

This is a preprint of the following article, which is available from <http://mdolab.engin.umich.edu>

Gaetan K. W. Kenway, Graeme J. Kennedy, and Joaquim R. R. A. Martins, "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations", *AIAA Journal*, 52(5):935-951, 2014. doi:10.2514/1.J052255

The published article may differ from this preprint, and is available by following the DOI above.

Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations

Gaetan K. W. Kenway,¹ Graeme J. Kennedy,² Joaquim R. R. A. Martins³

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

Abstract Aeroelastic systems achieve the best performance when the aerodynamic shape and structural sizing are optimized concurrently, but such an optimization is challenging when high-fidelity aerodynamic and structural models are required. This paper addresses this challenge through several significant improvements. Fully coupled Newton–Krylov methods are presented for the solution of aerostructural systems and for the corresponding adjoint systems. The coupled adjoint method presented can compute gradients with respect to thousands of multidisciplinary design variables accurately and efficiently. This is enabled by several improvements in the computation of the multidisciplinary terms in the coupled adjoint. The parallel scalability of the methods is demonstrated for a full aircraft configuration using an Euler computational fluid dynamics model with more than 8×10^6 state variables and a detailed structural finite element model of the wing with more than 1×10^6 degrees of freedom. The coupled Newton–Krylov methods are shown to improve the convergence rate of both the aerostructural solution and the coupled adjoint derivative computations. Gradient computations of aerodynamic and structural functions with respect to both aerodynamic shape and structural sizing variables are verified, and scaling is demonstrated to $\mathcal{O}(10^3)$ variables. The accuracy and scalability of the presented methods make it possible to perform aerostructural optimizations of full aircraft configurations with respect to hundreds of external shape and structural sizing design variables, leading to optimal aeroelastic tailoring.

Nomenclature

\mathcal{A}	aerodynamic residuals
C_p	pressure coefficient
F	structural load vector
F_A	aerodynamic load vector
I	function of interest
K	structural stiffness matrix
$K_{\mathcal{M}}$	mesh stiffness matrix
N_P	number of processors
\mathcal{R}	all residuals, $\mathcal{R}^T = [\mathcal{A}^T, \mathcal{S}^T]$
$R_{\mathcal{M}}$	mesh restriction operator
r	vector between computational fluid dynamics surface mesh point and structural mesh

\mathcal{S}	structural residuals
T	displacement transfer matrix
T^T	load transfer matrix
u	structural state variables
u_A	displacements of computational fluid dynamics surface mesh
u_r	rotational components of displacement
u_t	translational components of displacement
\mathcal{W}	mesh deformation operator
w	aerodynamic state variables
X_J	jig shape of computational fluid dynamics surface mesh or computational structural mechanics mesh
X_S	spatial coordinates of computational fluid dynamics surface mesh
X_{SW}	intermediate mesh from Bézier cubic spline interpolation
X_V	spatial coordinates of computational fluid dynamics volume mesh
x	design variable vector
Y_S	subset of Y_V with specified displacements
Y_V	coarse version of volume mesh, X_V
ϵ_A	convergence tolerance for aerodynamic solution
ϵ_{AS}	convergence tolerance for aerostructural solution
ϵ_S	convergence tolerance for structural solution
ϵ_{SA}	convergence tolerance for aerostructural adjoint solution
θ	under-relaxation factor
ϕ	structural adjoint vector
ψ	aerodynamic adjoint vector

1 Introduction

Coupling aerodynamic and structural numerical models to compute the static aeroelastic shape of lifting surfaces is essential when designing lifting surfaces that are flexible. Even small changes in shape can have a large effect on the aerodynamic performance, and multiple flow conditions result in multiple shapes. This is particularly important for swept wings, where bend-twist coupling can result in large changes in the twist distribution [1]. The analysis of static aeroelastic shapes was introduced as soon as the first simple numerical models for aerodynamics and structures matured. Skoog Brown [2], for example, coupled a lifting line model to a beam theory model to obtain the flying shape of a swept wing, and earlier work exists that considered even simpler models [3]. In recent years, this coupling has become even more important, because the trend has been to increase the aspect ratio of aircraft wings, making them more flexible. Wing flexibility impacts not only the static flying shape of the wing but also its dynamics, resulting in aeroelastic phenomena such as flutter and aileron reversal. In the present work, however, we restrict ourselves to the analysis of the static aeroelastic shape, which we refer to as *aerostructural* analysis.

The design of aircraft wings has benefited greatly from the use of numerical optimization techniques [4, 5], and the need to include the relevant disciplines in such design problems led to the field of multidisciplinary

design optimization (MDO) [6]. One of the earliest examples of aerostructural optimization was by Haftka [7], who combined a lifting line aerodynamic model with structural finite-element analysis to iteratively obtain the deformed shape and minimize the weight subject to drag and stress constraints. This enabled the comparison of trade-offs between structural weight and induced drag for aluminum and composite wings.

Grossman *et al.* investigated the aerostructural analysis and optimization of a sailplane [8] and a subsonic forward-swept transport wing [9]. They showed that a sequential optimization approach—where aerodynamic optimization and structural sizing are performed iteratively in sequence—produced a suboptimal result relative to a fully integrated aerostructural optimization. This failure of sequential optimization to produce the optimal result is further explained by Chittick and Martins [10].

With the advent of higher fidelity modeling in both structures and aerodynamics, numerical optimization has been extensively applied to each discipline separately. On the structures side, increasingly detailed finite-element models have been used in wing structural sizing optimization [11]. The increase in fidelity in the structural model is required for a more refined sizing of the structure while considering complex structural failure constraints, leading to a better estimate of the optimized structural weight. The fidelity of the models used for aerodynamic shape optimization has also been increasing, and it is now possible to use computational fluid dynamics (CFD) to optimize a design with respect to hundreds of design variables [12, 13, 14, 15, 16, 17]. In the design of transonic wings it is particularly important to use high-fidelity models to correctly predict the drag. In addition, to take advantage of these models, large numbers of airfoil shape variables are required to effectively reduce the wave drag of a wing. Unfortunately, aerodynamic shape optimization alone is insufficient for wing design, since it is impossible to perform the trade-offs for wing thickness, span, and sweep, which require a model of how the wing weight varies with respect to these parameters. To handle large numbers of design variables, the efforts above employed gradient-based optimization algorithms together with adjoint methods to compute the required gradients efficiently.

Given the importance of coupling the aerodynamics and the structures in wing design, coupling high-fidelity versions of these models for analysis and design optimization is a natural extension of the work cited so far. Various techniques have been proposed over the years for coupling CFD to computational structural mechanics (CSM) solvers, with contributions in load and displacement transfer schemes [18, 19, 20, 21] and solution techniques for solving the coupled system of equations [22, 23].

On the design optimization side, Maute *et al.* [24] pioneered high-fidelity aerostructural optimization by coupling an Euler flow solver to a linear finite-element model of the structure. They developed a nonlinear block Gauss–Seidel method with relaxation for the solution of the aerostructural system, and they used a three-field formulation to deform the CFD mesh by a spring analogy method. They demonstrated the method by performing an optimization with respect to five design variables using gradients computed with the direct method. The number of design variables was limited to such a small number because the cost of computing the gradient with the direct method is proportional to the number of design variables. Martins *et al.* [25] proposed the use of a coupled adjoint method for aerostructural design optimization using Euler CFD and linear finite-element analysis in a two-field formulation. They showed that the cost of computing the gradient using this method could be made nearly independent of the number of design variables, and they computed gradients with respect to thousands of variables [26]. Then, they applied this method to the aerostructural design of a supersonic business jet with respect to 97 shape and sizing variables [27]. One of the main advances in this work was the elimination of the minimum airfoil thickness constraints that must usually be enforced in aerodynamic shape optimization. By including the structure and simultaneously optimizing the wing with respect to the aerodynamic shape and structural sizing, the optimization was able to trade-off between drag and structural weight to naturally determine the optimal thickness-to-chord ratio. The consideration of these aerostructural trade-offs also enables the optimization of wing planform variables, such as span and sweep, but that work did not include these variables due to limitations in the CFD mesh movement. In addition, there were several other shortcomings that we address in the present work. While the implementation of the coupled adjoint method was shown to scale well with the number of design variables, it did not scale well with the number of structural surface degrees of freedom, limiting the complexity of the structural model. This was in part because some of the partial derivatives in the coupled adjoint terms were computed using finite differences, which also limited the accuracy of the derivatives. Furthermore, the structural solver was a serial code with a limited choice of element types. Finally, the optimization took into consideration only two flight conditions: one for cruise performance and another for a maneuver condition.

Maute *et al.* [28] presented another coupled adjoint formulation using discrete-analytical derivatives.

They compared it to the direct method and showed that the accuracy of the two methods is identical. To improve the robustness and efficiency of aerostructural solution methods, Barcelos *et al.* [29] developed a class of Newton–Krylov–Schur methods for solving the coupled nonlinear fluid-structure-mesh deformation problem. Their approach used an approximate Newton method for the solution of the nonlinear coupled equations. At each iteration, a Schur-complement approach is used to solve the coupled linear system that results from a linearization of the residual. They found that their technique is more robust and efficient than the original Gauss–Seidel method by Maute *et al.* [24]. More recently, Barcelos *et al.* [30] presented an aerostructural solution technique that couples the RANS equations to a linear structural model and a mesh-deformation strategy. They used the direct method for computing the coupled derivatives, and they applied it to solve an optimization problem with five design variables.

There has been a continued interest in the development of coupled adjoint methods. For example, Brezillon *et al.* [31] describe ongoing work at DLR, German Aerospace Center, related to high-fidelity aerostructural analysis and optimization capabilities. Similar efforts are described by Ghazlane *et al.* [32]. In both of these efforts, the number of design variables and flight conditions was limited.

In spite of all these developments over the last decade, we are still missing a high-fidelity fully integrated aerostructural design optimization approach that is scalable with respect to the number of design variables and to the number of degrees of freedom in the aerodynamic and structural models. Meeting these requirements is a challenging proposition, as evidenced by the fact that even after more than a decade of strong interest by several research groups and industry, the coupled adjoint method has still not been successfully applied to problems with hundreds of design variables. With the exception of Martins *et al.* [26], the scaling of the computational cost of the coupled adjoint solution with respect to the number of design variables has not been reported. The coupled aerostructural analysis must also be fast enough so that multiple flight conditions and all relevant load cases are considered. This is needed to realize the full potential of this approach to produce practical wing designs with optimal static aeroelastic tailoring.

Thus, the objective of the present work is to develop an approach that enables the simultaneous design optimization of aerodynamic shape and structural sizing with respect to hundreds (or even thousands) of variables with as many practical considerations as possible. In this paper, we focus on the description and benchmarking of the coupled solution approach and the coupled adjoint method. The application to aerostructural design optimization of an aircraft configuration is presented in separate paper by the authors [33].

Our approach follows the work of Martins *et al.* [27, 26, 34], but addresses the shortcomings mentioned above. We use a more advanced flow solver with an automatic differentiation adjoint (ADjoint) [35] and a new parallel structural solver that also has adjoint sensitivity analysis capability [36]. We eliminate the high computational cost and the low accuracy that result from evaluating the off-diagonal coupled adjoint terms with finite differences. We achieve this by computing all the partial-derivative terms required to form the coupled adjoint equations using only methods that are accurate to machine precision. Depending on the particular partial-derivative term, a combination of analytic, forward, or reverse algorithmic differentiation (AD) methods is used, and special care is taken to ensure the scalability of the parallel computations.

When it comes to accuracy, no previous work on coupled adjoint methods for large scale CFD and CSM has presented a definite verification of the coupled aerostructural derivatives. As we show in this paper, finite differencing may verify derivatives to a few digits, but it is not sufficient to demonstrate that the coupled adjoint implementation is entirely consistent.

A more sophisticated mesh-deformation scheme and a corresponding adjoint are developed, where the partial derivative terms are computed using the reverse mode of AD. This enables the mesh-deformation derivatives to be computed with a cost independent of the number of structural degrees of freedom and the number of design variables. These improvements allow us to consider substantially larger finite-element structural models with $\mathcal{O}(10^6)$ degrees of freedom and aerodynamic models with $\mathcal{O}(10^7)$ degrees of freedom, which is unprecedented in aerostructural design optimization.

Being able to handle this many degrees of freedom enables us to use a model of the aircraft wingbox structure that is sufficiently detailed to make accurate predictions of the quantities of interest. The model is able to compute accurate displacements to ensure that the correct wing flying shape, and also computes stresses accurately to insure that the failure constraints are realistic. Finally, the structural model provides a better estimate of how the wingbox mass changes with respect to structural and geometric design variables.

We also present improvements to the solution of the coupled system and its adjoint. To increase the efficiency of the coupled solver, we implement Aitken acceleration in the nonlinear block Gauss–Seidel method and we introduce a fully coupled Newton–Krylov solution approach that can handle highly flexible aircraft structures. To improve the solution of the aerostructural adjoint equations, we propose a new coupled-Krylov approach.

Thus, the coupled adjoint approach presented herein allows us to compute the gradient of a function of interest with respect to thousands of design variables in approximately half the time required for one aerostructural solution. In turn, this will enable the multi-point aerostructural design optimization of aircraft configurations, which is the focus of another paper by the authors [33]. Some progress has also been made in applying this methodology to multi-mission aerostructural design optimization [37].

The remainder of this paper is organized as follows. Section 2 describes the aerodynamic solver, the structural solver, and how they are coupled, providing the details of the load and displacement transfers, as well as the various coupled solution strategies. Section 3 derives the coupled adjoint equations and explains how they are implemented and solved. The results are presented in Section 4, where we verify the gradients, compare the various solution methods, and demonstrate the scalability of the coupled adjoint method.

2 Aerostructural Analysis

This section outlines the components of the aerostructural analysis. Euler CFD is used for the aerodynamic analysis and a linear finite-element model is used for the structural analysis. In addition to describing the two disciplinary solvers, we also describe the load and displacement transfer technique, the mesh-deformation approach, and two strategies for solving the coupled system.

1 Aerodynamic Solver

Most large commercial aircraft operate in the transonic flight regime and thus, at a minimum, the Euler equations must be solved to estimate the induced drag and wave drag on the aircraft. Although the inclusion of viscous effects is obviously important to accurately predict the overall drag, significant insight can be gained by using the Euler equations. The aerodynamic solver used in this work is SUMb, which solves the Euler, laminar Navier–Stokes, or Reynolds averaged Navier–Stokes (RANS) equations in either steady, unsteady, or time-spectral modes [38]. SUMb was originally written for the analysis of large-scale turbo-machinery flows but has also successfully been employed for external flow applications [15]. SUMb employs the finite-volume method on structured, body fitted, multi-block grids. Once the equations are discretized, SUMb solves a set of nonlinear equations with the following form:

$$\mathcal{A}(w) = 0, \quad (1)$$

where w is the vector of flow state variables. The aerodynamic residuals \mathcal{A} include both the inviscid and artificial dissipation terms. When the solver is used for CFD analysis only, the equations are solved using a preconditioned matrix-free Newton–Krylov approach that was recently implemented in SUMb [39]. GMRES [40] is used to approximately solve the successive linear systems. A suitable initial iterate that is sufficiently close to the basin of attraction is obtained using a five-stage explicit Runge–Kutta multigrid time stepping scheme.

