



# Adjoint-based analysis and optimization of beam-like structures subjected to dynamic loads

David Solano<sup>1</sup> · Darshan Sarojini<sup>1</sup> · Dushhyanth Rajaram<sup>1</sup> · Dimitri N. Mavris<sup>1</sup>

Received: 3 May 2021 / Revised: 21 October 2021 / Accepted: 28 October 2021 / Published online: 18 January 2022  
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

## Abstract

This paper presents the development of a framework to directly size and optimize a structure under dynamic loads with a special focus on enabling early aircraft design. Low degrees of freedom (DoF) beam models are used in lieu of high DoF models to represent the structure, thereby significantly reducing the computational cost (both runtime and memory). For a given residual system of equations, the adjoint method for time-dependent problems is solved using the stable backward difference formula scheme. This generalization allows for the implementation of a generic, equation-agnostic adjoint solver. Constraint aggregation techniques are extended to both spatial and temporal domains to constrain the strength-based stress yield criterion throughout the dynamic simulation. Consequently, the structure can directly be sized and optimized under the dynamic loads, without the need to rely on conversion to equivalent static loads. Finally, the developed framework is leveraged to demonstrate and quantify the consequence of neglecting dynamic loads on a notional ‘wing-like’ wingbox structure subjected to several dynamic gust load profiles.

**Keywords** Adjoint method · Dynamic loads · Structural optimization · Beam models · Aircraft structural design

## 1 Introduction

Designing components to sustain worst-case loads, as defined by Federal Aviation Administration (FAA) through Airworthiness Standards in 14 CFR Part 25 Subpart-C Structure (FAA 2019), is the primary role of structural sizing in the aircraft design process. The regulations mandate that the aircraft must be structurally sound under *dynamic loads*. Current aircraft design procedures heavily rely on computational tools to size the structure to ensure compliance with the regulations. Broadly speaking, the following procedure is typically adopted to size the structure: (1) dynamic loads are first computed by coupling aerodynamics, structural dynamics, flight mechanics, and controls (Shearer and Cesnik 2007; Meirovitch and Tuzcu 2004; Patil 1999; Karpel et al. 2004); (2) then, the dynamic loads are converted into a set of critical static load cases (Kim and Park 2010; Kang et al. 2005); and finally, (3)

the structure is approximately represented as either a shell model (Kennedy and Martins 2010, 2013, 2014a, b; Cavagna et al. 2009, 2011; Kenway et al. 2014; Kennedy et al. 2008; Kenway et al. 2012) or a beam model (Takahashi and Lemonds 2015; Chauhan and Martins 2018) and sized for critical static load cases.

The process of converting the dynamic loads into equivalent static loads typically leads to a large number of load cases that the structure must be sized for. Conventional tube-and-wing aircraft design methods have matured to the extent that thousands of equivalent static load cases can be systematically reduced to a smaller set of critical load cases (Sinha et al. 2021; Dharmasaroja et al. 2017). However, for novel aircraft concepts and configurations, such heuristic, albeit systematic reductions, may no longer be valid. For highly flexible or multi-body systems, the sizing and optimization algorithm will likely favor different designs depending on whether the problem is static or dynamic because the goals of the study may be different. For instance, sizing and optimization based on equivalent static loads tend to favor designs with higher stiffness, whereas under dynamic loads, the algorithms tend to drive the structure’s eigenfrequencies away from the driving frequency of the system (Stolpe et al. 2018; Stolpe 2014). Therefore, methods applicable in the

---

Responsible Editor: Ming Zhou

✉ Darshan Sarojini  
dsarshan@gatech.edu

<sup>1</sup> Georgia Institute of Technology, Atlanta, GA 30332, USA

early design must account for dynamics explicitly without relying on ad hoc conversions to equivalent static load cases.

Explicit consideration of dynamical loads is challenging, especially in early design. One, shell models (Cavagna et al. 2009, 2011; Kennedy and Martins 2010, 2013, 2014a; Kenway et al. 2014; Kennedy et al. 2008; Kenway et al. 2012) typically employed in early design have large degrees of freedom (DoF), making them computationally expensive. Two, shell models offer control over the thicknesses, composite ply orientations, etc., leading to a large design space and as a result, many expensive function and gradient calls when used in many-query exercises. Consequently, sizing for dynamical loads with shell models renders many-query exercises impractical. To avoid compute intensive analyses, this work advocates for the use of low DoF beam models (Nguyen 2008; Nguyen and Tuzcu 2009; Nguyen et al. 2014, 2016; Ting et al. 2014; Drela 1990, 1999; Patil 1999; Patil et al. 2001; Raghavan 2009; Shearer and Cesnik 2007; Palacios and Cesnik 2005) or modal representations (Hurty 1965; Waszak and Schmidt 1988; Schmidt and Raney 2001; Pedro and Bigg 2005; Karpel 1999; Karpel et al. 2004, 2005; Moulin and Karpel 2007; Raveh et al. 2000, 2001). Between beam models and modal analysis, the former is nonlinear, and hence preferred for highly flexible wings (Patil 1999). Beam models are a powerful tool for structural analysis and optimization. Variational asymptotic method-based approaches (Hodges 2006) have been successfully used in the analysis and design of rotorcraft blades (Ku et al. 2007; Li et al. 2008; Hodges and Yu 2007), wind turbines (Richards et al. 2014; Lee et al. 2002), and very flexible aircraft such as HALE (Cesnik et al. 2012).

Initial structural sizing and optimization of the wing-box by representing it as a beam has been demonstrated by Takahashi and Lemonds (2015). Recently, an open-source python tool named OpenAeroStruct (Jasa et al. 2018; Chauhan and Martins 2018) has been shown to provide reasonable weight estimates at the early stages of aircraft design. Sarojini (2021) compared the beam theory framework in this paper to a shell model for the PEGASUS aircraft and showed a 6% error in weight estimate, but with a 7.6× speed-up in computational efficiency. It should be noted that the references cited earlier used Euler–Bernoulli (EB) beam theory to compute the stresses and failure criteria during the structural sizing process. A current limitation is the simplification to static critical loads and static structural analysis instead of dynamic loads and structural dynamic analysis. This work addresses this limitation.

Another issue concerning direct consideration of dynamical loads is related to the implementation and computational cost of the adjoint method for gradient computations. One, the adjoint method requires the entire state history of the forward-simulation to be stored and used in the adjoint simulation (Boopathy and Kennedy

2017, 2019). For models with large DoF, these data can amount to gigabytes of data for a single simulation. Two, analytical derivation of the adjoint is tedious and requires considerable effort to implement (Sarojini et al. 2020). Any additional changes, such as the inclusion of complex cross-sections or accurate techniques to compute stresses, require re-derivation and re-implementation of the adjoint-based gradient computation. Advances in automatic differentiation (AD) enable computation of derivatives by accumulating values during code execution to generate numerical derivative evaluations rather than derivative expressions. AD can be applied to regular code with minimal change, allowing branching, loops, and recursion (Baydin et al. 2018). Despite its name, AD does not fully automate differentiation and can yield inefficient code if naively implemented (Margossian 2019). Symbolic differentiation offers yet another solution by allowing practitioners to write regular code using symbols, leading to an easily modifiable and modular framework. However, it is memory and compute intensive (Margossian 2019). Hence, symbolic differentiation's use is impractical for large DoF models such as a shell model where symbolics must be used for each node in the computational mesh. Since low DoF beam models are sufficiently accurate for early-stage aircraft structural weight estimation (Takahashi and Lemonds 2015; Chauhan and Martins 2018) under static loads, symbolics can be used in addition to AD to perform efficient gradient computations. This work uses a hybrid approach that leverages both symbolic and AD provided by the CasADi package. The core of CasADi consists of a symbolic framework that requires users to construct expressions that automatically define differentiable functions. These general-purpose expressions have no notion of optimization and are best likened with expressions in, e.g., MATLAB's Symbolic Toolbox or Python's SymPy package. Once the expressions have been implemented, they are used to efficiently obtain new expressions for derivatives using AD (Andersson et al. 2012). The result can be interpreted as a hybrid of symbolic differentiation and AD that is capable of automatically generating extremely fast C code.

Finally, this work attempts to tackle another challenge encountered in implementing structural sizing for dynamic loads concerning handling time-dependent constraints in structural optimization. Throughout a dynamic maneuver, it must be ensured that the stress in the discretized representation of the structure is below the yield stress of the material, i.e., the stress at every point in the structure must be less than the yield stress throughout the dynamic maneuver. As it is impossible to know a priori, the time and location at which the stress will exceed the yield stress, constraints must be placed at every spatial and temporal location. In literature, a large number of such constraints are typically handled

using *spatial* constraint aggregation techniques (Poon and Martins 2007; Martins and Poon 2005; Kennedy and Hicken 2015). Aggregation in both space and time has been given limited attention (Boopathy and Kennedy 2017, 2019) for computationally expensive high-fidelity simulations. This study demonstrates one of the first empirical demonstrations of constraint aggregation in both space and time for beam models.

In summary, this work presents the development of a ground-up implementation of an equation-agnostic adjoint-based framework to directly size and optimize a structure under dynamic loads with a special focus on early aircraft design. Low DoF beam models are used in place of high DoF models to represent the structure, thereby significantly reducing the computational cost (runtime and memory). Via several sizing and optimization problems on a notional ‘wing-like’ wingbox, it will be demonstrated that (1) loads applied to a structure produce higher stresses when analyzed dynamically rather than statically. Thus, the resulting structure has larger thicknesses, thereby reinforcing the necessity of considering dynamics in early-stage aircraft structural sizing, and (2) the structure can be sized directly for dynamic loads, thereby obviating the need for conversion to equivalent critical static loads.

The remainder of the paper is organized as follows: Sect. 2 provides background on the adjoint method (Sect. 2.1) and constraint aggregation (Sect. 2.2). Section 3 presents the framework to perform structural optimization using the adjoint method and constraint aggregation when applied to the equations of a nonlinear beam theory. Section 4 demonstrates the proposed approach on a case study of optimization of a wing-like structure subjected to dynamic loads.

## 2 Background

### 2.1 Adjoint method

The governing equations for many systems of interest are second-order differential equations in time. It is possible to convert any second-order system into a first-order system using standard approaches. Any set of governing equations can be written in a generic form as follows:

$$\mathbf{R}(\boldsymbol{\mu}, \mathbf{x}, \dot{\mathbf{x}}, t) = \mathbf{0}, \tag{1}$$

where  $\boldsymbol{\mu}$  represents the design variables,  $\mathbf{x}$  and  $\dot{\mathbf{x}}$  are the state variables and their first derivatives with respect to time (denoted by  $t$ ), respectively. Note that the chosen form of representing the system lends itself to a simple derivation of the adjoint system of equations.

Several methods exist for integrating the numerical differential algebraic equations (DAEs) arising from the governing

equations. Each method treats the discrete system of algebraic equations in a different manner (for instance, requiring solutions from different time steps). Therefore, the adjoint system must be derived separately for each method. In this work, the adjoint system of equations is derived for the backward difference formula (BDF) scheme due to its stability properties. The state approximation of the BDF scheme (Curtiss and Hirschfelder 1952; Boopathy and Kennedy 2017) is given by

$$\mathbf{S}_i = \frac{1}{\Delta t} \sum_{p=0}^P \alpha_p \mathbf{x}^{t-p} - \dot{\mathbf{x}}^t, \tag{2}$$

where  $\alpha_p$  are the BDF coefficients, and  $\Delta t$  the time step.

In this work, a second-order ( $P = 2$ ) BDF scheme is implemented. The time derivative at a time instant is defined as

$$\dot{\mathbf{x}}^t = \alpha_0 \mathbf{x}^t + \alpha_1 \mathbf{x}^{t-1} + \alpha_2 \mathbf{x}^{t-2} \tag{3}$$

with  $\alpha_0 = \frac{1.5}{\Delta t}$ ,  $\alpha_1 = \frac{-2}{\Delta t}$ , and  $\alpha_2 = \frac{0.5}{\Delta t}$ . At each time step, Newton’s method is used to solve the nonlinear system given by Eq. 1. Newton’s method uses the following linear solve at each iteration

$$\left[ \frac{\partial \mathbf{R}}{\partial \mathbf{x}^t} + \alpha_0 \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}^t} \right] \delta_{\mathbf{x}} = -\mathbf{r}^t, \tag{4}$$

where  $\frac{\partial \mathbf{R}}{\partial \mathbf{x}^t}$  and  $\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}^t}$  are the Jacobians of the residual system, and  $\mathbf{r}$  is the residual vector when Eq. 1 is evaluated for the current guess of  $\mathbf{x}$  and  $\dot{\mathbf{x}}$ . If the residual vector is nonzero, Newton’s method creates a new guess for the state variable using

$$\mathbf{x}^t = \mathbf{x}^t + \delta_{\mathbf{x}}. \tag{5}$$

The algorithm for the dynamic simulation’s forward solve is given in “Appendix B”—Algorithm 1.

