An Adjoint-based Derivative Evaluation Method for Time-dependent Aeroelastic Optimization of Flexible Aircraft

Graeme J. Kennedy* and Joaquim R. R. A. Martins†

University of Michigan, Department of Aerospace Engineering, Ann Arbor, MI, USA

The goal of this paper is to develop techniques to enable the use of aeroelastic constraints within a high-fidelity design optimization framework. As a first-step towards this goal we have developed a fully-coupled aeroelastic analysis tool that includes a coupled structural and aerodynamic analysis as well as rigid-body degrees of freedom. This work departs from previous efforts in two important ways: first, we use solution techniques that are tailored for high-performance parallel computing; second, we implement a fully-coupled adjoint method using a coupled-Krylov approach for the evaluation of derivatives of time-dependent functions of interest. The coupled Newton–Krylov approach enables us to perform simulations in the full structural space without modal reduction. The time-dependent adjoint approach enables us to evaluate the gradient of functions of interest for cases with hundreds or thousands of design variables in a cost similar to the time-dependent simulation. In order to demonstrate our framework, we verify the gradient accuracy of some preliminary simulations of three transport aircraft wings with increasing span subject to gust loads.

I. Introduction

The use of advanced materials and the increasing importance placed on fuel burn reduction is leading to more flexible civil transport aircraft with larger wing spans. Recent studies have demonstrated that even longer spans could lead to further fuel burn reductions, even with existing technology [1]. Evidence of this shift can also be observed in recent aircraft trends, including the 787, 787-8, A350, and 777-X, which all incorporate larger wing spans than previous aircraft within the same class. However, flexible aircraft, with larger wing spans present new analysis and design challenges. In particular, aeroelastic effects including control reversal, gust loading, and flutter will become increasingly important design drivers for the next generation of transport aircraft.

In order to study the effect of aircraft flexibility, several authors have developed aeroelastic analysis tools that integrate aerodynamic and structural analysis coupled to the rigid-body degrees of freedom using low or medium-fidelity methods [2, 3, 4, 5]. The simultaneous consideration of these three disciplines is essential when the rigid-body modes and lowest structural modes interact, producing a fully coupled response. In order to enhance these fully coupled aeroelastic models, several authors have employed geometrically nonlinear beam analyses that account for following forces and wing-area effects that cannot be predicted by linear theories [5, 4]. These nonlinear effects are especially pronounced for large-deflections encountered during gust or maneuver conditions. Both geometrically exact formulation [5] and co-rotationals [6, 3] beam element formulations have been successfully applied to these problems. These methods are especially effective for aircraft with slender, high-aspect ratio wings, such as high-altitude long-endurance (HALE) aircraft. Most of the work performed with these tools has focused on developing novel control schemes for existing aircraft [7, 4]. Fewer studies have addressed incorporating aeroelastic design constraints within a broad aircraft design problem [8]. In addition, studies which do consider aircraft design problem employ computationally expensive gradient-free methods [3].

Other authors have focused on design optimization using static aeroelastic analysis with high-fidelity tools [9, 10, 11]. The optimization results from these studies typically exhibit passive aeroelastic tailoring such that the optimized wing twists under load to alleviate the tip-loading and reduce the bending moments near the wing root [1, 12]. A significant amount of work has been devoted to developing gradient-based design optimization strategies for medium and high-fidelity analysis tools [9, 10, 13, 11]. These methods are effective for complex design optimization problems with thousands of design variables [11]. However, these studies typically ignore dynamic aeroelastic effects. The resulting wings are usually slender and incorporate significant span-extensions. The full merit of these designs cannot be properly assessed without considering the possible increased susceptibility to adverse aeroelastic effects.

The goal of this paper is to begin to close the gap between low-fidelity methods that include aeroelastic effects and high-fidelity optimization techniques that typically focus on static aeroelastic tailoring. As a first-step towards

---

*Postdoctoral Research Fellow, AIAA Member
†Associate Professor, AIAA Associate Fellow

this goal we have developed a fully-coupled aeroelastic analysis tool that includes structural, aerodynamic and rigid-body response of the aircraft. This work departs from previous efforts in two important ways: First, we use solution techniques that are appropriate for high-performance parallel computing. We solve the governing equations using a parallel Newton–Krylov technique that is effective for distributed, computationally expensive methods. A second distinguishing feature of the present work is that we implement a fully-coupled adjoint method using a coupled-Krylov approach for the evaluation of derivatives of time-dependent functions of interest. These methods are required in order to develop a full time-dependent simulation capability using high-fidelity methods. The coupled Newton–Krylov approach enables us to perform simulations in the full structural space without modal reduction. While such reduction is essential for state-space control design and implementation, the modal approach makes nonlinear analysis more difficult [4]. The time-dependent adjoint approach enables us to evaluate the gradient of functions of interest for cases with hundreds or thousands of design variables, in a cost similar to the time-dependent simulation. As part of the parallel analysis framework, we partition the aerodynamic and structural analyses into two non-overlapping processor sub-groups. This parallel partition is required for high-performance multidisciplinary analysis where the application codes may require significant computational resources that cannot be shared between two discipline codes within the same address space [13, 11].

II.   Literature review

In this section, we present an overview of literature related to the primary topics of this paper: aeroelastic analysis and time-dependent adjoint implementations.

Several authors have developed fully-coupled aeroelastic analysis tools that incorporate aerodynamic, structural and rigid-body dynamics. Drela [2] developed a fully coupled aeroelastic analysis tool, called ASWING, where the time-history of the structural, aerodynamic and rigid-body states are obtained using a fully coupled Newton method. The structural analysis consists of a geometrically nonlinear beam code coupled with a lifting-line aerodynamic model with wind-aligned trailing vorticity. A Prandtl–Glauert compressibility correction is used to account for compressibility. Shearer and Cesnik [5] developed an analysis tool to simulate very flexible aircraft during flight using a finite-state aerodynamic model coupled to a nonlinear beam model. Shearer and Cesnik [7] later used this model to perform design of a controller for trajectory tracking. Haghighat *et al.* [3, 4] developed an aeroservoelastic design optimization tool for simultaneous control and structural sizing of an aircraft subject to gust loads. Their approach included a control-law design procedure based on a model-predictive controller. The design optimization was performed using a gradient-free augmented-Lagrangian particle swarm optimizer (ALPSO) [14].

The simulation of time-dependent phenomena using high-fidelity methods is inherently computationally expensive. Optimization problems requiring the analysis of these time-dependent phenomena can quickly become impractical if the number of function evaluations required to obtain a solution grows too rapidly. Gradient-based optimization methods typically require fewer function evaluations than gradient-free methods, but are only more computationally efficient if an adjoint-based sensitivity method can be employed. As a result, several authors have developed time-dependent adjoint implementations for a variety of problems. Nadarajah and Jameson [15] and Nadarajah *et al.* [16] developed a continuous adjoint approach for control and design optimization of unsteady and periodic phenomena. Rumpfkeil and Zingg [17] developed a discrete adjoint approach for time-dependent aerodynamic flows. They later applied their framework to minimize noise from the blunt trailing edge of an airfoil [18]. In an entirely different application, Kennedy and Hansen [19] developed a time-dependent adjoint for the optimization of composite cure cycles. Other authors have developed analysis and optimization frameworks specifically for periodic phenomena. Nadarajah *et al.* [16] developed a nonlinear frequency domain method for the analysis of periodic flows. Mader and Martins [20] developed a time-spectral analysis and design optimization technique for the analysis of periodic motions and applied their framework to the design of a series of flying wings.

Other authors have focused on using high-fidelity methods for static aeroelastic tailoring of aircraft wings. Kennedy and Martins [13] presented a coupled Newton–Krylov approach for static aeroelastic design optimization of aircraft using a medium-fidelity aerodynamic surface panel method coupled to a high-fidelity structural analysis tool. They demonstrated that the Newton–Krylov approach is efficient and robust for slender, flexible structures. Kenway *et al.* [11] presented a coupled aerostructural analysis tool for gradient-based design optimization. The aerostructural analysis consisted of a high-fidelity structural model coupled to Euler-based CFD analysis. They presented an extensive comparison of solution methods and found that the Newton–Krylov technique was the most robust and computationally efficient. Later, Liem *et al.* [12] applied the framework developed by Kenway *et al.* [11] to the design of a full transport aircraft configuration by considering multiple operating points for multiple missions over the full operational envelope of the aircraft.

In this work, we adapt several elements of the MACH framework (MDO of Aircraft configurations with High-

fidelity) to time-dependent aeroelastic analysis of flexible aircraft. The MACH framework have previously been used to perform static aeroelastic design optimization of full aircraft configurations [13, 1, 11, 12]. In particular, we use two analysis tools from MACH: TriPan, a three-dimensional parallel panel method, and the Toolkit for the Analysis of Composite Structures (TACS), an advanced structural analysis tool with parallel solution and derivative evaluation capabilities [13].

The remainder of this paper is organized as follows: In Section III we present the time-dependent solution framework for the aeroelastic equations of motion and the corresponding adjoint-based derivative evaluation method. In Section IV we present the derivation of the full time-dependent aeroelastic equations of motion, including the equations of motion for the rigid-body degrees of freedom, the structural degrees of freedom and the time-dependent aerodynamic analysis. In Section V we present the fully coupled approximate Newton–Krylov solution technique required for each time step as well as the fully-coupled Krylov method for the solution of the adjoint system at each time step. The main difficulty in implementing these methods is the implementation of the preconditioning technique. Here we present a technique based on a Schur-complement approach. Finally, in Section VII, we outline conclusions and recommendations from the present study.

III.   Aeroelastic analysis and time-dependent adjoint framework

A.   Time integration scheme

The equations of motion of a flexible aircraft can expressed in the following form:

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

where $\mathbf{x}$ are the design variables, t is the time parameter, and $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the state variables for all disciplines and their first and second time derivatives, respectively. We have selected to express the equations of motion in this residual form for two reasons: first, this form makes the derivation of the adjoint equations simpler and more compact; second, the equations of motion cannot be reduced to the canonical form: $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ since the mass-matrix associated with the equations of motion for finite-elements shells is not invertible. This is due to the use of drilling degrees of freedom that do not contribute to the kinetic energy of the structure [21, 22].

We integrate equation (1) in time using a backwards-difference formula method with over the interval $t = 0$, $t = T$. We use $N + 1$ points on this time interval with constant time step $\Delta t = T/N$, from the initial point $\mathbf{q}_0$ to the final condition $\mathbf{q}_N$. In this integration scheme, the time derivatives at step k, are approximated using m-th order accurate backwards-difference formula based on the values of the state variables only at current and previous points as follows:

$$
\begin{aligned}
\dot{\mathbf{q}}_k &= \frac{1}{\Delta t} \sum_{i=0}^{m} \alpha_i \mathbf{q}_{k-i} + \mathcal{O}(\Delta t^m), \\
\ddot{\mathbf{q}}_k &= \frac{1}{\Delta t^2} \sum_{i=0}^{2m} \beta_i \mathbf{q}_{k-i} + \mathcal{O}(\Delta t^m).
\end{aligned}
\tag{2}
$$

Note that the first-derivative approximation requires state variable values at $m + 1$ points, while the second derivative approximation requires state variables at $2m + 1$ points.