2 Structural Solver

The structural solver used in this work is the Toolkit for the Analysis of Composite Structures (TACS) [36]. This is a parallel finite-element solver that includes both static linear and geometrically nonlinear analysis capabilities. However, only linear analysis is considered in this work. TACS includes an adjoint solver that is able to handle the structural design variables, which in our case are the thicknesses of the structural members. Parallelism is achieved within TACS by using an element-based partitioning of the finite-element mesh. This partitioning is used to parallelize the factorization of the stiffness matrix, the computation and assembly of the stiffness matrix and structural residuals, and the computation of the functions of interest and their derivatives. Typically, the factorization of the stiffness matrix is the most costly operation. To parallelize the matrix factorization and back solutions, TACS uses a Schur-complement-based parallel direct solver. In this technique, each processor independently computes the local contribution to a reduced linear system that is formed from the all unknowns on the domain interface. This reduced problem is the global

Schur complement. In TACS, the local contributions to the global Schur complement are computed using a block-based parallel factorization. After the global Schur complement is computed, it is factored in parallel using a sparse block-cyclic algorithm that achieves excellent parallel performance. This direct method enables us to solve poorly conditioned thin-shell structural problems with condition numbers $\mathcal{O}(10^9)$ in an efficient manner. For general nonlinear analysis the structural discipline residuals are

$$\mathcal{S}(u) = 0 \quad (2)$$

where u is the vector of structural displacements. For linear analysis, this equation can be written as $\mathcal{S}(u) = Ku - F$, where K is the linear stiffness matrix and F is the load vector.

3 Load and Displacement Transfer

To perform aerostructural analysis, we must couple the two disciplines to form a single analysis capable of determining the shape and aerodynamic characteristics of a lifting surface for a given flight condition. Specifically, we must determine how the loads computed by the aerodynamic analysis are transferred to the structural analysis (load transfer) and how the displacements computed by the structural analysis deform the wetted aerodynamic surface (displacement transfer).

Our load and displacement transfer scheme follows the work of Brown [18], later employed by Martins *et al.* [26]. In this approach, rigid links are used to extrapolate the displacements from the structural surface to the CFD surface, as shown in Fig. 1. These rigid links are constructed between the aerodynamic surface mesh points and the points on the structural model lying closest to this set of points. The consistent force vector is determined by employing the method of virtual work, ensuring that the force transfer is conservative. The integration of the forces is performed on the aerodynamic mesh and is transmitted back through the rigid links to the structure. The two primary advantages of this scheme are that it is consistent and conservative by construction, and that it may be used to transfer loads and displacements between aerodynamic and structural meshes that are not coincident.

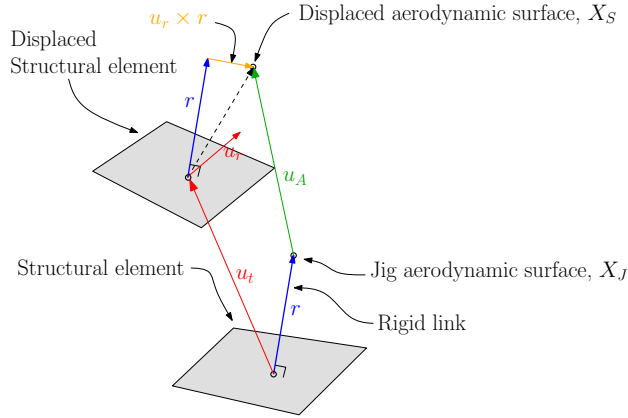


Figure 1: Load-displacement transfer operation

The design jig shape, X_J , is uniquely determined by the optimization design variables x . A single coordinate on the perturbed surface, X_S , is given by

$$\begin{aligned} X_S &= X_J + u_A \\ &= X_J + u_t + u_r \times r \end{aligned} \quad (3)$$

where r is the vector that connects a point on the aerodynamic surface to the closest location on the structural mesh, u_t is the translational component of the displacement, and u_r is the small-angle approximation of the rotations in the global reference frame, as shown in Fig. 1. We can compactly represent the displacement transfer from u to u_A as

$$u_A = Tu = \left(\frac{\partial u_A}{\partial u} \right) u \quad (4)$$

where T is the generalized transfer matrix. To transfer loads from the aerodynamic discipline to the structural discipline, we employ the transpose operation,

$$F = T^T F_A = \left(\frac{\partial F}{\partial F_A} \right) F_A \quad (5)$$

where F_A are the forces on the CFD nodes. More details on these transfers can be found in Brown [18] and Martins *et al.* [26].

4 Mesh-Deformation Algorithm

CFD-based aerodynamic shape optimization requires the perturbation of the mesh to consider changes in the shape design variables. However, unless the surface perturbations are of the same order as that of the mesh off-wall spacing, at least a portion of the volume mesh must be modified in order to prevent the formation of cells with negative volumes. Aerostructural analysis imposes additional mesh-deformation costs that are not present in aerodynamic shape optimization. Whereas aerodynamic shape optimization deforms the mesh only once per design iteration, aerostructural analysis requires a mesh deformation for every displacement transfer. Hence, up to $\mathcal{O}(10^2)$ mesh deformations may be necessary for an aerostructural analysis, depending on the multidisciplinary solution strategy. Thus, it is extremely important to have an efficient mesh-deformation algorithm. We use a hybrid algebraic-linear elasticity method similar to the work of Martins *et al.* [34]. The idea behind this approach is to apply a linear-elasticity-based scheme to a coarse approximation of the mesh to account for large low-frequency perturbations, and to use an algebraic deformation approach to attenuate small high-frequency perturbations.

The mesh perturbation proceeds by first selecting the nodes in a coarse version of the mesh, Y_V , which are called *supernodes*. This is accomplished using a restriction operator, $R_{\mathcal{M}}$:

$$Y_V = R_{\mathcal{M}} X_V. \quad (6)$$

The locations of the unknown interior supernodes are determined by solving a 3D pseudo-structural linear elasticity problem using solid elements formed by the supernodes. Prescribed displacement conditions are used at all boundary nodes, Y_S , and are determined as follows: 1. The nodal displacements on the aerodynamic surfaces are determined from the design variables and structural state vector. 2. The nodes on the far-field boundaries remain fixed, while the nodes on symmetry conditions are constrained to remain in the plane of symmetry. This symmetric linear system of equations can be written

$$K_{uu} Y_V - K_{us} Y_S = 0 \quad (7)$$

where the matrices K_{uu} and K_{us} are components of the mesh stiffness matrix, $K_{\mathcal{M}}$, when the unknown degrees of freedom are placed first, followed by the specified degrees of freedom, as shown below:

$$K_{\mathcal{M}} = \begin{bmatrix} K_{uu} & K_{us} \\ K_{su} & K_{ss} \end{bmatrix}. \quad (8)$$

Next, an intermediate mesh, X_{SW} , is computed using the coarse approximation with a trivariate Bézier cubic spline interpolation algorithm. Finally, any deviations between the desired surface and the intermediate interpolation approximation are attenuated using a modified transfinite interpolation algorithm operating on the difference between the two surfaces. A summary of the full mesh-deformation procedure is shown in Fig. 2.

The goal of this scheme is to obtain a high-quality perturbed mesh—like that obtained using a linear elasticity scheme—for all cells, but with a much lower computational cost. The mesh-deformation algorithm employed in this work is described in more detail in previous work by the authors [41].

The current approach can replicate the approach of Martins *et al.* [34] by selecting only the block corners as supernodes. However, if we allow more supernodes, the larger finite-element problem can more easily capture large surface modifications due to either design variables or structural displacements, especially when large rotations are involved. Additionally, the current implementation is fully parallel, making it scalable and greatly reducing the time required to solve the linear system of equations and the time required to repopulate the deformed mesh. This implementation has the advantage that all surface deformations are modeled exactly.

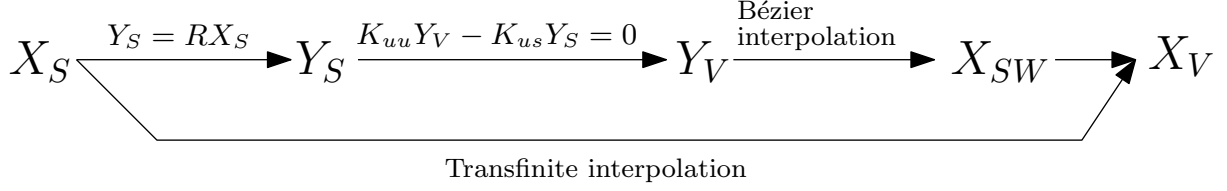


Figure 2: Solid mesh warping procedure

5 Aerostructural Analysis Methods

We formulate the aerostructural analysis problem using a two-field formulation. The aerodynamic analysis is the first field and the structural analysis is the second. Using this formulation, we can write the governing equations of both disciplines as a function of the fluid states, w , structural states, u , and design variables x . The latter is a vector consisting of global variables and local variables. Global variables affect the two disciplines directly, while local variables affect only a single discipline in a direct manner. Geometric variables that change the aircraft wetted surface—or outer mold line (OML)—are global variables. For example, an airfoil thickness variable will not only change the aerodynamic shape but also the height of the ribs and spars in the internal structure. Local variables include the angle of attack, which directly affects only the aerodynamics, and the thicknesses of the structural members, which affect only the structures. Combining the residual equations from the aerodynamic and structural disciplines, previously introduced in Eqs. (1) and (2), we write the combined residual of the multidisciplinary system as

$$\mathcal{R} = \begin{bmatrix} \mathcal{A}(w, u; x) \\ \mathcal{S}(u, w; x) \end{bmatrix} = 0. \quad (9)$$

The aerostructural analysis consists in finding a solution, (w, u) , that satisfies these coupled residual equations.

5.1 Nonlinear Block Gauss–Seidel Method (NLBGS)

The traditional process for solving the coupled aerostructural equations (9) is to use a nonlinear block Gauss–Seidel (NLBGS) method [26, 24]. In this approach, the aerodynamic analysis is first partially converged and the aerodynamic forces are evaluated. The forces are then transferred to the structural analysis and the corresponding displacements are computed. Finally, the displacements are transferred back to the aerodynamic analysis, the mesh is deformed, a new CFD solution is found, and this iterative loop continues until the coupled convergence criterion is met. The NLBGS procedure is listed in Algorithm 1.

Three tolerances are defined for the coupled analysis. The relative tolerances, ϵ_A and ϵ_S , are the tolerances required by the aerodynamic and structural disciplines, respectively, for each NLBGS iteration. The aerodynamic solver tolerance is typically $\mathcal{O}(10^{-1})$, while ϵ_S is typically $\mathcal{O}(10^{-3})$ or smaller. The third tolerance, ϵ_{AS} , is the aerostructural solution tolerance and represents the feasibility of the interdisciplinary coupling. Typical values for ϵ_{AS} range from 10^{-3} for an engineering solution accurate to three decimal places to 10^{-6} , which is the typical value used when performing design optimization. The CFD convergence is evaluated at the beginning of the k^{th} iteration using the structural displacements and aerodynamic states from the previous iteration. The structural convergence is evaluated at the beginning of the structural solution using the previous structural states with the current force vector. Aitken acceleration [42] (Line 22 of Algorithm 1) is employed to dynamically choose the under-relaxation factor to accelerate convergence. One advantage of this method is that each disciplinary solver can be used without modification. For tightly coupled aerostructural problems with large displacements, however, this method may converge slowly or not at all, as observed by Barcelos *et al.* [29].

5.2 Coupled Newton–Krylov Method (CNK)

The second approach that we use for the aerostructural solution is a fully coupled Newton–Krylov (CNK) method. Our implementation closely follows previous work by the authors on a lower fidelity case [36]. Algorithm 2 lists the pseudocode for evaluating the coupled nonlinear residual, \mathcal{R} .

Algorithm 1 Nonlinear block Gauss–Seidel method

```
1: Given:  $w^{(0)}, u^{(0)}, X_J, k_{\max}$ 
2: Initialize:  $\theta \in (0, 1]$ 
3: for  $k \leftarrow 1, k_{\max}$  do
4:    $X_S^{(k)} \leftarrow Tu^{(k-1)} + X_J$  ▷ Transfer displacements from CSM to CFD
5:    $X_V^{(k)} \leftarrow \mathcal{W}(X_S^{(k)})$  ▷ Deform volume mesh to match surface
6:    $\mathcal{A}^{(k)} \leftarrow \mathcal{A}(w^{(k-1)}, X_V^{(k)})$  ▷ Evaluate initial CFD residual
7:   while  $\|\mathcal{A}(w_i; x)\| > \epsilon_{\mathcal{A}} \|\mathcal{A}^{(k)}\|$  do ▷ CFD partial convergence criterion
8:      $w_{i+1}^{(k-1)} \leftarrow w_i^{(k-1)} + \Delta w_i$  ▷ Iterate CFD
9:      $i \leftarrow i + 1$ 
10:  end while
11:   $w^{(k)} \leftarrow w_i^{(k-1)}$  ▷ New CFD solution
12:   $F_A^{(k)} \leftarrow F_A(w^{(k)}, X_S^{(k)})$  ▷ Evaluate aerodynamics forces
13:   $F^{(k)} \leftarrow T^T F_A^{(k)}$  ▷ Transfer forces from CFD to CSM
14:   $\mathcal{S}^{(k)} \leftarrow \mathcal{S}(u^{(k-1)}, F^{(k)})$  ▷ Evaluate initial CSM residual
15:  while  $\|\mathcal{S}(u^{(k)}; x)\| > \epsilon_{\mathcal{S}} \|\mathcal{S}^{(k)}\|$  do ▷ CSM partial convergence criterion
16:     $u_{i+1}^{(k-1)} \leftarrow u_i^{(k-1)} + \Delta u_i$  ▷ Iterate CSM
17:     $i \leftarrow i + 1$ 
18:  end while
19:   $u^{(k)} \leftarrow u_i^{(k-1)}$  ▷ Temporary new CSM solution
20:   $\Delta u^{(k)} \leftarrow u^{(k)} - u^{(k-1)}$  ▷ Compute displacement increment
21:  if  $k > 1$  then
22:     $\theta \leftarrow \theta \left( 1 - \frac{(\Delta u^{(k)} - \Delta u^{(k-1)}) \cdot \Delta u^{(k)}}{\|\Delta u^{(k)} - \Delta u^{(k-1)}\|^2} \right)$  ▷ Adapt under-relaxation with Aitken acceleration
23:  end if
24:   $u^{(k+1)} \leftarrow u^{(k)} + \theta \Delta u^{(k)}$  ▷ Compute new structural states
25:  if  $\|\mathcal{A}^{(k)}\| < \epsilon_{AS} \|\mathcal{A}^{(1)}\|$  and  $\|\mathcal{S}^{(k)}\| < \epsilon_{AS} \|\mathcal{S}^{(1)}\|$  then ▷ Aerostructural convergence criterion
26:    break
27:  end if
28: end for
```

Algorithm 2 Coupled nonlinear residual computation

```
1: function  $\mathcal{R}(w, u)$ 
2:    $X_S \leftarrow Tu + X_J$  ▷ Transfer displacements
3:    $X_V \leftarrow \mathcal{W}(X_S)$  ▷ Deform volume mesh to match surface
4:    $\mathcal{A} \leftarrow \mathcal{A}(w, X_V)$  ▷ Evaluate CFD residuals
5:    $F_A \leftarrow F_A(w, X_S)$  ▷ Evaluate aerodynamics forces
6:    $F \leftarrow T^T F_A$  ▷ Transfer forces
7:    $\mathcal{S} \leftarrow \mathcal{S}(u, F)$  ▷ Evaluate CSM residuals
8:    $\mathcal{R} \leftarrow (\mathcal{A}, \mathcal{S})$  ▷ Combine residuals
9: return  $\mathcal{R}$ 
10: end function
```

The procedure is similar to one iteration of the NLBGS method, except that instead of computing an approximate solution update to the state variables, we evaluate only residuals. The communication and mesh-deformation costs of this coupled nonlinear residual evaluation are comparable to the cost of a

single NLBGS iteration. Provided the communication costs are low and the mesh-deformation algorithm is efficient, the coupled residual evaluation is inexpensive. We use an inexact Newton–Krylov approach to solve the coupled equations (9) by computing the approximate Newton update,

$$\begin{bmatrix} \frac{\partial \mathcal{A}}{\partial w} & \frac{\partial \mathcal{A}}{\partial u} \\ \frac{\partial \mathcal{S}}{\partial w} & \frac{\partial \mathcal{S}}{\partial u} \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta u \end{bmatrix} = - \begin{bmatrix} \mathcal{A}(w) \\ \mathcal{S}(u) \end{bmatrix}. \quad (10)$$

After this update is computed, the state variables w and u are updated with Δw and Δu , respectively.