In the context of design through numerical optimization, an objective/constraint functional can be expressed as a sum over time as

$$f(\boldsymbol{\mu}) = \int_0^T F(\boldsymbol{\mu}, \mathbf{x}, \dot{\mathbf{x}}, t) dt \approx \sum_{t=0}^T \Delta t F^t(\boldsymbol{\mu}, \mathbf{x}^t, \dot{\mathbf{x}}^t). \tag{6}$$

Here, the integrand  $F$  can be any arbitrary function of the state and design variables. What follows is a summary of the equations that will be used for the discrete-time adjoint system of equations (Boopathy and Kennedy 2017). The Lagrangian can be written as follows:

$$\mathbf{L} = \sum_{t=0}^T \Delta t F^t + \sum_{i=0}^n \Delta t (\lambda^i)^T \mathbf{R}^i + \sum_{i=0}^T (\boldsymbol{\phi}^i)^T \mathbf{S}^i, \tag{7}$$

where  $\mathbf{R}^t$  is the residual of the governing equations,  $\mathbf{S}^t$  denotes the state approximation resulting from the BDF scheme,  $\lambda$  and  $\phi$  are the corresponding Lagrange multipliers, respectively,  $\Delta t$  denotes the step size in time, and superscript  $t$  denotes position in time.

Setting the partial derivative of the Lagrangian w.r.t. to the state variables to zero results in a linear system of the following form at each time step,  $t$

$$\left(\frac{\partial \mathbf{R}^t}{\partial \mathbf{x}} + \alpha_0 \frac{\partial \mathbf{R}^t}{\partial \dot{\mathbf{x}}}\right)^T \lambda^t = -\left(\frac{\partial F^t}{\partial \mathbf{x}} + \alpha_0 \frac{\partial F^t}{\partial \dot{\mathbf{x}}}\right) - \left(\sum_{p=1}^P \alpha_p \frac{\partial \mathbf{R}^{t+p}}{\partial \mathbf{x}} \lambda_{t+p} - \sum_{p=1}^P \alpha_p \frac{\partial F^{t+p}}{\partial \dot{\mathbf{x}}}\right). \tag{8}$$

Once the adjoint variables at each time step are computed, the derivative of the functional of interest w.r.t. design variables can be found as

$$\frac{df}{d\boldsymbol{\mu}} = \sum_{t=0}^T \Delta t \frac{\partial F^t}{\partial \boldsymbol{\mu}} + \sum_{t=0}^T \Delta t (\lambda^t)^T \frac{\partial \mathbf{R}^t}{\partial \boldsymbol{\mu}}. \tag{9}$$

The formulation presented here results in a general method to obtain the gradient for a system of residual equations time-marched using a BDF scheme. The adjoint variable is solved at each time step using Eq. 8. The four quantities required are as follows:

- $\frac{\partial \mathbf{R}}{\partial \mathbf{x}}$ —the Jacobian matrix of the derivative of the residual equations w.r.t. the state variables,
- $\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}}$ —the Jacobian matrix of the derivative of the residual equations w.r.t. the first time derivative of the state variables
- $\frac{\partial F}{\partial \mathbf{x}}$ —the vector of derivatives of the function of interest w.r.t. the state variables,
- $\frac{\partial F}{\partial \dot{\mathbf{x}}}$ —the vector of derivatives of the function of interest w.r.t. the first time derivative of the state variables.

The gradient is then computed using Eq. 9 for which the following two additional quantities are required:

- $\frac{\partial F}{\partial \boldsymbol{\mu}}$ —the vector of derivatives of the function of interest w.r.t. the design variables,
- $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}}$ —the matrix of derivatives of the residual equations w.r.t. the design variables.

The numerical algorithm to obtain the gradient using the adjoint method is presented in “Appendix B”—Algorithm 2. An illustrative example that shows the application of the formulation above to a spring–mass–damper system to obtain the gradient of the integral of the potential energy with respect to the spring stiffness can be found in prior work by Sarojini et al. (2020).

## 2.2 Constraint aggregation

Consider a constraint  $g$  at location  $i$  in the domain and at time step  $t$  of the dynamic simulation

$$g_i^t \leq 0, \quad i = 1, \dots, N, \quad t = 0, \dots, T. \tag{10}$$

Note that the equation above results in a very large number of constraints  $\mathcal{O}(NT)$ . A large number of constraints results in a challenging-to-solve optimization problem. Hence, constraint aggregation methods are used (Kennedy and Hicken 2015). One possible constraint aggregation method is the Kreisselmeier–Steinhauser (KS) function given by

$$c_{ks} = m + \frac{1}{\rho_{ks}} \ln \left[ \sum_{i=1}^N e^{\rho_{ks}(g_i - m)} \right], \tag{11}$$

where  $m = \max g_i$ , and  $\rho_{ks}$  is a constraint aggregation parameter. Note that Eq. 11 is conventionally used when constraints are aggregated over space. However, the peak stress on the beam could occur at any time instant between 0 and  $T$ . As it is impossible to know a priori the time instant at which the peak load occurs, the constraints must also be aggregated over time. In this work, the KS functional in Eq. 11 is modified to integrate over both space and time as

$$c_{ks} = m + \frac{1}{\rho_{ks}} \ln \left[ \sum_{i=1}^N \sum_{t=0}^T e^{\rho_{ks}(g_i^t - m)} \right], \tag{12}$$

where  $m = \max g_i^t$ .

## 3 Adjoint method applied to a nonlinear beam theory

We first introduce the nomenclature of the symbolic framework. The symbolic variables denoted as  $\mathcal{V}^{\text{sym}}()$ . Operations on the symbolic variables result in symbolic expressions denoted as  $\mathcal{E}^{\text{sym}}()$ . A symbolic function,  $\mathcal{F}^{\text{sym}}()$ , of a symbolic expression is a function that accepts numeric inputs for the symbolic variables and computes numeric values of the symbolic expression.

The nonlinear beam theory used in this study (Drela 1999), when discretized, leads to a system of DAE. A summary of the equations was presented in a prior work by Sarojini et al. (2018). The quantities tracked at each node and at each time step are shown in Table 1. They include nodal positions, Euler angles, translational and angular velocities, accelerations, and reaction forces and moments. It must be noted that all the dynamic simulations in this manuscript use a damping ratio of 0.02, which corresponds to a continuous metallic structure (Drela 1999) and the simulations are initialized with an unloaded and undeflected structure.

For the static studies, the tip loads are applied as boundary conditions.

The process to create a symbolic beam model is shown as an extended design structure matrix (XDSM; Lambe and Martins 2012b) in Fig. 1. The number of cross-sections  $N$  and number of time steps  $T$  determine the size of the state vector. As seen in Table 1, 18 quantities are tracked per node, resulting in a state vector of size  $18N$  per time step. Thus, the symbolic variable for the state vector can be written as  $\sqrt{V}^{\text{sym}} \mathbf{x} \in \mathbb{S}^{18N \times T}$ . Some of the computations require a slice, in time, of the state vector, denoted as  $\sqrt{V}^{\text{sym}} \mathbf{x}^t \in \mathbb{S}^{18N}$ .

For each cross-section, the number of design variables  $M$  is dictated by the practitioners design choices. For example, a rectangular beam cross-section has width and height as design variables. A box cross-section, in general, has six design variables, i.e., width  $w$ , height  $h$ , thickness of the top flange  $t_{\text{top}}$ , thickness of the bottom flange  $t_{\text{bot}}$ , thickness of the left web  $t_{\text{left}}$ , and thickness of the right web  $t_{\text{right}}$ . For

aircraft wingbox structural optimization, the width and height are fixed by the airfoil and the locations of the spars, and the thicknesses are set as design variables. The appropriate number of symbolic cross-section design variables  $\sqrt{V}^{\text{sym}} \mathbf{cs}_{\text{dv}} \in \mathbb{S}^{MN}$  are generated.

The function *Symbolic beam properties* implements appropriate equations to compute the beam stiffness  $\mathbf{K}_i$  and mass matrices  $\mathbf{M}_i$  at each cross-section  $i$ . Analytical expressions exist for simple cross-sections such as rectangular and box beams. If the cross-section is represented as an airfoil, appropriate equations in literature may be used to obtain the beam matrices (Chauhan and Martins 2018). It is also possible to use higher-order beam theories such as a VABS (Yu 2011; Cesnik and Hodges 1997) or BECAS (Blasques 2012). If these higher-order theories are used, this block is an implicit function performing a linear solve. The final result is a set of symbolic expressions given as

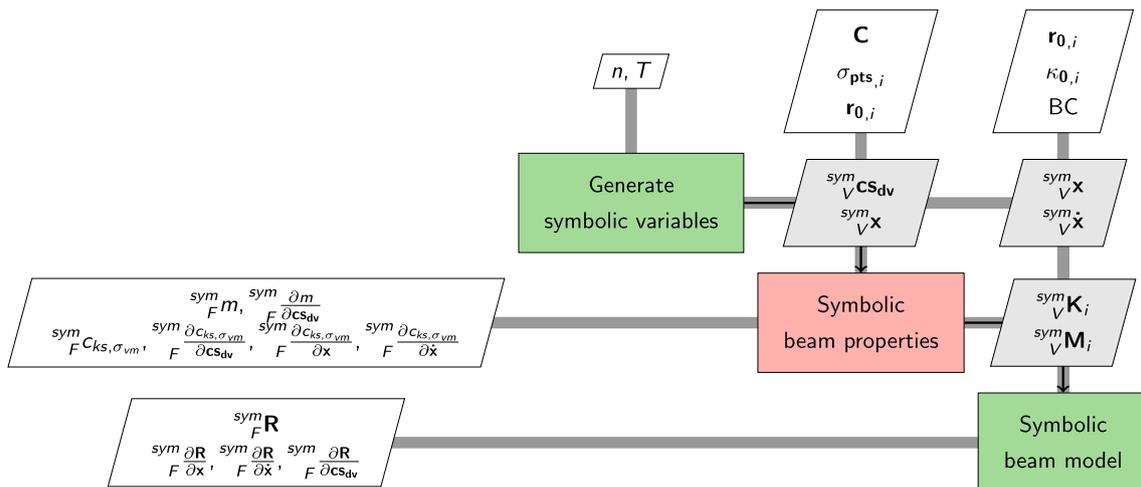
$$\sqrt{V}^{\text{sym}} \mathbf{K}_i \in \mathbb{S}^{6 \times 6} = f(\sqrt{V}^{\text{sym}} \mathbf{cs}_{\text{dv}}), \tag{13}$$

$$\sqrt{V}^{\text{sym}} \mathbf{M}_i \in \mathbb{S}^{6 \times 6} = f(\sqrt{V}^{\text{sym}} \mathbf{cs}_{\text{dv}}). \tag{14}$$

The function *Symbolic beam properties* also computes the mass and aggregated stress of the beam. Information from the state vector  $\sqrt{V}^{\text{sym}} \mathbf{x}$  is used to compute while performing the aggregation. As an example, the computation of axial stresses in a rectangular beam with EB assumptions is given by

**Table 1** Quantities tracked when solving a 1-D beam problem

Variable	Body-fixed axes for “i”th node
$\mathbf{r}_i = \{x_i, y_i, z_i\}$	Position
$\dot{\mathbf{r}}_i = \mathbf{u}_i = \{u_i, v_i, w_i\}$	Nodal velocities
$\ddot{\mathbf{r}}_i = \dot{\mathbf{u}}_i$	Nodal acceleration
$\theta_i = \{\psi, \theta, \phi\}$	Euler angles triad from body-fixed axes to csn axes
$\omega_i = \{p_i, q_i, r_i\}$	Nodal angular rates
$\dot{\omega}_i$	Nodal angular acceleration
$\mathbf{F}_i = \{F_x, F_y, F_z\}$	Force resultant
$\mathbf{M}_i = \{M_x, M_y, M_z\}$	Moment resultant



**Fig. 1** Generation of symbolic beam represented as a XDSM (Lambe and Martins 2012a).  $i = 1, \dots, N$  is the index for cross-sections along the span of the beam,  $t = 0, \dots, T$  the index for time steps in the dynamic simulation,  $\mathbf{C}$  the material properties matrix,  $\sigma_{\text{pts},i}$  the cross-

section locations of the stress recovery points,  $\mathbf{r}_0$  the undeformed beam axis node locations,  $\kappa_{0,i}$  the undeformed curvature matrix, and BC the boundary conditions

$$\sigma_i^t = \frac{M}{I} y = \frac{M_i^t}{w_i h_i^3} \frac{h_i}{2} = \frac{6M_i^t}{w_i h_i^2}, \quad (15)$$

where  $M_i^t$ , an element of the state vector  ${}^{\text{sym}}_{\mathbf{V}} \mathbf{x}$ , is the moment at cross-section  $i$  and time  $t$ . The corresponding stress constraint is given by

$$g_i^t = \frac{6M_i^t}{w_i h_i^2} - 1 \leq 0, \quad i = 1, \dots, N; \quad t = 0, \dots, T. \quad (16)$$

