NASA/CR-20250007060

Nastran Integration for MBSA&E Framework Development and Assessment

Thomas Nascenzi
M4 Engineering, Inc., Long Beach, California

September 2025

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION.
  Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM.
  Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT.
  Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION.
  Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION.
  Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION.
  English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/CR-20250007060

Nastran Integration for MBSA&E Framework Development and Assessment

Thomas Nascenzi
M4 Engineering, Inc., Long Beach, California

Prepared under Contract 80GRC023CA047

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

September 2025

Level of Review: This material has been technically reviewed by expert reviewer(s).

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

Table of Contents

List of Figures

List of Tables

Introduction

The development of high-fidelity multidisciplinary analysis and optimization (MDAO) capabilities is essential for advancing modern aerospace vehicle design. Traditional workflows often suffer from limited interoperability between structural solvers, aerodynamic tools, and optimization frameworks. To address these limitations, this project focuses on integrating NASTRAN, a structural solver, into the OpenMDAO framework.

This effort was conducted under NASA Contract 80GRC023CA047 by M4 Engineering, Inc., as part of the broader initiative to enhance Model-Based Systems Analysis and Engineering (MBSA&E) capabilities. The primary objective of the project was to develop a robust, reusable, and modular Python-based interface that allows NASTRAN to interface within OpenMDAO workflows and improve structural optimization capabilities. This enables engineers to leverage NASTRAN's structural analysis and sensitivity capabilities while benefiting from OpenMDAO's flexibility and advanced optimization drivers.

In support of this objective, multiple demonstration models were created across various NASTRAN solution sequences (SOLs), including static, aeroelastic, flutter, and buckling analyses. These models were used to evaluate the accuracy, robustness, and computational performance of the integration. Additionally, capabilities were developed to extract sensitivity data, aggregate constraint outputs, and streamline the generation of OpenMDAO components from existing NASTRAN input decks.

The work described herein also included integration of third-party aerodynamic tools (e.g., transonic CFD solvers and GROM) with OpenMDAO, development of representative aircraft configurations such as the truss-braced wing (TTBW) and blended-wing body (BWB), and final deployment into NASA's Aviary conceptual design framework. The outcomes of this project provide a foundation for future multidisciplinary optimizations using high-fidelity structural models and demonstrates some of the challenges.

Nastran Integration in OpenMDAO

A series of example models were developed to help understand Nastran's design optimization SOL 200 compatibility with OpenMDAO for a variety of analysis types. Five Nastran design optimization models were created for each analysis type. Additionally, an equivalent OpenMDAO problem was created for each example. The OpenMDAO problems relied on Nastran for the analysis and sensitivity calculation, but performed the optimization process independently. Ad-hoc Python scripts were created to facilitate the writing and reading of optimization variables to and from the Nastran model and response. This exercise helps inform future development of generalized OpenMDAO and Nastran compatibility software.

Within Nastran optimization, there are two primary cards that are analogous to OpenMDAO inputs and outputs. The first is DESVAR, which defines a design variable for optimization. DESVARs can control many aspects of the model including mesh grid point locations and shell element thicknesses. In the OpenMDAO example problems, analogous inputs are defined for each DESVAR in the corresponding Nastran model.

The second Nastran card of interest is the DRESP card. The DRESP is used to request specific model responses to be used for objectives or constraints. In the OpenMDAO example problems, analogous outputs are defined for each DRESP in the corresponding Nastran model.

The compute function in the OpenMDAO problems were set up to edit, run, and parse the Nastran model. First, the inputs are written to the DESVAR cards in the BDF. Next, Nastran is run on the BDF using the Python subprocess module. The analysis produces a set of output files, including a human-readable file that can be parsed using standard Python functionality. All responses specified using the DRESP cards will be extracted and assigned to their corresponding component outputs. The Nastran run also produces partial derivatives. The compute partial function in the OpenMDAO problems were set up to just parse the resulting sensitivity information from the previous compute function. An overview of the OpenMDAO and Nastran process is shown in Figure 1.

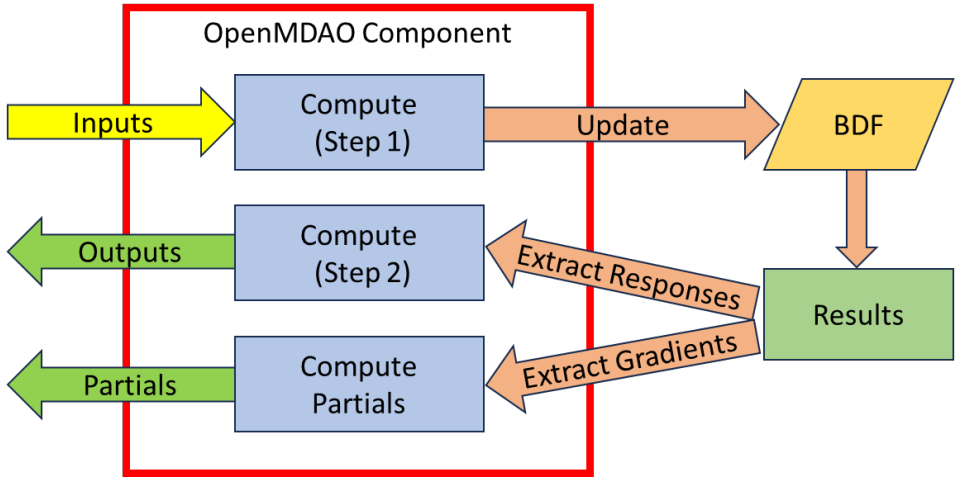


Figure 1: A flowchart showing the *compute* and *compute_partials* functions for the OpenMDAO example problems.

Development of Nastran SOL Examples

SOL 101: Linear Statics

A simple cantilevered beam model was set up to look at linear statics optimization. The linear statics analysis applies a simple upward force at one end of the beam. The optimization problem tries to maximize the length of the beam with constraints on stress, deflection, and weight. The length and radius of the beam are modeled as design variables. This optimization was performed as a Nastran-only design optimization and as an

OpenMDAO problem using Nastran for the analysis. Figure 2 provides a visualization of the cantilevered beam problem. Table 1 provides an overview of the optimization problem, while Table 2 provides the results of both versions of the optimization.

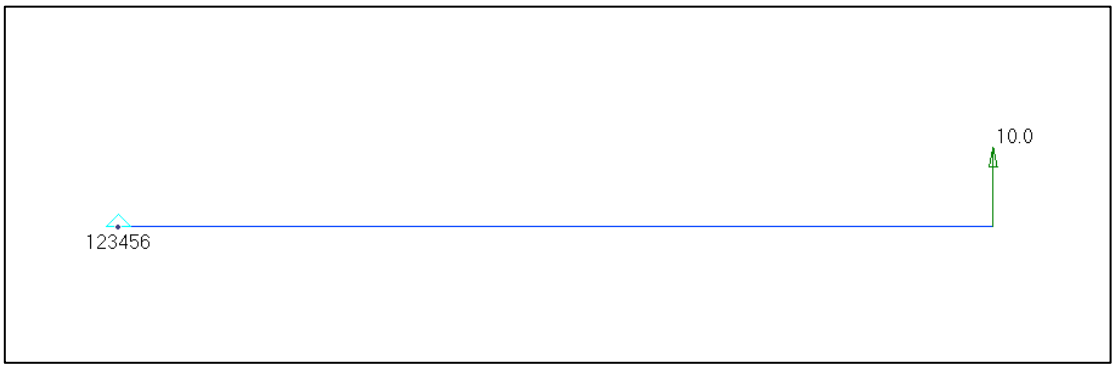


Figure 2: Visualization of the cantilevered beam with one end fixed in all DOFs (left end) and a 10lb force (right end)

Table 1: Optimization problem statement for the cantilevered beam.

Maximize:	• Beam length
W.r.t.:	• Beam radius • Beam length
Subject to:	• Mass < 0.0259slinch • Z-displacement < 0.1in • Stress < 26667psi

Table 2: Comparison of Nastran and OpenMDAO results for the cantilevered beam problem.

	NX NASTRAN	MSC NASTRAN	OpenMDAO (SLSQP)	OpenMDAO (SNOPT)
Iterations	12	4	17(157)	3(3)
Feasible	Yes	Yes	Yes	Yes
Radius	0.098470	.098522	0.098474	.098474
Weight (obj.)	1.5779e-5	1.5796e-5	1.5781e-5	1.5781e-5
Time	1.551s	4.138	169.184s	2.7195s

SOL 105: Buckling

A cantilevered beam model made of shell elements was set up to look at buckling optimization. The linear statics component of the model contains a simple upward force at one end. The optimization problem tries to minimize the weight of the beam with constraints on strain and panel buckling. The width of the box and the sandwich panel ply and core thicknesses are modeled as design variables. This optimization was performed as a Nastran-only design optimization and as an OpenMDAO problem using Nastran for the analysis. Figure 3 shows a visualization of the linear statics results for the unoptimized design. Figure 4 shows the buckling results

for the unoptimized design. Table 3 provides an overview of the optimization problem, while Table 4 provides the results of both versions of the optimization. Table 4 and Table 5 give a comparison of the solution time and processes for both problems.

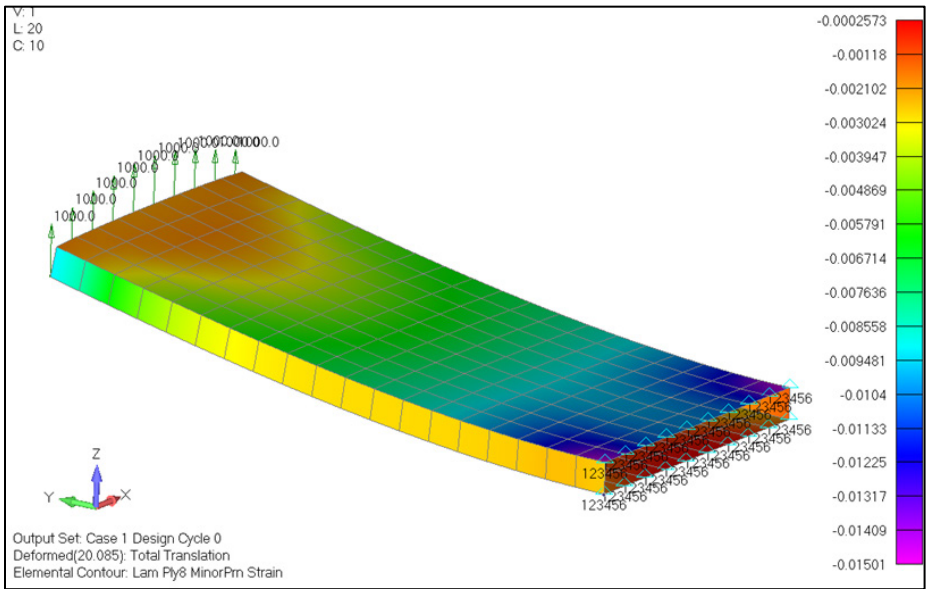


Figure 3: Visualization of the initial statics loads problem. The model is fixed at one end, and a force is applied at the other end.

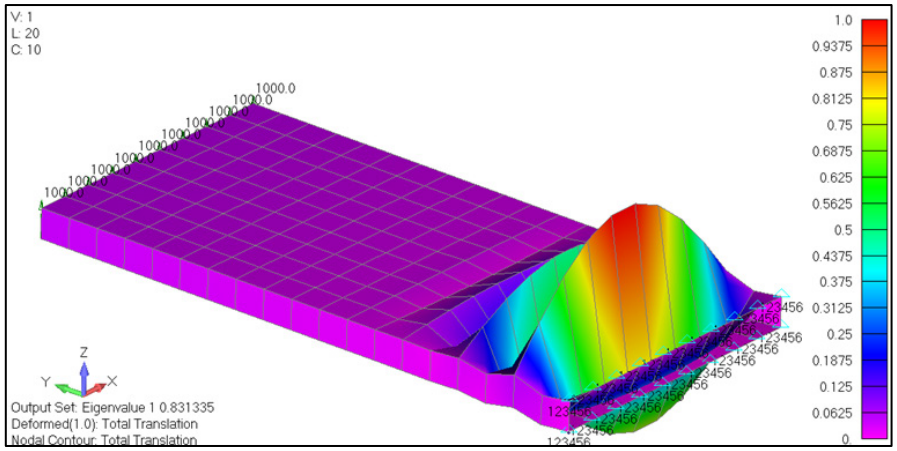


Figure 4: Visualization of the lowest eigenvalue buckling mode in the unoptimized model.

Table 3: Optimization problem for buckling analysis.

Minimize:	• Weight
W.r.t.:	• Box width • Core thickness • Ply thickness
Subject to:	• Buckling eigenvalues > 2.0 • Strain < 4000με

Table 4: Comparison of the Nastran and OpenMDAO optimization results for the bucking analysis.

	Nastran	OpenMDAO (SLSQP)
Iterations	481	170
Feasible	Yes	Yes
Weight (obj.)	.3835	0.3682
Width	12.91	13.27
Ply 1 T	0.00775	0.01725
Ply 2 T	0.1937	0.2382
Ply 3 T	0.00775	0.01925
Core T	2.705	1.337
Time	4.8 min	4.5 min

Table 5:Buckling Example: OpenMDAO Solution Time Breakdown

	Initialize Time(s)	Time per Iteration	Iterations	Total Time(s)
Wrapper Overhead	1.008	0.059	170	11.038
OpenMDAO Processes	0.076	0.023	170	3.986
Nastran Initialization and Output	0.00	0.67	170	113.9
Nastran Solve Times	0.00	0.67	170	113.9
Total Time	1.084	1.4215	170	242.824

Table 6: Buckling Example: Nastran Solution Time Breakdown

	Initialize Time(s)	Time per Iteration	Iterations	Total Time(s)
Total Time(s)	0.33	0.67	430	288.43

SOL 144: Static Aeroelastic

A relatively simple model of the X-57 Maxwell was created to demonstrate static aeroelastic analysis and optimization. The analysis consists of a single 2.5g pull up maneuver with two trim variables: angle-of-attack and elevator deflection. The optimization problem minimizes the weight of the wing with constraints on wing strain. Wing skin thicknesses are setup as design variables. This optimization was performed as a Nastran-only design optimization, and as an OpenMDAO problem using Nastran for the analysis.

Figure 5 shows the structural elements of the model while Figure 6 shows the aero elements. Figure 7 provides a visualization of the structural skin elements controlled by the wing thickness design variables. Table 7 provides an overview of the optimization problem, while Table 8 provides the results of both versions of the optimization.

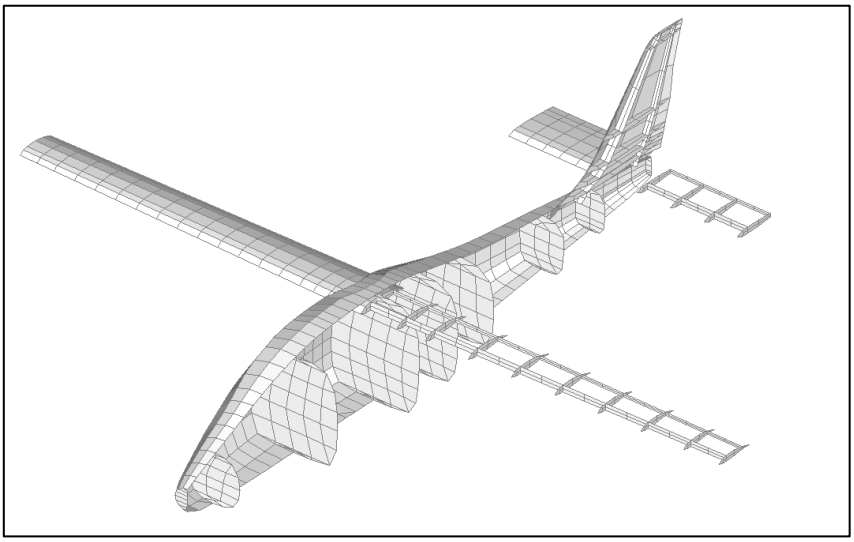


Figure 5: Visualization of the structure of the aeroelastic model.

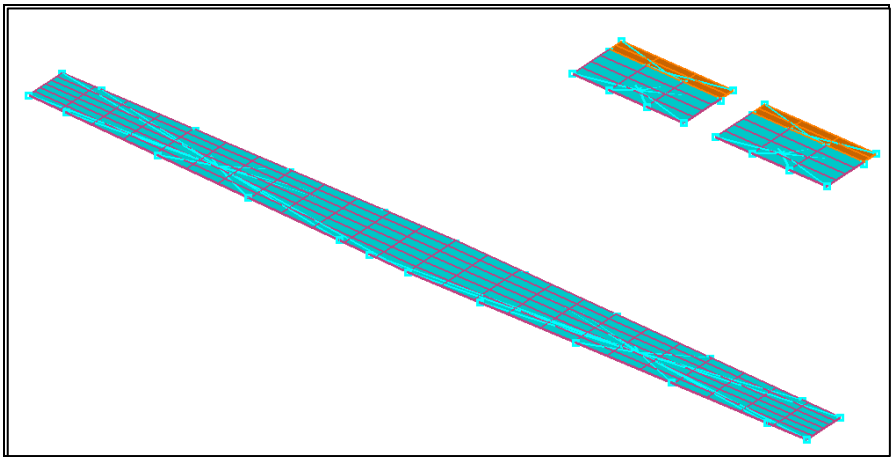


Figure 6: Visualization of the VLM lifting surfaces used for the aero analysis. The orange boxes function as control surfaces for the trim solution.

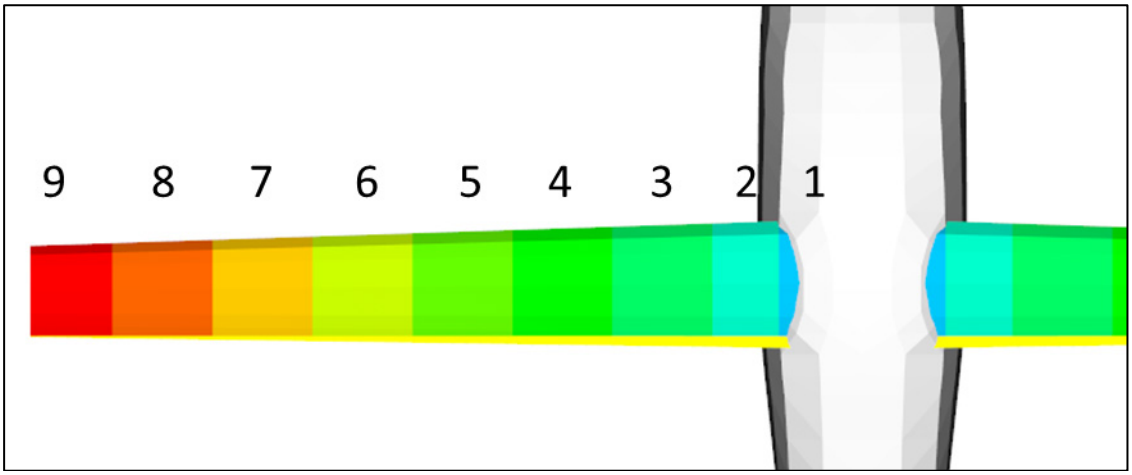


Figure 7: A visualization of the nine skin-thickness design regions on the wing. The skin thicknesses are mirrored to the other side of the wing.

Table 7: A summary of the optimization problem for the aeroelastic model.

Minimize:	• Weight
W.r.t.:	• Wing region skin thicknesses
Subject to:	• VM Stress < 40ksi

Table 8: Comparison of the Nastran and OpenMDAO optimization results for the static aeroelastic analysis

	NX Nastran	MSC Nastran	OpenMDAO (SLSQP)	OpenMDAO(SNOPT)
Iterations	18	3	13	7(7)
	Yes	Yes	Yes	Yes
Weight (obj.)	2574 lb	2554 lb	2554 lb	2554 lb
	0.049172	.050559	0.0508	.0508
Region 2 T	0.040167	.041768	0.417	.0417
	0.029474	.031091	0.0311	.03107
Region 4 T	0.026154	.020082	0.0197	.0197
	0.01928	0.0100 (Min)	0.0100 (Min)	0.0100 (Min)
Region 6 T	0.02225	0.0100 (Min)	0.0100 (Min)	0.0100 (Min)
	0.029271	0.0100 (Min)	0.0100 (Min)	0.0100 (Min)
Region 8 T	0.029271	0.0100 (Min)	0.0100 (Min)	0.0100 (Min)
	0.047283	0.0100 (Min)	0.0100 (Min)	0.0100 (Min)
Time	52.533s	2.885s	18.3518s	13.46s

SOL 145: Flutter

A FE model was created based on a 3D-printed scaled flutter model for flutter analysis and optimization. The analysis consists of a flutter analysis speed-sweep at a constant altitude (sea-level). The optimization problem minimizes the weight of the wing with a constraint specifying that flutter cannot appear below 60m/s. Wing skin thicknesses are setup as design variables. The optimization was performed using only Nastran and as a OpenMDAO problem using Nastran for analysis.

Figure 8 shows the structural elements of the model while Figure 9 shows the aero elements. Figure 10 provides a visualization of the modeshape occurring at 50m/s in the unoptimized model. Table 9 provides and overview of the optimization problem, while Table 10 provides the results of both versions of the optimization. Table 11 and Table 12 show the details of the OpenMDAO and Nastran process and the time breakdown respectively.

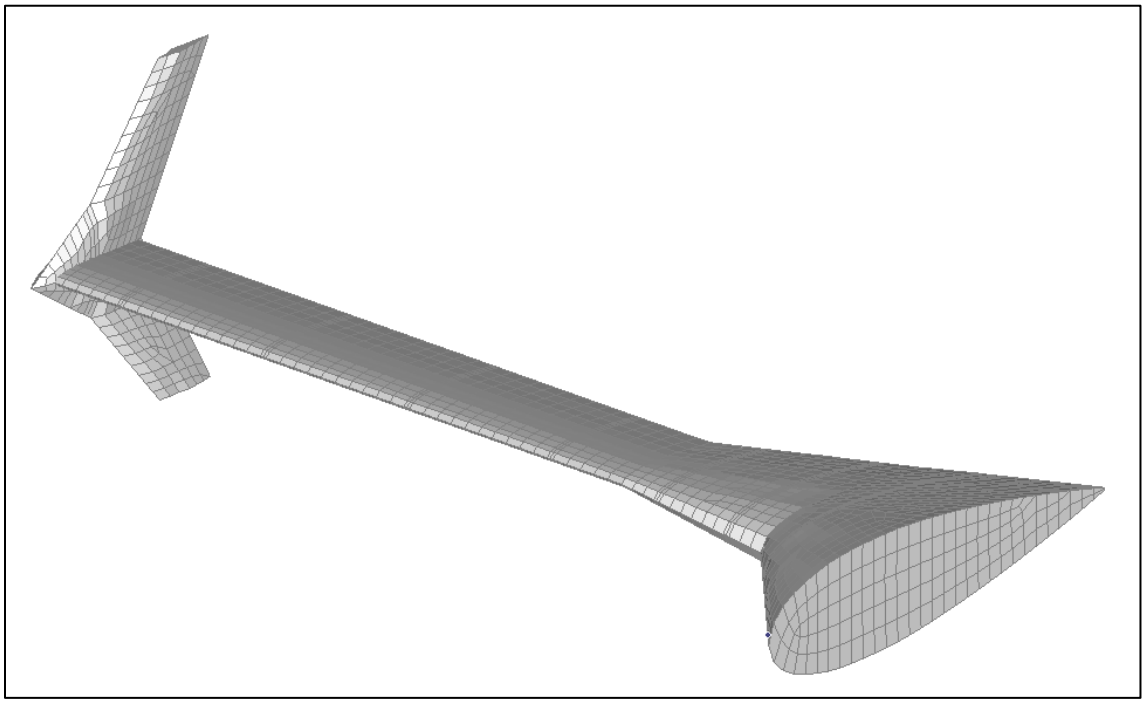


Figure 8: A visualization of the structural elements for flutter analysis.