Inserting the generic BDF formula (2) into the equations of motion (1), yields the following implicit nonlinear equation for the k-th time step:

$$\mathbf{R}(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k) = 0. \tag{3}$$

We solve this nonlinear equation at each time step using a Newton method for a sequence of state variable $\mathbf{q}_k^n$, until the residuals of each discipline are each reduced below a specified tolerance. The linear system of equations for the n-th update $\Delta q_k^n = \mathbf{q}_k^{n+1} - \mathbf{q}_k^n$, is

$$\frac{\partial \mathbf{R}}{\partial \mathbf{q}_k} \Delta \mathbf{q}_k^n = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{q}} + \frac{\alpha_0}{\Delta t} \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{q}}} + \frac{\beta_0}{\Delta t^2} \frac{\partial \mathbf{R}}{\partial \ddot{\mathbf{q}}} \right] \Delta \mathbf{q}_k^n = -\mathbf{R}(\ddot{\mathbf{q}}_k^n, \dot{\mathbf{q}}_k^n, \mathbf{q}_k^n), \tag{4}$$

where we solve the resulting linear system using a Krylov-subspace method. Once the solution to the Newton update is obtained, the following updates are applied to the state variables and their approximate time-derivatives:

$$
\begin{aligned}
\mathbf{q}_k^{n+1} &= \mathbf{q}_k^n + \Delta \mathbf{q}_k^n \\
\dot{\mathbf{q}}_k^{n+1} &= \dot{\mathbf{q}}_k^n + \frac{\alpha_0}{\Delta t} \Delta \mathbf{q}_k^n \\
\ddot{\mathbf{q}}_k^{n+1} &= \ddot{\mathbf{q}}_k^n + \frac{\beta_0}{\Delta t^2} \Delta \mathbf{q}_k^n
\end{aligned}
$$

American Institute of Aeronautics and Astronautics

Instead of forming the exact Jacobian at each iteration, we use an approximate Jacobian that is less expensive to evaluate. We outline the details of this approximation below. Furthermore, we solve the linear system (4) using a Krylov subspace method [23]. Therefore, following convention, we refer to this algorithm as an approximate Newton–Krylov solution method.

The implicit BDF family of time-integration schemes are not self-starting. As a result, at time step $j < m$ there are not enough points to apply the full m-th order BDF formula (2). Instead, we use a start-up strategy where a j-th order BDF formula is applied for the first $j = 1, \ldots, m$ points at which point the full m-th order accurate BDF integrations scheme can be applied. Note that at the first $2m$ points, the second-order approximate formula must be modified based on both the initial starting condition $\dot{\mathbf{q}}_0$ and state variable values at the previous time steps.

As part of the optimization problem, we evaluate constraint or objective functions based on the time-history of the state variables. These functions can be written in the generic form, $f(\mathbf{x}, \mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_N)$. Note that this is a discrete function of the state variables at the discrete points, $\mathbf{q}_k$, therefore, any time derivative that are required for the evaluation of f, must be approximated using finite-difference formula.

B. Adjoint derivation

We have implemented an adjoint method for the efficient evaluation of the gradients of functions of interest. The cost of the adjoint method scales weakly with the number of design variables and is therefore best suited to optimization problems with few functions of interest and many design variables. Our goal is to implement an approach which can be applied to aeroelastic design optimization with computationally expensive high-fidelity analysis methods.

We derive the adjoint equations using a Lagrangian approach. In this approach, the Lagrangian, $\mathcal{L}$, is defined as follows:

$$\mathcal{L} = f(\mathbf{x}, \mathbf{q}_0, \ldots, \mathbf{q}_N) + \sum_{k=0}^{N} \boldsymbol{\lambda}_k^T \mathbf{R}_k(\ddot{\mathbf{q}}_k, \dot{\mathbf{q}}_k, \mathbf{q}_k), \tag{5}$$

where f is the function of interest, $\boldsymbol{\lambda}_k$ are the Lagrange multipliers at each time step k and $\mathbf{R}_k$ are the residuals evaluated at the k-th time step.

In order to determine the values of the Lagrange multipliers, $\boldsymbol{\lambda}_k$, we evaluate the partial derivative of Lagrangian with respect to the state variables at the k-th time step:

$$\nabla_{\mathbf{q}_k} \mathcal{L} = \frac{\partial f}{\partial \mathbf{q}_k} + \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{R}_k}{\partial \mathbf{q}} + \sum_{i=0}^{m} \boldsymbol{\lambda}_{k+i}^T \frac{\alpha_i}{\Delta t} \frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}} + \sum_{i=0}^{2m} \boldsymbol{\lambda}_{k+i}^T \frac{\beta_i}{\Delta t} \frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}}.$$

Rearranging the terms, and taking the transpose of the above expression results in the following equation for the Langrange multipliers at the k-th time step:

$$\left[\frac{\partial \mathbf{R}_k}{\partial \mathbf{q}} + \frac{\alpha_0}{\Delta t} \frac{\partial \mathbf{R}_k}{\partial \dot{\mathbf{q}}} + \frac{\beta_0}{\Delta t} \frac{\partial \mathbf{R}_k}{\partial \ddot{\mathbf{q}}} \right]^T \boldsymbol{\lambda}_k = -\frac{\partial f}{\partial \mathbf{q}_k}^T - \sum_{i=1}^{m} \frac{\alpha_i}{\Delta t} \frac{\partial \mathbf{R}_{k+i}}{\partial \dot{\mathbf{q}}}^T \boldsymbol{\lambda}_{k+i} - \sum_{i=1}^{2m} \frac{\beta_i}{\Delta t} \frac{\partial \mathbf{R}_{k+i}}{\partial \ddot{\mathbf{q}}}^T \boldsymbol{\lambda}_{k+i} \tag{6}$$

Note that in this expression, the right-hand-side for the Langrange multipliers at the current time step depends only on future values of the Lagrange multipliers. As a result, it is necessary to obtain the Lagrange multipliers by marching backwards in time starting from the last point $k = N$, and proceeding backwards to the initial point $k = 0$. Note that Equation (6) requires the same Jacobian as the Newton update in Equation (4). In this instance, however, it is essential that the exact Jacobian be employed any approximation may result in gradient inaccuracies that produce poor optimizer convergence behavior. We solve Equation (6) using a fully coupled Krylov method described in Section V.

Once the values of the Lagrange multipliers have been obtained, the total derivative of the function of interest can be evaluated as follows:

$$\nabla_{\mathbf{x}} f \equiv \nabla_{\mathbf{x}} \mathcal{L} = \frac{\partial f}{\partial \mathbf{x}} + \sum_{k=0}^{N} \boldsymbol{\lambda}_k^T \frac{\partial \mathbf{R}_k}{\partial \mathbf{x}}. \tag{7}$$

Note that in practice it is necessary to store the entire time-history of the simulation for the forward solve and only $2m$ right-hand sides for the backwards adjoint solve. Some authors have had success with skipping time steps during the adjoint solution phase thereby reducing storage requires for the forward time-integration. Within the present framework, we compute the matrix-vector product contributions required by all right-hand-sides immediately following the calculation of the Lagrange multipliers at time step k. Following this update step, we also compute the contribution to the total derivative (7) from time-step k. Therefore, within this framework, we only store the Lagrange multipliers for the current time step, but accumulate the right-hand-sides for the next $2m + 1$ time steps.

American Institute of Aeronautics and Astronautics

IV.   The equations of motion

In this section, we derive the equations of motion for a flexible aircraft and outline the most significant terms required for the general framework presented in Section III.

Within this work, we subdivide the state vector $\mathbf{q}$ into the rigid-body degrees of freedom, $\mathbf{q}_R$, the aerodynamic state variables $\mathbf{q}_A$ and the structural state variables $\mathbf{q}_S$. We plan to employ both beam-type structural models and shell-element-based structural models of the fully flexible aircraft. As a result, we make no prior assumptions about the degrees of freedom present in $\mathbf{q}_S$. For similar reasons, we make no prior assumption about the content of the aerodynamic state variable vector $\mathbf{q}_A$.

A.   Dynamics of an elastic body

In this work, we derive the equations of motion using Lagrange's equations for quasi-coordinates [24]. In this derivation, we require an expression for the kinetic energy of the elastic body and the potential energy within the system. Figure 1 shows the initial and deformed configurations of the elastic body as well as the inertial reference frame, $\vec{\mathcal{F}}_{\mathbf{I}}$, and the body-fixed reference frame, $\vec{\mathcal{F}}_{\mathbf{B}}$. Note that the origin of the body-fixed frame is not necessarily at the center of mass and is not required to lie within the body itself.
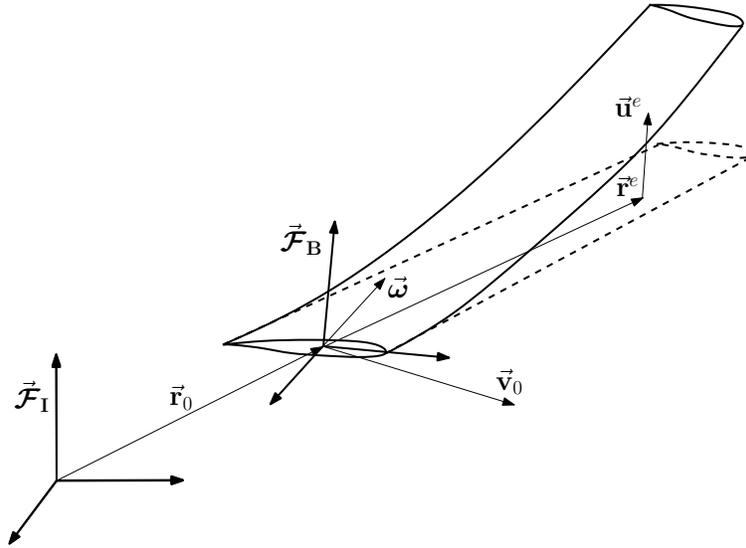

Figure 1: An illustration of the elastic body deformed under load in the body-fixed reference frame.