Aggregating the above stress using Eq. 12 yields  ${}^{\text{sym}}_{\mathbf{F}} c_{\text{ks},\sigma_{\text{vm}}}$ . Thus the following symbolic functions are generated.

$${}^{\text{sym}}_{\mathbf{V}} m \rightarrow \mathbb{R} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}), \quad (17)$$

$${}^{\text{sym}}_{\mathbf{F}} c_{\text{ks},\sigma_{\text{vm}}} \rightarrow \mathbb{R} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}). \quad (18)$$

The symbolic expression of the constraint can be differentiated using AD with respect to the state vector and the design variables to obtain

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial c_{\text{ks},\sigma_{\text{vm}}}}{\partial \mathbf{cs}_{\text{dv}}} \rightarrow \mathbb{R}^{MN} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}^t), \quad (19)$$

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial c_{\text{ks},\sigma_{\text{vm}}}}{\partial \mathbf{x}} \rightarrow \mathbb{R}^{18N} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}), \quad (20)$$

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial c_{\text{ks},\sigma_{\text{vm}}}}{\partial \dot{\mathbf{x}}} \rightarrow \mathbb{R}^{18N} = \mathbf{0}. \quad (21)$$

Similarly, the gradient of the objective function is found as

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial m}{\partial \mathbf{cs}_{\text{dv}}} \rightarrow \mathbb{R}^M = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}). \quad (22)$$

Finally, the residual equations of the beam are generated as symbolic expressions in the *Symbolic beam model* block. The resulting symbolic function

$${}^{\text{sym}}_{\mathbf{F}} \mathbf{R} \rightarrow \mathbb{R}^{18N} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}, {}^{\text{sym}}_{\mathbf{V}} \dot{\mathbf{x}}) \quad (23)$$

accepts numeric values of the cross-section design variables, guesses values of the state vector and the time derivative of the state vector, and returns the residual vector  $\in \mathbb{R}^{18N}$ . As mentioned earlier, the symbolic expression of the residual equation can be differentiated using AD with respect to the state vector, and the time derivative of the state vector to obtain the Jacobian matrices

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial \mathbf{R}}{\partial \mathbf{x}} \rightarrow \mathbb{R}^{18N \times 18N} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}, {}^{\text{sym}}_{\mathbf{V}} \dot{\mathbf{x}}), \quad (24)$$

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \rightarrow \mathbb{R}^{18N \times 18N} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}, {}^{\text{sym}}_{\mathbf{V}} \dot{\mathbf{x}}) \quad (25)$$

and with respect to the cross-section design variables to obtain

$${}^{\text{sym}}_{\mathbf{F}} \frac{\partial \mathbf{R}}{\partial \mathbf{cs}_{\text{dv}}} \rightarrow \mathbb{R}^{18 \times MN} = f({}^{\text{sym}}_{\mathbf{V}} \mathbf{cs}_{\text{dv}}, {}^{\text{sym}}_{\mathbf{V}} \mathbf{x}^t, {}^{\text{sym}}_{\mathbf{V}} \dot{\mathbf{x}}^t). \quad (26)$$

The process described thus far provides the following six partial derivative quantities needed for the forward and adjoint solvers mentioned earlier in Sect. 2.1:  $\frac{\partial \mathbf{R}}{\partial \mathbf{x}}$ : Eq. 24,  $\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}}$ : Eq. 25,  $\frac{\partial F}{\partial \mathbf{x}}$ : Eq. 20,  $\frac{\partial F}{\partial \dot{\mathbf{x}}}$ : Eq. 21,  $\frac{\partial F}{\partial \mu}$ : Eq. 19, and  $\frac{\partial \mathbf{R}}{\partial \mu}$ : Eq. 26.

It must be noted that the developed framework is agnostic to the type of beam theory used for stress recovery. The framework may be used if one of the two following criteria are met: (1) analytical equations are used for stress recovery, or (2) the gradient of the stress with respect to the design variables is provided. The test case in Sect. 4 uses a box cross-section beam shown in Fig. 14, with stresses recovered at 12 extremity points. For completeness, ‘‘Appendix A’’ contains the standard formulae to compute the beam properties and stresses for a box beam using EB assumptions. This assumption meets criterion 1 stated above. However, the beam theory used in this work is illustrative. Using Timoshenko theory would involve using different analytical equations, and is supported by the framework. If higher accuracy is desired, higher-order beam theory such as VABS (Yu 2011) may be used. To successfully use VABS—that does not have analytical equations for the stress recovery—criterion 2 must be satisfied, i.e., the burden of providing the gradient is on the analysis code.

## 4 Structural optimization

The test case considered is a ‘wing-like’ geometry shown in Fig. 2 consisting of four cross-sections, each with four design variables. The structure is cantilevered at cross-section 1 at the end left. The optimization problem setup is given in Eq. 27.

$$\begin{aligned}
 &\text{minimize} && \text{mass} \\
 &\text{subject to} && \mathbf{R}(\mu, \mathbf{x}, \dot{\mathbf{x}}, t, P, E) = \mathbf{0} \\
 & && c_{ks,\sigma} \leq 0 \\
 & && 0.003 \leq t_{\text{left},i} \leq 0.25 \\
 & && 0.003 \leq t_{\text{top},i} \leq 1.5 \\
 & && 0.003 \leq t_{\text{right},i} \leq 0.25 \\
 & && 0.003 \leq t_{\text{bot},i} \leq 1.5 \\
 &\text{data} && P(t) \text{ (N)}, E \text{ (Pa)}, \\
 & && \sigma_y \text{ (Pa)}, \rho \text{ (kg/m}^3\text{)}.
 \end{aligned} \tag{27}$$

In this problem,  $P(t)$  is the applied external load as a function of time,  $E$ ,  $\rho$  and  $\sigma_y$  are the Young’s modulus, density, and yield stress of the material of the beam, respectively, and  $c_{ks,\sigma}$  denotes the aggregated constraint of the stress failure criteria over space and time. For all optimization problems considered in this work, a relative change of  $10^{-8}$  in the value of the objective function and a relative change of  $10^{-10}$  in design variables between successive iterations were used as termination criteria.

### 4.1 Adjoint validation

The implementation of the adjoint method is validated by computing the derivative of the aggregated stress constraint  $c_{ks,\sigma}$  for von Mises stress with respect to the 16 design variables using the adjoint method and compared with those computed using a central difference stencil. A tip load of  $F_x = 1 \times 10^6$  N and  $F_z = F_x = 1 \times 10^6$  N is applied for 3 s. The simulation is advanced with a time step of 0.01 s. The resulting adjoint gradient vector’s components are shown in Table 2. Note that the adjoint gradients are within 2% error from their finite difference equivalents, making them suitable for optimization.

To test the scalability of the method, additional cross-sections are introduced in-between the four cross-sections

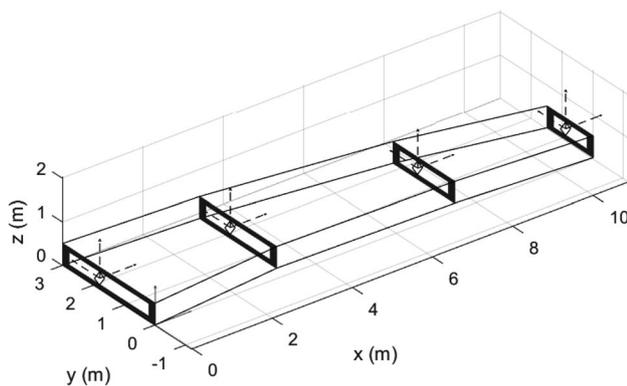


Fig. 2 Wing-like box beam with sweep, taper, and dihedral

seen in Fig. 2. Each additional cross-section adds four design variables to the problem. Table 3 compares the wall-time to compute the gradient using finite differences and the adjoint method as the number of design variables is varied. It can be clearly observed that the wall-time to compute the gradient using finite differences becomes infeasible as the number of design variables increases.

### 4.2 Study 1: static loading stress validation

Consider sizing the structure for the following conditions:

- Tip load of  $F_x = F_y = 0$  N,  $F_z = 2 \times 10^6$  N,
- Midpoint von Mises stress constraints,
- Initial guess to optimizer set to the upper bounds,
- Aluminum material with properties:  $E = 7 \times 10^9$  Pa,  $\rho = 2700$  kg/m<sup>3</sup>,  $\sigma_y = 276 \times 10^6$  Pa.

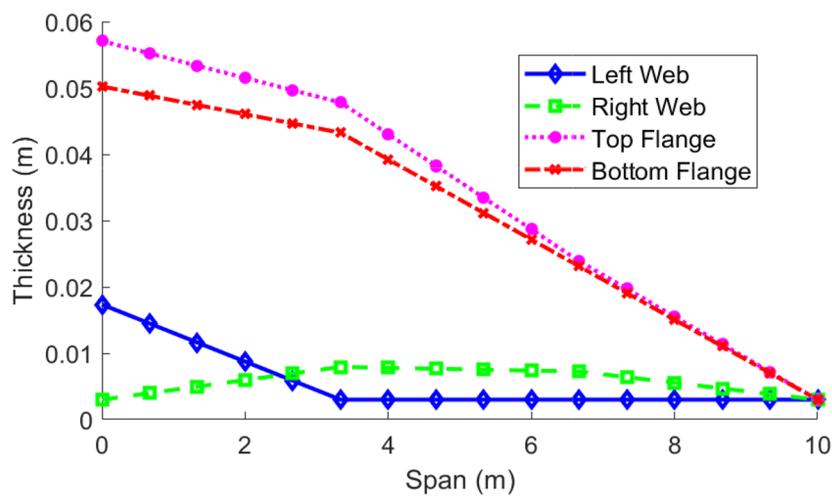
MATLAB’s fmincon function was used to perform the optimization with objective and constraint gradients supplied.

Table 2 Gradient comparison between adjoint method and central differences (step size:  $10^{-8}$ ) for 16 design variables (DV) for von Mises stress aggregated constraint  $c_{ks,\sigma}$

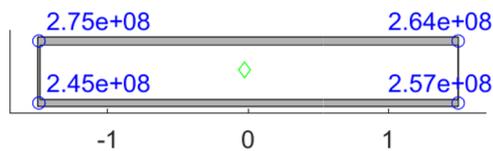
DV	FD	Adjoint	Error (%)
$t_{\text{left}_1}$	-0.14267	-0.14268	0.00505
$t_{\text{left}_2}$	-0.03586	-0.03587	0.021513
$t_{\text{left}_3}$	0.001537	0.001519	-1.21537
$t_{\text{left}_4}$	0.001505	0.001488	-1.12154
$t_{\text{top}_1}$	-2.72883	-2.72898	0.005573
$t_{\text{top}_2}$	-0.64338	-0.64348	0.016194
$t_{\text{top}_3}$	0.002966	0.002918	-1.61295
$t_{\text{top}_4}$	0.004861	0.004807	-1.10627
$t_{\text{right}_1}$	-0.14268	-0.14269	0.006823
$t_{\text{right}_2}$	-0.03596	-0.03597	0.02369
$t_{\text{right}_3}$	0.00125	0.001236	-1.1515
$t_{\text{right}_4}$	0.001358	0.001342	-1.13942
$t_{\text{bot}_1}$	-2.72883	-2.72898	0.005666
$t_{\text{bot}_2}$	-0.64336	-0.64346	0.016564
$t_{\text{bot}_3}$	0.002986	0.00294	-1.53316
$t_{\text{bot}_4}$	0.004865	0.004812	-1.09163

Table 3 Comparison of time (in seconds) to compute the gradient between finite-difference and adjoint method

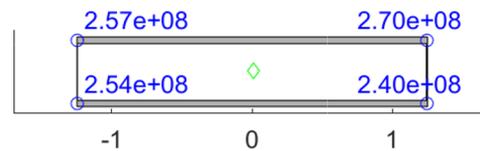
# DV	FD time (s)	Adjoint time (s)
20	6.581719	0.32008
40	17.78193	0.3244
100	87.24383	0.654731
500	1989.553	2.704534
1000	7917.029	5.341992



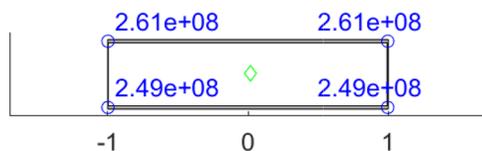
(a) Thickness as a function of span



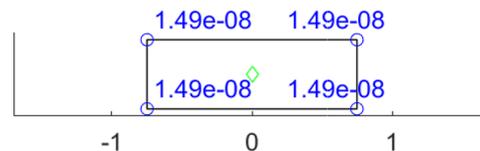
(b) Beam stresses on cross-section 1



(c) Beam stresses on cross-section 2



(d) Beam stresses on cross-section 3



(e) Beam stresses on cross-section 4

**Fig. 3** Optimum solution for study 1. In **a** the optimized thickness distribution as a function of the span is shown. In **b**, **c**, **d**, and **e** the von-Mises stresses (in Pa) are shown in blue, the axial stresses (in Pa) are shown in magenta, and the shear stresses are shown in red. (Color figure online)

The optimizer converged in 114 iterations satisfying constraints with a tolerance of  $1.074 \times 10^{-7}$ . The thickness distributions along the span are shown in Fig. 3a.