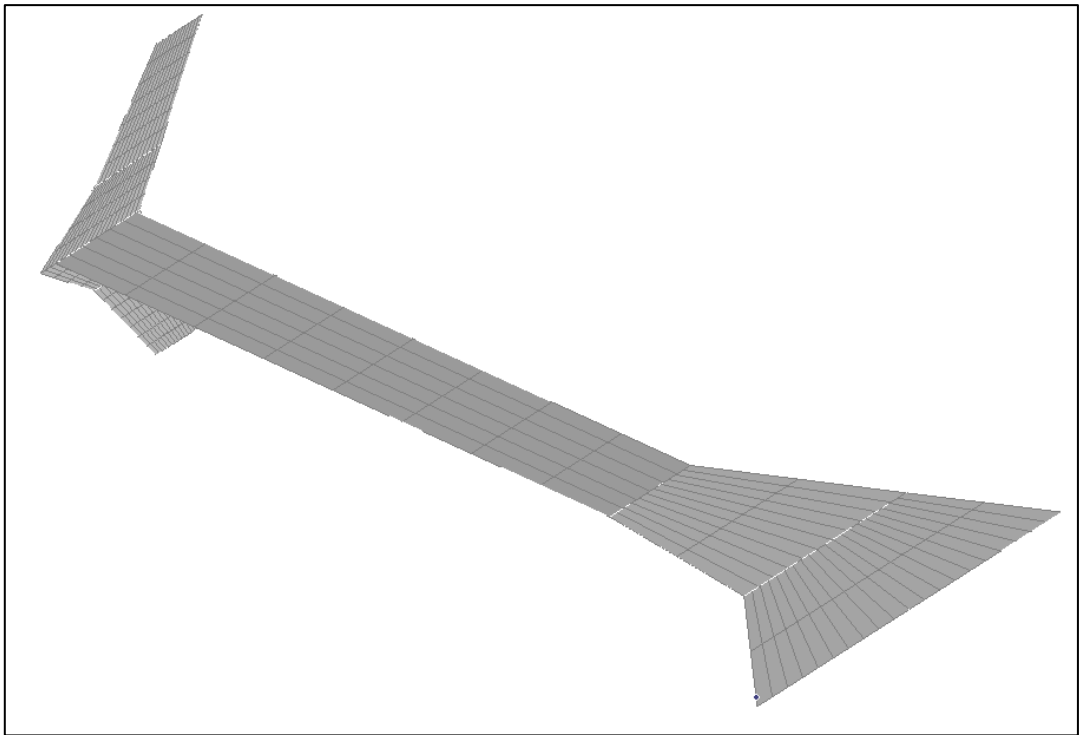


Figure 9: A visualization of the VLM lifting surface elements for flutter analysis.

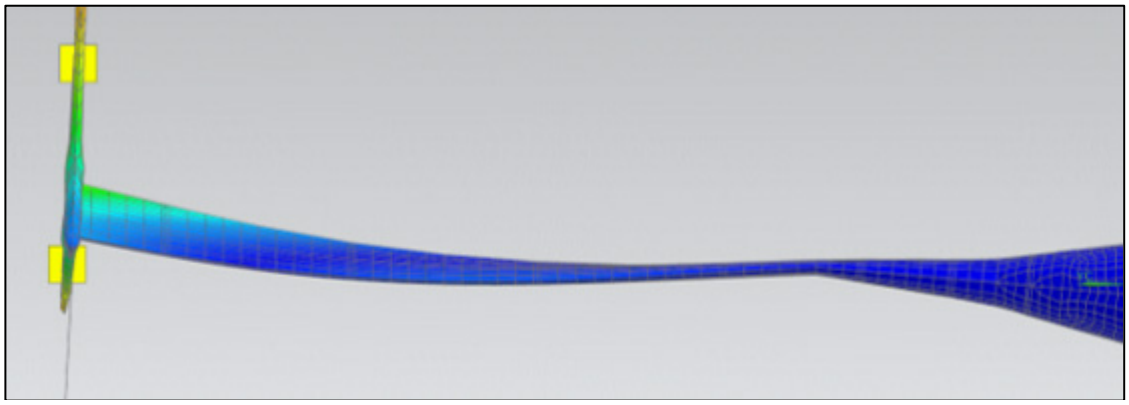


Figure 10: Flutter modeshape appearing at 50m/s for the unoptimized model.

Table 9: A summary of the optimization problem for flutter analysis.

Minimize:	• Weight
W.r.t.:	• Wing region skin thicknesses
Subject to:	• First flutter condition > 60 m/s (sea-level)

Table 10: Comparison of the Nastran and OpenMDAO optimization results for the flutter analysis.

	Nastran	**OpenMDAO (SLSQP)**
Iterations	100(max)	43
Feasible	Yes	Yes
Weight (obj.)	3.022kg	2.965kg
Region 1 T	.00508(min)	.00508 (min)
Region 2 T	.00254(min)	.00254 (min)
Region 3 T	.00254(min)	.00254 (min)
Region 4 T	.00254(min)	.00254(min)
Region 5 T	.00254(min)	.00254(min)
Region 6 T	.00254(min)	.00272
Region 7 T	.00368	.00418
Region 8 T	.00441	.00369
Region 9 T	.00419	.00319
Region 10 T	.00254(min)	.00283
Time	1.563 minutes	4.12 minutes

Table 11: Flutter Example: OpenMDAO Solution Time Breakdown

	Initialize Time(s)	Time per Iteration	Iterations	Total Time(s)
Wrapper Overhead	1.2	0.126	43	6.618
OpenMDAO Processes	1.375	0.23	43	11.265
Nastran Initialization and Output	0.00	4.45	43	191.35
Nastran Solve Times	0.00	0.88	43	37.84
Total Time	2.575	5.686	43	247.073

Table 12: Flutter Example: Nastran Solution Time Breakdown

	Initialize Time(s)	Time per Iteration	Iterations	Total Time(s)
Total Time(s)	4.13	0.88	100	92.13

SOL 146: Dynamic Aeroservoelastic

A dynamic aeroservoelastic model was created based on the scaled flutter model featured in the flutter analysis. The structural and aerodynamic models are identical except for the addition of an outboard control surfaces. This outboard control surface is actuated with step functions to produce a transient response.

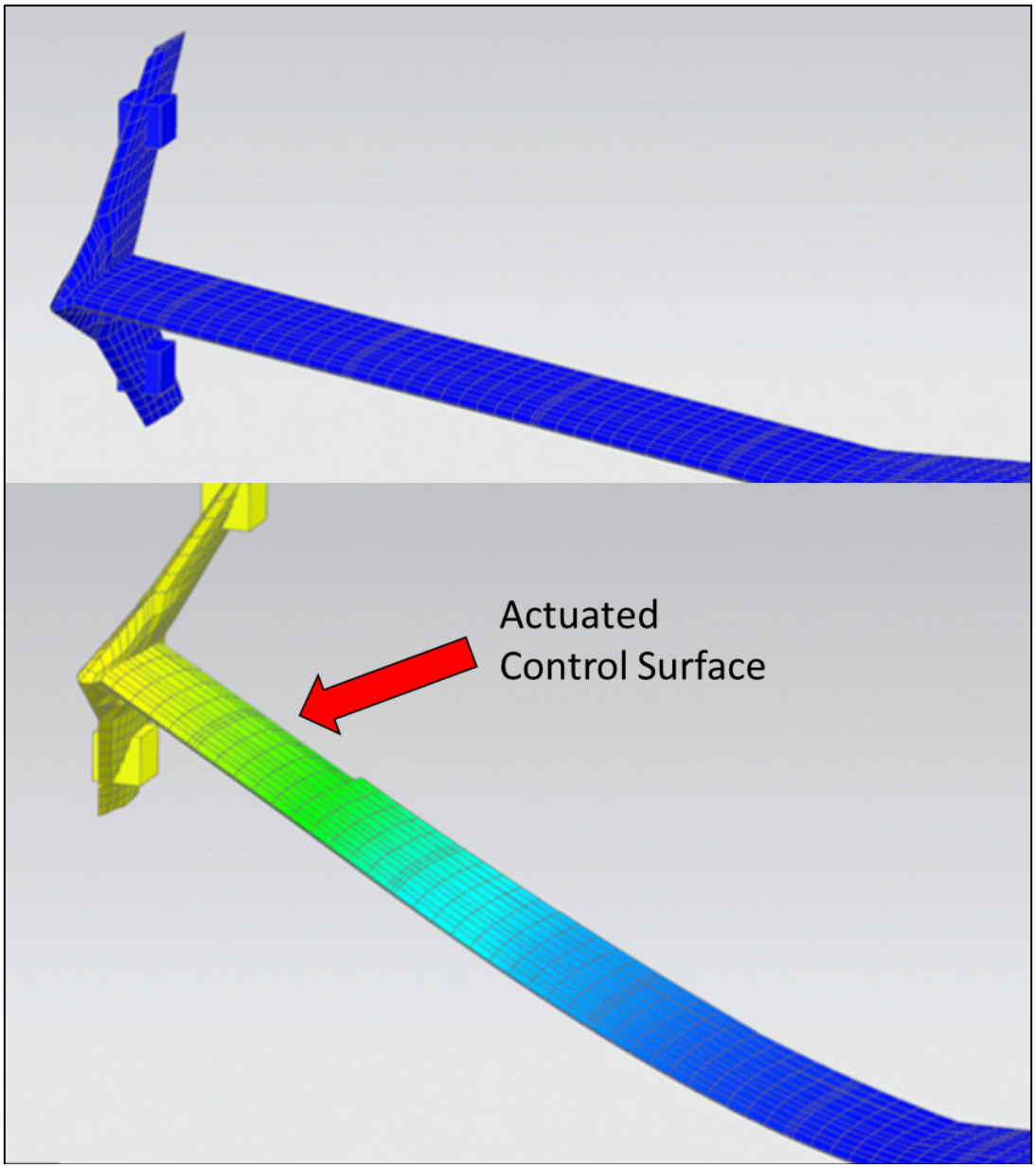


Figure 11: Wing deflection at two different time stamps. The top image shows the model before any control surface actuation. The bottom image shows the model sometime after the outboard control surface was extended trailing-edge down.

Nastran Sensitivity Benchmarking

The Nastran user manuals are unclear as to when the adjoint method is used to calculate design sensitivities. Therefore, we created a benchmark problem to examine computation time growth as the number of design variables change. Figure 12 shows computation time growth as the number of non-shape design variables grows. It shows that with auto-adjoint enabled, the computation time does not grow as the number of design variables grow. However, Figure 13 shows that computation time does grow if the number of shape design variables increase. This is because shape sensitivities are calculated using the finite difference method in Nastran. All computation times were obtained from a design optimization run with a linear statics analysis. We plan to perform additional benchmarks for additional analysis types to understand if analysis type influences sensitivity computation method.

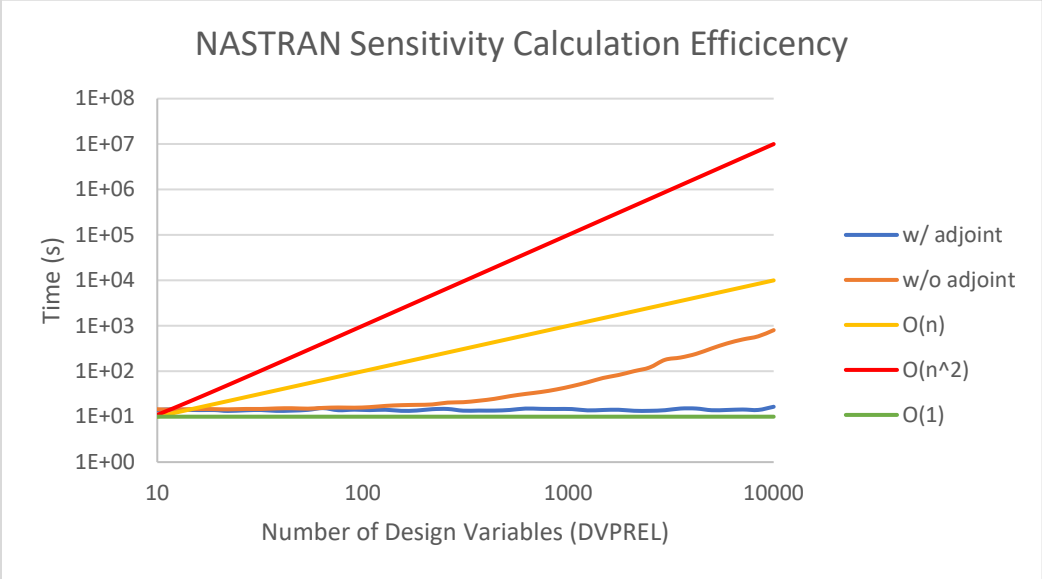


Figure 12: A graph showing the computational growth of Nastran sensitivity calculation as the number of non-shape design variables increases.

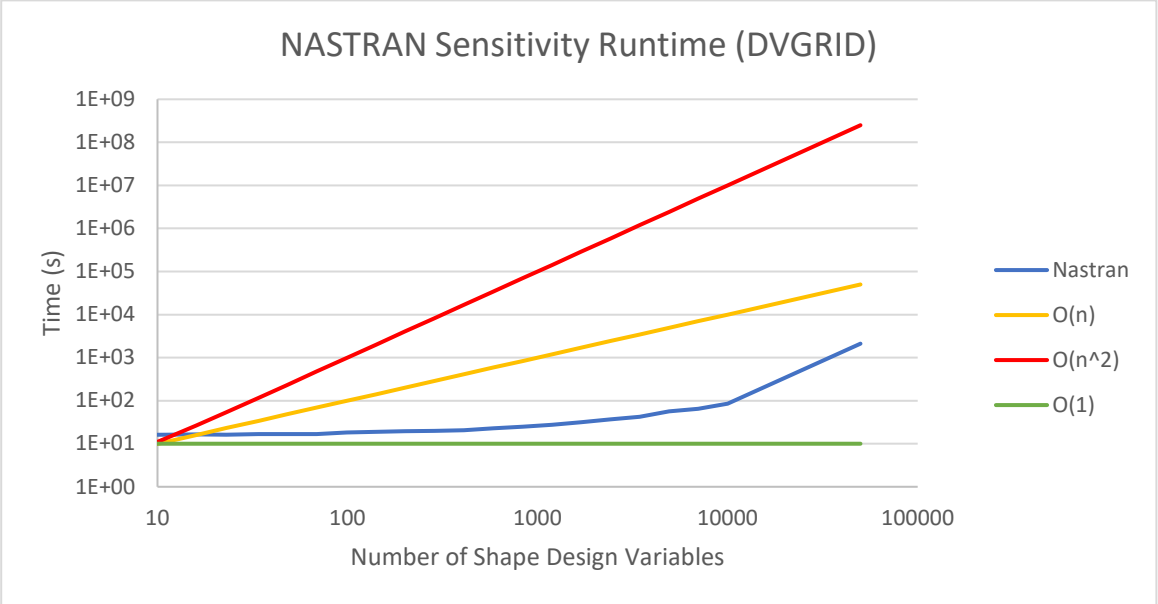


Figure 13: A graph showing the computational growth of Nastran sensitivity calculation as the number of shape design variables increases.

Response and Sensitivity Extraction

There is an option in Nastran to request a human-readable form of the design responses and sensitivity information. However, this information is embedded in a larger file containing much more information. The larger file makes parsing the sensitivity information a time-consuming process.

Nastran features the ability to request that any tables saved during the run be written to file. Additionally, Nastran allows the user to insert DMAP code into any Nastran submodule. Together, this enables us to write just the tables needed and bypass using the larger output file for sensitivity information. Table 13 shows an example code insertion and save-to-file call for the table R1VALRG. A description of the Nastran

tables relevant to optimization is provided in Table 14.

Table 13: Example Nastran code to write the R1VALRG table to file.

```
COMPILE SUBDMAP = EXITOPT $ Add code to the submodule EXITOPT
ALTER 62 $ Insert code after line 62
OUTPUT4 R1VALRG,,,,//-1/92///16 $ Save table to location 92
CEND
ASSIGN OUTPUT4 = 'resp.op4', UNIT=92 $ Saves table to file
```

Table 14: A description of the optimization-relevant Nastran tables.

Table Name	Table Description
R1VALRG	Matrix of initial values of the retained first level (direct) responses.
R2VALRG	Matrix of initial values of the retained second level (synthetic) responses.
R3VALRG	Matrix of initial values of the retained third level responses.
DSCM2	Design sensitivity coefficient matrix.
DSCMCOL	Correlation table for design sensitivity coefficient matrix.

Python Module Development

Initializing an OpenMDAO Component and Problem from an Existing NASTRAN SOL 200

The focus of this part of the Python module is to automatically create an OpenMDAO component and problem from an existing NASTRAN design optimization SOL 200. This is done with a number of functions. The primary input for these functions is simply a file path to the existing NASTRAN model.

The first function is to write a modified version of the original NASTRAN model. First, the modified version adds a number of ALTER statements to the model. ALTER statements are used in insert code into NASTRAN submodules. The ALTERs added add code to extract NASTRAN response and sensitivity information. The second modification is the addition of a DSARPT card. The DSARPT card can be used to tell NASTRAN to only compute sensitivity information instead of running the optimization. This will enable OpenMDAO to be the optimization driver while still using NASTRAN responses and sensitivities.

The second function is used to create an OpenMDAO explicit component for the NASTRAN model. Much of the functionality of the explicit component is the same model to model. However, some aspects need to be created uniquely for each model. A summary of the functionality is shown in Table 15.

Table 15: A summary of the functionality of the NASTRAN component for OpenMDAO.

Function	Pre-Written Component	Automatically Generated
Initialize	Declares options • bdf_path • work_dir	None
Setup	• Reads BDF • Sets up work dir	• Declares inputs • Declares outputs • Calls super

Compute	• Modifies work bdf
	  • Updates DESVARs
	  • Updates GRIDs
	• Runs NASTRAN
	• Parses response
		None

Compute Partials	• Parses sensitivities matrix	None

The third function is used to write an OpenMDAO problem based on the information contained in the NASTRAN model. Although finite element analysis can be one of many disciplines contained in a OpenMDAO problem, NASTRAN optimization contains all of the information necessary for stand-alone optimization, including objectives, constraints, and design variables. Therefore, both a component and a problem can be defined from an existing NASTRAN optimization model.

Constraint Aggregation

One obstacle that needed to be overcome to integrate NASTRAN with OpenMDAO was NASTRAN's constraint screening functionality. Before NASTRAN calculates model sensitivities, it performs constraint screening to limit the number of sensitivities to be calculated. This can be seen in a figure from the NASTRAN optimization user manual, and is shown in Figure 14. The constraints are screened based on two criteria. The first criterium is the truncation threshold. NASTRAN will truncate all constrained responses that are not within a certain threshold of the constraint bounds. The second criteria is maximum number of constraints per constraint region. Typically, NASTRAN constraints are applied to a region of elements. The maximum number of constraints simply limits the number of possible constraints per region. An overview of this process can be seen in Figure 15, which was taken out of the NASTRAN optimization user manual [1].

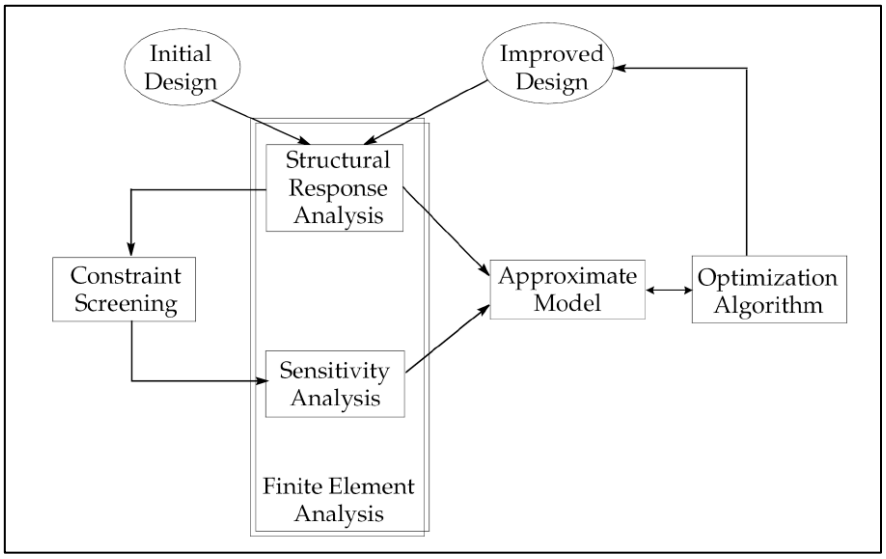


Figure 14: NASTRAN optimization flowchart.

