International Forum on Aeroelasticity and Structural Dynamics
IFASD 2024
17-21 June 2024, The Hague, The Netherlands

AEROELASTIC MISSION OPTIMIZATION OF VERY FLEXIBLE VEHICLES INCLUDING TAKE-OFF AND CLIMB CONSTRAINTS

Christopher A. Lupp[1]

[1]Air Force Research Laboratory
Aerospace Systems Directorate
Wright-Patterson AFB, Ohio, 45433, USA
christopher.lupp@us.af.mil

Keywords: aeroelasticity, conceptual aircraft design, multi-disciplinary design optimization, gradient-based optimization, time-domain

Abstract: An increase in high aspect-ratio aircraft designs in recent years has led to more flexible vehicles. Increased flexibility may, however, result in nonlinear aeroelastic behavior which must be captured during the design stage. Time-domain simulations, in particular, must be used to assess a flexible vehicle's stability across a mission. However, in the context of gradient-based vehicle and trajectory optimization, considering time-domain analyses for nonlinear aeroelastic systems has remained elusive due to the complexity of determining sensitivities of a time-marched solution. This paper develops a time-domain solution capability which uses a collocation method including gradient information for optimization problems. The aeroelastic time-domain solution is used to formulate take-off, climb, and landing constraints. The constraint formulations are demonstrated on a multi-disciplinary design optimization problem of a blended wing body. Finally, a fully transient formulation for the nonlinear aeroelastic system is presented and the ramification discussed.

NOMENCLATURE

D	drag force, N	M	Mach number
I_{sp}	specific impulse, s^{-1}	N_f	number of functions
L	lift force, N	T	thrust, N
T	thrust force, N	T_0	static thrust, N
g	gravitational acceleration, $\frac{m}{s^2}$	TET	turbine entry temperature
h	geodesic altitude, m	a_c	speed of sound at cruise, m/s
m	aircraft mass, kg	C_D	aircraft drag coefficient
r	horizontal distance, m	C_L	aircraft lift coefficient
v	airspeed, $\frac{m}{s}$	M_c	cruise Mach number
		P_B	body frame offset vector

P_w	local beam frame offset vector
p	current atmospheric pressure, Pa
p_0	atmospheric pressure at sea level, Pa
R	aircraft range, m
sfc	specific fuel consumption
t	current atmospheric temperature, K
t_0	atmospheric temperature at sea level, K
v_c	cruise speed, m/s
W_f	fuel weight, N
W_s	structural weight, N
w_x, w_y, w_z	local beam frame coordinate system vectors
x_B, y_B, z_B	body frame coordinates
x_G, y_G, z_G	global frame coordinates
α	angle of attack, $^\circ$
γ	flight path angle, $^\circ$
Π	overall pressure ratio of the engine
η_g	engine gas generator efficiency
η_i	engine inlet efficiency
η_d	efficiency of the inlet
η_f	efficiency of the fan
η_n	efficiency of the nozzle
η_t	efficiency of the turbine
κ	isentropic constant
μ	engine bypass ratio
ρ	current atmospheric density, $\frac{kg}{m^3}$
ρ_0	atmospheric density at sea level, $\frac{kg}{m^3}$
ζ_t	engine throttle setting
ζ_n	numerical relaxation factor
h	vector containing all nodal positions and rotations
h_w	vector containing the nodal positions and rotations
Q_1, Q_2	matrices of the state space equations
$\mathbb{K}(s)$	matrix exponential
s	coordinate along the beam axis
u	control inputs (in the state space equations)
v_B	translational body velocity
β	body velocities
ε^{el}	element strain vector
ε_x	extensional strain
κ_x	twist curvature
κ_y	bending curvature about w_y
κ_z	bending curvature about w_z
ω_B	rotational body velocity
θ_B	orientation of the body frame
C^{BG}	rotation matrix from the body to the global frame

1 INTRODUCTION

The design of very flexible aircraft poses challenges, some of which arise from transient, nonlinear aeroelastic effects. These are not limited to flutter and post-flutter behavior, but may include

handling qualities, response to gust disturbances, etc. As very flexible aircraft may encounter a wide range of displacements over their operational envelope, the stability of the vehicle may differ greatly depending on structural displacements as well as transients.

While the importance of flutter on vehicle design has been investigated since the 1970's, a wealth of studies regarding the dynamic behavior of very flexible aircraft were only obtained after the development of the NASA/Aerovironment Helios prototype. Helios encountered large bending (and a large range of) deflections during several test flights. However, during a 2003 test flight, the vehicle encountered unusually large bending displacements due to atmospheric disturbances, became unstable, and disintegrated after exceeding the aircraft's design speed. The loss of Helios proved to be a watershed moment for time-domain, nonlinear aeroelasticity. After the incident, NASA tasked a Mishap Investigation Board (MIB) with determining factors that contributed and led to the loss of the vehicle. As a part of its findings [1], the MIB recommended that NASA "[d]evelop more advanced, multidisciplinary (structures, aeroelastic, aerodynamics, atmospheric, materials, propulsion, controls, etc) ***'time-domain'*** analysis methods appropriate to highly flexible, morphing vehicles."

Prior to the Helios accident, aeroelastic investigations centered around frequency-domain studies (e.g., [2, 3]). However, the MIB findings resulted in research efforts regarding time-domain, geometrically nonlinear aeroelasticity. Work by Su and Cesnik [4, 5], Shearer and Cesnik [6], extended the nonlinear aeroelastic framework, UM/NAST, into the time domain. Patil and Hodges [7] extended their NATASHA framework for transient nonlinear aeroelasticity and investigated the dynamic behavior of a Helios-like vehicle. Similarly, Wang et al. [8] studied the transient behavior of a High Altitude Long Endurance (HALE) aircraft including unsteady aerodynamics.

In particular, Cesnik and coworkers focused on very flexible HALE aircraft and the coupled nature of geometrical nonlinear aeroelasticity and rigid-body degrees of freedom. While this work on transient nonlinear aeroelasticity resulted in the design of the University of Michigan's X-HALE aircraft [9], the design was not obtained from an optimization process (gradient-based or -free). Optimization problems including geometrically nonlinear flutter constraints have been conducted by Variyar [10] and by Lupp and Cesnik [11], this work was conducted in the frequency domain and cannot be utilized for a time-domain trajectory optimization.

Meanwhile, transient trajectory optimization and simultaneous vehicle Multi-disciplinary Design Optimization (MDO) have become a recent focus of research. A tool for conducting using collocation for transient optimization problems, Dymos [12], was developed at NASA within the context of the OpenMDAO framework, featuring methods for computationally efficient evaluation of derivatives [13]. Dymos has since been used for trajectory optimization of urban air mobility vehicles by Falck and coworkers [14] as well as Hendricks and coworkers [15]. Jasa, Mader, and Martins [16] conducted trajectory optimizations of a supersonic aircraft, subject to thermal constraints. Hendricks et al. also conducted a trajectory optimization, optimizing the vehicle's propulsion system [17]. Lin, Carpenter, and de Weck [18] investigated a simultaneous trajectory and vehicle design of a sounding rocket. Lupp et al. [19] used Dymos to include transient flight mechanics as well as thermal and power subsystem constraints for the design of a HALE aircraft.

However, trajectory optimization including geometrically nonlinear aeroelastic analyses (in the time-domain) have not been applied to gradient-based optimization. Holden and Kroo [20] investigated optimization problems enforcing aeroelastic stability using a beam structural model

and a vortex lattice method (VLM) aerodynamic solution. While they used a collocation method in an optimization problem to solve nonlinear aeroelastic equations of motion (EOM) in the time domain, the dynamic constraint was used to enforce flutter stability and did not include rigid body degrees of freedom. More recently, del Carre and Palacios [21] investigated trajectory optimization including geometrical nonlinear aeroelastic effects and flight mechanics. They modeled a HALE aircraft, subject to large deformations, and optimized it's catapult launch and take-off trajectory, but did not include vehicle design variables (beyond control surface deflections). However, del Carre and Palacios used a Bayesian optimization process. As a gradient-free method, this approach will likely scale poorly with increasing numbers of design variables typical of large MDO problems.

This paper presents a methodology for conducting gradient-based trajectory optimizations of very flexible vehicles, which may encounter geometrical nonlinearities. Structural modeling is accounted for by a strain-based, geometrically exact beam formulation. Within the context of a new nonlinear aeroelastic framework, Perseids, a theoretical formulation is presented for conducting collocation-based time-domain simulations including rigid-body degrees of freedom. The process is fully differentiated and conducive to large numbers of design variables.

The transient simulation is applied to both take-off and climb segments for a multi-disciplinary design optimization of a Blended Wing Body (BWB) aircraft. The methodology used is applicable to performance assessments of aircraft, but not constraining of dynamic aeroelastic instabilities. Finally, a fully transient, nonlinear aeroelastic formulation (including rigid body degrees of freedom) that provides gradients for optimization is presented. Ramifications of the formulation on the solution and optimization process are discussed.