We express all vectors shown in Figure 1 within the body-fixed reference frame, except for the position vector, $\vec{\mathbf{r}}_0$ which is expressed in the inertial coordinate axis. The origin of the body-fixed frame has velocity $\mathbf{v}_0$ and angular velocity $\boldsymbol{\omega}$. The initial, undeformed position of a point within the elastic body is given by $\mathbf{r}^e$ and the displacement of a point in the body under load is given by $\mathbf{u}^e$. As a result, the velocity of an arbitrary point within the body can be written as follows:

$$\mathbf{v} = \mathbf{v}_0 + \dot{\mathbf{u}}^e + \boldsymbol{\omega}^\times (\mathbf{r}^e + \mathbf{u}^e) \tag{8}$$

where we have employed matrix notation for the cross product operation [25] such that:

$$\mathbf{a} \times \mathbf{b} = \mathbf{a}^\times \mathbf{b},$$

where $\mathbf{a}^\times$ is a 3×3 skew-symmetric matrix with components

$$\mathbf{a}^\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}.$$

Based on Equation (8), the kinetic energy can now be expressed as a function of the rigid-body velocity and the

point-wise elastic deformation as follows:

$$
\begin{aligned}
T &= \frac{1}{2} \int_\Omega \mathbf{v}^T \mathbf{v} \mathrm{d}m, \\
&= \frac{1}{2} \int_\Omega \left(\mathbf{v}_0 + \dot{\mathbf{u}}^e - (\mathbf{r}^e + \mathbf{u}^e)^\times \boldsymbol{\omega}\right)^T \left(\mathbf{v}_0 + \dot{\mathbf{u}}^e - (\mathbf{r}^e + \mathbf{u}^e)^\times \boldsymbol{\omega}\right) \mathrm{d}m, \\
&= \frac{1}{2} \begin{bmatrix} \mathbf{v}_0^T & \boldsymbol{\omega}^T & \dot{\mathbf{q}}_S^T \end{bmatrix} \begin{bmatrix} m\mathbf{1} & -\hat{\mathbf{c}}^\times & \mathbf{p} \\ \hat{\mathbf{c}}^\times & \hat{\mathbf{J}} & \hat{\mathbf{h}} \\ \mathbf{p}^T & \hat{\mathbf{h}}^T & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega} \\ \dot{\mathbf{q}}_S \end{bmatrix}, \\
&= \frac{1}{2} \left(m\mathbf{v}_0^T \mathbf{v}_0 + \boldsymbol{\omega}^T \hat{\mathbf{J}} \boldsymbol{\omega} + \dot{\mathbf{q}}_S^T \mathbf{M} \dot{\mathbf{q}}_S\right) - \mathbf{v}_0^T \hat{\mathbf{c}}^\times \boldsymbol{\omega} + \mathbf{v}_0^T \mathbf{p} \dot{\mathbf{u}} + \boldsymbol{\omega}^T \hat{\mathbf{h}} \dot{\mathbf{q}}_S.
\end{aligned}
\tag{9}
$$

Where we have used the following expression for the finite-element shape functions $\mathbf{u}^e = \mathbf{N}\mathbf{q}_S$. The matrices $\hat{\mathbf{c}}$, $\hat{\mathbf{J}}$, $\mathbf{p}$ and $\hat{\mathbf{h}}$ are defined as follows:

$$
\hat{\mathbf{c}} = \int_\Omega (\mathbf{r}^e + \mathbf{u}^e)\mathrm{d}m \qquad\qquad \mathbf{p} = \int_\Omega \mathbf{N}\mathrm{d}m
$$

$$
\hat{\mathbf{J}} = -\int_\Omega (\mathbf{r}^e + \mathbf{u}^e)^\times (\mathbf{r}^e + \mathbf{u}^e)^\times \mathrm{d}m \quad \hat{\mathbf{h}} = \int_\Omega (\mathbf{r}^e + \mathbf{u}^e)^\times \mathbf{N}\mathrm{d}m
$$

where the symbol $\hat{}$ denotes quantities that depend on the state of elastic deformation.

In the following work, we simplify the equations of motion by making the assumption that quantities within the kinetic energy expression of the form $(\mathbf{r}^e)^\times \mathbf{u}^e$ can be neglected. For relatively rigid aircraft, this is a reasonable assumption, however, these terms cannot be neglected when highly-flexible structures are analyzed. However, the general framework for aeroelastic analysis and adjoint-based derivative evaluation can be extended to accomodate the more general case. With this assumption, the kinetic energy (9) can now be expressed as follows:

$$
T = \frac{1}{2} \left(m\mathbf{v}_0^T \mathbf{v}_0 + \boldsymbol{\omega}^T \mathbf{J} \boldsymbol{\omega} + \dot{\mathbf{q}}_S^T \mathbf{M} \dot{\mathbf{q}}_S\right) - \mathbf{v}_0^T \mathbf{c}^\times \boldsymbol{\omega} + \mathbf{v}_0^T \mathbf{p}\dot{\mathbf{u}} + \boldsymbol{\omega}^T \mathbf{h}\dot{\mathbf{q}}_S,
\tag{10}
$$

where the matrices, $\mathbf{c}$, $\mathbf{J}$ and $\mathbf{h}$ are defined as follows:

$$
\mathbf{c} = \int_\Omega \mathbf{r}^e \mathrm{d}m, \qquad \mathbf{J} = -\int_\Omega \mathbf{r}^{e\times} \mathbf{r}^{e\times} \mathrm{d}m, \qquad \mathbf{h} = \int_\Omega \mathbf{r}^{e\times} \mathbf{N}\mathrm{d}m.
$$

The potential energy of the elastic body is due to both the strain energy of the elastic deformation and the potential energy due to gravity. As a result, the total potential energy can be written as follows:

$$
U = m\mathbf{g}^T \mathbf{r}_0 + \mathbf{q}_S^T \mathbf{K}\mathbf{q}_S,
\tag{11}
$$

where $\mathbf{g}$ is the acceleration due to gravity expressed in the inertial coordinate frame and $\mathbf{K}$ is the stiffness matrix. Note that we have assumed a linear structure, however, geometric nonlinearities due to large deformations could also be taken into account within this framework without significant modification.

The equations of motion for an elastic body can be derived using Lagrange's equations expressed using quasi-coordinates [24] as follows:

$$
\begin{aligned}
\frac{d}{dt}\left(\frac{\partial L}{\partial \mathbf{v}_0}^T\right) + \boldsymbol{\omega}^\times \frac{\partial L}{\partial \mathbf{v}_0}^T - \mathbf{C}_{bi}\frac{\partial L}{\partial \mathbf{r}_0}^T &= \mathbf{f}_b, \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \boldsymbol{\omega}}^T\right) + \mathbf{v}_0^\times \frac{\partial L}{\partial \mathbf{v}_0}^T + \boldsymbol{\omega}^\times \frac{\partial L}{\partial \boldsymbol{\omega}}^T - \mathbf{S}^{-T}\frac{\partial L}{\partial \boldsymbol{\theta}}^T &= \mathbf{g}_b, \\
\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{q}}_S}^T\right) - \frac{\partial L}{\partial \mathbf{q}_S}^T &= \mathbf{f},
\end{aligned}
\tag{12}
$$

where L is the Lagrangian $L = T - U$. Here, $\boldsymbol{\theta}$ is an Euler-angle sequence, $\mathbf{C}_{bi}$ is the rotation matrix between the inertial and body-fixed coordinate frames and $\mathbf{S}$ is a matrix defined such that $\boldsymbol{\omega} = \mathbf{S}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$. The vectors $\mathbf{f}_R$ and $\mathbf{g}_R$ are the total aerodynamic forces and torques acting on the body expressed in the body-fixed frame, and $\mathbf{f}$ is the consistent aerodynamic force vector. These terms will be described in greater detail below.

The equations of motion can now be derived using the expressions for the kinetic and potential energy in Equation (10) and Equation (11) respectively in combination with the equations of motion in terms of quasi-coordinates (12). This results in the following two equations for the rigid-body motion:

$$m\dot{\mathbf{v}}_0 - \mathbf{c}^\times \dot{\boldsymbol{\omega}} + \mathbf{p}\ddot{\mathbf{q}}_S + \boldsymbol{\omega}^\times(m\mathbf{v}_0 - \mathbf{c}^\times\boldsymbol{\omega} + \mathbf{p}\dot{\mathbf{q}}_S) - m\mathbf{C}_{bi}\mathbf{g} - \mathbf{f}_R = 0,$$
$$\mathbf{c}^\times\dot{\mathbf{v}}_0 + \mathbf{J}\dot{\boldsymbol{\omega}} + \mathbf{h}\ddot{\mathbf{q}}_S + \mathbf{c}^\times\boldsymbol{\omega}^\times\mathbf{v}_0 + \boldsymbol{\omega}^\times\mathbf{J}\boldsymbol{\omega} + \mathbf{v}_0^\times\mathbf{p}\dot{\mathbf{q}}_S + \boldsymbol{\omega}^\times\mathbf{h}\dot{\mathbf{q}}_S - \mathbf{c}^\times\mathbf{C}_{bi}\mathbf{g} - \mathbf{g}_R = 0,$$
(13)

while the elastic degrees of freedom are governed by the following equation of motion:

$$\mathbf{p}^T\dot{\mathbf{v}}_0 + \mathbf{h}^T\dot{\boldsymbol{\omega}} + \mathbf{M}\ddot{\mathbf{q}}_S + \mathbf{K}\mathbf{q}_S - \mathbf{p}^T\mathbf{C}_{bi}\mathbf{g} - \mathbf{f} = 0.$$
(14)

In addition to the governing equations derived above, it is necessary to integrate the following kinematic equations to obtain the trajectory of the aircraft:

$$\dot{\mathbf{r}}_0 = \mathbf{C}_{bi}^T\mathbf{v}_0,$$
$$\dot{\boldsymbol{\theta}} = \mathbf{S}^{-1}\boldsymbol{\omega}.$$
(15)

The governing equations for the rigid-body motion of the flexible aircraft are the concatenation of the system dynamics (13) and the kinematic relationships (15). For the remainder of this paper, we write these governing equations in the following residual form:

$$\mathbf{R}_R(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) = 0.$$
(16)

Note that the aerodynamic forces contribute directly to the rigid body dynamics (13), as well as the structural equations of motion (14). The structures is linked to the rigid-body degrees of freedom through the inertial terms in Equation (13). In addition to this direct coupling, there is also aeroelastic coupling between the aerodynamic forces that change as the aircraft shape deforms elastically. As a result, the aerodynamics forces also depend on the structural displacement.

Within the context of this study, these two systems of equations are treated differently. Equation (16) represents 12 equations governing the rigid-body components of the motion, while Equation (14) contains all the degrees of freedom in the finite-element mesh. We solve Equation (14) using parallel finite-element methods, where the equations of motion and the solution procedure must be distributed amongst a group of structural processors to compute the solution in a computationally efficient manner. Furthermore, this group of structural processors are distinct from the group of aerodynamic processors responsible for computing the aerodynamic solution.

B. Structural analysis

The structural analysis is performed using the Toolkit for the Analysis of Composite Structures (TACS), a parallel finite-element code designed for the analysis of thin-walled composite structures using either geometrically linear or nonlinear analysis. The residuals of the equations of motion governing the structural degrees of freedom (14) are written here as:

$$\mathbf{R}_S(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) = 0.$$
(17)

The Jacobian of the structural residuals (17) with respect to the structural state variables, $\mathbf{q}_S$, consists of contributions from two terms: the tangent stiffness matrix and a non-symmetric term resulting from the application of the aerodynamic forces. This Jacobian can be written as follows:

$$\frac{\partial\mathbf{R}_S}{\partial\mathbf{q}_S} = \mathbf{K} - \frac{\partial\mathbf{f}}{\partial\mathbf{q}_S},$$
(18)

where $\mathbf{K}$ is the tangent stiffness matrix and $\partial\mathbf{f}/\partial\mathbf{q}_S$, is the derivative of the aerodynamic forces with respect to the structural state variables. In practice, computing products with the exact Jacobian requires contributions from both the aerodynamic and structural processor groups. For preconditioning operations, and inexact solutions, we do not compute the contributions from this second term, however, for accurate gradient-evaluation, it is essential to ensure that this term is properly computed.