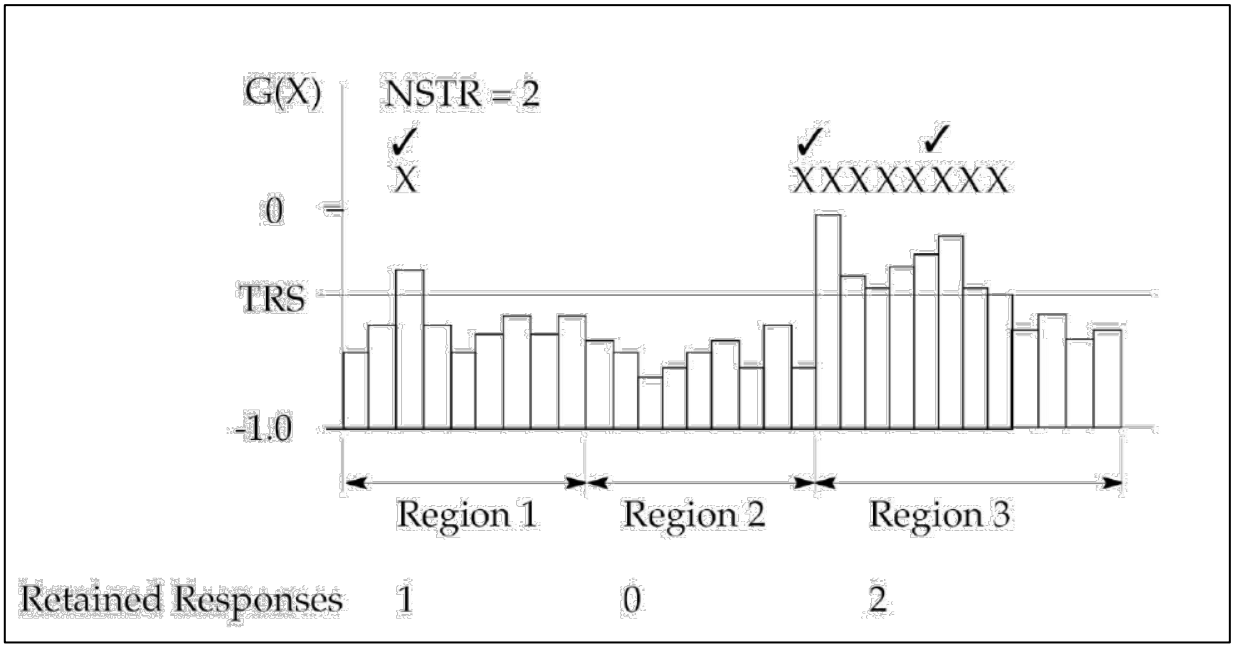


Figure 15: Graph explaining NASTRAN constraint screening. The NSTR value is the maximum number of constraints per region criteria. The TRS value is the truncation threshold for constraint screening. G(X) is the normalized constraint value with a positive value indicating that the constraint is violated.

One consequence of NASTRAN's constraint screening is that the number of responses changes iteration to iteration. Unfortunately, OpenMDAO is not set up to handle a varying number of outputs from a component. One solution to this problem is to set a very low truncation threshold so that no responses are truncated based on the truncation threshold. Additionally, the maximum number of constraints could be set to a value of 1. Together, both of these constraints guarantee that a single response will be calculated for every constraint region. However, the element characterized by the response may change iteration to iteration as the design changes and individual elements become more or less critical. This could potentially create a noisy response, which would be difficult for a gradient-based optimizer to handle.

Constraint aggregation is an alternative solution that we settled on, specifically the Kreisselmeier-

Steinhauser (KS) function.

$$c_{DKS}(\rho) = \max_i g_i + \frac{1}{\rho} \ln \left[\sum_{i=1}^{n_t} e^{\rho(g_i - \max_i g_i)} \right]$$

The KS function can take a variable number of responses and condense them down in to a single response value. This guarantees a constant number of responses and helps smooth out a response when there is critical element or mode switching.

Constraint aggregation was implemented in both the *compute* and *compute_partials* functions of the NASTRAN component for OpenMDAO. The *compute* function simply plugs the response into the KS function to obtain a single output for each constraint region. The *compute_partials* function utilizes the derivative of the KS function (along with the chain rule) to combine the partials derivatives into a single partial derivative for each constraint region.

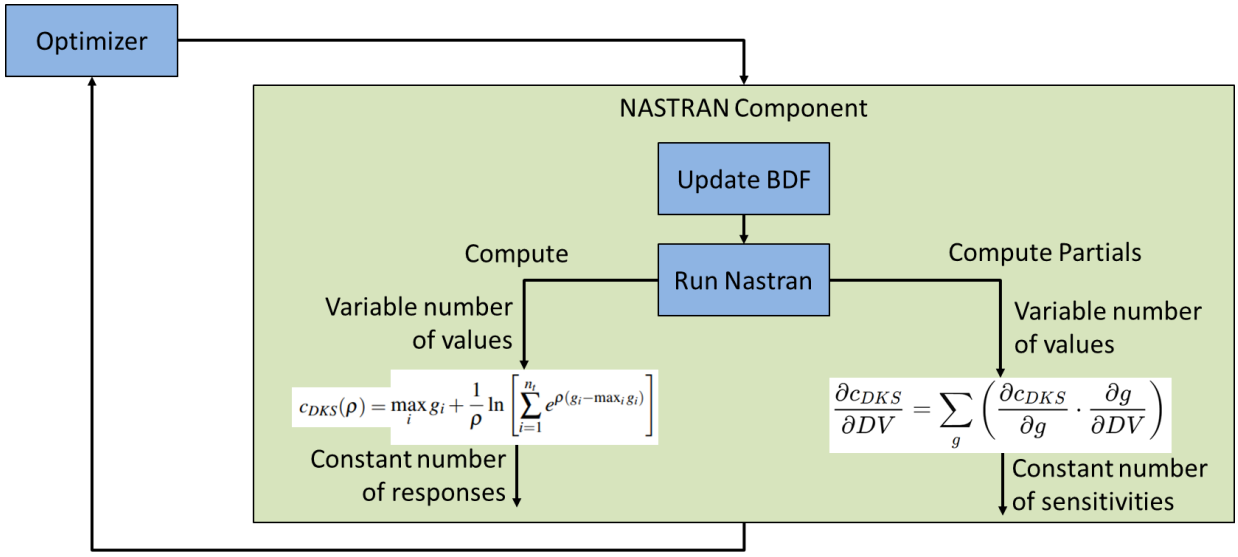


Figure 16: A flowchart outlining how the NASTRAN component for OpenMDAO utilizes constraint aggregation.

Constraint Aggregation- Global Level

As we are dealing with potentially very large models with Jacobians on the order of a million entries, a global aggregation (super aggregation) method was developed to help speed up optimization times. When quick results are needed for a model, a general method is to reduce tolerance constraints on the optimization convergence. This adjustment helps minimize lengthy run out times when the model is near a solution. However, for large models, achieving solutions even near convergence can be time-consuming, as the optimization must deal with making all design variables near optimal and below constraint limits simultaneously. To deal with this challenge, we propose a model that uses the KS function again to aggregate NASTRAN constraints into 1(or multiple) global constraints. These global constraints emphasize regions closest to the constraint values and force regions not near constraint limits to take large step sizes and quickly near the solution region. As a result, solutions can be approximated in minutes to hours rather than hours to days for the largest models.

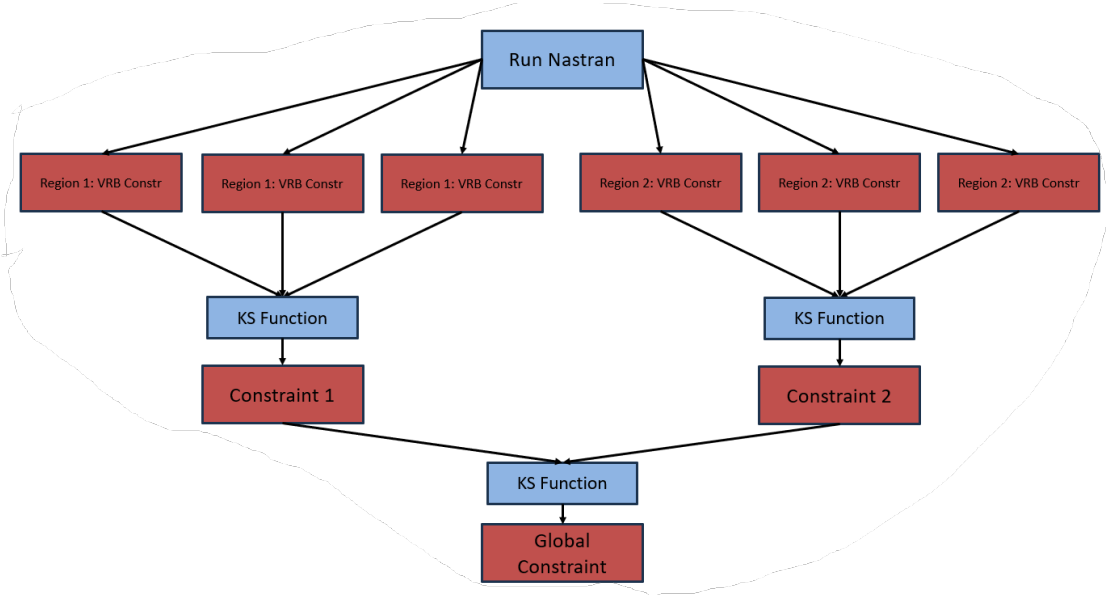


Figure 17: A flowchart outlining the global aggregation method in OpenMDAO

Partials are aggregated in the same way as the local KS aggregation, effectively cutting down on the Jacobian size by a factor equal to the number of constraints.

	Desvar 1	Desvar 2	Desvar 3	Desvar 4	Desvar 5
Constr 1	$\frac{\partial C1}{\partial D1}$	$\frac{\partial C1}{\partial D2}$	$\frac{\partial C1}{\partial D3}$	$\frac{\partial C1}{\partial D4}$	$\frac{\partial C1}{\partial D5}$
Constr 2	$\frac{\partial C2}{\partial D1}$	$\frac{\partial C2}{\partial D2}$	$\frac{\partial C2}{\partial D3}$	$\frac{\partial C2}{\partial D4}$	$\frac{\partial C2}{\partial D5}$
Constr 3	$\frac{\partial C3}{\partial D1}$	$\frac{\partial C3}{\partial D2}$	$\frac{\partial C3}{\partial D3}$	$\frac{\partial C3}{\partial D4}$	$\frac{\partial C3}{\partial D5}$
Constr 4	$\frac{\partial C4}{\partial D1}$	$\frac{\partial C4}{\partial D2}$	$\frac{\partial C4}{\partial D3}$	$\frac{\partial C4}{\partial D4}$	$\frac{\partial C4}{\partial D5}$
Constr 5	$\frac{\partial C5}{\partial D1}$	$\frac{\partial C5}{\partial D2}$	$\frac{\partial C5}{\partial D3}$	$\frac{\partial C5}{\partial D4}$	$\frac{\partial C5}{\partial D5}$
Constr 6	$\frac{\partial C6}{\partial D1}$	$\frac{\partial C6}{\partial D2}$	$\frac{\partial C6}{\partial D3}$	$\frac{\partial C6}{\partial D4}$	$\frac{\partial C6}{\partial D5}$
Constr 7	$\frac{\partial C7}{\partial D1}$	$\frac{\partial C7}{\partial D2}$	$\frac{\partial C7}{\partial D3}$	$\frac{\partial C7}{\partial D4}$	$\frac{\partial C7}{\partial D5}$
Constr 8	$\frac{\partial C8}{\partial D1}$	$\frac{\partial C8}{\partial D2}$	$\frac{\partial C8}{\partial D3}$	$\frac{\partial C8}{\partial D4}$	$\frac{\partial C8}{\partial D5}$
Constr 9	$\frac{\partial C9}{\partial D1}$	$\frac{\partial C9}{\partial D2}$	$\frac{\partial C9}{\partial D3}$	$\frac{\partial C9}{\partial D4}$	$\frac{\partial C9}{\partial D5}$
Constr 10	$\frac{\partial C10}{\partial D1}$	$\frac{\partial C10}{\partial D2}$	$\frac{\partial C10}{\partial D3}$	$\frac{\partial C10}{\partial D4}$	$\frac{\partial C10}{\partial D5}$

	D1	D2	D3	D4	D5
Global Constr	$\frac{\partial G1}{\partial D1}$	$\frac{\partial G1}{\partial D2}$	$\frac{\partial G1}{\partial D3}$	$\frac{\partial G1}{\partial D4}$	$\frac{\partial G1}{\partial D5}$

Figure 18: Comparing Jacobians before and after global aggregation

This method is implemented in the MDAOComps compute and compute partials functions as well as the Nastran.io parsing function. The user can specify in their problem which constraints to aggregate together using the *super_aggr* optional argument. This is provided as a toggle for each constraint, so aggregated and non-aggregated constraints can be mixed in the same problem. The user also has the option to specify the ρ value, which specifies the extent to which the partials are smoothed. A low ρ value (~50), will result in fast but imprecise solutions while a larger ρ (~500) will result in longer solutions with higher precision. It has been found that values much higher than ~500 will result in longer and less precise solutions than normal, non-aggregated solutions. While useful cases for this method are quite narrow, it has been found to be helpful for

cases with over 200 constraints (100 design variables), and in need of only approximate solutions. Shown in Figure 19, the super aggregated method beats the normal in solution time starting at around 100 design variables.

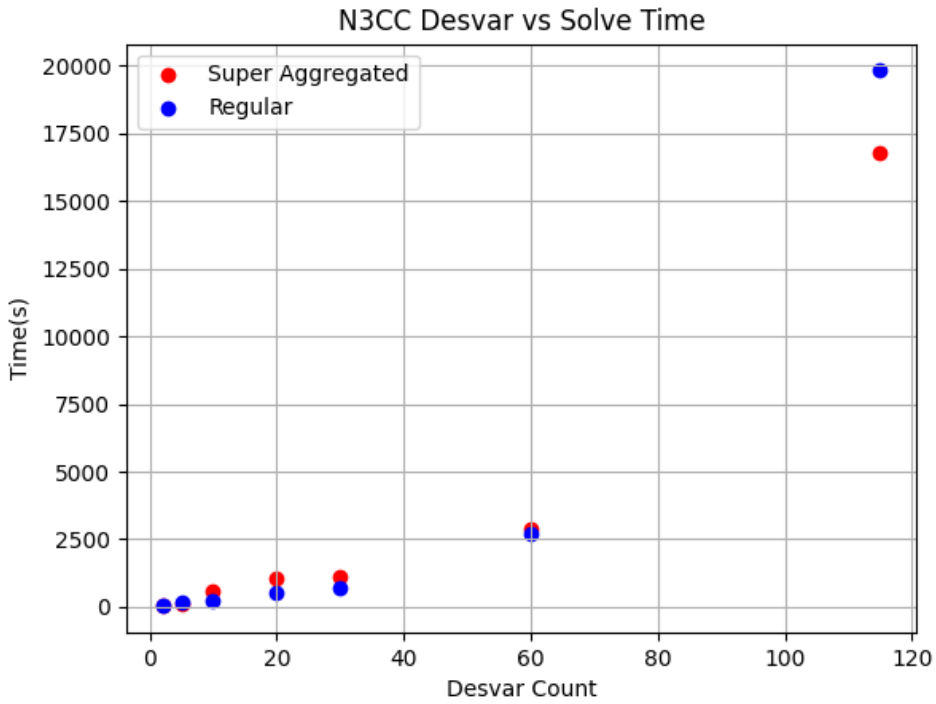


Figure 19: Comparison of solution times for super aggregated and normal methods on N3CC

Additional Integrations with OpenMDAO

Integration with the Transonic Aerodynamics Solver

Progress has been made on integrating the existing transonic aerodynamics solver within OpenMDAO. The transonic solver is based on the **U**nsteady **T**ransonic **S**mall **D**isturbance (UTSD) formulation for potential flow theory which calculates the aerodynamic influence coefficients (AICs) nonlinearly and is used to replace the AICs computed by NASTRAN. This process will be summarized below.

The aerodynamic influence coefficients matrix in NASTRAN provides a relationship between the lifting pressure coefficient and the downwash velocity induced by the inclination angle relative to freestream on each j-set surface panel element [2].

$$\{w_j\} = [A_{jj}]\{cp_j\} \tag{1}$$

The downwash velocities are related to the k-set degrees of freedom displacements on each aerodynamic grid element through the following

$$\{w_j\} = [D^1_{jk} + ikD^2_{jk}]\{u_k\} \tag{2}$$

The loads on the k-set degrees of freedom for each aerodynamic element are found from the following integration of the lifting pressure coefficients

$$\{P_k\} = q[S_{kj}]\{cp_j\} \tag{3}$$

Combining Eqs. (1-3) gives the final expression for the loads on the aerodynamic grid points.

$$\{P_k\} = q\left[S_{kj}\right]\left[A_{jj}\right]^{-1}\left[D_{jk}^1 + ikD_{jk}^2\right]\{u_k\} \tag{4}$$

The lifting surface methods in NASTRAN such as Doublet-Lattice compute $\left[A_{jj}\right]$ as a function of Mach number and reduced frequency [2]. However, since they are based on linearized aerodynamic potential theory, they fail to account for the nonlinear behavior present in transonic flow. Through Direct Matrix Input (DMI), the linear aerodynamic influence coefficients matrices computed by Doublet-Lattice can be replaced with those containing the transonic nonlinearities.

Starting with the transonic small-disturbance equation

$$-M_\infty^2\phi_{tt} - 2M_\infty^2\phi_{xt} + C\phi_{xx} + \phi_{yy} + \phi_{zz} = 0 \tag{5}$$

Where the nonlinear term C is

$$C = 1 - M_\infty^2 - (\gamma + 1)M_\infty^2\phi_x \tag{6}$$

Equation (5) can be cast in conservation form and simplified to give

$$\frac{\partial(-M_\infty^2\phi_t - 2M_\infty^2\phi_x)}{\partial t} + \frac{\partial\left((1 - M_\infty^2)\phi_x - \frac{\gamma+1}{2}M_\infty^2\phi_x^2\right)}{\partial x} + \frac{\partial\phi_y}{\partial y} + \frac{\partial\phi_z}{\partial z} = 0 \tag{7}$$

Next, prior to transforming Eq. (7) into computational coordinates we leverage the meshing assumptions described in [3] [4] such that grid-shearing is done only in the XY-plane. This eliminates many of the cross terms in the grid metric terms when the derivatives in $x, y,$ and z are transformed into $\xi, \eta,$ and ζ. Specifically, this introduces the following relations between the computational coordinates of $\xi, \eta,$ and ζ with $x, y,$ and z such that

$$\xi = \xi(x, y) \tag{8}$$

$$\eta = \eta(y) = y \tag{9}$$

$$\zeta = \zeta(z) = z \tag{10}$$

As a result, the following grid metrics are identical to zero:

$$\xi_z = \eta_x = \eta_z = \zeta_x = \zeta_y = 0 \tag{11}$$

This simplifies the first order derivatives to

$$\phi_x = \xi_x\phi_\xi \tag{12a}$$

$$\phi_y = \xi_y\phi_\xi + \eta_y\phi_\eta \tag{12b}$$

$$\phi_z = \zeta_z\phi_\zeta \tag{12c}$$

and the second order derivatives to

$$\phi_{xx} = \xi_x^2 \phi_{\xi\xi} \tag{13a}$$

$$\phi_{yy} = \xi_y \frac{\partial\left(\xi_y \phi_\xi + \eta_y \phi_\eta\right)}{\partial \xi} + \eta_y \frac{\partial\left(\xi_y \phi_\xi + \eta_y \phi_\eta\right)}{\partial \eta} \tag{13b}$$

$$\phi_{zz} = \zeta_z^2 \phi_{\zeta\zeta} \tag{13c}$$

We can then transform equation 7 into computational coordinates

$$\frac{\partial\left(M_\infty^2 \phi_t + 2M_\infty^2 \xi_x \phi_\xi\right)}{\partial t} = \frac{\partial\left(E\xi_x^2 \phi_\xi + F\xi_x^3 \left(\phi_\xi\right)^2 + \xi_y^2 \phi_\xi + \xi_y \eta_y \phi_\eta\right)}{\partial \xi} + \frac{\partial\left(\eta_y \xi_y \phi_\xi + \eta_y^2 \phi_\eta\right)}{\partial \eta} + \frac{\partial\left(\zeta_z^2 \phi_\zeta\right)}{\partial \zeta} \tag{14}$$

where $E = 1 - M_\infty^2$ and $F = -\frac{\gamma+1}{2} M_\infty^2$. We then cast Eq. (14) into the form

$$[B]\{\phi_{tt}\} + [C]\{\phi_t\} = \{R(\phi)\} \tag{15}$$

where the matrix $[C]$ contains the formulation for ϕ_{xt}.

Discretization of Eq. (15) is performed on a finite-difference volume mesh and is solved using a time-linearized AF algorithm described in Ref. [3] which yields a nonlinear steady-state solution for the potential, $\{\phi_s\}$, for a fixed set of downwash values $\{w_s\}$ such that

$$\{R(\phi_s, w_s)\} = 0 \tag{16}$$

Using Newton's method, a perturbation about (ϕ_s, w_s) of Eq. (15) gives the following

$$[B]\{\delta\phi_{tt}\} + [C]\{\delta\phi_t\} = \left[\frac{\partial R}{\partial \phi}\right]\delta\phi + \left[\frac{\partial R}{\partial w}\right]\delta w \tag{17}$$

where $\delta\phi$ is the perturbation potential and δw is some perturbation downwash. Equation (17) can then be transformed into the frequency domain

$$[B]\{-\omega^2 \delta\phi\} + [C]\{i\omega\delta\phi\} = \left[\frac{\partial R}{\partial \phi}\right]\delta\phi + \left[\frac{\partial R}{\partial w}\right]\delta w \tag{18}$$

and solved for the perturbation potential

$$\delta\phi = \left[-\omega^2[B] + i\omega[C] - \left[\frac{\partial R}{\partial \phi}\right]\right]^{-1}\left[\frac{\partial R}{\partial w}\right]\delta w \tag{19}$$

The perturbation potential is related to the perturbation pressure coefficient, δcp, through the following relationship [3]

$$\delta cp = -2\frac{\partial(\delta\phi)}{\partial x} - 2\frac{\partial(\delta\phi)}{\partial t} = -2\frac{\partial(\delta\phi)}{\partial x} - 2i\omega\delta\phi \tag{20}$$

It is important to note here the distinction between the aerodynamic elements in NASTRAN e.g., those created

from CAERO1 elements, and the grid points used for solution of the unsteady TSD formulation of Eq. (15). Of the latter, the grid points on the lifting surfaces will be denoted as the m-set. This distinction is important because the mesh resolution required for the solution of Eq. (15) is generally larger than the number of aerodynamic elements needed for Doublet-Lattice. With this in mind, an operator $[D]$ is defined which relates the perturbation potential in the volume mesh to the lifting pressure coefficients on grid points in the m-set using Eq. (20). Next, in order to transform the lifting pressure coefficients from the m-set to the j-set a second operator $[G]$ is defined which acts as a mapping function from the m-set to the j-set used in NASTRAN. The resulting relation for the j-set lifting pressure coefficients is shown here.

$$\{cp_j\} = [G][D]\left[-\omega^2[B] + i\omega[C] - \left[\frac{\partial R}{\partial \phi}\right]\right]^{-1}\left[\frac{\partial R}{\partial w}\right]\delta w \tag{21}$$

Finally, by defining a third operator $[H]$ which maps downwash velocities from the j-set – that is $\{w_j\}$, to the m-set grid points such that

$$\delta w = [H]\{w_j\} \tag{22}$$

the final expression for the lifting pressure coefficients on the j-set can be written.

$$\{cp_j\} = [G][D]\left[-\omega^2[B] + i\omega[C] - \left[\frac{\partial R}{\partial \phi}\right]\right]^{-1}\left[\frac{\partial R}{\partial w}\right][H]\{w_j\} \tag{23}$$

Examination of Eqs. (1) and (23) reveals a direct expression for the inverse of the aerodynamic influence matrix

$$[A_{jj}]^{-1} = [G][D]\left[-\omega^2[B] + i\omega[C] - \left[\frac{\partial R}{\partial \phi}\right]\right]^{-1}\left[\frac{\partial R}{\partial w}\right][H] \tag{24}$$

that is a function of freestream Mach number, reduced frequency, and contains the transonic nonlinear behavior from the converged TSD solution $\{R(\phi_s, w_s)\}$. The effect and purpose of the $[D]$, $[G]$, and $[H]$ operators are illustrated in Figure 20 for the F5 fighter wing.