2 NUMERICAL METHODS

Multi-disciplinary Design Optimization (MDO) has become a wide-spread tool used during vehicle conceptual design. Gradient-based optimization, in particular, enables designs incorporating very large numbers of design variables. However, these methods require robust and efficient access to derivative information. This section describes the MDO theory, methods used to determine derivatives, and time integration techniques applicable to MDO problems used within this work.

2.1 Multi-Disciplinary Design Optimization

Multi-disciplinary Design Optimization (MDO) is a subset of the larger optimization field and encompasses optimization problems that feature multiple, distinct disciplines that interact with each other. Multi-disciplinary problems are common in aerospace engineering, notably including aeroelasticity, as well as overall aircraft design.

Researchers have formulated many different MDO architectures that vary in complexity and degree of discipline-coupling. Some organizations favor architectures that feature loosely-coupled MDO formulations, due to the "siloed" nature of their corporate structure [22]. On the other hand, tighter coupling, which may enable greater performance gains, is often found in academic and research-centric studies. Within this work, a tightly coupled approach was chosen, using the modular analysis and unified derivatives (MAUD) MDO architecture [23] implemented within the OpenMDAO framework [24].

More disciplines that are tightly coupled often results in larger numbers of design variables. As a result, many traditional methods for design space exploration, such as Design of Experiments

(DOE) or gradient-free optimization methods (e.g., genetic algorithms) become intractable due to the *curse of dimensionality*.

Gradient-based algorithms, by contrast, are able to negotiate these larger design spaces efficiently. Fundamentally, these methods implement Newton-Raphson or Newton-Raphson-like root-finding algorithms and apply them to the derivatives of the objective function. Unlike gradient-free methods, which use heuristics to determine convergence, the convergence criteria for gradient-based algorithms is:

$$\frac{\partial f}{\partial x} \stackrel{!}{=} 0, \tag{1}$$

or

$$\left|\frac{\partial f}{\partial x}\right| \leq \epsilon, \tag{2}$$

where f is the objective function, x is the vector of design variables, and ϵ is a user-defined numerical tolerance.

2.2 Automatic Differentiation

Automatic Differentiation (AD), also referred to as algorithmic differentiation, is a method of obtaining gradients that has become increasingly popular in recent years. Generally, AD describes tools that are able to determine gradients by decomposing the individual operations in computer programs.

While AD can be implemented in a variety of forms, it fundamentally decomposes the software into a series of operations (e.g., lines of code) V_i, to which the derivatives are known. Additionally, local variables are stored in an array v_i. Every line of code is differentiated and the total derivatives of the functions with respect to the design variables of interest are determined either via forward- or back-substitution. Similar to semi-analytical methods [25, 26], there are two methods for determining the total derivatives using AD. The forward mode is given by:

$$\left(\mathcal{I} - D_V\right) D_v = \mathcal{I}, \tag{3}$$

or the reverse mode is given by:

$$\left(\mathcal{I} - D_V\right)^T D_v^T = \mathcal{I}. \tag{4}$$

The matrices in Equations 3 and 4 are:

$$D_V = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{\partial V_2}{\partial v_1} & 0 & 0 & \cdots & 0 \\ \frac{\partial V_3}{\partial v_1} & \frac{\partial V_3}{\partial v_2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ \frac{\partial V_n}{\partial v_1} & \frac{\partial V_n}{\partial v_2} & \cdots & \frac{\partial V_n}{\partial v_{n-1}} & 0 \end{bmatrix} \tag{5}$$

$$D_v = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \frac{dv_2}{dv_1} & 0 & 0 & \cdots & 0 \\ \frac{dv_3}{dv_1} & \frac{dv_3}{dv_2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ \frac{dv_n}{dv_1} & \frac{dv_n}{dv_2} & \cdots & \frac{dv_n}{dv_{n-1}} & 0 \end{bmatrix} \tag{6}$$

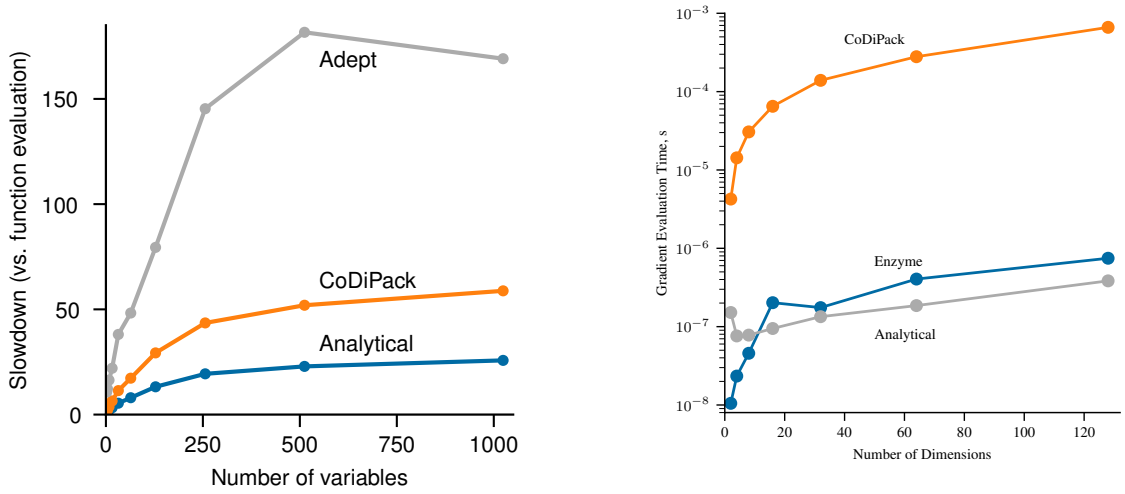


Figure 1: Performance and scalability benchmark comparing Enzyme, CoDiPack, and the analytical gradient solution. The results were obtained by averaging 100 benchmark runs.

The forward mode solution is akin to solving for one column of D_V from Equation 3 using forward substitution. These operations are executed together with the original code. The reverse mode solves the derivatives using back substitution. This requires storing the individual operations and variables in what is typically called a tape. As with the analytical methods, the forward and reverse modes are applicable for different problems. The forward mode is typically more efficient when the number of functions, N_f, is greater than the number of variables of interest, N_x, while the reverse mode is faster for $N_x > N_f$.

If implemented correctly, AD derivatives are exact to machine precision, as all individual operations are differentiated analytically. However, the implementation using AD can be challenging as potentially significant changes to the software's build system and source code are required. Two main approaches exist for applying AD: source code transformation tools (such as Tapenade [27]) and operator overloading tools (such as CoDiPack [28]). However, both require significant modifications to existing code to function and, especially in the case of operator overloading, may result in a substantial performance penalty compared to the primal function evaluation.

Recently, another class of AD tool has emerged: compiler-based AD. Moses and Churavy [29] introduced a method for efficiently determining derivatives of source code at compile time using the LLVM compiler [30]. To use this method, the source code must be lightly modified (using annotations, such as pragma statements in C/C++). LLVM then compiles the source into LLVM IR (intermediate representation) and performs a code optimization step. After this initial compiler optimization, the source code is differentiated with the Enzyme AD compiler plugin, generating the requested gradient functions. Next, another compiler optimization step is conducted to optimize the generated gradients, before generating the final binary files (e.g., libraries or executables). This results in very efficient gradients compared to operator overloading tools, as many operator overloading libraries prevent compiler optimization of the primal function evaluation as well as the gradient.

A performance comparison of various AD tools was conducted using the n-dimensional Rosenbrock function [31] (Figure 1). While CoDiPack outperforms other operator overloading li-

braries (Adept), it incurs a significant performance penalty compared to the evaluation of the analytical gradient. Enzyme AD, by comparison, performs comparably to the analytical gradient evaluation, even outperforming it for small numbers of variables[1]. While not benchmarked in this work, Moses and Churavy [29] observed performance improvements using Enzyme AD compared to Tapenade. Due to the performance advantages, Perseids uses a combination of CoDiPack and Enzyme to obtain gradients.

2.3 Assembling Total Derivatives

Another method for determining coupled derivatives is using the unified derivative equations (UDE) [23]. Assuming functions in residual form, the UDE in forward mode are:

$$[\mathcal{I}] = \left[\frac{\partial \mathcal{R}}{\partial v}\right] \left[\frac{dv}{dr}\right], \tag{7}$$

or reverse (adjoint) mode:

$$[\mathcal{I}] = \left[\frac{\partial \mathcal{R}}{\partial v}\right]^T \left[\frac{dv}{dr}\right]^T, \tag{8}$$

where v is the vector of all outputs of the disciplines in the MDO problem and r is the vector of all residual values. Equations 7 and 8 form the basis for OpenMDAO [24] to determine the total derivatives used by gradient-based optimizers. Note, to determine the problem's total derivatives, only the partials of the individual disciplines are required and no derivatives of the coupling terms must be provided. This eases the implementation of coupled analyses, as only the derivatives of the analyses are required, instead of also having to provide derivatives of the coupling (as is the case in the coupled adjoint approach).