C. Load and displacement transfer scheme

We employ a parallel load and displacement transfer scheme to transfer the aerodynamic loads to the structural model and to extrapolate the displacements from the structural model to the aerodynamic surface. Our approach is an extension of the work of Brown [26] that is constructed to be consistent and conservative. In a departure from Brown's

approach, we use an adaptive-technique to ensure that the loads from the aerodynamic mesh are integrated with sufficient accuracy.

In the present approach, we extrapolate the structural displacements are extrapolated to the aerodynamic surface mesh through a series of rigid links. These links connect the nodes on the aerodynamic surface to the closest point on the structure. The nodal displacement on the aerodynamic surface is extrapolated based on the displacement and rotation at the connecting point on the structural mesh as follows,

$$\mathbf{u}^a = \mathbf{u}^e + \boldsymbol{\theta}^\times \mathbf{r}_\ell, \tag{19}$$

where $\mathbf{u}^a$ and $\mathbf{u}^e$ are the displacements of the aerodynamic and structural points respectively, $\boldsymbol{\theta}$ are a set of small rotations at the structural surface and $\mathbf{r}_\ell$ is the rigid link. Next, we use the method of virtual work in conjunction with the displacement extrapolation (19), to construct an expression for the forces and moments acting at the nodes of the structural mesh. The virtual work of the aerodynamic pressure forces is,

$$\delta W = \int_{\Omega_A} \mathbf{t}^T \delta \mathbf{u}^a \, \mathrm{d}\Omega_A = \int_{\Omega_A} \left(\mathbf{t}^T \delta \mathbf{u}^e - \mathbf{t}^T \mathbf{r}_\ell^\times \delta \boldsymbol{\theta} \right) \mathrm{d}\Omega_A = \mathbf{f}^T \delta \mathbf{q}_S, \tag{20}$$

where $\mathbf{t}$ is the aerodynamic surface traction and $\mathbf{f}$ is the consistent force vector. We approximate the integral in Equation (20) using a quadrature scheme that matches the order of the finite-element mesh. In addition, we apply an adaptive refinement technique that splits high-aspect ratio aerodynamic cells to ensure that the integration is sufficiently accurate.

D.   Aerodynamic analysis

In this work, we perform the aerodynamic analysis using TriPan, a parallel, unstructured three-dimensional panel code for calculating the aerodynamic forces and moments for inviscid, external lifting flows governed by the Prandtl–Glauert equation [13]. TriPan uses a discretization based on constant source and doublet singularity elements distributed over the surface represented by quadrilateral and triangular panels [27, 28]. The aerodynamic forces and moments are calculated using surface pressure integration, while the induced drag is calculated using a Trefftz-plane integration scheme [29].

For the present work, it was necessary to modify TriPan to enable the simulation of unsteady flows with moving aerodynamic surfaces. Moving aerodynamic surfaces account for the relative motion of the structure within the body-fixed aerodynamic reference frame. For unsteady aerodynamic flows, the zero normal flow boundary condition must be modified to reflect the velocity of the moving surface. This modified boundary condition can be written as follows:

$$(\nabla \Phi + \mathbf{v}_f) \cdot \mathbf{n} = 0 \tag{21}$$

where $\mathbf{n}$ is the surface normal and $\mathbf{v}_f$ is the local onset velocity of the fluid. The onset fluid velocity must be opposite to the velocity of the aerodynamic surface. Therefore, the surface velocity is a combination of the rigid motion and elastic deformation as follows:

$$\left(\nabla \Phi - \left(\mathbf{v}_0 + \dot{\mathbf{u}}^a + \boldsymbol{\omega}^\times (\mathbf{r}^a + \mathbf{u}^a) \right) \right) \cdot \mathbf{n} = 0, \tag{22}$$

where $\mathbf{v}_0$ and $\boldsymbol{\omega}$ are the velocity and angular velocity of the origin of the body-fixed frame, $\mathbf{u}^a$ is the displacement of the aerodynamic surface and $\mathbf{r}^a$ is the position of the point on the aerodynamic surface in the body-fixed frame.

The full system of discretized boundary conditions produce a system of equations expressed in terms of the doublet strengths, $\mathbf{q}_A$. The resulting set of governing equations are represented as follows:

$$\mathbf{R}_A(\dot{\mathbf{q}}, \mathbf{q}) = 0, \tag{23}$$

where there is no second-order time-derivative dependence. The governing equations represented by Equation (23) are solved in parallel using PETSc [30, 31]. The dense aerodynamic influence coefficient matrix is distributed amongst the aerodynamic processors by assigning a segment of contiguous rows to each processor based on a domain decomposition of the surface mesh performed using METIS [32]. On each processor, the matrix is stored in a dense column-major matrix storage format. More details of the solution technique are presented in Kennedy and Martins [13]

V.   The Krylov-based solution methods

In this section, we outline the details of the solution methods used within the aeroelastic analysis framework. In particular, we describe the approximate Newton–Krylov solution method applied at each stage of the time-integration

scheme, as well as the fully-coupled Krylov solution method used to obtain the adjoint variables at each time-step of the time-dependent adjoint computation. In both cases, we use Krylov-subspace methods to solve the linear systems that are obtained as part of the solution procedure. Implementing an efficient and effective preconditioning technique is the single most difficult task when using Krylov methods. Within this work, we use a hybrid preconditioning method consisting of a block-Jacobi preconditioner for the aerodynamic states and a Schur-complement-based preconditioner for the combined structural and rigid-body degrees of freedom. The preconditioner for the aerodynamic block is based on the Krylov-subspace method GMRES [33], and as a result, a flexible Krylov-subspace method must be used for the global system. For the work presented here, we use flexible GMRES (FGMRES) [34] for both the approximate Newton–Krylov solution method and the adjoint calculations.

The overall efficiency of the simulation and the adjoint method are directly related to the efficiency of the Krylov methods used to solve the coupled linear systems at each iteration. Therefore, we have attempted to optimize the performance of the operations required to implement a Krylov-subspace methods without sacrificing the modularity of the underlying code. In order to adhere to this requirement we have used an abstract vector class that defines all the vector operations necessary to implement a Krylov subspace method. We have created a vector class for aeroelastic analysis derived from the abstract vector class that contains the aerodynamic, structural and rigid-body degrees of freedom, as a hybrid-vector of the native TACS solution vector, TriPan PETSc vector, and a 12-degree of freedom array with the rigid-body degrees of freedom. Since the vectors for each discipline are stored in the native format, no conversion is necessary for any operation. This enables a fast and clear implementation within the existing code.

A. The Jacobian of the equations of motion

The global system of governing equations is the concatenation of the residuals for the rigid-body degrees of freedom, the structural equations of motion, and the residuals governing the aerodynamics. This system can be written as follows:

$$\mathbf{R}(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}, t) = \begin{bmatrix} \mathbf{R}_R(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \\ \mathbf{R}_S(\ddot{\mathbf{q}}, \dot{\mathbf{q}}, \mathbf{q}) \\ \mathbf{R}_A(\dot{\mathbf{q}}, \mathbf{q}) \end{bmatrix} = 0, \tag{24}$$

where the vector $\mathbf{q}$ consists of the concatenation of all state variables as follows $\mathbf{q}^T = \begin{bmatrix} \mathbf{q}_R^T & \mathbf{q}_S^T & \mathbf{q}_A^T \end{bmatrix}$.

At each time step we obtain a new initial guess based on the values of the state variables and their derivatives at the previous iteration

$$\mathbf{q}_k^0 = \mathbf{q}_{k-1} + \Delta t \dot{\mathbf{q}}_{k-1} + \frac{\Delta t^2}{2} \ddot{\mathbf{q}}_{k-1}. \tag{25}$$

We have found that this initial estimate provides a good starting point for Newton's method, and therefore, no start-up strategy is required. At each iteration of Newton's method, a linear system with the following form must be solved:

$$\left[\alpha \frac{\partial \mathbf{R}}{\partial \mathbf{q}} + \beta \frac{\partial \mathbf{R}}{\partial \dot{\mathbf{q}}} + \gamma \frac{\partial \mathbf{R}}{\partial \ddot{\mathbf{q}}} \right] \mathbf{x} = \mathbf{b},$$

written in component form, this equation becomes:

$$\begin{bmatrix} \mathbf{D}_R + \mathbf{S}_R + \mathbf{A}_R & \mathbf{S}_{RS} + \mathbf{A}_{RS} & \mathbf{A}_{RA} \\ \mathbf{S}_{SR} + \mathbf{A}_{SR} & \mathbf{S}_S + \mathbf{A}_S & \mathbf{A}_{SA} \\ \mathbf{A}_{AR} & \mathbf{A}_{AS} & \mathbf{A}_A \end{bmatrix} \begin{bmatrix} \mathbf{x}_R \\ \mathbf{x}_S \\ \mathbf{x}_A \end{bmatrix} = \begin{bmatrix} \mathbf{b}_R \\ \mathbf{b}_S \\ \mathbf{b}_A \end{bmatrix}. \tag{26}$$

Here the letters $\mathbf{D}$, $\mathbf{S}$, and $\mathbf{A}$ indicate the primary discipline responsible for the calculation of the given term and represent the rigid-body dynamics, structures and aerodynamics, respectively.

The Jacobian of the rigid-body dynamics, $\mathbf{D}_R$, is a linear combination of the derivatives of the rigid state variables $\mathbf{q}_R$ and their first time derivatives, as follows:

$$\mathbf{D}_R = \begin{bmatrix} \beta \mathbf{C}_{bi} & \alpha(\mathbf{C}_{bi}\dot{\mathbf{r}}_0)^\times \mathbf{S} & -\alpha \mathbf{1} & 0 \\ 0 & \alpha(\dot{\mathbf{S}} + (\mathbf{S}\dot{\boldsymbol{\theta}})^\times \mathbf{S}) + \beta \mathbf{S} & 0 & -\alpha \mathbf{1} \\ 0 & -\alpha\, m(\mathbf{C}_{bi}\mathbf{g})^\times \mathbf{S} & m(\beta \mathbf{1} + \alpha \boldsymbol{\omega}^\times) & -\beta \mathbf{c}^\times + \alpha((\mathbf{c}^\times \boldsymbol{\omega})^\times - m\mathbf{v}_0^\times - \boldsymbol{\omega}^\times \mathbf{c}^\times) \\ 0 & -\alpha \mathbf{c}^\times(\mathbf{C}_{bi}\mathbf{g})^\times \mathbf{S} & \beta \mathbf{c}^\times + \alpha \mathbf{c}^\times \boldsymbol{\omega}^\times & \beta \mathbf{J} + \alpha\left(\boldsymbol{\omega}^\times \mathbf{J} - (\mathbf{J}\boldsymbol{\omega})^\times - \mathbf{c}^\times \mathbf{v}_0^\times\right) \end{bmatrix},$$

where $\mathbf{C}_{bi}$, $\mathbf{S}$ and $\mathbf{g}$ are the rotation matrix, the angular rate matrix, and the acceleration due to gravity, respectively. The Jacobian $\mathbf{D}_R$, is a 12×12 matrix calculated and stored on all structural processors during the computation.