To reduce the memory requirement, we use a matrix-free method. The matrix-vector products required by the Krylov method are approximated with finite differences. In theory, with this approach, we need only residual evaluations to solve the coupled problem. However, Krylov methods require effective preconditioning for acceptable performance, especially for very large systems with millions of degrees of freedom. We have implemented a block-Jacobi preconditioner that reuses the preconditioner of the original discipline solvers. This has two advantages: 1) the preconditioners for each discipline can be applied in parallel, because this approach eliminates contributions from the off-diagonal terms, which are never explicitly stored, and 2) since both of our discipline solvers use implicit methods, this preconditioner simply reuses the same linear solution method from the original solvers. Since the coupled preconditioning method varies from iteration to iteration, we must use a flexible variant of a Krylov method to solve the coupled Newton update (10). We choose to use FGMRES [40], which has been shown to work well for a wide range of large asymmetric systems. The Eisenstat–Walker method is used to adaptively select the forcing tolerance of the linear system solution to prevent oversolving during the initial iterations [43].

As with most Newton-type methods, this approach may not converge for certain initial conditions. We have not yet investigated the start-up behavior for the coupled Newton–Krylov method in detail. Our approach is to run a few relaxed NLBGS iterations (Algorithm 1) before switching to the Newton–Krylov algorithm. While this worked for the results presented in this paper, it is possible that other methods may be required for other cases, especially for even more flexible designs. The main advantage of the Newton–Krylov method is the elimination of the under-relaxation parameter, which leads to faster convergence, especially for problems with more flexible structures.

3 Coupled Adjoint Derivative Computation

1 Coupled Adjoint Overview

The critical component of the aerostructural framework is an efficient method for computing derivatives. Using the approach outlined by Martins *et al.* [26], we write the adjoint equations for the coupled aerostructural system. A more general derivation of this theory and its connection to other derivative computation methods can be found in Martins and Hwang [44]. The residuals, state variables, and adjoint variables for the aerodynamic discipline are \mathcal{A} , w , and ψ , respectively, and the corresponding variables for the structural discipline are \mathcal{S} , u , and ϕ . The total derivative of the function of interest, I , is

$$\frac{dI}{dx} = \frac{\partial I}{\partial x} + \begin{bmatrix} \frac{\partial I}{\partial w} & \frac{\partial I}{\partial u} \end{bmatrix} \begin{bmatrix} \frac{dw}{dx} \\ \frac{du}{dx} \end{bmatrix}. \quad (11)$$

We make a distinction between the partial derivatives, denoted “ ∂ ,” and the total derivatives, denoted “ d .” The total derivatives indicate that the derivative requires the solution of the system governing of equations. We write the total derivative of the residuals as

$$\begin{bmatrix} \frac{d\mathcal{A}}{dx} \\ \frac{d\mathcal{S}}{dx} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathcal{A}}{\partial x} \\ \frac{\partial \mathcal{S}}{\partial x} \end{bmatrix} + \begin{bmatrix} \frac{\partial \mathcal{A}}{\partial w} & \frac{\partial \mathcal{A}}{\partial u} \\ \frac{\partial \mathcal{S}}{\partial w} & \frac{\partial \mathcal{S}}{\partial u} \end{bmatrix} \begin{bmatrix} \frac{dw}{dx} \\ \frac{du}{dx} \end{bmatrix} = 0. \quad (12)$$

Substituting the solution of Eq. (12) into Eq. (11) to eliminate the total derivatives we obtain

$$\frac{dI}{dx} = \frac{\partial I}{\partial x} - \underbrace{\begin{bmatrix} \frac{\partial I}{\partial w} & \frac{\partial I}{\partial u} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathcal{A}}{\partial \mathcal{S}} & \frac{\partial \mathcal{A}}{\partial \mathcal{S}} \\ \frac{\partial \mathcal{A}}{\partial w} & \frac{\partial \mathcal{A}}{\partial u} \end{bmatrix}^{-1} \begin{bmatrix} \frac{\partial \mathcal{A}}{\partial \mathcal{S}} \\ \frac{\partial \mathcal{A}}{\partial \mathcal{S}} \end{bmatrix}}_{\Psi^T}. \quad (13)$$

There are two techniques for solving this equation. One option is to solve Eq. (12) for $[dw/dx^T \ du/dx^T]^T$ for a given design variable and to use Eq. (11) to compute the total derivative. This is known as the direct method. If, however, there are more design variables than functions of interest—which is usually the case for high-fidelity aerostructural optimization—it is computationally more efficient to use the adjoint method and to solve for the coupled adjoint vector, $\Psi = [\psi^T \ \phi^T]^T$. Rearranging Eq. (13) yields

$$\begin{bmatrix} \frac{\partial \mathcal{A}}{\partial w} & \frac{\partial \mathcal{A}}{\partial u} \end{bmatrix}^T \begin{bmatrix} \psi \\ \phi \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial w} & \frac{\partial I}{\partial u} \end{bmatrix}^T. \quad (14)$$

These are the coupled adjoint equations, which are solved once for each function of interest. After the solution for the coupled adjoint equations (14) is obtained, the following equation can be used to compute the total derivative:

$$\frac{dI}{dx} = \frac{\partial I}{\partial x} - \psi^T \left(\frac{\partial \mathcal{A}}{\partial x} \right) - \phi^T \left(\frac{\partial \mathcal{S}}{\partial x} \right). \quad (15)$$

The details of the computation of the required partial-derivative terms are explained in the following sections.

2 Coupled Adjoint Implementation

Implementing the coupled adjoint, including all the required partial-derivative terms, is a challenging endeavor. Furthermore, ensuring that the partial-derivative computations and solution methods are efficient and exhibit good parallel scalability is even more difficult. In this section, we present some important aspects involved in the solution of Eq. (14) and in the computation of the required partial-derivative terms. The choice between the various approaches is justified in terms of trade-offs between cost of implementation, computational effort, and memory requirements.

2.1 Aerodynamic Residual Partial Derivatives

The aerodynamic diagonal block, $\partial \mathcal{A} / \partial w$, in the coupled adjoint equations (14), represents the derivative of the aerodynamic residuals with respect to the aerodynamic states and can be used directly from the aerodynamic adjoint formulation, which was implemented by Mader *et al.* [45]. This term is computed exactly using the reverse-mode AD of Tapenade [46] and then stored.

The first off-diagonal block, $\partial \mathcal{A} / \partial u$, contains the derivatives of the aerodynamic residual with respect to the structural displacements. This is a challenging matrix to compute in a two-state aerostructural formulation. What makes this term particularly challenging is that a single structural degree of freedom (DOF) can affect many aerodynamic cells. Using the hybrid-mesh deformation scheme described in Section 2.4, any structural DOF that perturbs a supernode results in a dense column due to the solution of the linear system of equations. With a large number of structural surface DOFs and a large number of supernodes, storing this matrix would require an excessive amount of memory. Instead, we compute this term in a matrix-free fashion using the chain rule:

$$\left(\frac{\partial \mathcal{A}}{\partial u} \right)^T \psi = \left(\frac{\partial X_S}{\partial u} \right)^T \left(\frac{\partial X_V}{\partial X_S} \right)^T \left(\frac{\partial \mathcal{A}}{\partial X_V} \right)^T \psi \quad (16)$$

where $\partial \mathcal{A} / \partial X_V$ contains the derivatives of the aerodynamic residual with respect to variations of all the volume mesh coordinates. The sparse structure of this matrix is due to the stencil used for the finite-volume computation. We compute this matrix using reverse-mode AD, in a similar fashion to $\partial \mathcal{A} / \partial w$. This

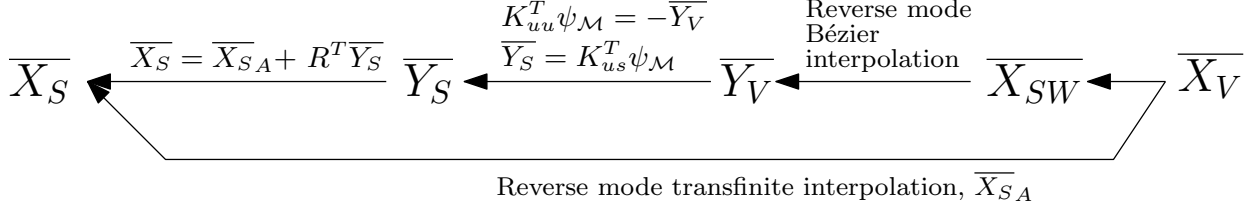


Figure 3: Adjoint procedure for solid mesh warping

matrix is computed in parallel and stored in distributed memory, which allows us to compute the transpose matrix-vector products in a relatively inexpensive manner.

The matrix of derivatives of the mesh deformation, $\partial X_V / \partial X_S$, requires a careful implementation to be computationally efficient. For the transpose matrix-vector products required for the adjoint computation, the normal deformation procedure—previously shown in Fig. 2—proceeds in reverse. Reverse-mode AD is used to compute the variation of all the surface nodes (\overline{X}_S) and the variation of the volume supernodes (\overline{Y}_V). The variation of the volume supernodes is related to the surface supernodes through the solution of the mesh adjoint equation given by

$$K_{uu}^T \psi_M = -\overline{Y}_V, \quad (17)$$

where ψ_M is the mesh adjoint. Finally, the variation \overline{Y}_S is propagated to the full mesh surface using the transpose of the restriction operator. This procedure is illustrated in Fig. 3.

This computation is performed just once for each transpose matrix-vector product. To ensure overall scalability of the coupled adjoint solution, all of the computations are executed in parallel with acceptable efficiency. In this case, since the linear system of equations for the mesh deformation is symmetric and fixed, a parallel LU decomposition is performed once using the distributed version of SuperLU [47]. Subsequent mesh perturbations and adjoint solutions can be computed quickly with a back-substitution operation. Finally, since X_S is given by $X_S = X_J + Tu$, the partial-derivative matrix $(\partial X_S / \partial u)^T$ is simply T^T , which is the load transfer operation.

This way of computing $\partial \mathcal{A} / \partial u$ contrasts with the finite differences used by Martins *et al.* [27], where each structural surface DOF was perturbed in turn, followed by a mesh deformation to re-evaluate the aerodynamic residuals. The computational cost of these finite differences scaled directly with the number of structural surface DOFs, and it became clear that this approach would not be suitable for the $\mathcal{O}(10^4)$ surface DOFs used in the present work. As we will see in Section 4, this method allows us to quickly compute the aerostructural gradient with structural meshes with over 1 million DOFs and problems with thousands of design variables.

The final aerodynamic residual partial derivative, $\partial \mathcal{A} / \partial x$, is computed using another chain rule:

$$\psi^T \left(\frac{\partial \mathcal{A}}{\partial x} \right) = \psi^T \left(\frac{\partial \mathcal{A}}{\partial X_V} \right) \left(\frac{\partial X_V}{\partial X_S} \right) \left(\frac{\partial X_S}{\partial x} \right). \quad (18)$$

The two right-most terms are identical to the terms in the computation of $\partial \mathcal{A} / \partial u$ (16), and the computation of the remaining term, $\partial X_S / \partial x$, is described in Section 2.4.

2.2 Structural Residual Partial Derivatives

The other off-diagonal block in the coupled adjoint equations (14), $\partial \mathcal{S} / \partial w$, represents the derivatives of the structural residuals with respect to the aerodynamic states. The only contribution to this matrix is due to

the aerodynamic forces, F_A , and is given by,

$$\begin{aligned}
\left(\frac{\partial \mathcal{S}}{\partial w}\right)^T \phi &= -\left(\frac{\partial F}{\partial w}\right)^T \phi \\
&= -\left(\frac{\partial F_A}{\partial w}\right)^T \left(\frac{\partial F}{\partial F_A}\right)^T \phi \\
&= -\left(\frac{\partial F_A}{\partial w}\right)^T T \phi.
\end{aligned} \tag{19}$$

We compute and store $\partial F_A/\partial w$ and $\partial F_A/\partial X_S$, which we refer to as the *coupling matrices*. These terms are computed using reverse-mode AD applied to the CFD force evaluation routine. Unlike $\partial \mathcal{A}/\partial w$ or $\partial \mathcal{A}/\partial X_V$, only cells on the wetted surface have nonzero components. Thus, the cost of storing these terms is very low.