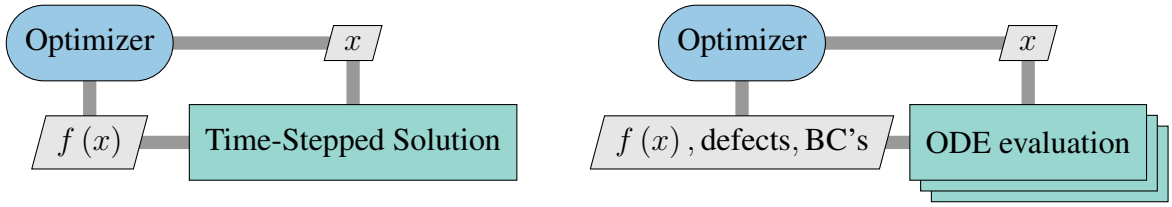
Similar to the coupled adjoint approach, every design variable requires one solution of Equation 7 (forward mode). Conversely, Equation 8 (reverse mode) must be solved once for every objective or constraint, making the choice of solution mode depend on the problem definition.

2.4 Time Integration Methods within MDO

The trajectory optimization problems investigated in this work require a transient solution of the nonlinear aeroelastic EOM. One approach to solving the equations of motion in the time-domain is to use an integration method (e.g., the trapezoidal or generalized-α integration schemes [32] used in University of Michigan's Nonlinear Aeroelastic Simulation Toolbox (UM/NAST)). For explicit time-integration schemes, the process is inherently sequential (Figure 2a. Furthermore, within the context of optimizations, determining derivatives for a time-marched solution can be computationally expensive, as the perturbation methods (such as finite-differencing or complex step) may be computationally prohibitive and other methods such as AD would require a very large tape to record a transient solution, resulting in large amounts of system memory needed and long derivative evaluation times.

Another approach, used in this work, is the use of colocation or pseudo-spectral methods such as Legendre-Gauss-Lobatto (LGL) [33] or Radau [34] methods. While a time-stepped solution (such as Euler time-integration) is sequential by nature, the Ordinary Differential Equations (ODE)s in the collocation method can be evaluated in parallel (Figure 2b). By contrast,

[1]Note: Enzyme AD is able to outperform user-written gradient functions in some cases due to its use of two compiler optimization cycles. As a result the function being differentiated has already been optimized once prior to generating gradients, which, in turn, are optimized. Hand-written gradient functions may not utilize code optimizations of the initial function.



(a) Time-stepped solution

(b) Collocation methods

Figure 2: Conceptual comparison of a time-stepped (sequential) time-domain simulation and a colocation method used in an optimization problem.

the collocation methods have been used successfully for transient, gradient-based optimization problems [12, 14].

To leverage OpenMDAO, this work uses the Dymos framework [12, 14]. As most available aeroelastic tools use explicit time-integrated solutions, this work first derives the formulation (Section 3) needed by the collocation-based transient solution.

3 GEOMETRICALLY NONLINEAR AEROELASTIC FORMULATION

The underlying formulation used in the strain-based, nonlinear beam solver was initially derived at the Active Aeroelasticity and Structures Research Laboratory (A^2SRL) at the University of Michigan by several generations of graduate students advised by and collaborating with Prof. Carlos Cesnik (Cesnik and Brown [35], Shearer and Cesnik [36], Su and Cesnik [5], among others). This section describes the theoretical basis of the Perseids aeroelastic framework for nonlinear, coupled, static analyses.

3.1 Coordinate Systems and Conventions

Three main coordinate systems exist for the description of the geometrically nonlinear beam solution subject to rigid body motion: the global (G), the body (B), and local node (w) frames (Figure 3). A vector, P_B, describes the offset of the body from the global frame and the orientation with respect to the global frame is defined by the quaternion vector, ζ:

$$P_B = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix}, \qquad \zeta = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}. \tag{9}$$

Rigid body motion of the body frame (relative to the global frame) is captured by three linear and three angular velocities:

$$b = \begin{bmatrix} p_B \\ \theta_B \end{bmatrix}, \qquad \dot{b} = \beta = \begin{bmatrix} v_B \\ \omega_B \end{bmatrix}. \tag{10}$$

Each structural member contains a user-defined number of beam elements. A geometrically nonlinear, constant strain, three-noded beam element is used. The strain states for every beam

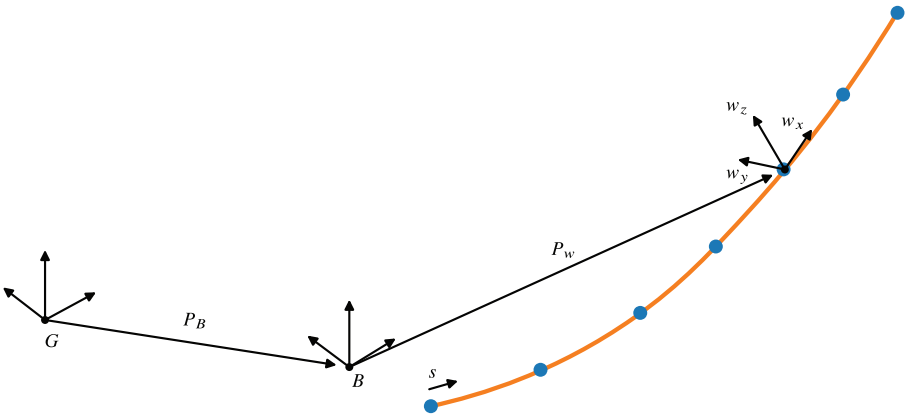


Figure 3: Global, body, and nodal/local coordinate system definitions used in the Perseids aeroelastic toolbox and UM/NAST theoretical formulation.

element are: extensional, twist, and two bending curvatures (in-plane and out-of-plane):

$$\varepsilon^{el} = \begin{bmatrix} \varepsilon_x \\ \kappa_x \\ \kappa_y \\ \kappa_z \end{bmatrix}. \tag{11}$$

The w frame, located at every node, describes the beam nodes' location and orientation in relation to origin of the body frame (Figure 3). Analogous to P_B, the vector P_w describes the offset of the w frame from the body frame. A column vector h can be defined that contains a point's spatial position and orientation information. The vector P_w and the coordinate system unit vectors (described in the body frame) w_x, w_y, and w_z, which are stacked to obtain the vector h_w:

$$h_w\left(s\right) = \begin{bmatrix} P_w(s) \\ w_x(s) \\ w_y(s) \\ w_z(s) \end{bmatrix}. \tag{12}$$

For the spatial information of the same point in the global frame, the corresponding h vector is defined as:

$$h\left(s\right) = \begin{bmatrix} P_b + P_w(s) \\ w_x(s) \\ w_y(s) \\ w_z(s) \end{bmatrix}. \tag{13}$$

The rotational transformation from the body to the global frame can be expressed via the quaternion vector (Eq. 9):

$$C^{BG} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{14}$$

Finally, the kinematic relationship between the strain (state variable), the boundary node h_{BC}, and the beam displacements (represented in the vector h) is captured by the matrix exponential:

$$h_w(s) = e^{\mathbb{K}(s-s_0)} h_{BC} \tag{15}$$

with

$$\mathbb{K}(s) = \begin{bmatrix} 0 & 1+\varepsilon_x & 0 & 0 \\ 0 & 0 & \kappa_z & -\kappa_y \\ 0 & -\kappa_z & 0 & \kappa_x \\ 0 & \kappa_y & -\kappa_x & 0 \end{bmatrix}_{12\times 12}. \tag{16}$$

For further details on the strain-based formulation, the reader is encouraged to review the comprehensive description provided by Su and Cesnik [4, 5].

3.2 Static Aeroelastic Solution

The static governing equations of the static aeroelastic problem were introduced by Cesnik and Brown [35], Shearer and Cesnik [36], and Su and Cesnik [5]:

$$\underbrace{\begin{bmatrix} K_F & K_c^T \\ K_c & 0 \end{bmatrix}}_{K_k} \underbrace{\begin{bmatrix} \varepsilon \\ \lambda_c \end{bmatrix}}_{y_{s,k}} = \underbrace{\begin{bmatrix} f \\ f_c \end{bmatrix}}_{f_k}. \tag{17}$$

The lower row of the equations contains the absolute and relative constraints for the system, which are enforced using Lagrange multipliers, λ_c. Previously, Equation 17 has been solved using a recursive, load-stepped approach using a constant relaxation factor, ζ_n:

$$y_{s,k+1} = \zeta_n y_{s,k} + (1 - \zeta_n) K_k^{-1} f_k. \tag{18}$$

Note that the inverse of K_k is not computed, but rather the entire $K_k^{-1} f_k$ term is determined by solving the linear system from Equation 17. This solution approach has been applied successfully in the UM/NAST for static aeroelastic studies.