American Institute of Aeronautics and Astronautics

The structural matrices in Equation (26), are only calculated on the structural processors, and do not require communication with the aerodynamic processors. The four matrices required for the full Jacobian are as follows:

$$\mathbf{S}_R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\alpha(\mathbf{p}\dot{\mathbf{q}}_S)^\times \\ 0 & 0 & -\alpha(\mathbf{p}\dot{\mathbf{q}}_S)^\times & -\alpha(\mathbf{h}\dot{\mathbf{q}}_S)^\times \end{bmatrix}, \qquad \mathbf{S}_S = \alpha\mathbf{K} + \gamma\mathbf{M},$$

$$\mathbf{S}_{RS} = \begin{bmatrix} 0 \\ 0 \\ \gamma\mathbf{p} + \beta\boldsymbol{\omega}^\times\mathbf{p} \\ \gamma\mathbf{h} + \beta(\mathbf{v}_0^\times\mathbf{p} + \boldsymbol{\omega}^\times\mathbf{h}) \end{bmatrix}, \qquad \mathbf{S}_{SR} = \begin{bmatrix} 0 & -\alpha\mathbf{p}^T(\mathbf{C}_{bi}\mathbf{g})^\times\mathbf{S} & \beta\mathbf{p}^T & \beta\mathbf{h}^T \end{bmatrix}.$$

In our implementation, the matrices $\mathbf{S}_{RS}$ and $\mathbf{S}_{SR}$ are stored as a series of vectors where only the non-zero vectors are computed and stored. Note that the matrix $\mathbf{S}_S$ is a linear combination of the stiffness matrix and the mass matrix and, as a result, is non-singular.

The aerodynamic coupling matrices contribute to all components of the coefficient matrix, five of these matrices require communication between the aerodynamic and structural processors, while the remaining four matrices require no communication.

The aerodynamic Jacobian components that couple to the rigid-body equations of motion are:

$$\mathbf{A}_R = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\dfrac{\partial\mathbf{f}_R}{\partial\mathbf{v}_0} & -\dfrac{\partial\mathbf{f}_R}{\partial\boldsymbol{\omega}} \\ 0 & 0 & -\dfrac{\partial\mathbf{g}_R}{\partial\mathbf{v}_0} & -\dfrac{\partial\mathbf{g}_R}{\partial\boldsymbol{\omega}} \end{bmatrix}, \qquad \mathbf{A}_{RS} = \begin{bmatrix} 0 \\ 0 \\ -\alpha\dfrac{\partial\mathbf{f}_R}{\partial\mathbf{X}_A}\dfrac{\partial\mathbf{X}_A}{\partial\mathbf{q}_S} - \beta\dfrac{\partial\mathbf{f}_R}{\partial\mathbf{V}_A}\dfrac{\partial\mathbf{V}_A}{\partial\dot{\mathbf{q}}_S} \\ -\alpha\dfrac{\partial\mathbf{g}_R}{\partial\mathbf{X}_A}\dfrac{\partial\mathbf{X}_A}{\partial\mathbf{q}_S} - \beta\dfrac{\partial\mathbf{g}_R}{\partial\mathbf{V}_A}\dfrac{\partial\mathbf{V}_A}{\partial\dot{\mathbf{q}}_S} \end{bmatrix},$$

$$\mathbf{A}_{RA} = \begin{bmatrix} 0 \\ 0 \\ -\alpha\dfrac{\partial\mathbf{f}_R}{\partial\mathbf{q}_A} - \beta\dfrac{\partial\mathbf{f}_R}{\partial\dot{\mathbf{q}}_A} \\ -\alpha\dfrac{\partial\mathbf{g}_R}{\partial\mathbf{q}_A} - \beta\dfrac{\partial\mathbf{g}_R}{\partial\dot{\mathbf{q}}_A} \end{bmatrix},$$

where the first term $\mathbf{A}_{RS}$ requires communication between the aerodynamic and structural processors. The components $\partial\mathbf{X}_A/\partial\mathbf{q}_S$ and $\partial\mathbf{V}_A/\partial\dot{\mathbf{q}}_S$ denote the derivative of the aerodynamic nodal locations with respect to the structural state variables and the derivative of the aerodynamic nodal velocities with respect to the rate of change of the structural state variables. These two terms must be computed on the structural processors and transfered to the aerodynamic processors where the matrix-vector product can be completed. Note that all components of the second term, $\mathbf{A}_{RA}$, can be computed on the aerodynamic processors without communication.

Next, we examine the terms that couple the aerodynamic forces to the structural equations of motion. These terms can be computed as follows:

$$\mathbf{A}_{SR} = -\alpha\frac{\partial\mathbf{f}}{\partial\mathbf{F}_A}\frac{\partial\mathbf{F}_A}{\partial\mathbf{q}_R} \qquad \mathbf{A}_S = -\alpha\frac{\partial\mathbf{f}}{\partial\mathbf{F}_A}\frac{\partial\mathbf{F}_A}{\partial\mathbf{X}_A}\frac{\partial\mathbf{X}_A}{\partial\mathbf{q}_S} - \beta\frac{\partial\mathbf{f}}{\partial\mathbf{F}_A}\frac{\partial\mathbf{F}_A}{\partial\mathbf{V}_A}\frac{\partial\mathbf{V}_A}{\partial\dot{\mathbf{q}}_S},$$

$$\mathbf{A}_{SA} = -\alpha\frac{\partial\mathbf{f}}{\partial\mathbf{F}_A}\frac{\partial\mathbf{F}_A}{\partial\mathbf{q}_A} - \beta\frac{\partial\mathbf{f}}{\partial\mathbf{F}_A}\frac{\partial\mathbf{F}_A}{\partial\dot{\mathbf{q}}_A},$$

where $\mathbf{F}_A$ represents an intermediate vector of forces within the load transfer operation that is stored on the aerodynamic processors. Note that the terms $\mathbf{A}_{SR}$ and $\mathbf{A}_{SA}$ require one-way communication from the aerodynamic processors to the structural processors, while the term $\mathbf{A}_S$ requires communication first from the structural processors to the aerodynamic processors, then a reverse communication from the aerodynamic to the structural processors.

The last set of terms represet the coupling to the aerodynamic residual equations. These remaining coupling terms can be computed as follows:

$$\mathbf{A}_{AR} = \alpha\frac{\partial\mathbf{R}_A}{\partial\mathbf{q}_R}, \qquad \mathbf{A}_{AS} = \alpha\frac{\partial\mathbf{R}_A}{\partial\mathbf{X}_A}\frac{\partial\mathbf{X}_A}{\partial\mathbf{q}_S} + \beta\frac{\partial\mathbf{R}_A}{\partial\mathbf{V}_A}\frac{\partial\mathbf{V}_A}{\partial\dot{\mathbf{q}}_S}, \qquad \mathbf{A}_A = \alpha\frac{\partial\mathbf{R}_A}{\partial\mathbf{q}_A}.$$

Note that only the term $\mathbf{A}_{AS}$ requires communication between the aerodynamic and structural processors.

B. Preconditioning methods

The preconditioner for the approximate Newton–Krylov method is based on a simplified form of the matrix (26), formed by discarding certain components from the Jacobian. The discarded components include all terms that require communication between the aerodynamic and structural processors. As a result, the preconditioner is based on the following matrix:

$$
\mathbf{M}_p = \begin{bmatrix} \mathbf{D}_R + \mathbf{S}_R + \mathbf{A}_R & \mathbf{S}_{RS} & 0 \\ \mathbf{S}_{SR} & \mathbf{S}_S & 0 \\ 0 & 0 & \mathbf{A}_A \end{bmatrix},
\tag{27}
$$

and the application of the precondition can occur without communication between aerodynamic and structural processor groups.

To construct the preconditioner, it is necessary to first factor the block matrix associated with the structural processors, $\mathbf{S}_S$, and computing the preconditioner for the aerodynamic analysis. Once the matrix $\mathbf{S}_S$ is factored, the Schur complement, $\mathbf{G}$, of the structural and rigid-body degrees of freedom can be computed as follows:

$$
\mathbf{G} = (\mathbf{D}_R + \mathbf{S}_R + \mathbf{A}_R) - \mathbf{S}_{RS}\mathbf{S}_S^{-1}\mathbf{S}_{RS}.
\tag{28}
$$

Once these operations are completed, the application of the preconditioner on an input vector, $\mathbf{y} = \mathbf{M}_p^{-1}\mathbf{x}$, can be computed using the following operations:

$$
\begin{aligned}
\mathbf{t}_S &= \mathbf{S}_S^{-1}\mathbf{x}_S, \\
\mathbf{y}_R &= \mathbf{G}^{-1}(\mathbf{x}_R - \mathbf{S}_{RS}\mathbf{t}_S), \\
\mathbf{y}_S &= \mathbf{S}_S^{-1}(\mathbf{x}_S - \mathbf{S}_{SR}\mathbf{y}_R), \\
\mathbf{y}_A &= \mathbf{A}_A^{-1}\mathbf{x}_A,
\end{aligned}
\tag{29}
$$

where $\mathbf{t}_S$ is a temporary structural vector. Note that both disciplines can apply their portion of the preconditioner simultaneously since the aerodynamic and structural disciplines are located on separate processor groups.

We also construct the preconditioner for the transpose system using the same matrix as above (27). The preconditioner for the transpose system, $\mathbf{y} = \mathbf{M}_p^{-T}\mathbf{x}$, can be obtained using a similar approach by performing the following operations:

$$
\begin{aligned}
\mathbf{t}_S &= \mathbf{S}_S^{-T}\mathbf{x}_S, \\
\mathbf{y}_R &= \mathbf{G}^{-T}(\mathbf{x}_R - \mathbf{S}_{SR}^T\mathbf{t}_S), \\
\mathbf{y}_S &= \mathbf{S}_S^{-T}(\mathbf{x}_S - \mathbf{S}_{RS}^T\mathbf{y}_R), \\
\mathbf{y}_A &= \mathbf{A}_A^{-T}\mathbf{x}_A,
\end{aligned}
\tag{30}
$$

where the same Schur complement factorization can be used for both the forward and transpose preconditioning operations.

C. Additional adjoint terms

In addition to solving the adjoint equation (6), it is also necessary to compute the inner product of the adjoint solution vector with the derivative of the residuals with respect to the design variables. In this section, we outline the computation of these additional terms that are essential for the total derivative evaluation.

In practice we subdivide the design variables into three groups: the geometric design variables that modify both the aerodynamic and structural disciplines through the nodal locations, the material design variables which only directly affect the constitutive relationships within the structural finite-element discretization and the variables associated with the initial conditions of the analysis, such as the initial angle of attack or the initial velocity. We write this partition as:

$$
\mathbf{x}^T = \begin{bmatrix} \mathbf{x}_G^T & \mathbf{x}_M^T & \mathbf{x}_F^T \end{bmatrix}
\tag{31}
$$

where $\mathbf{x}_G$, $\mathbf{x}_M$, and $\mathbf{x}_F$ are the geometric, material and initial condition variables respectively.