The remaining diagonal block, $\partial \mathcal{S}/\partial u$, represents the derivative of the structural residuals with respect to the structural states. For linear aerostructural analysis, this term is *not* simply the linear stiffness matrix, K . Given the structural residuals (2), we find that the applied forces, F , are actually explicit functions of u . As the flexible structure deforms, the surface normals on the CFD mesh change orientation, giving rise to a nonlinear following force. To obtain derivatives, this effect must be included. To obtain the matrix-vector products required by the Krylov-type solution strategy, we compute this term in the following manner:

$$\begin{aligned}
\left(\frac{\partial \mathcal{S}}{\partial u}\right)^T \phi &= \left(K - \left(\frac{\partial F}{\partial u}\right)^T\right) \phi \\
&= \left(K - \left(\frac{\partial F}{\partial F_A}\right) \left(\frac{\partial F_A}{\partial X_S}\right)^T \left(\frac{\partial X_S}{\partial u}\right)\right) \phi \\
&= \left(K - T^T \left(\frac{\partial F_A}{\partial X_S}\right)^T T\right) \phi.
\end{aligned} \tag{20}$$

This product is formed by transferring the ϕ vector to the aerodynamics using a displacement transfer, finding the $\partial F_A/\partial X_S$ transpose matrix-vector product, and then transferring the result back to the structural solver using a load transfer. Since this operation requires two synchronous data transfers, these products are evaluated only as required, and the exact $\partial \mathcal{S}/\partial u$ is never formed.

The partial derivatives of the structural residuals with respect to the design variables, $\partial \mathcal{S}/\partial x$, required in total-derivative equation (13), are evaluated analytically.

The product of the partial derivatives and the adjoint vector $\psi^T \partial \mathcal{S}/\partial x$ is evaluated analytically on an element-by-element basis, where geometric and material variables are handled using different approaches. For the geometric variables, the computation proceeds in two stages. First, the product $\psi^T \partial \mathcal{S}/\partial X_J$ is computed, where X_J are all the node locations. Next, the product of $\psi^T \partial \mathcal{S}/\partial X_J$ and the derivative of the node locations with respect to the design variables, $\partial X_J/\partial x$, is computed. This computation is arranged so that it is efficient when the number of geometric design variables is large. For the derivative of the material design variables, the sparse dependency of the design variables is exploited, and only elements with a nonzero contribution to $\psi^T \partial \mathcal{S}/\partial x$ are computed.

2.3 Partial Derivatives of the Function of Interest

The right-hand side of the coupled adjoint system (14) is the derivative of the function of interest with respect to the system states. We consider only aerodynamic or structural functions, that is, functions that could be used in a single-discipline analysis. The derivatives of typical aerodynamic functions such as C_L , C_D , and C_M are computed with reverse-mode AD, in a similar fashion to the coupling matrices. We analytically evaluate the derivatives of typical structural functions, such as individual element stresses, σ_i , or the Kreisselmeier–Steinhauser (KS) [48, 49] aggregation functions. For the structural functions, $\partial I/\partial w$ is zero. However, for

aerodynamic functions involving surface pressure or traction integration, $\partial I/\partial u$ is nonzero and is given by

$$\begin{aligned}\frac{\partial I}{\partial u} &= \left(\frac{\partial I}{\partial X_S} \right) \left(\frac{\partial X_S}{\partial u} \right) \\ &= \left(\frac{\partial I}{\partial X_S} \right) T.\end{aligned}\tag{21}$$

The resulting procedure is similar to that used in the computation of $\partial \mathcal{S}/\partial u$, expressed in Eq. (20).

2.4 Partial Derivatives in the Total-Derivative Equation

The evaluation of the partial-derivative terms in the total-derivative equation (15) is challenging because the load and displacement transfer operations have a dependency on the geometric design variables. Figure 1 shows how the displaced surface coordinates, X_S , are extrapolated from the structural domain. The complication arises from the fact that the length of the extrapolation vector, r , changes with the design variables. Accounting for this change leads to a transfer scheme that remains accurate for large geometry changes, but it complicates the computation of the derivatives.

For aerodynamic functions, we compute $\partial I/\partial X_S$ using reverse-mode AD, similarly to what was done for the coupling matrices in Eq. (19). Then, for each design variable, we evaluate the derivative of the perturbed shape, X_S , with respect to the design variables, x , by computing

$$\begin{aligned}\frac{\partial I}{\partial x} &= \frac{\partial I}{\partial X_S} \left(\frac{\partial X_S}{\partial x} \right) \\ &= \frac{\partial I}{\partial X_S} \left(\frac{\partial X_J}{\partial x} + \frac{\partial (u_r \times r)}{\partial x} \right).\end{aligned}\tag{22}$$

Since the transfer-operation matrix T is never explicitly formed, we evaluate a single entry of the resulting vector sequentially in a matrix-free fashion. A similar correction must be made for the derivative of the structural states with respect to the geometric design variables, since the load transfer depends on the design variables.

Finally, the evaluation of the $\partial X_S/\partial x$ term is used to complete the computation of $\partial \mathcal{A}/\partial x$. These operations are relatively costly, since in our implementation they require a synchronous transfer of data from both disciplines for each design variable. As we will see in Section 4.4, the cost of this computation scales very weakly with the number of design variables, and thus makes a small contribution to the overall computational cost of computing the gradient.

3 Coupled Adjoint Solution

As with the aerostructural system of equations, we consider two approaches to solve the coupled adjoint system (14): a segregated approach and a monolithic approach. The segregated approach is the lagged coupled adjoint approach [27], which corresponds to a linear block Gauss–Seidel algorithm. We also investigate a monolithic approach, which uses a Krylov method applied directly to the coupled linear system.

3.1 Linear Block Gauss–Seidel Method (LBGS)

The analog of the NLBGS method for coupled adjoint equations is the linear block Gauss–Seidel (LBGS) method. This method expresses the interdisciplinary coupling as additional forcing terms to the right-hand side of each set of disciplinary adjoint equations. If we separate the single matrix equation into the two blocks we obtain:

$$\begin{aligned}\left(\frac{\partial \mathcal{A}}{\partial w} \right)^T \psi^{(k)} &= \left(\frac{\partial I}{\partial w} \right)^T - \left(\frac{\partial \mathcal{S}}{\partial w} \right)^T \phi^{(k-1)} \\ \left(\frac{\partial \mathcal{S}}{\partial u} \right)_K \phi^{(k)} &= \left(\frac{\partial I}{\partial u} \right)^T - \left(\frac{\partial \mathcal{A}}{\partial u} \right)^T \psi^{(k)} - \left(\frac{\partial \mathcal{S}}{\partial u} \right)_F^T \phi^{(k-1)}.\end{aligned}\tag{23}$$

The subscripts K and F in the term $\partial \mathcal{S}/\partial u$ represent the contributions from the stiffness matrix and external forces, respectively. Since the external-force component is costly to compute and involves synchronous

communication between disciplines, it is lagged. The same applies to the contribution from the off-diagonal term. We can iterate between these two equations until we reach the desired convergence level. The main advantage of this approach is that the discipline adjoint solvers can be re-utilized by simply adding the appropriate right-hand side forcing terms. In practice, we only partially converge each discipline before a data exchange is made. Additionally, depending on the problem, it may be necessary to apply an under-relaxation factor to the structural update for enhanced stability. The pseudocode for this approach is listed in Algorithm 3.

Algorithm 3 Block Gauss–Seidel coupled adjoint solution method

```

1: Given:  $\psi^{(0)}, \phi^{(0)}, k_{\max}, \theta \in (0, 1]$  ▷  $\psi^{(0)}, \phi^{(0)}$  can be from a previous solution
2: if Aerodynamic Function then
3:    $\frac{\partial I}{\partial u} \leftarrow T^T \left( \frac{\partial I}{\partial X_S} \right)$  ▷ Cross-discipline function partial derivative
4: end if
5: for  $k \leftarrow 1, k_{\max}$  do
6:    $\phi_A \leftarrow T\phi^{(k-1)}$  ▷ Transfer structural adjoint
7:    $R_A^{(k)} \leftarrow \left( \frac{\partial I}{\partial w} \right)^T - \left( \frac{\partial A}{\partial w} \right)^T \psi^{(k-1)} - \left( \frac{\partial F_A}{\partial w} \right)^T \phi_A$  ▷ Aerodynamic adjoint RHS
8:    $\left( \frac{\partial A}{\partial w} \right)^T \Delta\psi = R_A^{(k)}$  ▷ Partially solve CFD adjoint update
9:    $\psi^{(k)} \leftarrow \psi^{(k-1)} + \Delta\psi$  ▷ Update aerodynamic adjoint
10:   $P \leftarrow T^T \left( \frac{\partial X_V}{\partial X_S} \right)^T \left( \frac{\partial A}{\partial X_V} \right)^T \psi^{(k)}$  ▷ Transfer aerodynamic contribution to structural RHS
11:   $Q \leftarrow T^T \left( \frac{\partial F_A}{\partial X_S} \right)^T T\phi^{(k-1)}$  ▷ Force  $\frac{\partial S}{\partial u}$  contribution
12:   $R_S^{(k)} \leftarrow \left( \frac{\partial I}{\partial u} \right)^T - K\phi^{(k-1)} - P - Q$  ▷ Compute structural RHS
13:   $K\Delta\phi = R_S^{(k)}$  ▷ Partially solve CSM adjoint update
14:   $\phi^{(k)} \leftarrow \phi^{(k-1)} + \theta\Delta\phi$  ▷ Under-relaxed structural adjoint update
15:  if  $\|R_A^{(k)}\| < \epsilon_{SA} \|R_A^{(1)}\|$  and  $\|R_S^{(k)}\| < \epsilon_{SA} \|R_S^{(1)}\|$  then ▷ Coupled adjoint convergence check
16:    break
17:  end if
18: end for

```

3.2 Coupled Krylov Method (CK)

The second approach that we use to solve the coupled adjoint equation (14) is a fully coupled monolithic method. As with the coupled Newton–Krylov method, this method follows previous work with lower fidelity systems [36]. This is the first use of a monolithic solution method for the coupled adjoint equations on a large-scale aerostructural problem.

Krylov subspace methods are particularly attractive for this type of problem, since they require only matrix-vector products, allowing the use of our pre-existing matrix-free adjoint operators. In our case, the diagonal blocks of the Jacobian in Eq. (14) are stored, but the off-diagonal terms are not explicitly stored. Effective preconditioning is critical to the performance of Krylov methods, especially on large systems of equations such as the ones we are solving.

We choose to use a block-Jacobi preconditioner for the coupled system, since this approach ignores the off-diagonal terms and allows the aerodynamic and structural preconditioning to be carried out in parallel. For the aerodynamic block preconditioner we reuse the preconditioned Krylov subspace method used for the aerodynamic adjoint. In this case however, we run a fixed number of GMRES iterations, typically between 10 and 20.

For the structural block of the preconditioner, we use the matrix factorization of K . The pseudocode for the linear adjoint operator for the coupled system is listed in Algorithm 4. The matrix-vector products are computed in a matrix-free fashion.

Algorithm 4 Coupled Krylov method linear operator

```
1: function MULT( $X$ )                                ▷ Compute Jacobian-vector product with  $X$ 
2:    $(X_A, X_S) \leftarrow X$                           ▷ Extract aerodynamic and structural components
3:    $Y_A \leftarrow \left(\frac{\partial \mathcal{A}}{\partial w}\right)^T X_A$ 
    $Y_S \leftarrow K X_S$                                 ▷ Evaluate diagonal contributions in parallel
4:    $Y_A \leftarrow Y_A + \left(\frac{\partial F_A}{\partial w}\right)^T T X_S$         ▷ Add aerodynamic off-diagonal term
5:    $Y_S \leftarrow Y_S + T^T \left(\frac{\partial X_V}{\partial X_S}\right)^T \left(\frac{\partial \mathcal{A}}{\partial X_V}\right)^T X_A$     ▷ Add structural off-diagonal term
6:    $Y_S \leftarrow Y_S + T^T \left(\frac{\partial F_A}{\partial X_S}\right)^T T X_S$         ▷ Add force  $\frac{\partial \mathcal{S}}{\partial u}$  contribution
7:    $Y \leftarrow (Y_A, Y_S)$                             ▷ Combine aerodynamic and structural components
8: return  $Y$ 
9: end function
```

4 Memory Requirements

We now consider the additional memory requirements for the fully coupled nonlinear and adjoint solution methods. One advantage of the Gauss–Seidel strategy is that the coupled solution does not require any significant additional memory. This allows the aerostructural analysis and optimization to inherit the same memory footprint as the stand-alone discipline solvers.

If a Krylov method with a long recurrence relationship is used to solve Eq. (10) or Eq. (14), significant additional memory is required. In the current framework, we use restarted FGMRES(m) (where m is the size of the Krylov subspace) to solve Eqs. (10) and (14). This requires $2m$ vectors of size \mathcal{R} to be stored in memory. For matrix-free disciplinary methods, these vectors require additional memory.

If Newton–Krylov methods are used to solve the disciplinary systems, however, it may be possible to reduce the memory requirement of the individual solvers when a coupled method is used. When our CFD solver is used stand-alone, a preconditioned GMRES(s) algorithm is used to approximately solve the Newton update equation and tightly solve the adjoint equation. The system is preconditioned using restrictive additive Schwartz and ILU(p) on each of the subdomains, and it requires s subspace vectors. For the coupled solution, it is possible to “exchange” some of the Krylov vectors used for discipline preconditioning to the outer Newton–Krylov or Krylov solver.

A fixed number of subspace vectors, n , can be used for either the outer FGMRES solver or the discipline preconditioner. If we choose $m = \lfloor (n - s)/2 \rfloor$ the coupled methods and the Gauss–Seidel methods will use the same memory. In effect, we are transferring computational work (and memory) from the discipline solvers to the coupled solver. This results in more outer iterations with fewer expensive inner iterations. The increase in the number of outer nonlinear or linear iterations also increases the computational cost of evaluating the off-diagonal terms. Eventually, this increasing cost leads to diminishing returns and there is a range of s values that leads to the best performance. This trade-off is investigated in Section 4.3.

4 Results

1 Common Research Model Test Case

The common research model (CRM) [50] wing-body-tail configuration is used for the studies presented in this section. This configuration exhibits design features typical of a modern wide-body long-range aircraft. The full configuration is used for the CFD analysis but only the wing structure is modeled. For the structural analysis, we created a structural model conforming to the OML that is representative of a modern aircraft wingbox. We use linear, second-order MITC shell elements [51] to model the ribs, spars, skins, stringers, and rib stiffeners. The upper wing skin and stringers are made of Aluminum 7050. The remainder of the primary wing structure is assumed to be manufactured from Aluminum 2024. The general arrangement of the CRM geometry, along with a detailed view of the wing structure, is shown in Fig. 4. Table 1 lists the key parameters for the CRM configuration.