Another solution approach is to formulate the static problem in residual form:

$$\mathcal{R}_s = f_k - K_k y_s. \tag{19}$$

While this may appear to be a linear system, the dependency of the stiffness matrix, K, on the static states, y_s, results in nonlinear residual equations. To avoid convergence issues when applying a root-finding algorithm, a relaxation factor, ζ_n, is included to obtain the residual for the k-th iteration:

$$\mathcal{R}_{s,k} = \zeta_n\left(k\right) f_k - K_k y_{s,k}. \tag{20}$$

Solving the nonlinear static problem within an MDO context therefore becomes an implicit process, with the states y_s as an input. The states can be determined using a nonlinear Newton solver combined with a Krylov method (GMRES [37, 38]) for linear solutions (determining gradients). To improve convergence speed and stability, Perseids employs an Armijo-Goldstein linesearch [39]. The implicit formulation offers the advantage of using Newton's method (which converges quadratically) and typically yields tighter convergence than the pseudo-time-stepping approach.

4 DESIGN STUDIES

A gradient-based MDO problem is formulated to demonstrate the transient trajectory optimization including nonlinear aeroelasticity. This section describes the models used and the assumptions made for each phase of flight. Finally, a gradient-based optimization is conducted to demonstrate the theoretical formulations in this work.

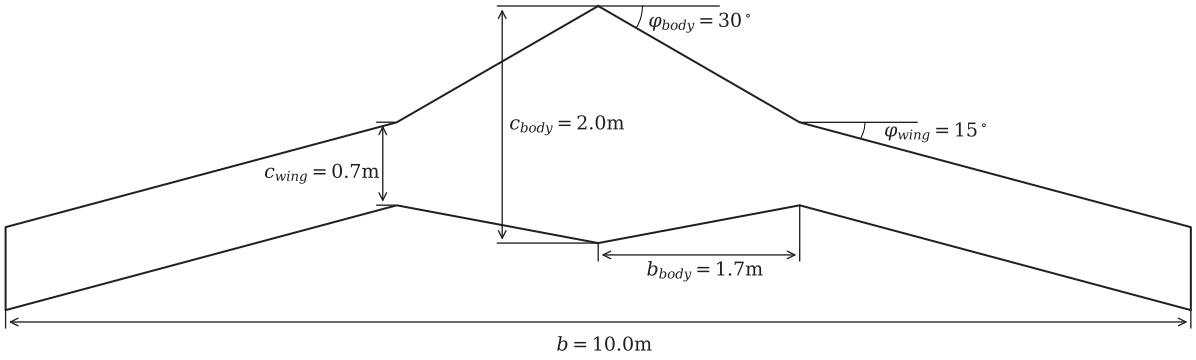
4.1 Blended Wing Body Model

A notional BWB configuration is used within this work to explore the application of the theoretical formulation on a sample MDO problem. The BWB model allows the parametric modification of the aerodynamic planform, structural geometry, and engine. This section describes the assumptions and parameterization of the aircraft model.

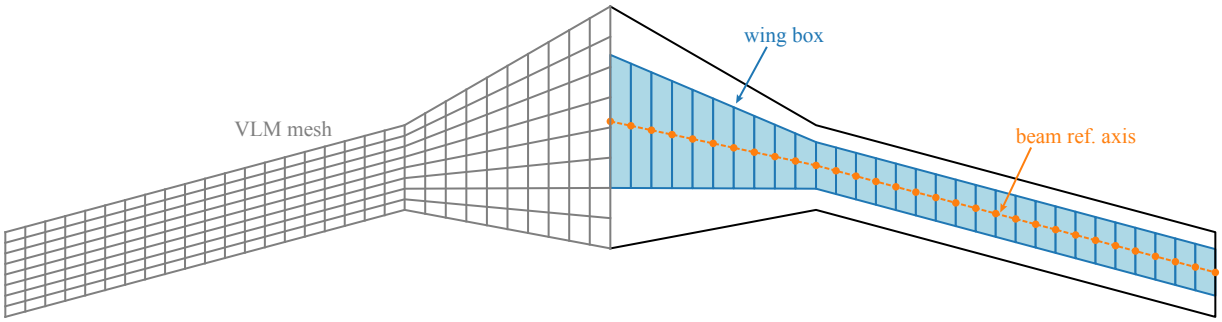
4.1.1 Geometry

The BWB consists of two wing sections: the body and wing sections (Figure 4a). The body is a trapezoidal configuration, while the wing section is defined by a swept planform of constant chord. The entire planform can be uniquely defined using six geometric design variables: the body span (b_{body}), body chord (c_{body}), body sweep angle (φ_{body}), total wing span (b), wing chord (c_{wing}), and wing sweep angle (φ_{wing}). In addition to the planform variables, five spline control points are used to define both the planform twist and thickness distributions.

While these parameters are used to determine the wing planform, no analyses are conducted directly using it. Instead, aerodynamic (VLM) and structural models (wing box definition, shell and/or beam model) are generated from the design variables. While a shell model has been created for the BWB, the structural models used in this work utilize equivalent beam properties and are automatically generated after every design variable update (Figure 4b). The meshing process provides analytical gradients, therefore making this approach feasible for gradient-based optimization.



(a) Wing planform



(b) Analysis geometry

Figure 4: Planform and analysis geometry of the baseline BWB model.

4.1.2 Weights

While the BWB model supports detailed weight analyses based on structural and component sizing, Raymer's weight estimations for fighter aircraft [40] are used within this work. Raymer's equations were determined from statistical regressions of existing designs. As such, it reasonably predicts the weight of conventional configurations, but will have shortcomings for underrepresented design configurations. It is noteworthy, that when these equations were developed, BWB configurations were not common. Despite the possible errors associated with using this method for this work, it provides a simple model to account for weight and enable design tradeoffs.

Another rationale for choosing this weight model for this work was the implicit nature of Raymer's method. This arises from the coupled nature of the weight of individual components and the design gross weight of the vehicle. The design gross weight is the sum of all components:

$$W_{dg} = \sum W_i. \tag{21}$$

However, the wing weight of the BWB is not determined from the Raymer weight equations. Instead, the structural wing weight is approximated from the wing box dimensions and thicknesses. This approach adds fidelity to the weight approximation, while still utilizing the statistical data that anchors Raymer's method.

4.1.3 Aerodynamics

A VLM solver, Open AeroStruct [41], serves as the aerodynamic solver for this work. Open AeroStruct was created at the University of Michigan and uses OpenMDAO to set up and solve the VLM governing equations. OpenAeroStruct offers mesh manipulation utilities, however, these are not used within this work, instead employing custom tools to remesh the planform for every design iteration.

It should also be noted that while OpenAeroStruct is useful for simple studies and not representative of industry tools, within this work it serves as a surrogate and could be replaced with higher fidelity tools. For the purposes of this study, OpenAeroStruct will be run from an external data center, emulating the aerodynamic discipline being evaluated at a different organization. OpenAeroStruct's simplicity is therefore beneficial to limit the computational costs of this study.

4.1.4 Engine

The engine model and its fuel consumption is a key factor in a fuel burn minimization problem. For conceptual design studies, engines are often pre-selected and data provided. In other cases, simple surrogate models may be provided by the engine manufacturers that allow for some variation in engine sizing parameters without exposing the (proprietary) engine model. The BWB model utilizes a parametric engine model proposed by Torenbeek [42] to enable tradeoffs between the vehicle and engine designs.

In Torenbeek's model, the thrust to static thrust ratio is a function of environmental quantities such as the density ratio and Mach number, and the engine throttle setting (ζ_t), as well as the engine bypass ratio (μ):

$$\frac{T}{T_0} = \zeta_t \left(\frac{\rho}{\rho_0} \right) e^{-0.35M\left(\frac{\rho}{\rho_0}\right)\sqrt{\mu}}. \tag{22}$$

The specific fuel consumption (sfc) is provided by the following equation:

$$sfc = \frac{0.697\sqrt{\frac{t}{t_0}}\left(\phi - \vartheta - \frac{\chi}{\eta_c}\right)}{\sqrt{5\eta_n\left(1+\eta_f\eta_t\mu\right)\left(G + 0.2M^2\mu\frac{\eta_d}{\eta_f\eta_t}\right)} - M\left(1+\mu\right)}, \tag{23}$$

with:

$$\chi = \vartheta\left(\Pi^{\frac{\kappa-1}{\kappa}} - 1\right); \tag{24}$$

$$\vartheta = 1 + \frac{\kappa - 1}{2}M^2; \tag{25}$$

the gas generator function,

$$G = \left(\phi - \frac{\chi}{\eta_c}\right)\left(1 - \frac{1.01}{\left(\eta_i^{\frac{\kappa-1}{\kappa}}\left(\chi+\vartheta\right)\right)\left(1 - \frac{\chi}{\phi\eta_c\eta_t}\right)}\right); \tag{26}$$

and the normalized turbine entry temperature,

$$\phi = \frac{TET}{t}. \tag{27}$$

For this study, the fan efficiency, η_f, is set at 0.85, the compressor efficiency, η_c, at 0.84, the turbine efficiency, η_t, at 0.87, and the nozzle efficiency, η_n, at 0.98. The gas generator efficiency

$$\eta_g = 1 - \frac{0.7M^2\left(1-\eta_d\right)}{1 + 0.2M^2} \tag{28}$$

and inlet efficiency:

$$\eta_i = 1 - \left(1.3 + 0.25\mu\right)\frac{\Delta p}{p} \tag{29}$$

are determined parametrically. For this work, the pressure loss, $\frac{\Delta p}{p}$, at the inlet is assumed to be two percent.