The geometric design variables modify both the aerodynamic surface nodal locations $\mathbf{X}_A$ and the structural nodal locations $\mathbf{X}_S$. As a result, the product of the adjoint variables with the derivative of the residuals with respect to the geometric design variables has contributions from all three disciplines:

$$
\boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}_G} = \boldsymbol{\lambda}_R^T \frac{\partial \mathbf{R}_R}{\partial \mathbf{x}_G} + \boldsymbol{\lambda}_S^T \frac{\partial \mathbf{R}_S}{\partial \mathbf{X}_S} \frac{\partial \mathbf{X}_S}{\partial \mathbf{x}_G} + \boldsymbol{\lambda}_A^T \frac{\partial \mathbf{R}_A}{\partial \mathbf{X}_A} \frac{\partial \mathbf{X}_A}{\partial \mathbf{x}_G}
\tag{32}
$$

The material design variables, however, only modify the structural design variables and the inertial constants m, $\mathbf{c}$, and $\mathbf{J}$. As a result, the derivative with respect to the material design variables can be written as follows:

$$\boldsymbol{\lambda}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}_M} = \boldsymbol{\lambda}_R^T \frac{\partial \mathbf{R}_R}{\partial \mathbf{x}_M} + \boldsymbol{\lambda}_S^T \frac{\partial \mathbf{R}_S}{\partial \mathbf{x}_M} \tag{33}$$

Finally, the design variables $\mathbf{x}_F$ only modify the initial conditions for the first iteration. Therefore, the design sensitivity can be written as follows:

$$\boldsymbol{\lambda}_0^T \frac{\partial \mathbf{R}_0}{\partial \mathbf{x}_F} = \boldsymbol{\lambda}_{0R}^T \frac{\partial \mathbf{R}_{0R}}{\partial \mathbf{x}_F}, \tag{34}$$

where the contribution only arises from the first time step.

The total derivative (7) must be evaluated with the inner product of the derivative of the residuals with respect to the geometric (32), material (32) and initial-condition (32) design variables to obtain an accurate gradient.

VI.   Results

In the following section, we present results that demonstrate the efficiency of the both the solution method and the adjoint implementation presented in the preceding sections. For these studies we use a model based roughly on data for a Boeing 777-200ER and test the effect of increasing aspect ratio and span on gust response. For this study we use the three wings depicted in Figure 2, which have 60.9 m, 70 m, and 80 m spans, respectively. The aspect ratios for these wings, based on total projected area, are 8.7, 10.7, and 13.2, respectively. The aerodynamic model consists of a wing-tail configuration with a panel mesh consisting of 1440 surface panels, with 60 chord-wise and 20 span-wise panels on the main wing.
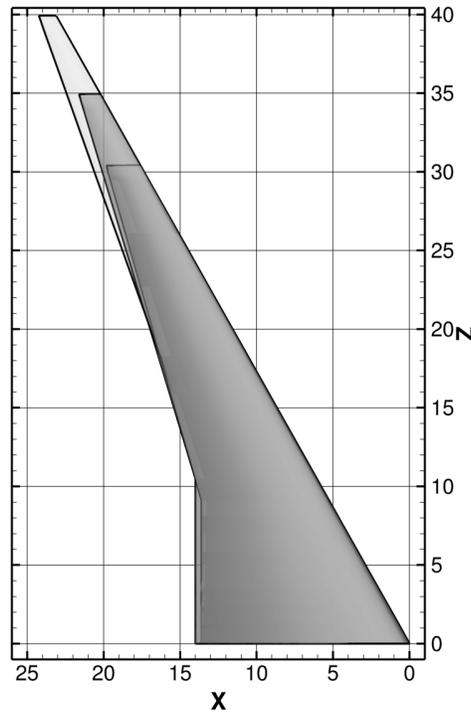

Figure 2: The wing planforms for the baseline, 70 m and 80 m wings.

For the purpose of this study, we use structural thickness distributions based on the results from a structural weight minimization design under fixed aerodynamic loads. Each wing-box consists of a two spar arrangement with a constant rib-pitch of 70 cm. The baseline, 70 m, and 80 m wings have 40, 46, and 53 ribs, respectively. Within the optimization problem we use six loading conditions obtained from three points in the maneuver envelope, 2.5g stall, 2.5g dive, and -1g dive, with two mass conditions at 100% and 10% fuel loads. Figure 3 shows the resulting thickness distributions for the baseline, 70 m, and 80 m wings, respectively. The structural models consist of a series of meshes with third-order MITC9 elements with between 140 000 and 160 000 degrees of freedom.
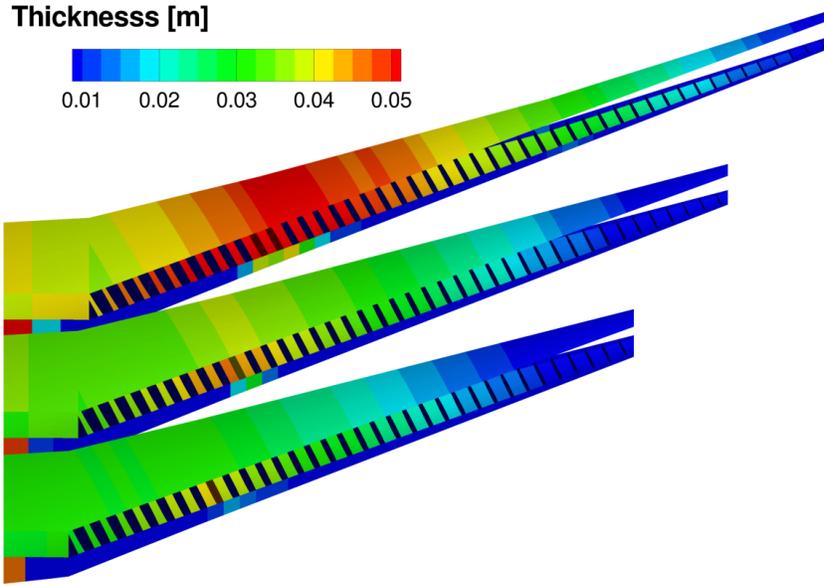
Figure 3: The optimal structural thickness distribution for the baseline, 70 m and 80 m wings, respectively

In the following series of simulations, the discrete gusts are simulated using a $1 - \cos$ gust distributions defined as follows:

$$V_{\text{wind}} = \frac{V_{\text{gust}}}{2} \left(1 - \cos \left(\frac{2\pi x}{H} \right) \right)$$

where V_{wind} is the local wind velocity at position x, and V_{gust} is the gust velocity taken here to be $V_{\text{gust}} = 20$ m/s. We use a series of three different gust lengths with $H = 20$, 50, and 100 m. The velocity profile of the different gusts used in the simulation are shown in Figure 4. Prior to the gust encounter, the aircraft is in steady, level flight at $M = 0.5$ at $20,000$ ft, resulting in an airspeed of 158 m/s. Based on this airspeed, the nominal gust durations are 0.19 s, 0.38 s, and 0.63 s, respectively. For the gust encounter simulations, we use 250 time steps, and a total simulation time of 3.15 seconds, which is 5 times the maximum gust length. As a result, the solution time step is $\Delta t = 0.013$.
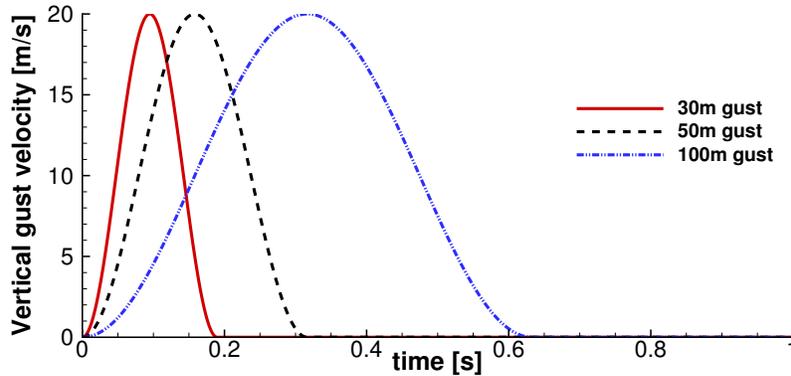
Figure 4: The gust velocity profiles for the 30 m, 50 m, and 100 m discrete gusts

A. Solution performance

In order to examine the computational performance of the Newton–Krylov solution method, we compare the computational time required to compute the first 50 time steps of the gust response on 4, 8, and 16 processors. In our studies thus far, we have found that the computational time of the Newton–Krylov approach is relatively insensitive to the flexibility of the structure, so in this study we present only the results for the baseline wing subject to the 30 m gust. To

American Institute of Aeronautics and Astronautics

examine the effect of modifying the aerodynamic and structural processor group sizes, we perform simulations with various processor group partitions for 3:1, 1:1, and 1:3 ratios of structural to aerodynamic processors.
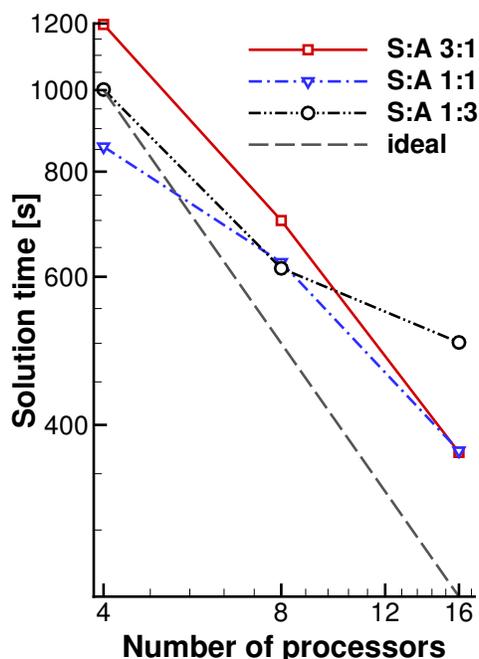

Figure 5: The aeroelastic gust simulation time for the 4, 8 and 16 processor cases with different aerodynamic and structural processor splits.

Figure 5 shows the solution time for the 4, 8, and 16 processor cases for all processor group partitions. The total solution time decreases for increasing numbers of processors however, the overall times do not exhibit ideal speed up behavior. Of all the partitions, the 1:1 split of structural and aerodynamic processor groups typically achieves the best solution times. Figure 6 shows the proportion of time spent in various operations performed during the solution procedure, normalized to the fastest result. The solution time is divided into four contributions: the time required to solve the linear system for the Newton update, the time required to compute the Jacobian and residual, the time required to compute coupling terms and, for only the structural processors, the time required to compute the factorization of the stiffness matrix. The time required to obtain the Newton-update and the computation of coupling terms both contribute to the less-than-ideal parallel performance of the algorithm. The relatively poor scalability of these computations can be traced to the load and displacement transfer scheme. The load and displacement transfer utilizes a collective communication paradigm that achieves excellent parallel efficiency for large, static aeroelastic problems [11], but does not yield good performance for the relatively smaller aeroelastic problems demonstrated here. In future, we will address this poor efficiency by improving the load and displacement transfer scheme by implementing an approach using non-blocking communication.