The von Mises stresses on cross-sections 1 through 4 are shown in Fig. 3b through e, respectively. It can be observed that the maximum stress on CS 1 ( $2.75 \times 10^8$  Pa) is close to the yield stress value ( $2.76 \times 10^8$  Pa). The maximum stresses on CS 2 ( $2.70 \times 10^8$  Pa) and CS 3 ( $2.61 \times 10^8$  Pa) are lower, but still close to the yield value. These results imply that the constraint aggregation in both space and time is working as expected, and the optimizer converges to a feasible optimum solution. Note that on the tip cross-section, i.e., CS 4, the stresses are essentially 0 Pa. The optimizer thins the cross-section to the lower bound. This behavior highlights a limitation of the proposed method in that the stresses at the tip cross-section cannot be captured accurately.

### 4.3 Study 2: static loading weight optimization

Consider sizing the structure for the following conditions:

- Tip load in ‘lift, drag, and moment’ directions of  $F_x = 0$  N,  $F_y = 1 \times 10^5$  N,  $F_z = 2 \times 10^6$  N,  $M_x = 1 \times 10^6$  N m,  $M_y = M_z = 0$  Nm,
- Initial guess to optimizer set to the upper bounds,
- Aluminum material with properties:  $E = 7 \times 10^9$  Pa,  $\rho = 2700$  kg/m<sup>3</sup>,  $\sigma_y = 276 \times 10^6$  Pa.

The following five constraint cases were considered:

1. Corner points axial stress only,
2. Midpoint von Mises stress only,
3. Flange shear stress only,

- 4. Midpoint von Mises stress and flange shear stress,
- 5. Midpoint von Mises stress, corner point axial stress, and flange shear stress.

Figure 4 compares the optimal thicknesses computed by the optimizer. The behavior of the results for the different constraint cases is summarized as follows:

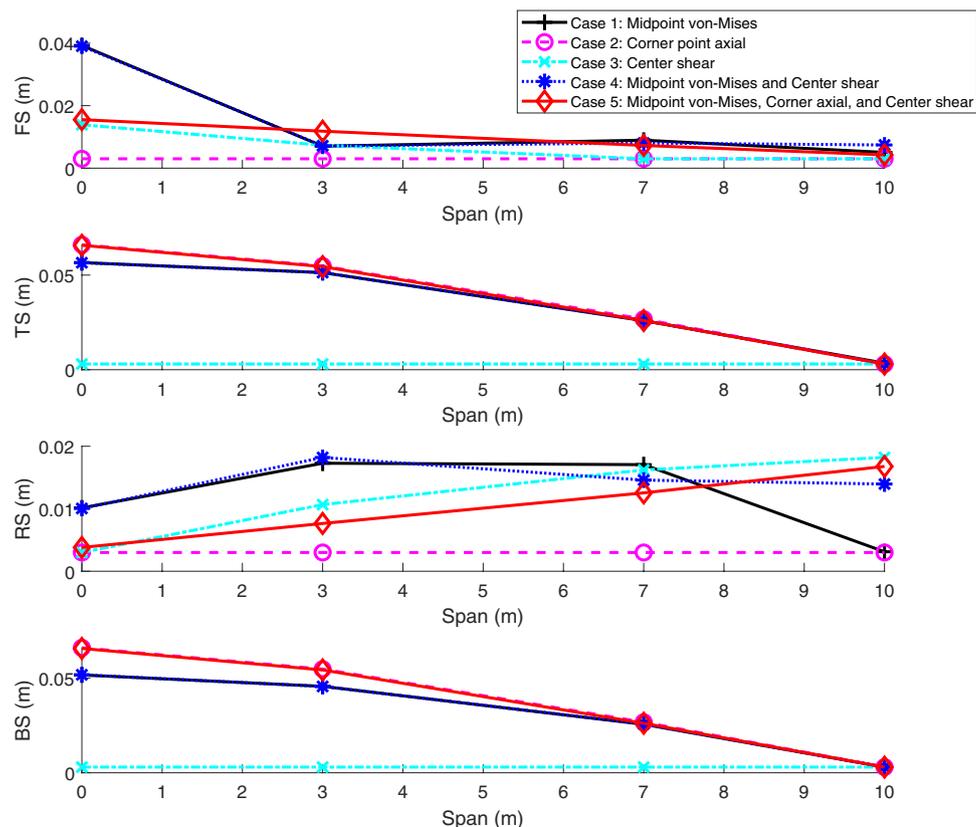
1. Axial stresses only depend on distance from the neutral point. The optimizer favors increasing the top and bottom flange thicknesses, while the left and right webs are at the lower bounds, as their thicknesses do not affect axial stress with the loading profile considered, as  $F_z > F_y$ , while also  $I_{zz} > I_{yy}$ .
2. The top and bottom flanges were still sized by the axial stress component in the von Mises resultant. The shear flow in the right and left webs contribute to their final-sized thicknesses. The internal torsional moment has a large negative value at the root while having a positive value at the tip. This difference in signs and magnitudes sizes the right and left webs differently.
3. If only the flange shear is used to size, the top and bottom flanges are sized to their minimums. This is due to not having to deal with axial stress and dealing with relatively low shear stress due to the cross-sectional inertia

in the top–bottom direction. On the other hand, the webs did get a size increment due to the shear stresses from two sources—the torsional shear and the force shear—which size the right web. Unlike torsional shear, which depends on the torsional loads, shear stresses depend on the shear forces experienced by the beam, which are constant along its span. Also, since the torsion at the tip is nonzero, the optimizer sizes one of the flanges to counter the stresses.

4. When the von Mises corner points include the shear at the center of the flanges, it results in thicker left and right webs with even the tip section sized, resulting in a lesser need for the top–bottom flange thickness.
5. If all constraints are enabled, the beam ends up with the highest weight, and thickness develops along all the different flanges, with an outcome similar to that of Case 4.

Table 4 compares the sized structure for the different constraint cases. As expected, the case that uses all constraints ends up with the highest thickness for all sections, as it takes into account every high point of stress within the beam. Due to width being greater than the height, the thicknesses of the top skin and bottom skin have a larger impact on the total weight than the thicknesses of the front and rear spars.

**Fig. 4** Optimum thickness distribution for different constraints in study 2. FS: front spar, TS: top skin, RS: rear spar, and BS: bottom skin



### 4.4 Study 3: constant load dynamic analysis

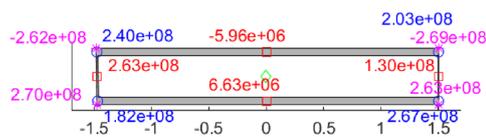
The purpose of this study is to compare the optimized structure obtained from a static analysis to that obtained from a dynamic analysis under identical loads (i.e., a constant tip load). This study is conducted under conditions similar to Study 2. When simulated dynamically, the loading conditions of  $F_x = 0\text{ N}$ ,  $F_y = 1 \times 10^5\text{ N}$ ,  $F_z = 2 \times 10^6\text{ N}$ ,  $M_x = 1 \times 10^6\text{ Nm}$ , and  $M_y = M_z = 0\text{ Nm}$  at the tip are held constant throughout the dynamic analysis. The simulation is run for 5 s with a step size of 0.01 s. Stress constraints are imposed on midpoint von Mises stress, corner point axial stress, and flange shear stress.

First, consider the optimum solution when the simulation is run statically. The stresses on the optimized root cross-section are shown in Fig. 5a. As observed in earlier studies, it can be seen that the maximum stresses are close to, but lower than the yield stress of  $2.76 \times 10^8\text{ Pa}$ .

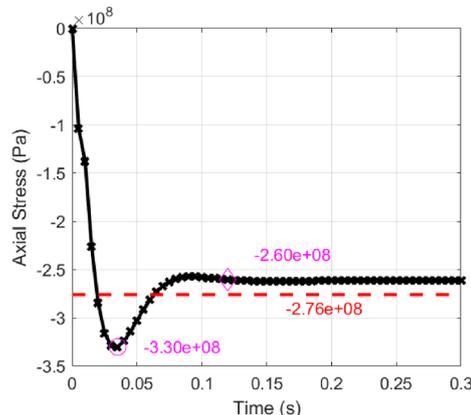
The optimized beam is next subjected to the same constant tip load, but is analyzed dynamically. Figure 5b shows the axial stress time history of the top left corner point. Note that the peak stress of  $-3.27 \times 10^8\text{ Pa}$  is much higher than the allowable yield stress. Also note that as the effects of dynamics damp out, the steady state value of  $-2.62 \times 10^8\text{ Pa}$  is indeed the value seen in Fig. 5a for the same top left corner point.

This simple example illustrates why larger thicknesses are achieved when a dynamic simulation is performed instead of a static simulation under identical loads. The thickness distribution of the optimum solutions is shown in Fig. 6. It can be seen that all four thicknesses for the dynamic analysis have been sized to larger values when compared to a static analysis. Table 5 compares the results of the optimization obtained from static analysis against that obtained from dynamic analysis. It can be seen that the weight increases by 10.8% due to the dynamics.

The breakdown of execution time is reported in Table 5. The generation of the symbolic beam model is a one-time cost. Thereafter, at each iteration of the optimizer, note that the execution times for the forward solve and each gradient computation using the adjoint solver are of the same order of magnitude. The computational time is higher due to the five stress gradient computations. As shown in Table 4 earlier, the optimum point found considering only midpoint von Mises stresses and that found considering all three (von



(a) von-Mises stresses (in Pa, shown in blue) and axial stresses (in Pa, shown in magenta), and shear stresses (Pa, shown in red) for a statically analyzed and optimized beam



(b) Axial stress at top left point for a dynamically analyzed and optimized beam

Fig. 5 Illustration of the effects of dynamic analysis and optimization when initialized with a statically converged optimal

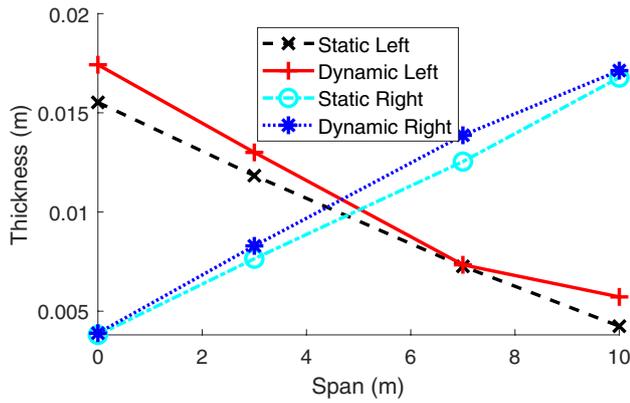
Mises, axial, and shear) produce comparable results. Thus, larger optimization problems considering only the von Mises stress constraint would yield acceptable results with a far lower computational cost (Table 6).