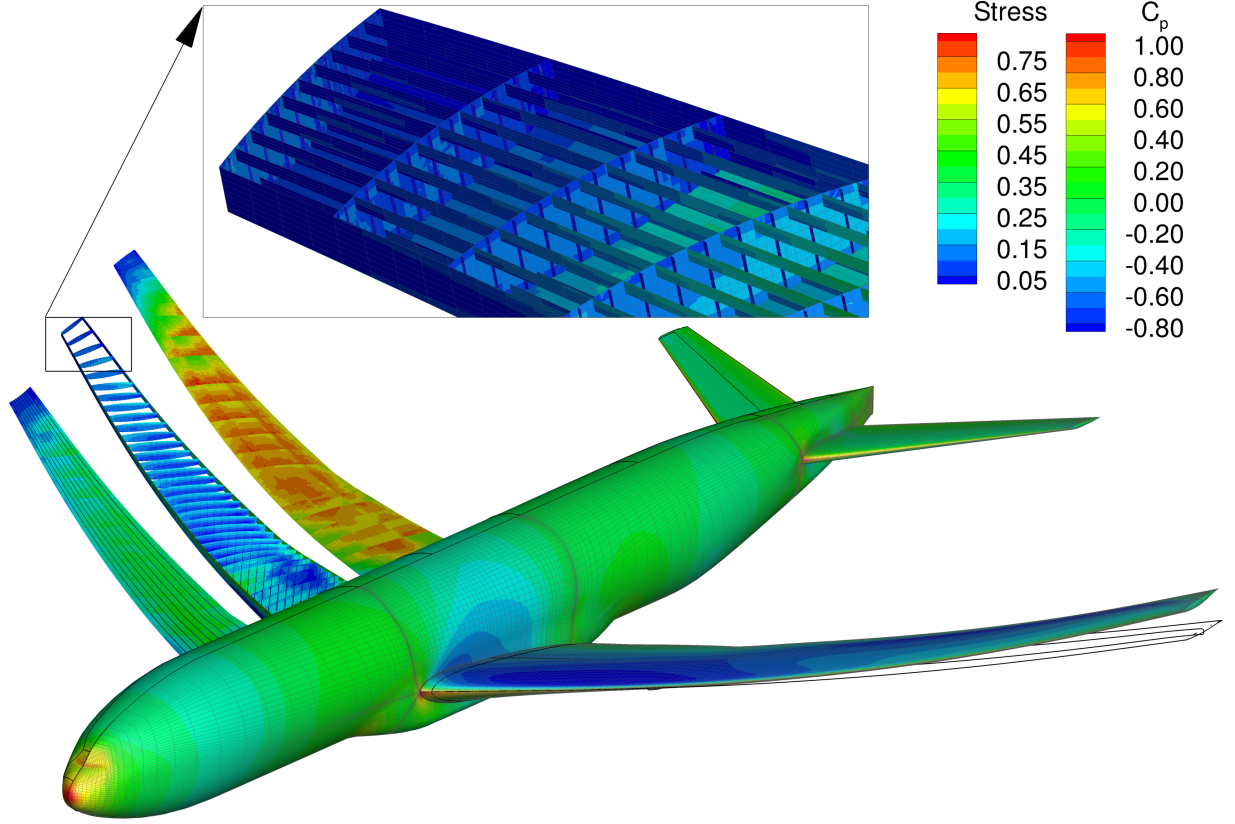


Figure 4: Common research model surface pressure coefficient at Mach=0.85 and 1 g cruise condition (right); displaced shape and stresses non-dimensionalized by the yield stress at the 2.5 g load condition and detailed view of the structure (left).

Table 1: Common research model specifications

Parameter	Value	Units
Cruise Mach number	0.85	-
Cruise lift coefficient	0.5	-
Span	58.6	m
Aspect ratio	9.0	-
Reference wing area	383.7	m ²
Sweep (leading edge)	37.4	°
Maximum take-off weight (MTOW)	298 000	kg
Operational empty weight	138 100	kg

2 Gradient Verification

Accurate derivatives are critical for the performance of any gradient-based optimization. As described in Section 3, all partial-derivative terms for the coupled adjoint equation (14) and total-derivative equation (15) are computed either with AD (using Tapenade [46]) or analytically. These terms can be evaluated with near machine precision, allowing for extremely accurate derivatives. To ensure an entirely consistent formulation and implementation, we verify the coupled adjoint using the complex step method [52, 53] as a benchmark

applied to the entire coupled system. In our experience, if only some of the digits in the derivatives are verified, certain bugs in the computations might go unnoticed and emerge more obviously in other cases. Therefore, we do not use finite-difference formulae to verify the results herein.

We modified and created “complexified” versions of all the analysis modules that allow for the complex perturbation of any design variable. When an aerostructural solution is computed with a complex perturbation on a design variable, the resulting complex parts in the functions of interest correspond to the derivatives of those functions. Unlike finite differences, the complex step method does not suffer from subtractive cancellation. With the elimination of subtractive-cancellation errors, we can choose an arbitrarily small step size and reduce the truncation error to machine precision. For iterative methods, the complex perturbations on the function of interest converge as the nonlinear system converges [53].

The verification is carried out with the level-1 discretization given in Table 5. The aerostructural analysis and adjoint systems are converged to a relative tolerance of $\epsilon_{AS} = \epsilon_{SA} = 10^{-8}$. It is generally not possible to obtain any further convergence for the aerostructural system, since the condition number of the structural Jacobian is typically $\mathcal{O}(10^9)$ for the shell structures used in this work. The resulting derivatives agree to $\mathcal{O}(10^{-5})$. Obtaining better agreement becomes difficult because small discrepancies between the real and complex codes often appear, probably due to differences in compiler implementations and optimizations of certain built-in functions that do not always produce bit-compatible results. However, the level of accuracy achieved with the current coupled adjoint implementation is more than sufficient for gradient-based optimization.

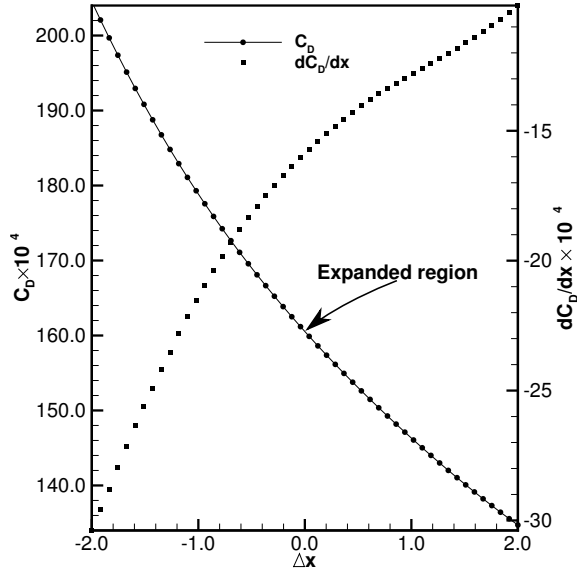
For the verification, we compute the derivatives of two aerodynamic functions, C_L and C_D , and a structural function, the KS stress aggregation function. The KS function aggregates the von Mises stress on the lower skin and lower stringers. The derivatives of these three functions are computed with respect to global and local design variables. Global design variables explicitly affect the residuals of all governing equations, while local variables directly affect only one discipline. The chosen global variables are wing span and sweep. The local variables are the angle of attack α (an aerodynamic variable) and the skin thickness of one lower surface skin panel at the root (a local structural variable). Table 2 shows the derivatives computed using the coupled adjoint method, the complex-step method, and the forward finite-difference formula. A complex-step size of $10^{-40}j$ is used for the complex-step method. For the finite-difference derivatives, we show the result for the optimal step size for each variable. Seven step sizes ranging from 10^{-1} to 10^{-7} were considered. We then use the derivative computed with the optimum step, h_{opt} , i.e., the step that minimizes the sum of the truncation error and the subtractive-cancellation error.

Table 2: Coupled adjoint derivative verification

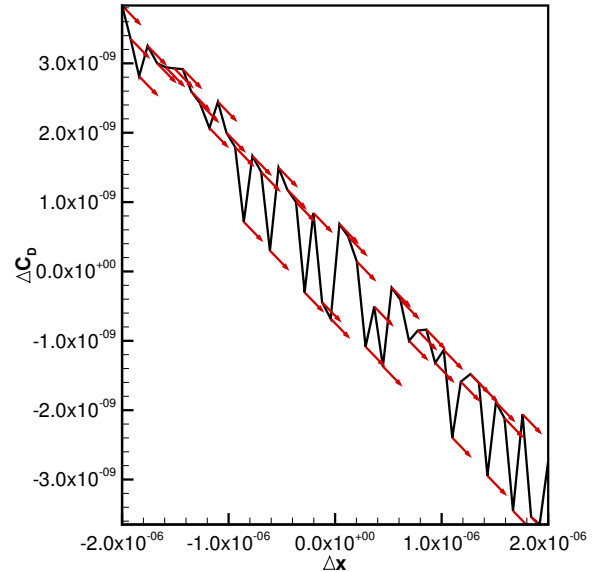
Function	Variable	Complex-step	Coupled adjoint		Finite-difference		
		Value	Value	Relative Error	Value	Relative Error	h_{opt}
C_L	Span	0.027265249	0.027266123	-3.21×10^{-5}	0.027858610	-2.18×10^{-2}	1×10^{-6}
	Sweep	-0.023601035	-0.023595480	2.35×10^{-4}	-0.023634861	-1.43×10^{-3}	1×10^{-4}
	α	0.143953101	0.143953092	5.96×10^{-8}	0.143632330	2.23×10^{-3}	1×10^{-6}
	t_{skin}	0.145264733	0.145266846	-1.45×10^{-5}	0.144828960	3.00×10^{-3}	1×10^{-5}
C_D	Span	0.000954129	0.000953880	2.61×10^{-4}	0.000911573	4.46×10^{-2}	1×10^{-5}
	Sweep	-0.001386782	-0.001388121	-9.65×10^{-4}	-0.001292728	6.78×10^{-2}	1×10^{-3}
	α	0.006336909	0.006336908	1.55×10^{-7}	0.006234932	1.61×10^{-2}	1×10^{-1}
	t_{skin}	0.007273797	0.007273570	3.12×10^{-5}	0.007141012	1.83×10^{-2}	1×10^{-6}
KS	Span	0.020412974	0.020414310	-6.55×10^{-5}	0.020939720	-2.58×10^{-2}	1×10^{-3}
	Sweep	-0.006399502	-0.006391973	1.18×10^{-3}	-0.005821129	9.04×10^{-2}	1×10^{-3}
	α	0.086843094	0.086843079	1.70×10^{-7}	0.092562735	-6.59×10^{-2}	1×10^{-3}
	t_{skin}	-0.660819304	-0.660817730	2.38×10^{-6}	-0.291121540	5.59×10^{-1}	1×10^{-5}

The derivatives computed with the coupled adjoint method differ from the complex-step method by relative errors of 10^{-3} to 10^{-6} . The derivatives with respect to the local aerodynamic variable, α , are typically more accurate due primarily to a more straightforward total-derivative computation. The finite-difference derivatives are much less reliable and typically differ from the complex-step derivatives by 10^{-1} to 10^{-3} . The limited accuracy of the finite-difference gradients reduces their effectiveness in verifying a coupled adjoint implementation, since errors of order $\mathcal{O}(10^{-2})$ and below cannot be detected.

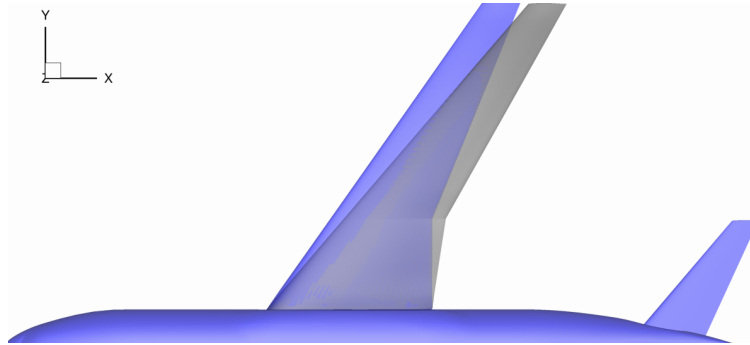
We also performed a study to demonstrate that the design space resulting from our aerostructural analysis techniques is relatively smooth and continuous with respect to the geometric variables. Geometric variables directly affect both disciplines through mesh changes, and therefore the derivatives with respect to these variables are more challenging to compute. A shearing sweep design variable that displaces all coordinates in the x -direction along the span of the wing is used in this study. Figure 5c shows the extent of the sweep change about the reference value, from $\Delta x = -2$ to $\Delta x = 2$. Figure 5a plots the derivative of the drag coefficient over this sweep range. We can see that both the drag coefficient and its derivative with respect to sweep vary smoothly at this scale. Figure 5b shows a detailed view where the variation in sweep is six orders of magnitude smaller than the one shown in Fig. 5a. Here, we can now see the noise inherent in the aerostructural solution, which is only $\mathcal{O}(10^{-7})C_D$. The reasons for such a low noise is the level of convergence are that we are able to achieve in the aerostructural solution, and the fact that the mesh movement is relatively smooth. Figure 5b also shows vectors representing the derivative computed with the coupled adjoint method. In spite of the noise in the drag coefficient, the derivative still indicate the correct trend of the noisy function.



(a) Gradient of C_D with respect to shearing sweep for a range of sweep values



(b) Noise level in C_D solution. Vectors show the coupled adjoint computed derivative.



(c) Visualization of the change in sweep

Figure 5: Verification of smoothness of coupled adjoint derivatives

To employ gradient-based optimization effectively, we require smooth functions with continuous first derivatives. We have taken great care to ensure that all computational components of the aerostructural analysis, including the geometry manipulation, the mesh deformation, and the two discipline solvers, have

smooth responses; we pass full-precision data between components and strive for the best possible numerical precision. It is clear from Fig. 5a that the derivatives are not only smooth but also continuous. The smooth design space and the accurate coupled adjoint derivatives demonstrate that the current framework is well suited for large-scale gradient-based optimization. This has also been demonstrated by successfully performing the aerostructural optimization of an aircraft configuration with respect to hundreds of design variables [33].

3 Solution Method Comparison

In this section, we compare the two strategies for the solution of the aerostructural system and the two strategies for the solution of the adjoint system. The comparison is carried out using the level-2 discretization listed in Table 5. We consider two load cases: a 1 g cruise condition with only moderate elastic deformation and a 2.5 g maneuver condition with significantly more deflection. We consider several variations of the NLBGS and CNK methods. First we consider four fixed values of the under-relaxation parameter: $\theta = 0.25, 0.5, 0.75$, and 1.0 (the latter corresponds to no under-relaxation). For these analyses, $\theta^{(k)} = \theta^{(0)}$ for all iterations. An additional analysis with Aitken acceleration ($\theta^{(0)} = 0.5$) is also performed. For all the NLBGS analyses, $\epsilon_A = 0.1$ and $\epsilon_S = 1 \times 10^{-3}$. However, since we are using a direct factorization method for the structural analysis, a lower error than ϵ_S is usually achieved.