4.2 Mission Description and Modeling

The total aircraft mission is subdivided into multiple mission phases: take-off, climb, and cruise. Each of these phases is modeled differently. While the take-off and climb segments are transient and modeled in Dymos, the cruise segment is approximated using Breguet's range equation. This section details the underlying assumptions and equations of motion for the individual flight phases.

4.2.1 Take-Off

The take-off phase follows the balanced field length approach by Raymer [40] and is itself divided into distinct segments: roll, rotation, initial climb, and rejected take-off. During the roll phase, the ODE is expressed as:

$$F_r = mg - L\cos\alpha - T\sin\alpha, \tag{30}$$

$$\dot{v} = \frac{T\cos\alpha - D - F_r\mu_r}{m}, \tag{31}$$

$$\dot{r} = v. \tag{32}$$

The roll phase extends until v_1 (in accordance to certification requirements), before the aircraft transitions to the rotate and then the climb segment. The ODE for the climb segment adds the altitude and flight path angle as states [43]:

$$\dot{v} = \frac{T}{m}\cos\alpha - \frac{D}{m} - g\sin\gamma, \tag{33}$$

$$\dot{\gamma} = \frac{T}{mv}\sin\alpha + \frac{L}{mv} - \frac{g\cos\gamma}{v}, \tag{34}$$

$$\dot{h} = v\sin\gamma, \tag{35}$$

$$\dot{r} = v\cos\gamma. \tag{36}$$

The angle of attack, α, and the thrust, T, serve as control variables and are driven by the optimizer. The runway length along with the altitude at the end of the initial climb are constrained in this study. The required runway length constraint equates to the class of runway the aircraft is designed for and is implemented as an inequality constraint:

$$r_{end} \leq r_{max}. \tag{37}$$

Likewise, the altitude is constrained as an inequality:

$$h_{end} \leq h_{min} = 35\text{ft}, \tag{38}$$

and represents the certification requirement that the aircraft be able to climb to 35 ft by the end of the runway. Additional constraints are enforced on the vehicle aerodynamics during climb to ensure stall free flight[2]:

$$C_L(t) \leq 1.0 \tag{39}$$

Nonlinear aeroelasticity is accounted within this ODE by obtaining the lift and drag values from the static aeroelastic solution in Perseids (Figure 5). To this end, the atmospheric, aerodynamic, and structural properties, as well as states (speed) are passed to the static aeroelastic solution. The resulting lift and drag are then transferred to the flight mechanics equations. The resulting method takes aeroelastic performance into account. Furthermore, (steady) structural stress during take-off and climb can be constrained using the aeroelastic solution (though not exercised in this work).

[2]The vehicle aerodynamics use potential flow. However, the stall constraint prevents the optimizer selecting angles of attack that would result in a stall under realistic conditions.

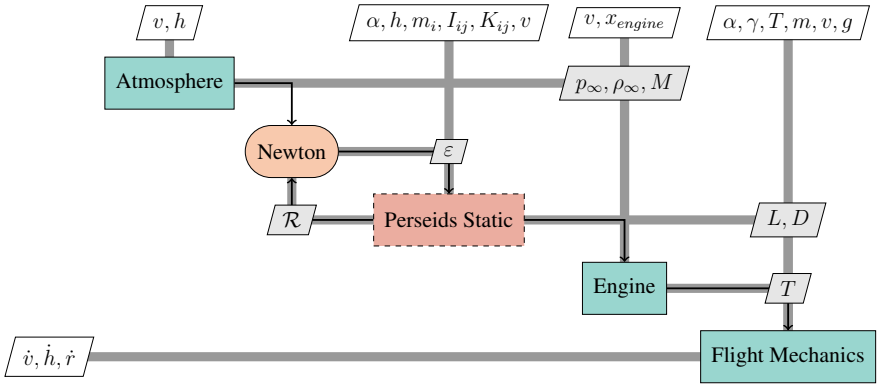


Figure 5: XDSM diagram of the ODE including static, geometrically nonlinear aeroelastic effects.

4.2.2 Climb

The climb phase of the flight is modeled according to the ODE in Equations 33–36. Again, for this phase, the angle of attack and the thrust are used as control variables by the optimizer. The climb phase begins at $200\,\mathrm{m}$ and ends at cruise altitude ($6\,\mathrm{km}$). Unlike the initial climb during take-off, a constraint on the flight path angle is enforced upon reaching cruise altitude, so that the aircraft stops climbing ($\gamma \overset{!}{=} 0$).

4.2.3 Cruise

The cruise phase is driven by a required aircraft range, with the vehicle fuel burn being the quantity of interest (and the objective of the optimization). The fuel burn is determined from the aircraft range. The Breguet range equation[3] can be written as [45–47]:

$$R = \frac{v_c\, C_L}{C_D\, sfc} \ln\left(\frac{W_1}{W_2}\right). \tag{40}$$

$$R = \frac{v_c\, C_L}{C_D\, sfc} \ln\left(\frac{m_1}{m_2}\right) \tag{41}$$

$$= \frac{M_c\, a_c\, C_L}{C_D\, sfc} \ln\left(\frac{m_1}{m_2}\right) \tag{42}$$

The initial mass, m_1, is the combined weight of the structure and fuel, while the weight at the end of the trip, m_2, is just the structural weight (no fuel reserves considered), i.e.,

$$m_1 = m_s + m_f \tag{43}$$

$$m_2 = m_s \tag{44}$$

The fuel weight and fuel mass are thereby determined by rearranging the range equation:

$$m_f = m_s \left(e^{R\frac{C_D\, sfc}{C_L\, v_c}} - 1\right) \tag{45}$$

The fuel mass (fuel burn) serves as the objective for this design study.

[3] While the range equation derives its name from Louis Breguet, as Cavcar [44] notes, the origin of the equation can be traced to multiple sources.

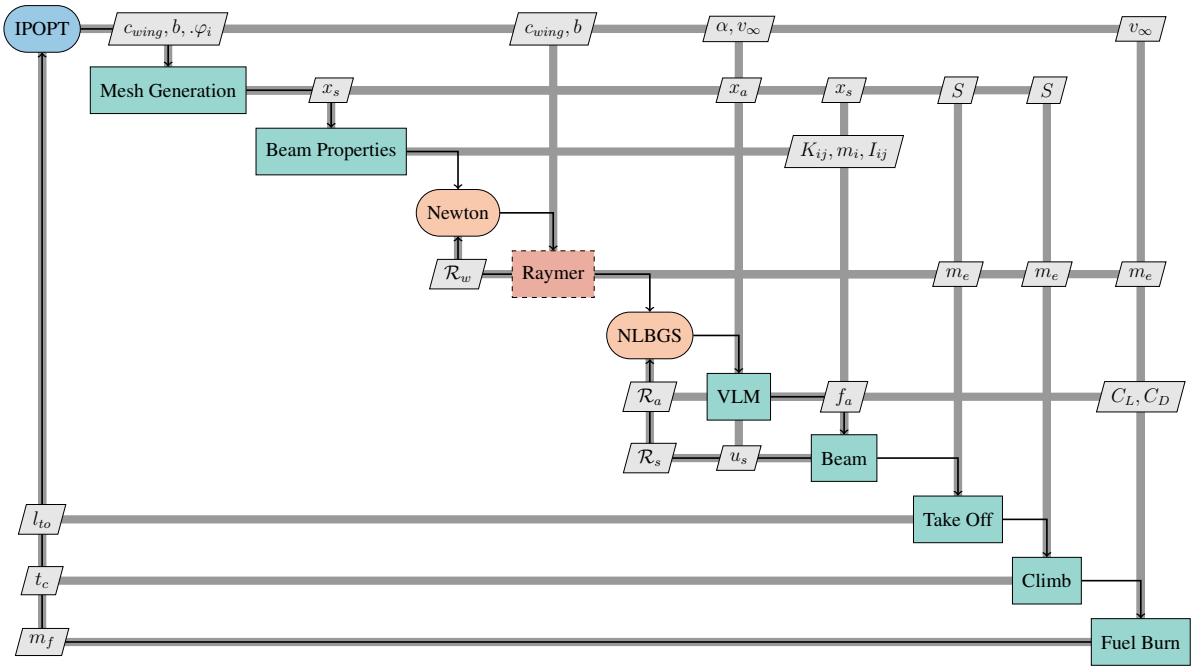


Figure 6: XDSM diagram of the BWB optimization subject to take-off and climb constraints.