### 4.5 Study 4: gust load dynamic optimization

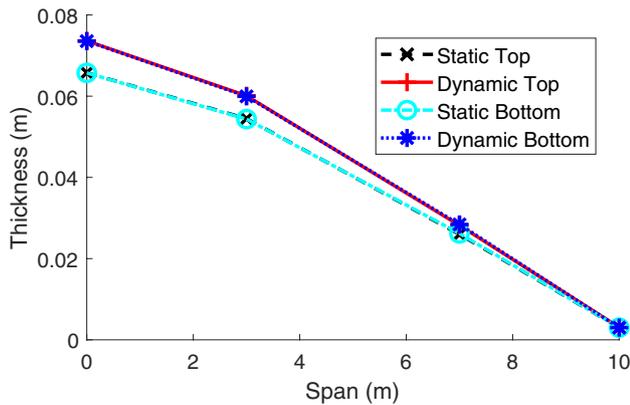
Several representative load profiles could be used to perform a sensible comparison between structures obtained from static versus dynamic sizing and optimization. To construct a meaningful dynamic simulation, a series of 13 gust profiles were simulated by means of a heavy elliptically distributed load, which was scaled by multipliers that mimic the profiles specified in the 14 CFR Part 25 Regulations for gust velocities. A 1-minus-cosine gust is selected as a test case, defined in Eq. 28 as

Table 4 Comparison of structural sizing for different constraints

	Case 1	Case 2	Case 3	Case 4	Case 5
Optimizer iterations	133	44	30	111	87
Optimum weight (kg)	4901.52	5275.87	624.85	4917.23	5386.58
Constraint tolerance	$9.14 \times 10^{-8}$	$9.13 \times 10^{-7}$	$5.14 \times 10^{-7}$	$6.13 \times 10^{-8}$	$8.97 \times 10^{-7}$
Execution time (s)	12.039	6.138	2.955	12.688	12.544



(a) Web thicknesses



(b) Flange thickness

Fig. 6 Optimum solution for study 3

Table 5 Comparison of structural sizing and optimization results from static and dynamic analysis

	Static	Dynamic
Optimization iterations	87	174
Weight (kg)	5386.58	5935.87
Execution time (s)	12.54	335.568

$$U = \frac{U_{ds}}{2} \left( 1 - \cos \frac{\pi V \cdot t}{H} \right) \quad (28)$$

based on a medium regional aircraft, where  $U$  is the multiplier for the loads at time  $t$  of the gust and  $U_{ds}$  is the multiplier for the peak design load being simulated. The value for  $U_{ds}$  is selected from each of the 10 gust profiles, with a maximum value of 1 for the gust corresponding to the highest design gust velocity, and less than 1 for the remaining gusts, sized in accordance with CFR 25.341 gust and turbulence loads. The gust penetration distance  $H$  oscillates from 30 to 350 ft. at an altitude of 30,000 ft. and a velocity

Table 6 Breakdown of computation time of dynamic solver

Task	Time (s)
Generate symbolic beam model	47.53
Time marching forward solve	3.12
von Mises stress gradient	3.90
Compressive axial stress gradient	3.92
Tensile axial stress gradient	3.82
Positive shear stress gradient	4.15
Negative shear stress gradient	4.10

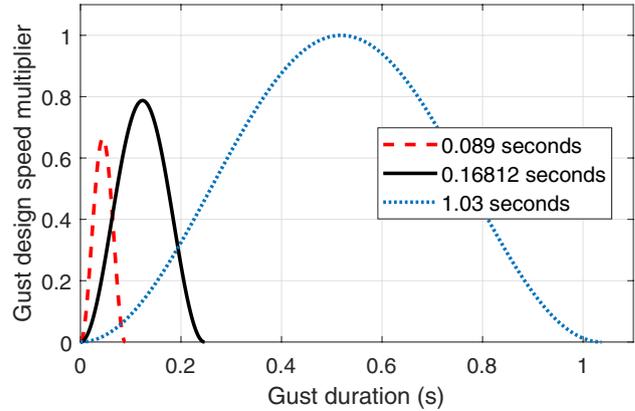


Fig. 7 Simulated gust profiles based on 14 CFR Regulations. Thirteen profiles are simulated from 0.089 to 1.03 s. Only three are shown for clarity

$V$  at Mach 0.6. Figure 7 shows an illustration of the gust profiles vs. time.

These gusts are used to size and optimize an aluminum structure with the method presented in this work. The optimizer used the von Mises aggregated stress constraint to size a box beam with the same geometry as the beam shown in Fig. 2. To illustrate the stress before and after the optimization process, four reference points were tracked, as shown in Fig. 8.

For the shortest period gust, Fig. 9 shows the stress at these points before and after optimization. Before optimization, note that points on the inner sections violate the constraints, whereas points on the outer sections are severely under-loaded. After optimization, smaller oscillations in the stress response to the gust are observed. Note that the maximum stress at points on all four cross-sections is below the yield stress (shown as a horizontal line). Some sections are slightly under-loaded as the thicknesses of the sections reached the lower bounds specified in the optimization problem statement. The optimization results in a mass of 393.6 kg.

Figure 10 illustrates the optimization iteration history. Note that some constraints are violated in the beginning

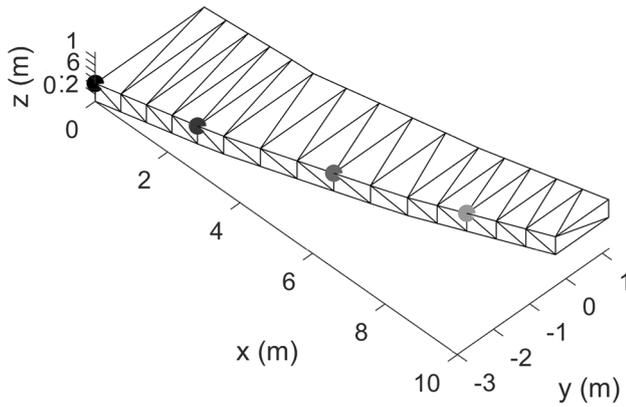


Fig. 8 Reference points tracked during optimization

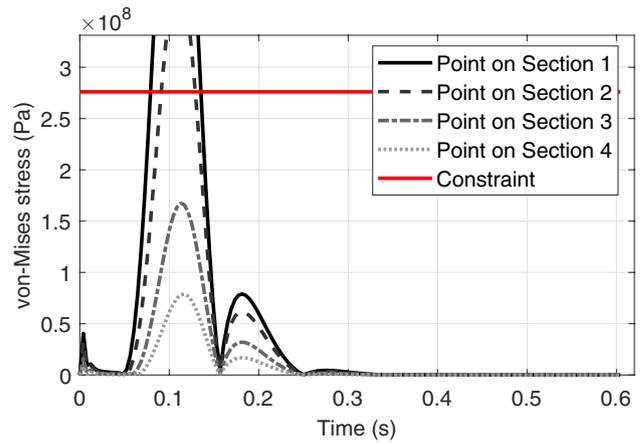
because the optimizer first opts to lower the overall thickness of the beam before focusing on satisfying the constraints, and finally honing in on the optimum.

#### 4.6 Study 5: multiple gust loads sizing and optimization

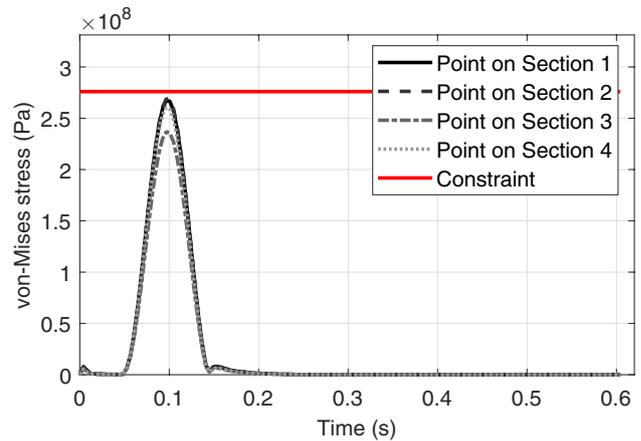
To quantify the impact and consequence of sizing and optimizing a structure using the equivalent static load method rather than considering the dynamics directly, a final optimization study was performed with 13 elliptically distributed load profiles designed to mimic actual gusts in accordance with the 14 CFR Part 25 Regulations. As mentioned earlier, the time variation of gust load magnitudes is modeled using multipliers. Equivalent static loads for the gusts are extracted from the dynamic simulations using a method presented by Park (2011).

The structure was first sized and optimized for the 13 gusts taken one at a time. Figure 11 compares the optimized weight using the equivalent load to that of its dynamic counterpart as a function of the gust period. Note that the equivalent static method under-predicts the optimum mass compared to the dynamic optimization. The equivalent static sizing begins yielding structures with heavier mass as the gust period gets wider, missing the peak that occurs at the gust period of 0.17 s.

Finally, an optimization study was run with all 13 gust load cases as constraints, simultaneously. The dynamic sizing and optimization resulted in a beam with a mass of 400.2 kg, whereas the structure obtained by the aggregated equivalent static analysis resulted in a mass of 388.9 kg. While the overall difference (about 2.8%) in total optimized weight between both approaches is small, it is mainly due to the consideration of the longer gust periods. When the optimized design produced by the equivalent load method was subjected to dynamic analysis with the gust loads,



(a) von-Mises stress before optimization



(b) von-Mises stress after optimization

Fig. 9 von Mises stress before and after optimization for gust period 0.16 s

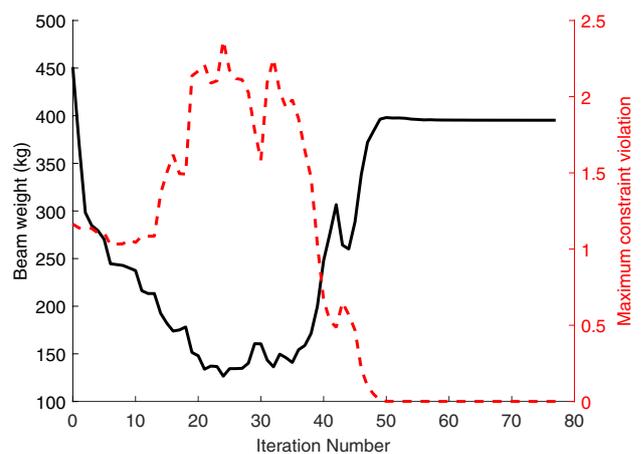
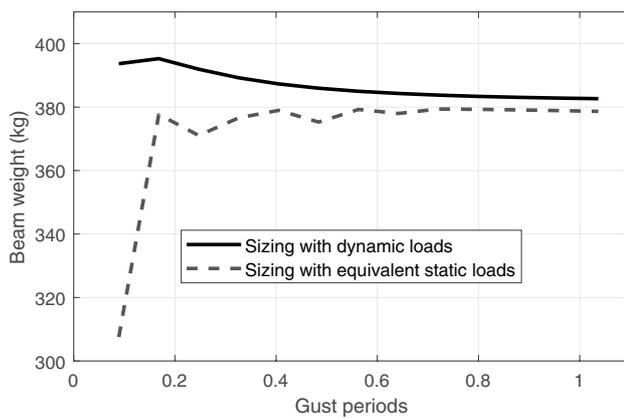


Fig. 10 Optimization history for 0.16-s dynamic gust sizing



**Fig. 11** Dynamic vs. equivalent static sizing of different gust profiles

the structure did not satisfy the yield constraint. This was particularly pronounced when subjected to the 0.17-s gust period.

## 5 Conclusion and future work

This work presented the development and implementation of an equation and cross-section agnostic beam element-based dynamic structural analysis and optimization framework with a special emphasis on enabling early-stage structural design and optimization. A general adjoint solver was implemented for the backwards difference formula scheme. The framework utilizes a hybrid approach leveraging a mix of symbolic expressions and AD that provides a convenient way to consider arbitrary cross-section shapes, use physics-based stress recovery formulae at the desired level of fidelity, and add an arbitrary points number of points on the cross-section to recover stresses. The framework efficiently computes gradients using the adjoint method, thus allowing for rapid static and dynamic sizing and gradient-based structural optimization using any publicly available optimization algorithm. Finally, this work demonstrated aggregation of stress constraints in both space and time.

The gradient computed by the adjoint-based implementation was compared and validated against finite differences. Study 1 demonstrated the ability of the framework to minimize the structural weight for a static tip load. Study 2 highlighted the effects of different types of stresses on the design variables by comparing optimal solutions obtained by considering different types of stresses in the optimization problem. Studies 3, 4, and 5 explicitly investigated the ramifications of neglecting dynamics by posing and solving a series of equivalent static and dynamic optimization problems. For a realistic gust load sizing scenario, study 5 revealed a difference in the optimal weight when the time history of loads was considered as opposed to the optimal

weight obtained by considering a steady state static load or a sequence of equivalent static loads.

A particular consequence of utilizing the beam theory and associated stress recovery formulae is the inability to accurately compute the stresses at the tip cross-section. Future work will include further accuracy enhancements to the stiffness and stress recovery computations using the VAM. Other improvements will involve the implementation of constraints important to aerospace applications, such as deflection and buckling constraints.

## Appendix A: Analytical equations of a box cross-section beam

In this work, we demonstrate the capability on a structure that is ‘wing-like,’ as seen in Fig. 2. The geometry has typical aircraft wing planform features such as taper, dihedral, and sweep. Each cross-section is a hollow rectangle. It should be noted that the geometry used here has a lower aspect ratio than most commercial aircraft. Beam assumptions generally hold better for high aspect ratio structures.