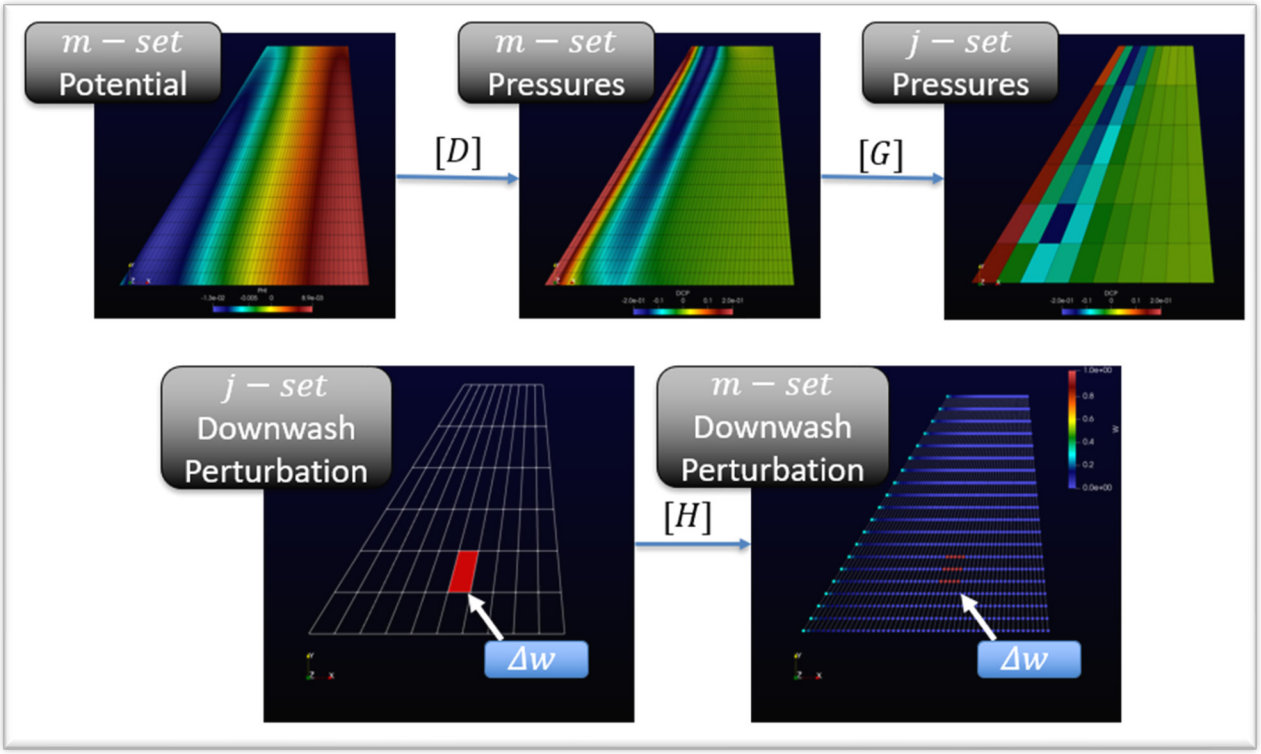


Figure 20: Visualization of the various matrix operators in Eq. (18) for the F5 fighter wing

The $[A_{jj}]$ matrix computed from the unsteady transonic small disturbance equation will replace the aerodynamic influence matrix computed in NASTRAN to introduce the transonic nonlinear aerodynamics into the NASTRAN solutions for static aeroelasticity and flutter and ultimately in NASTRAN's optimization solution.

NASTRAN Response Sensitivity Verification

In order to test this for obtaining response sensitivities, a simple NASTRAN aerodynamic model was created that consisted of a wing and a horizontal tail. Figure 21 illustrates the NASTRAN model for use in NASTRAN Doublet-Lattice (DLM) as well as the mesh used in the UTSD code. It should be noted here the differences between the NASTRAN DLM mesh which consists of surface panels and the UTSD mesh which consists of a volume mesh.

The first verification step was to confirm that the AIC matrix in NASTRAN was correctly getting replaced by the AIC matrix computed by UTSD when running NASTRAN's design optimization using the static aeroelastic solution. Figure 22 below shows the resulting predictions for C_{L_α} across a range of Mach numbers from NASTRAN, UTSD, and NASTRAN using the UTSD AIC matrix which demonstrates that the aerodynamic solution within NASTRAN is correctly getting replaced with the UTSD aerodynamics for NASTRAN's static aeroelastic solution.

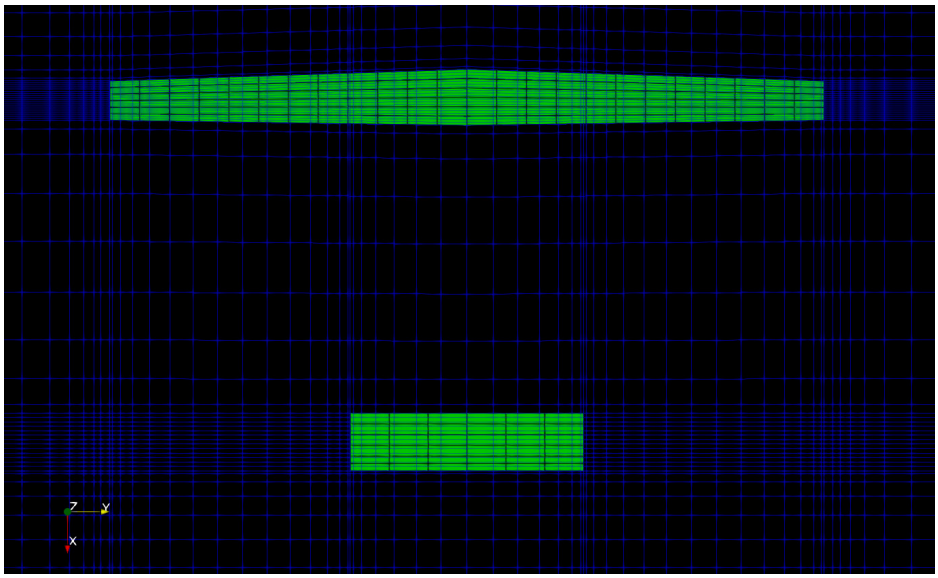


Figure 21: Simple wing and tail model with UTSD mesh in blue and NASTRAN DLM mesh in green

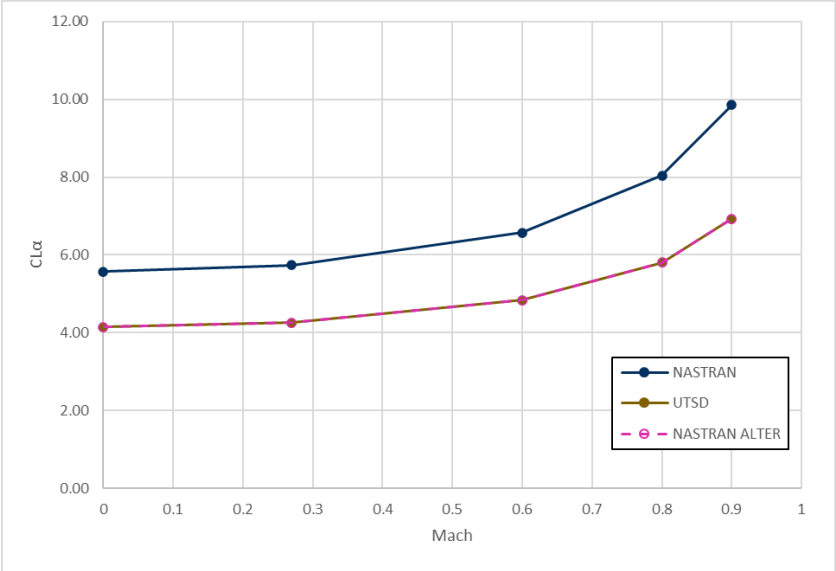
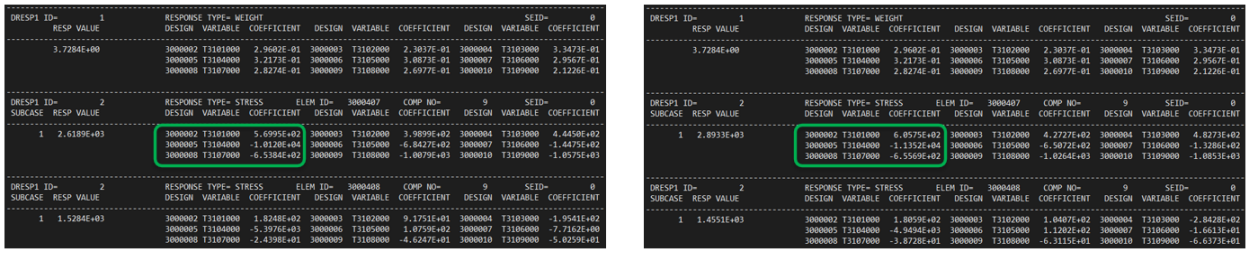


Figure 22: Lift curve slope prediction across Mach for NASTRAN (blue), UTSD (gold), and NASTRAN with UTSD AICs (pink)

The next step was to verify that the AIC matrix was correctly getting replaced for NASTRAN's Design Optimization solution. This was performed to ensure that the method for replacing the AIC matrix within NASTRAN was valid for both solution 144 (static aeroelastic) as well as solution 200 (design optimization). The resulting lift curve slope predictions matched those in Figure 22.

Finally, the design optimization model was run both with NASTRAN's DLM aerodynamics as well as the nonlinear aerodynamics from UTSD to assess the influence on the response sensitivities. This can be checked by inspection of the *f06* file output from NASTRAN which lists the response sensitivity coefficients which mathematically are the partial derivatives of a response with respect to a design variable [5]. Figure 23 shows a snippet of the *f06* files for the simple wing and tail model where the sensitivity coefficients are shown for the stress responses. From the figure it is clear that the stress response sensitivity coefficients have changed with the updated aerodynamics while the weight response sensitivity coefficients have remained the same.



DRESP1 ID= 1 RESPONSE TYPE= WEIGHT SEID= 0

RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
3.7284E+00	3000002	T3101000	2.9602E-01	3000003	T3102000	2.3037E-01	3000004	T3103000	3.3473E-01
	3000005	T3104000	3.2173E-01	3000006	T3105000	3.0873E-01	3000007	T3106000	2.9567E-01
	3000008	T3107000	2.8274E-01	3000009	T3108000	2.6977E-01	3000010	T3109000	2.1226E-01

DRESP1 ID= 2 RESPONSE TYPE= STRESS ELEM ID= 3000407 COMP NO= 9 SEID= 0

SUBCASE	RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
1	2.6189E+03	3000002	T3101000	5.6995E+02	3000003	T3102000	3.9899E+02	3000004	T3103000	4.4450E+02
		3000005	T3104000	-1.0128E+04	3000006	T3105000	-6.8427E+02	3000007	T3106000	-1.4475E+02
		3000008	T3107000	-6.5384E+02	3000009	T3108000	-1.0079E+03	3000010	T3109000	-1.0575E+03

DRESP1 ID= 2 RESPONSE TYPE= STRESS ELEM ID= 3000408 COMP NO= 9 SEID= 0

SUBCASE	RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
1	1.5284E+03	3000002	T3101000	1.8248E+02	3000003	T3102000	9.1751E+01	3000004	T3103000	-1.9541E+02
		3000005	T3104000	-5.3976E+03	3000006	T3105000	1.0759E+02	3000007	T3106000	-2.7162E+00
		3000008	T3107000	-2.4398E+01	3000009	T3108000	-4.6247E+01	3000010	T3109000	-5.0259E+01

DRESP1 ID= 1 RESPONSE TYPE= WEIGHT SEID= 0

RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
3.7284E+00	3000002	T3101000	2.9602E-01	3000003	T3102000	2.3037E-01	3000004	T3103000	3.3473E-01
	3000005	T3104000	3.2173E-01	3000006	T3105000	3.0873E-01	3000007	T3106000	2.9567E-01
	3000008	T3107000	2.8274E-01	3000009	T3108000	2.6977E-01	3000010	T3109000	2.1226E-01

DRESP1 ID= 2 RESPONSE TYPE= STRESS ELEM ID= 3000407 COMP NO= 9 SEID= 0

SUBCASE	RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
1	2.8933E+03	3000002	T3101000	6.0575E+02	3000003	T3102000	4.2727E+02	3000004	T3103000	4.8273E+02
		3000005	T3104000	-1.1352E+04	3000006	T3105000	-6.5072E+02	3000007	T3106000	-1.3286E+02
		3000008	T3107000	-6.5569E+02	3000009	T3108000	-1.0264E+03	3000010	T3109000	-1.0853E+03

DRESP1 ID= 2 RESPONSE TYPE= STRESS ELEM ID= 3000408 COMP NO= 9 SEID= 0

SUBCASE	RESP VALUE	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT	DESIGN	VARIABLE	COEFFICIENT
1	1.4551E+03	3000002	T3101000	1.8059E+02	3000003	T3102000	1.0407E+02	3000004	T3103000	-2.8428E+02
		3000005	T3104000	-4.9494E+03	3000006	T3105000	1.1202E+02	3000007	T3106000	-1.6613E+01
		3000008	T3107000	-3.8728E+01	3000009	T3108000	-6.3115E+01	3000010	T3109000	-6.6373E+01

Figure 23: Response sensitivity coefficients from NASTRAN with DLM aerodynamics (left) and UTSD aerodynamics (right)

To verify that the model correctly could replace the AIC matrix and arrive to the same optimized solution, 4 different models were run through an OpenMDAO solution. At the lower mach number, we expect the two solutions to be identical, and at the transitional Mach number they should diverge slightly as the aerodynamics deviate. From Table 16 we see that our results are as expected and deviate by a small margin for the Mach 0.9 case.

Table 16: AIC Matrix Replacement Normalized Weight Comparison

Mach	Replaced AIC Weight	Default AIC Weight
M=0.6	1.128	1.128
M=0.9	1.0843	1.0813

Code Enhancements to Transonic Code

Ultimately, the goal for integration of the UTSD transonic code is to perform the design sensitivity and optimization of a transonic truss-braced wing configuration like that shown below. Currently, the UTSD does not have the capability to simulate multiple lifting surface if any of the surfaces have dihedral due to the mesh assumptions listed in Eqs. (8-10).

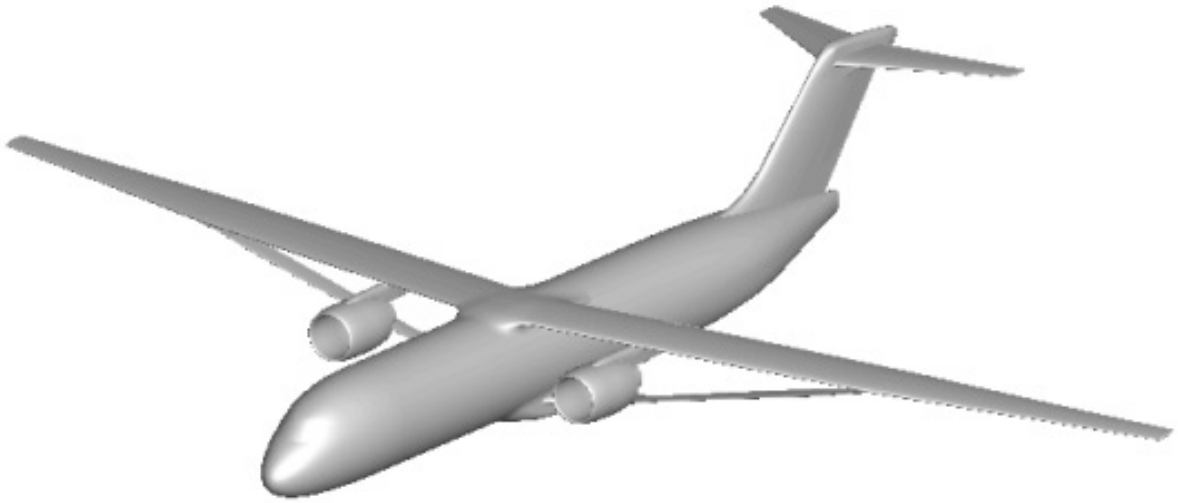


Figure 24: Transonic truss-braced wing configuration from [6]

Currently, the UTSD code is being updated to support simulations of lifting surfaces with dihedral. This involves modifying the meshing assumptions to allow grid-shearing in the YZ-plane which results in the following grid transformation relations

$$\xi = \xi(x, y) \tag{25}$$

$$\eta = \eta(y) = y \tag{26}$$

$$\zeta = \zeta(y, z) \tag{27}$$

which in turn modifies the resulting derivative terms

$$\phi_x = \xi_x \phi_\xi \tag{28a}$$

$$\phi_y = \xi_y \phi_\xi + \eta_y \phi_\eta + \zeta_y \phi_\zeta \tag{28b}$$

$$\phi_z = \zeta_z \phi_\zeta \tag{28c}$$

and the second order derivatives to

$$\phi_{xx} = \xi_x^2 \phi_{\xi\xi} \tag{29a}$$

$$\phi_{yy} = \xi_y \frac{\partial\left(\xi_y \phi_\xi + \eta_y \phi_\eta + \zeta_y \phi_\zeta\right)}{\partial \xi} + \eta_y \frac{\partial\left(\xi_y \phi_\xi + \eta_y \phi_\eta + \zeta_y \phi_\zeta\right)}{\partial \eta} + \zeta_y \frac{\partial\left(\xi_y \phi_\xi + \eta_y \phi_\eta + \zeta_y \phi_\zeta\right)}{\partial \zeta} \tag{29b}$$

$$\phi_{zz} = \zeta_z^2 \phi_{\zeta\zeta} \tag{29c}$$

While the general equation for the AIC matrix in Eq. (24) remains the same, the formulation for the Jacobian $\left[\frac{\partial R}{\partial \phi}\right]$ is modified with the updated formulation for the ϕ_{yy} derivative shown in Eq. (29b). This work is in progress with the code updates being tested currently.

Integration with GROM

GROM (Generalized Reduced Order Model) is a framework developed by M4 to allow for creation of reduced order aeroservoelastic state space models that can be interpolated across a range of flight conditions. The flowchart for generalized reduced order model generation in GROM is shown in Figure 25 where the system is ultimately reduced to the following state-space representation through the calculation of the A/B/C/D matrices.

$$\begin{aligned} \{\dot{x}\} &= [A]\{x\} + [B]\{u\} \\ \{y\} &= [C]\{x\} + [D]\{u\} \end{aligned}$$

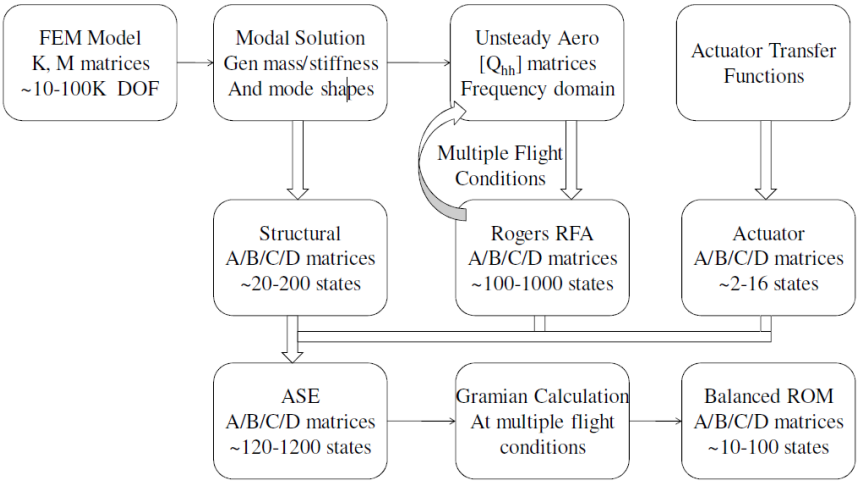


Figure 25: GROM flowchart [7]

The motivation behind the development of GROM was to advance the process of control law development, especially in the design of control systems required to provide flutter suppression, gust load alleviation, and ride quality enhancement. In this project, GROM has been integrated within the developed openmdao optimization package to leverage its capabilities and expand the design space.

To develop the optimization problem, an OpenMDAO explicit component has been developed for Nastran solution 146, under which, GROM is used to compute the A/B/C/D matrices, compute the system's frequency response, and obtain the Mode 7 frequency. Note here that Mode 7 frequency computed through GROM is selected as a constraint for demonstration purposes herein. However, the intent for our customers is to be able to use GROM for more advanced usage in optimization.
The optimization example can be defined as follows for the PawPrint aircraft:

$$\begin{cases} \min(weight) \\ 0.05" \leq t_{sp} \leq 0.15" \\ 0.3\ Hz \leq f \leq 100\ Hz \end{cases}$$

Where the objective function is the structural weight, the design variable is the thickness of the spars t_{sp} (Figure 26), and the constraint is the Mode 7 GROM based frequency *f*.

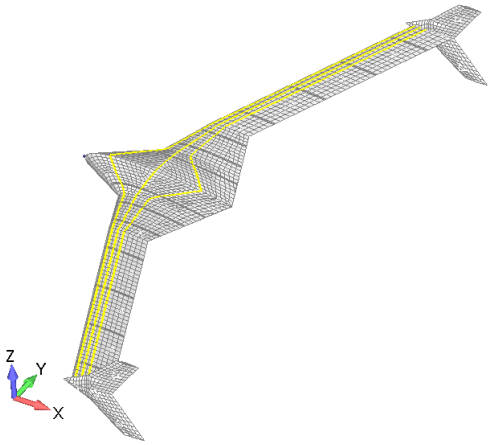


Figure 26: Highlighted spars for scaled flutter aircraft model

The frequency response computed through GROM is shown in Figure 27 for the first 8 modes (6 rigid modes, Modes 1-6 and 2 elastic modes, Modes 7-8), where a good agreement is shown with the response obtained from Nastran.