B.   Gradient-evaluation performance

In order to perform an assessment of the efficiency of the gradient calculation, we compare the computational time required to compute the gradient of four functions of interest for the baseline wing on 4, 8, and 16 processors. Figure 7 shows the speed up of the gradient calculation for the 4, 8, and 16 processor cases for different ratios of structural to aerodynamic processor group sizes. It should be noted that the parallel efficiency of these cases is superior to the parallel efficiency exhibited by the solution technique. This superior parallel efficiency is due in part to the larger proportion of time required to assemble the exact Jacobian at each iteration of the time-dependent adjoint method.

Figure 8 shows the proportion of time spent in the various stages of the gradient evaluation method. Note that a larger proportion of time is spent in the linear solution method for the adjoint-based gradient evaluation method than for the solution method shown in Figure 6. This can be attributed to the fact that the linear adjoint systems must be solved to a tighter tolerance to maintain gradient accuracy, and that four adjoint solutions must be computed at each step. Note that while the linear solution time will increase with the number of functions, the set up time remains constant and is therefore amortized with a small memory penalty due to the increase in storage requirements.
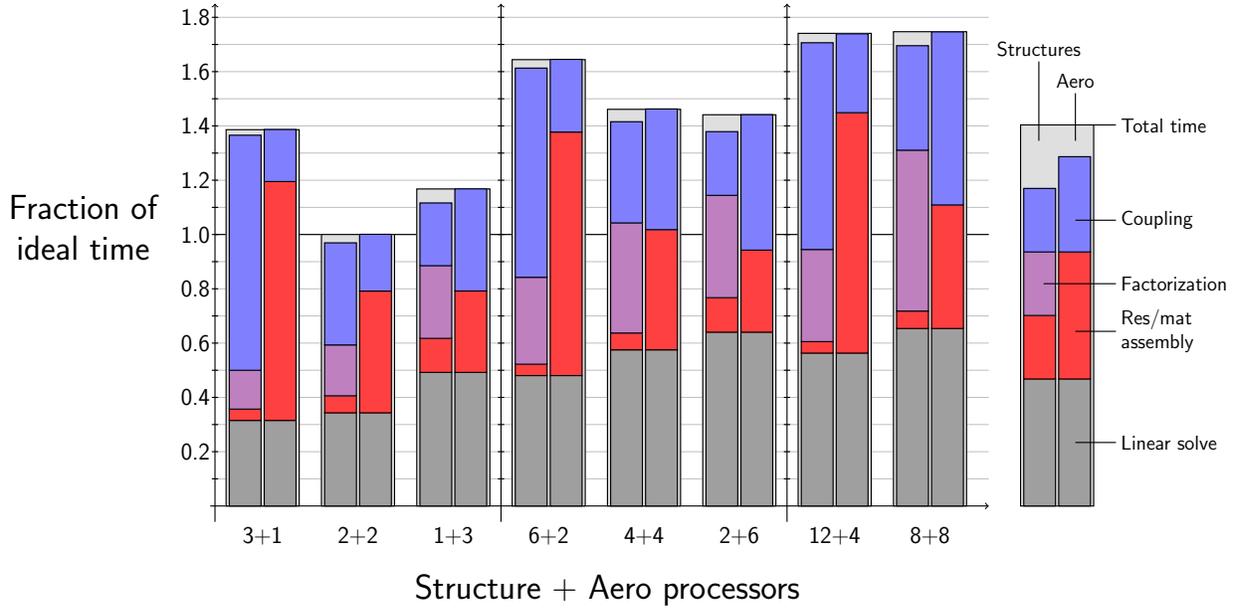
Figure 6: Solution performance breakdown for the first 50 time steps of 30 m gust encounter for the baseline wing for the 4, 8 and 16 processor cases.

C.   Accuracy verification

In this section we evaluate the accuracy of the adjoint implementation described above using a finite-difference comparison of a simple aeroelastic test case. For the purposes of this study, we use the aeroelastic wing model shown in Figure 9 which is a straight, untapered wing with an aspect ratio of 10. The aerodynamic model consists of a surface mesh with 600 surface panels and 630 aerodynamic nodes. The structural mesh consists of 576, third order MITC9 [35] shell elements, with 2130 nodes and approximately 12 700 structural degrees of freedom. The structural properties of the wing are based on aircraft-grade aluminium with a Young's modulus of 70 GPa, a Poisson's ratio of 0.3 and a density of 2700 kg/m^3. For the purposes of this preliminary verification we do not include inertial loads from fuel. We simulate a glide at sea-level conditions with an initial forward velocity of 65 m/s and an initial angle of attack of 1.3º.

For the purposes of this study, we test the adjoint implementation using a stress constraint applied at the final time step of the simulation. The purpose of the stress constraint is to limit the maximum von Mises stress within the structural model. Instead of using the maximum stress value directly, we use a Kreisselmeier-Stienhauser (KS) constraint aggregation technique which provides a differentiable estimate of the maximum von Mises stress in the finite-element mesh [36, 37]. The KS function is computed as follows,

$$f(\mathbf{x}, \mathbf{q}_N) = f_{\max} + \frac{1}{\rho} \ln \left[\sum \exp \left(\rho \left(\frac{\sigma_{vm}}{\sigma_{ys}} - f_{\max} \right) \right) \right], \tag{35}$$

where ρ is a weighting parameter that we set as $\rho = 20$ for the purposes of this study. Here, σ_{vm} is the von Mises stress, σ_{ys} is the yield stress and $f_{\max}$ is the maximum ratio, $f_{\max} = \max(\sigma_{vm}/\sigma_{ys})$, over the entire finite-element mesh. Note that the summation in Equation (35) is performed over all Gauss quadrature points within the finite-element mesh.

Figure 10 shows a comparison between the gradient obtained using the time-dependent adjoint method and the gradient obtained using a central finite-difference approximation with a step size of $\Delta x = 10^{-6}$. Here, we compare the gradient values for 10 structural thicknesses within the finite-element mesh. From Figure 10, it is clear that all gradient components match to between 2 and 3 digits of accuracy. It is difficult to assess whether this discrepancy is due to implementation errors within the adjoint method or is a result of the truncation or cancellation errors in the finite-difference technique. Nevertheless, this level of accuracy is sufficient for many simple optimization problems. In future, we will perform a better comparison of gradient accuracy using the complex-step method [38].
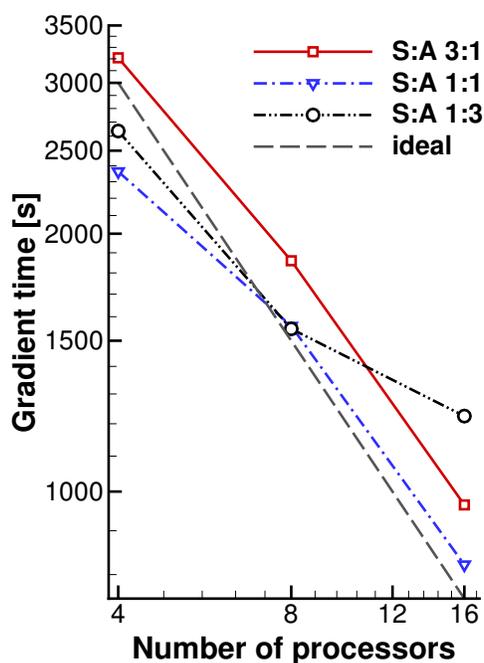
American Institute of Aeronautics and Astronautics

Figure 7: The adjoint-based derivative evaluation time for the 4, 8 and 16 processor cases with different aerodynamic and structural processor splits.

VII. Conclusions

High-fidelity static aeroelastic design optimization techniques have been successfully applied to aircraft design problems with thousands of shape, planform, and structural sizing variables. However, a significant issue with these results is that they neglect dynamic aeroelastic effects and potentially over-predict the benefits of wing-span extension without accounting for the additional weight penalties from adverse aeroelastic effects. Therefore, designs based on static aeroelastic optimization must be viewed as incomplete until their susceptibility to adverse aeroelastic phenomena is addressed. As a first step towards the development of a high-fidelity aeroelastic design tool, we have derived and verified a time-dependent aeroelastic optimization framework that uses a medium-fidelity panel method coupled to a high-fidelity structural finite-element model. The additional difficulties of aeroelastic design optimization are apparent: a large additional simulation cost due to stiff, time-dependent equations and the additional adjoint-evaluation cost. Nevertheless, the additional benefit of including aeroelastic effects within a high-fidelity framework outweigh the challenges: greater confidence in results, and fewer, costly late-stage design modifications required to fix unexpected aeroelastic behavior.

Acknowledgement

References

[1] Kennedy, G. J. and Martins, J. R. R. A., "A Comparison of Metallic and Composite Aircraft Wings Using Aerostructural Design Optimization," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, September 2012.

[2] Drela, M., "Integrated simulation model for preliminary aerodynamic, structural and control-law design of aircraft," *Proceedings of the 40th AIAA Structures Dynamics and Materials Conference*, St. Louis, MO, 1999, AIAA 99-1394.

[3] Haghighat, S., Martins, J. R. R. A., and Liu, H. H. T., "Aeroservoelastic Design Optimization of a Flexible Wing," *Journal of Aircraft*, Vol. 49, No. 2, 2012, pp. 432–443. doi:10.2514/1.C031344.