For the NLBGS analysis, the aerodynamic Newton–Krylov solver uses GMRES(75). This system is preconditioned with additive Schwartz [54] (overlap 1) and ILU(2) on each of the subdomains. For the coupled Newton–Krylov algorithm, three non-Aitken accelerated relaxed Gauss–Seidel start-up iterations are performed for the 1 g case, and five start-up iterations are performed for the 2.5 g case. Three variants of the algorithm are evaluated, each with a different subspace size for the aerodynamic part of the coupled block-Jacobi preconditioner. To ensure a fair comparison using constant memory, the size of the FGMRES outer subspace is $m = \lfloor (75 - s)/2 \rfloor$, which yields similar memory usage for the CFD processors. While the structural processors have additional memory requirements, these requirements are small compared to the memory required for the stiffness matrix and associated factorization, so they are acceptable.

Table 3 lists the computational times for each of the test combinations. The N_{iter} column is the number of iterations required for convergence, which has a slightly different meaning depending on the solution method. For the NLBGS methods, this is the number of iterations required for convergence. For the CNK methods, this is the total number of nonlinear residual evaluations. Since the cost of the preconditioned nonlinear residual evaluations varies with the strength of the preconditioner, i.e., the value of s , the number of iterations is not directly comparable between the two classes of methods. The “—” symbol signifies that a given parameter is not relevant for the particular case.

A number of trends can be determined from these results. For both load cases, fixed under-relaxation parameters with values of 0.25 or 1.0 consistently performed poorly, while values of 0.5 and 0.75 performed reasonably well. The lowest under-relaxation of 0.25 is very robust, but it takes a large number of iterations to converge. Conversely, the solution without under-relaxation ($\theta = 1.0$) consistently over- and under-shoots the actual structural displacement, especially for the 2.5 g condition, leading to a large number of iterations. The Aitken acceleration reduces the required number of Gauss–Seidel iterations considerably, resulting in the second-best Gauss–Seidel solution time for each of the two load cases. The advantage of the Aitken acceleration is that the optimal value of the θ parameter, which is problem dependent, does not need to be known beforehand. For example, the optimal θ for the 1 g condition is 0.75, while for the 2.5 g load case it is 0.5.

Given the ease of implementation and increase in performance, and assuming that we do not know the optimal under-relaxation parameter in advance, we recommend the Aitken-accelerated form of the Gauss–Seidel method among the segregated analysis methods.

The CNK method generally performs better than the Gauss–Seidel methods, and the lowest solution times for each of the two flight conditions are achieved by this method. The effect of the strength of the inner preconditioner on the number of inner iterations is evident in Table 3. As expected, $s = 5$ requires the most outer iterations, with each iteration having the lowest computational cost. For the largest number of outer iterations ($m = 35$), the number of iterations increases considerably, exceeding 200. Since our scheme is very efficient, a small inner subspace is still viable. However, better solution times are obtained using a slighter stronger preconditioner. The solution times with $s = 10$ and $s = 20$ are similar and suggest that for computational problems with this discretization level, 10–20 is a reasonable range for this parameter. For

Table 3: Aerostructural solution comparison for nonlinear block Gauss–Seidel (NLBGS) with various under-relaxation strategies and coupled Newton–Krylov (CNK)

Load Factor	Solver	$\theta^{(0)}$	Aitken accel.	Inner size (s)	Outer size (m)	N_{iter}	Time (s)			
							CFD	Mesh	CSM	Total
1.0	NLBGS	0.25	N	—	—	49	384.4	6.2	23.0	395.6
1.0	NLBGS	0.50	N	—	—	22	218.7	2.8	20.3	223.7
1.0	NLBGS	0.75	N	—	—	16	190.3	2.0	19.7	193.9
1.0	NLBGS	1.00	N	—	—	22	216.8	2.8	21.8	221.9
1.0	NLBGS	0.50	Y	—	—	18	200.3	2.3	19.9	204.5
1.0	CNK	0.50	N	5	35	202	207.2	25.3	32.5	239.5
1.0	CNK	0.50	N	10	32	79	162.4	9.8	24.1	175.5
1.0	CNK	0.50	N	20	27	56	183.6	3.9	22.2	193.0
2.5	NLBGS	0.25	N	—	—	52	349.9	6.6	23.2	358.9
2.5	NLBGS	0.50	N	—	—	28	254.9	3.5	20.9	261.3
2.5	NLBGS	0.75	N	—	—	25	272.2	3.1	20.6	278.0
2.5	NLBGS	1.00	N	—	—	52	363.0	6.6	23.2	375.0
2.5	NLBGS	0.50	Y	—	—	25	263.1	3.1	20.6	268.8
2.5	CNK	0.50	N	5	35	225	234.8	28.2	34.8	270.8
2.5	CNK	0.50	N	10	32	118	214.6	14.7	26.8	233.8
2.5	CNK	0.50	N	20	27	55	193.9	6.8	22.3	203.2

the 1 g load case, the best CNK solution time is 9% faster than the best NLBGS time, and it is 22% faster for the 2.5 g load case. Not surprisingly, the advantage of a monolithic strategy is more pronounced in the more strongly coupled 2.5 g load condition.

We perform a similar comparison for the solution of the coupled adjoint system. For the LBGS solver, we consider two fixed under-relaxation parameter values, 0.5 and 1.0, as well as three relative tolerances for the aerodynamic adjoint solution, $\epsilon_A = 0.1, 0.25$ and 0.5 . The same three variants of the CNK solver are considered with inner aerodynamic subspace sizes of 5, 10, and 20. As with the nonlinear solution comparison, the memory requirements are fixed. The LBGS solver uses GMRES(75), while the coupled solver uses GMRES(s) for the aerodynamic preconditioner and FGMRES($\lfloor (75 - s)/2 \rfloor$) for the outer subspace. The results are given in Table 4.

The benefits of the monolithic solution method are even more compelling for the adjoint solution. As is the case for the nonlinear solution, all the Gauss–Seidel solutions are able to converge, albeit with some difficulty. For the 1 g flight condition, the under-relaxation parameter of 1.0 yields faster solution times, while for the 2.5 g case, a value of 0.5 is faster. Both cases converge faster with the relatively weak tolerances of only 0.5. Additional accuracy for each approximate solution is unnecessary and only increases the computational cost. The wide range of solution times for the LBGS method indicates a strong dependence on both the forcing tolerances and the under-relaxation factor. All three variants of the coupled Krylov solver converge without issue and are typically significantly faster than the best LBGS combination. The observation regarding the small inner subspace sizes ($s \approx 5$) made for the aerostructural solution applies to the adjoint solution to an even greater extent. While the reverse-mode mesh-deformation derivatives are computed efficiently, they are more costly than their forward equivalent.

The cumulative time required for the mesh-deformation derivatives represents a much larger proportion of the total solution time in all the coupled solution results. It represents approximately 40% of the overall time for the $s = 5$ case and approximately 15% for the $s = 20$ case. Values of $s = 10$ yielded the best solution times. However, the optimal values of s are problem dependent and may also depend on other solution parameters, such as the level of fill used for the aerodynamic preconditioner. Generally, s values ranging from 10 to 20 result in fast and robust convergence, a trend that is similar to the aerostructural solution case.

A comparison of the best LBGS solution times with the best CK times shows a 19% reduction in time for

Table 4: Adjoint solution method comparison

Load Factor	Solver	$\theta^{(0)}$	$\epsilon_{\mathcal{A}}$	Inner size (s)	Outer size (m)	N_{iter}	Time (s)			
							CFD	Mesh	CSM	Total
1.0	LBGS	0.5	0.10	—	—	21	542.6	9.1	2.1	554.0
1.0	LBGS	0.5	0.25	—	—	22	304.2	9.5	2.3	316.1
1.0	LBGS	0.5	0.50	—	—	23	141.7	10.0	2.3	154.1
1.0	LBGS	1.0	0.10	—	—	17	375.9	7.4	1.7	385.0
1.0	LBGS	1.0	0.25	—	—	16	158.9	7.0	1.6	167.6
1.0	LBGS	1.0	0.50	—	—	21	111.6	9.1	2.1	123.0
1.0	CK	—	—	5	35	120	78.2	53.4	10.6	136.3
1.0	CK	—	—	10	32	58	71.3	25.1	5.1	99.6
1.0	CK	—	—	20	27	40	99.8	17.3	3.5	118.5
2.5	LBGS	0.5	0.10	—	—	21	552.4	9.1	2.1	563.8
2.5	LBGS	0.5	0.25	—	—	21	311.1	9.2	2.1	322.5
2.5	LBGS	0.5	0.50	—	—	24	139.2	10.5	2.4	152.2
2.5	LBGS	1.0	0.10	—	—	79	4518.4	34.5	7.8	4561.3
2.5	LBGS	1.0	0.25	—	—	42	308.1	18.2	4.2	330.1
2.5	LBGS	1.0	0.50	—	—	46	175.9	20.0	4.5	200.8
2.5	CK	—	—	5	35	125	81.2	55.5	11.0	141.8
2.5	CK	—	—	10	32	53	77.6	27.4	5.6	107.4
2.5	CK	—	—	20	27	47	115.9	20.4	4.1	137.9

the 1 g load case and a 29% reduction for the 2.5 g case. We conclude that for the type of static aerostructural deformations we expect to find on a subsonic transport aircraft wing, the fully coupled solution methods (nonlinear and adjoint) can yield computational savings in the order of 10 to 30%.

4 Parallel Solution Scalability

We now examine the parallel scalability of the aerostructural solution and adjoint techniques. We consider a sequence of three discretization levels, where each level doubles the spatial resolution of the previous level. For the CFD solver, each level increases the number of cells by a factor of eight. Since the structural solver uses shell elements, doubling the spatial resolution results in four times the number of elements and approximately four times the number of structural degrees of freedom. Table 5 lists the sizes of each level of discretization and the number of processors used for each level.

This weak scaling study attempts to keep the number of degrees of freedom per processor roughly constant. For the CSM problem, near-perfect load balancing is achieved for each mesh level, but this is not the case for the CFD problem. The load imbalance is computed via $\text{Imbalance} = \max(N_i) / N_{\text{even}} - 1$, where N_i is the number of cells on each processor and $N_{\text{even}} = N_{\text{total}} / N_P$ is the number of cells resulting from a perfect balancing. The load imbalance for mesh level 3 indicates that at least one critical processor is assigned 25% more computational work than is required on average, leading to computation times 25% longer than would result from perfect load balancing.

Table 5: Mesh sizes for the three levels used in the parallel scalability study

Level	CFD				CSM			
	Cells	DOFs	N_P	Imbalance	Elements	Nodes	DOFs	N_P
1	263 762	1 318 810	8	1.5%	13 512	13 522	81 132	2
2	2 110 096	10 550 480	64	13.1%	54 048	54 030	324 180	8
3	16 288 0768	84 403 840	512	25.2%	216 192	216 118	1 296 708	32

Solutions are generated for a Mach number of 0.85, a fixed angle of attack of 2° , and a static pressure of 30 000 Pa, which is close to the 1 g cruise flight condition. The nonlinear equations are solved using the NLBGS method, while the adjoint equations are solved with the coupled Krylov method. The adjoint solution also evaluates the gradient of the lift with respect to 462 design variables. Both the nonlinear and adjoint solutions are converged to relative tolerance $\epsilon_{AS} = \epsilon_{SA} = 10^{-6}$. Figure 6 shows the C_p contours for each converged solution, as well as the von Mises stress values as a fraction of yield stress on the upper surface of the wing box. A section of the computational meshes near the wing root for each discipline is also provided for visual reference. Comparisons of key aerodynamic and structural performance results are given in Table 6.

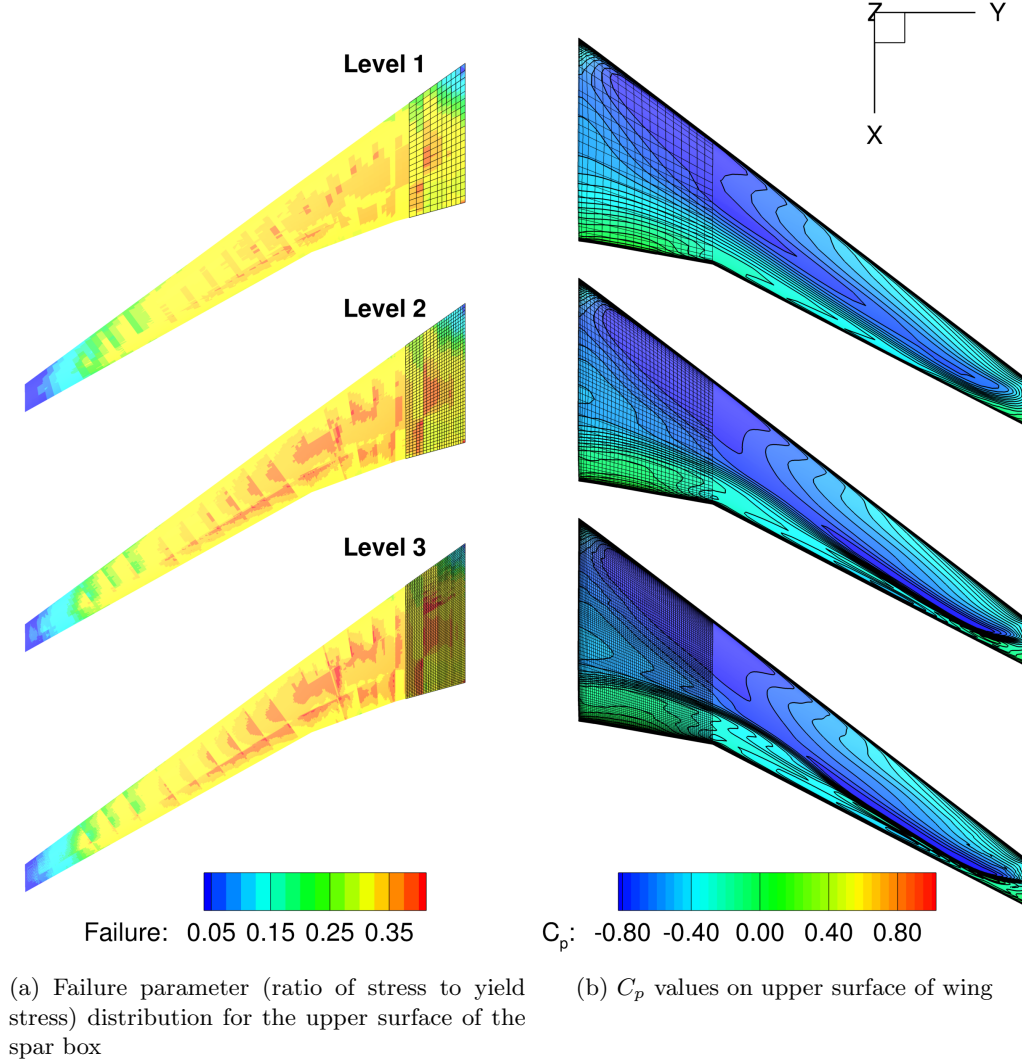


Figure 6: CSM and CFD solution comparison for each mesh level

The general characteristics of the solutions at each mesh level are similar, but a closer investigation reveals several differences. As the CFD mesh is refined, the normal shock on the upper surface of the wing becomes more pronounced, leading to a more accurate prediction of the wave drag. The lift coefficient is predicted reasonably well at all three mesh levels, while the drag coefficient reduces considerably as the mesh is refined because of the reduction of spurious drag caused by the artificial dissipation scheme. As expected from finite-element theory, the higher resolution mesh is less stiff, resulting in larger displacements

and higher stress levels.