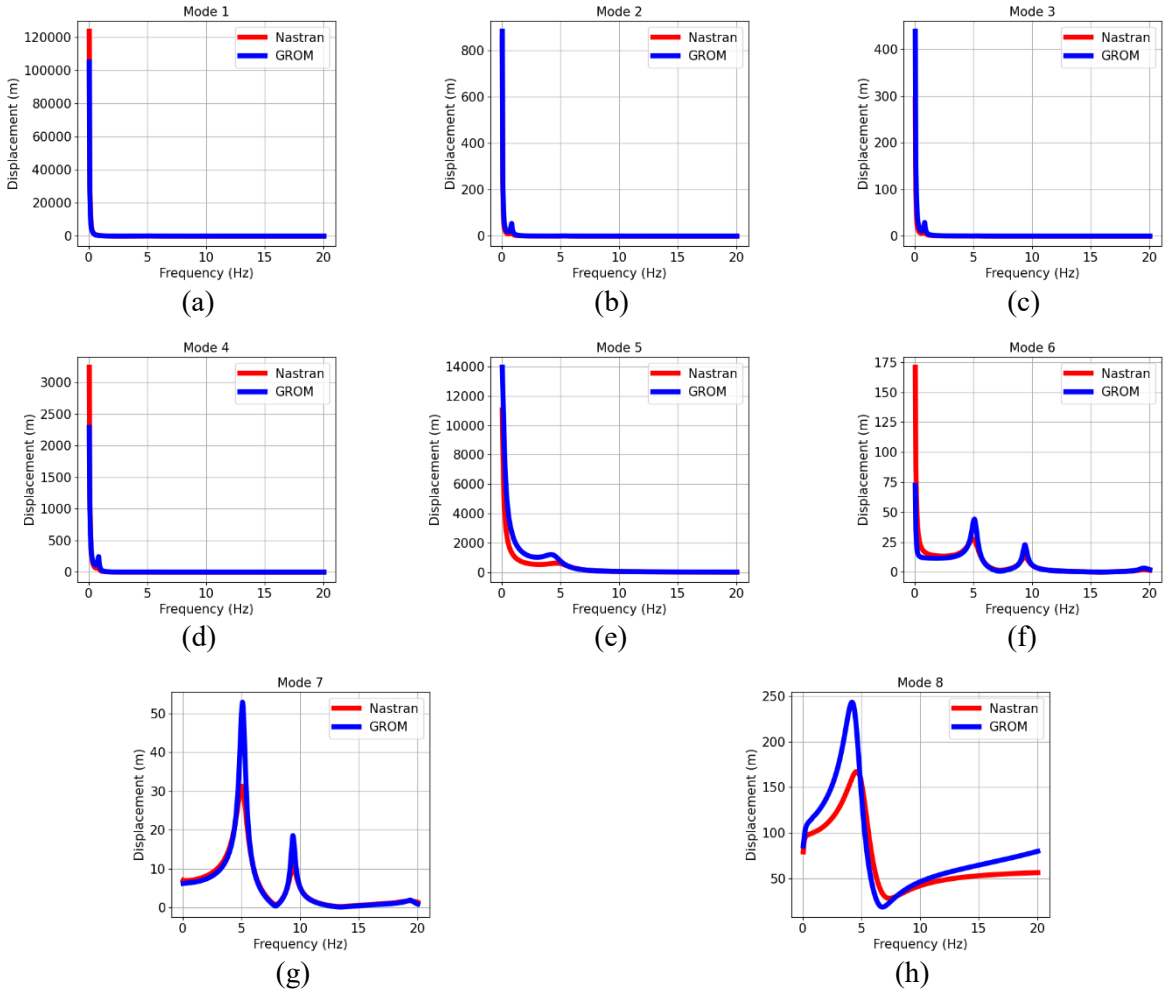
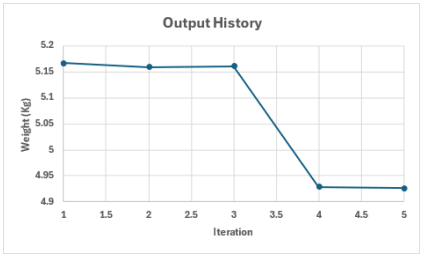
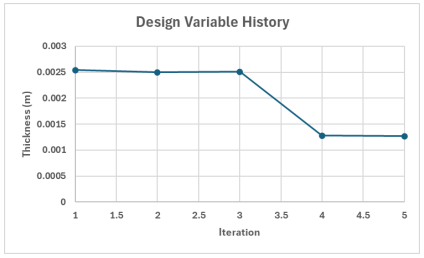


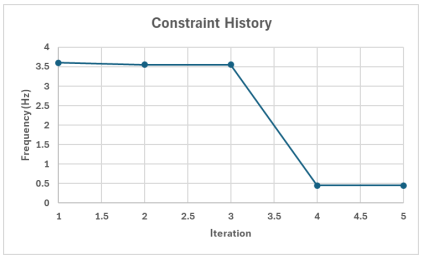
Figure 27: Comparison of the frequency response obtained from Nastran and GROM for Modes 1-8 (a-h, respectively).



(a)



(b)



(c)

Figure 28: Optimization history

The optimization results and history of the defined problem above is presented in Figure 28, showing that the optimization converged in 5 iterations, reducing the weight from around 5.17 Kg to around 4.93 Kg by reducing the spars thickness from 0.1" to 0.05" and hitting the lower bound of the frequency constraint (0.3 Hz).

Finite Element Model Development

M4 proposed to create three NASTRAN models which would be used in large scaled optimization problems. The first demonstration will be using a tube-and-wing configuration. The structural finite element model will be based on an OpenVSP model. This model can be seen in Figure 29.

Material System

The materials chosen were Aluminum 2024 T3 ("MMPDS 07," 2012) and a Cytec laminate comprised of UD Carbon: Cytec MTM45-1/12K HTS5632. Only the wing skin was made up of the composite material. The rest of the plane structure (skins, ribs, spars, frames, bulkheads) was made up of Aluminum 2024 T3. Both the aluminum and the laminate were modeled as PSHELL elements. Material properties for the UD Carbon laminate were taken from OHC tests done on a 50/40/10 ply layup of the laminate. Table 17 and Table 18 outline the mechanical properties of both the aluminum and composite materials.

Table 17: Properties of FEM aluminum material.

Properties	E, Elastic Modulus (Msi)	Shear Modulus (Msi)	Density (lb/in³)	Poisson's Ratio
Aluminum 2024 T3	10.5	10.7	0.10	0.33

Table 18: Properties of FEM composite material.

Properties	E1 (Msi)	E2 (Msi)	G12 (Msi)	ρ(lb/in³)	Poisson's Ratio
UD Carbon: Cytec MTM45-1/ 12K HTS5632 50/40/10 Laminate	10.25	4.40	3.50	0.055	0.433

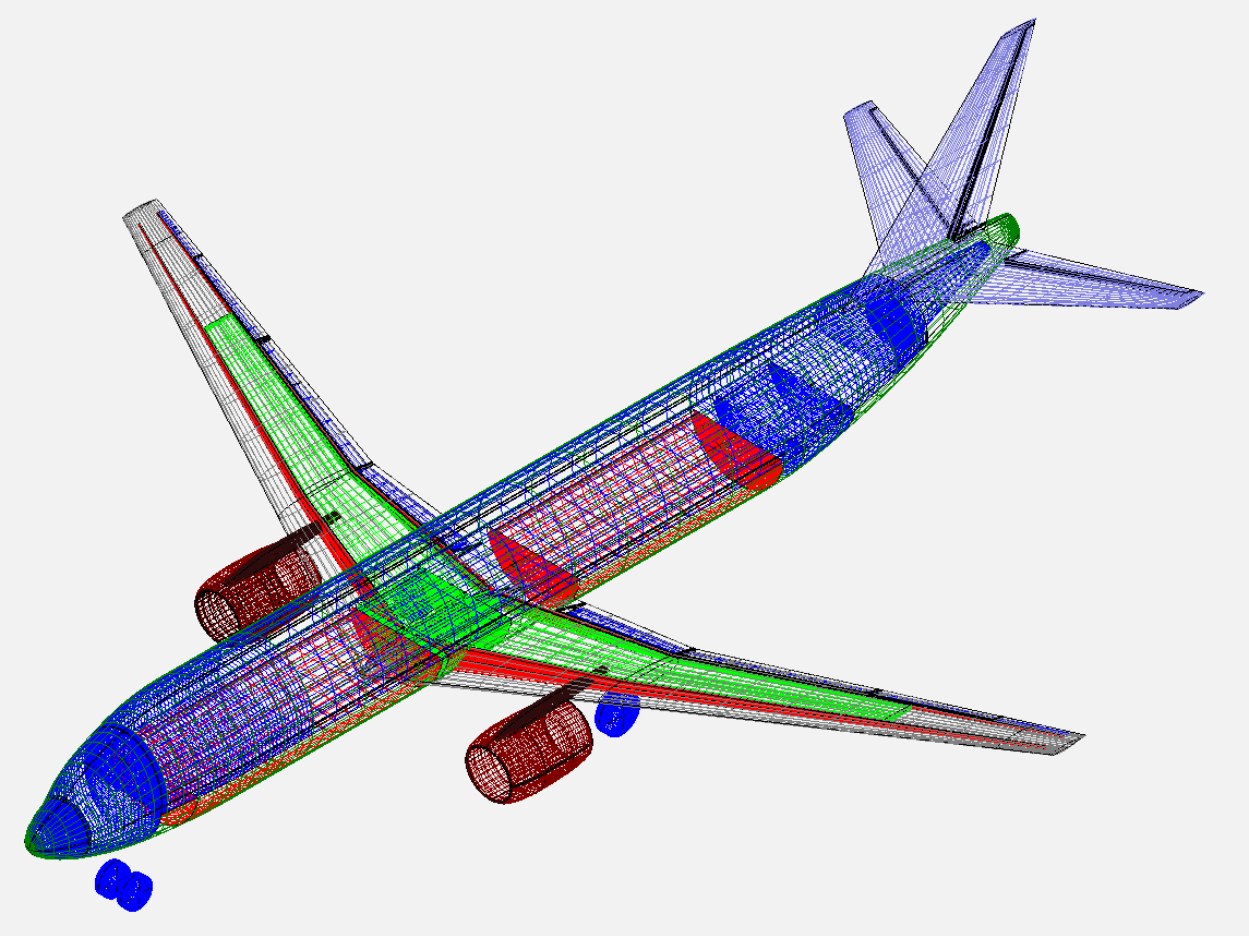


Figure 29: The N3CC OpenVSP model received.

N3CC FEM Development

The property regions defined were for differentiating the structure and non-structural components. Non-structural components are represented by either NSM smeared mass or point masses. External components such as the landing gear and powerplants were defined as point masses. Estimates for where each non-structural component was in the OpenVSP model were obtained, and property regions created in the FEM model that roughly corresponded to those locations in the OpenVSP model were defined.

For example, the Avionics component was a volume located at the front of the fuselage. The location along the length of the aircraft model of this volume was taken and a single property was defined for all elements in between these bounds. From the Aviary output, the mass of the Avionics component was taken, and using the area of the elements in property region a NSM value was calculated and applied to the FEM property.

A similar process was performed on all other fuselage, non-structural components where an additional task of defining the bounds in the vertical direction became necessary. For example, the passenger and cargo holds were would overlap in the lengthwise direction of the model, but the passenger section would be in the top half of the fuselage while the cargo holds would in the bottom half. The wing contained fuel tank masses, the dimensions of which were also obtained similarly with respect to the wing chord and span. Table 19 and Table 20 below summarize the non-structural components defined in the FEM.

Table 19: A list of non-structural components in the FEM

Non-Structural Component	Mass (lbs)	FEM Representation
Avionics	2056	Smeared NSM
Crew	675	Smeared NSM
PAX	46376	Smeared NSM
Forward Cargo	2,695	Smeared NSM
Aft Cargo	2,695	Smeared NSM
APU	1018	Smeared NSM
Aft hydraulics	222	Smeared NSM
Electric Hydraulic Bay	2,893	Smeared NSM
Nose Landing Gear	649	Point Mass
Main Landing Gear	4,755	Point Mass
Fuel Tanks	21,101	Point Mass
Powerplant	12,588	Point Mass

Table 20: Mass Property Comparison between Models

	VSP3 Model (results from Aviary)	NASTRAN FEM
Total Component NSM (lbs)	96,395	96,395
Xcg location measured from nose (in)	690.68	690.68

Once the property regions for the nonstructural components were defined, NSM was added and subtracted from different regions to mimic the overall non-structural component center of gravity location. Specifically, 880lbs of mass was moved from the Forward cargo bay to the aft cargo bay. In Figure 30 below, the Forward cargo bay is highlighted in pink and the aft cargo bay in blue.

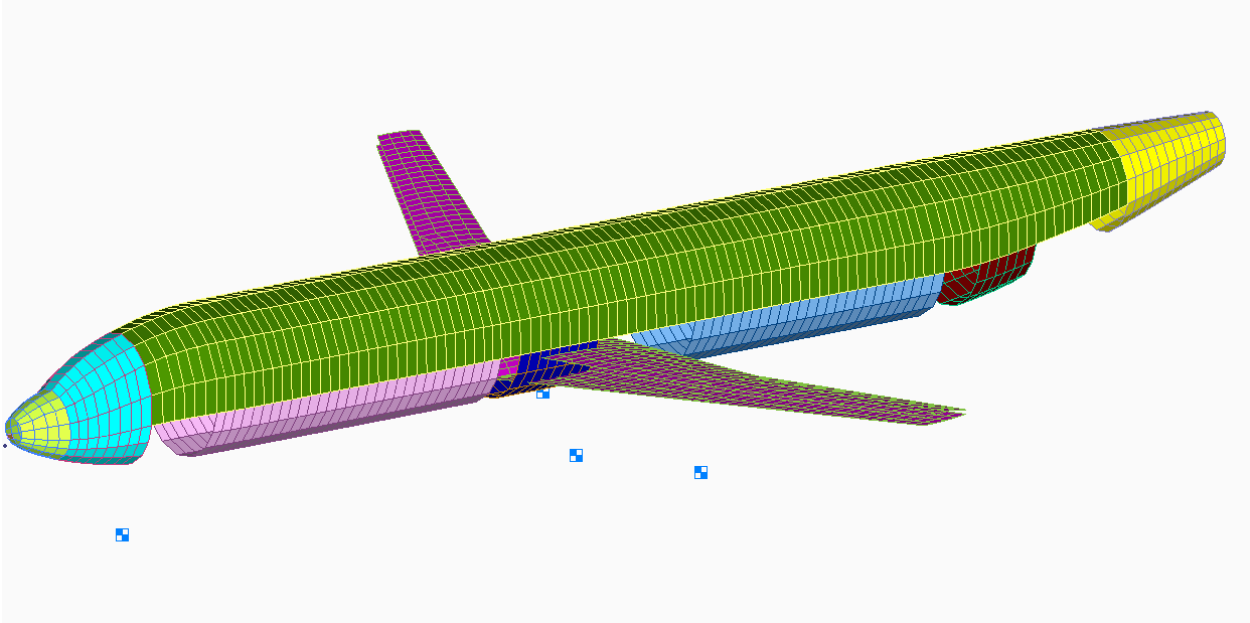


Figure 30: FEM regions with different NSM representing the non-structural components.

In addition to specifying the non-structural components in the model, the property regions would also serve as the design variables defined during the optimization process. The remaining portion of the fuselage was given independent properties. Likewise, each "band" of wing skin between ribs were given individual properties. Since a design variable can be tied to only one property region, all NSM properties would be sized independently. For example, the inner portion of the wing skin contained NSM representing the fuel tank mass. Bands of properties were defined on this portion of the wing, all with the same NSM. However, the portion of the wing that didn't carry the fuel tank mass was also defined in bands of property regions. As a result, the wing skin excluding the fuel tanks and the wing skin including the fuel tanks were all to be sized independently. In the case of this project, a decision was made to not size the non-fuel tank carrying mass wing sections, which are highlighted in pink in Figure 31 below.

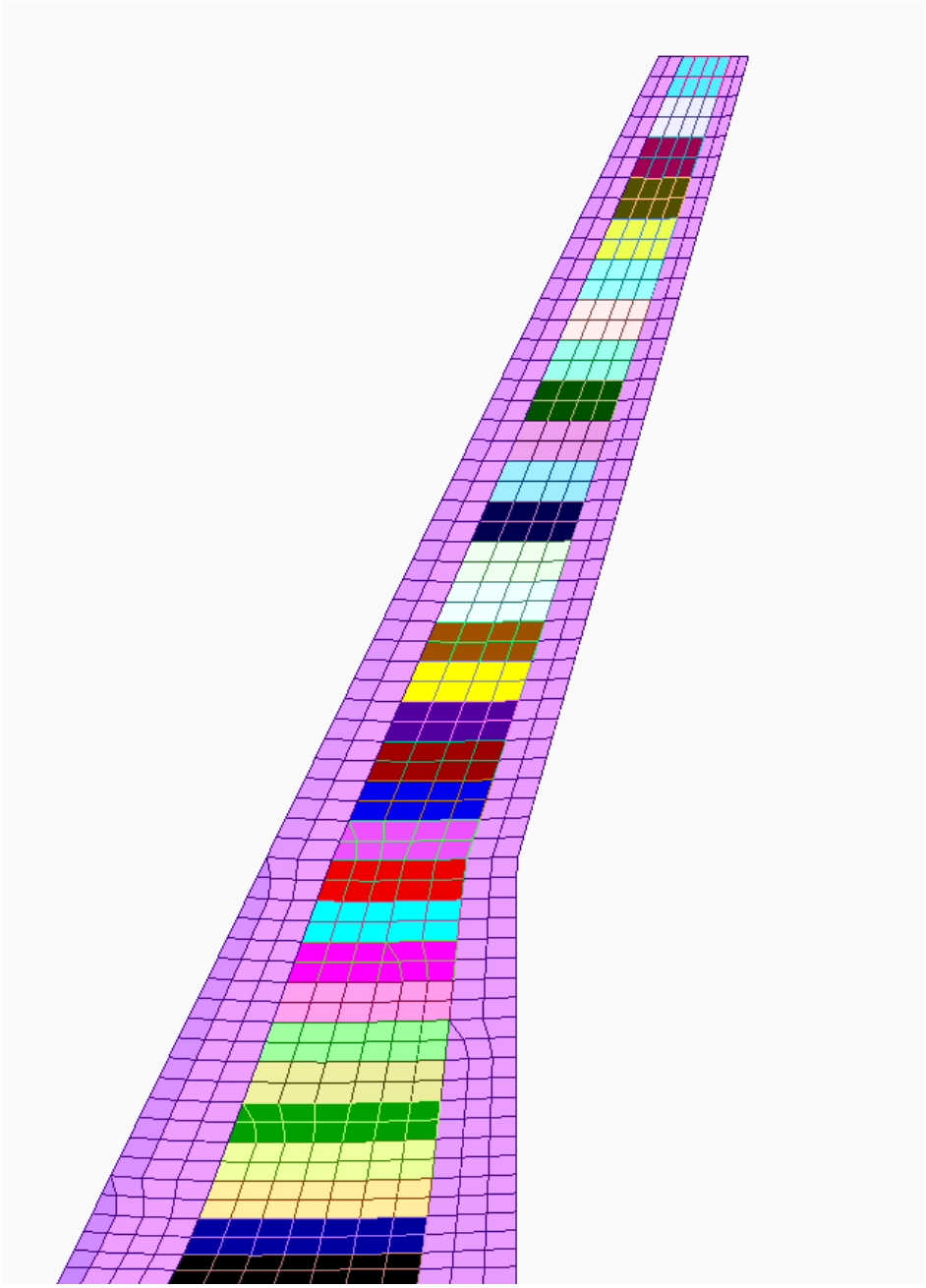


Figure 31: Different colors represent property regions defined along the wing.

Moving away from the aircraft structure, an aero model was also defined as load cases used in optimization were expected to compose of aero loads. Figure 32 below shows the splines, aero panels, and control surfaces (defined in orange). The control surfaces were all explicitly defined as found in the OpenVSP model. The wing also included independently defined inner and outer flaps, and ailerons.

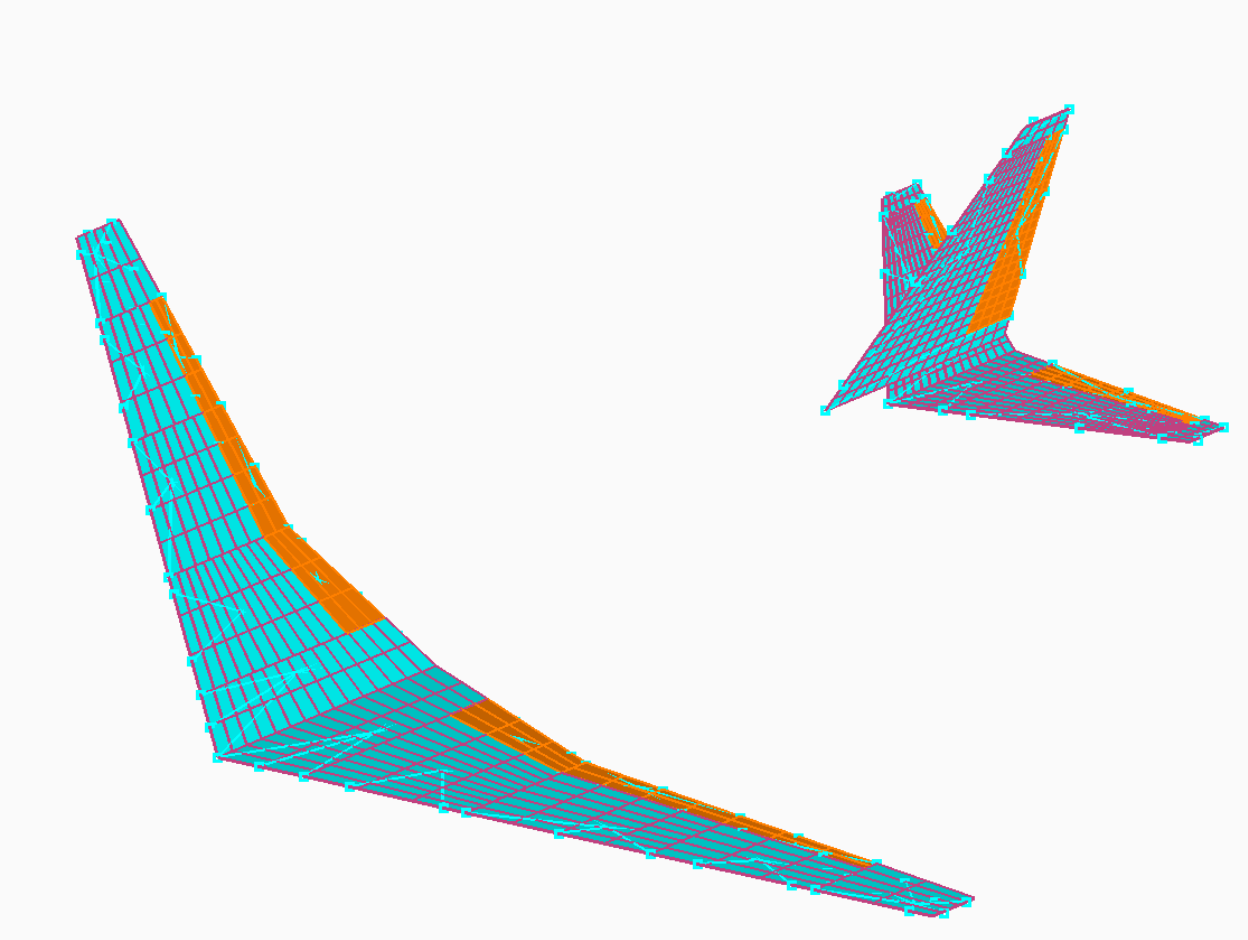


Figure 32: The aero model of the N3CC vehicle.