### Objective function

The objective function is the total wing weight, which is given by the following equations:

$$f_m = \sum_{i=1}^{N-1} m_i, \quad (29)$$

$$m_i = \rho V_i \longrightarrow V_i = \frac{1}{3} s_i (A_{i+1} + A_i + \sqrt{A_i A_{i+1}}). \quad (30)$$

### Cross-sectional properties

In general, the beam axis can be chosen to be at any location within the cross-section. This makes the beam axis location potentially different from the elastic axis, the tension axis, and the geometric CG as shown in Fig. 14. The design variables of each box cross-section  $i$  of the beam are shown in Fig. 12 and can be defined by the following quantities:

- $h_i$ : height,
- $w_i$ : width,
- $t_{\text{top},i}$ : top thickness,
- $t_{\text{bot},i}$ : bottom thickness,
- $t_{\text{right},i}$ : right thickness,
- $t_{\text{left},i}$ : left thickness.

In general, the beam axis can be chosen to be at any location within the cross-section. This makes the beam axis location

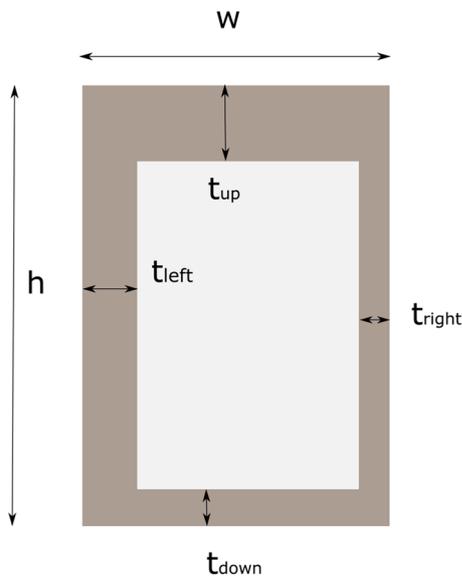


Fig. 12 Rectangular box cross-section assumption and parameters

potentially different from the elastic axis, the tension axis, and the geometric CG. Such a scenario for an airfoil cross-section is shown in Fig. 13.

We use the notation  $(\_)_{j,i}$  to denote a property at cross-sectional  $i$  and component  $j$  can be top, left, bottom, or right corresponding to 1, 2, 3, or 4 respectively. The area of the cross-section in inertial  $XYZ$  frame is given by

$$A_i = A_{1,i} + A_{2,i} + A_{3,i} + A_{4,i}, \tag{31}$$

where

$$A_{1,i} = t_{top,i} W_i, \tag{32}$$

$$A_{2,i} = t_{left,i} (h_i - t_{top,i} - t_{bot,i}), \tag{33}$$

$$A_{3,i} = t_{bot,i} W_i, \tag{34}$$

$$A_{4,i} = t_{right,i} (h_i - t_{top,i} - t_{bot,i}). \tag{35}$$

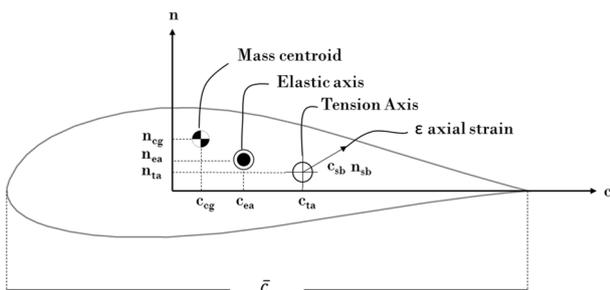


Fig. 13 Cross-section properties. Figure based on Drela (1999)

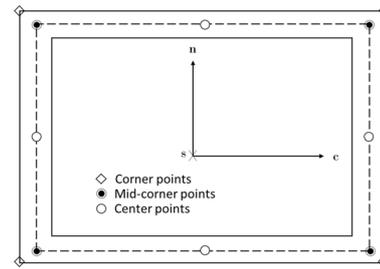


Fig. 14 Points in each beam cross-section where stresses are computed

The area moments of inertia in the inertial  $XYZ$  frame are given by

$$I_{xx,i} = \sum_{j=1}^4 I_{xx,j,i} + A_i (x_{i,j,x} - c_{ea,x,i})^2, \tag{36}$$

$$I_{zz,i} = \sum_{j=1}^4 I_{zz,j,i} + A_i (x_{i,j,z} - n_{ea,z,i})^2, \tag{37}$$

$$I_{xz,i} = \sum_{j=1}^4 -A_i (x_{i,j,z} - n_{ea,z,i})(x_{i,j,x} - c_{ea,x,i}), \tag{38}$$

where the terms of the top flange are given by

$$I_{xx,1,i} = \frac{w_i t_{top,i}^3}{12}, \tag{39}$$

$$I_{zz,1,i} = \frac{w_i^3 t_{top,i}}{12}, \tag{40}$$

$$x_{i,1,z} = \frac{h_i - t_{top,i}}{2}, \tag{41}$$

$$x_{i,1,x} = 0. \tag{42}$$

The terms of the left web are given by

$$I_{xx,2,i} = \frac{t_{left,i} (h_i - t_{top,i} - t_{bot,i})^3}{12}, \tag{43}$$

$$I_{zz,2,i} = \frac{t_{left,i}^3 (h_i - t_{top,i} - t_{bot,i})}{12}, \tag{44}$$

$$x_{i,2,z} = \frac{h}{2} - t_{top,i} - \frac{h_i - t_{top,i} - t_{bot,i}}{2} \tag{45}$$

$$x_{i,2,x} = \frac{t_{\text{left},i} - w_i}{2}. \tag{46}$$

The terms of the bottom flange are given by

$$I_{xx,3,i} = \frac{w_i t_{\text{bot},i}^3}{12}, \tag{47}$$

$$I_{zz,3,i} = \frac{w_i^3 t_{\text{bot},i}}{12}, \tag{48}$$

$$x_{i,3,z} = \frac{t_{\text{bot},i} - h_i}{2}, \tag{49}$$

$$x_{i,3,x} = 0. \tag{50}$$

The terms of the right web are given by

$$I_{xx,4,i} = \frac{t_{\text{right},i}(h_i - t_{\text{top},i} - t_{\text{bot},i})^3}{12}, \tag{51}$$

$$I_{zz,4,i} = \frac{t_{\text{right},i}^3(h_i - t_{\text{top},i} - t_{\text{bot},i})}{12}, \tag{52}$$

$$x_{i,4,z} = \frac{h}{2} - t_{\text{top},i} - \frac{h_i - t_{\text{top},i} - t_{\text{bot},i}}{2}, \tag{53}$$

$$x_{i,4,x} = \frac{w_i - t_{\text{right},i}}{2}. \tag{54}$$

The modulus weighted centroid is used to compute the elastic axis. We make the assumption here that the tension axis coincides with the beam axis.

$$c_{\text{ea},x,i} = \frac{\sum_{j=1}^4 A_j E_j x_{i,j,x}}{\sum_{j=1}^4 A_j E_j}, \tag{55}$$

$$n_{\text{ea},z,i} = \frac{\sum_{j=1}^4 A_j E_j x_{i,j,z}}{\sum_{j=1}^4 A_j E_j}, \tag{56}$$

$$c_{\text{ta},x,i} = 0, \tag{57}$$

$$n_{\text{ta},z,i} = 0. \tag{58}$$

The torsional stiffness is given by

$$J_i = \frac{2 \left( \left( h_i - \frac{t_{\text{top},i}}{2} - \frac{t_{\text{bot},i}}{2} \right) \left( w_i - \frac{t_{\text{right},i}}{2} - \frac{t_{\text{left},i}}{2} \right) \right)^2}{\left( \frac{w_i - \frac{t_{\text{right},i}}{2} - \frac{t_{\text{left},i}}{2}}{\left( \frac{t_{\text{top},i}}{2} + \frac{t_{\text{bot},i}}{2} \right)} + \frac{h_i - \frac{t_{\text{top},i}}{2} - \frac{t_{\text{bot},i}}{2}}{\left( \frac{t_{\text{right},i}}{2} + \frac{t_{\text{left},i}}{2} \right)} \right)}. \tag{59}$$

The moment/curvature stiffness submatrix in the inertial XYZ frame is given by

$$\underline{\underline{EI}}_{XYZ,i} = \begin{bmatrix} E_i I_{xx,i} & 0 & E_i I_{xz,i} \\ 0 & G_i J_i & 0 \\ E_i I_{xz,i} & 0 & E_i I_{zz,i} \end{bmatrix}. \tag{60}$$

It is transformed into the local csn frame as

$$\underline{\underline{EI}}_{\text{csn}} = [T] \underline{\underline{EI}}_{XYZ} [T]^T. \tag{61}$$

For shear stress computation, the stiffness is given by

$$Q_{x,i} = w_i t_{\text{top},i} \left( \frac{h_i}{2} - n_{\text{ea},z,i} - \frac{t_{\text{top},i}}{2} \right) + \frac{1}{2} t_{\text{right},i} \left( \frac{h_i}{2} - n_{\text{ea},z,i} - t_{\text{top},i} \right)^2 + \frac{1}{2} t_{\text{left},i} \left( \frac{h_i}{2} - n_{\text{ea},z,i} - t_{\text{top},i} \right)^2, \tag{62}$$

$$Q_{z,i} = h_i t_{\text{left},i} \left( \frac{w_i}{2} - c_{\text{ea},x,i} - \frac{t_{\text{left},i}}{2} \right) + \frac{1}{2} t_{\text{top},i} \left( \frac{w_i}{2} - c_{\text{ea},x,i} - t_{\text{left},i} \right)^2 + \frac{1}{2} t_{\text{bot},i} \left( \frac{w_i}{2} - c_{\text{ea},x,i} - t_{\text{left},i} \right)^2. \tag{63}$$

The center of gravity of the cross-section is given by

$$r_{\text{cg},c,i} = \frac{\sum_{j=1}^4 A_j \rho_j x_{i,j,x}}{\sum_{j=1}^4 A_j \rho_j}, \tag{64}$$

$$r_{\text{cg},n,i} = \frac{\sum_{i=1}^4 A_i \rho_i x_{i,j,z}}{\sum_{j=1}^4 A_j \rho_j}. \tag{65}$$

### Stress recovery

The beam theory tracks the states defined in Table 1. For node  $i = 1, \dots, N$  and time  $t = t_0, \dots, t_{\text{final}}$ , the following information is obtained:

$$\mathbf{x}(i, t) = \begin{bmatrix} \mathbf{r}(i, t) \\ \boldsymbol{\theta}(i, t) \\ \mathbf{F}(i, t) \\ \mathbf{M}(i, t) \\ \dot{\mathbf{r}}(i, t) \\ \dot{\boldsymbol{\theta}}(i, t) \end{bmatrix}. \tag{66}$$

This resultant state vector is obtained in the inertial XYZ frame. From the state vector, the set of internal forces and moments can be computed in the csn frame by performing the following transformation:

$$\mathbf{M}_{\text{csn}} = [T]\mathbf{M}_{\text{XYZ}}, \tag{67}$$

$$\mathbf{F}_{\text{csn}} = [T]\mathbf{F}_{\text{XYZ}}, \tag{68}$$

where

$$\mathbf{M}_{\text{csn}} = \begin{bmatrix} M_c(i, t) \\ M_s(i, t) \\ M_n(i, t) \end{bmatrix}, \quad \mathbf{F}_{\text{csn}} = \begin{bmatrix} F_c(i, t) \\ F_s(i, t) \\ F_n(i, t) \end{bmatrix}. \tag{69}$$

The internal forces and moments at each cross-section are used to compute the state of stress in the beam. At each cross-section, points  $j = 1, \dots, 12$  are considered as shown in Fig. 14. The locations of these points,  $x_{c,j}$  and  $x_{n,j}$  along the local c and n directions, respectively, are specified as inputs to the simulation, and are directly dependent on the design variables shown in Fig. 12.

