEL WING SUBJECT TO CFD-BASED GUST AND FLUTTER CONSTRAINTS

Andrew S. Thelen,[*] Kevin E. Jacobson,[†] Bret K. Stanford[‡1]

[1]NASA Langley Research Center, Hampton, VA, 23681

Keywords: Optimization, Flutter, Gust, Linearized Frequency Domain, Doublet Lattice Method

Abstract: The linearized frequency-domain method was recently implemented in the stabilized finite element solver in NASA's FUN3D code. Previous work by the authors used this method for enforcing flutter constraints during gradient-based optimizations. More recently, the solver was expanded to account for continuous (also known as stochastic) gust responses. This paper expands on recent Common Research Model wing optimization work, which demonstrated gradient-based optimization with flutter and stochastic gust constraints, among others. While that work utilized FUN3D for static aeroelastic solutions but relied on doublet lattice aerodynamics for gust and flutter responses, the present work replaces these unsteady aerodynamic analyses with those of FUN3D's linearized frequency-domain solver. With analytic derivatives available, gradient-based optimization is performed through the use of the OpenMDAO/MPhys libraries with over 700 shape, structural, and aerodynamic design variables and over 10 nonlinear constraints. Comparisons of analysis results and optimized designs are made between doublet lattice and linearized frequency-domain solutions.

1 Introduction

Flutter and gust responses must be taken into account when certifying a new aircraft. When optimizing an aircraft, it is therefore useful to enforce nonlinear constraints on gust and flutter responses. Historically, this has typically been done using panel-based methods, namely the Doublet Lattice Method (DLM), often corrected using limited Computational Fluid Dynamics (CFD) or experimental data. Higher fidelity CFD has also seen limited use for aeroelastic constraints during optimization. These methods, while in theory more accurate than panel-based methods, require orders of magnitude more time to evaluate, especially when using time-domain CFD. To reduce the cost of these unsteady CFD calculations, the Linearized Frequency-Domain (LFD) method presumes that the flow field is harmonic and linearized about a nonlinear state, allowing the flow to be solved for a small set of frequencies rather than marching through time.

LFD has recently been implemented in the Stabilized Finite Element (SFE) solver [1, 2] in the NASA FUN3D code [3–6]. This has been used for enforcing flutter constraints during gradient-based optimizations [7] of the weakened AGARD 445.6 wing. More recently, the LFD solver was expanded in Ref. [8] to account for continuous (also known as stochastic) gust

[*] Research Aerospace Engineer, Aeroelasticity Branch, andrew.s.thelen@nasa.gov.

[†] Research Aerospace Engineer, Aeroelasticity Branch.

[‡] Research Aerospace Engineer, Aeroelasticity Branch.

responses [9, 10], which analyze how a structure reacts to typical atmospheric velocity variations during flight. These aeroelastic responses, and their analytic derivatives, have been implemented with the help of the open-source MPhys [11, 12] Python library for OpenMDAO [13]. This software library, a collaborative effort by government, academia, and industry groups, seeks to standardize multi-physics analyses in OpenMDAO. Using these standards, so-called MPhys “builders” for aerodynamics, structures, load/displacement transfer, and geometry can be used to set up a coupled aeroelastic OpenMDAO problem. Due to the modularity of these builders, solvers can be interchanged easily at the top level. This, for instance, allows aeroelastic design problems to be tested with panel-based aerodynamics before swapping these lower fidelity aerodynamics with CFD. In addition to providing modularity for discipline coupling, a recent addition to MPhys, so-called “remote components”, provides an automated way to evaluate computationally expensive OpenMDAO problems remotely on High-Performance Computing (HPC) jobs.

While the LFD gust and flutter optimization work was originally demonstrated on the AGARD wing [7, 8], recent work by the authors [14] sought to work towards a more industry-relevant design case, which was chosen to be the NASA Common Research Model (CRM) wing [15]. This recent work focused on typical CRM design cases in the literature, with the addition of the gust and flutter constraints. Static aeroelastic analyses used either an in-house Vortex Lattice Method (VLM) or FUN3D coupled to an in-house structural solver (PyShell) via the Matching-based Extrapolation of Loads and Displacements (MELD) transfer scheme [16]. In this previous work, unsteady aerodynamics (coupled to PyShell via MELD) during the optimizations were only provided by an in-house DLM. The present work replaces these DLM-based aerodynamics with those of the FUN3D LFD solver.

This paper is organized as follows. First, Sect. 2 describes the optimization problem of interest and other analysis details, including the use of remote components. Section 3 provides results of the optimization problems. This includes convergence metrics, flutter and gust analysis results at the optima, and comparisons of LFD and DLM aerodynamics at the same design points. The paper ends with conclusions and comments on future work in Sect. 4.

2 Methodology

This section provides details on methodology used in the optimization problems. Section 2.1 provides details on the optimization problem, while Sect. 2.2 describes the use of remote components.

2.1 Design problem definition

The current design problem [14] is based on CRM optimization work in the literature. Table 1 provides parameters used for the mission definition, as well as atmospheric conditions and parameters that affect fuel and total mass. In this work, the goal is to optimize the wing jig shape for minimum fuel burn of a typical transport aircraft mission. A steady level cruise flight condition (Mach 0.85 at 37,000 ft altitude) is included, as well as a Mach 0.64, 2.5g pull up maneuver at sea level, where a stress constraint is enforced. While other literature has included a -1g push-down maneuver as well, this is excluded in the present work to reduce computational cost and because the pull-up maneuver is the stronger design driver. Various other constraints are included, including some related to geometry such as planform area, stiffeners via stiffness smearing [17], and spar depth. Other constraints enforce static aeroelastic loads (i.e., load factor) and consistency (i.e., fuel mass used in the structural analysis must equal computed fuel mass based on the particular mission). Table 2 provides the list of design variables and nonlinear constraints used for the optimization. The optimizer chosen is the Sequential Least Squares Programming (SLSQP) algorithm in the pyOptSparse [18] SciPy wrapper with a tolerance of 1×10^{-5}.

Table 1: General parameters used for fuel and weight calculations, as well as atmospheric conditions.

Category	Description	Value	Units
Fuel and weight	Design range	14,306.7	km
	Specific fuel consumption	0.53	kg/(kg$_f$×s)
	Fuel density	810	kg/m^3
	Reserve fuel mass	15,000	kg
	Fixed weight	1.4×10^6	N
	Cruise fuel used	50	% of mission fuel burn
	Maneuver fuel used	100	% of mission fuel burn
Atmospheric conditions	Cruise Mach number	0.85	-
	Cruise altitude	37,000	ft
	Maneuver Mach number	0.64	-
	Maneuver altitude	Sea level	-
	Flutter Mach number	0.9	-
	Flutter altitude	37,000	ft

Table 2: Objective, design variables, and constraints for the optimization problems.