N3CC Analysis

M4 planned to use a variety of load case scenarios to initially size the N3CC FEM. The first load cases defined are defined in Table 21 below. The loads were defined to be from a variety of scenarios where all property regions being optimized would be affected.

Table 21: Initial Sizing Loads for the N3CC FEM.

Load	Description
2.5G vertical acceleration	Pullup maneuver
-1G vertical acceleration	Pushdown maneuver
14 psi distributed load	Representing a pressurized cabin
2G Load on Landing Gear	A landing "bump" where vertical force equivalent to 2Gs is applied to the landing gear

For the optimization process and to complement the initial load cases, constraints needed to be defined. M4 used the mechanical properties of materials in the model along with a 1.5 Safety Factor to define stress and

strain limits in the model. Skin thicknesses were used as design variables for a weight minimization optimization problem. These skin thicknesses were redefined as fixed layups of 50/40/10 with a DESVAR controlling total thickness and DVPRELs scaling plies with total thickness. This was done to avoid unnecessary complexity and reduce the size of our optimization problems. This optimization problem was set up both in OpenMDAO (utilizing Nastran sensitivities), and in Nastran alone. In OpenMDAO *super_aggr* was toggled off for more precise solutions.

Transonic Truss Braced Wing Development

The second demonstration will be using a transonic truss braced wing model. The structural finite element model will be based on an OpenVSP model. This model can be seen in Figure 33.

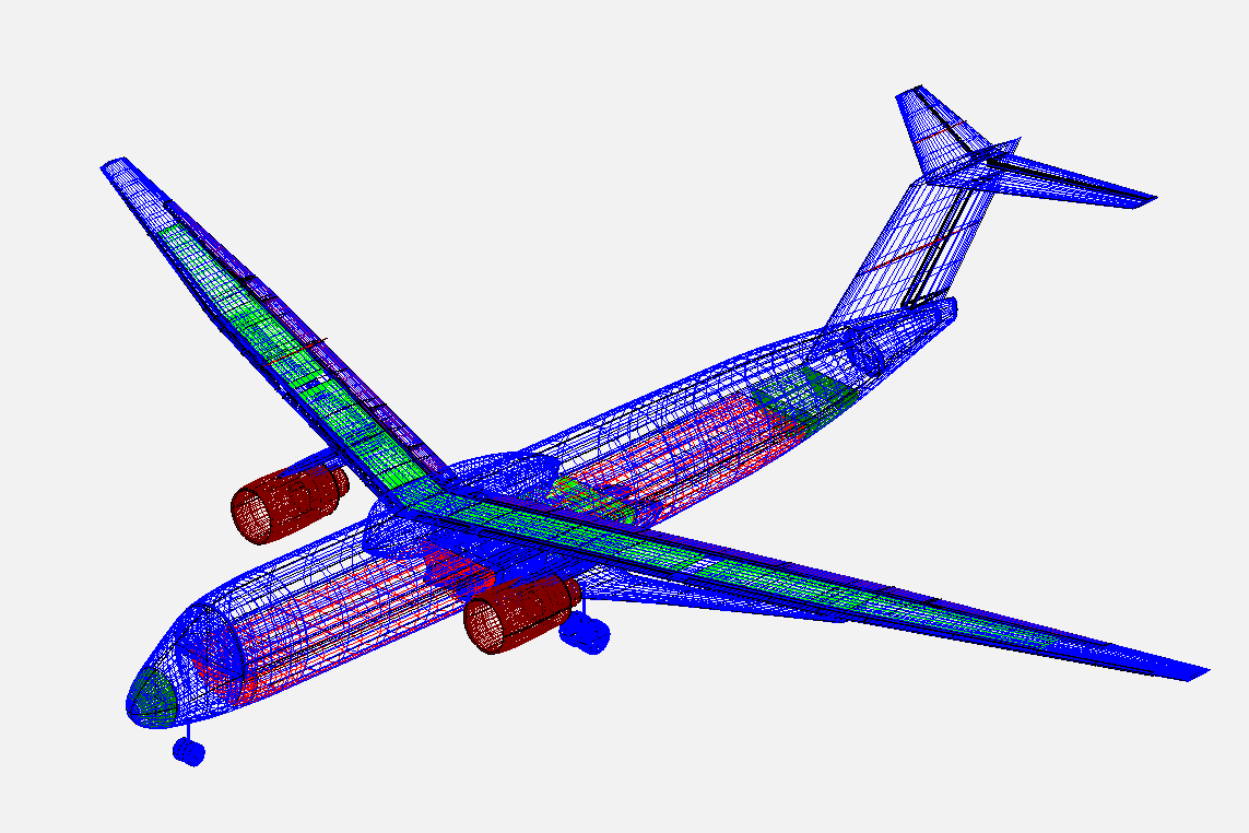


Figure 33: The TTBW OpenVSP model recieved.

The property regions defined were for differentiating the structure and non-structural components. Non-structural components are represented by either NSM smeared mass or point masses. External components such as the landing gear and powerplants were defined as point masses. Estimates for where each non-structural component was in the OpenVSP model were obtained, and property regions created in the FEM model that roughly corresponded to those locations in the OpenVSP model were defined.

For example, the Avionics component was a volume located at the front of the fuselage. The location along the length of the aircraft model of this volume was taken and a single property was defined for all elements in between these bounds. From the Aviary output, the mass of the Avionics component was taken, and using the area of the elements in property region a NSM value was calculated and applied to the FEM property. A similar process was performed on all other fuselage, non-structural components where an additional task of

defining the bounds in the vertical direction became necessary. For example, the passenger and cargo holds would overlap in the lengthwise direction of the model, but the passenger section would be in the top half of the fuselage while the cargo holds would be in the bottom half. The wing contained fuel tank masses, the dimensions of which were also obtained similarly with respect to the wing chord and span. Table 22 and Table 23 below summarize the non-structural components defined in the FEM.

Table 22: A list of non-structural components in the FEM

Non-Structural Component	Mass (lbs)	FEM Representation
Avionics	1,396	Smeared NSM
Crew	952	Smeared NSM
PAX	43,448	Smeared NSM
Forward Cargo	3,465	Smeared NSM
Aft Cargo	3,465	Smeared NSM
APU	1,000	Smeared NSM
Aft Hydraulics	289	Smeared NSM
Fuel Tanks	27,120	Smeared NSM
Control Surface	1841	Smeared NSM
De-Icing	254	Smeared NSM
Nose Landing Gear	735	Point Mass
Main Landing Gear	5,857	Point Mass
Electric Hydraulic Bay	2,594	Point Mass
Powerplant	14,700	Point Mass

Table 23: Mass Property Comparison between Models

	VSP3 Model (results from Aviary)	NASTRAN FEM
Total Component NSM (lbs)	107,117	107,115
Xcg location measured from nose (in)	680.74	702.53

The Xcg shows a difference of 3.2% between the VSP3 Model and the NASTRAN FEM. This is due to the slight differences in models between the FEM and VSP3 Model, and will be fixed by shifting weight from aft to fore cargo, or shifting locations of other NSMs.

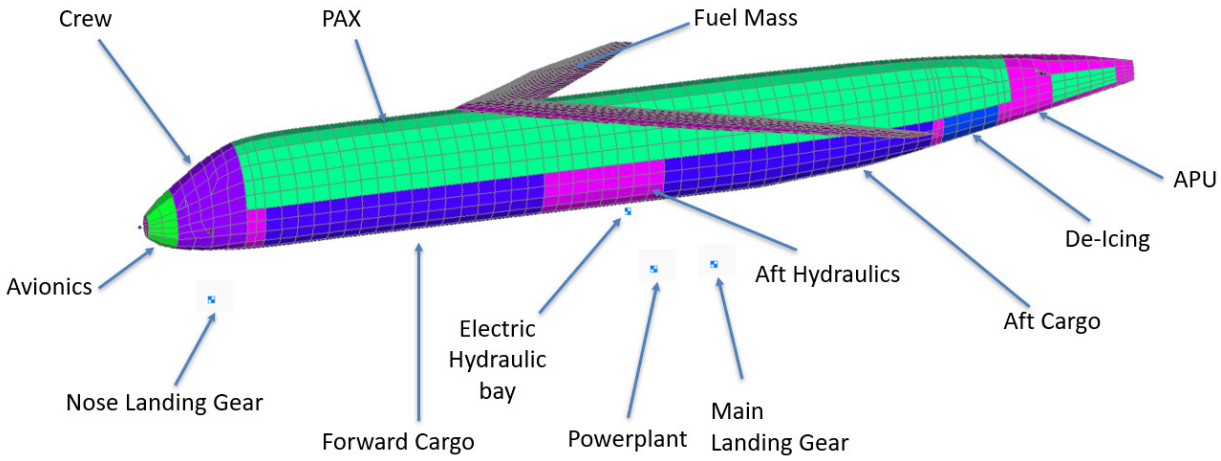


Figure 34: FEM regions with different NSM representing the non-structural components.

In addition to specifying the non-structural components in the model, the property regions would also serve as the design variables defined during the optimization process. The remaining portion of the fuselage was given independent properties. Likewise, each "band" of wing skin between ribs were given individual properties and design variables. Sections of the wing containing fuel tanks were represented by smeared NSM DVPREL all controlled by a single fuel design variable. The trailing edge skin is not sized in this case.

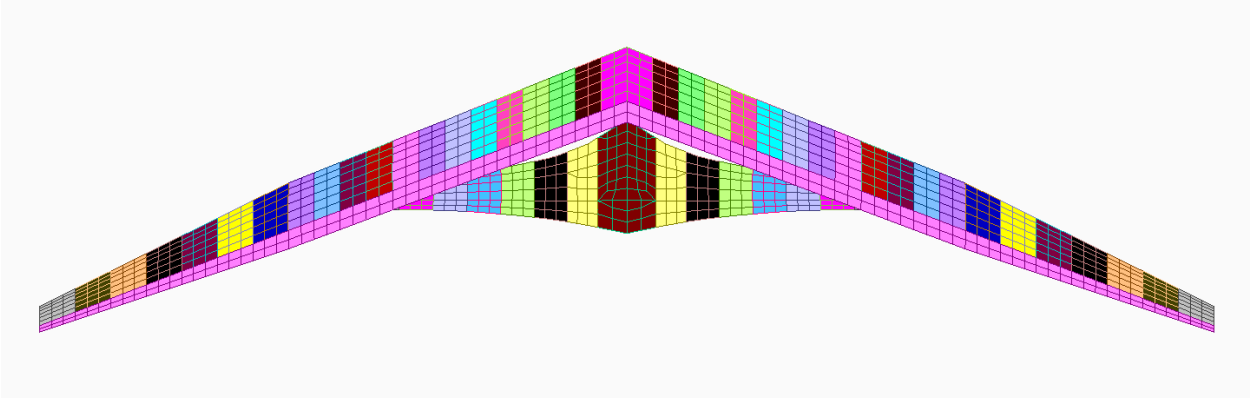


Figure 35: Different colors represent property regions defined along the wing.

Moving away from the aircraft structure, an aero model was also defined as load cases used in optimization were expected to compose of aero loads. Figure 36 below shows the splines, aero panels, and control surfaces (defined in orange). The control surfaces were all explicitly defined as found in the OpenVSP model. The wing also included independently defined both inner and outer flaps and ailerons.

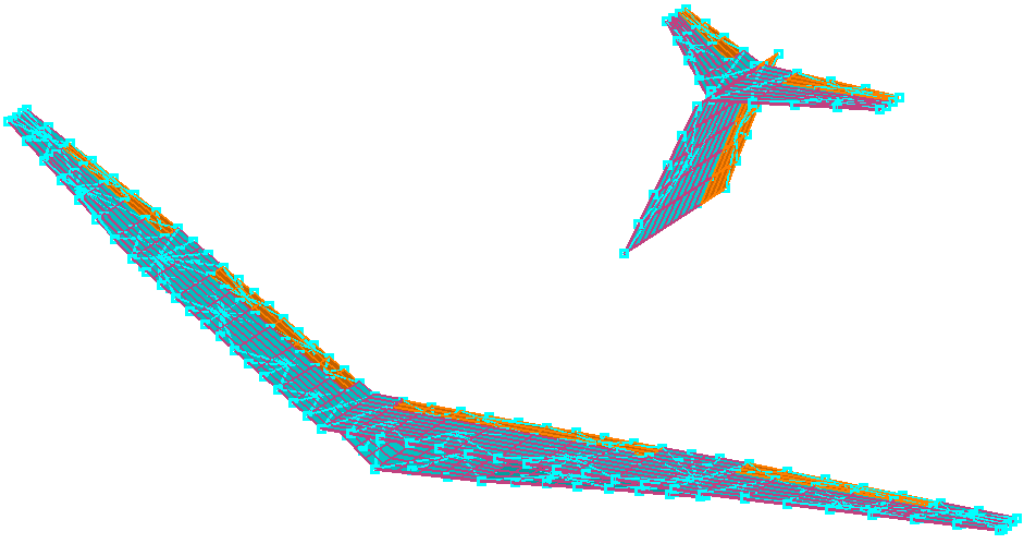


Figure 36 The aero model of the TTBW vehicle

TTBW Analysis

M4 planned to use a variety of load case scenarios to initially size the TTBW FEM. The first load cases defined are defined in Table 24 below. The loads were defined to be from a variety of scenarios where all property regions being optimized would be affected.

Table 24: Initial Sizing Loads for the TTBW FEM

Load	Description
2.5G vertical acceleration	Pullup maneuver
-1G vertical acceleration	Pushdown maneuver
14 psi distributed load	Representing a pressurized cabin
2G Load on Landing Gear	A landing "bump" where vertical force equivalent to 2Gs is applied to the landing gear

For the optimization process and to complement the initial load cases, constraints needed to be defined. M4 used the mechanical properties of materials in the model along with a 1.5 Safety Factor to define stress and strain limits in the model. Shown below are the results from the Nastran standalone optimization as well as the integrated OpenMDAO optimization.

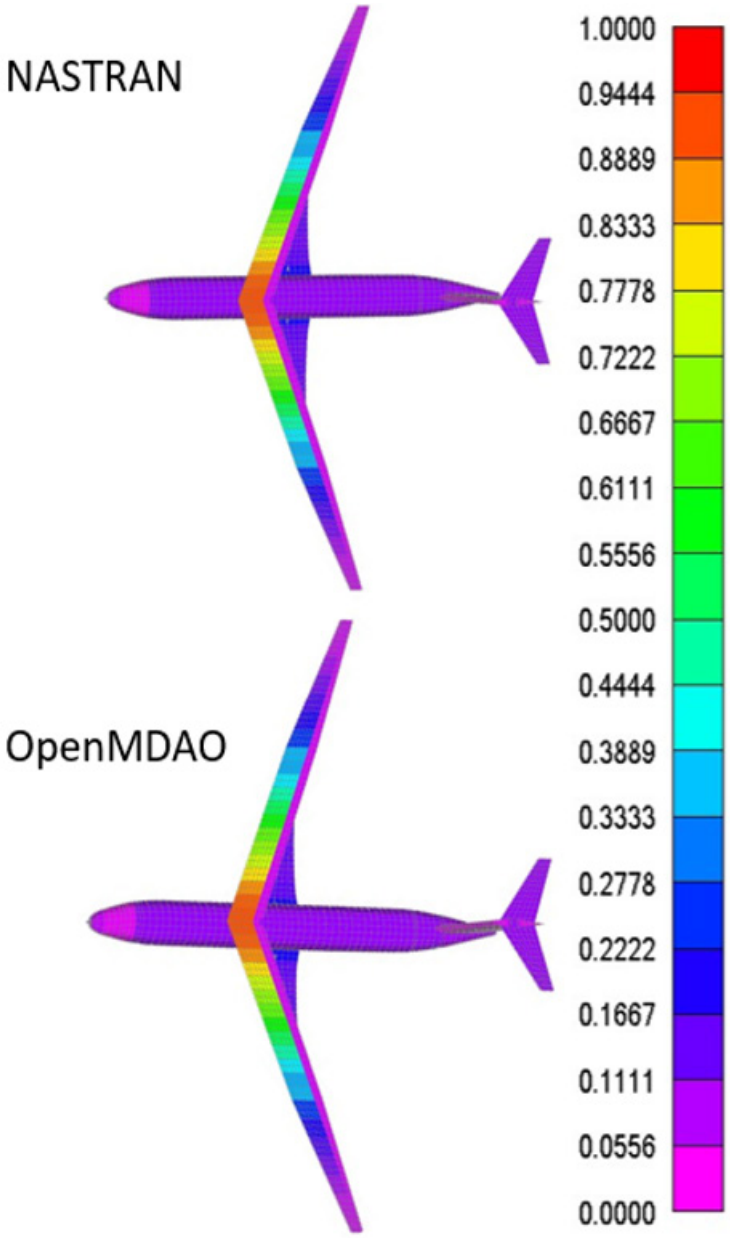


Figure 37: Nastran and OpenMDAO wing thickness comparison

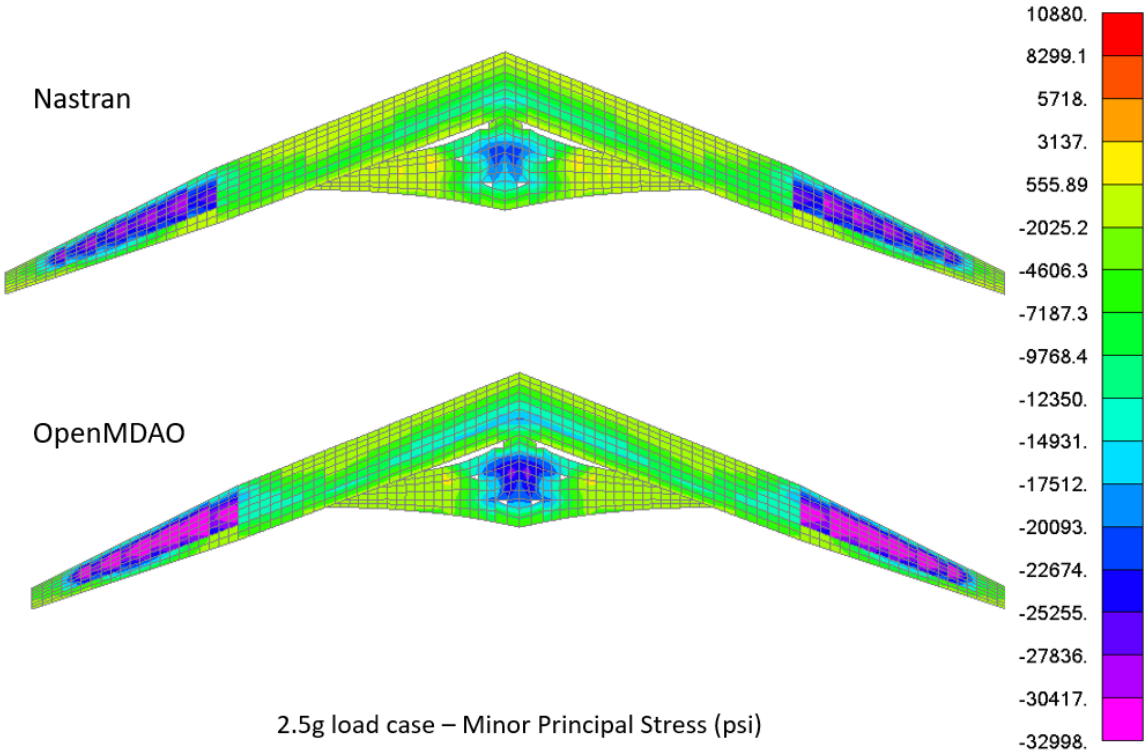


Figure 38: Nastran and OpenMDAO stress comparison.

Even though buckling of the truss is a concern, a buckling analysis wasn't included in the overall optimization because it would add to computation time. Instead, a post-optimization buckling analysis was performed. Figure 39 shows the first buckling mode of the strain constrained optimized wing. It's eigenvalue is above 2.0, which indicates that buckling will not be a concern for the strain sized wing structure.

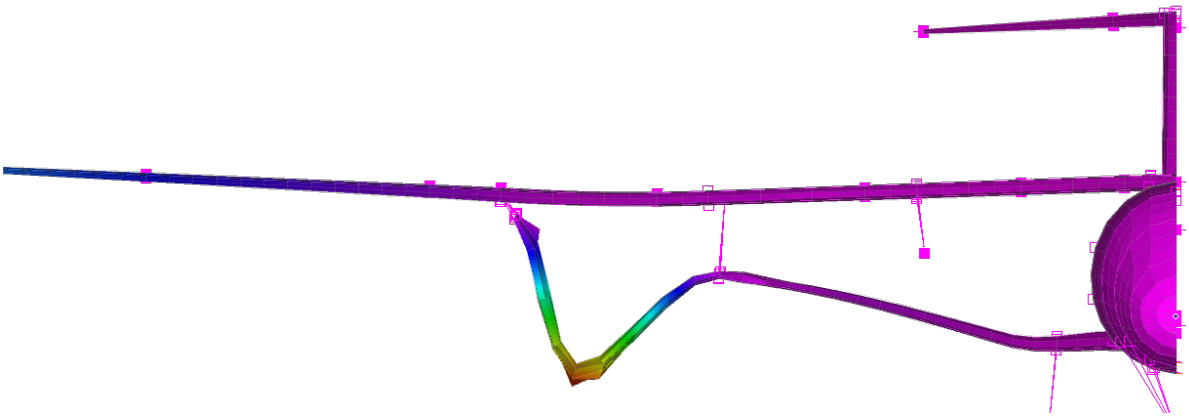


Figure 39: First buckling mode at the -1g maneuver with an eigenvalue of 2.65.