4.3 Gradient-Based Results

A gradient-based MDO problem is formulated to demonstrate the transient trajectory optimization including nonlinear aeroelasticity. The design problem is formulated as a fuel burn minimization problem consisting of take-off, climb, and cruise phases (Figure 6). The optimization formulation is given by:

$$
\begin{aligned}
&\textbf{minimize:} && m_f, \\
&\textbf{with respect to:} \\
& && x = [b_{wing}, c_{wing}, \alpha_b, \varphi_i, t_i]^T, \\
&\textbf{subject to:} \\
& && C_{L,cruise} = 0.5, \\
& && V_{fuel} \leq V_{wingbox}, \\
& && |\sigma_{Mises}| \leq \sigma_{Al}\ , \\
& && r_{to} \leq 1500\text{m}, \\
& && h_{to} \leq 35\text{ft}, \\
& && t_{climb} \leq 400\text{s}.
\end{aligned}
$$

The design variables for the problem are the wing span, wing chord, body angle of attack, wing twist, and wing box thicknesses. The cruise lift coefficient, fuel volume, cruise stress, take-off length, altitude (at the end of the runway), and climb time are constrained. The IPOPT optimizer [48] is used to drive the problem due to its ability to navigate large numbers of design variables and constraints (as imposed by the collocation methods used by Dymos).

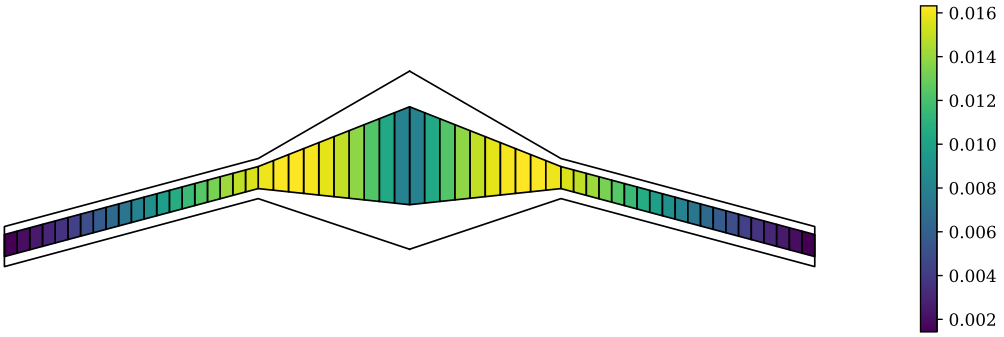


Figure 7: Thickness distribution of the optimal BWB wingbox (in meters).

The optimization converged in 187 iterations resulting in the vehicle depicted in Figure 7. While the optimization converged, it should be noted that neither the take-off, nor the climb constraints were active. Again, it should be noted, that this optimization accounts for nonlinear aeroelasticity from a performance perspective.

5 FULLY TRANSIENT TRAJECTORY OPTIMIZATION FORMULATION

The trajectory optimization studies presented within this work assumed static aeroelastic responses. As a result, there is no structural coupling between time steps. This approximation holds true for quasi-steady problems. However, as mentioned in the introduction to this work, fully transient aeroelastic solutions with tight coupling of rigid-body degrees of freedom may be necessary for very flexible aircraft. This section present the theoretical formulation needed to transition from static to fully unsteady aeroelastic trajectory optimization.

The coupled aeroelastic equations of motion that govern Perseids were originally developed by Shearer [32] and Su [5] for use in UM/NAST:

$$\begin{bmatrix} M_{FF} & M_{FB} \\ M_{BF} & M_{BB} \end{bmatrix} \begin{bmatrix} \ddot{\varepsilon} \\ \dot{\beta} \end{bmatrix} + \begin{bmatrix} C_{FF} & C_{FB} \\ C_{BF} & C_{BB} \end{bmatrix} \begin{bmatrix} \dot{\varepsilon} \\ \beta \end{bmatrix} + \begin{bmatrix} K_{FF} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon \\ b \end{bmatrix} = \begin{bmatrix} f_F \\ f_B \end{bmatrix}, \tag{46}$$

$$-\frac{1}{2}\Omega_\zeta \zeta = \dot{\zeta}, \tag{47}$$

$$\begin{bmatrix} C^{GB} \\ 0 \end{bmatrix} \beta = \dot{P}_B. \tag{48}$$

In addition to strain states, the equations of motion include nonlinear rigid body degrees of freedom. While not necessary for all analyses, the EOM (Equations 46–48) can be augmented with unsteady aerodynamics using Peters' finite state aerodynamics [49]:

$$F_1 \begin{bmatrix} \ddot{\varepsilon} \\ \dot{\beta} \end{bmatrix} + F_2 \begin{bmatrix} \dot{\varepsilon} \\ \beta \end{bmatrix} \beta + F_3 \lambda = \dot{\lambda}. \tag{49}$$

To solve the problem with Dymos the EOM must be reformulated into state-space form. The governing equations (Equations 46–49) were rearranged by Brown [2] to obtain a set of first order ODE:

$$Q_1 \dot{y} = Q_2 y + f\left(y, \dot{y}, u, v_g\right) \tag{50}$$

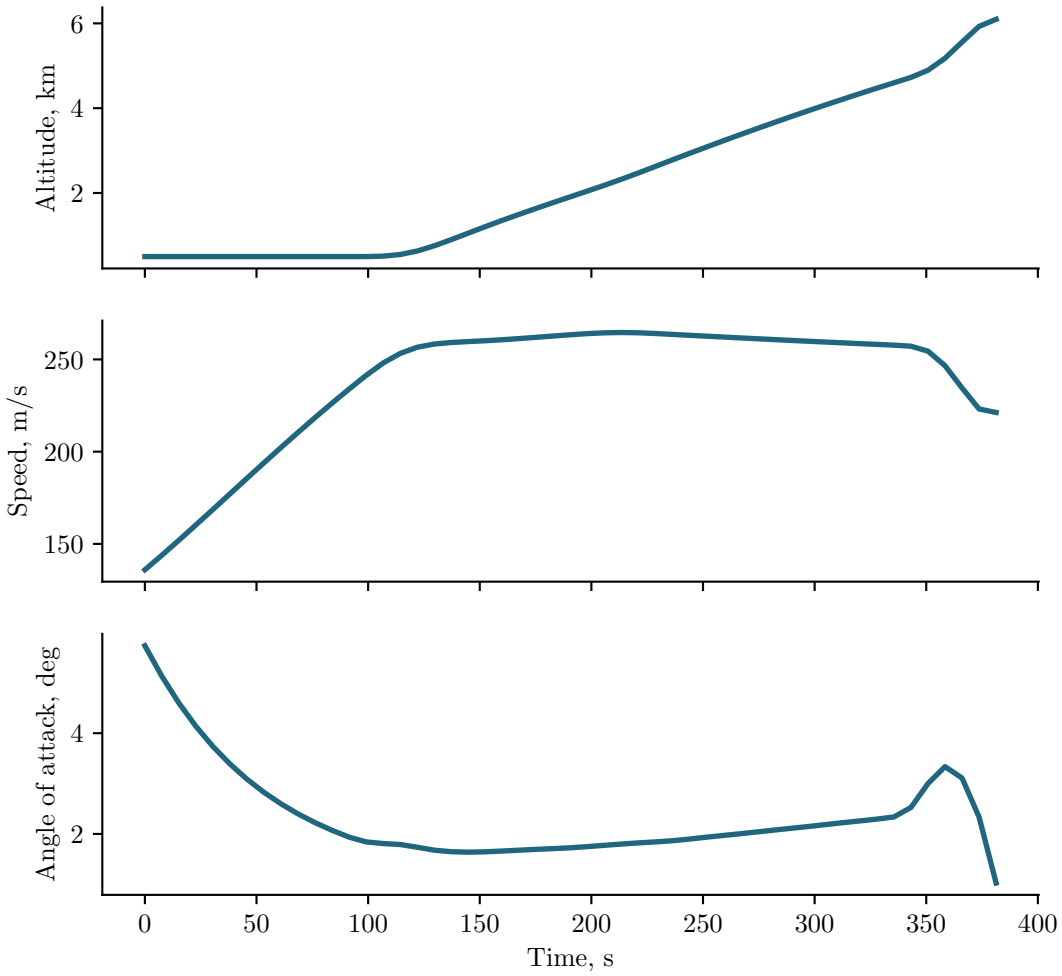


Figure 8: Climb trajectory of the BWB.