	Description	Specification	Quantity
minimize:	Cruise fuel burn		
with respect to:	Root chord differential	$\in [-3, 3]$ m	1
	Tip chord differential	$\in [-1, 1]$ m	1
	Span differential	$\in [-10, 10]$ m	1
	Sweep differential	$\in [-10, 10]$ deg	1
	Wing thickness	$t_{wing} \geq 0.5 \times t_{wing,0}$	5
	Wing twist differential	$\in [-10, 10]$ deg	5
	Fuel scaler	$\in [0, 1]$	1
	Shell thickness	$\in [0.003, 0.03]$ m	240
	Stiffener thickness	$\in [0.003, 0.03]$ m	240
	Stiffener height	$\in [0.03, 0.15]$ m	240
	Static angle of attack	$\in [-10, 10]$ deg	3
	Total		**738**
subject to:	Fuel mismatch	$= 0$	1
	Static load factor	$\eta = [1, 2.5, 1]^T$	3
	Wing area	$A \geq 205.8$ m^2	1
	Available fuel mass	$\geq 63,464.28$ kg	1
	Trailing edge spar depth	≥ 0.19 m	1
	Stiffener aspect ratio	$KS_{AR} \leq 1$	1
	Stiffener vs. shell thickness	$KS_{t/t_s} \leq 1$	1
	Pull-up stress	$1.5 KS_\sigma \leq 1$	1
	Transonic flutter	$KS_{fl} \leq 0$	1
	Cruise gust	$\leq 1.6 \times 10^7$ N-m	1
	Total		**12**

Compared to Ref. [15], the present work adds constraints on flutter and stochastic gust responses. The authors previously demonstrate this addition using DLM-based aerodynamics [14]. Flutter constraints are computed using a Kreisselmeier-Steinhauser (KS) [19] aggregate of modal damping, as computed by a p-k method implementation. Meanwhile, stochastic gust constraints are based on steady root bending moment combined with the gust-induced unsteady Root Mean Square (RMS), where the von Kàrmàn gust spectrum [10] is used to model atmospheric velocity perturbations. The upper limit of the combined root bending moment is explained in Thelen et al. [14]. A discrete gust constraint was excluded from the present results as it was inactive throughout an optimization that utilized panel aerodynamics.

Table 3 lists the three optimization cases to be demonstrated. The new optimization case, labeled as “FUN3D”, has the same objective function and constraints as the “FUN3D+DLM” and “VLM+DLM” cases shown in Thelen et al. [14], except with the flutter and gust constraints utilizing LFD instead of DLM. While the gust constraint is still computed about the Mach 0.85 cruise condition, the updated flutter constraint is computed at cruise flight conditions with Mach number increased to 0.9. This adds an additional equality constraint (load factor must equal one) and design variable (angle of attack) not included in the DLM-based optimizations. This change was made to ensure that the flutter constraint is active at the optimum for demonstrative purposes, as the constraint was inactive in the FUN3D+DLM optimization and transonic effects seem to be relatively weak at the baseline design [20]. Gust and flutter LFD analyses are both computed about statically deflected geometries. Gust analyses include pitch and plunge rigid body modes, whereas the flutter analyses do not; these are excluded because the present p-k method implementation cannot currently handle a Phugoid mode instability. Both analyses use 9 modes (2 rigid body and 7 flexible for gust, 9 flexible modes for flutter) and 11 reduced frequencies. All FUN3D analyses are currently limited to Euler equations solved on a fairly coarse mesh with 60,742 nodes. Cruise drag is augmented with empirical profile drag to help account for viscosity in the fuel burn calculation. Computing RANS-based steady and unsteady aerodynamics on a better resolved mesh is an area of future work.

2.2 Use of remote components

With the analyses being implemented with OpenMDAO/MPhys, the FUN3D-based optimizations also make use of a new addition of the MPhys library, so-called remote components. This class of component, derived from the OpenMDAO ExplicitComponent class, allows an OpenMDAO problem to be evaluated remotely on an HPC compute node, replicating inputs and outputs locally. This, for instance, provides the ability to launch an optimization in the background on an HPC login node, which evaluates the computationally expensive analyses on HPC compute nodes as needed. In a client-server arrangement of nested OpenMDAO problems, these OpenMDAO components use the ZeroMQ library [21] and ssh port forwarding to communicate (via

Table 3: Summary of the optimization problems to be compared.

Description	Steady aerodynamics	Unsteady aerodynamics	Transonic flutter constraints
FUN3D	FUN3D Euler*	FUN3D Euler LFD	Mach 0.9, cruise altitude, load factor of 1
FUN3D+DLM	FUN3D Euler*	DLM	Mach 0.85
VLM+DLM	VLM	DLM	Mach 0.85

*augmented with empirical profile drag to approximately account for viscosity encoded JSON dictionaries) with the server-side analyses; meanwhile, pbs4py[1] is used for HPC job management. As the optimization progresses, function and gradient evaluation times are logged so that the HPC jobs can be relaunched when remaining job time is estimated to be too short for an additional design evaluation.

In addition, the optimization is launched using a small number of processors so that separate flight conditions may be evaluated in parallel on separate HPC jobs (via the ParallelGroup OpenMDAO class). For the FUN3D+DLM optimization, this involved evaluating cruise and pull-up maneuver conditions using 15 compute nodes each, while the DLM evaluations are performed on a third job on one compute node. For the FUN3D optimization, cruise, maneuver, and flutter flight conditions were evaluated on 10 nodes each. Overall, this allowed a single optimization to span numerous HPC jobs running in parallel, with the total number of nodes exceeding the allowed maximum for a single job, without having to manually restart the optimization each time an HPC job expired.

3 Results

This section summarizes results of the optimization problems. First, convergence metrics are given for the three cases: FUN3D, FUN3D+DLM, and VLM+DLM. Then, gust and flutter responses are examined at the optimal designs. Finally, comparisons are made between DLM- and LFD-based gust and flutter responses.

3.1 Optimization convergence metrics

Figure 1 shows convergence metrics of the three optimization cases, including the objective function, maximum constraint violation, and first-order optimality [14,22]. The FUN3D+DLM and VLM+DLM results, originally shown in Thelen et al. [14], are plotted against the new FUN3D result in black. In these plots, note that the FUN3D+DLM result was started from the VLM+DLM optimum in a effort to reduce the number of iterations. This effort was largely ineffective, likely because the VLM+DLM and FUN3D+DLM optimal designs are too far apart. The FUN3D case, meanwhile, was started from the baseline design; presumably, starting from the FUN3D+DLM optimum may have reduced the required number of iterations by a larger amount because the optima are more similar (as suggested by planforms shown later).

For reference, Fig. 2 shows convergence of all constraints. Constraint convergence is further categorized in Fig. 3, 4, and 5. Figure 3, for instance, illustrates which normalized constraints

[1] Available at https://github.com/nasa/pbs4py

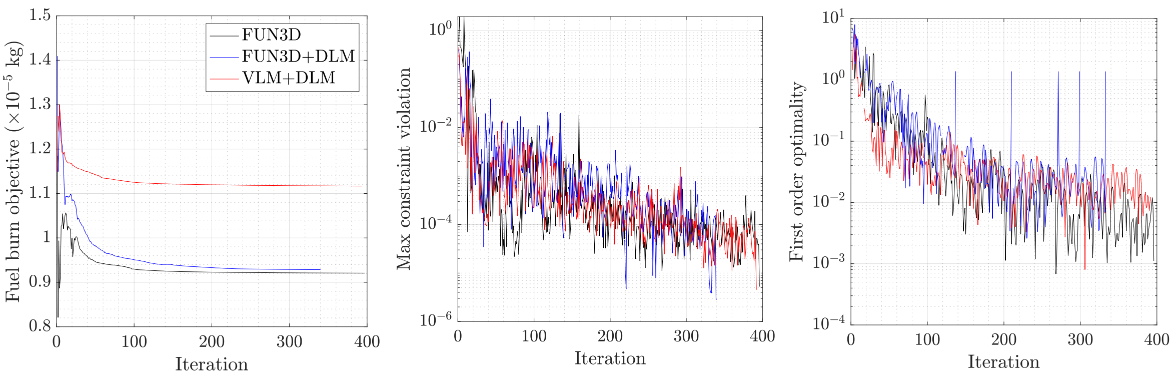

Figure 1: Optimization convergence of the fuel burn objective (left), maximum constraint violation (center), and first-order optimality (right).