Generic Blended Wing Body

The BWB OpenVSP model provides details of the OML, as well as locations of structural members (spars, bulkheads, ribs) and nonstructural masses (NSM). This model is utilized further in conjunction with the Aviary framework, providing a table of aerodynamic lift and drag coefficients.

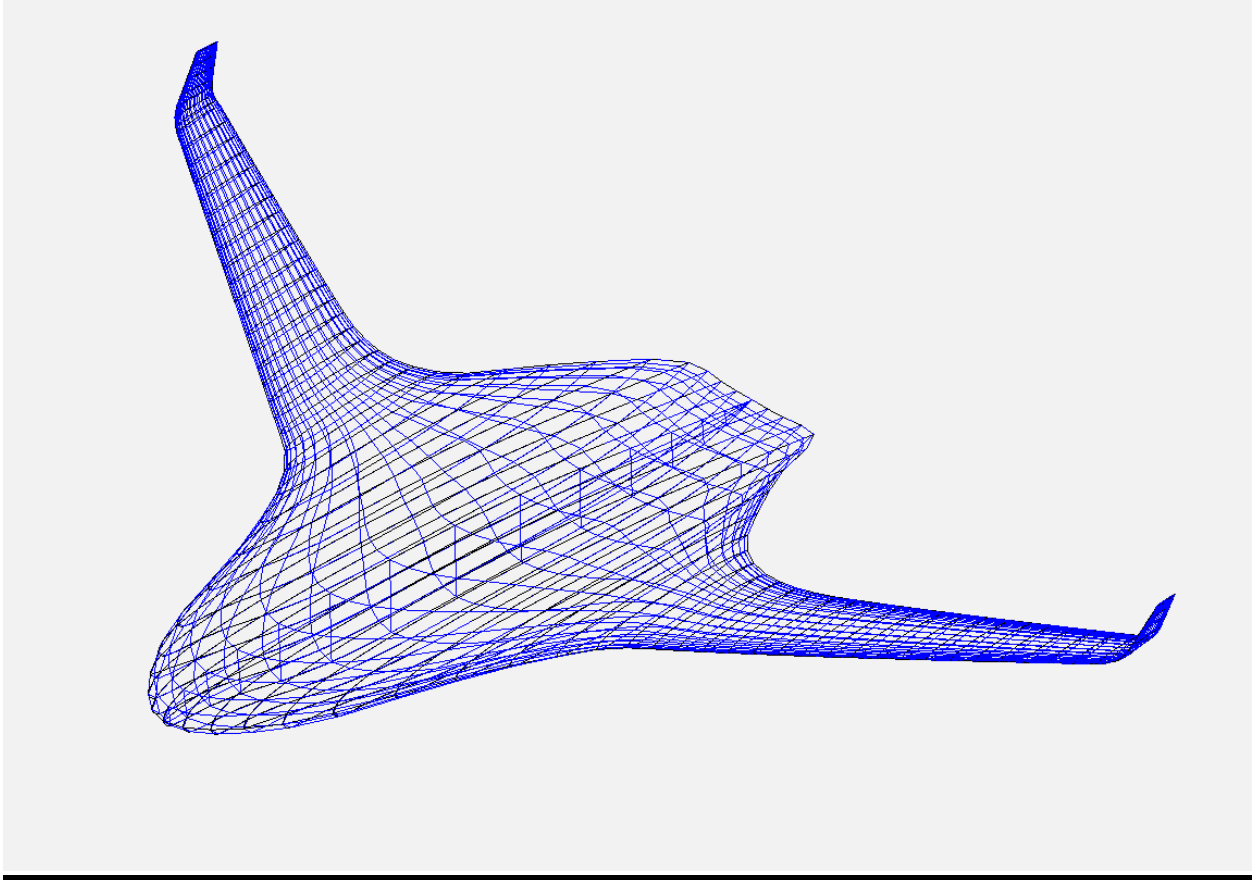


Figure 40: BWB OpenVSP Model

BWB FEM Development

The materials chosen were a Cytec composite comprised of UD Carbon: Cytec MTM45-1/12K HTS5632. The composite is modeled as a PSHELL utilizing the properties of a 50/40/10 layup. This data is obtained from material tests of the laminate, properties shown in Table 1.

Table 1: Properties of FEM composite material.

Properties	E1 (Msi)	E2 (Msi)	G12 (Msi)	ρ(lb/in³)	Poisson's Ratio
UD Carbon: Cytec MTM45-1/ 12K HTS5632 50/40/10 Laminate	10.25	4.40	3.50	0.055	0.433

The property regions defined were for differentiating the structure and non-structural components. Non-structural components are represented by either NSM smeared mass or point masses. External components such as the landing gear and powerplants were defined as point masses. Estimates for where each non-structural component was located as well as its total mass were initially guessed based on scaling of a Boeing 747 to match span.

Table 3: A list of non-structural components in the FEM

Non-Structural Component	Mass (lbs)	FEM Representation
Avionics	2083.95	Smeared NSM
Crew	675	Smeared NSM
PAX	132000	Smeared NSM
Forward Cargo	DYNAMIC	Smeared NSM
Aft Cargo	DYNAMIC	Smeared NSM
APU	2768.94	Smeared NSM
Aft hydraulics	991.2	Smeared NSM
Electric Hydraulic Bay	4063.32	Smeared NSM
Fuel Tanks	68175.31	Smeared NSM
Powerplant	19200	Point Mass
Landing gear	16964.79	Point Mass

Table 4: Mass Property Comparison between Models

	VSP3 Model (results from Aviary)	NASTRAN FEM
Total Component NSM (lbs)	96,395	96,395
Xcg location measured from nose (in)	690.68	690.68

Once the property regions for the nonstructural components were defined, NSM was added and subtracted from different regions to mimic the overall non-structural component center of gravity location. Specifically, 880lbs of mass was moved from the Forward cargo bay to the aft cargo bay. In Figure 2 below, the Forward cargo bay is highlighted in pink and the aft cargo bay in blue.

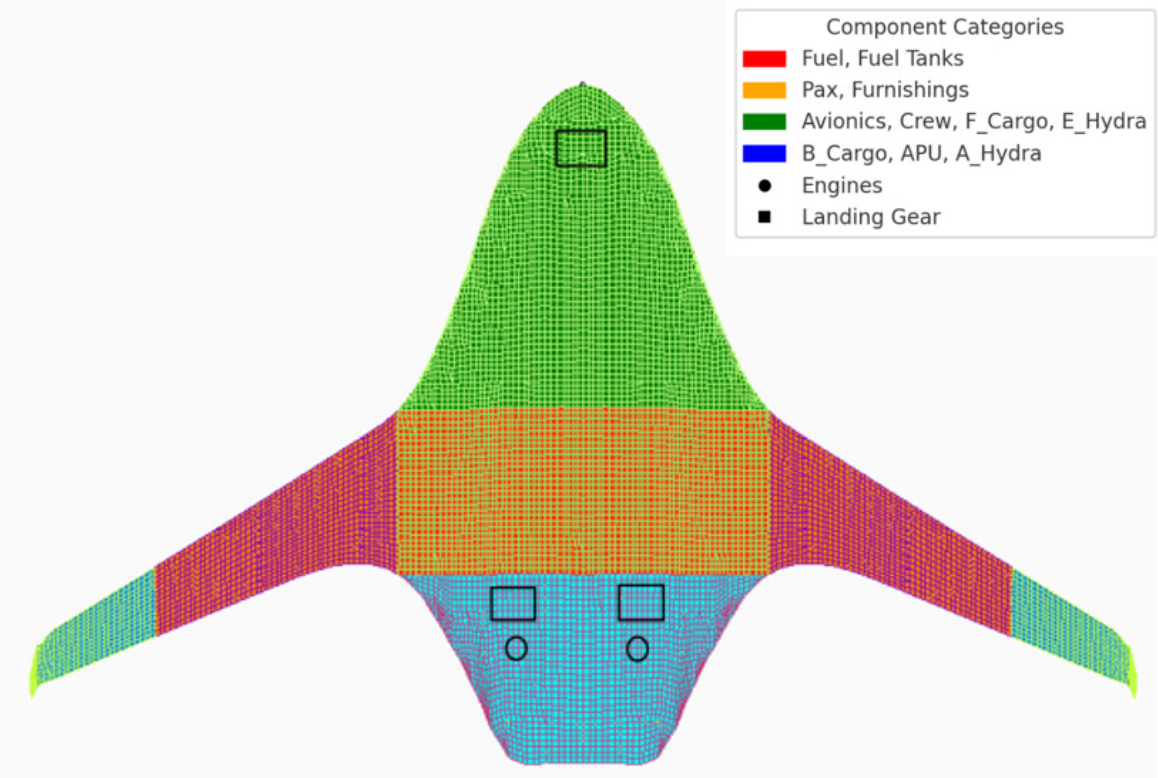


Figure 41: FEM regions with different NSM representing the non-structural components.

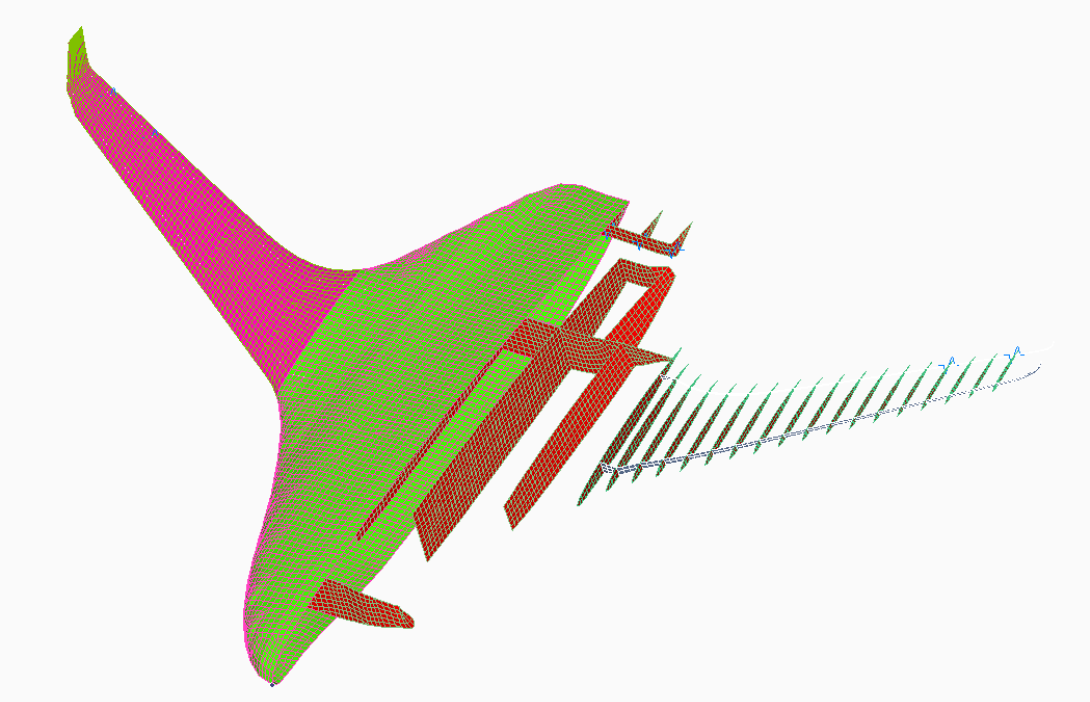


Figure 42: BWB Structural Cross-Section

In addition to specifying the non-structural components in the model, the property regions would also serve as the design variables defined during the optimization process. The remaining portion of the fuselage was given independent properties. Likewise, each “band” of wing skin between ribs were given individual properties. Since a design variable can be tied to only one property region, all NSM properties would be sized independently. For example, the inner portion of the wing skin contained NSM representing the fuel tank mass. Bands of properties were defined on this portion of the wing, all with the same NSM. However, the portion of the wing that didn’t carry the fuel tank mass was also defined in bands of property regions. As a result, the wing skin excluding the fuel tanks and the wing skin including the fuel tanks were all to be sized independently.

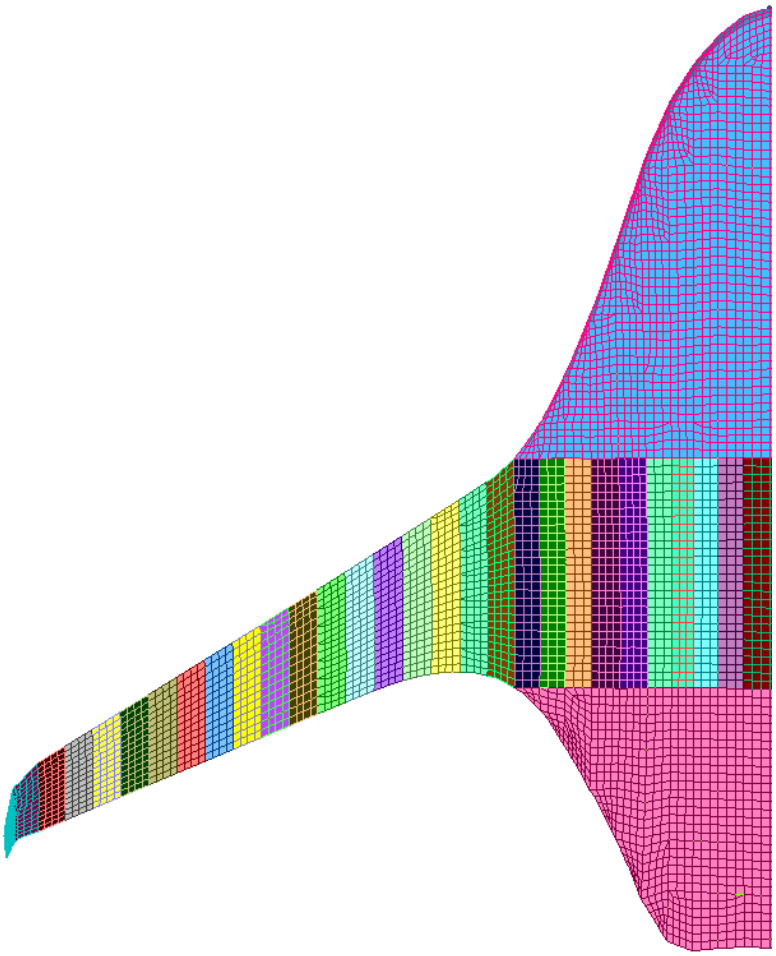


Figure 43: Different colors represent property regions defined along the wing.

Moving away from the aircraft structure, an aero model was also defined as load cases used in optimization were expected to compose of aero loads. Figure 5 below shows the splines, aero panels, and control surfaces (defined in orange). The control surfaces were all explicitly defined as found in the OpenVSP model. The wing also included independently defined inner and outer flaps, and ailerons.

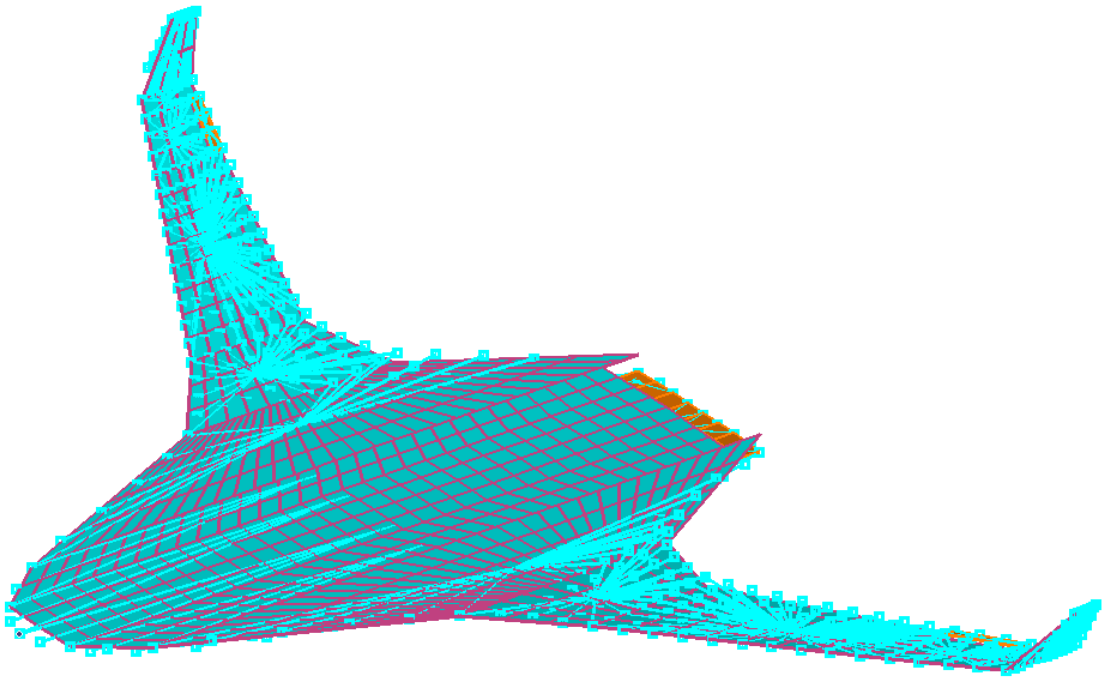


Figure 44: The aero model of the N3CC vehicle.

BWB Analysis

For the BWB aircraft, a single 2.5g pullup load case was chosen for optimization. It was decided to exclude the pressure load case due to the complex nature of the pressure vessel shape.

Table 5: Initial Sizing Loads for the N3CC FEM.

Load	Description
2.5G vertical acceleration	Pullup maneuver

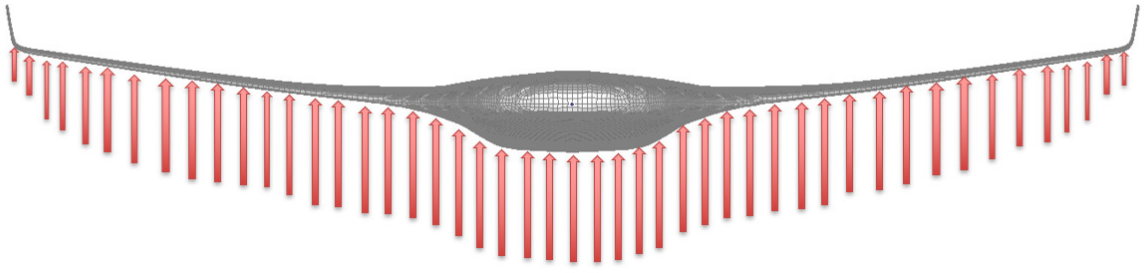


Figure 45: BWB 2.5g Pullup Load Case

For the optimization process and to complement the initial load cases, constraints needed to be defined. M4 used the mechanical properties of materials in the model along with a 1.5 Safety Factor to define stress and strain limits in the model. Skin thicknesses were used as design variables for a weight minimization optimization problem. These skin thicknesses were redefined as fixed layups of 50/40/10 with a DESVAR controlling total thickness and DVPRELs scaling plies with total thickness. This was done to avoid unnecessary complexity and reduce the size of our optimization problems. This optimization problem was set up both in OpenMDAO (utilizing Nastran sensitivities), and in Nastran alone. In OpenMDAO *super_aggr* was toggled off for more precise solutions.

Aviary Integration

Aviary N3CC Integration

M4 began the integration of the previously developed OpenMDAO Nastran system into the tube and wing Aviary framework. This process involved overriding empirical Aviary subsystems with Nastran definitions as well as introducing Nastran structural design variables into the global aviary design space as shown in Figure 46.

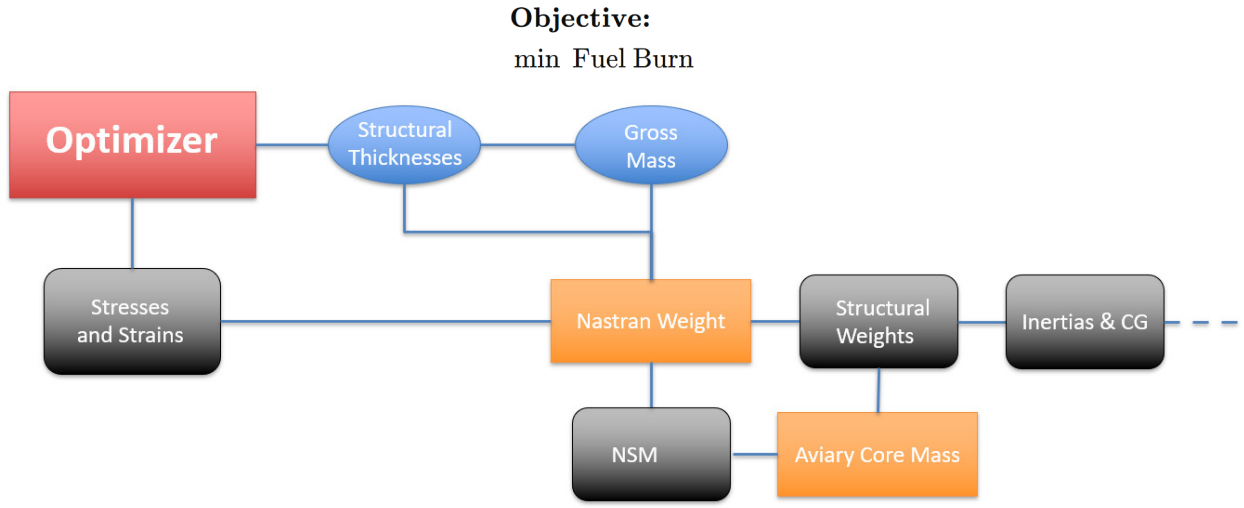


Figure 46: Nastran Subsystem Diagram

The Nastran subsystem takes in Aviary NSM and utilizes the OpenVSP mapping to correlate Aviary data to smeared Nastran NSM. It also takes in the design gross mass and similarly maps it to a smeared fuel NSM on the wings. The design gross mass is taken as a global design variable, and thus, fuel mass must be computed internally as gross mass minus empty mass to meet gross mass requirements. The new structural masses from Nastran are then fed downstream to update the Aviary core mass subsystem. Additionally, the inertias and CG location are fed downstream to be later used by mission processes as well as post-mission stability analysis. The basic mission layout is described in Table 25. Two different mission profiles were tested, one with a fuel burn objective, and the other with an empty weight objective. The design variables and constraints remain consistent for both objectives: one objective calculates fuel burn using an Aviary module, while the other incorporates empty weight as an output from the Nastran subsystem. Since the aerodynamic system relies on fixed lift coefficient (CL) and drag coefficient (CD) tables and does not account for variations in other aerodynamic shape parameters (such as twist, incidence, and span) due to the fixed outer mold line (OML), the two objectives are found to be effectively equivalent. From this point on we will be solely using the empty weight objective as it uses fewer model connections and converges faster.