[4] Haghighat, S., Liu, H. H. T., and Martins, J. R. R. A., "A Model Predictive Gust Load Alleviation Controller for a Highly Flexible Aircraft," *Journal of Guidance, Control and Dynamics*, Vol. 36, 2012, pp. 1751–1766. doi:10.2514/1.57013.
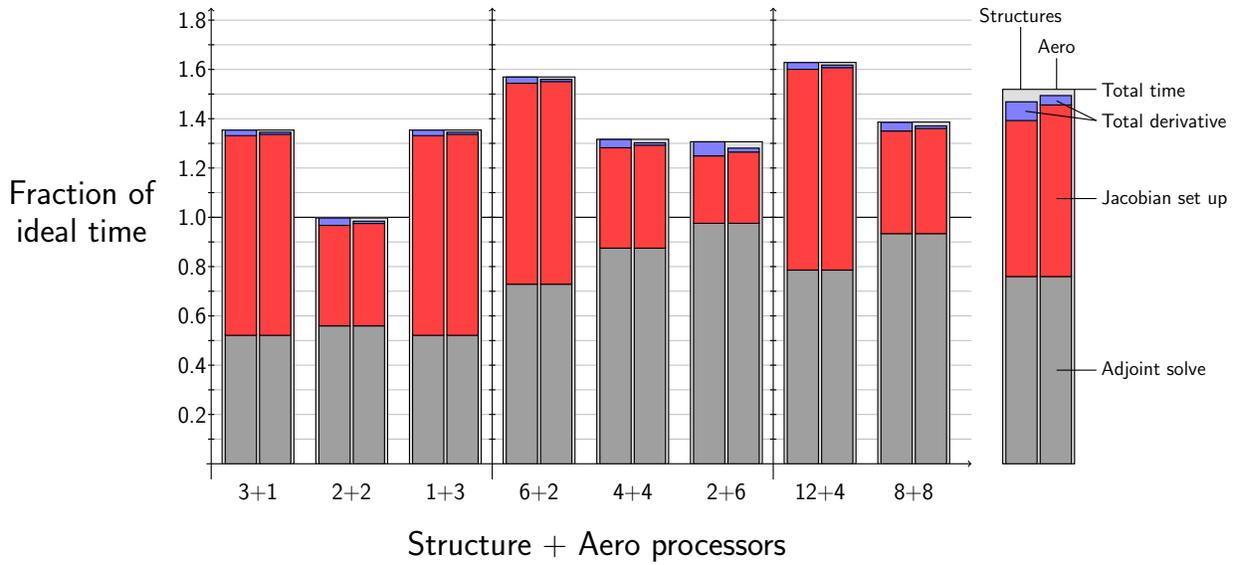
Figure 8: The adjoint-based derivative evaluation performance breakdown for the first 50 time steps of 30 m gust encounter for the baseline wing for the 4, 8 and 16 processor cases.
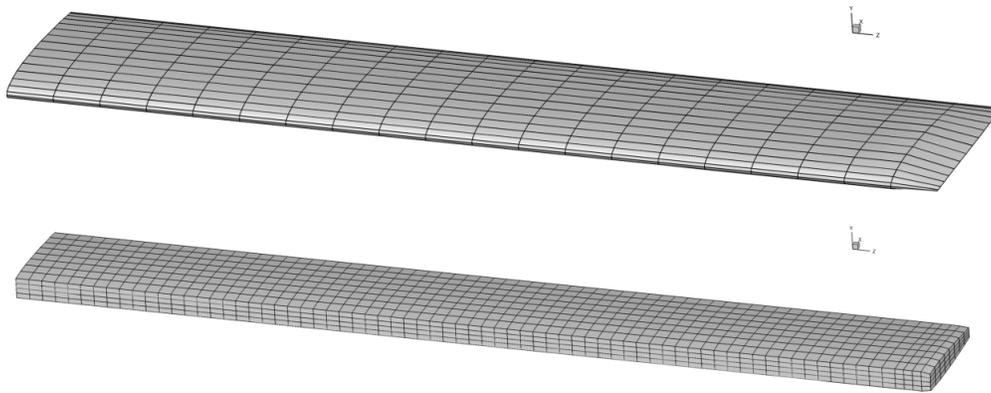

Figure 9: The aeroelastic model used for the finite-difference accuracy results.
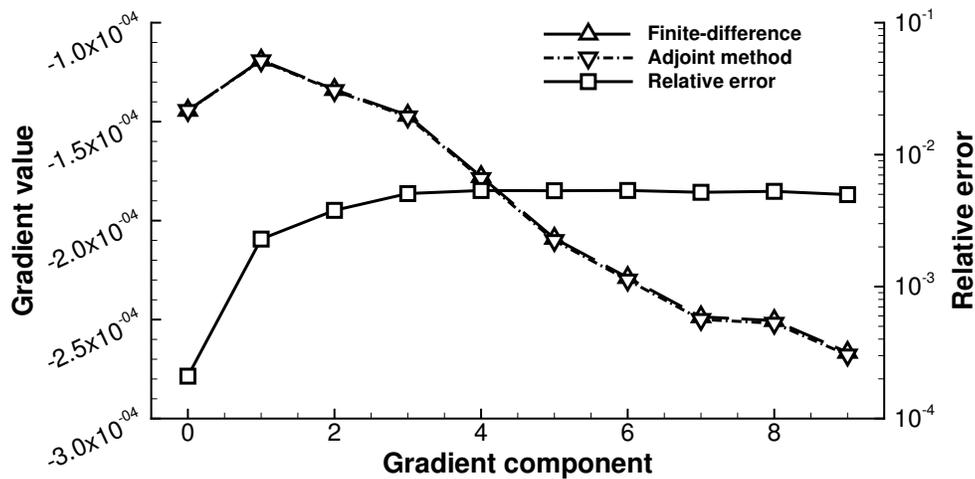

Figure 10: Comparison between the finite-difference results and the adjoint method using a step size of $\Delta x = 10^{-6}$.

American Institute of Aeronautics and Astronautics

[5] Shearer, C. M. and Cesnik, C. E., "Nonlinear Flight Dynamics of Very Flexible Aircraft," *Journal of Aircraft*, Vol. 44, 2007, pp. 1528 –1545. doi:10.2514/1.27606.

[6] Chimakurthi, S. K., Stanford, B. K., Cesnik, C. E. S., and Shyy, W., "Flapping Wing CFD/CSD Aeroelastic Formulation Based on a Corotational Shell Finite Element," *50th AIAA / ASME / ASCE / AHS / ASC Structures, Structural Dynamics, and Materials Conference*, May 2009, AIAA-2009-2412.

[7] Shearer, C. M. and Cesnik, C. E. S., "Trajectory Control for Very Flexible Aircraft," *Journal of Guidance, Control and Dynamics*, Vol. 31, No. 2, 2008. doi:10.2514/1.29335.

[8] Fidkowski, K., Engelsen, F., Willcox, K. E., and Kroo, I., "Stochastic Gust Analysis Techniques for Aircraft Conceptual Design," *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria BC, Canada*, September 2008, AIAA Paper Number 2008-5848.

[9] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "High-fidelity aerostructural design optimization of a supersonic business jet," *Journal of Aircraft*, Vol. 41, No. 3, 2004, pp. 523–530. doi:10.2514/1.11478.

[10] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., "A coupled–adjoint sensitivity analysis method for high–fidelity aero–structural design," *Optimization and Engineering*, Vol. 6, 2005, pp. 33–62. doi:10.1023/B:OPTE.0000048536.47956.62.

[11] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., "A Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations," *AIAA Journal*, 2013, In press.

[12] Liem, R. P., Kenway, G. K. W., and Martins, J. R. R. A., "Multi-point, multi-mission, high-fidelity aerostructural optimization of a long-range aircraft configuration," *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, September 2012.

[13] Kennedy, G. J. and Martins, J. R. R. A., "Parallel solution methods for aerostructural analysis and design optimization," *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, Fort Worth, TX, September 2010, AIAA 2010-9308.

[14] Jansen, P. and Perez, R., "Constrained structural design optimization via a parallel augmented Lagrangian particle swarm optimization approach," *Computers & Structures*, Vol. 89, No. 1314, 2011, pp. 1352 – 1366. doi:10.1016/j.compstruc.2011.03.011.

[15] Nadarajah, S. and Jameson, A., "Optimal control of unsteady flows using a time accurate method," *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference*, September 2002, AIAA 2002-5436.

[16] Nadarajah, S., McMullen, M., and Jameson, A., "Optimal control of unsteady flows using a time accurate and nonlinear frequency domain methods," *16th Computational Fluid Dynamics Conference*, June 2003, AIAA 2003-3875.

[17] Rumpfkeil, M. P. and Zingg, D., "The optimal control of unsteady flows with a discrete adjoint method," *Journal of Optimization and Engineering*, Vol. 11, 2010, pp. 5–22. doi:10.1007/s11081-008-9035-5.

[18] Rumpfkeil, M. P. and Zingg, D., "A Hybrid Algorithm for Far-Field Noise Minimization," *Computers and Fluids*, Vol. 39, 2010, pp. 1516–1528. doi:10.1016/j.compfluid.2010.05.006.

[19] Kennedy, G. J. and Hansen, J. S., "The hybrid-adjoint method: A semi-analytic gradient evaluation technique applied to composite cure cycle optimization," *Optimization and Engineering*, Vol. 11, 2010, pp. 23–43. doi:10.1007/s11081-008-9068-9.

[20] Mader, C. A. and Martins, J. R. R. A., "Derivatives for Time-Spectral Computational Fluid Dynamics Using an Automatic Differentiation Adjoint," *AIAA Journal*, Vol. 50, 12/2012 2012, pp. 2809–2819. doi:10.2514/1.J051658.

[21] Hughes, T. J. R. and Brezzi, F., "On drilling degrees of freedom," *Computer Methods in Applied Mechanics and Engineering*, Vol. 72, 1989, pp. 105–121.

[22] Fox, D. and Simo, J., "A drill rotation formulation for geometrically exact shells," *Computer Methods in Applied Mechanics and Engineering*, Vol. 98, No. 3, 1992, pp. 329 – 343. doi:10.1016/0045-7825(92)90002-2.

[23] Saad, Y., *Numerical methods for large eigenvalue problems*, Algorithms and architectures for advanced scientific computing, Manchester University Press, 1992.

[24] Meirovitch, L. and Stemple, T., "Hybrid equations of motion for flexible multibody systems using quasicoordinates," *Journal of Guidance, Control and Dynamics*, Vol. 18, No. 4, 1995, pp. 678–688.

[25] Hughes, P. C., *Spacecraft attitude dynamics*, Dover books on engineering, Dover Publications, 2004.

[26] Brown, S., "Displacement extrapolation for CFD+CSM aeroelastic analysis," AIAA Paper 97-1090, 1997.

[27] Katz, J. and Plotkin, A., *Low–Speed Aerodynamics*, McGraw–Hill Inc., 1991.

[28] Hess, J. and Smith, A., "Calculation of potential flow about arbitrary bodies," *Progress in Aerospace Sciences*, Vol. 8, 1967, pp. 1–138.

[29] Smith, S., "A computational and experimental study of nonlinear aspects of induced drag," Tech. Rep. NASA TP 3598, National Aeronautics and Space Administration, Ames Research Center, Moffett Field, CA, 94035-1000, 1996.

American Institute of Aeronautics and Astronautics

[30] Balay, S., Buschelman, K., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Smith, B. F., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[31] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[32] Karypis, G. and Kumar, V., "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, Vol. 20, No. 1, 1998, pp. 359–392.

[33] Saad, Y. and Schultz, M. H., "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. doi:10.1137/0907058.

[34] Saad, Y., "A fexible inner-outer preconditioned GMRES algorithm," *SIAM Journal on Scientific Computing*, Vol. 14, No. 2, 1993, pp. 461–469. doi:10.1137/0914028.

[35] Bathe, K.-J. and Dvorkin, E. N., "A formulation of general shell elements – the use of mixed interpolation of tensorial components," *International Journal for Numerical Methods in Engineering*, Vol. 22, 1986, pp. 697–722. doi:10.1002/nme.1620220312.

[36] Wrenn, G., "An indirect method for numerical optimization using the Kreisselmeier-Steinhauser function," NASA Technical Report CR-4220, 1989.

[37] Poon, N. and Martins, J. R. R. A., "An adaptive approach to constraint aggregation using adjoint sensitivity analysis," *Structural and Multidisciplinary Optimization*, Vol. 34, 2007, pp. 61–73. doi:10.1007/s00158-006-0061-7.

[38] Martins, J. R. R. A., Sturdza, P., and Alonso, J. J., "The complex-step derivative approximation," *ACM Transactions on Mathematical Software*, Vol. 29, No. 3, Sept. 2003, pp. 245–262. doi:10.1145/838250.838251.