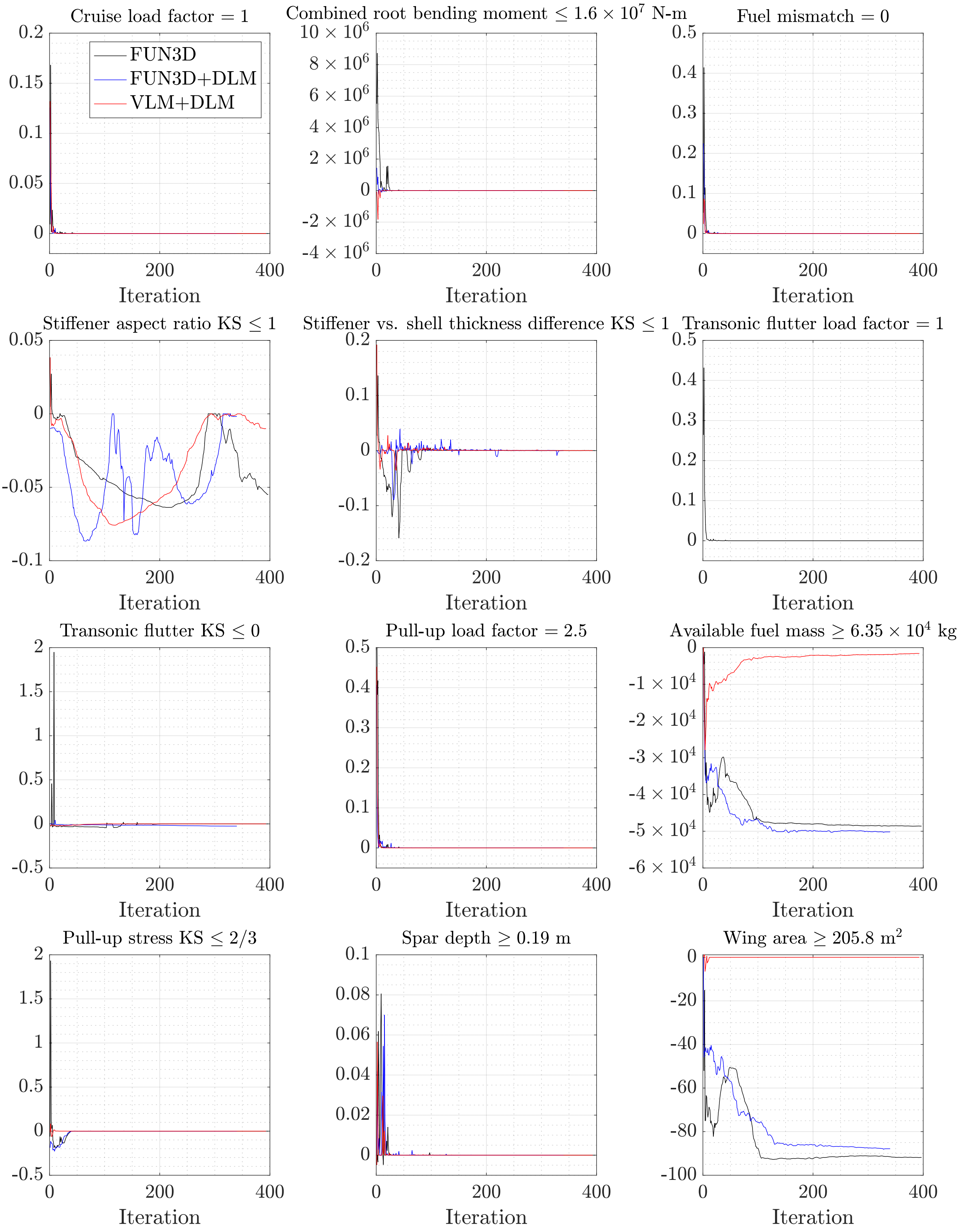

Figure 2: Constraint convergence, where inequality constraints must be less than or equal to zero while equality constraints must be zero. Black, blue, and red lines are the FUN3D, FUN3D+DLM, and VLM+DLM optimization cases, respectively.

are violated at a given design evaluation, with the threshold being the optimization constraint tolerance of 1×10^{-5} (i.e., which constraints are greater than 1×10^{-5}). Similarly, Fig. 4 illustrates which normalized constraints are currently active (i.e., on the cusp of being violated and likely driving the design process) with a somewhat arbitrary threshold of -1×10^{-4}. Figure 5 shows which normalized constraint is currently the largest, thus indicating which constraint may be the most difficult to satisfy.

Shown in Fig. 1, the FUN3D case obtained a lower fuel burn than the FUN3D+DLM case. This is perhaps surprising, as the FUN3D+DLM flutter constraint is inactive while the FUN3D one is active. Shown later in Sect. 3.3, this is likely due to DLM predicting a larger gust RMS than

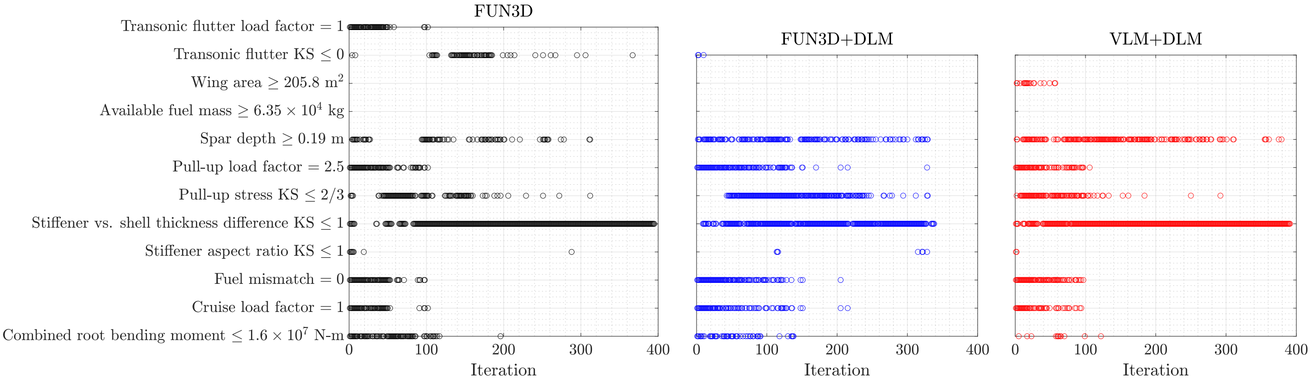

Figure 3: Violated constraints based on a tolerance of 1×10^{-5}.