Table 25: Description of Aviary phases

Phase	Phase Design Variables	Phase Constraints	Phase Comps
Pre-Mission	• Structural thicknesses • Gross mass	• Structural stresses/strains • Mass residual	• Subsystem weights • Fuel mass • Inertias • CG
Takeoff	N/A	N/A	• T/O distance, mass, final alt, VTO
Climb/Cruise/Descent/Reserve	• Times • Distance • Altitude (5 segments fitted to 3rd order poly	• Init and final altitudes and Mach num fixed • Time, Mach, and altitude bounds	• Throttle setting (path constrained)
Landing	N/A	N/A	• Ground roll • Approach velocity
Post-Mission	N/A	• Range residual	• Static and dynamic stability • Fuel burn

Nastran Model Setup

Non-structural masses were integrated into the model using DESVAR cards to facilitate access to partial derivatives. Corresponding DVPRELs (for skin) and DVCRELs (for engine and landing gear) were generated for each smeared property. The relationship between DESVAR and DVPREL is depicted in Figure 47, where COEFi is calculated as the reciprocal of the smeared area. For the gross mass DESVAR, C0 is assigned a value equal to the negative of the empty mass, ensuring the proper allocation of fuel weight to the wing sections.

$$P_i = C0 + \sum_i \text{COEFi} \cdot \text{DVIDi}$$

Figure 47: DVPREL Relation

The system is integrated into Aviary using the subsystem builder, connecting various inputs and outputs as demonstrated in the code snippet in Figure 29. Nastran thicknesses are elevated to the global level as design variables, while weights and inertias are promoted to override Aviary's default calculations. Additionally, Aviary NSM values are linked to their corresponding Nastran equivalents.

Analysis

The N3CC model was analyzed with 4 different methods. Namely the Nastran-wrapped OpenMDAO, NX Nastran, MSC Nastran, and the Aviary integrated models are shown here. The Nastran-wrapped OpenMDAO as well as the Nastran-wrapped Aviary models were both run utilizing the SNOPT optimizer. NX Nastran and MSC Nastran both run on internal optimizers for which little tuning is available. These are both setup as Sol 200 optimizations. A comparison of results for these models is presented below in Figure 9 and Figure 10. Optimized weights are then compared in Table 7. Weights for the 3 non-Aviary problems are derived from the initial Aviary results table from an optimization without Nastran. After integration of the Nastran subsystem, a new slightly different optimized fuel weight is obtained. For completeness, this weight is plugged back in to the OpenMDAO problem and a final optimized weight is solved for. From the results, we can see that

the Aviary global optimization performs notably worse than the independent optimization, likely due to the larger nature of the problem. For convergence on the Nastran-wrapped Aviary problem, lower convergence criteria must be input to obtain convergence in a reasonable time. Also of note, MSC Nastran performs significantly worse than the other optimizations. This is mainly due to the lack of tuning option availability rather than a more general problem with MSC as the optimization converges quickly to this less than optimal solution. These results demonstrate the advantage of working within OpenMDAO, in that a significant amount of fine tuning is readily available to the user.

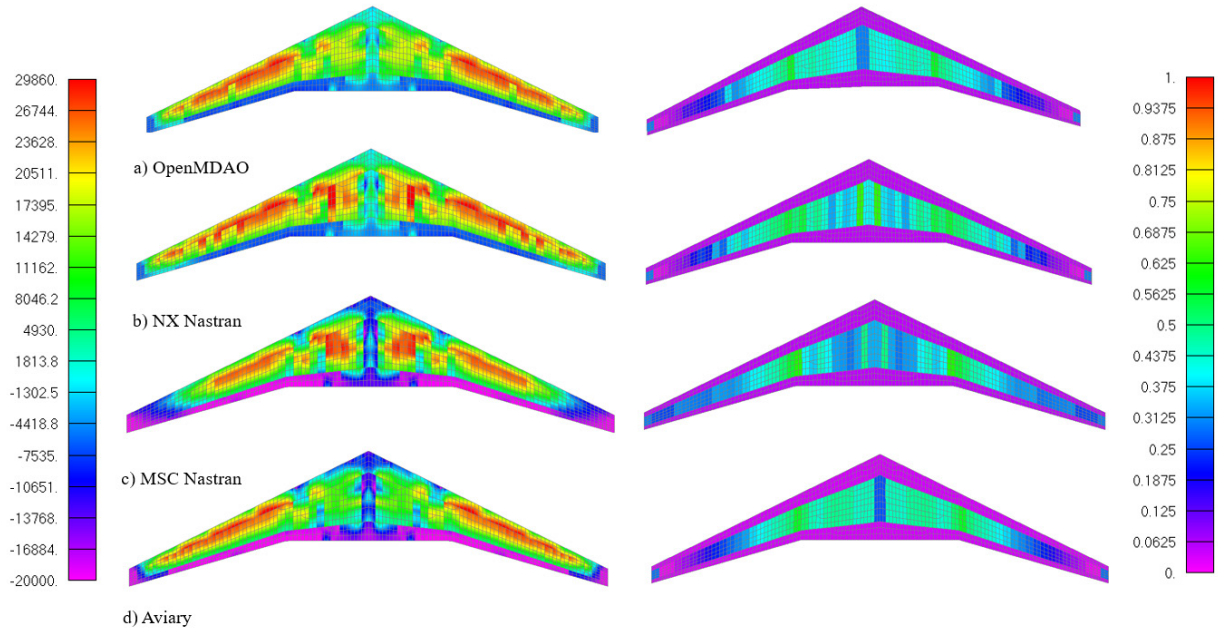


Figure 43: Comparison of optimized wing for Nastran, OpenMDAO, and integrated Aviary models

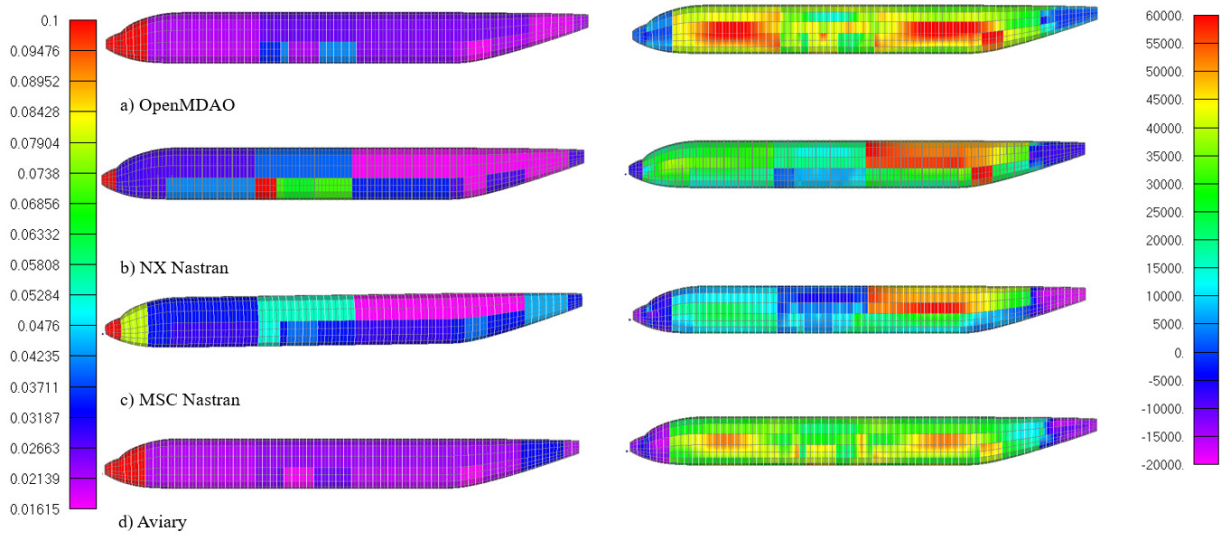


Figure 44: Comparison of optimized fuselage for Nastran, OpenMDAO, and integrated Aviary models

Table 30: N3CC OpenMDAO and Nastran Results Comparison

Component	NX Nastran	MSC Nastran	OpenMDAO	Aviary	Updated Fuel OpenMDAO
Wing Struct Weight(lb)	5,612.57	7,508.30	5,492.245	5,927.55	5,492.245
Fuel Weight (lb)	27,037	27,037	27,037	27,068.28	27,068.28
Fuselage Weight(lb)	66,921.76	68,184.30	66,846.8	66,957.7	66,878.79
Total Weight(lb)	128.291	131,449.17	128,095.92	128,673.4	128,159.19

Aviary-Nastran TTBW Integration

A similar integration was done with the Transonic Truss-Braced Wing concept. The nonstructural masses shown in the model development were assigned to DVPREL cards and promoted to global optimization. To determine the accuracy of the empirical methods compared to the Nastran integration, the final optimized weights are compared below.

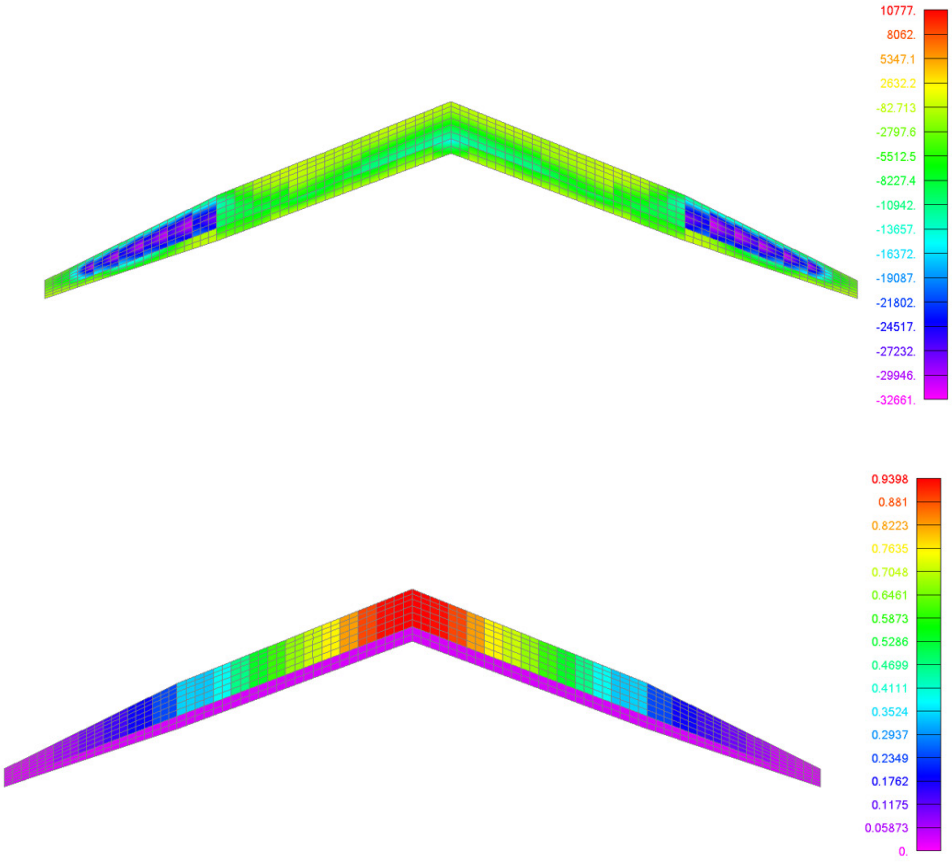


Figure 9: Aviary Optimized Stresses (Top) and Thicknesses (Bottom)

Table 26: TTBW Optimization Comparison

	Aviary Solo	Aviary Integrated
Wing + Strut + Jury Structural Weight (lb)	18,939.2	11,514.7
Fuselage Structural Weight (lb)	17,564.9	7,316
Horizontal Structural Tail (lb)	1,352.7	1,429.1
Vertical Structural Tail (lb)	1,847.9	1,175.5
Total Structural Weight (lb)	**39,704.7**	**21,435.3**
Fuel Weight - NSM (lb)	26,121.5	25,456.8
NSM Weight except Fuel Weight (lb)	81,634.5	80,638.38
Total Component NSM (lb)	**107,756**	**106,095.2**
Total Weight (lb)	**147,460.7**	**127,530.5**

Aviary Nastran BWB Integration

M4 began the integration of the previously developed OpenMDAO Nastran system into the Blended Wing Body Aviary framework. This process involved overriding empirical Aviary subsystems with Nastran definitions as well as introducing Nastran structural design variables into the global aviary design space as shown in Figure 7. For the BWB, empirical data from a B747 was taken as an initial starting point for much of the nonstructural mass, as well as the mission and flight parameters. These values were then optimized by the aviary system and more accurate nonstructural masses were computed. The BWB keeps in line with the N3CC model, aiming for a range of 3500nmi. As a result of its much larger size, the mission climb rate is significantly reduced from that of a B747.

The Nastran subsystem takes in Aviary NSM and utilizes the OpenVSP mapping to correlate Aviary data to smeared Nastran NSM. It also takes in the design gross mass and similarly maps it to a smeared fuel NSM on the wings. The design gross mass is taken as a global design variable, and thus, fuel mass must be computed internally as gross mass minus empty mass to meet gross mass requirements. The new structural masses from Nastran are then fed downstream to update the Aviary core mass subsystem. Additionally, the inertias and CG location are fed downstream to be later used by mission processes as well as post-mission stability analysis. The basic mission layout is described in Table 6. Two different mission profiles were tested, one with a fuel burn objective, and the other with an empty weight objective. The design variables and constraints remain consistent for both objectives: one objective calculates fuel burn using an Aviary module, while the other incorporates empty weight as an output from the Nastran subsystem. Since the aerodynamic system relies on fixed lift coefficient (CL) and drag coefficient (CD) tables and does not account for variations in other aerodynamic shape parameters (such as twist, incidence, and span) due to the fixed outer mold line

(OML), the two objectives are found to be effectively equivalent. For this model, tabulated aero data was computed from a panel method. From this point on we will be solely using the empty weight objective as it uses fewer model connections and converges faster.

Nastran Model Setup

Non-structural masses were integrated into the model using DESVAR cards to facilitate access to partial derivatives. Corresponding DVPRELs (for skin) and DVCRELs (for engine and landing gear) were generated for each smeared property. The system is integrated into Aviary using the subsystem builder, connecting various inputs and outputs. Nastran thicknesses are elevated to the global level as design variables, while weights and inertias are promoted to override Aviary's default calculations. Additionally, Aviary NSM values are linked to their corresponding Nastran equivalents.

BWB Analysis Results

The BWB model was analyzed with 4 different methods. Namely the Nastran-wrapped OpenMDAO, NX Nastran, MSC Nastran, and the Aviary integrated models are shown here. The Nastran-wrapped OpenMDAO as well as the Nastran-wrapped Aviary models were both run utilizing the SNOPT optimizer. NX Nastran and MSC Nastran both run on internal optimizers for which little tuning is available. These are both setup as Sol 200 optimizations. A comparison of results for these models is presented below in Figure 9. Optimized weights are then compared in Table 7. Weights for the 3 non-Aviary problems are derived from the initial Aviary results table from an optimization without Nastran. After integration of the Nastran subsystem, a new slightly different optimized fuel weight is obtained. For completeness, this weight is plugged back in to the OpenMDAO problem and a final optimized weight is solved for. From the results, we can see that the Aviary global optimization performs notably worse than the independent optimization, likely due to the larger nature of the problem. For convergence on the Nastran-wrapped Aviary problem, lower convergence criteria must be input to obtain convergence in a reasonable time. Also of note, MSC Nastran performs significantly worse than the other optimizations. This is mainly due to the lack of tuning option availability rather than a more general problem with MSC as the optimization converges quickly to this less than optimal solution. These results demonstrate the advantage of working within OpenMDAO, in that a significant amount of fine tuning is readily available to the user.

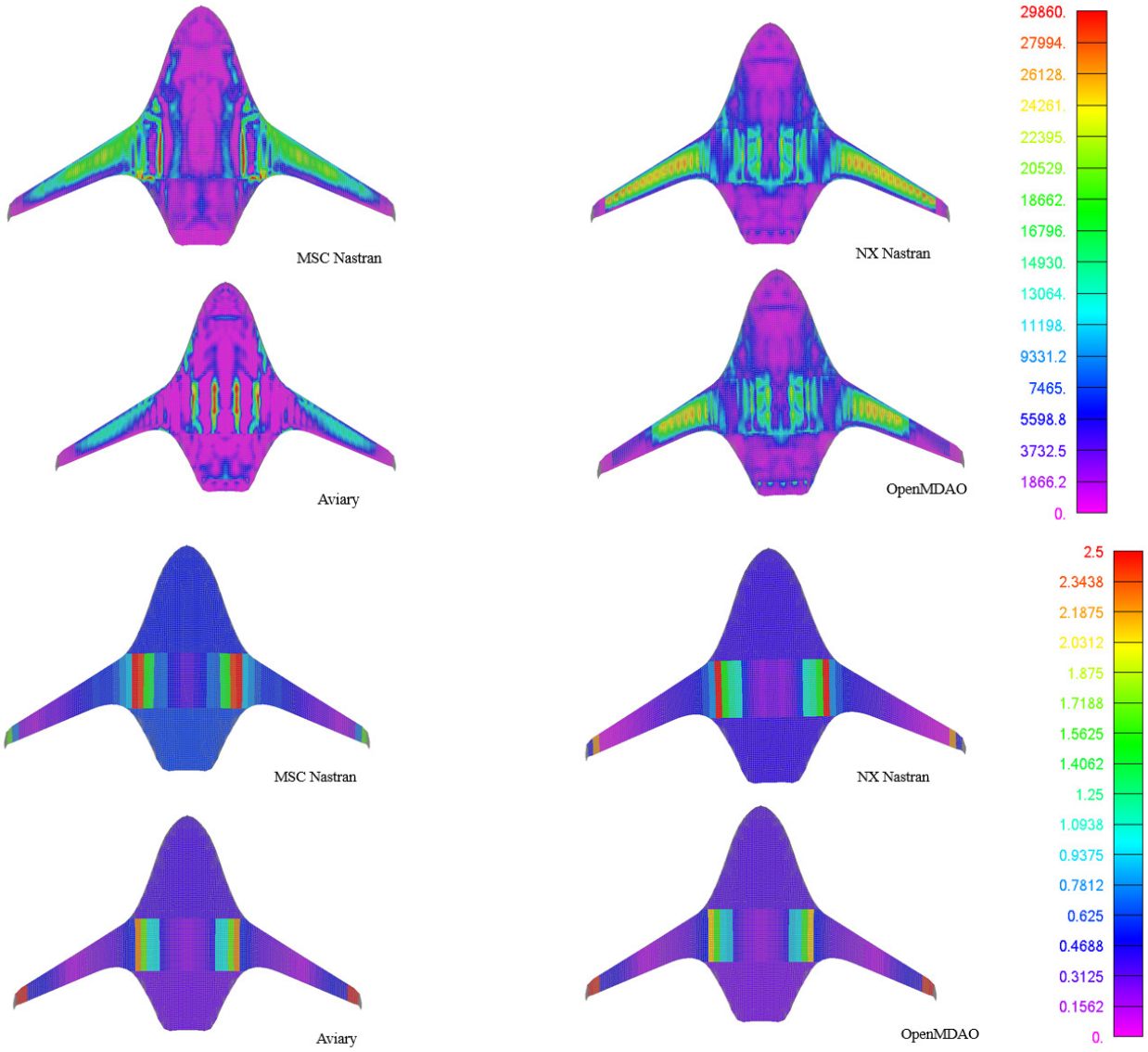


Figure 9: Comparison of optimized wing stresses and thicknesses, respectively for Nastran, OpenMDAO, and integrated Aviary models

Table 7: BWB OpenMDAO and Nastran Results Comparison

Component	NX Nastran	MSC Nastran	OpenMDAO	Aviary Integrated
Struct Weight(lb)	188,253	196,563	187,894	194,678
Fuel Weight (lb)	68,175	68,175	68,175	68,175
Total Weight(lb)	462,991	489,557	462,119	468,274

Discussion and Conclusions

The integration of NASTRAN with OpenMDAO has proven to be both technically feasible and practically valuable for enhancing multidisciplinary design workflows. By automating the generation of OpenMDAO components directly from existing NASTRAN SOL 200 models, this effort enables engineers to repurpose mature structural models within high-level optimization frameworks without substantial rework.
The demonstration cases, including linear statics, buckling, static aeroelastic, and flutter, validated the interface across a wide range of NASTRAN solution sequences. Results obtained through OpenMDAO closely matched those from native NASTRAN optimization runs, confirming the fidelity of the integration. Furthermore, performance benchmarking revealed that the primary computational cost remains in the solver itself, with minimal overhead introduced by the Python wrappers and OpenMDAO infrastructure. The added ability to extract and use sensitivity matrices directly, bypassing large output files, significantly improved data parsing efficiency.

Additional value was demonstrated by integrating third-party aerodynamic solvers, such as a transonic CFD tool and GROM, and applying the framework to realistic configurations including the TTBW, BWB, and N3CC aircraft. These integrations, along with deployment into NASA's Aviary framework, highlight the extensibility of the approach and its readiness for broader application in conceptual and preliminary design environments. Creating optimization problems in this manner results in order of magnitude decreases in cost as compared to single physics optimizations. While the optimality of the solutions is lacking compared to these standalone optimizations, global solutions provide an efficient suitable starting point for design iteration.
In conclusion, this work delivers a reusable, extensible, and validated framework for embedding high-fidelity structural models within larger multidisciplinary analysis and optimization processes. It lays a solid foundation for future research involving full-vehicle optimization, surrogate modeling, and multi-physics integration within aircraft optimization problems.

Bibliography

[1] Siemens, Design Sensitivity and Optimization User's Guide, 2019.

[2] Siemens, "NX NASTRAN Aeroelastic Analysis User's Guide".

[3] J. T. Batina, "A Finite-Difference Approximate-Factorization Algorithm for Solution of the Unsteady Transonic Small-Disturbance Equation," NASA-TP-3129, 1992.

[4] J. T. Batina, "Unsteady Transonic Flow Calculations for Interfering Lifting Surface Configurations," *Journal of Aircraft,* May 1986.

[5] Siemens, "Design Sensitivity and Optimization User's Guide".

[6] R. E. Bartels, B. K. Stanford and J. M. Waite, "Performance Enhancement of the Flexible Transonic Truss-Braced Wing Aircraft Using Variable-Camber Continuous Trailing-Edge Flaps," AIAA 2019-3160, 2019.

[7] R. e. al., "Development of Generalized Aeroservoelastic Reduced Order Models," in *SDM Conference,* , Palm Springs, California, 2009.