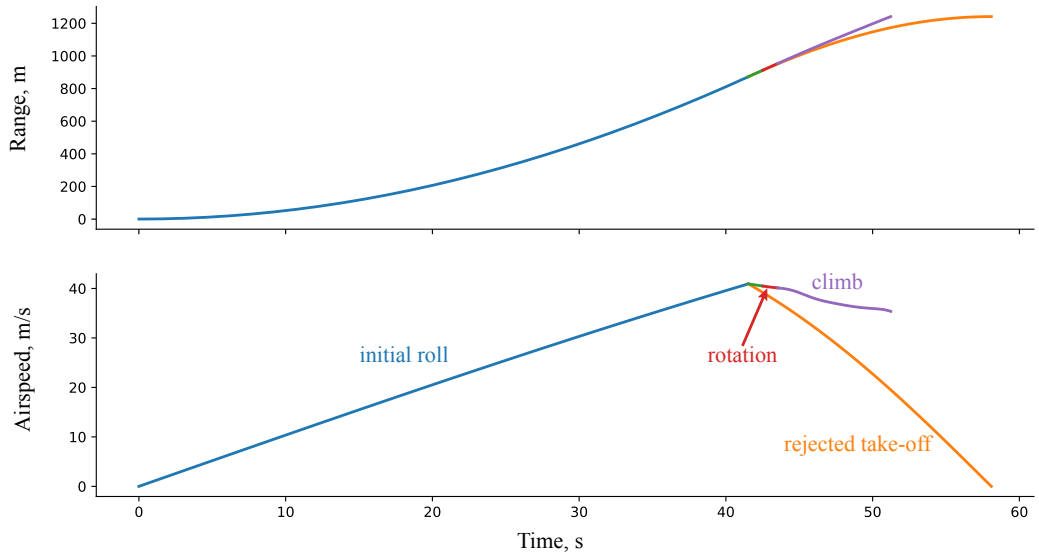


Figure 9: Take-off trajectories for the optimal BWB configuration.

In this system of equations, the system states y (including rigid-body degrees of freedom (DOF) and unsteady aerodynamics) and control states u are:

$$y = \begin{bmatrix} \varepsilon \\ \dot{\varepsilon} \\ \beta \\ \zeta \\ P_B \end{bmatrix}, \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}. \tag{51}$$

Note, that control inputs are user defined (based on the vehicle) and include all control surfaces and forces (such as thrust) that are of interest to the problem. The linearized ODE in Equation 50 can be rearrange to obtain:

$$\dot{y} = \tilde{f}\left(y, u\right) \tag{52}$$

$$= \underbrace{Q_1^{-1} Q_2}_{A} y + \underbrace{Q_1^{-1} \frac{\partial f}{\partial u}}_{B} u. \tag{53}$$

Equation 52 forms the basis for an ODE component within Perseids which is then provided to Dymos to obtain the time-domain component. The strains, ε, and constraint states, λ_c, are passed from the static solution to the dynamic solver component. With these initial conditions, the dynamic component solves the equations of motion in the time-domain by determining the state rate, $\dot{y}$ (rather than the linearized matrices that are required for frequency domain solutions) at each point of evaluation.

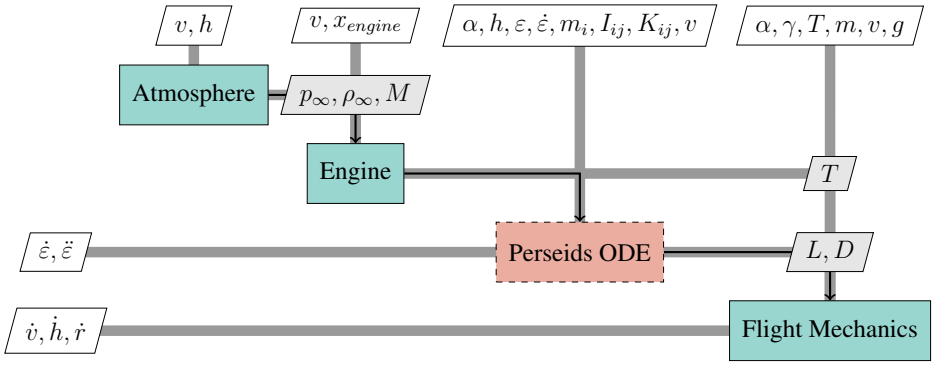


Figure 10: XDSM diagram of a fully coupled dynamic aeroelastic trajectory optimization.

This procedure can either be applied to the entire aeroelastic system (including rigid body degrees of freedom) or only including the strain states. Within this work, the gradient-based MDO problem was augmented by the strain states only. The resulting states were used to determine the lift and drag of the aircraft, which were transferred to the same flight mechanics equations used for the quasi steady case (Figure 10).

While the dynamic gradient-based optimization was run for this paper, it did not converge. A likely cause for this can be found in the increase in numbers of states and constraints the optimizer needs to negotiation. The quasi-steady optimization features 5 states per collocation point, whereas the dynamic aeroelastic problem features 125. This results in additional computational cost per iteration as well as a significant burden to the optimizer.

6 CONCLUDING REMARKS

This paper presented a geometrically nonlinear formulation for including quasi-steady aeroelastic effects into trajectory optimization. The method was applied to take-off and climb trajectory optimization, allowing designers to investigate aeroelastic performance and its effect on aircraft requirements and certification specifications. The quasi-steady formulation was amended to incorporate full structural dynamic states. This formulation could enable future transient, geometrically nonlinear aeroelastic optimizations of very flexible vehicles (such as HALE aircraft) during critical flight phases (climb, gust response, roll constraints, etc.).

However, optimizations using the fully transient formulation failed to converged when using collocation methods. Exposing large numbers of states as design variables and constraints to the optimizer proves challenging and counterproductive to achieving MDO problems with larger number of disciplines and load cases. As such, it appears necessary to solve the transient problem either by using a sub-optimization problem or a nonlinear solver, to avoid overburdening the top-level optimizer. Future studies are required to demonstrate the practicality of such an approach.

7 REFERENCES

[1] Noll, T. E., Brown, J. M., Perez-Davis, M. E., et al. (2004). Investigation of the Helios Prototype Aircraft Mishap. Mishap Report, NASA.

[2] Brown, E. (2003). *Integrated strain actuation in aircraft with highly flexible composite wings*. Ph.D., Massachussetts Institute of Technology, Cambridge, MA.

[3] Patil, M. J., Hodges, D. H., and Cesnik, C. E. S. (2001). Nonlinear Aeroelasticity and Flight Dynamics of High-Altitude Long-Endurance Aircraft. *Journal of Aircraft*, 38(1), 88–94. ISSN 0021-8669, 1533-3868. doi:10.2514/2.2738.

[4] Cesnik, C. E. S. and Su, W. (2005). Nonlinear Aeroelastic Modeling and Analysis of Fully Flexible Aircraft. In *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Austin, Texas: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-065-9, pp. 1–27. doi:10.2514/6.2005-2169.

[5] Su, W. and Cesnik, C. E. S. (2011). Dynamic Response of Highly Flexible Flying Wings. *AIAA Journal*, 49(2), 324–339. ISSN 0001-1452, 1533-385X. doi:10.2514/1.J050496.

[6] Shearer, C. and Cesnik, C. (2006). Trajectory Control of Very Flexible Aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. Keystone, Colorado: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-046-8, pp. 1–43. doi:10.2514/6.2006-6316.

[7] Patil, M. J. and Hodges, D. H. (2006). Flight Dynamics of Highly Flexible Flying Wings. *Journal of Aircraft*, 43(6), 1790–1799. ISSN 0021-8669, 1533-3868. doi:10.2514/1.17640.

[8] Wang, Z., Chen, P. C., Liu, D. D., et al. (2006). Time Domain Nonlinear Aeroelastic Analysis for HALE Wings. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference
 14th AIAA/ASME/AHS Adaptive Structures Conference
 7th*. Newport, Rhode Island: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-040-6, pp. 1–19. doi:10.2514/6.2006-1640.

[9] Cesnik, C., Senatore, P., Su, W., et al. (2010). X-HALE: A Very Flexible UAV for Nonlinear Aeroelastic Tests. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference
 18th AIAA/ASME/AHS Adaptive Structures Conference
 12th*. Orlando, Florida: American Institute of Aeronautics and Astronautics. ISBN 978-1-60086-961-7, pp. 1–22. doi:10.2514/6.2010-2715.

[10] Variyar, A., Economon, T. D., and Alonso, J. J. (2017). Design and Optimization of Unconventional Aircraft Configurations with Aeroelastic Constraints. In *55th AIAA Aerospace Sciences Meeting*. Grapevine, Texas: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-447-3, pp. 1–13. doi:10.2514/6.2017-0463.

[11] Lupp, C. A. and Cesnik, C. E. S. (2019). A Gradient-Based Flutter Constraint Including Geometrically Nonlinear Deformations. In *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-578-4, pp. 1–22. doi:10.2514/6.2019-1212.