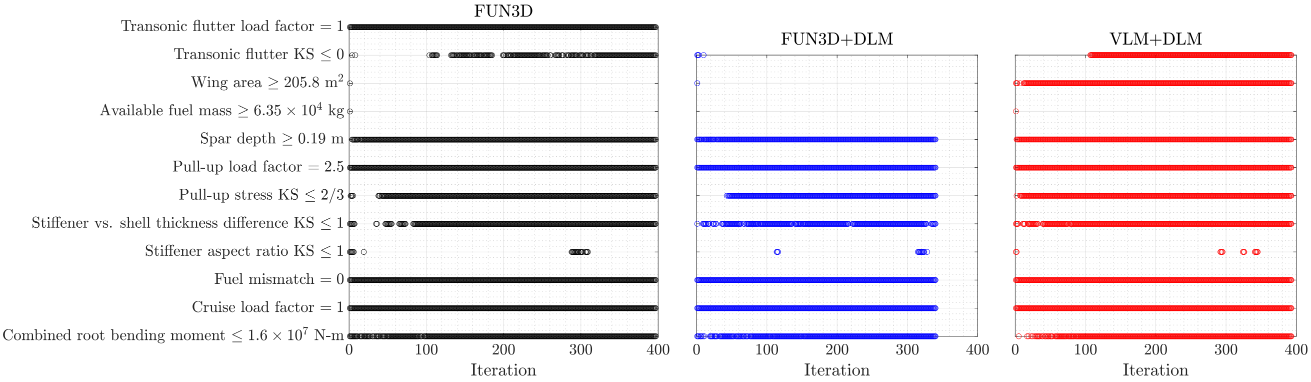

Figure 4: Active constraints with -1×10^{-4} taken to be the threshold.

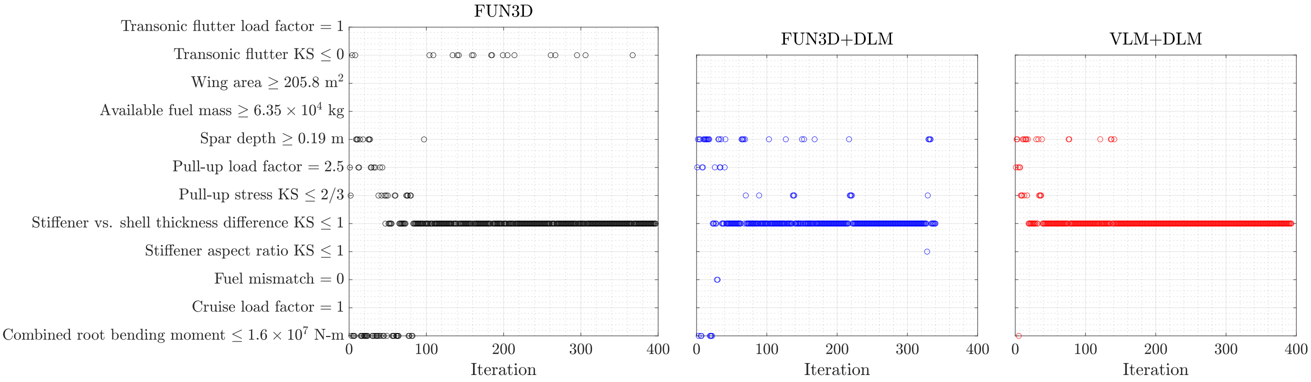

Figure 5: Maximum normalized constraint during optimization convergence.

LFD at the optimum (thus being more restrictive on the design space), which is the opposite trend seen at the baseline design. In the center plot of Fig. 1, the maximum constraint violation is seen to converge to around 1×10^{-5} in 300-400 iterations. Figures 2, 3, and 5 suggest that aeroelastic constraints are mostly well behaved and tend to converge relatively quickly (with the exception of the FUN3D flutter constraint). Conversely, constraints relating to structural geometry, particularly the stiffener vs. shell thickness constraint (and to a lesser extent, spar depth), tend to be the maximum violation the vast majority of the time. This perhaps indicates that they are more difficult to satisfy, potentially due to stronger nonlinearity of the functions, than the more computationally expensive aeroelastic constraints.

Shown in Fig. 1 on the right, optimality is shown to drop by 3-4 orders of magnitude. Here, note that the optimality of the FUN3D+DLM case indicated a numerical issue by having several spikes (with values of around 1). The issue was related to insufficient flow adjoint convergence, which occasionally led to poor gradient accuracy, and has since been resolved by adjusting FUN3D solver settings. A gradient accuracy issue was also suggested by the required function and derivative calls, 341 and 314, respectively, as there were several optimization iterations with poorly converging line searches. For comparison, the FUN3D case required 396 and 391 function and gradient evaluations, while the VLM+DLM case required 392 and 390, respectively.

Perhaps worth noting in Fig. 4 is that the wing area constraint, which serves as a surrogate for ensuring adequate wing volume, is only active for the VLM+DLM case as the FUN3D+DLM and FUN3D cases yield a larger, more swept wing. This can be seen in Fig. 6, which illustrates the baseline and optimized planform shapes. The constraint on available fuel mass, in addition, is inactive for all optimization cases, although it nearly becomes active in the VLM+DLM case as shown in Fig. 2. Nonetheless, in Fig. 6, it is interesting to note how closely the FUN3D and FUN3D+DLM planforms resemble each other. This may indicate that these two designs are much closer to each other than the third optimum; starting the FUN3D case from the FUN3D+DLM optimum may have therefore led to more significant savings when compared to restarting from the VLM+DLM optimum. Physically, this may mean that the FUN3D-computed fuel burn trend (which depends on lift to drag ratio) drives sweep to larger values due to wave

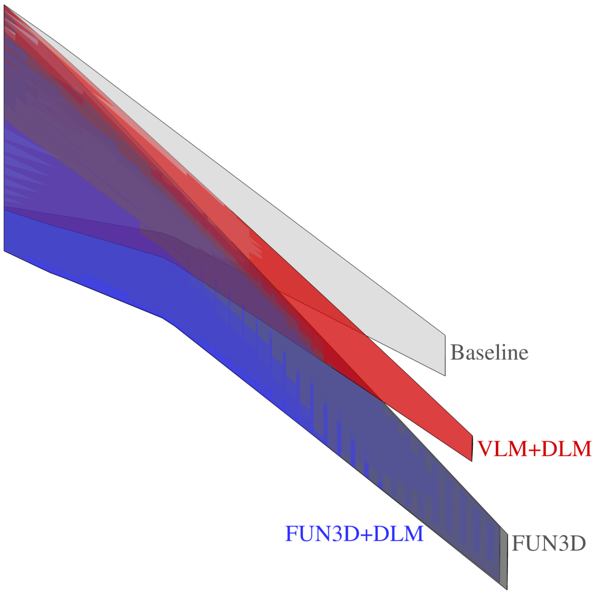

Figure 6: Baseline and optimized planforms.

drag associated with shocks. For these highly swept wings, DLM may be better correlated with LFD due to the lower effective Mach number, which may reduce transonic effects. Other factors are likely at play as well; for instance, the root bending moment would be lower with a shorter span wing, which potentially drove the optimizer from a wing with larger span and less sweep.