Given a cross-section  $i$ , the equations to compute the beam state of stress are described next. At the four corner points, the axial stress,  $\sigma_{ss}$  for a corner point  $j$  at time  $t$  is computed by

$$\sigma_{\text{ss}_{i,j,t}}^{\text{corner}} = \frac{N_{i,t}}{A_i} - x_{n,j} \frac{M_{c,i,t}}{EI_{cc_i}} + x_{c,j} \frac{M_{n,i,t}}{EI_{nn_i}}. \tag{70}$$

The corner points, however, do not consider the torsional stresses due to the shear flow resulting from torsion within the beam. For that, given a mid-line and assuming a closed thin-walled section, the shear flow and torsional shear can be assumed to be constant and equal to

$$q_{i,t} = \frac{T_{i,t}}{\bar{A}} \longrightarrow \tau_{s_{i,j,t}}^{\text{mid}} = \frac{T_{i,t}}{2\bar{A}t_{i,j}}. \tag{71}$$

For the mid-corner points, a resultant shear is calculated by computing the von Mises resultant shear flow of both flanges that intersect the corner:

$$(\tau_{r_{i,j,t}}^{\text{mid}})^2 = (\tau_{s_{i,p,t}}^{\text{mid}})^2 + (\tau_{s_{i,q,t}}^{\text{mid}})^2 - \tau_{s_{i,p,t}}^{\text{mid}} \tau_{s_{i,q,t}}^{\text{mid}}, \tag{72}$$

where  $p$  and  $q$  are flanges that intersect at the corner  $j$ . Next, an axial stress resultant is computed, taking into account the position of the corner points with respect to the neutral axis:

$$\sigma_{\text{ss}_{i,j,t}}^{\text{mid}} = \frac{N_{i,t}}{A_i} - \left(x_{n,j} - \frac{t_{i,p}}{2}\right) \frac{Mc_{i,t}}{EI_{cc_i}} + \left(x_{c,j} - \frac{t_{i,q}}{2}\right) \frac{Mn_{i,t}}{EI_{nn_i}}, \tag{73}$$

where  $p$  and  $q$  are flanges that intersect at the corner  $j$ . After both the shear and normal stresses are computed, a net stress is obtained by generating a von Mises resultant at the corners, since the torsion and normal stress are perpendicular to each other:

$$(\sigma_{\text{vm}_{i,j,t}}^{\text{mid}})^2 = (\sigma_{\text{ss}_{i,j,t}}^{\text{mid}})^2 + (\tau_{r_{i,j,t}}^{\text{mid}})^2 - \sigma_{\text{ss}_{i,j,t}}^{\text{mid}} \tau_{r_{i,j,t}}^{\text{mid}}. \tag{74}$$

At the center of the flanges, the loads are assumed to be mostly composed of shear loads. These points were set up in place to capture beam stresses that lie along or close to the neutral axis of the beam. A shear flow through a cross-section of total thickness  $t$  is calculated by

$$q = \tau t \rightarrow \tau = \frac{VQ}{It}, \tag{75}$$

$$\tau_{c_{i,t}} = \frac{V_{c_{i,t}} Q_{n_i}}{I_{nn_i}(t_{\text{top},i} + t_{\text{bot},i})}, \tag{76}$$

$$\tau_{n_{i,t}} = \frac{V_{n_{i,t}} Q_{c_i}}{I_{cc_i}(t_{\text{right},i} + t_{\text{left},i})}. \tag{77}$$

The shear flow occurs due to torsional and transverse loads. The shear stress is calculated by combining torsional and force shear stresses specific to each flanges direction:

$$\tau_{r_{c,i,t}} = \tau_{s_{c,i,t}} + \tau_{c_{i,t}}, \tag{78}$$

$$\tau_{r_{n,i,t}} = \tau_{s_{n,i,t}} + \tau_{n_{i,t}}. \tag{79}$$

### Appendix B: Forward and adjoint solver

Consider the beam governing equations written in residual form  $\mathbf{R} = \mathbf{0}$ . The solver will attempt to find the values of the unknowns (elements of the state vector  $\mathbf{x}$ ) by driving the residuals of nonlinear system of equations to zero. To start the time marching, the states from the trim solution can be given as the initial conditions. The residual system is solved using Newton iterations. The pseudo-code for the solver is given in Algorithm 1. The required symbolic functions for the Jacobians are given in Eqs. 24 and 25.

**Algorithm 1:** Forward Solver: Dynamic Simulation using BDF scheme

```

Input: Symbolic functions:  $\frac{\text{sym } \partial \mathbf{R}}{F \partial \mathbf{x}}, \frac{\text{sym } \partial \mathbf{R}}{F \partial \dot{\mathbf{x}}}$ 
Input: Design variables:  $\mathbf{cs}_{\text{dv},i} \forall i = 1, \dots, N$ 
1 while  $t < t_{\text{final}}$  do
2    $t = t + \Delta t$ 
3   Guess from previous time-step
4    $\mathbf{x}_t = \mathbf{x}_{t-1}$   $\dot{\mathbf{x}}_t = \mathbf{0}$ 
5   BDF coefficients  $k_0 = \frac{1.5}{\Delta t}, k_1 = \frac{-2}{\Delta t}$ 
6   Convergence Flag = False
7    $\mathbf{x} = \mathbf{x}_t, \dot{\mathbf{x}} = \dot{\mathbf{x}}_t$ 
8   while Convergence Flag  $\neq$  True do
9     Compute residual vector
10     $\mathbf{r} = \frac{\text{sym } \partial \mathbf{R}}{F}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{cs}_{\text{dv}})$ 
11    Compute Jacobians
12     $\frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \frac{\text{sym } \partial \mathbf{R}}{F \partial \mathbf{x}}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{cs}_{\text{dv},i} \forall i)$ 
13     $\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} = \frac{\text{sym } \partial \mathbf{R}}{F \partial \dot{\mathbf{x}}}(\mathbf{x}, \dot{\mathbf{x}}, \mathbf{cs}_{\text{dv},i} \forall i)$ 
14    Solve linear system
15     $[\frac{\partial \mathbf{R}}{\partial \mathbf{x}} + k_0 \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}}] \delta \mathbf{x} = -\mathbf{r}$ 
16    if  $\|\mathbf{r}\|_2 < \epsilon$  then
17      Convergence Flag = True
18    end
19    else
20      Guess for next iteration
21       $\mathbf{x} = \mathbf{x} + \delta \mathbf{x}$   $\dot{\mathbf{x}} = k_0 \mathbf{x} + k_1 \mathbf{x}_{t-1}$ 
22    end
23  end
24   $\mathbf{x}_t = \mathbf{x}, \dot{\mathbf{x}}_t = \dot{\mathbf{x}}$ 
25 end
Output:  $\mathbf{x}_t, \dot{\mathbf{x}}_t \forall t = 0, \dots, T$ 

```

The forward solver provides the time history of the state variables  $\mathbf{x}_t, \dot{\mathbf{x}}_t \forall t = 0, \dots, T$ . The adjoint solver uses the time history to obtain the required gradient of a function of interest. The pseudo-code for the solver is given in Algorithm 2. It should be noted that the function *AdjointSolver()* uses the entire time history of the solution as seen on lines 4 and 5. The function *GradientComputation()* only uses the state at a given time step.

In this work, the adjoint solver is used to obtain the gradient of the aggregated stresses  $F = c_{\text{ks},\sigma_{\text{vm}}}$  with respect to the cross-sectional design variables  $\boldsymbol{\mu} = \mathbf{cs}_{\text{dv},i}$ .

**Algorithm 2:** Gradient using Adjoint Solver

```

1 Function AdjointSolver()
2   Input: Forward solver solution:
3      $\mathbf{x}_t, \dot{\mathbf{x}}_t \forall t = 0, \dots, T$ 
4   Input: Design variables:
5      $\mathbf{cs}_{\text{dv},i} \forall i = 1, \dots, N$ 
6   Input: Symbolic functions:  $\frac{\text{sym } \partial \mathbf{R}}{F \partial \mathbf{x}}, \frac{\text{sym } \partial \mathbf{R}}{F \partial \dot{\mathbf{x}}}, \frac{\text{sym } \partial F}{F \partial \mathbf{x}}, \frac{\text{sym } \partial F}{F \partial \dot{\mathbf{x}}}$ 
7   for  $t \leftarrow t_{\text{final}}$  to 0 do
8     BDF coefficients  $k_0 = \frac{1.5}{\Delta t},$ 
9      $k_1 = \frac{-2}{\Delta t}, k_2 = \frac{0..5}{\Delta t}$ 
10    Compute Jacobians
11     $\frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \frac{\text{sym } \partial \mathbf{R}}{F \partial \mathbf{x}}(\mathbf{x}_t \& \dot{\mathbf{x}}_t \forall t, \mathbf{cs}_{\text{dv},i} \forall i)$ 
12     $\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} = \frac{\text{sym } \partial \mathbf{R}}{F \partial \dot{\mathbf{x}}}(\mathbf{x}_t \& \dot{\mathbf{x}}_t \forall t, \mathbf{cs}_{\text{dv},i} \forall i)$ 
13    Compute partials of F
14     $\frac{\partial F}{\partial \mathbf{x}} = \frac{\text{sym } \partial F}{F \partial \mathbf{x}}(\mathbf{x}_t \& \dot{\mathbf{x}}_t \forall t, \mathbf{cs}_{\text{dv},i} \forall i)$ 
15     $\frac{\partial F}{\partial \dot{\mathbf{x}}} = \frac{\text{sym } \partial F}{F \partial \dot{\mathbf{x}}}(\mathbf{x}_t, \dot{\mathbf{x}}_t \forall t, \mathbf{cs}_{\text{dv},i} \forall i)$ 
16    Create linear system
17     $A = [\frac{\partial \mathbf{R}}{\partial \mathbf{x}} + k_0 \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}}]$ 
18     $b = k_0 \frac{\partial F}{\partial \dot{\mathbf{x}}} + \frac{\partial F}{\partial \mathbf{x}} + k_1 (\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \lambda_{t+1}) +$ 
19     $k_2 (\frac{\partial \mathbf{R}}{\partial \dot{\mathbf{x}}} \lambda_{t+2})$ 
20    Solve linear system  $A \lambda_t = b$ 
21  end
22 Output:  $\lambda_t \forall t = 0, \dots, T$ 
23
24 Function GradientComputation()
25 Input: Adjoint solution:
26    $\lambda_t \forall t = 0, \dots, T$ 
27 Input: Forward solver solution:
28    $\mathbf{x}_t, \dot{\mathbf{x}}_t \forall t = 0, \dots, T$ 
29 Input: Symbolic functions:  $\frac{\text{sym } \partial F}{F \partial \boldsymbol{\mu}}, \frac{\text{sym } \partial \mathbf{R}}{F \partial \boldsymbol{\mu}}$ 
30  $\frac{\partial F}{\partial \boldsymbol{\mu}} = 0, \frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}} = 0$ 
31 for  $t \leftarrow 0$  to  $t_{\text{final}}$  do
32    $\frac{\partial F}{\partial \boldsymbol{\mu}} =$ 
33    $\frac{\partial F}{\partial \boldsymbol{\mu}} + \Delta t \frac{\text{sym } \partial F}{F \partial \boldsymbol{\mu}}(\mathbf{x}_t, \dot{\mathbf{x}}_t, \mathbf{cs}_{\text{dv},i} \forall i)$ 
34    $\frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}} = \frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}} +$ 
35    $\Delta t \frac{\text{sym } \partial \mathbf{R}}{F \partial \boldsymbol{\mu}}(\mathbf{x}_t, \dot{\mathbf{x}}_t, \mathbf{cs}_{\text{dv},i} \forall i) \lambda_t$ 
36 end
37 Compute gradient  $\frac{dF}{d\boldsymbol{\mu}} = \frac{\partial F}{\partial \boldsymbol{\mu}} + \frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}}$ 
38 Output:  $\frac{dF}{d\boldsymbol{\mu}}$ 

```

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s00158-021-03141-5>.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

**Replication of results** The following codes are made available with this paper:

– MATLAB code for the equation-agnostic forward solver (Algorithm 1) and adjoint solver (Algorithm 2).

– Spring–mass–damper illustrative example to validate the solvers written using the CasADi symbolic framework.

If the information provided in the paper is not sufficient, interested readers are welcome to contact the authors for further explanations.