[12] Falck, R. D. and Gray, J. S. (2019). Optimal Control within the Context of Multidisciplinary Design, Analysis, and Optimization. In *AIAA Scitech 2019 Forum*. San Diego, California: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-578-4, pp. 1–17. doi:10.2514/6.2019-0976.

[13] Gray, J. S., Hearn, T. A., and Naylor, B. A. (2019). Using Graph Coloring To Compute Total Derivatives More Efficiently in OpenMDAO. pp. 1–15. doi:10.2514/6.2019-3108.

[14] Falck, R. D., Ingraham, D., and Aretskin-Hariton, E. (2018). Multidisciplinary Optimization of Urban-Air-Mobility Class Aircraft Trajectories with Acoustic Constraints. In *2018 AIAA/IEEE Electric Aircraft Technologies Symposium*. Cincinnati, Ohio: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-572-2, pp. 1–7. doi: 10.2514/6.2018-4985.

[15] Hendricks, E. S., Aretskin-Hariton, E., Ingraham, D., et al. (2020). Multidisciplinary Optimization of an Electric Quadrotor Urban Air Mobility Aircraft. In *AIAA Aviation Forum*. Virtual Event: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-598-2, pp. 1–25. doi:10.2514/6.2020-3176.

[16] Jasa, J. P., Mader, C. A., and Martins, J. R. R. A. (2018). Trajectory Optimization of a Supersonic Aircraft with a Thermal Fuel Management System. In *2018 Multidisciplinary Analysis and Optimization Conference*. Atlanta, Georgia: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-550-0, pp. 1–16. doi:10.2514/6.2018-3884.

[17] Hendricks, E. S., Falck, R. D., and Gray, J. S. (2017). Simultaneous Propulsion System and Trajectory Optimization. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. Denver, Colorado: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-507-4, pp. 1–18. doi:10.2514/6.2017-4435.

[18] Lin, B., Carpenter, M., and de Weck, O. (2020). Simultaneous Vehicle and Trajectory Design using Convex Optimization. In *AIAA Scitech 2020 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-595-1, pp. 1–18. doi:10.2514/6.2020-0160.

[19] Lupp, C. A., Clark, D. L., Aksland, C. T., et al. (2022). Mission and Shape Optimization of a HALE Aircraft including Transient Power and Thermal Constraints. In *AIAA AVIATION 2022 Forum*. Chicago, IL & Virtual: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-635-4, pp. 1–26. doi:10.2514/6.2022-3935.

[20] Holden, M. and Kroo, I. (1996). A collocation method for aeroelastic optimization. In *6th Symposium on Multidisciplinary Analysis and Optimization*. Bellevue,WA,U.S.A.: American Institute of Aeronautics and Astronautics, pp. 1–6. doi:10.2514/6.1996-3980.

[21] del Carre, A. and Palacios, R. (2020). Simulation and Optimization of Takeoff Maneuvers of Very Flexible Aircraft. *Journal of Aircraft*, 57(6), 1097–1110. ISSN 1533-3868. doi: 10.2514/1.C035901.

[22] Sobieszczanski-Sobieski, J., Agte, J. S., and Sandusky, R. R. (2000). Bilevel Integrated System Synthesis. *AIAA Journal*, 38(1), 164–172. ISSN 0001-1452, 1533-385X. doi: 10.2514/2.937.

[23] Martins, J. R. R. A. and Hwang, J. T. (2013). Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models. *AIAA Journal*, 51(11), 28.

[24] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., et al. (2019). OpenMDAO: an open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4), 1075–1104. ISSN 1615-147X, 1615-1488. doi: 10.1007/s00158-019-02211-z.

[25] Fox, R. L. (1965). Constraint surface normals for structural synthesis techniques. *AIAA Journal*, 3(8), 1517–1518. ISSN 0001-1452, 1533-385X. doi:10.2514/3.3182.

[26] Arora, J. S. and Haug, E. J. (1979). Methods of Design Sensitivity Analysis in Structural Optimization. *AIAA Journal*, 17(9), 970–974. ISSN 0001-1452, 1533-385X. doi:10.2514/3.61260.

[27] Hascoët, L. and Pascual, V. (2013). The Tapenade Automatic Differentiation tool: Principles, Model, and Specification. *ACM Transactions On Mathematical Software*, 39(3).

[28] Sagebaum, M., Albring, T., and Gauger, N. R. (2019). High-Performance Derivative Computations using CoDiPack. *ACM Transactions on Mathematical Software*, 45(4), 1–26. ISSN 0098-3500, 1557-7295. doi:10.1145/3356900.

[29] Moses, W. S. and Churavy, V. (2020). Instead of Rewriting Foreign Code for Machine Learning, Automatically Synthesize Fast Gradients. *Advances in Neural Information Processing Systems*, 33.

[30] Lattner, C. and Adve, V. (2004). LLVM: A compilation framework for lifelong program analysis & transformation. In *International Symposium on Code Generation and Optimization*. San Jose, CA, USA: IEEE. ISBN 978-0-7695-2102-2, pp. 75–86. doi:10.1109/CGO.2004.1281665.

[31] Rosenbrock, H. H. (1960). An Automatic Method for finding the Greatest of Least Value of a Function. *The Computer Journal*, 3(3), 175–184.

[32] Shearer, C. M. (2006). *Coupled nonlinear flight dynamics, aeroelasticity, and control of very flexible aircraft*. Ph.D. thesis, University of Michigan, Ann Arbor, Michigan.

[33] Herman, A. L. and Conway, B. A. (1996). Direct optimization using collocation based on high-order Gauss-Lobatto quadrature rules. *Journal of Guidance, Control, and Dynamics*, 19(3), 592–599. ISSN 0731-5090, 1533-3884. doi:10.2514/3.21662.

[34] Garg, D., Patterson, M., Darby, C., et al. (2009). Direct Trajectory Optimization and Costate Estimation of General Optimal Control Problems Using a Radau Pseudospectral Method. In *AIAA Guidance, Navigation, and Control Conference*. Chicago, Illinois: American Institute of Aeronautics and Astronautics. ISBN 978-1-60086-978-5, pp. 1–29. doi:10.2514/6.2009-5989.

[35] Cesnik, C. E. S. and Brown, E. (2002). Modeling of High Aspect Ratio Active Flexible Wings for Roll Control. In *43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Denver, Colorado: American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-117-5, pp. 1–15. doi:10.2514/6.2002-1719.

[36] Shearer, C. M. and Cesnik, C. E. S. (2007). Nonlinear Flight Dynamics of Very Flexible Aircraft. *Journal of Aircraft*, 44(5), 1528–1545. ISSN 0021-8669, 1533-3868. doi:10.2514/1.27606.

[37] Saad, Y. and Schultz, M. H. (1986). GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3), 856–869. ISSN 0196-5204, 2168-3417. doi:10.1137/0907058.

[38] Jones, E., Oliphant, T., Peterson, P., et al. (2001). SciPy: Open source scientific tools for Python.

[39] Armijo, L. (1966). Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16(1), 1–3. ISSN 0030-8730, 0030-8730. doi:10.2140/pjm.1966.16.1.

[40] Raymer, D. (2018). *Aircraft design: A conceptual approach*. AIAA education series. American Institute of Aeronautics and Astronautics, Incorporated. ISBN 978-1-62410-490-9.

[41] Jasa, J. P., Hwang, J. T., and Martins, J. R. R. A. (2018). Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4), 1815–1827. ISSN 1615-1488. doi:10.1007/s00158-018-1912-8.

[42] Torenbeek, E. (1982). *Synthesis of Subsonic Airplane Design*. Dordrecht: Springer, 1 ed. ISBN 978-90-247-2724-7.

[43] Bryson, A. E. (1999). *Dynamic Optimization*. Menlo Park, CA: Addison Wesley Longman, 1 ed. ISBN 0-201-59790-X.

[44] Cavcar, M. (2006). Bréguet Range Equation? *Journal of Aircraft*, 43(5), 1542–1544. ISSN 0021-8669, 1533-3868. doi:10.2514/1.17696.

[45] Devillers, R. (1918). La Dynamique de l'Avion. Tech. rep., Librairie Aéronautique, Paris.

[46] Coffin, J. G. (1920). A Study of Airplane Range and Useful Loads. NACA-TR-69, NACA.

[47] Breguet, L. (1923). Calcul du poids de combustible consummé par un avion en vol ascendant. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 177, 870–872.

[48] Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57. ISSN 0025-5610, 1436-4646. doi:10.1007/s10107-004-0559-y.

[49] Peters, D. A., Karunamoorthy, S., and Cao, W.-M. (1995). Finite state induced flow models. I - Two-dimensional thin airfoil. *Journal of Aircraft*, 32(2), 313–322. ISSN 0021-8669, 1533-3868. doi:10.2514/3.46718.

COPYRIGHT STATEMENT