3.2 Flutter and gust responses at the optima

Figure 7 shows the p-k flutter result at the FUN3D optimum. The corresponding flutter constraint is active, with modes 1, 3, and 4 all coming close to straying into the black "keep-out" zone (above the black curve being infeasible) in the dynamic pressure-damping space. For reference, Fig. 8 shows the wind-off mode shapes corresponding to these three modes. Here, the first normal mode is in general a first bending mode, mode 2 is generally a second bending

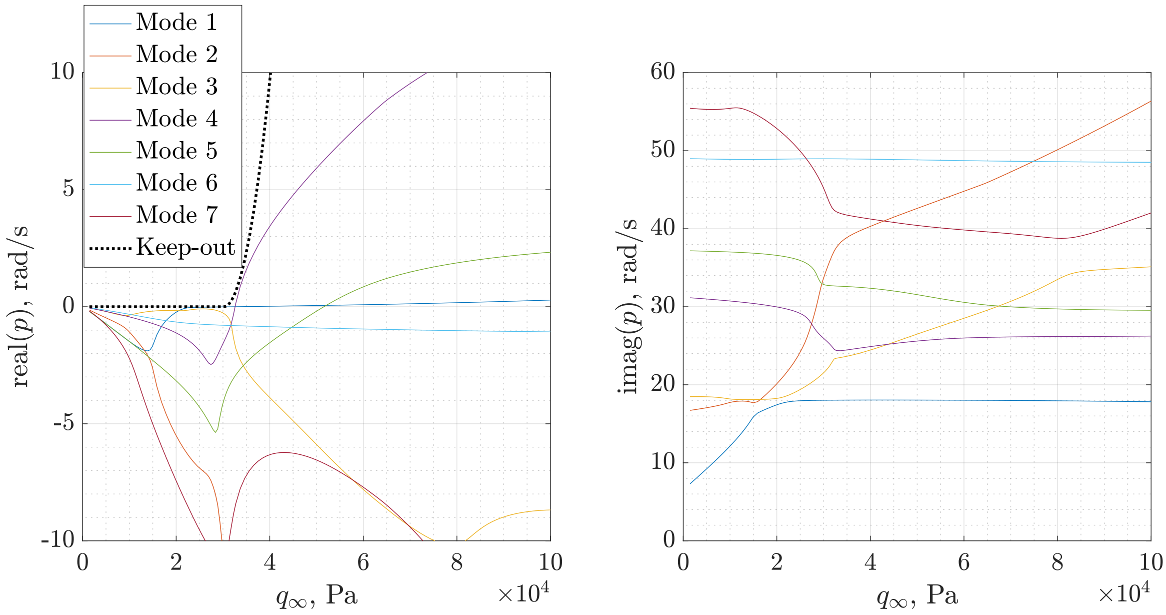

Figure 7: Flutter results at the FUN3D optimum.

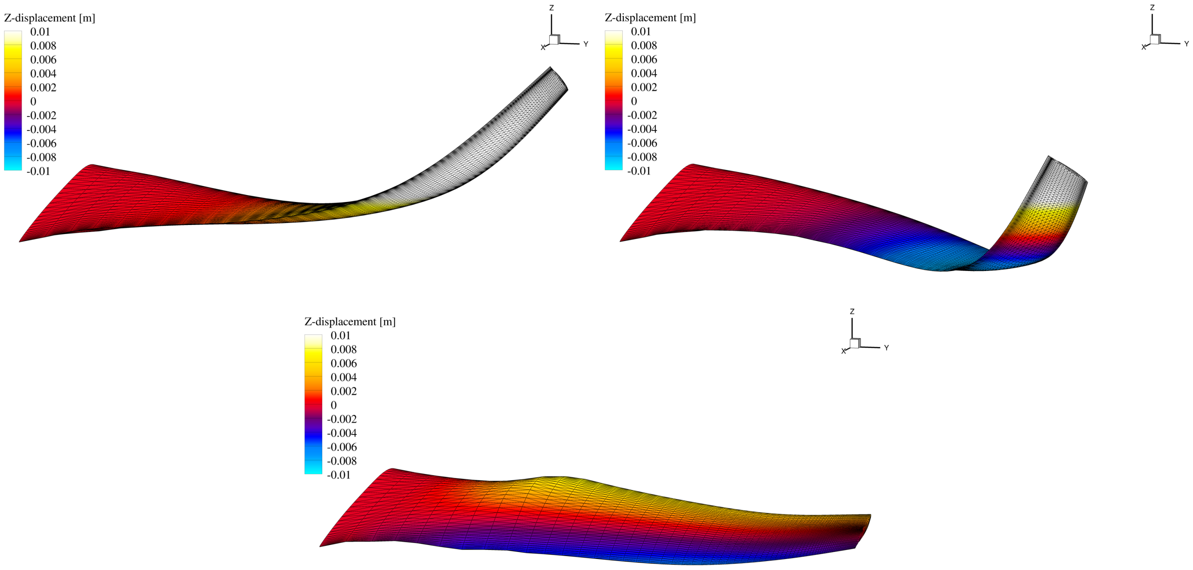

Figure 8: Wind-off mode 1 (left), 3 (right), and 4 (bottom) shapes at the FUN3D optimum, colored by modal z-displacements, where modal displacement about the statically deflected geometry is scaled by 200.

mode with some in-plane motion, while mode 4 is a second torsional mode. While mode 3 never encounters flutter, mode 1 causes a "soft" flutter onset while mode 4 causes a "hard" one. Finally, Fig. 9 shows the corresponding LFD-computed complex pressures and mode shape corresponding to the hard flutter crossing of mode 4. Based on pressure contours, this flutter mechanism is largely driven by oscillatory pressures at the outboard leading edge, particularly near the shock. In addition, while it is the fourth mode that goes unstable in the damping plot, there is in fact a large amount of participation with modes 1 and 2. While not shown, the soft flutter mechanism of mode 1 involves a large amount of in-plane bending.

Figures 10 and 11 show the DLM-computed flutter results at the FUN3D+DLM and DLM optima. The flutter constraint was ultimately inactive for the FUN3D+DLM case, with modes 4 and 5 eventually encountering flutter beyond the dynamic pressure range of interest. The

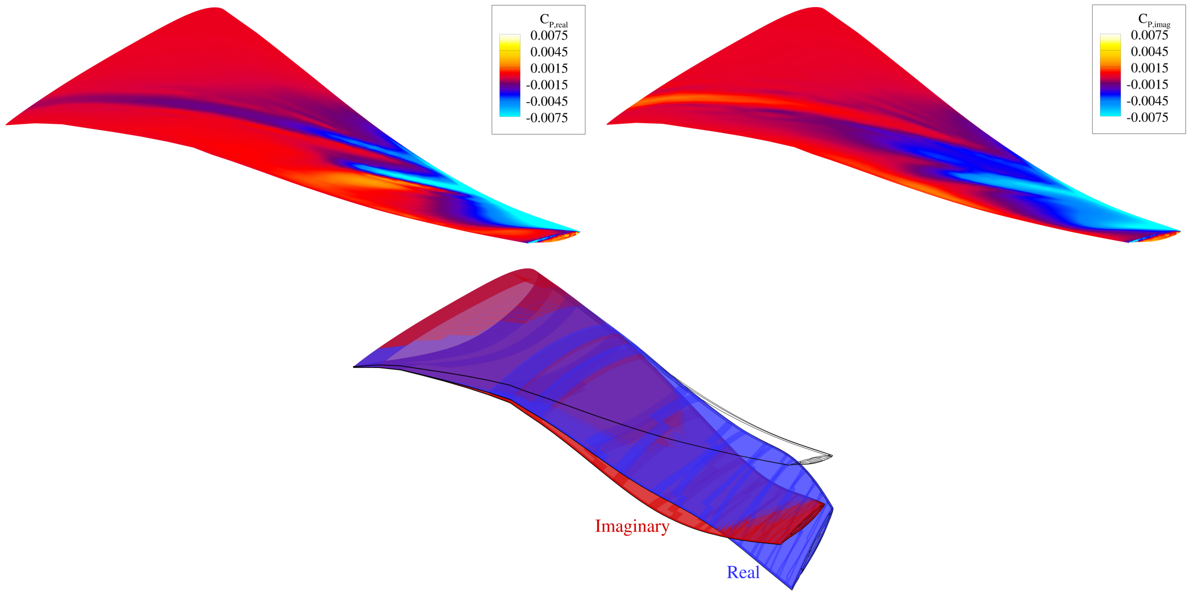

Figure 9: Flutter complex pressure (top) and mode shape (bottom) of the hard flutter crossing at the FUN3D optimum.