We now examine the scalability of our methods by breaking down the time required to compute each solution and adjoint. These computations, as well as all the others in this paper, were performed using Intel Xeon E5540 processors connected with a 4x-DDR non-blocking InfiniBand fabric interconnect [55]. A detailed computational-time breakdown of the derivative computations performed for the above converged aerostructural analysis is given in Table 7. The fraction column is nondimensionalized using the aerostructural analysis time for that level.

The aerostructural solution times for all three mesh levels show that the majority of the solution time is taken by the CFD solution. The mesh deformation comprises only 2–3% of the total solution time. Similarly, the time required for transferring the loads and displacement is less than 1%.

We would expect the total solution time for an NLBGS solution to be the sum of the individual disciplinary solution times. However, since we are using a direct solution method for the CSM, the majority of the solution time is due to the matrix factorization, which we can overlap with the first CFD Gauss–Seidel iteration.

For the coupled adjoint solution, the one-time assembly costs account for the time required to compute the aerodynamic state residual matrix, $\partial\mathcal{A}/\partial w$, the spatial residual matrix, $\partial\mathcal{A}/\partial X_V$, and the two coupling matrices $\partial F_A/\partial w$ and $\partial F_A/\partial X_S$. This amounts to approximately 35% of the analysis time for the level-2 mesh, but it is quickly amortized over multiple adjoint solutions. Since the stiffness matrix for the structural discipline is symmetric, the original factorization for the structural solution method is reused for the structural adjoint. The bulk of the coupled adjoint solution time is spent in each of the disciplinary solvers, amounting to approximately 35% of the aerostructural solution time. The computational cost for the off-diagonal terms is dominated by the time required to compute $\partial\mathcal{A}/\partial u$, listed in Table 7 under “Mesh derivatives.” This amounts to 20–30% of the coupled adjoint solution time for the level-1 and level-2 meshes. The time required for the mesh derivatives on the larger mesh is disproportionately large. The culprit is the load balancing of the mesh-deformation algorithm. The mesh-deformation algorithm currently does not support block splitting, which means that once the number of processors exceeds the number of blocks in the mesh (144 in this case), the remaining processors are idle during mesh operations. If the level-3 mesh contained a correspondingly larger number of blocks, this behavior would not be observed. The total-derivative time includes the calculation of all partial-derivative terms in the total-derivative equation.

It is instructive to examine how the convergence characteristics of the nonlinear aerostructural solution and linear adjoint solutions change as the size of the computation increases by nearly two orders of magnitude. Figure 7 shows the nonlinear convergence and Fig. 8 shows the adjoint convergence.

For both the level-1 and level-2 meshes, convergence to 10^{-6} is achieved in approximately the same number of iterations (16 and 19, respectively), while engineering accuracy (10^{-3}) for the lift-to-drag ratio is achieved in approximately 10 iterations. However, the level-3 solution requires 36 iterations, and engineering accuracy is not achieved until iteration 20.

Moving from mesh level 2 to 3, the number of NLBGS iterations doubles, but the wall time increases by a factor of 4.3. Since a fixed aerodynamic forcing tolerance is used, each iteration is also more costly on the larger mesh. The performance for the coupled adjoint solution is similar. In this case, the cost of each iteration is similar for all three mesh levels, such that the number of iterations required for convergence in Fig. 8

Table 6: Aerostructural scaling results

Quantity	Level 1	Level 2	Level 3
C_L	0.4871	0.4960	0.4968
C_D	0.0186	0.0121	0.0109
C_M	−0.0250	−0.0224	−0.0216
Tip displacement (m)	1.54	1.61	1.62
Lower skin/stringer KS	0.328	0.365	0.491
Upper skin/stringer KS	0.459	0.492	0.723
Rib/spar KS	0.367	0.431	0.512
Nonlinear Gauss–Seidel iterations	16	19	36
Coupled Krylov adjoint iterations	24	46	117

Table 7: Coupled adjoint computational time comparison and breakdown

	Level 1		Level 2		Level 3	
	Time (s)	Fraction	Time (s)	Fraction	Time (s)	Fraction
Aerostructural solution	117.1	1.000	209.7	1.000	897.2	1.000
CFD solution	112.6	0.962	205.1	0.978	863.2	0.962
CSM solution	7.24	0.062	20.0	0.095	51.1	0.057
Mesh deformation	1.8	0.015	2.4	0.011	24.3	0.027
Coupled adjoint solution	53.2	0.454	112.7	0.537	579.6	0.646
Setup	42.9	0.366	71.3	0.340	52.6	0.059
CFD solution	43.7	0.371	90.1	0.430	241.3	0.269
CSM solution	0.9	0.008	4.0	0.019	19.3	0.022
Mesh derivatives	8.9	0.076	20.7	0.097	328.4	0.366
Total derivative	5.4	0.046	5.7	0.027	7.0	0.012
First gradient		0.823		0.913		0.716
Subsequent gradients		0.457		0.573		0.657

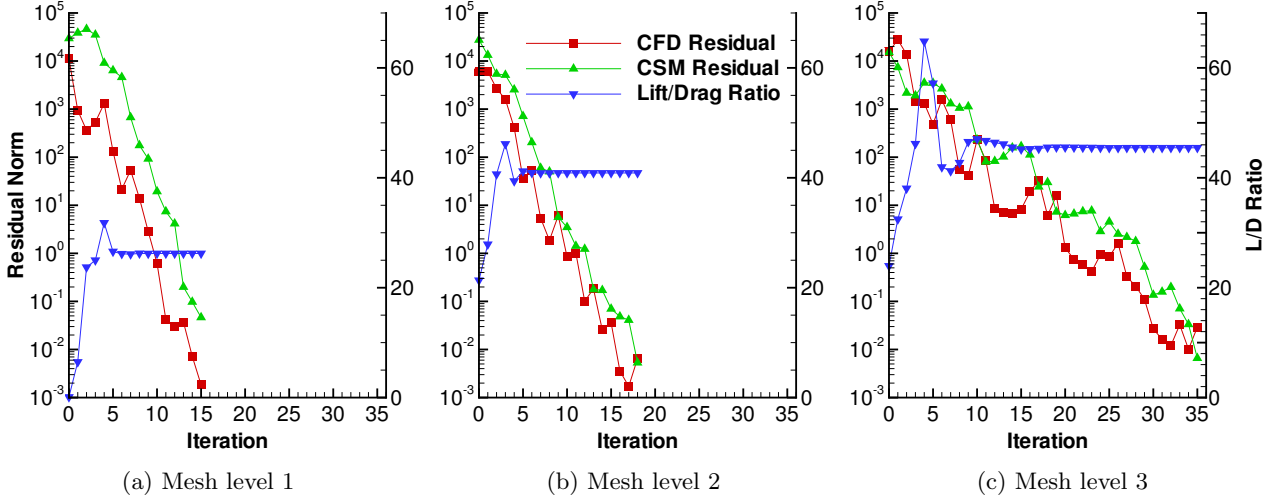


Figure 7: NLBGS convergence for the three mesh levels

is representative of the overall solution time. All of the adjoint solutions use the same aerodynamic preconditioning settings—ILU(1) and additive Schwartz(1)—resulting in nearly constant memory usage across the mesh levels. For mesh level 3, faster convergence times can be achieved by using stronger preconditioning, which reduces the condition number of the preconditioned system.

5 Design Variable Scalability

The main advantage of using the coupled adjoint method to compute the gradients of the functions of interest is that the computational cost is theoretically independent of the number of design variables. However, as described in Section 3.2, careful implementation of the partial-derivative terms $\partial I/\partial x$, $\partial A/\partial x$, $\partial S/\partial x$ in the total-derivative equation (15) is required to ensure that the computational cost is practically independent of the number of design variables.

We now consider the time required to compute the gradient of C_L with respect to thousands of design variables. The design variables are distributed according to Table 8 and contain both global geometric variables and local variables.

We compare the computational time required to compute the gradient for the coupled adjoint method and

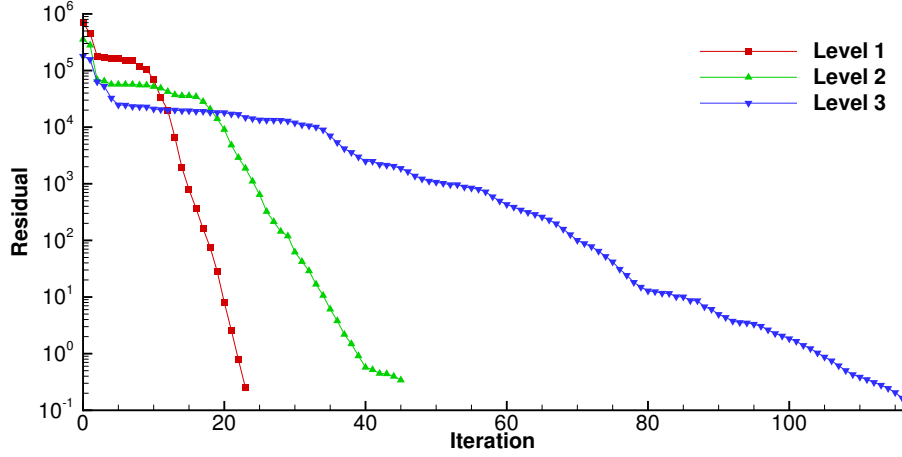


Figure 8: Convergence history for the lift adjoint using the coupled Krylov adjoint solver

Table 8: Design variables

Description	Quantity
<i>Global Variables</i>	
Span	1
Sweep	1
Chord	3
Twist	5
Shape	4818
<i>Aerodynamic variables</i>	
Angle of attack	1
Tail rotation	1
<i>Structural variables</i>	
Upper skin	54
Lower skin	54
Upper stringers	54
Lower stringers	54
Ribs	18
Rib stiffeners	18
Spars	36
<i>Total</i>	5120

for first-order finite differences. The level-2 discretization is used and the computational time is normalized by the time required for a single aerostructural solution. The results are shown in Fig. 9.

We expect the cost of finite differencing to be linearly dependent on the number of design variables. However, the slope is not equal to one but is significantly lower, since the solution for each design-variable perturbation uses the previous solution as a starting point, and it is closer to the converged state than a uniform-flow field solution. For each additional design variable, finite differencing requires a time equivalent

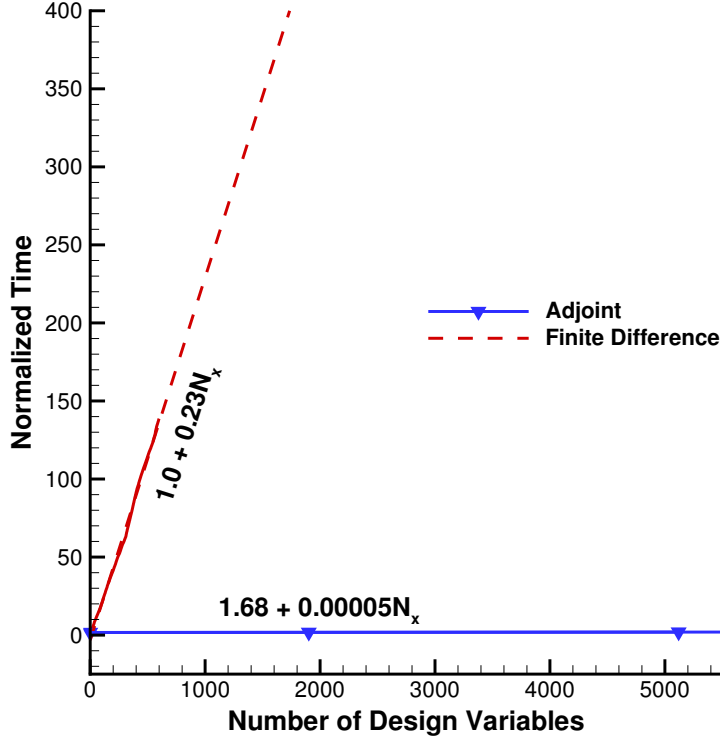


Figure 9: Gradient evaluation cost for first-order finite differencing and the coupled adjoint method versus number of design variables; one unit of normalized time corresponds to one aerostructural solution

to 23% of an aerostructural solution, resulting in a slope of 0.23.

The coupled adjoint method exhibits an extremely small slope. The main contributor to this slope is the design-variable-dependent load transfer, which requires a synchronous data transfer for each geometric design variable. Nevertheless, each additional design variable requires only 0.005% of the aerostructural solution time.

It is worth comparing the current results with the previous work of Martins *et al.* [26]. In that work, the coupled adjoint cost was found to scale with the number of design variables according to $3.4 + 0.01N_x$. Since the constant term in the equation includes the aerostructural solution, the coupled adjoint solution had a baseline cost of 2.4. The present method scales according to $1.67 + 5 \times 10^{-5}N_x$, as indicated in Fig. 9. This corresponds to a baseline cost for the coupled adjoint of 0.67, i.e., a 72% reduction relative to the previous implementation. This is primarily due to the elimination of the finite differencing that was used to compute the off-diagonal coupled adjoint terms. This improvement is even more significant in absolute terms because the aerostructural solution of the new implementation is also much more efficient. Additionally, the slope in the dependency on the number of design variables has been reduced by over two orders of magnitude. This is achieved by eliminating the use of finite-difference derivatives in the total-derivative equation (15).

We have shown that the new implementation of the coupled adjoint method exhibits extremely good design-variable scaling. The coupled computational cost can be considered practically independent of the number of design variables, and it is now feasible to compute coupled gradients with respect to thousands of design variables.

5 Conclusions

Strategies for the analysis and derivative computation of high-fidelity aerostructural systems have been presented. Two methods were implemented for solving the nonlinear aerostructural systems: a block Gauss–Seidel method with Aitken acceleration, and a fully coupled Newton–Krylov approach. Both methods performed well on the present problem of interest, with the latter method typically requiring 10% less

computational time than the former. With the proposed CNK approach, a typical aerostructural solution with 2×10^6 CFD cells and 300 000 structural degrees of freedom can be obtained in under 3 minutes using 52 aerodynamic and 4 structural processors.

A coupled adjoint was implemented for the aerostructural equations that significantly improves the accuracy and efficiency relative to previous work. By eliminating the computational expense of the finite differences previously used for the coupled adjoint off-diagonal terms, much larger coupled adjoint systems were solved in a scalable way. The dependency of the computational cost on the number of design variables was reduced to $\mathcal{O}(10^{-5})$, which can be considered negligible. A coupled adjoint gradient computation with respect to over 5 000 design variables was demonstrated. The elimination of finite differences dramatically improved the accuracy: when compared with the complex-step derivative approximation, an excellent agreement was demonstrated, with relative differences of $\mathcal{O}(10^{-5})$.