## References

- Andersson J, Åkesson J, Diehl M (2012) CasADI: a symbolic package for automatic differentiation and optimal control. In: Recent advances in algorithmic differentiation. Springer, pp 297–307. [https://doi.org/10.1007/978-3-642-30023-3\\_27](https://doi.org/10.1007/978-3-642-30023-3_27)
- Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. *J Mach Learn Res* 18(153):1–43. <https://www.jmlr.org/papers/volume18/17-468/17-468.pdf>
- Blasques JPAA (2012) User's Manual for BECAS: a cross section analysis tool for anisotropic and inhomogeneous beam sections of arbitrary geometry
- Boopathy K, Kennedy G (2017) Adjoint-based derivative evaluation methods for flexible multibody systems with rotorcraft applications. In: 55th AIAA aerospace sciences meeting, p 1671. <https://doi.org/10.2514/6.2017-1671>
- Boopathy K, Kennedy GJ (2019) Parallel finite element framework for rotorcraft multibody dynamics and discrete adjoint sensitivities. *AIAA J* 57(8):3159–3172. <https://doi.org/10.2514/1.J056585>
- Cavagna L, Ricci S, Riccobene L (2009) A fast tool for structural sizing, aeroelastic analysis and optimization in aircraft conceptual design. In: 50th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, 17th AIAA/ASME/AHS adaptive structures conference, 11th AIAA no. p 2571. <https://doi.org/10.2514/6.2009-2571>
- Cavagna L, Ricci S, Travaglini L (2011) NeoCASS: an integrated tool for structural sizing, aeroelastic analysis and MDO at conceptual design level. *Prog Aerosp Sci* 47(8):621–635. <https://doi.org/10.1016/j.paerosci.2011.08.006>
- Cesnik CE, Hodges DH (1997) VABS: a new concept for composite rotor blade cross-sectional modeling. *J Am Helicopter Soc* 42(1):27–38. <https://doi.org/10.4050/JAHS.42.27>
- Cesnik CE, Senatore PJ, Su W, Atkins EM, Shearer CM (2012) X-HALE: a very flexible unmanned aerial vehicle for nonlinear aeroelastic tests. *AIAA J* 50(12):2820–2833. <https://doi.org/10.2514/1.J051392>
- Chauhan SS, Martins JRRR (2018) Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using OpenAeroStruct. In: Proceedings of the 6th international conference on engineering optimization, EngOpt 2018. Springer, Lisbon, pp 418–431
- Curtiss CF, Hirschfelder JO (1952) Integration of stiff equations. *Proc Natl Acad Sci USA* 38(3):235. <https://doi.org/10.1073/pnas.38.3.235>
- Dharmasaroja A, Armstrong C, Murphy A, Robinson T, McGuinness S, Iorga N, Barron J (2017) Load case characterization for the aircraft structural design process. *AIAA J* 55(8):2783–2792. <https://doi.org/10.2514/1.J055544>
- Drela M (1990) Method for simultaneous wing aerodynamic and structural load prediction. *J Aircr* 27(8):692–699. <https://doi.org/10.2514/3.25342>
- Drela M (1999) Integrated simulation model for preliminary aerodynamic, structural, and control-law design of aircraft. In: 40th Structures, structural dynamics, and materials conference and exhibit, p 1394. <https://doi.org/10.2514/6.1999-1394>
- FAA (2019) Code of Federal Regulations, Title 14, Federal Aviation Regulations, Part 25—Airworthiness Standards: Transport Category Airplanes. Federal Aviation Administration
- Hodges DH (2006) Nonlinear composite beam theory. American Institute of Aeronautics and Astronautics
- Hodges D, Yu W (2007) A rigorous, engineer-friendly approach for modelling realistic, composite rotor blades. *Wind Energy Int J Prog Appl Wind Power Convers Technol* 10(2):179–193. <https://doi.org/10.1002/we.215>
- Hurty WC (1965) Dynamic analysis of structural systems using component modes. *AIAA J* 3(4):678–685. <https://doi.org/10.2514/3.2947>
- Jasa JP, Hwang JT, Martins JR (2018) Open-source coupled aerostructural optimization using Python. *Struct Multidisc Optim* 57(4):1815–1827. <https://doi.org/10.1007/s00158-018-1912-8>
- Kang BS, Park G, Arora J (2005) Optimization of flexible multibody dynamic systems using the equivalent static load method. *AIAA J* 43(4):846–852. <https://doi.org/10.2514/1.4294>
- Karpel M (1999) Reduced-order models for integrated aeroservoelastic optimization. *J Aircr* 36(1):146–155. <https://doi.org/10.2514/2.2420>
- Karpel M, Moulin B, Anguita L, Maderuelo C, Climent H (2004) Aeroservoelastic gust response analysis for the design of transport aircrafts. In: 45th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, p 1592. <https://doi.org/10.2514/6.2004-1592>
- Karpel M, Moulin B, Chen P (2005) Dynamic response of aeroservoelastic systems to gust excitation. *J Aircr* 42(5):1264–1272. <https://doi.org/10.2514/1.6678>
- Kennedy GJ, Hicken JE (2015) Improved constraint-aggregation methods. *Comput Methods Appl Mech Eng* 289:332–354. <https://doi.org/10.1016/j.cma.2015.02.017>
- Kennedy GJ, Martins JRRR (2010) Parallel solution methods for aerostructural analysis and design optimization. In: Proceedings of the 13th AIAA/ISSMO multidisciplinary analysis optimization conference, Fort Worth, TX. <https://doi.org/10.2514/6.2010-9308>
- Kennedy GJ, Martins JRRR (2013) An adjoint-based derivative evaluation method for time-dependent aeroelastic optimization of flexible aircraft. In: Proceedings of the 54th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Boston, MA. <https://doi.org/10.2514/6.2013-1530>
- Kennedy GJ, Martins JRRR (2014a) A parallel aerostructural optimization framework for aircraft design studies. *Struct Multidisc Optim*. <https://doi.org/10.1007/s00158-014-1108-9>
- Kennedy GJ, Martins JR (2014b) A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures. *Finite Elem Anal Des* 87:56–73. <https://doi.org/10.1016/j.finel.2014.04.011>
- Kennedy GJ, Martins JRRR, Hansen JS (2008) Aerostructural optimization of aircraft structures using asymmetric subspace optimization. In: 12th AIAA/ISSMO multidisciplinary analysis and optimization conference, Victoria, BC. AIAA. <https://doi.org/10.2514/6.2008-5847>
- Kenway GKW, Kennedy GJ, Martins JRRR (2012) A scalable parallel approach for high-fidelity aerostructural analysis and optimization. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, Honolulu, HI. <https://doi.org/10.2514/6.2012-1922>

- Kenway GW, Kennedy GJ, Martins JRR (2014) Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. *AIAA J* 52:935–951. <https://doi.org/10.2514/1.J052255>
- Kim YI, Park GJ (2010) Nonlinear dynamic response structural optimization using equivalent static loads. *Comput Methods Appl Mech Eng* 199(9–12):660–676. <https://doi.org/10.1016/j.cma.2009.10.014>
- Ku J, Volovoi V, Hodges D (2007) Multilevel-multiphase optimization of composite rotor blade with surrogate model. In: 48th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics, and materials conference, p 1900. <https://doi.org/10.2514/6.2007-1900>
- Lambe A, Martins J (2012a) Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Struct Multidisc Optim*. <https://doi.org/10.1007/s00158-012-0763-y>
- Lambe AB, Martins JRR (2012b) Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Struct Multidisc Optim* 46(2):273–284. <https://doi.org/10.1007/s00158-012-0763-y>
- Lee D, Hodges DH, Patil MJ (2002) Multi-flexible-body dynamic analysis of horizontal axis wind turbines. *Wind Energy Int J Prog Appl Wind Power Convers Technol* 5(4):281–300. <https://doi.org/10.1002/we.66>
- Li L, Volovoi VV, Hodges DH (2008) Cross-sectional design of composite rotor blades. *J Am Helicopter Soc* 53(3):240–251
- Margossian CC (2019) A review of automatic differentiation and its efficient implementation. *Wiley Interdiscip Rev Data Min Knowl Discov* 9(4):e1305. <https://doi.org/10.1002/widm.1305>
- Martins J, Poon NM (2005) On structural optimization using constraint aggregation. In: VI world congress on structural and multidisciplinary optimization WCSMO6, Rio de Janeiro, Brasil. Citeseer
- Meirovitch L, Tuzcu I (2004) Time simulations of the response of maneuvering flexible aircraft. *J Guid Control Dyn* 27(5):814–828. <https://doi.org/10.2514/1.2392>
- Moulin B, Karpel M (2007) Gust loads alleviation using special control surfaces. *J Aircr* 44(1):17–25. <https://doi.org/10.2514/1.19876>
- Nguyen N (2008) Integrated flight dynamic modeling of flexible aircraft with inertial force–propulsion–aeroelastic coupling. In: 46th AIAA aerospace sciences meeting and exhibit, p 194. <https://doi.org/10.2514/6.2008-194>
- Nguyen N, Tuzcu I (2009) Flight dynamics of flexible aircraft with aeroelastic and inertial force interactions. In: AIAA atmospheric flight mechanics conference, p 6045. <https://doi.org/10.2514/6.2009-6045>
- Nguyen NT, Ting E, Nguyen DY, Trinh KV (2014) Flutter analysis of mission-adaptive wing with variable camber continuous trailing edge flap. In: 55th AIAA/ASME/ASCE/AHS/SC structures, structural dynamics, and materials conference, p 0839. <https://doi.org/10.2514/6.2014-0839>
- Nguyen NT, Ting E, Lebofsky S (2016) Inertial force coupling to nonlinear aeroelasticity of flexible wing aircraft. In: 15th Dynamics specialists conference, p 1094. <https://doi.org/10.2514/6.2016-1094>
- Palacios R, Cesnik C (2005) Static nonlinear aeroelasticity of flexible slender wings in compressible flow. In: 46th AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, p 1945. <https://doi.org/10.2514/6.2005-1945>
- Park GJ (2011) Technical overview of the equivalent static loads method for non-linear static response structural optimization. *Struct Multidisc Optim* 50(3):319–337. <https://doi.org/10.1007/s00158-010-0530-x>
- Patil MJ (1999) Nonlinear aeroelastic analysis, flight dynamics, and control of a complete aircraft. PhD Thesis, Georgia Institute of Technology
- Patil MJ, Hodges DH, Cesnik CES (2001) Nonlinear aeroelasticity and flight dynamics of high-altitude long-endurance aircraft. *J Aircr* 38(1):88–94. <https://doi.org/10.2514/2.2738>
- Pedro J, Bigg CG (2005) Development of a flexible embedded aircraft/control system simulation facility. In: Proceedings of the AIAA modeling and simulation technologies, conference and exhibit, San Francisco, CA, USA. pp 1–25. <https://doi.org/10.2514/6.2005-5889>
- Poon NM, Martins JR (2007) An adaptive approach to constraint aggregation using adjoint sensitivity analysis. *Struct Multidisc Optim* 34(1):61–73. <https://doi.org/10.1007/s00158-006-0061-7>
- Raghavan B (2009) Flight dynamics and control of highly flexible flying-wings. PhD Thesis, Virginia Tech. <http://hdl.handle.net/10919/26829>. Accessed 14 July 2021
- Raveh DE, Levy Y, Karpel M (2000) Structural optimization using computational aerodynamics. *AIAA J* 38(10):1974–1982. <https://doi.org/10.2514/2.853>
- Raveh DE, Levy Y, Karpel M (2001) Efficient aeroelastic analysis using computational unsteady aerodynamics. *J Aircr* 38(3):547–556. <https://doi.org/10.2514/2.2795>
- Richards PW, Griffith DT, Hodges DH (2014) High-fidelity modeling of local effects of damage for derated offshore wind turbines. *J Phys Conf Ser* 524:012036. <https://doi.org/10.1088/1742-6596/524/1/012036>
- Sarojini D (2021) Structural analysis and optimization of aircraft wings through dimensional reduction. PhD Thesis, Georgia Institute of Technology
- Sarojini D, Duca R, Solano HD, Chakraborty I, Briceno SI, Mavris DN (2018) Framework to assess effects of structural flexibility on dynamic loads developed in maneuvering aircraft. In: 2018 Aviation technology, integration, and operations conference, p 4147. <https://doi.org/10.2514/6.2018-4147>
- Sarojini D, Rajaram D, Solano D, Mavris DN (2020) Adjoint-based structural optimization for beam-like structures subjected to dynamic loads. In: AIAA Scitech 2020 Forum, p. 0273. <https://doi.org/10.2514/6.2020-0273>
- Schmidt DK, Raney DL (2001) Modeling and simulation of flexible flight vehicles. *J Guid Control Dyn* 24(3):539–546. <https://doi.org/10.2514/2.4744>
- Shearer CM, Cesnik CE (2007) Nonlinear flight dynamics of very flexible aircraft. *J Aircr* 44(5):1528–1545. <https://doi.org/10.2514/1.27606>
- Sinha K, Klimmek T, Schulze M, Handojo V (2021) Loads analysis and structural optimization of a high aspect ratio, composite wing aircraft. *CEAS Aeronaut J*. <https://doi.org/10.1007/s13272-021-00494-x>
- Stolpe M (2014) On the equivalent static loads approach for dynamic response structural optimization. *Struct Multidisc Optim* 50(6):921–926. <https://doi.org/10.1007/s00158-014-1101-3>
- Stolpe M, Verbart A, Rojas-Labanda S (2018) The equivalent static loads method for structural optimization does not in general generate optimal designs. *Struct Multidisc Optim* 58(1):139–154. <https://doi.org/10.1007/s00158-017-1884-0>
- Takahashi TT, Lemonds T (2015) Transport category wing weight estimation using an optimizing beam-element structural formulation. In: 53rd AIAA aerospace sciences meeting, p 1898. <https://doi.org/10.2514/6.2015-1898>
- Ting E, Nguyen NT, Trinh KV (2014) Static aeroelastic and longitudinal trim model of flexible wing aircraft using finite-element vortex-lattice coupled solution. In: 55th AIAA/ASME/ASCE/AHS/SC structures, structural dynamics, and materials conference, p 0837. <https://doi.org/10.2514/6.2014-0837>
- Waszak MR, Schmidt DK (1988) Flight dynamics of aeroelastic vehicles. *J Aircr* 25(6):563–571. <https://doi.org/10.2514/3.45623>
- Yu W (2011) VABS manual for users