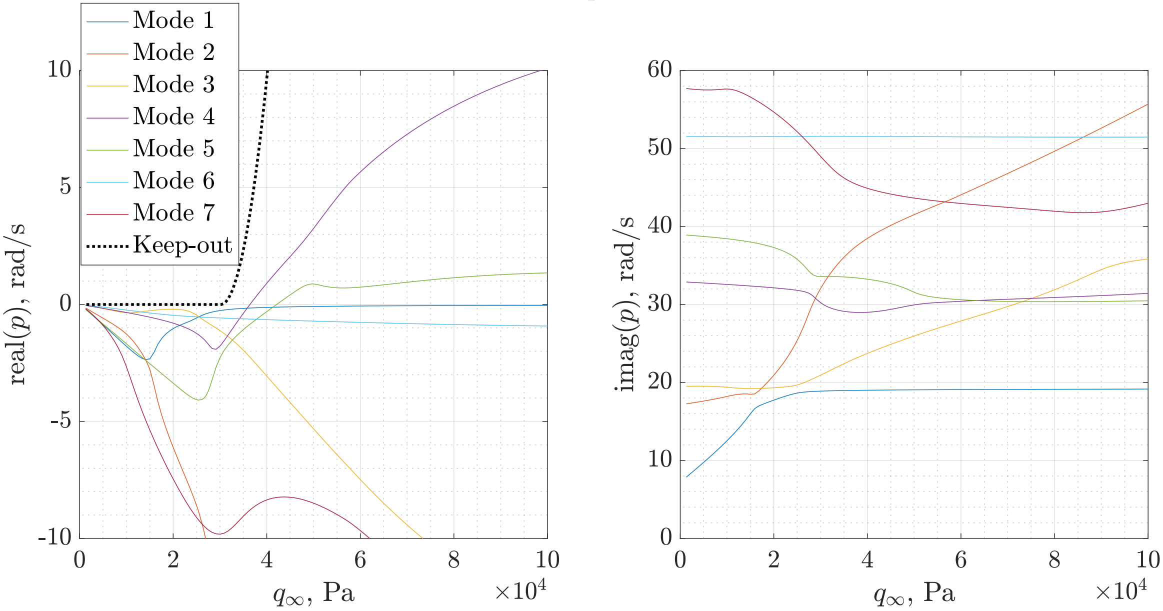

Figure 10: Flutter results at the FUN3D+DLM optimum.

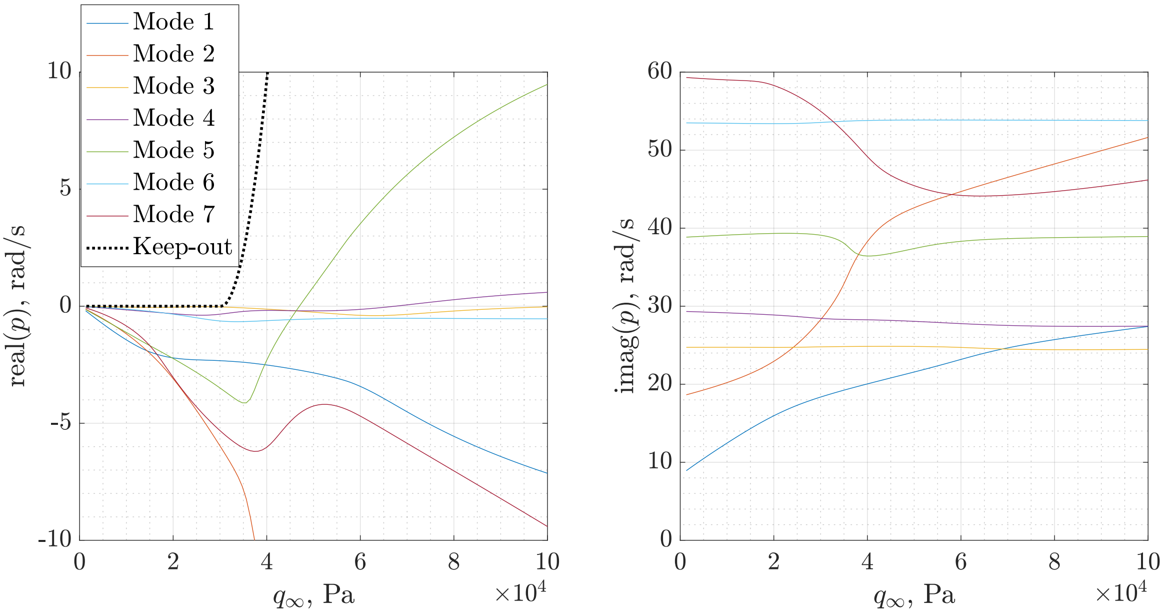

Figure 11: Flutter results at the VLM+DLM optimum.

flutter constraint for the VLM+DLM case was active, but only due to what is perhaps a soft hump mode in mode 3; hard flutter in mode 5 is encountered at a significantly higher dynamic pressure. Note that in these two results, flutter is computed at a Mach number of 0.85. The previous FUN3D result instead used a Mach number of 0.9 (at cruise altitude and load factor of 1) to help ensure an active flutter constraint for demonstrative purposes.

Figure 12 shows a comparison of gust-induced root bending moment Power Spectral Densities (PSDs) at the different optima. With the rigid body modes included in the gust analysis, and with the gust quantity of interest being based on a reaction load (as opposed to tip displacement), the PSD approaches zero at zero frequency. All designs have a peak at roughly the same

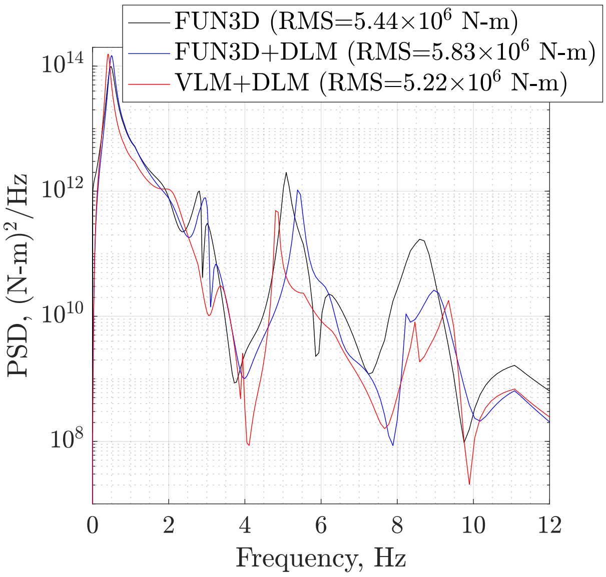

Figure 12: Gust-induced root bending moment PSD at the different optima.

frequency (around 0.5 Hz) as this initial peak is driven by motion in the rigid body modes, with some interaction with the first flexible mode. The FUN3D peak here is smaller than that of the FUN3D+DLM design, which leads to a smaller RMS; the VLM+DLM design, however, has the smallest RMS among them because the first peak at around 0.5 Hz appears to be narrower. Differences at higher frequencies, due to interactions of flexible modes, are less consequential to the RMS, with the next largest peak at roughly 5 Hz being two orders of magnitude smaller. This secondary peak may correspond to an approaching flutter instability; if the gust analysis atmospheric conditions were to approach those of flutter onset, then this gust PSD would grow to infinity at the flutter frequency [14].