A traditional linear block Gauss–Seidel method was compared with a new fully coupled Krylov method implementation. The latter method was shown to be approximately 30% faster than the former, with similar memory requirements.

The parallel technique of the present study were demonstrated to scale to CFD problem with over 16×10^6 cells, CSM discretizations with over 1×10^6 degrees of freedom, and approximately 100 000 coupling variables using 512 processors.

The accuracy and scalability of the methods presented is the key ingredient for high-fidelity aerostructural optimization. Using these methods, it is possible to perform aerostructural optimizations of full aircraft configurations using detailed CFD and CSM models, leading to optimal aeroelastic tailoring. Such aerostructural optimizations are the focus of other papers by the authors [33, 37].

6 Acknowledgments

The authors are grateful for the funding provided by the Natural Sciences and Engineering Research Council. Computations were performed on the GPC supercomputer at the SciNet HPC Consortium. SciNet is funded by the Canada Foundation for Innovation under the auspices of Compute Canada; the Government of Ontario; Ontario Research Fund – Research Excellence; and the University of Toronto.

References

- [1] Pai, S. I. and Sears, W. R., “Some Aeroelastic Properties of Swept Wings,” *Journal of the Aeronautical Sciences*, Vol. 16, No. 2, 1949, pp. 105–115.
- [2] Skoog, R. and Brown, H., “A Method for the Determination of the Spanwise Load Distribution of a Flexible Swept Wing at Subsonic Speeds,” Tech. Rep. TN-2222, National Advisory Committee for Aeronautics, 1951.
- [3] Diederich, F. W., “Calculation of the Aerodynamic Loading of Flexible Wings of Arbitrary Planform and Stiffness,” Tech. rep., NACA, Langley, VA, 1949, TN 1876.
- [4] Ashley, H., “On Making Things the Best — Aeronautical Uses of Optimization,” *Journal of Aircraft*, Vol. 19, No. 1, Jan. 1982, pp. 5–28.
- [5] Sobieszcanski-Sobieski, J. and Haftka, R. T., “Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments,” *Structural Optimization*, Vol. 14, No. 1, 1997, pp. 1–23. doi:[10.1007/BF011](https://doi.org/10.1007/BF011).
- [6] Martins, J. R. R. A. and Lambe, A. B., “Multidisciplinary Design Optimization: A Survey of Architectures,” *AIAA Journal*, Vol. 51, No. 9, Sep 2013, pp. 2049–2075. doi:[10.2514/1.J051895](https://doi.org/10.2514/1.J051895).
- [7] Haftka, R. T., “Optimization of Flexible Wing Structures Subject to Strength and Induced Drag Constraints,” *AIAA Journal*, Vol. 14, No. 8, 1977, pp. 1106–1177. doi:[10.2514/3.7400](https://doi.org/10.2514/3.7400).
- [8] Grossman, B., Strauch, G., Eppard, W. H., Gürdal, Z., and Haftka, R. T., “Integrated Aerodynamic/Structural Design of a Sailplane Wing,” *Journal of Aircraft*, Vol. 25, No. 9, 1988, pp. 855–860. doi:[10.2514/3.45670](https://doi.org/10.2514/3.45670).

- [9] Grossman, B., Haftka, R. T., Kao, P.-J., Polen, D. M., and Rais-Rohani, M., “Integrated Aerodynamic-Structural Design of a Transport Wing,” *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 1050–1056. doi:[10.2514/3.45980](https://doi.org/10.2514/3.45980).
- [10] Chittick, I. R. and Martins, J. R. R. A., “An Asymmetric Suboptimization Approach to Aerostructural Optimization,” *Optimization and Engineering*, Vol. 10, No. 1, March 2009, pp. 133–152. doi:[10.1007/s11081-008-9046-2](https://doi.org/10.1007/s11081-008-9046-2).
- [11] Venkatamaran, S. and Haftka, R. T., “Structural optimization complexity: what has Moore’s law done for us?” *Structural and Multidisciplinary Optimization*, Vol. 28, 2004, pp. 375–387.
- [12] Jameson, A., “Aerodynamic Design via Control Theory,” *Journal of Scientific Computing*, Vol. 3, No. 3, Sept. 1988, pp. 233–260.
- [13] Nielsen, E. J. and Anderson, W. K., “Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations,” *AIAA Journal*, Vol. 37, No. 11, 1999, pp. 1411–1419.
- [14] Hicken, J. E. and Zingg, D. W., “Aerodynamic Optimization Algorithm with Integrated Geometry Parameterization and Mesh Movement,” *AIAA Journal*, Vol. 48, No. 2, Feb. 2009, pp. 400–413.
- [15] Mader, C. A. and Martins, J. R. R. A., “Stability-Constrained Aerodynamic Shape Optimization of Flying Wings,” *Journal of Aircraft*, Vol. 50, No. 5, September 2013, pp. 1431–1449. doi:[10.2514/1.C031956](https://doi.org/10.2514/1.C031956).
- [16] Lyu, Z., Kenway, G. K., Paige, C., and Martins, J. R. R. A., “Automatic Differentiation Adjoint of the Reynolds-Averaged Navier–Stokes Equations with a Turbulence Model,” *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, Jul 2013. doi:[10.2514/6.2013-2581](https://doi.org/10.2514/6.2013-2581).
- [17] Lyu, Z. and Martins, J. R. R. A., “RANS-based Aerodynamic Shape Optimization of a Blended-Wing-Body Aircraft,” *21st AIAA Computational Fluid Dynamics Conference*, San Diego, CA, Jul 2013. doi:[10.2514/6.2013-2586](https://doi.org/10.2514/6.2013-2586).
- [18] Brown, S. A., “Displacement Extrapolation for CFD+CSM Aeroelastic Analysis,” *Proceedings of the 35th AIAA Aerospace Sciences Meeting*, Reno, NV, 1997, AIAA 1997-1090.
- [19] Maman, N. and Farhat, C., “Matching Fluid and Structure Meshes for Aeroelastic Computations: A Parallel Approach,” *Computers and Structures*, Vol. 54, No. 4, 1995, pp. 779–785.
- [20] Farhat, C., Lesoinne, M., and LeTallec, P., “Load and Motion Transfer Algorithms for Fluid/Structure Interaction Problems with Non-Matching Discrete Interfaces: Momentum and Energy Conservation, Optimal Discretization and Application to Aeroelasticity,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 157, No. 1–2, 1996, pp. 95–115.
- [21] Kamakoti, R. and Shyy, W., “Fluid–Structure Interaction for Aeroelastic Applications,” *Progress in Aerospace Sciences*, Vol. 40, 2004, pp. 535–558.
- [22] Felippa, C. A., Park, K., and Farhat, C., “Partitioned Analysis of Coupled Mechanical Systems,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2001, pp. 3247–3270.
- [23] Kim, J. Y., Aluru, N. R., and Tortorelli, D. A., “Improved Multi-Level Newton Solvers for Fully Coupled Multi-Physics Problems,” *International Journal for Numerical Methods in Engineering*, Vol. 58, 2003, pp. 463–480. doi:[10.1002/nme.751](https://doi.org/10.1002/nme.751).
- [24] Maute, K., Nikbay, M., and Farhat, C., “Coupled Analytical Sensitivity Analysis and Optimization of Three-Dimensional Nonlinear Aeroelastic Systems,” *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2051–2061.
- [25] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., “Aero-Structural Wing Design Optimization Using High-Fidelity Sensitivity Analysis,” *Proceedings of the CEAS Conference on Multidisciplinary Aircraft Design and Optimization*, edited by H. Höllinger, Köln, Germany, June 2001, pp. 211–226.

- [26] 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, No. 1, March 2005, pp. 33–62. doi:[10.1023/B:OPTE.0000048536.47956.62](https://doi.org/10.1023/B:OPTE.0000048536.47956.62).
- [27] Martins, J. R. R. A., Alonso, J. J., and Reuther, J. J., “High-Fidelity Aero-Structural Design Optimization of a Supersonic Business Jet,” *Journal of Aircraft*, Vol. 41, No. 3, 2004, pp. 523–530. doi:[10.2514/1.11478](https://doi.org/10.2514/1.11478).
- [28] Maute, K., Nikbay, M., and Farhat, C., “Sensitivity Analysis and Design Optimization of Three-Dimensional Non-Linear Aeroelastic Systems by the Adjoint Method,” *International Journal for Numerical Methods in Engineering*, Vol. 56, No. 6, 2003, pp. 911–933.
- [29] Barcelos, M., Bavestrello, H., and Maute, K., “A Schur-Newton-Krylov Solver for Steady-State Aeroelastic Analysis and Design Sensitivity Analysis,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 195, No. 17-18, 2006, pp. 2050–2069. doi:[10.1016/j.cma.2004.09.013](https://doi.org/10.1016/j.cma.2004.09.013).
- [30] Barcelos, M. and Maute, K., “Aeroelastic Design Optimization for Laminar and Turbulent Flows,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, No. 19–20, 2008, pp. 1813–1832. doi:[10.1016/j.cma.2007.03.009](https://doi.org/10.1016/j.cma.2007.03.009).
- [31] Brezellion, J., Ronzheimer, A., Haar, D., Abu-Zurayk, M., Lummer, K., Krug er, W., and Nattere, F. J., “Development and Application of Multi-Disciplinary Optimization Capabilities Based on High-Fidelity Methods,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2012, AIAA 2012–1757.
- [32] Ghazlane, I., Carrier, G., Dumont, A., and D esid eri, J.-A., “Aerostructural adjoint method for flexible wing optimization,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2012, AIAA 2012–1924.
- [33] Kenway, G. K. W. and Martins, J. R. R. A., “Multi-Point High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration,” *Journal of Aircraft*, 2013. doi:[10.2514/1.C032150](https://doi.org/10.2514/1.C032150), (In press).
- [34] Alonso, J. J., LeGresley, P., van der Weide, E., Martins, J. R. R. A., and Reuther, J. J., “pyMDO: A Framework for High-Fidelity Multi-Disciplinary Optimization,” *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, September 2004, AIAA 2004–4480.
- [35] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and van der Weide, E., “ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers,” *AIAA Journal*, Vol. 46, No. 4, April 2008, pp. 863–873. doi:[10.2514/1.29123](https://doi.org/10.2514/1.29123).
- [36] 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 and Optimization Conference*, Fort Worth, Texas, September 2010.
- [37] Liem, R. P., Kenway, G. K., and Martins, J. R. R. A., “Multi-point, multi-mission, high-fidelity aerostructural optimization of a long-range aircraft configuration,” *Proceedings of the 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Indianapolis, IN, Sept. 2012. doi:[10.2514/6.2012-5706](https://doi.org/10.2514/6.2012-5706).
- [38] van der Weide, E., Kalitzin, G., Schluter, J., and Alonso, J. J., “Unsteady Turbomachinery Computations Using Massively Parallel Platforms,” *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2006, AIAA 2006–0421.
- [39] Kenway, G. K., Kennedy, G. J., and Martins, J. R. R. A., “A Scalable Parallel Approach for High-Fidelity Aerostructural Analysis and Optimization,” *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2012, AIAA 2012–1922.

- [40] Saad, Y. and Schultz, M. H., “GMRES: A Generalized Minimal Residual Algorithm for Solving Non-symmetric Linear Systems,” *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. doi:[10.1137/0907058](https://doi.org/10.1137/0907058).
- [41] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “A CAD-Free Approach to High-Fidelity Aerostructural Optimization,” *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Fort Worth, TX, September 2010, AIAA–2010–9231.
- [42] Irons, B. M. and Tuck, R. C., “A Version of the Aitken Accelerator for Computer Iteration,” *International Journal for Numerical Methods in Engineering*, Vol. 1, No. 3, 1969, pp. 275–277.
- [43] Eisenstat, S. C. and Walker, H. F., “Choosing the Forcing Terms in an Inexact Newton Method,” *SIAM J. Sci. Comput.*, Vol. 17, No. 1, 1996, pp. 16–32.
- [44] Martins, J. R. R. A. and Hwang, J. T., “Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models,” *AIAA Journal*, Vol. 51, No. 11, 2013, pp. 2582–2599. doi:[10.2514/1.J052184](https://doi.org/10.2514/1.J052184).
- [45] Mader, C. A., Martins, J. R. R. A., Alonso, J. J., and van der Weide, E., “ADjoint: An Approach for the Rapid Development of Discrete Adjoint Solvers,” *AIAA Journal*, Vol. 46, No. 4, April 2008, pp. 863–873. doi:[10.2514/1.29123](https://doi.org/10.2514/1.29123).
- [46] Hascoët, L. and Pascual, V., “TAPENADE 2.1 User’s Guide,” Technical report 300, INRIA, 2004.
- [47] Li, X. S. and Demmel, J. W., “SuperLU DIST: A Scalable Distributed-Memory Sparse Direct Solver for Unsymmetric Linear Systems,” *ACM Trans. Mathematical Software*, Vol. 29, No. 2, 2003, pp. 110–140.
- [48] Wrenn, G., “An Indirect Method for Numerical Optimization Using the Kreisselmeier-Steinhauser Function,” NASA Technical Report CR-4220, NASA, 1989.
- [49] Poon, N. M. K. and Martins, J. R. R. A., “An Adaptive Approach to Constraint Aggregation Using Adjoint Sensitivity Analysis,” *Structures and Multidisciplinary Optimization*, Vol. 30, No. 1, 2007, pp. 61–73. doi:[10.1007/s00158-006-0061-7](https://doi.org/10.1007/s00158-006-0061-7).
- [50] Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., “Development of a Common Research Model for Applied CFD Validation Studies,” *26th AIAA Applied Aerodynamics Conference*, AIAA, Honolulu, Hawaii, August 2008.
- [51] Dvorkin, E. N. and Bathe, K.-J., “A Continuum Mechanics Based Four-Node Shell Element for General Nonlinear Analysis,” *Engineering Computations*, Vol. 1, No. 1, 1984, pp. 77–88.
- [52] Squire, W. and Trapp, G., “Using Complex Variables to Estimate Derivatives of Real Functions,” *SIAM Rev.*, Vol. 40, No. 1, March 1998, pp. 110–112. doi:[10.1137/S003614459631241X](https://doi.org/10.1137/S003614459631241X).
- [53] 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, 2003, pp. 245–262. doi:[10.1145/838250.838251](https://doi.org/10.1145/838250.838251).
- [54] Smith, B., Bjørstad, P., and Gropp, W., *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 2004.
- [55] Loken, C., Gruner, D., Groer, L., Peltier, R., Bunn, N., Craig, M., Henriques, T., Dempsey, J., Yu, C.-H., Chen, J., Dursi, L. J., Chong, J., Northrup, S., Pinto, J., Knecht, N., and Zon, R. V., “SciNet: Lessons Learned from Building a Power-Efficient Top-20 System and Data Centre,” *Journal of Physics: Conference Series*, Vol. 256, No. 1, 2010, pp. 012026.