3.3 Comparison of DLM and LFD aerodynamics

Flutter and gust analysis results are next compared at the same designs (baseline and FUN3D optimum) but using either LFD or DLM aerodynamics. Figure 13 shows the resulting flutter

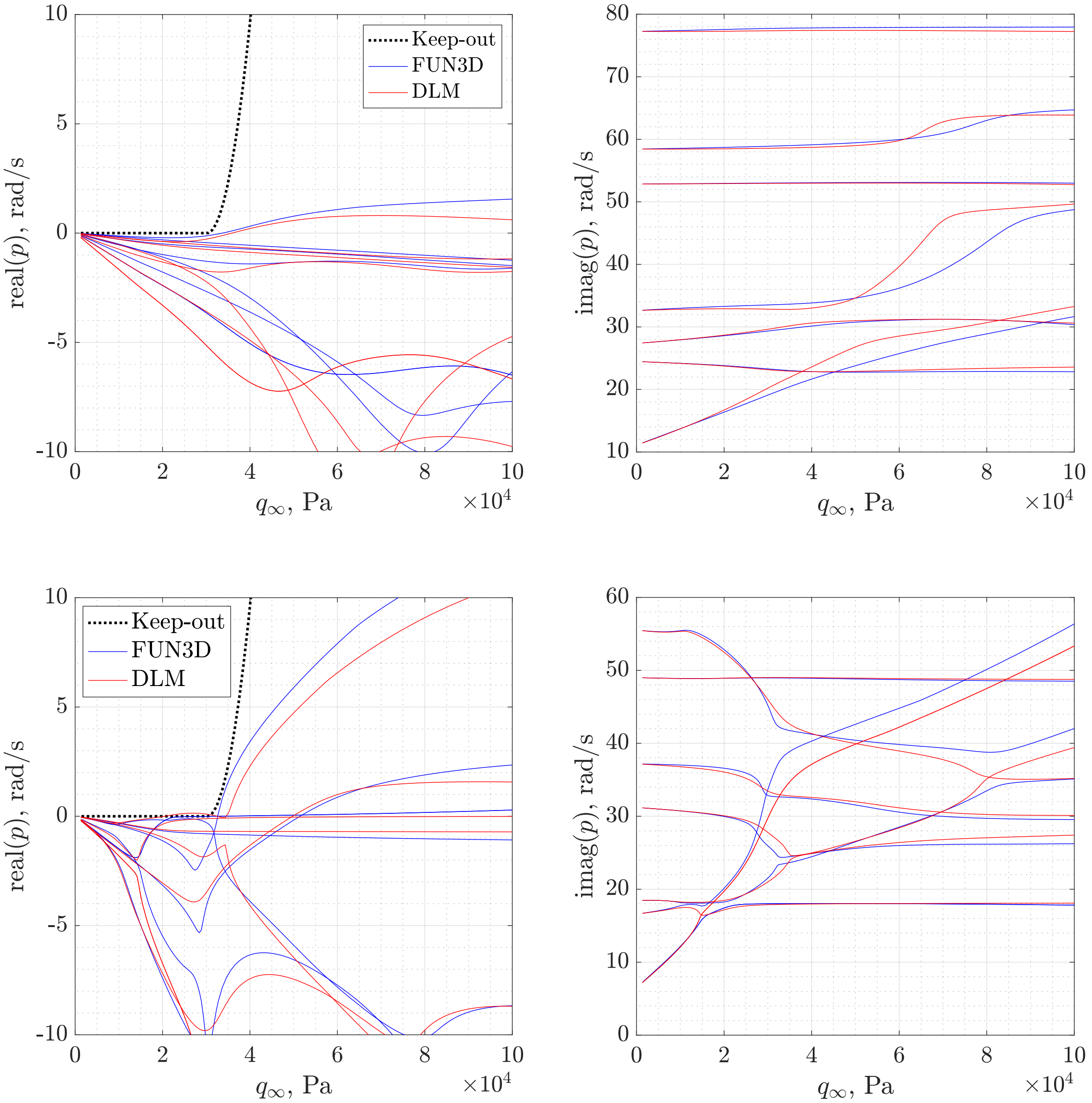

Figure 13: Comparison of DLM and LFD flutter at the baseline design (top) and FUN3D optimum (bottom).

results at these two designs. The soft flutter crossings at the baseline design, shown in the top plots, are fairly similar when using the different physics, indicating a fairly weak transonic effect at this Mach number as previously observed in Stanford & Massey [20]. At the FUN3D optimum, however, a hard flutter crossing at around 3.5×10^4 kg/(m-s^2) shows a larger difference between LFD and DLM, with LFD predicting flutter at a lower dynamic pressure (as expected). Interestingly, the DLM flutter result predicts a soft hump mode at a lower dynamic pressure which yields a violated KS flutter constraint value of 0.147 compared to -2.900×10^{-9} with LFD.

Figure 14 shows the gust-induced root bending moment PSDs at the baseline design and FUN3D optimum, using LFD and DLM aerodynamics. Looking at the largest peak at around 0.5 Hz, LFD predicts a larger peak at the baseline design but a smaller one at the optimal design. Conversely, the peak at around 5 Hz is significantly larger with LFD. This indicates that, at the dynamic pressure used by the gust analysis, one (or a combination of multiple) flexible modes are more lightly damped with LFD than with DLM in the vicinity of 5 Hz. Overall, due to the lower frequency peak, LFD predicts an RMS value significantly higher than DLM at the baseline design, but slightly smaller at the optimal design. As mentioned in Sect. 3.1, this larger RMS computed by DLM is more restrictive on the design space than LFD, thus leading to a larger fuel burn with FUN3D rather than FUN3D+DLM. This occurs even in spite of the flutter constraint, which is inactive in the FUN3D+DLM case but active in the FUN3D case.

4 Conclusion

Gradient-based optimization of the CRM wing with gust and flutter constraints has been demonstrated. This work built on previous work by the authors [14] which introduced the problem setup but relied on DLM-based unsteady aerodynamics (with either FUN3D- or VLM-based steady aerodynamics). The present work replaced these panel-based unsteady aerodynamics with those of the FUN3D LFD solver for both the flutter and stochastic gust analyses. The flutter analysis was evaluated at a slightly higher Mach number (0.9 rather than 0.85) to ensure

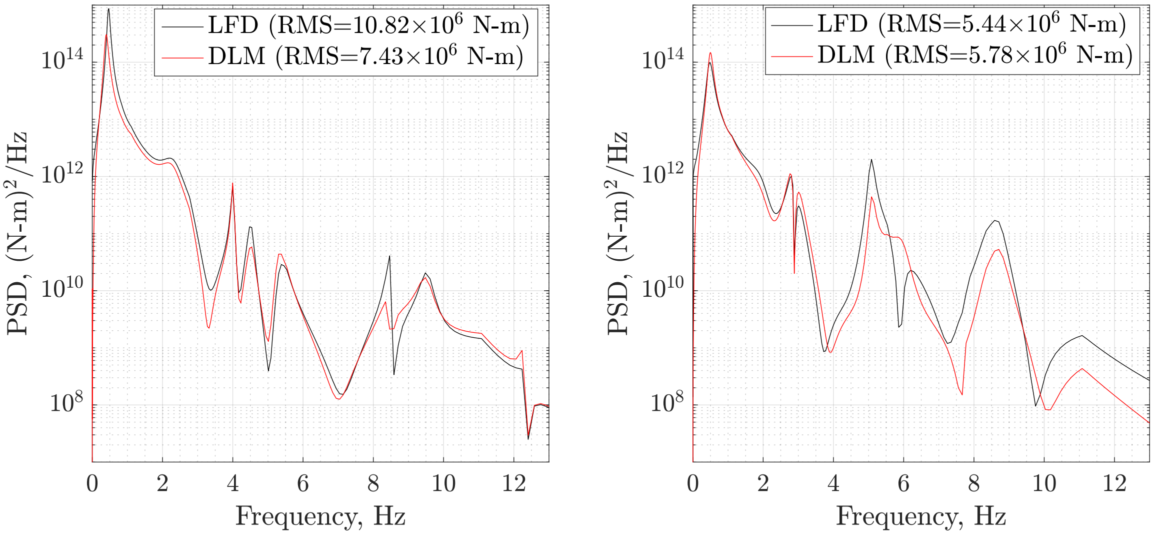

Figure 14: Comparison of DLM and LFD gust-induced root bending moment PSD at the baseline design (left) and FUN3D optimum (right).

a more active flutter constraint for demonstrative purposes. Comparisons were made with the existing optimizations that relied on DLM. LFD- and DLM-based gust and flutter responses are also compared at the optimized designs. Overall, large design differences were observed when moving from panel to CFD-based steady aerodynamics. Smaller differences were seen when replacing DLM with LFD, due to weak transonic effects in this case.

While performing a full gradient-based optimization with LFD-based unsteady aerodynamics represents a major hurdle, various other improvements and developments are planned for future work. For instance, while all present optimizations currently rely on Euler equations augmented with empirical profile drag (in an attempt to compute more realistic fuel burn values), RANS equations are the target aerodynamic fidelity level for future work. Moving to RANS will likely require code performance improvements to be able to run an optimization in a reasonable amount of time, especially when used for LFD as well. Another area of potential improvement is to include more flight conditions for the gust and flutter responses. With DLM readily available, many subsonic flight conditions could be included without increasing computational cost significantly. It may also be valuable to implement DLM correction methods in the literature and compare DLM and LFD to corrected DLM optimizations. Regardless of unsteady aerodynamics used, a longer term plan is to consider stresses as a gust quantity of interest rather than root bending moment. The latter serves as an intermediate step towards constraining gust-induced stresses, which would likely require rational function approximations to obtain accurate stresses from a modal decomposition representation.

5 ACKNOWLEDGEMENTS

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.

6 References

[1] Anderson, W. K., Newman, J. C., and Karman, S. L. (2018). Stabilized Finite Elements in FUN3D. *Journal of Aircraft*, 55(2). doi:10.2514/1.C034482.

[2] Anderson, W. K. and Newman, J. (2018). High-Order Stabilized Finite Elements on Dynamic Meshes. In *2018 AIAA Aerospace Sciences Meeting*. Kissimmee, Florida, USA. doi:10.2514/6.2018-1307.

[3] Anderson, W. K. and Bonhaus, D. L. (1994). An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids. *Computers & Fluids*, 23(1), 1–21. doi: 10.1016/0045-7930(94)90023-X.

[4] Biedron, R. T. and Thomas, J. L. (2009). Recent Enhancements to the FUN3D Flow Solver for Moving-Mesh Applications. In *AIAA 47th Aerospace Sciences Meeting*. Orlando, Florida, USA. doi:10.2514/6.2009-1360.

[5] Nielsen, E. J. and Anderson, W. K. (2002). Recent Improvements in Aerodynamic Design Optimization on Unstructured Meshes. *AIAA Journal*, 40, 1155–1163. doi:10.2514/2.1765.

[6] Wang, L., Diskin, B., Biedron, R. T., Nielsen, E. J., and Bauchau, O. A. (2017). Sensitivity Analysis of Multidisciplinary Rotorcraft Simulations. In *55th AIAA Aerospace Sciences Meeting*. Grapevine, Texas, USA. doi:10.2514/6.2017-1670.

[7] Jacobson, K. E., Stanford, B. K., Wood, S. L., and Anderson, W. K. (2021). Adjoint-based Sensitivities of Flutter Predictions based on the Linearized Frequency-domain Approach. In *59th AIAA Aerospace Sciences Meeting*. Virtual Event. doi:10.2514/6.2021-0282.

[8] Jacobson, K., Thelen, A., and Stanford, B. (2024). Linearized Frequency-Domain Gust Analysis and Adjoint-Based Sensitivities. In *AIAA SciTech*. Orlando, Florida, USA. doi: 10.2514/6.2024-2591.

[9] Stanford, B. K. (2018). Aeroservoelastic Optimization of a Transport Aircraft Wingbox under Stochastic Gust Constraints. *Journal of Aeroelasticity and Structural Dynamics*, 6(1), 21 – 41. doi:10.2514/6.2018-2837.

[10] Bisplinghoff, R. L., Ashley, H., and Halfman, R. L. (2013). *Aeroelasticity*. Courier Corporation.

[11] Yildirim, A., Jacobson, K. E., Stanford, B. K. Gray, J. S., Mader, C. A., Martins, R. R. A., and Kennedy, G. J. (2020). MPhys: A Modular Multiphysics Simulation Package using the OpenMDAO Framework. In *AIAA Aviation Forum*. Presentation Only, Virtual.

[12] Yildirim, A., Jacobson, K., Anibal, J., Stanford, B., Gray, J., Mader, C., Martins, J., and Kennedy, G. (2024). MPhys: A Modular Multiphysics Simulation Package using the OpenMDAO Framework. Submitted.

[13] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A. (2019). OpenMDAO: An Open-Source Framework for Multidisciplinary Design, Analysis, and Optimization. *Structural and Multidisciplinary Optimization*, 59, 1075–1104. doi:10.1007/s00158-019-02211-z.

[14] Thelen, A., Stanford, B., and Jacobson, K. (2024). Flutter and Stochastic Gust Constraint Studies of the Undeflected CRM using MPhys. In *AIAA SciTech Forum*. Orlando, Florida, USA. doi:10.2514/6.2024-2774.

[15] Brooks, T., Kenway, G., and Martins, J. (2018). Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings. *AIAA Journal*, 56(7). doi:10.2514/1.J056603.

[16] Kiviaho, J. F. and Kennedy, G. J. (2019). Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares. *AIAA Journal*, 57(5), 2237–2243. doi: 10.2514/1.J057318.

[17] Kennedy, G., Kenway, G., and Martins, J. (2014). A Comparison of Metallic, Composite and Nanocomposite Optimal Transonic Transport Wings. Tech. Rep. CR-2014-218185, NASA.

[18] Wu, N., Kenway, G., Mader, C. A., Jasa, J., and Martins, J. R. R. A. (2020). pyOptSparse: A Python Framework for Large-Scale Constrained Nonlinear Optimization of Sparse Systems. *Journal of Open Source Software*, 5(54), 2564. doi:10.21105/joss.02564.

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

[20] Stanford, B. and Massey, S. J. (2017). Uncertainty Quantification of the FUN3D-Predicted Flutter Boundary on the NASA CRM. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Grapevine, Texas, USA. doi: 10.2514/6.2017-1816.

[21] Hintjens, P. (2013). *ZeroMQ: Messaging for many Applications*. O'Reilly Media, Inc.

[22] Haeser, G. and de Melo, V. (2015). Convergence Detection for Optimization Algorithms: Approximate-KKT Stopping Criterion when Lagrange Multipliers are not Available. *Operations Research Letters*, 43, 484–488.